SAS® Viya® 3.4 for Linux: Deployment Guide
## Contents

**Chapter 1 / Introduction**  
Steps for a Successful Deployment ........................................... 1  
How Deployment Works ......................................................... 3  
Deployment Examples and Guidance ........................................ 5  
SAS Products and Supporting Components ................................. 18  
Contact SAS Technical Support .............................................. 19

**Chapter 2 / System Requirements** ......................................... 21  
Hardware Requirements ......................................................... 22  
Operating System Requirements ............................................ 28  
Server Software Requirements ............................................... 32  
Data Source and Storage Requirements ................................... 33  
Security Requirements .......................................................... 40  
User and Group Requirements ............................................... 43  
Client Requirements .............................................................. 52  
Deployment Tools .................................................................... 53

**Chapter 3 / Pre-installation Tasks** ........................................ 55  
Create a Mirror Repository ..................................................... 56  
Create a Playbook ................................................................ 58  
Enable Required Ports ............................................................ 60  
Configure SELinux .................................................................. 63  
Configure a Proxy Server ......................................................... 64  
Enable the Yum Cache ............................................................. 65  
Enable a Shared File System ..................................................... 65  
Perform Linux Tuning ............................................................... 66  
Confirm the Identities of the Hosts ............................................ 69  
(Optional) Enable Key-Based SSH Authentication ....................... 69  
Install Ansible ....................................................................... 70

**Chapter 4 / Installation** ......................................................... 73  
Edit the Inventory File ............................................................. 74  
Modify the vars.yml File .......................................................... 81  
Modify the sitedefault.yml File ............................................... 112  
SAS Viya and Multi-tenancy ..................................................... 112  
Deploy the Software ............................................................... 115  
Install with SAS 9.4 Software ................................................. 117  
Deployment Logs .................................................................... 117

**Chapter 5 / Post-installation Tasks** ........................................ 119  
Configure Security ................................................................. 120  
Configure the Machines ........................................................... 125  
Configure Data Access ............................................................. 131  
Configure Model Access .......................................................... 147  
Configure SAS Analytics for IoT .............................................. 147  
Configure SAS Data Preparation and SAS Data Agent ............... 148  
Configure SAS Data Quality .................................................... 148  
Configure SAS Data Science .................................................... 149  
Configure SAS Decision Manager .......................................... 149  
Configure SAS Event Stream Processing ................................. 149
Appendix 1

Chapter 7 / Completing the Deployment

Configure SAS Intelligent Decisioning .................................................. 155
Configure SAS Model Manager ............................................................. 155

Chapter 6 / Validating the Deployment .................................................. 157
Verify Access to SAS Logon and SAS Drive ........................................... 158
Log On to Your Version of SAS Studio ................................................ 158
SAS Viya File Locations ....................................................................... 159
Perform Installation Qualification on RPM Packages .............................. 160
Access CAS Server Monitor ................................................................. 162
Access SAS Environment Manager ...................................................... 163
Verify SAS Message Broker ............................................................... 163
Verify SAS Infrastructure Data Server ............................................... 163
Verify Elasticsearch ............................................................................ 164
Verify SAS Event Stream Manager Status ......................................... 165
Overview of Data Access Verification .................................................. 165
Verify SAS/ACCESS Interface to Amazon Redshift ................................. 166
Verify SAS/ACCESS Interface to DB2 .................................................. 167
Verify SAS/ACCESS Interface to Google BigQuery ............................... 168
Verify SAS/ACCESS Interface to Greenplum ...................................... 168
Verify SAS/ACCESS Interface to HAWQ ............................................. 168
Verify SAS/ACCESS Interface to Impala ............................................. 169
Verify SAS/ACCESS Interface to JDBC .............................................. 169
Verify SAS/ACCESS Interface to Microsoft SQL Server ......................... 170
Verify SAS/ACCESS Interface to MySQL ............................................ 170
Verify SAS/ACCESS Interface to Netezza .......................................... 171
Verify SAS/ACCESS Interface to ODBC ............................................ 171
Verify SAS/ACCESS Interface to Oracle ............................................ 172
Verify SAS/ACCESS Interface to PostgreSQL .................................... 172
Verify SAS/ACCESS Interface to SAP ASE ........................................ 173
Verify SAS/ACCESS Interface to SAP HANA ..................................... 173
Verify SAS/ACCESS Interface to SAP R/3 ......................................... 173
Verify SAS/ACCESS Interface to Snowflake ..................................... 174
Verify SAS/ACCESS Interface to Teradata ........................................ 174
Verify SAS/ACCESS Interface to Vertica .......................................... 175

Chapter 7 / Completing the Deployment ................................................. 177
Save Snapshot Directory Content ....................................................... 177
Share Important Deployment Information with the Administrators ..... 177
Next Steps for SAS Event Stream Processing Users ............................ 178
Refer to Additional Documentation .................................................... 178

Chapter 8 / Managing Your Software .................................................... 179
Overview ......................................................................................... 180
Updating Your SAS Viya Software ..................................................... 181
Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4 191
Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya .......... 197

Chapter 9 / Uninstalling SAS Viya .......................................................... 223
What deploy-cleanup Does ............................................................... 223
Use deploy-cleanup ........................................................................... 223
Uninstall SAS Embedded Process ..................................................... 225
Uninstall SASHDAT Plug-ins ............................................................. 225
Uninstall a Mirror Repository .......................................................... 225

Appendix 1 / Creating High Availability PostgreSQL Clusters .......... 227
Overview ......................................................................................... 227
Appendix 2 / Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop .............................. 239
   Supported Hadoop Distributions ........................................ 239
   Deployment Tasks for Hive and Spark Access .......................... 239
   Pre-deployment Hadoop and Spark Tasks .............................. 240
   Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop .......................... 241
   Verify SAS/ACCESS Interface to Spark ................................ 247
Appendix 3 / Hadoop Deployment: Configuring SAS In-Database Technologies ......................................................... 249
   Hadoop Prerequisites .................................................... 249
   Overview of the In-Database Deployment Package for Hadoop ...... 250
   SAS Embedded Process for SAS 9.4: Install and Deploy on SAS Viya ......................................................... 250
   (Optional) Deploy TLS Certificates .................................... 250
   Verify SAS Data Connect Accelerator for Hadoop ................. 251
   Verify SAS Data Connect Accelerator for Spark .................... 251
   Add the YARN Application CLASSPATH for MapR ................ 252
   Performance Tuning for the SAS Embedded Process ................ 253
Appendix 4 / Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS .......................................................... 257
   About CAS SASHDAT Access to HDFS ................................. 257
   Supported Hadoop Distributions ........................................ 258
   Overview of Deployment Tasks for HDFS for Existing Hadoop Clusters ......................................................... 258
   Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters ......................................................... 258
   Review the Passwordless Secure Shell Requirements ............... 259
   Kerberos Requirements ................................................. 259
   Deploying SAS Plug-ins for Hadoop ................................... 262
   Uninstalling SAS Plug-ins for Hadoop ................................. 268
   sashdat-install.sh Reference ............................................ 271
Appendix 5 / Teradata Deployment: Configuring SAS In-Database Technologies .......................................................... 275
   Prerequisites ..................................................................... 275
   Connections from SAS 9.4 Clients ....................................... 276
   Installing the SAS In-Database Deployment Package for Teradata ......................................................... 276
   Configuring SAS Data Quality Accelerator for Teradata ........... 278
Appendix 6 / Troubleshooting ...................................................... 285
   Nothing to Do Dialog ...................................................... 285
   PCA and KCLUS Procedures Were Not Found ......................... 286
   Timeout Dialog ............................................................. 286
   From Any Browser: Connection Is Not Private ....................... 287
   From Google Chrome: Connection Is Not Private ................... 287
   Unable to Read a Key ....................................................... 287
   After an Upgrade, One or More of the RabbitMQ Nodes Fails to Start ......................................................... 288
   CAS Controller Start-Up Fails ............................................. 289
   Invalid Value for Internal Host Name .................................... 289
   Project Creation Failed .................................................... 290
   Zypper Run Command Failed: Library Is Locked ...................... 290
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zypper Run Command Failed: Shared Vault Value Has Not Been Set</td>
<td>290</td>
</tr>
<tr>
<td>Deployment of More Than Five Machines Takes More Time</td>
<td>291</td>
</tr>
<tr>
<td>Connection Reset by Peer Network Problem</td>
<td>291</td>
</tr>
<tr>
<td>Internet Connectivity Problems</td>
<td>292</td>
</tr>
<tr>
<td>After an Upgrade, the Default Backup Schedule Is Not Working</td>
<td>294</td>
</tr>
<tr>
<td>Setup Failed Because of Symbolic Link Problem with RabbitMQ</td>
<td>294</td>
</tr>
<tr>
<td>After an Upgrade, an Additional Instance of Folder Services Might Exist</td>
<td>294</td>
</tr>
</tbody>
</table>
Steps for a Successful Deployment

Before You Begin

- Make sure that this guide is the correct one for the software that you ordered.
  
  ☑ If you accessed this guide directly from the Software Order Email (SOE), you are viewing the correct guide.
  
  ☑ If you did not access this guide directly from the SOE, check the date that you received the SOE from SAS to make sure that you have the correct guide:
    
    - If the date is August 20, 2019 or later, use this guide to deploy the software.
If the date is before August 20, 2019, see SAS Viya 3.4 for Linux: Deployment Guide (Before August 20, 2019) to deploy the software.

Note: After the software is ordered, SAS sends the SOE to your business or organization. The SOE contains a list of the software that you licensed, attachments that are required for deployment, and a link to the correct deployment guide.

If you are viewing a saved copy of a PDF of this guide, the content might be outdated because this guide is subject to continual updates. You can always view the latest SAS Viya deployment guides at the SAS Viya Install Center.

To use this guide successfully, you should have a working knowledge of Ansible and the Linux operating system and familiarity with the deployment tools that are provided by SAS. Also, you should understand the deployment types (full and programming-only) and that some products require that you perform either a programming-only or full deployment. For more information, see “How Deployment Works” on page 3.

In some cases, you must use different guides to deploy the software.

- You can create a Docker container that includes a programming-only deployment of the software. For more information, see SAS Viya for Containers: Deployment Guide.

- If your order includes SAS Data Preparation, SAS Data Agent is also included. You can deploy SAS Data Agent to enable data to move from an on-premises environment to SAS Data Preparation that is running in a private cloud or a public cloud. For more information, see SAS Data Agent on Linux: Deployment Guide.

  Note: Although SAS Data Agent is included in an order that includes SAS Data Preparation, it does not appear in the list of software in the SOE.

After you deploy the software, you can migrate resources from SAS 9 or earlier versions of SAS Viya. For information about what resources are supported for migration, see SAS Viya Administration: Promotion: Overview.

  Note: Migration is not supported for a programming-only deployment of SAS Viya.

### Step 1 — Prepare for the Deployment

1. Perform one of the following tasks:
   - To update, upgrade, or add software to an existing deployment, go directly to “Managing Your Software” on page 179.

     Note: To assist with an upgrade, additional Ansible playbooks are available in the SAS Viya Administration Resource Kit (SAS Viya ARK) on GitHub. For more information, see “(Optional) Using the SAS Viya Administration Resource Kit” on page 5.

   - To deploy a new instance of the software, continue with the following the steps.

2. Go to “System Requirements” on page 21 to learn about requirements for hardware, software, data sources, storage, users and groups, security, and clients.

   Note: To review the possible topologies, see “Deployment Examples and Guidance” on page 5.

3. Go to “Pre-installation Tasks” on page 55 to prepare your environment before you deploy the software.

### Step 2 — Perform the Deployment

1. Go to “Installation” on page 73 to deploy the software to a single machine or across multiple machines. The steps for running the playbook to install and configure the SAS Viya applications and the SAS/ACCESS software are included in this section.
Step 3 — Validate and Complete the Deployment

1. Go to “Validating the Deployment” on page 157 to log on to SAS Studio and perform validation tasks, such as verifying that the servers were deployed correctly and that SAS can access your data.

2. Go to “Completing the Deployment” on page 177 to learn about post-deployment best practices and to find documentation about administrative tasks.

Step 4 — Configure Access to Hadoop or Teradata

If you deployed SAS to access data in Hadoop or Teradata, the following tasks are most likely performed by a Hadoop or Teradata administrator, respectively.

- **Hadoop:**
  - To access data in Hive, see “Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop” on page 239.
  - To configure SAS Embedded Process with Hadoop, see “Hadoop Deployment: Configuring SAS In-Database Technologies” on page 249.
  - To configure CAS SASHDAT Access to HDFS, see “Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS” on page 257.

- **Teradata:** to configure SAS Embedded Process with Teradata, see “Teradata Deployment: Configuring SAS In-Database Technologies” on page 275.

Note: To find out more about deployment scenarios with Hadoop or Teradata, see “Deployment Examples and Guidance” on page 5.

How Deployment Works

Using Ansible to Deploy SAS Viya

The Basics

Ansible is used to deploy SAS Viya to one or multiple machines. Using Ansible gives you the most control over the deployment process.

- Ansible is configuration management software that provides a straightforward approach to deploying SAS Viya. To deploy using Ansible, you customize files for your environment, and then you run a command to deploy software according to the values in those files. The set of files, known collectively as “the playbook,” provides the instructions about what software is deployed on which machines. In this guide, “run the playbook” means to deploy or update SAS Viya software.

- The playbook that you run must first be customized for your order. You will use the SAS Orchestration Command Line Interface (CLI) to create the customized playbook. The instructions for downloading the SAS Orchestration CLI and for creating a playbook are provided in this guide. Also, the Software Order Email (SOE) that SAS sends to your business or organization contains a file attachment that is required in order to create the playbook. The file attachment in the SOE contains information that is specific to your order.
During the deployment process, the software to which you are entitled is downloaded from repositories that are maintained by SAS or from mirror repositories at your own site. Creating mirror repositories before running the playbook is optional for deployments on Red Hat Enterprise Linux, but is required for deployments on SUSE Linux. The instructions for using SAS Mirror Manager to create mirror repositories are provided in this guide.

Each time you run the playbook, Ansible automates a series of commands that securely access the latest SAS Viya software to which you are entitled.

To use Ansible, you must install it first. In this guide, the machine on which you install Ansible is called the “Ansible controller.” The Ansible controller must have SSH access to the machines on which you plan to deploy SAS Viya.

Files Used for Deployment

The following files are used to deploy SAS Viya using Ansible. Before you run the playbook, you will edit the files to specify the machines on which to deploy the software, which software to deploy, and site-specific configuration settings. Also, each filename is a reserved name that is required for running your playbook. Therefore, when you edit the file, be sure to save as the filename that is shown.

<table>
<thead>
<tr>
<th>File</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>inventory.ini</td>
<td>The inventory file that is used to deploy SAS Viya. You edit the inventory.ini file to map machines (or hosts) to the SAS Viya software components, which are represented as host groups within the inventory.ini file.</td>
</tr>
<tr>
<td>vars.yml</td>
<td>The vars.yml file includes the variables that enable you to customize your deployment. For example, you edit the vars.yml file to configure a data connector, to manage passwordless SSH settings, and so on.</td>
</tr>
<tr>
<td>sitedefault.yml (optional)</td>
<td>Typically, the sitedefault.xml is not used for the initial deployment. The sitedefault.yml file contains variables for more advanced implementations, such as setting up a high availability PostgreSQL cluster and enabling SAS Viya to run in a multi-tenancy mode.</td>
</tr>
</tbody>
</table>

Note: SAS provides sample files (templates) in your playbook, which you can repurpose for your deployment. After you create your playbook, look for these files in the sas_viya_playbook/samples subdirectory.

Deployment Types

By default, the playbook that is generated with the SAS Orchestration CLI enables you to deploy the full suite of products and user interfaces that you ordered. In the SAS documentation, this type of deployment is referred to as a “full deployment.”

Although SAS recommends a full deployment, you can generate a playbook that is used to perform a “programming-only deployment.” A programming-only deployment supports data scientists and programmers who use SAS Studio or direct programming interfaces such as Python or REST APIs. Understand that this type of deployment does not include SAS Drive, SAS Environment Manager, and the complete suite of services that are included with a full deployment. Also, a programming-only deployment does not support multi-tenancy. Therefore, make sure that you are providing your users with the features that they require.

Your version of SAS Studio depends on which type of deployment you perform. If you deploy a full environment, your environment contains both SAS Studio 5.1 and SAS Studio 4.4. By default, users will log on to SAS Studio 5.1. If you deploy a programming-only environment, your environment contains SAS Studio 4.4.

Note: To deploy SAS Data Science, perform a programming-only deployment. A full deployment does not support SAS Data Science.
Note: To deploy SAS Event Stream Manager or SAS Analytics for IoT, perform a full deployment. A programming-only deployment does not support SAS Event Stream Manager or SAS Analytics for IoT.

(Optional) Using the SAS Viya Administration Resource Kit

The SAS Viya Administration Resource Kit (SAS Viya ARK) provides additional Ansible playbooks and other tools that can assist in a deployment or upgrade. For example, playbooks are available to help prepare your environment before a deployment or upgrade or to complete required steps after a deployment or upgrade. Because playbooks automate a series of manual steps, using them can save time.

To learn more, see the SAS Viya Administration Resource Kit (SAS Viya ARK) on GitHub. See the README for each playbook or tool to understand what is supported and how and when to use it.

How a SAS Viya Deployment Differs from a SAS 9 Deployment

Besides the use of Ansible, the SAS Viya deployment differs from a SAS 9 deployment in the following ways:

- The SAS Deployment Wizard and the SAS Deployment Manager that support SAS 9.4 are not used to install and configure SAS Viya.
- Because the deployment model works with repositories that are native to your operating system, a SAS Software Depot is not required for your SAS Viya software.
- No SAS deployment tools must be installed on target machines. All deployment actions can be remotely executed from the Ansible controller.
- SAS/ACCESS software can be configured during installation.

Deployment Examples and Guidance

About the Deployment Examples

The deployment examples provide high-level guidance for deploying SAS Viya software. Refer to the following examples for topologies that best fit your environment and users and for tips on using the documentation.

When reviewing the examples, consider the following:

- Ansible is used to deploy SAS Viya. A benefit of using Ansible is that you can assign host groups to different machines, which can allow for dedicated servers and clustering.
- The inventory.ini file is shown in the examples. To illustrate collections of the host groups that are provided in the inventory.ini file, the following categories of software are depicted.
  - “Service Layer” represents the numerous features for the visual interfaces, advanced analytics, text analytics, data management and mining, and more. Also, the service layer includes the stateful services for SAS Viya, which include the following:
    - SAS Infrastructure Data Server, which is used to store user content. SAS Infrastructure Data Server is based on PostgreSQL.
    - SAS Message Broker, which is based on Pivotal’s messaging broker, RabbitMQ.
    - SAS Configuration Server, which uses HashiCorp’s Consul as a service configuration registry that serves as a central repository for configuration data, service discovery, and health status.
SAS Secrets Manager, which is based on HashiCorp Vault. SAS Secrets Manager uses Vault to store and generate secrets such as Transport Layer Security (TLS) certificates.

- An Apache HTTP Server, which serves static HTML content and proxies client connections.


- "CAS Server" provides the run-time environment where data management and analytics take place. The CAS server can be deployed in a distributed fashion across multiple machines or on a single machine.

- Distributing the CAS server across multiple machines allows for massively parallel processing (MPP) by users. An advantage to MPP is that, whenever possible, data is loaded into memory in parallel, which can result in faster load times. Also, the distributed CAS server can be configured for fault tolerance. For example, if a CAS worker fails, another CAS worker can use a redundant copy of the data in order to complete data analysis.

- Deploying the CAS server on a single machine allows for symmetric multi-processing (SMP) by users. A single-machine CAS server performs serial loads of data into memory from a supported data source. The in-memory analytic features of a distributed server are available to the single-machine CAS server.

- Data connectors, which are part of SAS/ACCESS software, must be deployed to one or more machines on which CAS is running. For scenarios in which CAS is deployed to multiple machines, data connectors are deployed to the CAS controller and to each CAS worker.

- A mirror repository is not shown in each example, but creating one might be part of your deployment. For example, a mirror repository is required for all deployments on SUSE Linux. For deployments on Red Hat Enterprise Linux, creating a mirror repository is optional. Also, if you plan to deploy multiple CAS servers or a multi-tenant environment, creating a mirror repository is highly recommended.

- Before you begin the deployment process, refer to the system requirements to understand hardware and software requirements and recommendations.

**Single Machine Deployment**

You can deploy the software, including the CAS server, to a single machine. This type of deployment can be useful for demonstration purposes or for deploying software for a specific group of users. For a single-machine deployment, the CAS server is deployed to support analytics and data-management processing in symmetric multiprocessing (SMP) mode. You can deploy the software to the same machine where Ansible is running, or you can deploy to a different, target machine.

The following example shows all software, including the CAS server, installed on one machine.

*Figure 1.1  SAS Viya on One Machine Using Ansible*

Note: Because the CAS server is deployed to one machine, the secondary CAS controller and CAS workers are not deployed.
Full Deployment with a Separate, Single-Machine CAS Server

In this section, Ansible is used to perform a full deployment. Also, a single-machine CAS server is deployed to support analytics and data-management processing in symmetric multiprocessing (SMP) mode.

When reviewing the examples, consider the following:

- A full deployment includes the service layer and the programming run time.
- To deploy the CAS server, the primary CAS controller is deployed to its own machine, and data connectors are shown as configured for use. A best practice is to configure the data connectors for use in the vars.yml file before running the playbook. Because a single-machine CAS server is deployed, the secondary CAS controller and the CAS workers are not deployed.

The following example shows a full deployment with single-machine CAS server and a separate machine for the service layer and the programming run time.

*Figure 1.2  SAS Viya on Two Machines*

The following example shows separate machines for the service layer and the programming run time.

*Figure 1.3  SAS Viya on Three Machines*
Full Deployment with a Distributed CAS Server

Ansible is used to deploy the service layer and the programming run time across two machines, and the CAS server is distributed across multiple machines. An advantage to deploying a distributed CAS server is that optimal processing can be achieved through massively parallel processing (MPP) for multiple users.

Consider the following:

- A full deployment includes the service layer and the programming run time.
- To deploy the CAS server, different host groups that are related to CAS are assigned to machines in the inventory.ini file before you run the playbook. The primary CAS controller, the secondary CAS controller, and each CAS worker are all deployed to different machines. Data connectors are shown as configured for use on each machine where a CAS controller or a worker is deployed. A best practice is to configure the data connector for use in the vars.yml file before you run the playbook.

**Note:** The secondary CAS controller, which acts as a backup controller, cannot be deployed to a machine with a CAS worker.

- When CAS is deployed across multiple machines, each machine requires passwordless SSH to communicate with the other machines. Passwordless SSH is set up by the playbook by default. You can edit the passwordless SSH settings in the vars.yml file.

Figure 1.4  Full Deployment with a Distributed CAS Server

**TIP** A shared file system is recommended. The shared file system should reside on a machine other than the primary CAS controller or the secondary CAS controller. However, both controllers would use the shared file system to store data and configuration information. If the CAS primary controller fails, the secondary CAS controller could then assume the controller role and use the same data. Information about enabling a shared file system is provided later in this guide.

Multiple CAS Servers

You can deploy multiple CAS servers to support your SAS Viya environment. After the initial deployment, you can add CAS servers to new machines, and each additional CAS server can support either SMP or MPP mode. For example, a single-machine CAS server that is deployed to support SMP mode can operate in the same environment as a distributed CAS server, which supports MPP mode.
Advantages of deploying multiple CAS servers include the following:

- Multiple CAS servers support the separation of the CAS server data and processing demands of multiple work groups, for security and resource management purposes. Also, multiple CAS servers allow for redundant environments that support disaster recovery.

- To support a development, test, and production environment, you can stage an additional CAS server to manage and process data before promotion to production environments.

Note: The capacity for each additional CAS server must be within your license agreement.

Figure 1.5 Guidance for Deploying Multiple CAS Servers

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A mirror repository is recommended to ensure that the software versions used for the deployment of the initial CAS server are used for the subsequent, additional deployments of CAS servers. For more information, see “Create a Mirror Repository” on page 56.</td>
</tr>
<tr>
<td>2</td>
<td>Ansible is used for the initial deployment of the SAS Viya environment, including the CAS server. The instructions for the initial deployment of the software are provided in this guide.</td>
</tr>
<tr>
<td>3</td>
<td>Ansible is used to add a new CAS server. For more information, see SAS Viya Administration: Add a CAS Server to a New Machine.</td>
</tr>
</tbody>
</table>

Cluster for High Availability PostgreSQL

SAS Viya uses High Availability (HA) PostgreSQL as the SAS Infrastructure Data Server. SAS Infrastructure Data Server stores user content such as reports, custom groups, comments, authorization rules, selected source definitions, attachments, audit records, and user preferences.
By default, Ansible deploys HA PostgreSQL as a single node on a single machine. The standard deployment consists of one PGPool and one PostgreSQL data node. However, you can deploy an HA PostgreSQL cluster to achieve higher performance and to support redundancy. For more information, see “Creating High Availability PostgreSQL Clusters” on page 227.

The following example shows an HA PostgreSQL horizontal cluster, where each data node is on a separate machine. Other topologies, such as a vertical cluster or a hybrid cluster, are supported.

Figure 1.6  High Availability PostgreSQL Horizontal Cluster

Multi-tenant Environment

SAS Viya supports a multi-tenant environment where a single instance of the software can serve multiple tenants. If you opt for multi-tenancy, you must enable it during the initial deployment. If you enable multi-tenancy, a provider is created during deployment. After deployment, you can onboard additional tenants. Each tenant has access to the licensed software but has no visibility into the data and workflows of other tenants.

Important: You cannot retrofit multi-tenancy in an environment where multi-tenancy was not enabled during the initial deployment. If you anticipate having multiple tenants, even if you have only one tenant initially, you must enable multi-tenancy. Changing from a single-tenant deployment to a multi-tenant deployment requires that you re-deploy your software.

To enable a multi-tenant environment and then to onboard the tenants, you must use this guide and SAS Viya Administration, which includes instructions for onboarding and managing tenants.
Figure 1.7  Guidance for Deploying and Configuring Multi-tenancy

1. A mirror repository is recommended to ensure that the same software versions are used across tenants. For more information, see “Create a Mirror Repository” on page 56.

2. The LDAP server is configured to support multi-tenancy. Information about the required LDAP configuration is provided in “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46 and “Set Up Accounts for Multi-tenant Deployments: Separate LDAP Server per Tenant” on page 48.

3. Ansible is used to deploy the provider. Before you run the playbook, you must enable multi-tenancy, which includes editing the vars.yml and sitedefault.yml files. For more information, see “SAS Viya and Multi-tenancy” on page 112.

4. Tenants are onboarded. For more information, see SAS Viya Administration: Multi-tenancy.

Table 1.2  Steps — Guidance for Deploying and Configuring Multi-tenancy

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A mirror repository is recommended to ensure that the same software versions are used across tenants. For more information, see “Create a Mirror Repository” on page 56.</td>
</tr>
<tr>
<td>2</td>
<td>The LDAP server is configured to support multi-tenancy. Information about the required LDAP configuration is provided in “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46 and “Set Up Accounts for Multi-tenant Deployments: Separate LDAP Server per Tenant” on page 48.</td>
</tr>
<tr>
<td>3</td>
<td>Ansible is used to deploy the provider. Before you run the playbook, you must enable multi-tenancy, which includes editing the vars.yml and sitedefault.yml files. For more information, see “SAS Viya and Multi-tenancy” on page 112.</td>
</tr>
<tr>
<td>4</td>
<td>Tenants are onboarded. For more information, see SAS Viya Administration: Multi-tenancy.</td>
</tr>
</tbody>
</table>

SAS Data Preparation and SAS Data Agent

If your software order includes SAS Data Preparation, SAS Data Agent is also included. SAS Data Agent enables data to move from an on-premises environment to SAS Data Preparation. SAS Data Agent is deployed when data must pass through a firewall to access SAS Data Preparation. SAS Data Agent is optimized to connect to SAS Data Preparation that is running on a private cloud or a public cloud.

Note: Although SAS Data Agent is included in an order that includes SAS Data Preparation, it does not appear in the list of software in the Software Order Email (SOE).

When deploying the software, consider the following:
The tasks that are required to deploy SAS Data Preparation are provided in this guide. The tasks to deploy a SAS Data Agent server and to register it to SAS Data Preparation are provided in SAS Data Agent on Linux: Deployment Guide.

SAS Data Preparation must be deployed and operational before you can register SAS Data Agent to SAS Data Preparation.

SAS Data Preparation and SAS Data Agent are deployed to different machines. To configure communications between SAS Data Preparation and SAS Data Agent, you must perform tasks on both machines.

If SAS Data Preparation is deployed within a SAS Viya multi-tenant environment, you can register one or more SAS Data Agent servers to communicate with each tenant. However, each SAS Data Agent server can be registered to only one tenant. In other words, you do not have to register a SAS Data Agent server to every tenant. You can register a SAS Data Agent server to the provider in a multi-tenant environment and to any tenants that are onboarded.

The tasks for deploying a multi-tenant environment are provided in this guide. The tasks for onboarding tenants are provided in SAS Viya Administration: Multi-tenancy.

Hadoop Integration: Access Data in Hive

The following examples provide guidance for deploying SAS Viya to support accessing data in Hive.

In the first example, SAS Data Connector to Hadoop is configured to allow the serial loading of data from Hive tables into a distributed CAS server.

Figure 1.8  Access Data in Hive: Serial Processing
Table 1.3  Steps — Access Data in Hive: Serial Processing

1. Ansible is used to deploy the SAS Viya environment, including a distributed CAS server. Before you begin the deployment process, an important task is to configure SAS Data Connector to Hadoop in the vars.yml file.

2. Ansible is run again to deploy the Hadoop JAR files to the Hadoop cluster, and then to the CAS controller and SAS programming nodes.

   Note: In this example, Ansible is shown to deploy Hadoop JAR files to the required nodes. As an alternative, SAS provides the hadoop_extract script, which you can run manually to collect and deploy the Hadoop JAR files.

   For more information about deploying the Hadoop JAR files, see “Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop” on page 239.

In the next example, SAS Data Connect Accelerator for Hadoop is configured to allow the parallel loading of Hive tables into a distributed CAS server. Also, the SAS Embedded Process, which is deployed to the Hadoop cluster, is used to provide high-speed parallel data transfer between Hive and the distributed CAS server. Processing occurs at the CAS server.

Figure 1.9  Access Data in Hive: Parallel Processing

Table 1.4  Steps — Access Data in Hive: Parallel Processing

1. Ansible is used to deploy the SAS Viya environment, which includes a distributed CAS server. Before you begin the deployment process, an important task is to configure SAS Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop in the vars.yml file.
Ansible is run again to collect the Hadoop JAR files from the Hadoop cluster, and then to deploy the files to the CAS controller and the SAS programming nodes.

Note: In this example, Ansible is shown to deploy Hadoop JAR files to the required nodes. As an alternative, SAS provides the hadoop_extract script, which you can run manually to collect and deploy the Hadoop JAR files.

For more information about deploying the Hadoop JAR files, see “Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop” on page 239.

The SAS Embedded Process is deployed to all nodes in the Hadoop cluster. For more information, see “Hadoop Deployment: Configuring SAS In-Database Technologies” on page 249.

Hadoop Integration: CAS SASHDAT Access to HDFS

SAS Plug-ins for Hadoop are deployed to the Hadoop cluster. SAS Plug-ins for Hadoop enable CAS to write SASHDAT file blocks evenly across the HDFS file system. This even distribution provides a balanced workload across the machines in the cluster and enables SAS Viya analytic processes to read SASHDAT tables very quickly.

To support CAS SASHDAT access to HDFS, two deployment examples are described: co-locating the CAS server with Hadoop, and configuring access to a remote Hadoop cluster.

CAS Server Co-located with Hadoop

In this first example, the CAS server is deployed across Hadoop nodes. Specifically, the CAS controller is deployed to the NameNode, and the CAS workers are deployed to all DataNodes or to a subset of the DataNodes. Also, SAS Plug-ins for Hadoop is deployed to the Hadoop NameNode and all the DataNodes.

An advantage of this type of deployment is that SASHDAT on HDFS can serve as a local cache for CAS tables, which facilitates fast saves and loads of CAS in-memory tables. If you have sufficient resources on your Hadoop cluster, consider deploying the CAS server on Hadoop nodes.

Here is an overview of the deployment process:
Table 1.5  Steps — CAS Deployed on All Hadoop Nodes

1. Ansible is used to deploy the CAS controller to the NameNode and to deploy CAS workers to the DataNodes. Also, the SAS Viya applications are deployed. Before you begin the deployment process, an important task is to set the environment variable for co-location and the Hadoop environment variable in the vars.yml file.

2. The Hadoop cluster is configured, and SAS Plug-ins for Hadoop is configured on the Hadoop NameNode and all the DataNodes. For more information, see “Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS” on page 257.

Note: SAS Plug-ins for Hadoop is provided as a parcel or a stack, which you can activate with Cloudera Manager or Ambari, respectively. As an alternative, SAS provides a script that you can run to install SAS Plug-ins for Hadoop.

Consider the following when deciding where to deploy the CAS controllers and the CAS workers:

- If your CAS license permits fewer than the total number of CPU cores in your Hadoop environment, you can deploy the CAS workers to all DataNodes and use a subset of the CPU cores.
- If the CAS workers are deployed to all DataNodes, passwordless SSH is not required for loading SASHDAT tables.
- If the CAS workers are deployed to a subset of the DataNodes, consider the following:
  - Passwordless SSH is required in order to load SASHDAT tables.
  - When you save a SASHDAT table, data is still written locally. Therefore, the `env.CAS_ENABLE_REMOTE_SAVE` environment variable does not have to be defined. Also, data is written in parallel only to those DataNodes on which a CAS worker is deployed.
  - SAS Plug-ins for Hadoop is configured on the Hadoop NameNode and all the DataNodes.
- If you move the CAS controller to a DataNode, SAS recommends that you define the `HADOOP_NAMENODE` environment variable.
Remote Access to HDFS

In this example, a distributed CAS server is deployed to machines that are not part of the Hadoop cluster. SAS Plug-ins for Hadoop is deployed to the Hadoop cluster, which enables a remote, parallel connection between the HDFS nodes and the CAS server.

Note: Passwordless SSH is required between the CAS nodes and the HDFS nodes.

Figure 1.11  Remote Access to HDFS

Table 1.6  Steps — Remote Access to HDFS

1. Ansible is used to deploy the SAS Viya environment, which includes a distributed CAS server. Before you begin the deployment process, an important task is to set the Hadoop environment variables in the vars.yml file.

2. The Hadoop cluster is configured, and SAS Plug-ins for Hadoop is configured on each Hadoop node. For more information, see “Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS” on page 257.

Note: SAS Plug-ins for Hadoop is provided as a parcel or a stack, which you can activate with Cloudera Manager or Ambari, respectively. As an alternative, SAS provides a script that you can run to install SAS Plug-ins for Hadoop.

Note: When users access HDFS remotely, CSV files cannot be saved back to HDFS.

Teradata Integration

SAS provides two data connectors that can enable access to data in Teradata.

- SAS Data Connector to Teradata enables you to load data serially from the Teradata data warehouse to the CAS server.
- SAS Data Connect Accelerator for Teradata enables you to load data in parallel using the SAS Embedded Process. SAS Embedded Process, which resides on the data appliance, is used to provide high-speed
parallel data transfer between the Teradata data warehouse and the CAS server. The data is processed at the CAS server.

The following example provides guidance for deploying SAS Viya to support accessing data in Teradata.

**Figure 1.12 Deployment with Teradata**

<table>
<thead>
<tr>
<th>Table 1.7 Steps</th>
<th>Deployment with Teradata</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Ansible is used to deploy the SAS Viya environment, which includes a distributed CAS server. Before you begin the deployment process, you must configure SAS Data Connector to Teradata and SAS Data Connect Accelerator for Teradata in the vars.yml file.</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>The SAS Embedded Process is deployed to the Teradata appliance. For more information, see “Teradata Deployment: Configuring SAS In-Database Technologies” on page 275.</td>
</tr>
</tbody>
</table>

**Elasticsearch Cluster**

Elasticsearch, which provides search capabilities for SAS Visual Investigator and SAS Intelligence and Investigation Management, is used to generate data for visualizations. In this example, Elasticsearch is deployed across multiple nodes in a clustered environment. An advantage of this deployment is that optimal processing can be achieved through spreading the Elasticsearch queries across multiple nodes. During deployment, the Elasticsearch master nodes and client nodes are deployed on separate machines. Data nodes are also on their own machines.

For more information about Elasticsearch, see the documentation at the following site:


**Note:** When you deploy the software in a production environment, consider the performance of multiple servers that run on the same machine. For example, because CAS, PostgreSQL and Elasticsearch are memory-intensive applications, SAS recommends that you do not deploy them on the same machine.
SAS Products and Supporting Components

This guide provides information for deploying the following software:
| SAS Adaptive Learning and Intelligent Agent System | SAS Visual Data Mining and Machine Learning 8.4 |
| SAS Analytics for IoT 7.1 | SAS Visual Forecasting 8.4 |
| SAS Assortment Planning 8.2 | SAS Visual Investigator 10.5.1 |
| SAS Cloud Analytic Services for Viya 3.4 | SAS Visual Statistics (on SAS Viya) 8.4 |
| SAS Data Preparation 2.2 | SAS Visual Text Analytics 8.4 |
| SAS Data Quality 3.4 | SAS/ACCESS Interface to Amazon Redshift (on SAS Viya) |
| SAS Data Science 8.3 (for more information about SAS Data Science, see the SAS Data Science product page) | SAS/ACCESS Interface to DB2 (on SAS Viya) |
| SAS Decision Manager 5.2 | SAS/ACCESS Interface to Google BigQuery (on SAS Viya) |
| SAS Demand Planning 8.2 | SAS/ACCESS Interface to Greenplum (on SAS Viya) |
| SAS Econometrics 8.4 | SAS/ACCESS Interface to Hadoop (on SAS Viya) |
| SAS Event Stream Manager 6.1 | SAS/ACCESS Interface to HAWQ (on SAS Viya) |
| SAS Event Stream Processing (SAS Viya Enabled) 6.1 | SAS/ACCESS Interface to Impala (on SAS Viya) |
| SAS Event Stream Processing for CAS 6.1 | SAS/ACCESS Interface to JDBC (on SAS Viya) |
| SAS Financial Planning 8.2 | SAS/ACCESS Interface to Microsoft SQL Server (on SAS Viya) |
| SAS In-Database Technologies for Hadoop (on SAS Viya) | SAS/ACCESS Interface to MySQL (on SAS Viya) |
| SAS In-Database Technologies for Spark (on SAS Viya) | SAS/ACCESS Interface to Netezza (on SAS Viya) |
| SAS In-Database Technologies for Teradata (on SAS Viya) | SAS/ACCESS Interface to ODBC (on SAS Viya) |
| SAS Intelligence and Investigation Management 1.4 | SAS/ACCESS Interface to Oracle (on SAS Viya) |
| SAS Intelligent Decisioning 5.3 | SAS/ACCESS Interface to PC Files (on SAS Viya) |
| SAS Markdown Optimization (on SAS Viya) 8.2 | SAS/ACCESS Interface to PostgreSQL (on SAS Viya) |
| SAS Mobile Investigator | SAS/ACCESS Interface to R/3 (on SAS Viya) |
| SAS Model Manager (on SAS Viya) 15.2 | SAS/ACCESS Interface to SAP ASE (on SAS Viya) |
| SAS Optimization 8.4 | SAS/ACCESS Interface to SAP HANA (on SAS Viya) |
| SAS Risk Engine 8.1 | SAS/ACCESS Interface to Snowflake (on SAS Viya) |
| SAS Size Optimization 4.2 (on SAS Viya) | SAS/ACCESS Interface to Spark (on SAS Viya) |
| SAS Studio 5.1 | SAS/ACCESS Interface to Teradata (on SAS Viya) |
| SAS Studio 4.4 | SAS/ACCESS Interface to Vertica (on SAS Viya) |
| SAS Text Analytics (for languages) | SAS/CONNECT (on SAS Viya) |
| SAS Visual Analytics (on SAS Viya) 8.4 | SAS/IML (on SAS Viya) |
| | SAS/QC (on SAS Viya) |

**Note:** Check your Software Order Email (SOE) for the list of software that you have licensed.

---

**Contact SAS Technical Support**

Technical support is available to all customers who license SAS software. However, you are encouraged to engage your designated on-site SAS support personnel as your first support contact. If your on-site SAS support personnel cannot resolve your issue, have them contact SAS Technical Support to report your problem.
Before you contact SAS Technical Support, explore the SAS Support website at support.sas.com/techsup/. This site offers access to the SAS Knowledge Base, as well as SAS communities, Technical Support contact options, and other support materials that might answer your questions.

When you contact SAS Technical Support, you are required to provide information, such as your SAS site number, company name, email address, and phone number, that identifies you as a licensed SAS software customer.
System Requirements

**Hardware Requirements**
- Host Requirements ......................................................... 22
- General Hardware Considerations ...................................... 23
- File System and Storage Requirements ................................. 26

**Operating System Requirements**
- Supported Operating Systems ........................................... 28
- Requirements for All Linux Platforms ................................. 28
- Additional Requirements for Red Hat Enterprise Linux and Oracle Linux ........................................... 29
- Additional Requirements for SUSE Linux ............................... 30
- (Optional) Additional Requirement for Multi-tenancy .......... 30
- (Optional) Additional Requirements for CAS Resource Management ........................................... 30
- Additional Requirement for SAS Visual Investigator and SAS Intelligence and Investigation Management ........................................... 31
- Additional Requirements for the ESP Server ............................. 31
- SAS Support for Alternative Operating Systems .................... 31

**Server Software Requirements**
- Java ............................................................................. 32
- Apache httpd ..................................................................... 32

**Data Source and Storage Requirements**
- Overview of Data Source Requirements .............................. 33
- Supported Data Sources .................................................. 33
- Data Sources Supported by SAS Visual Investigator and SAS Intelligence and Investigation Management ........................................... 34
- Hadoop Requirements ...................................................... 35
- Requirements to Transfer Data from SAS 9.4 ..................... 35
- Requirements for SAS/ACCESS Interface to Amazon Redshift ........................................... 36
- Requirements for SAS/ACCESS Interface to DB2 ................. 36
- Requirements for SAS/ACCESS Interface to Google BigQuery ........................................... 36
- Requirements for SAS/ACCESS Interface to Greenplum ........ 36
- Requirements for SAS/ACCESS Interface to Hadoop .......... 36
- Requirements for SAS In-Database Technologies for Hadoop ........................................... 36
- Requirements for SAS/ACCESS Interface to HAWQ ............. 37
- Requirements for SAS/ACCESS Interface to Impala ............... 37
- Requirements for SAS/ACCESS Interface to JDBC ............... 37
- Requirements for SAS/ACCESS Interface to Microsoft SQL Server ........................................... 37
- Requirements for SAS/ACCESS Interface to MySQL ............ 37
- Requirements for SAS/ACCESS Interface to Netezza .......... 38
- Requirements for SAS/ACCESS Interface to ODBC .......... 38
- Requirements for SAS/ACCESS Interface to Oracle .............. 38
- Requirements for SAS/ACCESS Interface to PC Files ........... 38
- Requirements for SAS/ACCESS Interface to PostgreSQL .... 38
Hardware Requirements

Host Requirements

Each target machine in your deployment must have all of the following attributes:

- A static IP address
  
The SAS Configuration Server component binds to a single private IP address per machine. If any of your intended hosts has multiple network interface cards (NICs), verify whether multiple NICs have been assigned IP addresses, including private IP addresses. To avoid an error during the deployment, you must edit the inventory file to add a `consul_bind_adapter` parameter. For more information, see “Edit the Inventory File” on page 74.

- A static host name
  
  Some networking environments, such as Dynamic Host Configuration Protocol (DHCP), and some cloud providers use dynamic host names or IP address assignments by default. Although it is possible to deploy the software successfully in these environments, any future change to either IP addresses or host names might result in an inoperative deployment. Therefore, SAS recommends that before you start the installation, you work with your network administrator to ensure that IP addresses and host names are static.
Important: On the Linux machine that will host the CAS server, make sure that the host name in `/etc/hosts` is specified in all lowercase letters. If you change the host name to comply with this requirement, verify that the image is stable and that all other services are working correctly before you start the deployment process.

- A host name that can be resolved to an IP address
  
  Reverse DNS resolution must be enabled. Your DNS server should have a mapping of each IP address to its corresponding fully qualified domain name (FQDN) for each machine in your deployment. As an alternative, both the FQDN and IP address of each machine can be added to its `/etc/hosts` file.

- An FQDN that is 64 characters or fewer in length
  
  This restriction is related to the implementation of Transport Layer Security (TLS). One of the specifications for the certificate revocation list is a 64-character limit for the common name (CN) attribute. For more information, see RFC 5280.

- The `/tmp` directory on the Ansible target machines must be on a partition that is mounted as executable. A deployment script must be able to execute from `/tmp`.

If you plan to deploy the software on multiple machines, make sure that the clock time is synchronized across all of them. For example, you can use a Network Time Protocol (NTP) server for this purpose.

**General Hardware Considerations**

A full deployment is the recommended type of deployment for most customer requirements. Most of the recommendations in this section apply to a full deployment. To deploy SAS Event Stream Manager or SAS Analytics for IoT, you must perform a full deployment. However, SAS Data Science is always deployed as programming-only.

SAS recommends installing a full deployment of SAS Viya on multiple machines for improved performance. However, SAS also strongly recommends consulting with a sizing expert to obtain an official hardware recommendation that is based on your deployment type, the estimated SAS workload, and the number of users. To request sizing expertise, contact your SAS account representative. If you need assistance in determining your SAS account representative, send an email to contactcenter@sas.com.

**CPU and RAM Recommendations**

SAS Viya has undergone rigorous performance testing with various hardware combinations. In addition to being tested on high-performing Intel Xeon E3-E7 series microprocessors, SAS Viya has also been tested with newer Intel chips, such as Intel Xeon Scalable Processors. SAS Viya also supports 64-bit AMD chipsets. Thirty-two-bit chipsets are not supported.

Consider the following as you prepare for the deployment process:

- The hardware guidelines in this guide reflect baseline standards. For a production environment, CPU, RAM, and disk resources should be increased after the expected amount of data to be processed and number of concurrent users are taken into consideration.

- Overall system performance will improve with the addition of both RAM and CPU cores.
  
  The CAS Server and the Programming Runtime must not exceed your licensed core count, but the microservices and web applications are not similarly restricted. Adding RAM to the CAS Server and Programming Runtime machines should improve performance. Adding both CPU cores and RAM to the machines that host the microservices can also improve performance.

- Test machines were equipped with RAM that had a minimum memory clock speed of 1600 MHz.
Architectural Considerations

SAS Viya is built for a scalable, flexible architecture. If the amount of data that is processed is relatively small, a programming-only deployment of a few SAS Viya products will perform well on a Linux VM running on a standard 4-core computer. A full deployment that includes the visual applications requires more resources. However, in both situations, SAS Viya can scale across many nodes to meet the requirements of a particular enterprise. Therefore, properly sizing your license and selecting the appropriate hardware must be based on an understanding of the planned usage of the software in your environment.

The SAS Viya architecture consists of three categories of components that you should consider as you plan your deployment. These components can all be installed on the same host, or they can be distributed over multiple hosts:

- **CAS Server**
  The CAS Server is required for all deployments, regardless of type (full or programming-only). It is licensed by CPU core, with a minimum license size of 4 cores.
  The amount of RAM that is required for the CAS Server is determined by the amount of data that is processed, and by the level of user activity in the environment. However, out of the box, the amount of RAM that is required to start the CAS Server is less than 1 GB.

- **Programming Runtime**
  The Programming Runtime consists of multiple components that are required for all deployments, regardless of type (full or programming-only). It includes the SAS compute server, SAS Foundation, SAS Studio, SAS Workspace Server, SAS/CONNECT Server, and any SAS/ACCESS engines that you have licensed.
  The number of CPU cores that are required for the Programming Runtime depends on your workload. This resource is constrained by the license that you purchased. The number of licensed cores for CAS represents a ceiling that also applies to the Programming Runtime. However, the minimum requirement is 2 cores. SAS recommends that you allocate at least 4 cores for optimal performance.
  The minimum required amount of RAM for the Programming Runtime is 4 GB. SAS recommends that you allocate at least 16 GB of RAM, or 4 GB for each CPU core.

- **Service Layer**
  This category consists of components that are required for a full deployment, as well as services that support specific SAS products. The components of the Services Layer are not usage restricted. They include the Core Services host group and all the other services that support SAS Viya analytics processing.
  The host groups that compose the Service Layer can be deployed on multiple hosts, and with as many CPU cores as are needed for optimal performance and availability.

**Baseline Hardware Requirements by Product**

The following table lists products that can be separately licensed and indicates the minimum RAM and number of CPU cores to support individual components when they are installed on a single machine. Testing was performed with a full deployment. These guidelines do not apply to a programming-only deployment.

The table represents what is required to start all system services and to enable a single user to operate against a small sample data set in order to validate operational functionality. The final row of the table indicates the requirements when all products are installed on the same machine. These out-of-the-box requirements should be increased for larger deployments.

Additional RAM should be added based on the expected amount of data that will be processed. More resources are required for multiple-user, production-scale deployments that use large data sets. These guidelines do not attempt to account for all ordering scenarios, but instead are intended to illustrate typical software orders.
Table 2.1 Minimum Hardware Requirements for Each Product

<table>
<thead>
<tr>
<th>Products</th>
<th>RAM (GB)</th>
<th>CPU Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Visual Analytics</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Visual Statistics</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Investigator</td>
<td>44</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Investigator and SAS Visual Analytics</td>
<td>52</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Visual Forecasting</td>
<td>56</td>
<td>12</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Visual Text Analytics</td>
<td>56</td>
<td>12</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Data Preparation</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Data Quality</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Decision Manager or SAS Intelligent Decisioning</td>
<td>48</td>
<td>12</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Model Manager</td>
<td>48</td>
<td>8</td>
</tr>
<tr>
<td>SAS Visual Analytics and SAS Analytics for IoT</td>
<td>56</td>
<td>12</td>
</tr>
<tr>
<td>All products listed above</td>
<td>68</td>
<td>16</td>
</tr>
</tbody>
</table>

SAS Viya installs executables and creates configuration directories in `/opt/sas/`. The minimum available disk space that is required to install and start a full deployment of SAS Viya is less than 50 GB. However, logs and operational data will quickly grow to exceed that amount. Therefore, the actual space that is required will depend on the amount of data and the level of activity in your specific deployment. For more information, see “File System and Storage Requirements” on page 26.

These out-of-the-box requirements will accommodate the CAS Server (to the extent that the resources comply with your product licensing terms) as well as the Programming Run-time for validation purposes. Be aware that the start-up times for the various services in the environment and the level of performance that they deliver will improve as CPU cores and memory are added.

**Hardware Requirements for Deep Learning**

Deep learning capabilities are included with SAS Visual Data Mining and Machine Learning and with SAS Analytics for IoT. Use the requirements for these products to prepare your target machines. The deep learning features are automatically installed.
A graphical processing unit (GPU) is not required on your computer in order to use deep learning features. However, a GPU provides additional functionality. To enable deep learning with GPU functionality, here are additional requirements:

- A powerful, CUDA-capable GPU. SAS has tested with GPUs that have the NVIDIA Compute Unified Device Architecture (CUDA).
  
  SAS recommends, and has tested on, servers with the NVIDIA Tesla P100 and V100 GPUs.

- The NVIDIA display driver, version 396.37 or later. SAS recommends using the latest version.
  
  SAS also recommends enabling NVIDIA driver persistence mode at all times. For more information, see the Driver Persistence section of the NVIDIA deployment documentation.


- `/lib64` is the first path that is defined for the LD_LIBRARY_PATH environment variable on the server where the GPU is installed.

### File System and Storage Requirements

#### Disk Space Considerations

Verify that at least 48 GB of disk space are available for the installation. This requirement applies to both full deployments and programming-only deployments. The installation files are automatically downloaded to the `/var/cache/yum` directory.

The software is installed in the `/opt` directory on each target machine. In many cases, this directory is in a file system with 50 GB or fewer of disk space. To increase available disk space for the installation, SAS recommends that you mount additional volumes at `/opt/sas` instead of to a subdirectory of `/opt/sas`.

Mounting a volume in the installation directories increases the difficulty of uninstalling the software or of moving the volume to another location at a later time.

CAS servers automatically cache blocks of data on disk when they are working with tables whose size exceeds the available memory. The CAS controller also uses disk space for the CAS cache directory. You can configure the location of this cache in the playbook. For more information, see “Set Up the CAS Cache Directory” on page 100.

If your order included Model Studio, consider allocating additional storage space in `/opt/sas/viya/config/data/cas/default/projects`. Model Studio copies the data source when the first Data node in the project is run. Therefore, the amount of space that is required depends on the number of saved projects and on the size of the data source. Each project that end users create will require space in this directory structure until the project is deleted. For optimal performance, create this directory structure on a high-performance storage appliance, such as a SAN or another multi-device appliance.

Additional space for logs is required in `/opt/sas/viya`. The amount that is required depends on the logging level that you have set. However, the minimum amount of disk space that is required for the installation and for logging is 48 GB.

If disk space is limited, SAS recommends that you create symbolic links from the installation or log directories to the partitions where sufficient disk space is available. For example, you can create a symbolic link from the log directory (`/var/log`) to a directory that has additional free space:

```
/var/log/sas/viya -> ../../opt/sas/viya/config/var/log/sas/
```

As part of your log management strategy, create symbolic links at the `/opt/sas` level in order to capture all logging activity from SAS Viya components. SAS Viya also includes administrative tools for managing hardware resources for CAS servers. For information about CAS Resource Management, see SAS Cloud Analytic Services: CAS Resource Management.
The Apache httpd component of the Apache HTTP Server logs to `/var/log/httpd`. The logs in this directory can grow very large. In addition to using symbolic links to change the log location, you should also implement a log rollover strategy. See the Apache documentation for guidance about log rotation.

The regular monitoring of disk space usage is a critical CAS administrative task. For more information, see:

- Monitor Disk Space in SAS Viya Administration: Monitoring for a full deployment
- Monitoring: How to (CAS Server Monitor) in SAS Viya Administration: Monitoring for a programming-only deployment

Requirements for Caslib Data Access

A caslib is an in-memory space that enables the CAS server to hold tables, access control lists, and data source information. All data is available to CAS through caslibs, and all operations in CAS that use data are performed with a caslib in place. Among other functions, caslibs provide access to data from the data source and access to in-memory tables that are copied from the data source. In some situations, caslibs require additional persistent storage.

When you add or edit a caslib definition, you can provide a path to another location with additional storage space, such as an external drive. Although different caslibs might have different storage requirements that are data source-dependent, SAS recommends that you configure the persistent storage for all caslibs in a single location. Using a single location for persistent storage enables you to easily manage backup and recovery. In the caslib definition, rather than using a path, you can also specify a mount point that has additional storage.

Multiple predefined system caslibs and the Public caslib have a default location for persistent storage: `/opt/sas/viya/config/data/cas/instance-name/name-of-Public-caslib`. You can specify the instance name when you edit the playbook. If you anticipate that many users will use browsers to access the user interfaces and to import data from files, additional space for this file system will be required. SAS recommends monitoring disk usage at `/opt/sas/viya/config/data/cas`.

Each CAS user has a personal caslib called CASUSER, and CAS administrators typically set it to write to the user’s home directory. This caslib might also require some additional disk space, depending on the individual user’s requirements.

Requirements for File Systems

The file system that contains the `/opt/sas/viya/config` directories must be mounted with support for Access Control Lists (ACLs).

You can deploy a primary CAS controller and a secondary CAS controller to support failover. A common, shared file system must be available to both controllers.

If you are deploying for multi-tenancy, plan to use a shared file system for all the path-based caslibs in each tenant because each tenant will have its own CAS controller. Mount the file system at `/opt/sas/tenant/config/cas/data`. For `tenant`, substitute `viya` for a single tenant, or substitute the name of one of your tenants.

Note: If you mount the shared file system before running the playbook, be aware that the installation runs in this path as root. The root user therefore requires permissions to modify the file system across the mount. After the installation has completed, root permissions are not required. To avoid this requirement, you can set up this file system post-deployment. For more information, see Set Up a Shared File System for CAS Controllers (Post-Deployment) in SAS Viya Administration.

Make sure that end users’ home directories ($HOME) are also shared so that they can be accessed on both the primary and secondary CAS controllers. When starting CAS sessions from SAS Studio, shared home directories ensure that the path for the CASUSER library is available during CAS session start-up.

Some caslibs use a path to a directory as the data source. Therefore, in an environment with a secondary controller, the directory must be a network location that is available at the same path on both controllers. A file system that is mounted at this location accommodates all caslibs that are created by SAS. Similarly, enable a
shared file system if you are deploying a secondary controller, and you are deploying the CAS server on a massively parallel processing (MPP) system. For more information, see “Enable a Shared File System” on page 65.

High-Availability Requirements

Some SAS Viya software components can be deployed with multiple instances to support high availability. A few additional requirements apply to these deployments.

- **SAS Message Broker**

  High availability for SAS Message Broker creates a cluster of machines that run RabbitMQ instances. These instances are susceptible to message loss if an outage causes them to detect a partition in the network. Set up an odd number of SAS Message Broker machines for optimal message queue management. By default, when a node in the cluster detects a partition, nodes that are on the side of the partition with fewer nodes can then automatically pause and enable all message queues to synchronize. For more information, see SAS Note 63804.

- **Compute server**

  When multiple instances of the compute server are installed, the required user home directories must be accessible to all of them.

- **SAS Studio 4.4 (user interface for programming-only deployments)**

  SAS Studio enables users to set preferences and store work. As a result, all instances of SAS Studio must be able to access the same saved configuration data.

  You must meet one of the following requirements to support SAS Studio 4.4 in a high-availability configuration:

  - User home directories must be accessible to all instances of SAS Studio.
  - SAS Studio must be configured to use a shared drive in a shared file system.

  This option requires additional setup to instruct SAS Studio to use the shared drive. For more information, see the description of the webdms.studioDataParentDirectory property in SAS Viya Administration: Configuration Properties.

---

**Operating System Requirements**

**Supported Operating Systems**

For the full list of supported operating systems, see [https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-operating-systems.html](https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-operating-systems.html).

In a multi-machine deployment, SAS recommends that all server machines have the same version of Linux, including the same distribution, release, and patch level. Note that all CAS Server machines must be running the same version of Linux.

**Requirements for All Linux Platforms**

The requirements in this section apply to all of the supported Linux operating systems.
Libraries and Packages

The typical Linux installation includes most of the packages and libraries that SAS requires. Problems can occur if default packages were removed from the base operating system (for example, X11 libraries and system utilities).

The following libraries and packages are required for Red Hat Enterprise Linux, Oracle Linux, and SUSE Linux:

- acl-2.2 or later
  The acl package is installed with Red Hat Enterprise Linux by default. For SUSE Linux, it is available in the base repositories.
- glibc-2.12-1.166.el6 and later (on Red Hat Enterprise Linux 6.x or the equivalent). Refer to RHBA-2015:1465 on the Red Hat Customer Portal to obtain the latest updated package list.
  glibc-2.17-107.el7 and later (on Red Hat Enterprise Linux 7.x or the equivalent). Refer to RHSA-2016:2573 on the Red Hat Customer Portal to obtain the latest updated package list.
  glibc-2.22 and later (on SUSE Linux)
- libpng (on Red Hat Enterprise Linux 6.x or the equivalent)
  libpng12 (on Red Hat Enterprise Linux 7.x, Oracle Linux 7.x, or SUSE Linux)
- libXp
  Note: For SUSE Linux, the package is named libXpm4.
- libXmu
- net-tools
- the numactl package
- systemd version 219-30 or later
- the X11/Xmotif (GUI) packages
- xterm

Verifying systemd

On Linux 7.x and SUSE Linux, verify that the systemd package on each machine is a supported version. Run the following command:

```
rpm -qa | grep systemd
```

For Red Hat or Oracle, if the version that is returned is not at least 219-30, run the following command to retrieve the most recent package:

```
yum update systemd
```

For SUSE, run the following command to retrieve systemd information:

```
zypper update systemd
```

Additional Requirements for Red Hat Enterprise Linux and Oracle Linux

SUSE Linux does not use yum as a deployment tool and therefore does not require a subscription service. If you are using SUSE Linux, you should skip this section.

A SAS Viya deployment requires the operating system to be registered with the Red Hat Network or Oracle Unbreakable Linux Network (ULN). Registration enables you to receive periodic software updates. For a SAS
software deployment, registration also enables yum to download software from SAS repositories. Verify that the machine where you perform the deployment (typically, the Ansible controller) is registered and that your subscription has been activated.

The Ansible controller must be connected to the Red Hat Network with a Server-Optional subscription in addition to the Base (operating-system) subscription. The managed nodes must also be registered to the Red Hat Network, but a Base subscription is sufficient.

To check whether the system is registered, run the following command on Red Hat Enterprise Linux:

```
subscription-manager version
```

The command returns information about the subscription service to which the system is registered. To check whether the subscription has been activated, run the following command:

```
subscription-manager list --available
```

A list of active subscriptions is returned.

For Oracle Linux, you periodically see a message stating that `This system is not registered with ULN` if your ULN subscription is not active. To register an Oracle Linux installation with the ULN, run the following command as the root user:

```
uln_register
```

On a machine that lacks a support contract with Oracle, you can set up a connection to the Oracle Public Yum Server. For more information, see [http://public-yum.oracle.com/](http://public-yum.oracle.com/).

If you have enabled Security-Enhanced Linux (SELinux) in your environment, you must perform additional steps to accommodate SELinux for a SAS Viya deployment with Ansible. For more information, see “Configure SELinux” on page 63.

The default shell, Bash, is required. You can use other shells, but Bash must be present.

In addition, the setuid mount option must be enabled for the file systems in which SAS software is installed. A few processes must be able to access these file systems at SAS run time.

**Additional Requirements for SUSE Linux**

The default shell, Bash, is required. If your machine is set to use a different shell, errors occur during the deployment process.

To avoid errors during playbook execution, verify that the `which` utility has been installed. Use the following command:

```
sudo zypper in which
```

**(Optional) Additional Requirement for Multi-tenancy**

To support multi-tenancy, the jq-1.5 package (or a later version) must be installed. It is required to enable the command-line interface. SAS recommends that you obtain the jq package from GitHub.

The onboarding playbook will fail to run if the jq package is not installed on the hosts that are listed in the CommandLine host group. For more information about onboarding tenants, see [How To (Provider Administrators)](SAS_Viya_Administration:_Multi-tenancy) in SAS Viya Administration: Multi-tenancy.

**(Optional) Additional Requirements for CAS Resource Management**

The CAS resource policies that you can configure on CAS servers include an optional method for managing hardware resources. To support CAS resource policies, verify that your environment meets the following requirements:
The libcgroup and libcgroup-tools Linux packages are installed. These packages are typically included with the operating system.

(Red Hat 6.x only) The cgconfig service has been started.

On Red Hat 6.x, this service is installed with the libcgroup package, but on some varieties of Linux, it is not automatically started.

The cgroups file system has been mounted.

For more information about CAS resource management, see SAS Cloud Analytic Services: CAS Resource Management in SAS Viya Administration.

Additional Requirement for SAS Visual Investigator and SAS Intelligence and Investigation Management

For SAS Visual Investigator and SAS Intelligence and Investigation Management, an additional utility is required: jq-1.5.

Install this utility on each machine in the deployment.

Run the following commands to obtain the required version of jq and install it in the required location:

curl -L -o jq https://github.com/stedolan/jq/releases/download/jq-1.5/jq-linux64
chmod +x ./jq
sudo cp jq /usr/bin

Additional Requirements for the ESP Server

Note: This information is relevant for users of SAS Event Stream Processing or SAS Analytics for IoT.

The ESP server libraries were built using gcc-4.4.7-16 and the Boost library 1.58. The Boost library 1.58 is automatically installed with SAS Event Stream Processing. The libraries were compiled using the following compiler options:

-D_REENTRANT
-D_THREAD_SAFE

All the SAS Event Stream Processing applications that you build with SAS Event Stream Processing Studio must also use the same compiler options.

The SAS Event Stream Processing libraries have been built using gcc-4.4.7-16 on Red Hat Enterprise Linux Server 6.7 using libc-2.12.so, libstdc++.so.6.0.13, and libgcc_s-4.4.7-20120601.so.1.

SAS Support for Alternative Operating Systems

SAS provides support on a limited basis for alternative operating system distributions that customers might select. For more information, see the official support policy statement at http://support.sas.com/techsup/pcn/altopsys.html.
Server Software Requirements

Java

A Java Runtime Environment (JRE) must be installed on every machine in your deployment. The playbook checks for a preinstalled version of Java that meets or exceeds the requirements. If one is found, it is used. Otherwise, the playbook attempts to install a recent version of OpenJDK and to set the path in a system configuration file. You can also specify the path to an existing JRE in the vars.yml file before you run your playbook.

For a list of supported JRE distributions and other requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-jre.html.

Third-party distributions of the JRE are supported as long as the version matches the one that is listed on the SAS Support website. However, IBM SDK, Java Technology Edition is not supported. In some cases, running `sudo yum install java` to install Java can result in the unintentional installation of the IBM JRE, which causes failures with an installation utility.

The current JRE options for SAS Viya have been tuned for OpenJDK and Oracle JRE. If you use a JRE from another vendor and experience performance issues, SAS might recommend using OpenJDK or Oracle JRE. You should also verify that the version of Java is the same on every CAS server machine. You can determine the current Java version on a Linux machine by running the following command:

```
java -version
```

Apache httpd

The deployment process automatically installs Apache httpd on the machines that you designate as targets for the HTTP proxy installation unless it has already been installed. Apache httpd with the mod_ssl module is required in order to create the Apache HTTP Server, which provides security and load balancing for multiple SAS Viya components. This server is also referred to as the reverse proxy server in this guide.

For a list of supported Apache HTTP Server versions, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-apache.html.

SAS recommends that you install Apache httpd and configure the Apache HTTP Server to use certificates that comply with the security policies at your enterprise before you start the deployment process. The playbook will automatically configure the certificates to secure the server. For more information, see “Transport Layer Security” on page 41.

A high-availability proxy environment is not installed by default, but is a supported configuration. For example, you can include multiple machine targets in the playbook to install httpd on multiple servers. A load balancer is then required to provide high availability for the Apache HTTP Server. Otherwise, you risk bringing the SAS Viya environment down if one httpd instance becomes unavailable.

To install redundant instances and to specify the machine target or targets for the Apache HTTP Server, use the [httpproxy] host group in the inventory file. For more information, see “Assign the Target Machines to Host Groups” on page 77. If you install Apache httpd before starting the deployment process, specify any machines where you have installed it for the [httpproxy] host group so that the deployment can add required software to them. However, because the Apache HTTP Server is required for internal communications among SAS Viya components, do not replace the Apache components that are installed by the playbook.

The Apache HTTP Server must be dedicated to a single SAS Viya deployment.
Data Source and Storage Requirements

Overview of Data Source Requirements

Note: If you are deploying only SAS Data Preparation in the cloud, and it will access databases that are only in a SAS Data Agent on-premises deployment, you should skip to “Security Requirements” on page 40.

Additional SAS software is required in order to enable data retrieval from a Hadoop data store and from various data storage appliances. Depending on your software order, the deployment process might install one or more SAS/ACCESS products or a SAS In-Database Technologies product. The data access software is installed on the machines that you designate as the CAS controller and the CAS workers in your inventory file. As a best practice, modify settings in the vars.yml file for each SAS/ACCESS or SAS In-Database product that you plan to use before you run the playbook.

After the deployment process has completed, you might be required to install additional software on your CAS machines. For example, the database client must be installed on the CAS controller. If you decide to use the multinode data transfer feature, the database client software is also required on your CAS workers. For more information about the multinode option, see Data Connectors: Using Data Transfer Modes with Data Connectors for DBMS Data Sources in the SAS Cloud Analytic Services: User’s Guide.

Refer to the section that corresponds to your SAS/ACCESS product or SAS In-Database Technologies product for additional system requirements that apply to your data sources.

Supported Data Sources

SAS Viya supports multiple external data sources.

Note: For the list of data sources that are supported by SAS Visual Investigator and SAS Intelligence and Investigation Management, see “Data Sources Supported by SAS Visual Investigator and SAS Intelligence and Investigation Management” on page 34.

The following data sources require a SAS/ACCESS product, which might have additional requirements:

- Amazon Redshift
- Apache Hive
- Apache Spark
- IBM DB2
- Google BigQuery
- Impala
- Data sources accessible with a Java Database Connectivity (JDBC) driver
- Microsoft SQL Server
- MySQL
- Data sources accessible with an ODBC driver
- Oracle
- PC files
- PostgreSQL
- SAP HANA
- Snowflake
Teradata

Vertica

Some of these data sources also offer an optional SAS In-Database technology bundle to support parallel execution. Refer to the corresponding section of this guide for the requirements.

Support for the following external data sources is limited to integration with SAS 9.4. These sources also require a SAS/ACCESS product but do not offer a data connector to CAS:

- Apache HAWQ
- Greenplum
- IBM Netezza
- SAP ASE
- SAP R/3

SAS Viya also supports data sources that use a SAS data connector that is included with CAS and is not separately licensed or configured. Support for the following data sources is automatically included:

- SASHDAT on UNIX
- SASHDAT on HDFS (distributed CAS server is required)
- LASR Analytic Server (SAS 9.4)
- SAS Scalable Performance Data Engine (SPDE)
- SAS data sets

SAS Viya also supports CSV files, which do not require a SAS data connector or SAS/ACCESS product and can be accessed directly.

If you purchased SAS Event Stream Processing for CAS, a full installation of SAS Event Stream Processing is a required data source.

A PostgreSQL database is also used as an internal data store, named SAS Infrastructure Data Server. It is based on PostgreSQL version 9 and is configured specifically to support SAS software by storing user content and preferences.

Note: If you plan to deploy multiple tenants, be aware that only a single SAS Infrastructure Data Server is supported for all tenants. However, an individual tenant can deploy and use a different external data source. The corresponding SAS/ACCESS product would be required.

Data Sources Supported by SAS Visual Investigator and SAS Intelligence and Investigation Management

SAS Visual Investigator and SAS Intelligence and Investigation Management support the following data sources:

- IBM DB2
- Microsoft SQL Server
- MySQL
- Oracle 12c
- PostgreSQL
- Teradata

Data from MySQL and Microsoft SQL Server data sources is accessible using SAS/ACCESS Interface to ODBC.
Hadoop Requirements

Supported Distributions and Connection Requirements

SAS Viya supports multiple third-party distributions of Hadoop.

Note: If you upgrade your Hadoop version and have already deployed SAS Viya with SASHDAT, you must perform steps to redeploy SAS Viya with Hadoop. For more information, see SAS Note 60118.

For the full list of supported Hadoop distributions, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-databases.html.

You can connect to data as follows:

- For SASHDAT on HDFS, CAS components are typically installed on all SAS servers in your deployment and on every machine in your Hadoop cluster. No additional SAS data connector setup is required.
  
  Loading a table from HDFS requires Read/Write access to the /tmp directory. Permission should be granted to the user ID of the session process, which is either the cas user ID or the host account of an end user who is starting a session.
  
  If your CAS server is installed on a single machine, data in HDFS cannot be accessed by SAS Viya.

- For Hive, SAS/ACCESS Interface to Hadoop and optionally SAS In-Database Technologies for Hadoop are required. These products have individual system requirements, which are documented below.

  Note: Apache Hadoop 0.23, 2.4.0, and 2.7.1 and later versions are supported only as a Hadoop cluster that is co-located with CAS for access to SASHDAT on HDFS.

Support for Apache Spark

SAS Viya supports the transfer of data from Apache Spark clusters to CAS for analytics processing. Spark data access requires one of the following combinations:

- SAS/ACCESS Interface to Hadoop and SAS/ACCESS Interface to Spark, or

- SAS In-Database Technologies for Hadoop and SAS In-Database Technologies for Spark

Spark must be installed on the Hadoop nodes. When the playbook detects the presence of Spark, the remaining configuration is performed automatically.

Some additional requirements apply. For more information, see “Requirements for SAS/ACCESS Interface to Spark” on page 39 or “Requirements for SAS In-Database Technologies for Spark” on page 39.

SAS Support for Alternative Releases of Hadoop Distributions

SAS identifies the specific set of Hadoop distributions that are supported with each SAS product release. The SAS policy that applies to alternative releases or distributions of Hadoop is documented at the following website: https://support.sas.com/en/documentation/third-party-software-reference/9-4/support-for-hadoop.html#alternativereleases. The same policy that applies to SAS 9.4 also applies to SAS Viya.

Requirements to Transfer Data from SAS 9.4

For SAS 9.4 deployments that are earlier than SAS 9.4 TS1M5 (SAS 9.4M5), SAS/CONNECT is required in the environment in order to transfer data from other SAS deployments and operating systems to SAS Viya. SAS/CONNECT can convert data from a non-UTF-8 encoded SAS session to the UTF-8 format that SAS Viya requires.

SAS/CONNECT is not included with a standard SAS Viya order, and must be separately licensed.
By contrast, SAS 9.4M5 and later are integrated with SAS Viya directly. As a result, SAS/CONNECT is no longer required in order to transfer data from SAS 9.4M5 and later. All SAS programming clients in a 9.4M5 or later environment can call procedures that are enabled in SAS Viya and submit DATA step code, operating directly on CAS data sources. Examples of SAS programming clients are SAS Studio, SAS Enterprise Guide, SAS Data Integration Studio, and SAS Data Management Studio.

SAS/CONNECT is still supported, but if you are running SAS 9.4M5 or later, it is no longer required in order to transfer data into SAS Viya.

**Requirements for SAS/ACCESS Interface to Amazon Redshift**

SAS/ACCESS Interface to Amazon Redshift (on SAS Viya) includes SAS Data Connector to Amazon Redshift.

For information about supported Amazon Redshift versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to DB2**

SAS/ACCESS Interface to DB2 (on SAS Viya) includes SAS Data Connector to DB2.

For information about supported IBM DB2 versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to Google BigQuery**

SAS/ACCESS Interface to Google BigQuery (on SAS Viya) requires SAS Foundation and SAS Viya.

The required Google BigQuery client library is included with SAS/ACCESS Interface to Google BigQuery. No additional software is required.

**Requirements for SAS/ACCESS Interface to Greenplum**

SAS/ACCESS Interface to Greenplum (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported Greenplum versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to Hadoop**

SAS/ACCESS Interface to Hadoop (on SAS Viya) includes SAS Data Connector to Hadoop.

In order to access data from Apache Spark, SAS/ACCESS Interface to Spark is also required. For more information, see “Requirements for SAS/ACCESS Interface to Spark” on page 39.

For information about supported Hadoop versions and additional requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS In-Database Technologies for Hadoop**

SAS In-Database Technologies for Hadoop includes SAS Data Connect Accelerator for Hadoop. SAS/ACCESS Interface to Hadoop is required and is separately licensed.

SAS Embedded Process is included with SAS In-Database Technologies for Hadoop. It must be installed on the Hadoop cluster. SAS Viya 3.4 requires version 15.0 or later of SAS Embedded Process.
The CAS controller communicates with the Hadoop NameNode and YARN nodes. SAS Embedded Process is installed on nodes where an instance of YARN Node Manager is running. All Embedded Process tasks that are running on YARN nodes communicate with the CAS controller before they communicate with a CAS worker. The Embedded Process tasks access the CAS grid using the CAS network port, GCPORT. You can set GCPORT to a specific port number when you configure the playbook. If you do not set GCPORT, ephemeral ports are used.

The CAS controller uses a JVM to communicate with Hadoop in order to initiate the data transfer. As a result, the path to libjvm.so must be set for the LD_LIBRARY_PATH environment variable on the machine where SAS In-Database Technologies for Hadoop is installed. You can configure this path automatically during the deployment by editing the CAS_SETTINGS section of the vars.yml file. For more information, see “Modify the vars.yml File” on page 81.

In order to access data from Apache Spark, SAS/ACCESS Interface to Spark is also required. For more information, see “Requirements for SAS/ACCESS Interface to Spark” on page 39.

For more information about supported Hadoop versions and additional requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to HAWQ**

SAS/ACCESS Interface to HAWQ (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported Apache HAWQ versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to Impala**

SAS/ACCESS Interface to Impala (on SAS Viya) includes SAS Data Connector to Impala.

For information about supported Impala versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to JDBC**

SAS/ACCESS Interface to JDBC (on SAS Viya) includes SAS Data Connector to JDBC. SAS/ACCESS Interface to JDBC enables access to relational databases by means of SQL and the Java Database Connectivity (JDBC) API.

For information about JDBC support and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to Microsoft SQL Server**

SAS/ACCESS Interface to Microsoft SQL Server (on SAS Viya) includes SAS Data Connector to Microsoft SQL Server.

For information about supported Microsoft SQL Server versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.

**Requirements for SAS/ACCESS Interface to MySQL**

SAS/ACCESS Interface to MySQL (on SAS Viya) includes SAS Data Connector to MySQL.

For information about supported MySQL versions and requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-databases.html.
Requirements for SAS/ACCESS Interface to Netezza

SAS/ACCESS Interface to Netezza requires SAS Foundation and SAS Viya.

For information about supported IBM Netezza versions and requirements, see:

Requirements for SAS/ACCESS Interface to ODBC

SAS/ACCESS Interface to ODBC (on SAS Viya) includes SAS Data Connector to ODBC. SAS/ACCESS Interface to ODBC enables access to multiple data source types by means of a generic ODBC driver.

For information about ODBC support, see:

Requirements for SAS/ACCESS Interface to Oracle

SAS/ACCESS Interface to Oracle (on SAS Viya) includes SAS Data Connector to Oracle.

For information about supported Oracle versions and requirements, see:

Requirements for SAS/ACCESS Interface to PC Files

SAS/ACCESS Interface to PC Files (on SAS Viya) includes SAS Data Connector to PC Files.

SAS/ACCESS Interface to PC Files enables access to the following file formats:
- .jmp
- .spss
- .stata
- .xlsx or .xls

No additional software is required.

Requirements for SAS/ACCESS Interface to PostgreSQL

SAS/ACCESS Interface to PostgreSQL (on SAS Viya) includes SAS Data Connector to PostgreSQL.

For information about supported PostgreSQL versions and requirements, see:

Requirements for SAS/ACCESS Interface to SAP ASE

SAS/ACCESS Interface to SAP ASE (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported SAP ASE versions and requirements, see:

Requirements for SAS/ACCESS Interface to SAP HANA

SAS/ACCESS Interface to SAP HANA (on SAS Viya) includes SAS Data Connector to SAP HANA.

For information about supported SAP HANA versions and requirements, see:
Requirements for SAS/ACCESS Interface to SAP R/3

SAS/ACCESS Interface to R/3 (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported SAP R/3 versions and requirements, see:

Requirements for SAS/ACCESS Interface to Snowflake

SAS/ACCESS Interface to Snowflake (on SAS Viya) requires SAS Foundation and SAS Viya.


Requirements for SAS/ACCESS Interface to Spark

SAS/ACCESS Interface to Spark requires SAS/ACCESS Interface to Hadoop, which is separately licensed.

For information about supported versions and requirements, see:

Requirements for SAS In-Database Technologies for Spark

SAS In-Database Technologies for Spark requires SAS In-Database Technologies for Hadoop, which is separately licensed.

SAS Embedded Process for Spark is included with SAS In-Database Technologies for Spark. SAS Embedded Process must also be installed on the Spark cluster to enable SAS Data Connect Accelerator for Spark. SAS recommends installing the latest version of SAS Embedded Process.

The CAS controller communicates with the Hadoop NameNode and YARN nodes. SAS Embedded Process is installed on nodes where an instance of YARN Node Manager is running. All Embedded Process tasks that are running on YARN nodes communicate with the CAS controller before they communicate with a CAS worker. The Embedded Process tasks access the CAS grid using the CAS network port, GCPORT. You can set GCPORT to a specific port number when you configure the playbook. If you do not set GCPORT, ephemeral ports are used.

For information about supported versions and requirements, see:

Requirements for SAS/ACCESS Interface to Teradata

SAS/ACCESS Interface to Teradata (on SAS Viya) includes SAS Data Connector to Teradata.

For information about supported Teradata versions and requirements, see:

Requirements for SAS In-Database Technologies for Teradata

SAS In-Database Technologies for Teradata (on SAS Viya) includes SAS Data Connect Accelerator for Teradata and SAS Data Quality Accelerator for Teradata.

SAS In-Database Technologies for Teradata requires SAS/ACCESS Interface to Teradata.

The SAS Embedded Process for Teradata is included with SAS In-Database Technologies for Teradata.
For information about supported Teradata versions and additional requirements, see:

**Requirements for SAS/ACCESS Interface to Vertica**

SAS/ACCESS Interface to Vertica (on SAS Viya) includes SAS Data Connector to Vertica.

For information about supported Vertica versions and requirements, see:

---

**Security Requirements**

**LDAP Requirements**

LDAP is required only for SAS Viya visual interfaces. It is not required in a programming-only deployment.

**For Full Deployments**

- SAS Viya must have Read access to your LDAP server.
- SAS Viya requires a userDN and password in order to bind to the LDAP server. Anonymous binding is supported for clients that are authenticating to the LDAP server.
- If the mail attribute is specified for LDAP accounts, it must have a non-null value that is unique for each user.
- LDAPS is supported, but the required certificates are not configured automatically by the deployment process.

For most user interfaces, configuring SAS Viya to access your LDAP server is a post-deployment task. For more information, see “Configure the Connection to Your Identity Provider” on page 122.

**Additional LDAP Requirements for Multi-tenant Environments**

LDAP and a full deployment are required to support multi-tenancy. You cannot use multiple LDAP servers for a single tenant. However, you can use one LDAP server per tenant, or you can use a single LDAP server for all tenants.

In the previous release of SAS Viya, each tenant required a separate organizational unit (OU). However, you can now specify custom LDAP properties for each tenant. A separate OU per tenant and an OU named “provider” are required only when you use a single LDAP server for all tenants, and only when you use the same LDAP directory structure for all tenants.

You should either set up or plan your tenant structure in LDAP before you start the deployment. Determine whether you will use the same directory structure for the users and groups within all tenants (a “fixed” LDAP structure), or will use a custom structure that varies per tenant. Based on these decisions, you can then perform tenant onboarding as a post-deployment task.

Multi-tenancy requires some post-deployment configuration in SAS Environment Manager. By default, the values that you specify for tenant LDAP connection parameters are automatically applied to the provider and to the users and groups within all tenants. However, you will have an option to selectively apply LDAP connection settings. This option enables you to deploy a custom directory structure for each tenant. For example, you can use a single LDAP server across all tenants while using custom parameters, such as the baseDNs or search filters, for each tenant. In such a case, you would select the option to **Apply configuration only to this tenant (provider)** for the provider’s group and user connections.
During tenant onboarding, if you select the option to apply the configuration only to the provider, you must use the fixed directory structure that is described in “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46. The reason is that SAS Viya requires an OU for the provider and separate OUs for each tenant if the option is not selected.

SAS Viya also supports a custom LDAP structure per tenant and does not require the use of separate OUs to define the provider and tenants. For more information about your options for setting up tenants in LDAP, see “Set Up Accounts for Multi-tenant Deployments: Separate LDAP Server per Tenant” on page 48.

Transport Layer Security

Transport Layer Security (TLS) is applied to many of the network connections in the deployment. These connections are secured by SAS Secrets Manager, which is based on HashiCorp Vault. In a full deployment that is also fully compliant with SAS security standards, the certificates are all signed by a root CA that is generated by SAS Secrets Manager and an intermediate certificate.

In a programming-only deployment, TLS is not enabled by default. In addition, SAS Secrets Manager is not installed. As a result, network connections from the Apache HTTP Server to back-end services are not encrypted. For information about enhancing the default security of a programming-only deployment, see Tasks to Harden Security for Your Linux Programming-Only Deployment in Encryption in SAS Viya: Data in Motion.

How Default Security Is Applied

An Apache HTTP server is used as a reverse proxy server to secure your environment. Default security settings use the Apache mod_ssl module to secure the server with self-signed certificates.

The playbook can automatically install Apache httpd with the mod_ssl module. This option uses default Apache security settings and self-signed certificates. These settings are reasonably secure, but they are not compliant with SAS security standards.

The playbook also inspects any existing certificates and the CA chain to determine whether they comply with SAS security requirements. If compliant certificates are found, they are used without changes. If only the default mod_ssl is found, the playbook generates a self-signed certificate and configures mod_ssl to use it.

You can add your own certificates after the completion of the deployment process, which will require a brief outage. If you do not add compliant certificates and instead keep the default security settings and certificates, end users will see a standard web browser warning message. SAS recommends replacing the certificates before giving end users access to the software.

The deployment process provides a default level of encryption for data at rest (stored data) and for data in motion (transmitted data). However, you should perform several additional actions to increase the level of security on your systems.

Enhance Default Security Settings

SAS recommends that you enhance the default security that is applied by the playbook. As a best practice, follow these steps before you start the deployment process:

1. Install the Apache httpd module and the Apache mod_ssl module on all the web servers in your environment.
2. Add certificates that conform to the policies at your enterprise.
3. Specify the location of the intermediate certificates and the root CA when you edit the playbook. For more information, see “Specify the Path to Certificates” on page 83.
4. Perform a full deployment rather than a programming-only deployment.

The playbook can then enhance the security of your software deployment automatically. It detects the CA chain that is configured for mod_ssl and incorporates it into the truststores for all other machines in your deployment. On machines that are targets for Consul deployment, the playbook performs additional security configuration.
(Optional) You can also perform these actions after the playbook has been run:

- Block external connections to port 80.
- Use HTTPS for access to the user interfaces from a web browser.
- Add custom certificates to the self-signed certificates that are provided on all machines in a full deployment. SAS self-signed certificates are valid for one year. Be sure to renew them before they expire.
- Upgrade the security protocol and ciphers that are enabled by default using the sas-ssl.conf file.
- Prevent administrators from altering the default permissions on subdirectories of `opt/sas/viya`. Use your preferred network monitoring or security tool to monitor permissions on subdirectories of `opt/sas/viya` after the deployment has completed.

For more information about setting up the Apache HTTP Server and configuring additional security settings, see *Encryption in SAS Viya: Data in Motion* in *SAS Viya Administration*.

**Transport Layer Security Requirements for the SAS Embedded Process**

If you are using the SAS Embedded Process, you can secure data transfers between your cluster and CAS. To use Transport Layer Security (TLS) with SAS Embedded Process, the following software is required on each node in the cluster:

- OpenSSL, version 1.0.1g or later
- Appropriate CA certificates to match the server certificates that are configured on the CAS server

**Additional Requirements for Multi-tenancy**

To enable a multi-tenant configuration, you must select a multi-tenant deployment in the sitedefault.yml file before you run the playbook. After the playbook has been run, the system includes a single tenant that enables the provisioning of other tenants and is called the provider. Additional tenants are onboarded as a post-deployment task.

Multi-tenant deployments use Access Control Lists (ACLs) to protect data and configuration of tenants and to restrict access to the provider. The file system that contains the `/opt/sas/viya/config` directories for both tenants and the provider must be mounted with support for ACLs.

You must also set up DNS records for tenant-specific subdomains. As the users within each tenant access SAS Viya components, the host name that they use to access the SAS deployment identifies their tenant membership.

Each tenant is reachable by a URL that is derived from the provider’s URL. Here is the format for a typical tenant URL:

```
tenant-ID.provider-URL
```

Here is an example of a provider URL:

```
sasviya.mycompany.com
```

Here is an example of a tenant URL:

```
mytenant.sasviya.mycompany.com
```

You must verify that the DNS server for your enterprise is configured to route to these address spaces. You can create a wildcard subdomain entry as a time-saving step.

Depending on how you set up LDAP and the connection to SAS Viya, the provider can be accessed in one of two ways after the deployment process has completed:
By logging in with the sasprovider user ID and password. These parameters are configured during tenant onboarding.

This login method is required if you are using multiple LDAP servers for your tenants or if a setting for the provider’s LDAP properties prevents the tenants from inheriting them. To complete tenant configuration, see How To (Provider Administrators) in SAS Viya Administration: Multi-tenancy.

By logging in as one of the users within the provider OU in LDAP.

You can do this if you are setting up tenants within a single LDAP server, and if each tenant has a uniform directory structure and a unique organizational unit (OU) defined, with a separate OU for the provider. To configure the LDAP server before you run the playbook, see “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46.

Enabling YARN Reservations in a Kerberos Environment

SAS Viya supports Kerberos authentication in Apache Hadoop YARN environments. The CAS administrator typically sets a parameter to enable CAS to reserve YARN memory space. For more information, see the configuration file option “cas.USEYARN=true | false” in SAS Viya Administration: SAS Cloud Analytic Services.

However, when the YARN memory reservation is enabled, an error occurs at CAS start-up, specifying that the Kerberos ticket-granting ticket cannot be found. As a workaround for this problem, you can authorize a Service Principal Name (SPN) or User Principal Name (UPN) to access your YARN environment. For details about the workaround, see SAS Note 63457.

User and Group Requirements

Overview: Required Accounts

In addition to an installation account with sudoers privileges, SAS Viya also requires several service accounts. These accounts are created and used like the service accounts that are created by a yum installation of MySQL. These accounts own the SAS Viya files and are used to run various processes.

The deployment process creates the service account named “sas” and adds it to a group named “sas.” The sas account is not used to perform the installation, but it must exist to provide ownership of all the installation files. After the installation has completed, it enables required components to run.

Most of the service accounts for SAS Viya must belong to the sas group. You cannot assign an alternative name to this group. By default, the SAS Viya deployment process will create the sas group as a local Linux group (in /etc/group) on each target computer that you define. However, you can explicitly create the sas group before you run the playbook. You can create it locally (in /etc/group), or in an LDAP scheme that is configured for the authentication provider of the Linux servers that will be your installation targets.

If you create the sas group locally, the group ID (GID) must be consistent across all servers in the SAS Viya environment.

For a list of all accounts, including those for optional SAS products that you might have ordered, see “User Accounts (Reference)” on page 49.

Set Up the User Account that Deploys the Software

The user account that is used to configure and start the deployment process must meet the following requirements:

- Super user (sudo) or root access.

  To verify that your user ID is included in the sudoers file, run the following command:
As an alternative, to verify your sudoers privileges, run this command:

```bash
sudo -v
```

Make sure that commands that can be run as “sudo” are unrestricted on the installation computer.

This user account must be able to access the following accounts as “sudo”: root, sas, and cas. Instructions for configuring the sas and cas accounts are provided in this guide.

- Appropriate permissions to create subdirectories in the directory where you saved the playbook. The recommended path is `/sas/install/sas_viya_playbook`. For more information, see “Store the Playbook” on page 60.
- A home directory.

## Set Up the cas Account

During pre-deployment validation, the playbook checks for a user account named cas and its membership in a group named sas.

This user account is critical to the deployment. As a member of the Super User role in the visual administration interface (SAS Environment Manager), the cas user account has unrestricted access to CAS. It functions as a back-end service account that owns CAS processes. You have the option to use the cas user account as the CAS Admin account.

The playbook does not create the cas user account. The cas user account can belong to multiple Linux groups, but it must also belong to the sas group.

To set up the required group and user account:

1. Create a Linux group named **sas**. A group with this name is required.
   
   This group can either be defined locally (in `/etc/group`) in Linux, or globally in LDAP. The computer must be using LDAP for authentication. You can use Pluggable Authentication Modules (PAM) or another method to enable LDAP authentication.

2. Create a user account. The recommended user name is **cas**.
   
   The cas user account can be defined either locally (in `/etc/passwd`) in Linux, or globally in LDAP. You can use PAM or another method to enable LDAP authentication on each CAS computer.
   
   **Note:** The cas account must exist on all future CAS hosts. If you create this account locally (in `/etc/passwd`), the UID for the cas account must be consistent across all hosts.

3. Make sure that the cas account has a login shell (`/bin/bash` is recommended).

4. Make the cas user a member of the sas group.

5. Verify that the user account exists on each computer where a CAS component will be running. In addition, verify that the account has a consistent UID and GID on all computers in your deployment.
   
   **Note:** Use the `usermod` command to align the UIDs of any mismatched user accounts. For any groups with mismatched GIDs, use the `groupmod` command.

6. If you decide to use an alternative name for the cas user account, configure the alternative account name for the CAS Admin User in the playbook. For more information, see “Set Up the CAS Admin User” on page 87.

An SSH public key is required in the `$HOME/.ssh` directory of the cas user. The playbook will perform this step automatically, or you can configure your own passwordless SSH for the cas user before the deployment. For more information, see “Set Up Passwordless SSH for CAS” on page 85.
For a programming-only deployment, you must assign a password to the cas account if you want to use it to access SAS Studio and CAS Server Monitor. If the cas account is a local account (using `/etc/passwd`), the password must be the same on all machines in your deployment.

**Set Up Additional User Accounts**

Additional user accounts are required in order to configure and run the software after the deployment process has completed. Requirements for user accounts vary according to the deployment type: a full deployment or a programming-only deployment. In a full deployment, you must prepare accounts for users of the visual interfaces. Additional requirements apply to users who require compute server functionality and to users in a programming-only deployment. For more information, see “Deployment Types” on page 4.

**Set Up Accounts for Users of the Visual Interfaces**

**Note:** This section applies only to a full deployment.

The following requirements apply to user accounts that access SAS Viya visual interfaces, including SAS Drive, SAS Environment Manager, and SAS Studio 5.1:

- Each user must be able to authenticate to the LDAP provider.
- (Optional) If you plan to configure front-end single sign-on (SSO), each user must be able to authenticate to the associated provider. This is an additional requirement rather than a replacement for the preceding requirement.
- User accounts that use the compute server must also meet the requirements that are listed in “Set Up Accounts for Compute Server Users” on page 45.

Users of the visual interfaces can also access the SAS Studio 4.4 programming-only interface. However, they must also meet the requirements that are listed in “Set Up Accounts for Programmers” on page 46.

**Set Up Accounts for Compute Server Users**

**Note:** This section applies only to a full deployment.

The SAS compute server and compute service enable users to submit SAS programs and jobs for processing. Several SAS Viya products use the compute service to process programs and statements that were written in the SAS language. For more information, see SAS Compute Server and Service in SAS Viya Administration: Programming Environment Servers.

Compute servers are launched in the operating system under the user ID of the requesting user. The SAS Logon service passes user credentials from the visual interfaces to the machines where compute servers are running. Therefore, the authentication scheme must be configured to enable the operating system on each compute server machine to recognize the users’ LDAP identities. For example, you can use PAM or SSSD to enable this integration.

The following products use compute server functionality: SAS Studio 5.x, SAS Visual Data Mining and Machine Learning, SAS Model Manager, SAS Visual Forecasting, SAS Decision Manager or SAS Intelligent Decisioning, and SAS Analytics for IoT.

Here are the requirements for the users of these products:

- LDAP accounts for the visual interfaces
- Host accounts that match their LDAP accounts on machines where the compute server is installed
- Consistent UIDs and GIDs on all machines in the deployment
- A home directory that can be accessed by the compute server each time that a process starts
If home directories will be local to the machine target for the compute server installation, you can enable the automatic creation of these directories. For more information, see “Enable Directory Creation for Users of the Compute Server” on page 125.

Set Up Accounts for Programmers

Note: This section applies only to a programming-only deployment.

In a programming-only deployment, users will not log on to the SAS Viya visual interfaces. Instead, they will log on to SAS Studio 4.4. Here are the requirements for programmers:

- A valid host account on the machine where the SAS Studio 4.4 web application runs
  
  Note: Any account that exists only on an LDAP server cannot log on to SAS Studio by default.

- A home directory

- Consistent UIDs and GIDs for these user accounts on all machines in the deployment

Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants

Note: This section applies only to a full deployment.

The LDAP directory structure that is described here is required if you decide to apply the same connection settings to the provider and to users and groups within all tenants during tenant onboarding. These steps are recommended for an environment in which a single LDAP server is used for all tenants. You can also use a custom structure that applies different settings to different tenants, and you can use a separate LDAP server per tenant. For more information, see “Set Up Accounts for Multi-tenant Deployments: Separate LDAP Server per Tenant” on page 48.

To configure the LDAP server:

1. Create the provider OU. Here is an example:

```
dc=example,dc=com
   ou=tenant-1
   ou=groups
   ou=users
ou=tenant-2
   ou=groups
   ou=users
...
ou=provider
   ou=groups
   ou=users
```

Here is an example that uses LDIF syntax:

```
dn: cn=sas,ou=groups,ou=provider,dc=sas,dc=com
   distinguishedName: cn=sas,ou=groups,ou=provider,dc=sas,dc=com
   displayName: Tenant-admin-group-for-provider
   gidNumber: value
   objectClass: groupOfUniqueNames
   objectClass: extensibleObject
   uniqueMember: uid=sas,ou=people,ou=provider,dc=sas,dc=com
   cn: sas

Note: The provider DN must be specified as provider.
```

2. If you did not create the sas group in LDAP, verify that the sas group has a consistent GID across all machines in the deployment.
You were instructed to create the sas group in “Set Up the cas Account” on page 44. Multi-tenant deployments apply extra security in the /opt/sas/viya/config directory. To ensure that users within the provider tenant can access critical services (including SAS Studio and the SAS compute server), all provider users must be members of the sas group. If they are not members of that group, the additional security will prevent them from using those servers and services.

3 For each tenant user that you define in LDAP, the following requirements apply:
   - Each uidNumber with gidNumber attributes has been specified.
   - Each user ID, across all tenants, is unique.
   - The homeDirectory attribute is set to a value that is appropriate for your environment.
   - The loginShell attribute is set to /bin/bash.

4 (Optional) Configure a cas user within the provider OU. The cas user can exist as a local user account, but it must have a consistent UID and GID across all machines in the deployment.

   Here is an example that uses LDIF syntax:

   ```
   dn: uid=cas,ou=people,ou=provider,dc=sas,dc=com
   uid: cas
   cn: cas
   sn: Admin
   loginShell: /bin/bash
   distinguishedName: uid=cas,ou=people,ou=provider,dc=sas,dc=com
   displayName: CAS-Administrator-and-process-owner
   userPassword: password
   objectClass: inetOrgPerson
   objectClass: extensibleObject
   uidNumber: value
   gidNumber: value
   homeDirectory: /home/cas
   ```

   You created the cas user (or its equivalent) in “Set Up the cas Account” on page 44. In addition to this optional LDAP user, an identical account must exist on each host.

5 If you are using the sitedefault.yml file for another purpose, remove the sas.identities.providers.ldap.connection block from the file before you run the playbook. You will instead use SAS Environment Manager to configure the LDAP connections as a post-deployment step. SAS recommends using this block in the sitedefault.yml file only to automate a deployment.

6 (Optional) If your provider or tenants will have secondary CAS controllers to enable failover, set up a shared file system. For more information, see “Enable a Shared File System” on page 65.

The deployment process automatically creates an internal user account for an administrator within the provider tenant. You can set up separate groups for administrative users and for non-administrative users within each tenant in LDAP, and you can add tenant users to one of these groups. The tenant creation process provides these groups with access to critical files and other resources that are otherwise restricted. The users that are defined within tenants should not be added to the sas group. Tenant users must instead be members of their tenant’s user group.

Tenant onboarding is a post-deployment task. With the directory structure that is described here, do not select the option to Apply configuration only to this tenant (provider) when you perform tenant onboarding. The settings are then applied automatically to all tenants. Any tenants that you add in the future will require similar directory settings. For more information, see How To (Provider Administrators) in SAS Viya Administration: Multi-tenancy.
Set Up Accounts for Multi-tenant Deployments: Separate LDAP Server per Tenant

Note: This section applies only to a full deployment.

SAS Viya supports an environment in which a separate LDAP server is used for each tenant. You can customize the LDAP directory structure or use your existing structure. To prepare your environment for tenant onboarding:

1. Before you run the playbook, plan and document the structure of the tenant spaces in the LDAP servers that you will allocate to tenants.

2. For each tenant user that you define in LDAP, the following requirements apply:
   - Each uidNumber with gidNumber attributes has been specified.
   - Each user ID, across all tenants, is unique.
   - The homeDirectory attribute is set to a value that is appropriate for your environment.
   - The loginShell attribute is set to `/bin/bash`.

3. Create the sas group in each LDAP server. Verify that the sas group has a consistent GID across all machines in the deployment.

   You were instructed to create the sas group in "Set Up the cas Account" on page 44. Multi-tenant deployments apply extra security in the `/opt/sas/viya/config` directory. To ensure that users within the provider tenant can access critical services (including SAS Studio and the SAS compute server), all provider users must be members of the sas group. If they are not members of that group, the additional security will prevent them from using those servers and services.

4. (Optional) Configure a cas user within the provider. The cas user can exist as a local user account, but it must have a consistent UID and GID across all machines in the deployment

   Here is an example that uses LDIF syntax:

   ```
   dn: uid=cas,ou=people,ou=provider,dc=sas,dc=com
   uid: cas
   cn: cas
   sn: Admin
   loginShell: /bin/bash
   distinguishedName: uid=cas,ou=people,ou=provider,dc=sas,dc=com
   displayName: CAS-Administrator-and-process-owner
   userPassword: password
   objectClass: inetOrgPerson
   objectClass: extensibleObject
   uidNumber: value
   gidNumber: value
   homeDirectory: /home/cas
   ```

   You were instructed to create the cas user (or its equivalent) in "Set Up the cas Account" on page 44. In addition to this optional LDAP user, an identical account must exist on each host.

5. If you are using the sitedefault.yml file for another purpose, remove the `sas.identities.providers.ldap.connection` block from the file before you run the playbook.

   You will instead use SAS Environment Manager to configure the LDAP connections as a post-deployment step. SAS recommends using this block in the sitedefault.yml file only to automate a deployment.

6. Once the deployment process has completed, log on to SAS Environment Manager using the sasboot account.

7. Select settings for the following parameters to enable the Identities service to authorize the provider:

   `sas.identities.providers.ldap.connection`
As you configure each parameter, consider whether to select the option to **Apply configuration only to this tenant (provider)**. This option restricts the application of a setting to the provider. If you do not select this option, the values for each parameter are applied to all tenants, and you must set up the directory structure as specified in “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46.

8  **(Optional)** If your provider or tenants will have secondary CAS controllers to enable failover, set up a shared file system. For more information, see “Enable a Shared File System” on page 65.

You can set up separate groups for administrative users and for non-administrative users within each tenant in LDAP, and you can add tenant users to one of these groups. The tenant onboarding process provides these groups with access to critical files and other resources that are otherwise restricted. The users that are defined within tenants should not be added to the sas group. Tenant users must instead be members of their tenant’s user group. For more information about tenant onboarding, see How To (Provider Administrators) in SAS Viya Administration: Multi-tenancy.

**User Accounts (Reference)**

This section provides reference information about user accounts that are required in order to deploy and to perform initial configuration of SAS Viya.

SAS recommends that you use a security management tool, such as Mozilla Network Security Services (NSS) or the Red Hat System Security Services Daemon (SSSD), to manage users and groups. Creating accounts in LDAP and then creating matching local accounts for users and for service ownership is not recommended.

The table identifies and describes SAS Viya user accounts. Because these accounts are required for the installation and for running services during the product’s normal operation, do not delete them or change their names. These user accounts do not require root or sudo privileges.

<table>
<thead>
<tr>
<th>Default Account Name and Group</th>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas; member of sas group</td>
<td>A service account without user restrictions. Required for the installation, and created automatically. A login shell is required. A password is not required. You can add a password after installation, if necessary, but make sure that it does not expire. The default user name is required. The sas group is an administration group, not a general user group.</td>
<td>The installation process sets user and group ownership permissions on all the installation files. This user must exist to enable ownership. After the installation has completed, this user account enables required components to run. The sas group is intended to allow access to administrative features, such as logs and backup. It is the group owner of many files on disk. Restrict membership in this group to administrators.</td>
</tr>
<tr>
<td>Default Account Name and Group</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>cas; member of sas group</td>
<td>The process owner of CAS processes. Create this user account and add it to the sas group before you start the deployment. A login shell is required. No default password is assigned, but a password is required if you plan to use this account as the CAS administrator. If you are using both operating-system and LDAP accounts, which are required for a full deployment, verify that this user has a single set of credentials that are valid for all applicable authentication providers. In addition, verify that this user is a member of the sas group. The cas user must be able to connect from the CAS controller to each CAS worker without providing a password. If the CAS server is running in an environment with multiple CAS workers, passwordless SSH can be configured by the Ansible playbook. For more information, see &quot;Accept the Passwordless SSH Default&quot; on page 85. The &quot;cas&quot; user name is recommended. This user name enables the deployment to assign SSH keys. To assign a different user name, modify the casenv_user parameter in the vars.yml file.</td>
<td></td>
</tr>
<tr>
<td>sasboot</td>
<td>Created during the deployment, with an expired password. After the deployment has completed, use this account to log on to the SAS visual interface in order to configure the connection to your identity provider and to set up user accounts. The sasboot account is typically not used after the accounts have been set up. However, it provides an indirect login option in case your identity provider becomes unavailable. The sasboot user is internal only to SAS. It does not exist on a host or in LDAP. For more information, see SAS Viya Administration: Identity Management.</td>
<td></td>
</tr>
<tr>
<td>user account for SAS Studio access</td>
<td>A user account that is defined on the operating system of the machine where SAS Studio will be installed (a host account).</td>
<td></td>
</tr>
</tbody>
</table>

Purpose:
- **cas; member of sas group**: Required for managing and enabling CAS. This user corresponds to the CAS (Superuser) role in the CAS administration interface, CAS Server Monitor.
- **sasboot**: Administrator account that is used for preliminary logon to the visual administration interface. This account is not recognized by SAS Studio, which is the programming interface.
- **user account for SAS Studio access**: User account that is used for preliminary logon to the programming administration interface.
User account that enables SAS Model Manager workflow clients to make service task calls. This account executes all service tasks for any workflow process that has been assigned the **SAS Model Manager** client identifier. (SAS Analytics for IoT has the same requirement.)

Use SAS Environment Manager to assign this account to the Workflow service after the deployment process has completed. You must also log in to one of the SAS Viya user interfaces as this user in order to obtain an initial OAuth token. For more information, see “Configure the Workflow Client User” on page 155.

The required user account named sas is created automatically by the deployment process. This user is the file owner of many of the files that are copied to the machine by the installation RPMs. When the deployment has completed, verify that the sas user account includes a login shell.

The sas user account enables required components to run, including the web application server for SAS Event Stream Processing Studio. For SAS Event Stream Processing, the installation directory path enables Write access per user group, and it is owned by the sas user. To grant permission to edit the configuration files, the administrator must add any user requiring write access to these files to the sas group. SAS Event Stream Manager users are authenticated in LDAP when they log on.

An SSH key for the sas user account is required in order to enable the SAS Infrastructure Data Server. This key is created during the deployment and is delivered to every pgpoolc and sasdatasvrc host that is listed in your inventory.ini file.

The following additional groups are required to support third-party components and are also added to `/etc/group` automatically:

- `apache`
- `postgres`
An additional user account, named sasrabbitmq, is created automatically as the owner of the RabbitMQ component. This component is also added to /etc/passwd automatically.

Services That Require Root Privileges

When the deployment process has completed, several services are automatically configured to run with root privileges. Do not downgrade (change from root privilege to another privilege) any of the following services. Doing so would result in an inoperable environment:

- **Apache httpd**—Supports the Apache HTTP Server, which provides security and load balancing for multiple SAS Viya components.
- **consul-template**—Supports SAS Configuration Server, which is based on HashiCorp Consul. SAS Configuration Server is a registry that contains service configuration data and status information. The consul-template process extracts configuration change data from the server and updates the appropriate service configuration file.
- **identsvcs and launchsvcs**—Authorize and perform the launching of the CAS server session. These services must run as root because, on Linux, the root identity is required in order to start a running process under a different identity. The launchsvcs process creates a CAS session under the identity of the user who submitted the request. The identsvcs process authenticates users when they attempt to connect to a CAS server with a user name and password using PAM.
- **RabbitMQ**—Supports SAS Message Broker, which is a message service that is based on Pivotal RabbitMQ. SAS Message Broker manages and routes messages among SAS Viya components.
  
  **Note:** SAS Message Broker is not used in a programming-only deployment.
- **vault**—Supports SAS Secrets Manager, which is based on HashiCorp Vault. It stores and generates secrets such as certificates.

Some supporting utilities that are included with SAS Viya must run with temporary root privileges. After the deployment has completed, the setuid bit is set to “root” by default for the following executables:

- **caslaunch**—Starts a CAS server and its supporting processes (identsvcs and launchsvcs). It reads and executes only one simple configuration file, which is under the control of root.
  
  **Note:** The setuid root requirement applies to this executable under one of these conditions: if the CAS session processes run under an individual user identity or if the CAS session processes authenticate to a source that requires root access. The PAM configuration might not require root privileges.
- **elssrv**—Launches processes via the identity of the requesting client or via a set of stored credentials.
- **sasauth**—Enables SAS Viya servers to authenticate connecting clients.
- **sasperm**—Delivered with the SAS Viya deployment, but is not used.

Client Requirements

Web Browsers

End users can access the product user interfaces for SAS Viya applications from a desktop computer, using a supported web browser. Because SAS software is not installed on this machine, the requirements are minimal. UNIX and 64-bit Windows operating systems are supported.

Some SAS Viya user interfaces include some advanced features that require recent versions of popular web browsers. For information about supported web browsers and the corresponding platforms to access SAS user interfaces, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-web-browsers.html.
Mobile Platform and Touchscreen Support

The SAS Visual Analytics Apps run natively on iOS, Android, and Windows 10, and provide the ability to view and explore reports using a touchscreen.

Some SAS Viya user interfaces are not currently supported on mobile devices.

For more information about mobile device support, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-web-browsers.html.

Database Drivers

Make sure that each client where users will access SAS software has the required database drivers already installed.

Screen Resolution

The minimum screen resolution for each client machine that will access the SAS Viya user interfaces is 1280 x 1024.

Deployment Tools

Ansible Controller Requirements

A typical Ansible deployment consists of at least one control machine (the Ansible controller) and multiple Ansible managed nodes (the machines where SAS software is installed). In a single-machine deployment, Ansible and all SAS software are installed on the Ansible controller. For more information, see “Install Ansible” on page 70.

In a distributed deployment, the managed nodes use a secure shell (SSH) framework for connections to the Ansible controller. Verify network connectivity between the controller and the managed nodes. Connectivity is also required among all machines in the deployment and from the controller to the SAS yum repositories.

For information about supported Ansible versions and other requirements, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-operating-systems.html#ansible.
Pre-installation Tasks

- **Create a Mirror Repository**
  - SAS Mirror Manager and the Mirror Repository
  - Using SAS Mirror Manager with a Proxy Server
  - Specify a Log Location
- **Create a Playbook**
  - Download the SAS Orchestration CLI
  - Create a Playbook with the SAS Orchestration CLI
  - Store the Playbook
- **Enable Required Ports**
  - Ports to Be Made Available
- **Configure SELinux**
  - Options for Deploying SAS Viya with SELinux
  - Disable SELinux and Deploy SAS Viya
  - Configure the Environment to Accommodate SELinux
  - Disable SELinux, Deploy SAS Viya, then Re-enable SELinux
- **Configure a Proxy Server**
  - Overview
  - Using curl
  - Using yum
- **Enable the Yum Cache**
- **Enable a Shared File System**
- **Perform Linux Tuning**
  - Set the MaxStartups Value
  - Set the ulimit Values
  - Set the Semaphore Values
  - Tune Memory Capacity
  - Change the Default Time-outs
  - (SUSE Linux Only) Change the Maximum Number of Operating System Tasks
- **Confirm the Identities of the Hosts**
- **(Optional) Enable Key-Based SSH Authentication**
- **Install Ansible**
  - Standard Ansible Installation
  - Streamlined Ansible Installation for Red Hat Enterprise Linux and Equivalent Distributions
  - Streamlined Ansible Installation for SUSE Linux
  - Test Your Ansible Installation
Create a Mirror Repository

Note: The process for creating a mirror repository for SAS Viya 3.4 is different from the one used in previous versions. If you are familiar with earlier versions of SAS Viya, you should not assume any similarities with the process used by those versions.

SAS Mirror Manager and the Mirror Repository

SAS Mirror Manager is a command-line utility for synchronizing a collection of SAS software repositories. Its primary use is to create and manage mirror repositories for software deployment. A mirror repository is required for all SAS Viya deployments on SUSE Linux. For Red Hat Enterprise Linux, a mirror repository is optional and should be used if your deployment does not have access to the internet or if you must always deploy the same version of software (such as for regulatory reasons). In addition, if you intend to eventually add tenants or additional CAS servers to your deployment, use a mirror repository to ensure that the same software is deployed on each machine.

As you select a location for your mirror repository, keep in mind that SAS Mirror Manager can be used to place the files in several locations, such as on a web server that serves the files by HTTP, or on a shared NFS mount. The default location for the download is the `sas_repos` directory of the installation user. Ensure that the default location or the location that you select has adequate space. Also ensure that the machine where the mirror repository will be located has adequate space.

To create a mirror repository with SAS Mirror Manager:

1. The Software Order Email (SOE) indicated that you should save the SAS_Viya_deployment_data.zip file attachment. If you have not already saved the file, save it now.

2. Download SAS Mirror Manager from the SAS Mirror Manager download site to the machine where you want to create your mirror repository. If you use Internet Explorer to download the Linux or Macintosh version, save the file as a .tgz file instead of a .gz file.
   
   Note: This step requires internet connectivity. If you receive warnings or errors regarding connectivity, see “Internet Connectivity Problems” on page 292.

3. Uncompress the downloaded file.

4. Run the following command:
   
   Note: Enter the command on a single line. Multiple lines are used here to improve readability.
   
   ```bash
   ./mirrormgr mirror --deployment-data path-to-deployment-zip-file-from-SOE --platform Linux-distribution-value --latest
   ```

   Here are the values that can be used for the --platform option for Linux:

   - Use `x64-redhat-linux-6` for all supported versions of Red Hat Enterprise Linux and its equivalent such as Oracle Linux.
   - Use `x64-suse-linux-12` for all supported versions of SUSE Linux.

   By default, the repositories are placed in the `sas_repos` directory in the installation user’s home directory. You can change this location by using the --path option, followed by the full directory location of the mirror destination. This guide refers to that location as `sas_repos`. However, if you want to use a different location, replace instances of `sas_repos` that are used in this guide with the actual location that you select.

   ```bash
   ./mirrormgr mirror --deployment-data path-to-deployment-zip-file-from-SOE --path location-of-mirror-repository --platform Linux-distribution-value --latest
   ```
The `sas_repos` directories are explained as follows:

- The entitlements.json is a list of the repositories to which you are entitled.
- The location_group_declarations.json file and the `sasmd` directory contain data that is used by the SAS Orchestration CLI to create the order-specific tools for your deployment.
- Any remaining directories are the software repositories, organized by native deployment tools:
  - `repos` contains yum files for Linux.
  - `win` contains MSI files for Windows.
  - `deb` contains APT files for Debian.

5 (Optional) After the initial download is complete, move the file structure to a web server or shared NFS mount. The destination machine does not have to be connected to the internet.

You can use tools like rsync and scp to move the files. Here is a typical command for rsync:

```
rsync -av --progress sas_repos target_machine:/var/www/html/pulp/
```

(Optional) If you are using Red Hat Satellite, you can work with your system administrator to move the files to your Red Hat Satellite Server.

### Using SAS Mirror Manager with a Proxy Server

If your environment requires a proxy server and is set up to use it, the SAS Mirror Manager commands will work automatically. However, if your environment is not set up to send data through the proxy, you can add an environment variable to the command to run SAS Mirror Manager. The environment variable identifies where the proxy is located and what is required to send data through it.

Use the environment variable that is appropriate for the target of the query that passes through the proxy. For example, if you are trying to reach a SAS repository, use the HTTPS environment variable because the SAS repository is on an HTTPS site. In most cases, the HTTPS environment variable is appropriate.

Here are some examples of SAS Mirror Manager commands that include environment variables.

**Note:** Specify these commands on a single line. Multiple lines are used here to improve readability.

**Example 1:** An HTTPS site.

```
https_proxy=http://user-name:password@internet-proxy-server-FQDN:proxy-port
```

**Example 2:** HTTPS with the certificate location.

If you use the `https_proxy` variable, the run command for SAS Mirror Manager might also require the `--cacert` option, which indicates the location of the certificate that the proxy must use. The proxy certificate will be one that your organization manages. Here is an example of the environment variable and the run command for SAS Mirror Manager used together.

```
https_proxy=https://proxyid:password@proxy.company.com:3129 /opt/sas/viya/home/bin/mirrormgr mirror --deployment-data SAS_Viya_deployment_data.zip --platform x64-redhat-linux-6 --path sas_repos --cacert ../proxycert.crt --latest
```

**Example 3:** An HTTP site.

```
http_proxy=http://user-name:password@internet-proxy-server-FQDN:proxy-port
```

**Example 4:** An HTTP site with the environment variable and the run command for SAS Mirror Manager used together.

```
http_proxy=http://proxyid:password@proxy.company.com:443 /opt/sas/viya/home/bin/mirrormgr mirror --deployment-data SAS_Viya_deployment_data.zip --platform x64-redhat-linux-6 --path sas_repos --latest
```
Specify a Log Location

The default location for SAS Mirror Manager logs is `user-home-directory/.local/share/mirrormgr/mirrormgr.log`. To specify an alternative log location:

```
./mirrormgr mirror --deployment-data path-to-deployment-zip-file-from-SOE --path location-of-mirror-repository --log-file location-of-mirror-repository/mirrormgr.log --platform Linux-distribution-value --latest
```

Here are the values that can be used for the `--platform` option for Linux:

- Use `x64-redhat-linux-6` for all supported versions of Red Hat Enterprise Linux and its equivalent such as Oracle Linux.
- Use `x64-suse-linux-12` for all supported versions of SUSE Linux.

Create a Playbook

The SAS Orchestration Command Line Interface (CLI) uses the order information that was included in your Software Order Email (SOE) to create a playbook for deploying your SAS Viya software. The SAS Orchestration CLI can be run on Linux or Windows and it requires the Java Runtime Environment 1.8. It also requires access to the internet, unless you are deploying from a mirror repository.

Before you use the SAS Orchestration CLI, ensure that the SAS_Viya_deployment_data.zip file attachment from your SOE is copied to a directory on a machine that runs the Linux, Macintosh, or Windows operating system.

Download the SAS Orchestration CLI

1. The SOE indicated that you should save the SAS_Viya_deployment_data.zip file attachment. If you have not already done so, save that file now.

2. Go to the [SAS Orchestration CLI download site](#) and download the SAS Orchestration CLI for the operating system where you stored the ZIP file.

   The SOE recommended that you save the ZIP file to a machine that runs Linux, which is where you will install your SAS Viya software. However, you can also store it on a machine that runs Macintosh or Windows. If you use Internet Explorer to download the Linux or Macintosh version, save the file as a .tgz file instead of a .gz file.

   **Note:** This step requires internet connectivity. If you receive warnings or errors regarding connectivity, see “Internet Connectivity Problems” on page 292.

3. Uncompress the TGZ file (Linux and Macintosh) or ZIP file (Windows) in the same location where you downloaded it. The result is a file named sas-orchestration on Linux or Macintosh or a file named sas-orchestration.exe on Windows.

Create a Playbook with the SAS Orchestration CLI

Basic Command

To create a playbook, use the command that is appropriate for the operating system where the SAS Orchestration CLI is located.
Note: The following commands are organized by the operating system where the SAS Orchestration CLI will run, rather than by the operating system where your SAS Viya software will be deployed. After you create the playbook, you can move it to the machine where you will deploy your software.

**Linux or Macintosh**

`./sas-orchestration build --input location-of-ZIP-file-including-file-name --platform deployment-platform-tag`

**Windows**

`.\sas-orchestration.exe build --input location-of-ZIP-file-including-file-name --platform deployment-platform-tag`

For `deployment-platform-tag`, if you deploy to Red Hat Enterprise Linux or an equivalent distribution, such as Oracle Linux, specify `redhat`. If you deploy to SUSE Linux, specify `suse`.

Using the SAS Orchestration CLI creates a new file named `SAS_Viya_playbook.tgz`.

**Options**

**Use a Proxy Server**

If you use an unauthenticated proxy to reach the internet, you must add the following options to the run command in order to make an outgoing connection:

--java-option "-Dhttps.proxyHost=proxy-server-IP-address-or-host-name"
--java-option "-Dhttps.proxyUser=user-name"
--java-option "-Dhttps.proxyPassword=user-password"

In addition, if the proxy server is not using the default proxy port of 80, you must also add the following option:

--java-option "-Dhttps.proxyPort=proxy-server-port-number"

For normal usage, when communicating with SAS, the options must start with `Dhttps`. If you have set up a mirror behind an HTTP server without TLS, then the options must start with `Dhttp`.

The Java options should not be combined into a single option. Here is an example of using the options on a Linux machine:

`./sas-orchestration --java-option "-Dhttps.proxyHost=my.proxy.com" --java-option "-Dhttps.proxyPort=1111" --java-option "-Dhttps.proxyUser=proxy" --java-option "-Dhttps.proxyPassword=password123" build --input /tmp/SAS_Viya_deployment_data.zip`

The `--java-option` tags must come before the `build` command.

**Specify the Deployment Type**

You can perform either of two types of deployment for SAS Viya software:

- Full deployment
  - Includes all the software to which you are entitled. A full deployment is required to enable multi-tenancy.

- Programming-only deployment
  - Excludes SAS Drive, most graphical user interfaces, and most services. It is the simplest and smallest type of deployment.

SAS typically recommends a full deployment. However, if your software includes SAS Data Science, it must be deployed as programming-only.

To deploy your software as programming-only, add the `--deployment-type` option and the programming value. Here is an example:

`./sas-orchestration build --input /sas/install/SAS_Viya_deployment_data.zip --platform redhat`
For more information about the deployment types, see “Deployment Types” on page 4.

Use a Mirror Repository

If you created a mirror repository with SAS Mirror Manager, you must include its location with the --repository-warehouse option.

```
./sas-orchestration build --input /sas/install/SAS_Viya_deployment_data.zip --platform redhat --repository-warehouse "URL-to-mirror-repository-content"
```

Here is an example for Windows:

```
./sas-orchestration build --input c:\sas\install\SAS_Viya_deployment_data.zip --repository-warehouse file:///sas_repos
```

Note: The repository warehouse URL must be available to all hosts in the deployment to retrieve packages from the repositories. For example, if the repository warehouse is file-based, then that location should be shared across hosts and should be shared at the same path on each of those hosts. For more information about URLs, consult with your system administrator.

For more information about SAS Mirror Manager, see “Create a Mirror Repository” on page 56.

Help with the Options

The SAS Orchestration CLI includes several options. To learn about all the options for the SAS Orchestration CLI, use the appropriate command:

**Linux or Macintosh**

```
./sas-orchestration build --help
```

**Windows**

```
./sas-orchestration.exe build --help
```

Store the Playbook

1. If necessary, move the SAS_Viya_playbook.tgz file to a directory on your Ansible controller that can be read by other users. The recommended location is /sas/install.

2. In the same directory where you have saved the playbook, uncompress it.

   ```
tar xf SAS_Viya_playbook.tgz
   ```

In addition, SAS recommends that you create a directory on each machine in your deployment for storing files that are used to deploy and maintain your software. The best practice is to use the same directory location on each machine. SAS recommends using /sas/install. This guide assumes that you will use /sas/install. However, if you do not use /sas/install, replace those instances in this guide with the actual location that you select.

Enable Required Ports

The following ports are used by SAS Viya and should be available before you begin to deploy your software. The same ports should also be available for any firewalls that are configured on the operating system or the network.
<table>
<thead>
<tr>
<th>Process</th>
<th>Required Port</th>
<th>Requires Allowed Inbound Traffic From</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache HTTP Server</td>
<td>80, 443</td>
<td>anywhere (SAS Viya servers, workstation)</td>
<td></td>
</tr>
<tr>
<td>Erlang Port Mapper Daemon (epmd) port</td>
<td>4369</td>
<td>SAS Viya servers only</td>
<td>For a single server deployment with no failover, these ports must be opened in order to use the PostgreSQL tools, pgAdmin and pgpoolAdmin. Each additional standby node gets the next sequentially available port, up to 5439.</td>
</tr>
<tr>
<td>SAS Infrastructure Data Server</td>
<td>5430–5439</td>
<td>SAS Viya servers only</td>
<td>For a single-server deployment with no failover, these ports must be opened in order to use the PostgreSQL tools, pgAdmin and pgpoolAdmin. Each additional standby node gets the next sequentially available port, up to 5439.</td>
</tr>
<tr>
<td>SAS Common Planning Data Server</td>
<td>5440-5449</td>
<td>SAS Viya servers only</td>
<td>For a single-server deployment with no failover, these ports must be opened in order to use the PostgreSQL tools, pgAdmin and pgpoolAdmin. Each additional standby node gets the next sequentially available port, up to 5449.</td>
</tr>
<tr>
<td>CAS Server</td>
<td>5570</td>
<td>SAS Viya servers and workstations</td>
<td>Used by clients to make binary connections to CAS.</td>
</tr>
<tr>
<td>SAS Message Broker</td>
<td>5671, 5672, 15672, 25672</td>
<td>SAS Viya servers only</td>
<td></td>
</tr>
<tr>
<td>SAS Studio</td>
<td>7080</td>
<td>SAS Viya servers only</td>
<td>Required only for SAS Studio 4.4 and for programming-only deployments.</td>
</tr>
<tr>
<td>SAS Secrets Manager</td>
<td>8200</td>
<td>SAS Viya servers only</td>
<td></td>
</tr>
<tr>
<td>SAS Configuration Server</td>
<td>8300, 8301, 8302, 8500, 8501</td>
<td>SAS Viya servers only</td>
<td>SAS uses HashiCorp Consul as its configuration server. Ports 8301 and 8302 must be open for both UDP and TCP traffic.</td>
</tr>
<tr>
<td>Object Spawner</td>
<td>8591</td>
<td>SAS Viya servers only</td>
<td></td>
</tr>
<tr>
<td>CAS Server Monitor</td>
<td>8777</td>
<td>anywhere (SAS Viya servers)</td>
<td>Used by clients to make REST HTTP calls to CAS, as with the Python REST interface.</td>
</tr>
<tr>
<td>SAS/CONNECT Spawner</td>
<td>17551</td>
<td>anywhere (SAS Viya servers, SAS 9.X servers, workstation)</td>
<td></td>
</tr>
</tbody>
</table>
### Process Table

<table>
<thead>
<tr>
<th>Process</th>
<th>Required Port</th>
<th>Requires Allowed Inbound Traffic From</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Model Manager or SAS Analytics for IoT launcher context</td>
<td>18201–18250</td>
<td>SAS Viya servers only</td>
<td>Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.</td>
</tr>
<tr>
<td>SAS Job Execution launcher context</td>
<td>18501–18600</td>
<td>SAS Viya servers only</td>
<td>Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.</td>
</tr>
<tr>
<td>SAS Visual Forecasting or SAS Analytics for IoT launcher context</td>
<td>18601–19000</td>
<td>SAS Viya servers only</td>
<td>Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.</td>
</tr>
</tbody>
</table>

If your order included SAS Event Stream Processing, any ports that will be used for ESP servers must be open to HTTP traffic. These ports are also required to enable the streaming analytics features of SAS Analytics for IoT. For more information, see [Setting Up and Using the ESP Server](#) in SAS Event Stream Processing.

The Linux operating system defines a specific series of network service ports as an ephemeral port range. These ports are designed for use as short-lived IP communications ports and are allocated automatically from within this range. For example, if you do not specify a port for **GCPORT** in the vars.yml file, the CAS server uses ephemeral ports for communications among the controller and workers by default. If a required port is within the range of the ephemeral ports for a host, another application can attempt to claim it and cause services to fail to start. Therefore, you must exclude the required ports in the table from the ports that can be allocated from within the ephemeral port range.

1. To determine the active ephemeral port range, run the following command on your host:
   ```bash
   sudo sysctl net.ipv4.ip_local_port_range
   ```
   The results contain two numbers:
   ```
   net.ipv4.ip_local_port_range = inclusive-lower-limit inclusive-upper-limit
   ```

2. To list any existing reserved ports, run the following command:
   ```bash
   sudo sysctl net.ipv4.ip_local_reserved_ports
   ```
   Here is an example of the results:
   ```
   net.ipv4.ip_local_reserved_ports = 23, 25, 53
   ```
   If no ports are reserved, no ports are listed in the results:
   ```
   net.ipv4.ip_local_reserved_ports =
   ```

3. After you determine the limits of the ephemeral port range, you must add any required ports from the table that are included in your ephemeral port range to the Linux system reserved ports list. Add ports to the reserved list as comma-separated values or as a range within quotation marks:
   ```bash
   sudo sysctl -w net.ipv4.ip_local_reserved_ports="ports-or-port-range"
   ```
   Here is an example:
   ```bash
   sudo sysctl -w net.ipv4.ip_local_reserved_ports="5672,15672,25672,4369,16060-16069,9200"
   ```
   Note: The `sysctl` command numerically sorts the port numbers regardless of the order that you specify.
Add an entry to the `/etc/sysctl.conf` file to make your changes permanent. Here is an example:

```bash
net.ipv4.ip_local_reserved_ports = 4369,5672,9200,15672,16060-16069,25672
```

## Configure SELinux

If Security-Enhanced Linux (SELinux) is enabled in your environment, it must be disabled or accommodated before you can use Ansible to deploy SAS Viya.

### Options for Deploying SAS Viya with SELinux

If SELinux is required in your environment, it is supported on Red Hat Enterprise Linux if you perform some additional tasks. On SUSE Linux, SELinux is not supported.

To determine the present status of SELinux in your environment:

```
sudo sestatus -v
```

- If you see a message that the command is not enabled, SELinux is not active in your environment. You can use Ansible to deploy SAS Viya.
- If a mode that is not `permissive` is returned, SELinux is enabled. In order to deploy SAS Viya, you must select one of the following options:
  - Disable SELinux and deploy SAS Viya.
  - Configure the environment to accommodate SELinux.
  - Disable SELinux, deploy SAS Viya, and then re-enable SELinux after the deployment has completed.

### Disable SELinux and Deploy SAS Viya

If SELinux is active in your environment, one option is to configure `permissive` mode for SELinux on all the target machines in your deployment.

**Note:** Permissive mode effectively disables SELinux.

To change the mode value to `permissive` on all target machines in your deployment:

```
sudo setenforce 0
sudo sed -i.bak -e 's/SELINUX=enforcing/SELINUX=permissive/g' /etc/selinux/config
```

### Configure the Environment to Accommodate SELinux

To deploy SAS Viya with SELinux enabled on all target machine in your deployment, perform all the following tasks:

1. Disable the pre-installation check that determines whether SELinux is active. For more information, see “Verify System Requirements” on page 82.

2. Configure SELinux to enable the Apache HTTP Server. By default, SELinux does not allow the Apache httpd component to access the network.

   Run the following command on any machines that are deployTargets for the `[httpproxy]` host group in the `inventory.ini` file:

   ```bash
   sudo setsebool -P httpd_can_network_connect 1
   ```
3 Make sure that the SELinux Policy deny_unknown status is set to allowed. Perform the following steps:
   a Run the following command to determine the current SELinux settings:
      `sudo sestatus -v`
   b Check the value of Policy deny_unknown status in the output. If the value is not allowed, you must change the policy setting.
   c As root, edit the `/etc/selinux/semanage.conf` file.
   d Add the following line:
      `handle-unknown=allow`
   e As root, run the following command to rebuild and reload the policy:
      `semodule -B`

**Disable SELinux, Deploy SAS Viya, then Re-enable SELinux**

A final option is to disable SELinux during the deployment and then re-enable it as soon as the deployment has completed. If you select this option, perform tasks 2 and 3, as described in “Configure the Environment to Accommodate SELinux”.

In addition, if you use key-based authentication to enable Ansible tasks to run on multiple machines without password prompts, perform an additional task. Any SSH keys that were generated while SELinux was in permissive mode will no longer work when you set SELinux back to enforcing mode. You must restore the context for your SSH keys as a post-deployment step. For more information, see “(Optional) Enable Key-Based SSH Authentication” on page 69.

---

**Configure a Proxy Server**

**Overview**

The SAS Viya deployment process uses both curl and yum to download RPM packages from SAS repositories. If your organization uses a forward HTTP proxy server, both curl and yum on each target deployment machine must be configured for forward proxy servers.

Refer to the Linux man pages for yum.conf and curl for more information about proxy settings.

**Using curl**

Curl uses the https_proxy and http_proxy environment variables to send requests to proxy servers. You can export these variables in a new shell profile script such as `/etc/profile.d/httpproxy.sh`. Here is an example of the `/etc/profile.d/httpproxy.sh` script:

```bash
export https_proxy=http://user-name:password@internet-proxy-server-FQDN:8080/
export http_proxy=http://user-name:password@internet-proxy-server-FQDN:8080/
```

In addition, ensure that HTTP requests between machines in the deployment are not routed through the proxy server during deployment by adding the IP addresses, host names, or domains for the SAS Viya machines to the no_proxy variable in your profile.d script. For example, if the SAS Viya machines are using the IP addresses,
10.255.47.131 and 10.255.47.132, and the host names, machine1.example.com and machine2.example.com, you can configure no_proxy as follows:

```bash
export no_proxy="localhost,127.0.0.1,.example.com,10.255.47.131,10.255.47.132"
```

If the profile script is properly configured, these environment variables are set at login for all users. Curl requests for HTTP or HTTPS resources should use the connection information from these variables.

### Using yum

Forward proxy server settings for yum can be configured in `/etc/yum.conf`. Here is an example of the `/etc/yum.conf` script:

```bash
proxy=internet-proxy-server-FQDN:8080/
proxy_username=user-name
proxy_password=password
```

### Enable the Yum Cache

**Note:** SUSE Linux does not use yum as a deployment tool. If you are using SUSE Linux or installing from a local mirror repository, skip this section.

By default, yum deletes downloaded files after a successful operation when they are no longer needed, minimizing the amount of storage space that yum uses. However, you can enable caching so that the files that yum downloads remain in cache directories. By using cached data, you can perform certain operations without a network connection.

In order to enable caching, add the following text to the `[main]` section of `/etc/yum.conf`.

```bash
keepcache = 1
```

This task should be performed on each machine in the deployment.

### Enable a Shared File System

If you are deploying SAS Cloud Analytic Services (CAS) on a massively parallel processing (MPP) system, and if your deployment will include a secondary CAS controller, you should enable a shared file system. The shared file system will be used to store data and configuration information that is used by the primary CAS controller and the secondary CAS controller. However, the shared file system should reside on a machine other than the primary CAS controller or the secondary CAS controller. If the primary CAS controller fails, the secondary CAS controller could then assume the controller role.

**Note:** If you prefer to set up the shared file system after the deployment is complete, skip the steps in this section and see Set Up a Shared File System for CAS Controllers (Post-Deployment) in SAS Viya Administration: SAS Cloud Analytic Services.

1. Identify the machine and the directory location that will be used to house the shared file system.

2. Create the `/opt/sas/viya/config/data/cas` directory on the machines that will be the primary CAS controller and the secondary CAS controller. Set up the `/opt/sas/viya/config/data/cas` directory with the following information:
   - Owner and group of the entire directory path: sas | sas
   - Permissions throughout the entire path: 755
3 Mount the shared file system on the machines that will be the primary CAS controller and the secondary CAS controller. Run the following commands on both machines:

**Note:** Enter the command on a single line. Multiple lines are used here to improve readability.

```
sudo mount IP-address-of-machine-with-shared-file-system:directory-location-of-shared-file-system
/opt/sas/viya/config/data/cas
```

The shared file system is mounted for the CAS server and for the tenant in a single-tenant deployment or for the provider in a multi-tenant deployment. For a multi-tenant deployment, you must repeat similar steps for each tenant that will use a secondary CAS controller. For more information about adding tenants, see Set Up a Secondary CAS Controller (Optional) in SAS Viya Administration: Multi-tenancy.

---

### Perform Linux Tuning

This section describes tuning that should be performed on your Linux machines before you deploy your software. For information about tuning that can be performed after you deploy your software, see Linux in SAS Viya Administration: Tuning.

#### Set the MaxStartups Value

The MaxStartups variable specifies the maximum number of unauthenticated concurrent connections. The default value is 10. Access to non-local (remote) HDFS blocks can result in more than 10 simultaneous connection attempts. Therefore, in Hadoop environments, SAS recommends that you increase MaxStartups to 100 on all nodes in the Hadoop cluster.

1 Open the `/etc/ssh/sshd_config` file.

2 Here is an example for setting the value for the MaxStartups variable to 100:

```
MaxStartups 100
```

In this example, if the value for MaxStartups is in the format of three numbers separated by colons, ensure that the first number is 100.

3 Save and close the `/etc/ssh/sshd_config` file.

4 Restart the sshd service.

   For Red Hat Enterprise Linux 6.7:
   ```
   sudo service sshd restart
   ```

   For Red Hat Enterprise Linux 7.x or SUSE Linux:
   ```
   sudo systemctl restart sshd
   ```

#### Set the ulimit Values

**Overview**

The Linux operating system provides mechanisms that enable you to set the maximum limit for the amount of resources that a process can consume. Here are some of the resource types:

- open file descriptors
- stack size
- processes available to a user ID
Each resource type with limits is stored in the appropriate file on each machine in your deployment.

Here is the format of the /etc/security/limits.conf file for setting the maximum number of open file descriptors:

```
*     -     nofile     value
```

The asterisk (*) indicates all user accounts.

For a single user account, * can be replaced with the user ID for that account. Here is an example:

```
account-name     -     nofile     value
```

This line is duplicated in the file for each user ID.

For a group, * can be replaced with the at symbol (@) followed by the group name. Here is an example:

```
@group-name     -     nofile     value
```

Set the Maximum Number of Open File Descriptors and Stack Size

For each machine in your deployment:

1. Open the /etc/security/limits.conf file.

2. Set the limit for open file descriptors as follows:
   - If PostgreSQL will be deployed on the machine, set the limit (using the nofile item) to 150000 for the sas user.
     ```
     sas     -     nofile     150000
     ```
   - If you are deploying SAS Visual Investigator or SAS Intelligence and Investigation Management and the machine is running Elasticsearch, set the limit to at least 65536 for the sas user.
     ```
     sas     -     nofile     65536
     ```
   - For all other machines in the deployment, set the limit for the sas account, the cas account, and any other account that will be used to run a CAS session, including the root user, to at least 48000.
     ```
     *     -     nofile     48000
     ```

   **Note:** If you are performing a single-machine deployment, use the highest limit (described in step 2) for all users.

   ```
   *     -     nofile     150000
   ```

3. For machines on which PostgreSQL will be deployed, set the limit for the stack size (using the stack item) to 10240 for the sas user.

   ```
   sas     -     stack     10240
   ```

4. For machines that will not have PostgreSQL deployed on them, do not set a limit for the stack size.

5. Save and close the /etc/security/limits.conf file.

Set the Maximum Number of Processes Available

For each machine in your deployment:

1. Open the appropriate file. For Red Hat Enterprise Linux 6.7 or an equivalent distribution, open /etc/security/limits.d/90-nproc.conf. For Red Hat Enterprise Linux 7.1 and greater or an equivalent distribution, open /etc/security/limits.d/20-nproc.conf. For SUSE Linux, open /etc/security/limits.conf.

2. Set the limit for the number of processes as follows:
If PostgreSQL will be deployed on the machine, set the limit (using the nproc item) to 100000 for the sas user.

```
sas     -     nproc     100000
```

For all other machines in the deployment, set the sas account, the cas account, and any other account that will be used to run a CAS session to at least 65536.

```
*     -     nproc     65536
```

**Note:** If you are performing a single-machine deployment, use the highest limit (described in step 2) for all users.

```
*     -     nproc     100000
```

3 Save and close the `*-nproc.conf` file.

### Set the Semaphore Values

For each machine where PostgreSQL will be deployed:

1 Open the `/etc/sysctl.conf` file.

2 Add the following lines or modify existing values as follows:

```
kernel.sem=512 32000 256 1024
net.core.somaxconn=2048
```

3 Save and close the `/etc/sysctl.conf` file.

4 Refresh the revised settings from the `/etc/sysctl.conf` file:

```
sudo sysctl -p
```

### Tune Memory Capacity

**Note:** If you are not deploying SAS Visual Investigator or SAS Intelligence and Investigation Management, skip this section.

On each machine that will be hosting Elasticsearch:

1 Open the `/etc/sysctl.conf` file.

2 Add the following lines to increase the maximum virtual memory:

```
vm.max_map_count=262144
vm.overcommit_memory=0
```

3 Save and close the `/etc/sysctl.conf` file.

4 Refresh the revised settings from the `/etc/sysctl.conf` file.

```
sudo sysctl -p
```

### Change the Default Time-outs

**Note:** The information in this section applies only to systems running Red Hat Enterprise Linux 7.1 and later or equivalent distributions, including SUSE 12.1 and later. If you are using an earlier Linux distribution, skip this section.

To change the default time-out values:
1. Open the `/etc/systemd/system.conf` file.

2. Find the two variables that control timeouts: `DefaultTimeoutStartSec` and `DefaultTimeoutStopSec`.

3. If the lines that contain these variables are not already uncommented, uncomment each line by removing the number sign (`#`).

4. Assign both the `DefaultTimeoutStartSec` and `DefaultTimeoutStopSec` variables a value of `1800s`.

   ```
   DefaultTimeoutStartSec=1800s
   DefaultTimeoutStopSec=1800s
   ```

5. Save and close the `/etc/systemd/system.conf` file.

(SUSE Linux Only) Change the Maximum Number of Operating System Tasks

If you are deploying on SUSE Linux, run the following commands to change the maximum number of operating system (OS) tasks that each user can run concurrently.

**Note:** Run these commands as a root or sudoer user.

```
sudo sed -i 's#.*UserTasks.*#UserTasksMax=50000#g' /etc/systemd/logind.conf
sudo systemctl restart systemd-logind
```

These commands allow the user to run 50000 tasks concurrently.

Confirm the Identities of the Hosts

Each machine in the deployment must have a fully qualified domain name (FQDN). To ensure that each machine in the deployment has the host name that you expect, run the `hostname`, `hostname -f`, and the `hostname -s` commands on each machine. If any of the machines are not named as you expect or do not have an FQDN, correct the issue and run the commands again to confirm the correction.

**Note:** For more information about the `hostname` command and its options, see the Linux man pages.

(Optional) Enable Key-Based SSH Authentication

**Note:** Key-based SSH authentication is optional, but it is recommended.

In order to run Ansible tasks on multiple hosts without being prompted for a password, you can create an SSH key pair and distribute the public key to the machines where SAS software will be installed. Performing this task provides a secure authentication mechanism for SSH logins and avoids the need for SSH password options when performing Ansible tasks.

Here is an example of setting up an SSH key pair. However, there are many methods for creating and propagating SSH keys.

**Note:** These steps assume that the `PasswordAuthentication` keyword has been enabled in the SSH daemon configuration file. It is also assumed that the user has a password that can be used for `ssh-copy-id` authentication.
1. Create an SSH key pair without a passphrase. The following example specifies the RSA key type. However, you can specify any key type that is supported by your SSH installation. Refer to the ssh-keygen man page for more information.

   ```bash
   ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
   ```

2. Copy the public key to each target host. Here is an example:

   ```bash
   ssh-copy-id target0.example.com
   ssh-copy-id target1.example.com
   ```

   If the machine where Ansible is installed is also a target host for installing SAS software, run `ssh-copy-id` against the Ansible host.

3. Verify that you can authenticate to all target hosts without being prompted for a password.

   If you enable SELinux by setting it to enforcing after the deployment has completed, you must restore the context of any SSH keys that were generated while SELinux was running in permissive mode. Here is an example of the command:

   ```bash
   restorecon -R -v /path-to-keys/.ssh
   ```

   For `path-to-keys`, substitute the location of your SSH key files.

---

**Install Ansible**

Ansible is third-party software that provides automation and flexibility for deploying software to multiple machines. You must install a supported version of Ansible.

**Note:** By default, Ansible uses five forks to deploy software. If you are deploying SAS Viya software on more than five machines, see “Deployment of More Than Five Machines Takes More Time” on page 291 for information about changing the number of forks that Ansible uses.

**Standard Ansible Installation**

The Ansible installation process is documented at [http://docs.ansible.com/ansible/latest/intro_installation.html](http://docs.ansible.com/ansible/latest/intro_installation.html).

You should always follow the Ansible documentation and choose the installation method that works best for your IT environment.

Not all versions of Ansible that are available for installation are supported by SAS Viya. For a list of supported Ansible versions, see: [https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-operating-systems.html#ansible](https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-operating-systems.html#ansible). On that same page, SAS provides a list of supported versions of Python. Python support is determined by the release of Ansible that you install.

**Streamlined Ansible Installation for Red Hat Enterprise Linux and Equivalent Distributions**

**Note:** Even though you are advised to follow the instructions in the Ansible documentation, streamlined installation instructions are provided here as a convenience. Before performing these instructions, ensure that they are appropriate for your site and that they comply with the IT policies in your organization.

These steps assume that you have sudo access to the machine where you are installing Ansible.

1. Run the following commands to attach the EPEL repository to your server. You can copy and paste this entire block of text for convenience.

   ```bash
   ## find out which release {6 or 7}
   if   grep -q -i "release 6" /etc/redhat-release ; then
   ```
majversion=6
elif grep -q -i "release 7" /etc/redhat-release ; then
    majversion=7
else
    echo "Apparently, running neither release 6.x nor 7.x"
fi
## Attach EPEL
# Display the available repositories
sudo yum repolist

2 To Install Python PIP and related packages:
    sudo yum install -y python python-setuptools python-devel openssl-devel
    sudo yum install -y python-pip gcc wget automake libffi-devel python-six

3 Because EPEL will no longer be required, you can remove it with the following command:
    sudo yum remove -y epel-release

4 Upgrade PIP and setuptools using one of the following methods, based on the version of Python you are running.

Specific versions of Python modules are required. Here are some examples:
    sudo pip install --upgrade pip==9.0.3
    sudo pip install pycparser==2.14
    sudo pip install idna==2.7

For Python 2.7 (and later within 2.7.x):
    sudo pip install --upgrade pip setuptools

5 To install a specific version of Ansible through PIP:
    sudo pip install ansible==2.7.12

Streamlined Ansible Installation for SUSE Linux

Note: Even though you are advised to follow the instructions in the Ansible documentation, streamlined installation instructions are provided here as a convenience. Before performing these instructions, ensure that they are appropriate for your site and that they comply with the IT policies in your organization.

These steps assume that you have sudo access to the machine where you are installing Ansible.

1 To install Python's setup tools:
    sudo zypper install python-setuptools

2 To Install Python PIP:
    sudo easy_install pip

3 To install a specific version of Ansible through PIP:
    sudo pip install ansible==2.7.12

Test Your Ansible Installation

1 To test the Ansible version:
    ansible --version

Here is an example of successful output:
To perform a basic ping test:

```bash
ansible localhost -m ping
```

Here is an example of successful output:

```
[WARNING]: Host file not found: /etc/ansible/hosts
[WARNING]: provided hosts list is empty, only localhost is available
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```
Installation

Edit the Inventory File
- Overview ................................................. 74
- Single Machine Deployment ......................... 74
- Multiple Machine Deployment ....................... 75

Modify the vars.yml File .................................. 81
- Set the Deployment Label ........................... 81
- Set the Pre-deployment Validation Parameters .... 81
- Specify Security Settings ............................ 83
- Specify the Path to Certificates .................... 83
- Change the Repository Warehouse ................. 84
- Define Multiple Invocations ....................... 84
- (Optional) Specify JRE ............................... 84
- Set Up Passwordless SSH for CAS ................. 85
- Define the CAS User Group ......................... 87
- Set Up the CAS Admin User ....................... 87
- Add Data Source Information ...................... 87
- Set the Java Location ............................... 100
- Set Up the CAS Cache Directory .................. 100
- Set the CAS Data Directory ....................... 101
- Set Up HDFS and Co-location ..................... 102
- Change the Load Balancer Settings ............... 103
- (Optional) Set the CAS Network Port .......... 104
- Declare User Name Case Insensitivity for Back-End Authentication 105
- Change the CAS Host and Port ................. 106
- Change the SAS/CONNECT Ports ................. 107
- Change SAS Studio Configuration Properties .... 107
- Change the Object Spawner Port ............ 109
- Create the autoexec_deployment.sas File .... 109
- Create the sasenv_deployment File ............. 110
- Create the sasv9_deployment.cfg File .......... 111
- (Optional) Disable SELinux Verification .... 112

Modify the sitedefault.yml File ..................... 112
- Properties for Backup and Restore .............. 112

SAS Viya and Multi-tenancy .......................... 112
- Overview of Multi-tenancy ....................... 112
- Considerations for Multi-tenancy and LDAP ... 113
- Enable Multi-tenancy ............................. 113

Deploy the Software .................................. 115
- Assessment Test .................................. 115
- Deployment Command ............................. 116
- Options ......................................... 116
Edit the Inventory File

Note: This chapter describes the initial deployment of your SAS Viya software only. For information about modifying an existing deployment with updated software, adding new software to an existing deployment or upgrading to a new version of SAS Viya, see “Managing Your Software” on page 179.

Overview

Ansible uses an inventory file to specify the machines to be included in a deployment and the software to be installed on them. For SAS Viya deployments, `sas_viya_playbook/inventory.ini` is used as the inventory file. If you used the recommended location for uncompressing your playbook, the file is located at `/sas/install/sas_viya_playbook/inventory.ini`.

However, if you do not want to manually complete the default inventory.ini file, you can copy an existing template from the `sas_viya_playbook/samples` subdirectory instead. This directory contains templates for different types of deployments, including a single-machine deployment, which is described later in this chapter. Copy the template that you want to use, rename it inventory.ini, and place it in the `sas_viya_playbook` directory. It replaces the existing inventory.ini file.

Each inventory file consists of two parts:

- **deployment target definition**
  A specification of each machine on which SAS Viya software will be deployed.

- **host group assignment list**
  A mapping of the installable groups of software and the machines on which they will be deployed. SAS Viya software is deployed as host groups, which are identified by square brackets ([ ]) in the inventory file. Each host group is preceded by comments that describe the purpose of the software in the host group. The comments also contain any requirements, constraints, and recommendations relating to host groups.

  Specify the machines on which a host group will be deployed by listing them under the host group name. A machine can have more than one host group deployed on it.

  Here is an example of a host group assignment list:

  ```
  # The CommandLine host group contains command line interfaces for remote interaction with services.
  [CommandLine]
  deployTarget
  deployTarget2
  ```

  More details about the deployment target definition and the host group assignment list are included in the following sections.

Note: Inventory files are generated for a specific software order. Do not copy files from one playbook and attempt to use them with another playbook.
Single Machine Deployment

This section is applicable only if you are performing a single-machine deployment. If you are performing a multi-machine deployment, skip this section and go to “Specify the Machines in the Deployment” on page 75.

1 From the `sas_viya_playbook` directory, copy the `inventory_local.ini` file from its location and paste the copy in the top level of the `sas_viya_playbook` directory. This command also changes the name of the file to `inventory.ini`.

   ```
   cp samples/inventory_local.ini inventory.ini
   ```

   **Note:** Using an inventory file in any location other than the root directory can seriously affect the deployment of your software. If you do not want to copy a sample file into the root directory, ensure that the inventory file that you do use is in the root directory.

2 The first line of the `inventory.ini` file is a deployment target definition that identifies the machine on which the SAS Viya software is being deployed. If you are using Ansible locally (on the same machine where you are deploying SAS Viya software), you should not revise the deployment target definition.

   If you are using Ansible remotely, you should modify the deployment target definition to replace `ansible_connection=` with `ansible_host=` and include the location of the machine where SAS Viya is being deployed. Here is an example:

   ```
   deployTarget ansible_host=host1.example.com
   ```

3 If the deployment target has more than one network adapter, add a parameter that specifies which one should be used for Consul. Without the parameter, a deployment target that has multiple private IP addresses will fail. Here are examples that use the parameter:

   For a local machine:
   ```
   deployTarget ansible_connection=local consul_bind_adapter=eth0
   ```

   For a remote machine:
   ```
   deployTarget ansible_host=host1.example.com consul_bind_adapter=eth0
   ```

4 Save and close the `inventory.ini` file.

Multiple Machine Deployment

Specify the Machines in the Deployment

The first section in the `inventory.ini` file identifies a deployment target for each target machine. It also specifies the connection information that is needed by Ansible to connect to each machine. The following format is used to specify the deployment target reference. It is located at the beginning of the `inventory.ini` file.

   ```
   deployTarget ansible_host=<machine address> ansible_user=<userid> ansible_ssh_private_key_file=<keyfile>
   ```

The following table describes the components of the deployment target reference:
Table 4.1  Descriptions of Components of the Deployment Target Reference

<table>
<thead>
<tr>
<th>Component of the Deployment Target Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployTarget</td>
<td>specifies the alias that is used by Ansible to refer to the physical machine definition. The default alias is <code>deployTarget</code>. In a multi-machine deployment, you specify multiple deployment targets. In this case, choose a different alias name for each deployment target. Choose a meaningful alias such as <code>ansible-controller</code>.</td>
</tr>
<tr>
<td>ansible_host</td>
<td>specifies any resolvable address for the target host, such as the IP address or fully qualified domain name.</td>
</tr>
<tr>
<td>ansible_user</td>
<td>specifies the user ID that is used by Ansible to connect to each of the remote machines and to run the deployment. Must have root or sudo permissions.</td>
</tr>
<tr>
<td>ansible_ssh_private_key_file</td>
<td>specifies the private key file that corresponds to the public key that was previously installed on each of the remote machines. This file typically resides in your <code>~/.ssh</code> directory.</td>
</tr>
</tbody>
</table>

Note: Do not use the same machine for more than one alias. See the example below where each machine has a different alias.

The following example specifies the deployment target to be used when SAS Viya software will be deployed on the machine that is running Ansible:

```bash
deployTarget ansible_connection=local
```

The following example lists the deployment targets for a multi-machine deployment:

```bash
sas-service ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-programming ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
cas-controller-1 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
cas-backup-1 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
cas-worker-1 ansible_host=host5.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
cas-worker-2 ansible_host=host6.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
cas-worker-3 ansible_host=host7.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
vi-main ansible_host=host8.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-elastic-master-1 ansible_host=host9.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-elastic-master-2 ansible_host=host10.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-elastic-client-1 ansible_host=host11.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-elastic-client-2 ansible_host=host12.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-elastic-data-1 ansible_host=host13.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
sas-elastic-data-2 ansible_host=host14.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
```

If any of the deployment targets has more than one network adapter, add a parameter that specifies which one should be used for Consul. Without the parameter, a deployment target that has multiple private IP addresses will fail. Here is an example that uses the parameter:

```bash
sas-service ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa
```
To specify a machine address that is used by other machines in the deployment, add the `internal_deployment_ipv4_override` parameter to the deployment target reference. Using such a parameter ensures that intra-deployment connections to that machine route through the preferred address. Here is an example:

```
sas-service ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa internal_deployment_ipv4_override=149.173.160.3
```

To specify the preferred machine address when the machine refers to itself, add the `self_deployment_ipv4_override` parameter to the deployment target reference. This parameter is useful for services that perform binds. Here is an example:

```
sas-programming ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ssh/id_rsa self_deployment_ipv4_override=127.0.0.1
```

### Assign the Target Machines to Host Groups

The second section in the inventory file is used to assign deployment targets to each host group. Under each group, assign machines to the group by using the appropriate alias.

Do not add white space in order to indent machine name entries.

Here is a typical assignment that uses the machines from the preceding example.

**Note:** The inventory file contains comments that precede each host group and that describe its function to help in assigning machines. Those comments have been removed from this example to improve readability.

```
[AdminServices]
sas-service

[AdvancedAnalytics]
sas-service

[CASServices]
sas-service

[CommandLine]
sas-service
sas-programming
cas-controller-1
cas-backup-1
cas-worker-1
cas-worker-2
cas-worker-3
vi-main
sas-elastic-master-1
sas-elastic-master-2
sas-elastic-client-1
sas-elastic-client-2
sas-elastic-data-1
sas-elastic-data-2

[ComputeServer]
sas-programming

[ComputeServices]
sas-service
```
[CoreServices]
sas-service

[DataMining]
sas-service

[DataServices]
sas-service

[HomeServices]
sas-service

[ModelServices]
sas-service

[Operations]
sas-programming

[ReportServices]
sas-service

[ReportViewerServices]
sas-service

[ScoringServices]
sas-service

[ThemeServices]
sas-service

[configuratn]
sas-service

[consul]
sas-service

[elasticsearch]
sas-elastic-master-1 ElasticSearch_IsMaster=true ElasticSearch_IsData=false ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=1000
sas-elastic-master-2 ElasticSearch_IsMaster=true ElasticSearch_IsData=false ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=1000
sas-elastic-client-1 ElasticSearch_IsMaster=false ElasticSearch_IsData=false ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=1000
sas-elastic-client-2 ElasticSearch_IsMaster=false ElasticSearch_IsData=false ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=1000
sas-elastic-data-1 ElasticSearch_IsMaster=false ElasticSearch_IsData=true ElasticSearch_HeapSize=16g ElasticSearch_QueueSize=1000
sas-elastic-data-2 ElasticSearch_IsMaster=false ElasticSearch_IsData=true ElasticSearch_HeapSize=16g ElasticSearch_QueueSize=1000

[httpproxy]
sas-service

[pgpoolc]
sas-service
Edit the Inventory File

Edit the Inventory File

[programming]
sas-programming

[rabbitmq]
sas-service

[sas_casserver_primary]
cas-controller-1

[sas_casserver_secondary]
cas-backup-1

[sas_casserver_worker]
cas-worker-1
cas-worker-2
cas-worker-3

[sasdatasvrc]
sas-service

[viprCommon]
vi-main

[viprEntity]
vi-main

[viprSand]
vi-main

[viprVi]
vi-main

[viprVsd]
vi-main

[sas_all:children]
AdminServices
AdvancedAnalytics
CASServices
CommandLine
ComputeServer
ComputeServices
CoreServices
DataMining
DataServices
HomeServices
ModelServices
Operations
ReportServices
ReportViewerServices
ScoringServices
ThemeServices
configuratn
consul
elasticsearch
httpproxy
Consider the following issues when editing the inventory file:

- SAS recommends that you do not remove any host groups from the list or any entries from the [sas_all:children] list unless you are an experienced Ansible user. A host group can have no entries under it, but the host group should not be removed, even if it is empty. Removing a host group that contains targeted machines from the [sas_all:children] list can result in critical tasks not being executed on those targeted machines.

- For information about adding CAS servers or adding worker and secondary controller nodes to an existing CAS server, see SAS Cloud Analytic Services: How To (Scripts) in SAS Viya Administration: SAS Cloud Analytic Services.

- If you are using HDFS co-located with CAS, then [sas_casserver_primary] and [sas_casserver_worker] should be assigned to machines in the Hadoop cluster.

- If you purchased the optional SAS Event Stream Processing for CAS component, it is automatically installed on all machines where CAS components are installed. However, SAS Event Stream Processing Studio, Streamviewer, and SAS Event Stream Manager must be placed in the corresponding host groups.

- Users deploying SAS Visual Investigator or SAS Intelligence and Investigation Management should note that the entry for [elasticsearch] contains information not included with the other host groups. For more information, see the comment preceding [elasticsearch] in the inventory file.

Note: If you are upgrading SAS Visual Investigator or SAS Intelligence and Investigation Management, the format of the list in the [elasticsearch] group has changed for SAS Visual Investigator 10.4 and later and SAS Intelligence and Investigation Management 1.2 and later. Here are the changes:

- A machine that is only a master node changes from
  
  ```
  machine-name ElasticSearch_HostType=master ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
  ```

  to
  
  ```
  machine-name ElasticSearch_IsMaster=true ElasticSearch_IsData=false ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
  ```

- A machine that is only a client node changes from
  
  ```
  machine-name ElasticSearch_HostType=client ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
  ```

  to
  
  ```
  machine-name ElasticSearch_IsMaster=false ElasticSearch_IsData=false ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
  ```

- A machine that is only a data node changes from
  
  ```
  machine-name ElasticSearch_HostType=data ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
  ```

  to
  
  ```
  machine-name ElasticSearch_HostType=data ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
  ```
A master node can also be a data node by changing the value of `ElasticSearch_IsData` from `false` to `true`.

The minimum recommended value for the `ElasticSearch_QueueSize` has changed from `10000` to `1000`.

If you are deploying SAS Visual Investigator or SAS Intelligence and Investigation Management, SAS recommends that you do not put Elasticsearch, CAS, and PostgreSQL on the same machine.

If the machines that you specify for `[pgpoolc]` or `[sasdatasvrc]` do not have an alias of `deployTarget` in the deployment target reference, you must open the `sas_viya_playbook/vars.yml` file and replace the instance of `deployTarget` under `INVOCATION VARIABLES` with the alias that you used in the deployment target reference:

```yaml
# Multiple invocation definitions
INVOCATION_VARIABLES:
  deployTarget:
```

After you have completed your edits, save and close the `inventory.ini` file.

Note: By default, your deployment includes a single-machine, single-node instance of HA PostgreSQL, which is used as the SAS Infrastructure Data Server. If your software order contains SAS Assortment Planning, SAS Demand Planning, SAS Financial Planning, or SAS Markdown Optimization, your deployment includes a single-machine, single-node instance of CPS PostgreSQL. To deploy HA PostgreSQL or CPS PostgreSQL with multiple nodes, see “Creating High Availability PostgreSQL Clusters” on page 227.

---

**Modify the vars.yml File**

As its name suggests, the `vars.yml` file contains deployment variables that enable you to customize your deployment to meet your needs. If you used the recommended location for uncompressing your playbook, the file is located at `/sas/install/sas_viya_playbook/vars.yml`.

Note: All entries in the `vars.yml` file are case-sensitive.

**Set the Deployment Label**

The `DEPLOYMENT_LABEL` variable is a unique name used to identify the deployment across multiple machines. A default value for `DEPLOYMENT_LABEL` is set by the playbook.

If you want to use a customized `DEPLOYMENT_LABEL`, replace the default entry with another name, within double quotation marks, that is appropriate for your deployment. The name can contain only lowercase alphabetic characters, numbers, and hyphens. Nonalphanumeric characters, including a space, are not allowed. Here is an example of a valid name:

```
DEPLOYMENT_LABEL: "va-04april2017"
```

Note: Do not change the value of `DEPLOYMENT_LABEL` after the software has been deployed, including when performing an add-on or an upgrade.

**Set the Pre-deployment Validation Parameters**

The setting of the `VERIFY_DEPLOYMENT` variable determines the extent of the pre-deployment validation that the playbook performs. If the variable is set to true (the default), all of the following actions take place. If the variable is set to false, only the Ansible version check is performed.
Check the Ansible Version

The playbook checks the installed Ansible version to determine whether it is at least the minimum supported version. If not, the playbook stops execution with a message.

Note: For information about supported Ansible versions, see “Ansible Controller Requirements” on page 53.

Verify Machine Properties

The playbook checks each machine in the deployment to ensure that the necessary conditions for deployment are met. If any of the following conditions is not met, a warning is given and the playbook stops execution.

1 Verify that the DEPLOYMENT_LABEL variable has content and contains only lowercase alphabetic characters, numbers, and hyphens.
   Note: For more information about the DEPLOYMENT_LABEL variable, see “Set the Deployment Label” on page 81.

2 Verify that a CAS primary controller host is defined.
   Note: For information about assigning software to machines, see “Specify the Machines in the Deployment” on page 75.

3 Verify that each machine's fully qualified domain name contains less than or equal to 64 characters.

4 Verify that each machine in the inventory file can successfully connect to every other machine in the inventory file.
   Note: For more information about modifying the inventory file, see “Specify the Machines in the Deployment” on page 75.

5 Verify that each machine's fully qualified domain name resolves to the same address for every other machine.

6 If the playbook finds that the http_proxy and or https_proxy environment variables in the profile.d script have been set, verify that the no_proxy environment variable has been set as well. For more information about configuring a proxy server, see “Configure a Proxy Server” on page 64.

7 If the sas user already exists, verify that it is part of the sas user group.

Create and Verify sas User and sas Group

If the sas user and sas group do not already exist, the playbook creates the sas user and places it in the sas group. If this validation fails, a warning is provided and the playbook stops execution.

Verify System Requirements

The playbook ensures that some system requirements are met. If any of the following requirement checks fail, a warning is provided and the playbook stops execution.

1 Verify that SELinux mode for each machine is disabled. The value should be set to **permissive**, which disables SELinux.

   This check is performed by default. You can deploy SAS Viya in an environment where SELinux is enabled. However, some additional configuration is required. For more information about the requirements to run with SELinux enabled, see “Configure SELinux” on page 63.

   You can temporarily set SELinux to **permissive** until the deployment has completed. However, if you instead deploy with SELinux enabled (that is, set to **enforcing**), you must add a key-value pair that
disables playbook checks for SELinux. For more information, see "(Optional) Disable SELinux Verification" on page 112.

2 Verify that systemd is at version 219-30 or later.

3 Verify that each machine has enough free disk space to accommodate the packages that are installed on that machine. The amount of free space depends on the deployment layout.

   Note: For more information about assigning packages to machines, see "Specify the Machines in the Deployment" on page 75.

   This check uses system software for package management. The playbook configures SAS repositories and associated certificates as required by the system software.

4 For each machine, verify the nofile and nproc settings for the install user.

   Note: For more information about setting ulimits, see "Set the ulimit Values" on page 66.

Specify Security Settings

The SECURE_CONSUL and DISABLE_CONSUL_HTTP_PORT variables in vars.yml work together to determine the status of the HTTP and HTTPS ports. You can set both variables to true or false with the following results.

- If you set SECURE_CONSUL to false, only the HTTP port (8500) will be available after the software is deployed.

- If you set SECURE_CONSUL to true, the results depend on how DISABLE_CONSUL_HTTP_PORT is set:
  - If you set DISABLE_CONSUL_HTTP_PORT to true, only the HTTPS port (8501) is available.
  - If you set DISABLE_CONSUL_HTTP_PORT to false, both the HTTP port (8500) and the HTTPS port (8501) will be available.

By default, SECURE_CONSUL is set to true and DISABLE_CONSUL_HTTP_PORT is set to true. Only the HTTPS port will be available after the software is deployed.

Specify the Path to Certificates

Note: By default, when SAS Viya is deployed, it installs Apache httpd with a self-signed certificate for use across the deployment. If you want to accept the default, skip this section. If, however, you already have httpd set up and configured, you must provide a value for the HTTPD_CERT_PATH variable as described here.

The SSLCertificateChainFile is a variable set in the httpd security configuration file at /etc/httpd/conf.d/ssl.conf. It is a location on your system containing certificate information. SAS recommends that the file at the location that SSLCertificateChainFile represents contain the root certificate authority (CA) and all intermediate certificates in the chain.

To set HTTPD_CERT_PATH:

1 Open the vars.yml file.

2 Set the value of HTTPD_CERT_PATH based on the following conditions.

   - If your SSLCertificateChainFile contains the root certificate authority (CA) and all intermediate certificates, remove the existing value for HTTPD_CERT_PATH. Ensure that all browsers and clients have the root CA in their truststore.

     Here is an example of the modified variable:

     HTTPD_CERT_PATH:
If your SSLCertificateChainFile contains the intermediate links but not the root CA, HTTPD_CERT_PATH should be the path to the file on the machine in the [httpproxy] host group in the inventory file that contains the root CA.

If your SSLCertificateChainFile contains no certificates and no root CA, HTTPD_CERT_PATH should be the path to the file on the machine in the [httpproxy] host group in the inventory file that contains the intermediate certificates and the root CA. Ensure that all the intermediate certificates are in the truststore of all browsers and clients.

Here is an example of the HTTPD_CERT_PATH variable with a value:

```
HTTPD_CERT_PATH: "/etc/pki/tls/certs/my-ca-chain.crt"
```

Note: The default value for HTTPD_CERT_PATH in the vars.yml file is the most likely location for the necessary file for Red Hat Enterprise Linux and equivalent distributions. If that file is in the default location, you do not need to make any changes. The default location for SUSE Linux is `/etc/apache2/ssl.crt/localhost.crt`.

1. Save and close the vars.yml file.

### Change the Repository Warehouse

When you generate the playbook with the SAS Orchestration CLI, the REPOSITORY_WAREHOUSE variable in the vars.yml file is set to the default repository warehouse or to the repository warehouse that was specified in the command-line option. If you are using a mirror repository, the value for REPOSITORY_WAREHOUSE should be the location of that mirror. If the target deployment systems use a different address to the mirror repository or if the mirror repository is moved after the initial deployment, you should change the mirror location by revising the REPOSITORY_WAREHOUSE value.

Note: If you are using a Red Hat Satellite Server, use a value of `none` to prevent the deployment from adding more repositories to the server.

```
REPOSITORY_WAREHOUSE: "URL-to-mirror-repository-content"
```

### Define Multiple Invocations

By default, Ansible deploys High Availability (HA) PostgreSQL as a single node on a single machine. If your software order contains SAS Assortment Planning, SAS Demand Planning, SAS Financial Planning, or SAS Markdown Optimization, Ansible also deploys an instance of CPS PostgreSQL as a single node on a single machine. The INVOCATION_VARIABLES block is used to set the parameters of an HA PostgreSQL cluster or a CPS PostgreSQL cluster on more than one machine. For details, see “Creating High Availability PostgreSQL Clusters” on page 227.

### (Optional) Specify JRE

The Java Runtime Environment (JRE) must be installed on each target machine to enable SAS Viya. By default, the playbook attempts to install a recent version of OpenJDK and to set the path in a system configuration file. You can instead supply the path to an existing JRE before you run the playbook. To use a pre-installed version of the JRE:

1. With a text editor, open the vars.yml file.

2. Set the value of the sas_install_java variable to `false`. For example:

   ```yaml
   sas_install_java: false
   ```

3. Add the file path to the JRE as the value of the sasenv_java_home variable. Be sure to include `jre` in the file path. For example:

   ```yaml
   sasenv_java_home: /path/to/jre
   ```
Save and close the vars.yml file.

For the supported versions of Java, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-jre.html.

Set Up Passwordless SSH for CAS

Manage Passwordless SSH

If SAS Cloud Analytic Server (CAS) is deployed on multiple machines, each machine requires passwordless SSH in order to communicate with the others. Passwordless SSH is set up by the playbook by default.

You have three choices for managing passwordless SSH:

- Allow SAS to create a default passwordless SSH with a single user. See “Accept the Passwordless SSH Default” on page 85 for more information about the default process.
- Use your own passwordless SSH. See “Use Your Own Passwordless SSH” on page 85.
- Use the deployment process to create a customized passwordless SSH. Customization can include users other than the default. See “Create Customized Passwordless SSH” on page 86.

Accept the Passwordless SSH Default

SAS Viya requires that a user account for CAS must be created before you deploy your software. SAS recommends that the user ID for this account be cas. If you use a different user ID and still accept the default for passwordless SSH, you must ensure that the correct user ID is included in vars.yml. In the sas_users block, ensure that the first ID matches your CAS account ID:

```yaml
sas_users:
  cas:
    group: sas
    password: ''
    setup_home: false
    shell:
    home:
```

The casenv_user variable must also be set to the CAS account ID.

If you accept the default, the deployment process occurs as follows:

1. SSH keys are set up for the CAS user account.
2. A set of keys is created for any other user that is defined in the sas_users field.
3. The private and public keys are copied to each host that the playbook runs against.
4. The ssh-keyscan utility is run from each host to every other host in the CAS cluster.
5. The user’s public key is added to the ~/.ssh/authorized_keys file.

Use Your Own Passwordless SSH

If you choose to use your own passwordless SSH, you must set the cas user to be a user that you have already configured for passwordless SSH. For details, see “Set Up the CAS Admin User” on page 87.

To prevent the deployment process from setting up passwordless SSH:

Modify the vars.yml File
1 Open the vars.yml file.

2 Set the `setup_sas_users` variable to `false`. Here is an example:

   ```yaml
   setup_sas_users: false
   ```

3 Save and close the vars.yml file.

Create Customized Passwordless SSH

To use the playbook to set up passwordless SSH for new users:

Note: If you add an existing user to the vars.yml file, but with an attribute that is different from an attribute that was set elsewhere, the user attribute in the vars.yml file takes precedence.

1 Open the vars.yml file. Here is an example of the properties to be edited for each new user:

   ```yaml
   setup_sas_users: true
   sas_users:
     cas:
       group: sas
       password: ''
       setup_home: false
       shell:
       home:
       setup_sas_packages: false
       extra_packages:
         libselinux-python: support copying files
   ```

2 Edit the fields as follows:

   a Ensure that the `setup_sas_users` variable is set to `true`.

   b Create a list of user accounts and attributes under `sas_users`.

       Here are the attributes:

       - `group` – the group to which the user belongs. If the group does not exist, it is created when the playbook runs.

       - `password` – the encoded password for the user account. If you do not want to assign a password to the user account, use quotation marks (') that indicate that no password is assigned.

           Note: The comments in the vars.yml file explain how to create an encrypted password.

       - `setup_home` – uses the value of `true` or `false`. Determines whether the shell and home values should be used by the deployment. To accept the default, use a value of `false`.

       - `shell` – the location of the shell for the user account to use. It can be used only if `setup_home` is set to `true`.

       - `home` – the location of the user directory to be created. It can be used only if `setup_home` is set to `true`.

   c As an option, to install any packages to be defined under `extra_packages`, set `setup_packages` to `true`.

   d Under `extra_packages`, specify one or more names of any additional packages to install along with a comment that describes its purpose. The administrator typically uses this field to specify additional packages for the deployment (such as Firefox or Git) as a convenience. The field is ignored if `setup_packages` is set to `false`.

3 Save and close the vars.yml file.
After you edit the fields and run the playbook, the following actions occur:

- If the setup_sas_packages variable is set to `true`, any listed extra packages are installed.
- After CAS is installed, SSH is set up for any users that are specified in sas_users.
- CAS is configured for passwordless SSH. In addition, when the CAS controller is started, the workers also start.

**Define the CAS User Group**

Ensure that the user group for your CAS user account is correct.

1. Open the vars.yml file.
2. For the casenv_group variable, insert the user group name.
3. Save and close the vars.yml file.

**Set Up the CAS Admin User**

The CAS admin user is a member of the Superuser role in CAS. If you are performing a full deployment, you should designate an LDAP user to be the CAS admin user. For programming-only deployments, the CAS admin user can be a host account. You can also grant access to the CAS Superuser role by adding users to the SAS Administrators custom group after the deployment. If casenv_admin_user remains disabled in the vars.yml file during deployment, the only CAS Superuser is the casenv_user (default `cas`). For more information about CAS roles, see Identity Management: CAS Roles in SAS Viya Administration.

1. Open the vars.yml file.
2. Uncomment the line that contains the casenv_admin_user variable. To uncomment, remove the number sign (`#`).
3. In that same field, insert the name of a user that exists and that can log on.
   ```yaml
   casenv_admin_user: valid-user
   ```
4. Save and close the vars.yml file.

When the deployment is complete, you should use this user to log on to CAS Server Monitor in programming-only deployments.

**Add Data Source Information**

Note: If you are deploying only SAS Data Preparation in the cloud and it accesses only databases in a SAS Data Agent on-premises deployment, skip this section.

**Overview of the Data Sources**

If your software order includes one or more SAS/ACCESS products, you must edit the vars.yml file to include information that is needed to configure those products during deployment. Also, if you plan to use Hadoop Distributed File System (HDFS), you must also edit the vars.yml file.

SAS Viya uses the sasenv_deployment and cas_settings files to configure environment variables for the data sources. To create those files at deployment, add values to the FOUNDATION_CONFIGURATION and CAS_SETTINGS blocks of the vars.yml file before you run the playbook. The vars.yml file contains typical examples of these blocks, which are commented out with number signs (`#`). The following sections contain
examples of these blocks that are appropriate for the specific SAS/ACCESS products. To customize the file, either uncomment the lines and edit the existing blocks or create new blocks using the example’s format.

**Note:** If you start a new block, ensure that each line in the block begins with three spaces and a number. Each numbered line should reflect its numerical order within the block.

If you copy and paste from this guide, preserve indents as shown in each example. An indent is equivalent to three spaces.

After you save the file, the Ansible script is run to update the sasenv_deployment and cas.settings files.

**SAS/ACCESS Interface to Amazon Redshift**

For the following steps, depending on how you have configured the Amazon Redshift ODBC driver that is delivered with SAS/ACCESS Interface to Amazon Redshift, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   ```yaml
   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ``

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment, remove the number sign (#).

4. Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   ```yaml
   #CAS_SETTINGS:
   1: ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
   2: ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
   3: ODBCHOME=/opt/sas/viya/home/lib64/accessclients
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ``

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6. Save and close the vars.yml file.

**SAS/ACCESS Interface to DB2**

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   ```yaml
   #FOUNDATION_CONFIGURATION:
   1: CLASSPATH=$CLASSPATH:$DB2-related-classpath
   2: DB2INSTANCE=DB2-instance
   3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-DB2-install
   ``

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment, remove the number sign (#).

4. Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:
Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

Save and close the vars.yml file.

SAS/ACCESS Interface to Google BigQuery

No changes are required to the vars.yml file for deploying SAS/ACCESS to Google BigQuery.

SAS/ACCESS Interface to Greenplum

For the following steps, depending on how you have configured the Greenplum ODBC driver delivered with SAS/ACCESS Interface to Greenplum, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:
   
   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib

   For bulk loading, add the following lines.
   
   5: GPHOME_LOADERS=Greenplum-bulk-loader-install-location
   6: GPLOAD_HOME=Greenplum-install-location
   7: GPLOAD_PORT=Greenplum-bulk-load-port

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
4. Save and close the vars.yml file.

SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop

For the following steps, if you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the playbook to specify a different location for the installation of the JRE.

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #FOUNDATION_CONFIGURATION:
   1: JAVA_HOME=location-of-your-Java-8-JRE
   2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server

   If you are using MapR, add the following line.
3: MAPR_HOME=location-of-MapR-file

3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4 Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #CAS_SETTINGS:
   1: JAVA_HOME=location-of-your-Java-8-JRE
   2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server

   If you are using MapR, add the following line.

   3: MAPR_HOME=location-of-MapR-file

5 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6 Save and close the vars.yml file.

SAS/ACCESS Interface to HAWQ

For the following steps, depending on how you have configured the Greenplum ODBC driver that is delivered with SAS/ACCESS Interface to HAWQ, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1 Open the vars.yml file.

2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib

3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4 Save and close the vars.yml file.

SAS/ACCESS Interface to Impala

For the following steps, depending on how you have configured your Impala ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1 Open the vars.yml file.

2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   Note: Enter the variable for LD_LIBRARY_PATH on a single line. Multiple lines are used here to improve readability.

   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3. `CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file`
4. `EASYSOFT_UNICODE=YES`
5. `LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver: /opt/cloudera/impalaodbc/lib/64`

Note: The EASYSOFT_UNICODE variable should be added only to set the encoding for the SAS client to UTF-8.

3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4 Under CAS_SETTINGS, add the following lines as shown, including the indentations, spaces, and numerical prefixes:

Note: Enter the variable for LD_LIBRARY_PATH on a single line. Multiple lines are used here to improve readability.

```yaml
#CAS_SETTINGS:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCBINST=location-of-your-odbcinst.ini-file-including-file-name
3: CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
```

5 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6 Save and close the vars.yml file.

**SAS/ACCESS Interface to JDBC**

For the following steps, if you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is `/usr/lib/jvm/jre-1.8.0`. The default should be used unless you edit the playbook to specify a different location for the installation of the JRE.

1 Open the vars.yml file.

2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```yaml
#FOUNDATION_CONFIGURATION:
1: JAVA_HOME=location-of-your-Java-8-JRE
2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4 Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```yaml
#CAS_SETTINGS:
1: JAVA_HOME=location-of-your-Java-8-JRE
2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

5 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6 Save and close the vars.yml file.
SAS/ACCESS Interface to Microsoft SQL Server

For the following steps, depending on how you have configured the Microsoft SQL Server ODBC driver that is delivered with SAS/ACCESS Interface to Microsoft SQL Server, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #CAS_SETTINGS:
   1: ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
   2: ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
   3: ODBCHOME=/opt/sas/viya/home/lib64/accessclients
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6. Save and close the vars.yml file.

SAS/ACCESS Interface to MySQL

To edit the vars.yml file:

1. Open the vars.yml file.

2. Under FOUNDATION_CONFIGURATION, add the following line as shown, including the indentions, spaces, and numerical prefix:

   #FOUNDATION_CONFIGURATION:
   1: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-MySQL-client

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Under CAS_SETTINGS, add the following line, including the spaces and the numerical prefix:

   #CAS_SETTINGS:
   1: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-MySQL-client

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6. Save and close the vars.yml file.
SAS/ACCESS Interface to Netezza

For the following steps, depending on how you have configured your Netezza ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```yaml
#FOUNDATION_CONFIGURATION:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: NZ_ODBC_INI_PATH=path-to-the-Netezza-configuration-files
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Netezza-client
```

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
4. Save and close the vars.yml file.

SAS/ACCESS Interface to ODBC

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines (including the spaces and the numerical prefixes), depending on the version of ODBC that you are using.

For DataDirect:

```yaml
#FOUNDATION_CONFIGURATION:
1: ODBCHOME=ODBC-home-directory
2: ODBCINI=location-of-your-odbc.ini-file-including-file-name
3: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

For iODBC:

```yaml
#FOUNDATION_CONFIGURATION:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

For unixODBC:

```yaml
#FOUNDATION_CONFIGURATION:
1: ODBC_SYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
2: ODBCINI=name-of-your-odbc.ini-file
3: ODBCINSTINI=name-of-your-odbcinst.ini-file
4: LD_LIBRARY_PATH=location-of-ODBC-driver-manager-library:$LD_LIBRARY_PATH
```

Note: For unixODBC, if ODBC_SYSINI is not set in your environment, ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
4 Under **CAS_SETTINGS**, add the following lines (including the indentions, spaces, and numerical prefixes), depending on the version of ODBC that you are using:

For DataDirect:

```
#CAS_SETTINGS:
1: ODBCHOME=ODBC-home-directory
2: ODBCINST=location-of-your-odbc.ini-file-including-file-name
3: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

For iODBC:

```
#CAS_SETTINGS:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINSTINI=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

For unixODBC:

```
#CAS_SETTINGS:
1: ODBCSYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
2: ODBCINI=name-of-your-odbc.ini-file
3: ODBCINSTINI=name-of-your-odbcinst.ini-file
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

**Note:** For unixODBC, if ODBCSYSINI is not set in your environment, ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

5 Uncomment the **CAS_SETTINGS** line. To uncomment the line, remove the number sign (#).

6 Save and close the vars.yml file.

---

**SAS/ACCESS Interface to Oracle**

To edit the vars.yml file:

1 Open the vars.yml file.

2 Under **FOUNDATION_CONFIGURATION**, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
1: ORACLE_HOME=Oracle-home-directory
2: TWO_TASK=ORACLE_SID
3: ORAENV_ASK=NO
4: SASORA=V9
5: PATH=$PATH:$ORACLE_HOME/bin
6: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

3 Uncomment the **FOUNDATION_CONFIGURATION** line. To uncomment the line, remove the number sign (#).

4 Under **CAS_SETTINGS**, add the following lines, including the spaces and the numerical prefixes.

```
#CAS_SETTINGS:
1: ORACLE_HOME=Oracle-home-directory
2: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

5 Uncomment the **CAS_SETTINGS** line. To uncomment the line, remove the number sign (#).

6 Save and close the vars.yml file.
SAS/ACCESS Interface to PostgreSQL

For the following steps, depending on how you have configured your PostgreSQL ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:
   
   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   3: PGCLIENTENCODING=encoding-for-the-PostgreSQL-client-that-matches-the-SAS-client-encoding
   4: PATH=$PATH:/opt/sas/spre/home/lib64/psql
   5: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/sas/spre/home/lib64

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
4. Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.
   
   #CAS_SETTINGS:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   3: PGCLIENTENCODING=UTF-8
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/sas/viya/home/lib64

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).
6. Save and close the vars.yml file.

SAS/ACCESS Interface to SAP ASE

For users of SAP ASE Open Client 15, to correctly copy the SAP ASE libraries for use with SAS/ACCESS Interface to SAP ASE, you must have Read/Write authority for SAP-ASE-home-directory/lib and SAP-ASE-home-directory/devlib to run SAP-ASE-home-directory/scripts/lnsyblib. Instructions for copying the SAP libraries are in the header comments in the lnsyblib file.

In SAS Viya, you must install two SAP ASE-stored procedures on the target SAP server. Three files have been included in the !SASROOT/misc/dbi directory to assist in the installation:

- sas-spcp.txt is a text file containing instructions on how to perform the installation.
- sas-spdf_15.txt is the first of two stored procedure scripts.
- sassp2df_15.txt is the second of two stored procedure scripts.

The process uses two SAP ASE (Sybase) facilities, defncopy and isql.

To edit the vars.yml file:

1. Open the vars.yml file.
2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:
   
   #FOUNDATION_CONFIGURATION:
   1: SYBASE=path-to-the-SAP-ASE-Open-Client-Home
   2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SYBASE/lib
3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Save and close the vars.yml file.

**SAS/ACCESS Interface to SAP HANA**

For the following steps, depending on how you have configured your SAP HANA ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1. Open the vars.yml file.

2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   ```yaml
   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client
   ```

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.

   ```yaml
   #CAS_SETTINGS:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client
   ```

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6. Save and close the vars.yml file.

**SAS/ACCESS Interface to SAP R/3**

To edit the vars.yml file:

1. Open the vars.yml file.

2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   ```yaml
   #FOUNDATION_CONFIGURATION:
   1: RFC_INI=path-to-the-R/3-ini-file
   2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-R/3-client
   ```

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Save and close the vars.yml file.

**SAS/ACCESS Interface to Snowflake**

For the following steps, depending on how you have configured your Snowflake ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:
1. Open the vars.yml file.

2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-odbc-driver-manager:path-to-the-Snowflake-odbc-driver

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #CAS_SETTINGS:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-odbc-driver-manager:path-to-the-Snowflake-odbc-driver

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6. Save and close the vars.yml file.

---

**SAS/ACCESS Interface to Spark and SAS In-Database Technologies for Spark**

For the following steps, if you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is `/usr/lib/jvm/jre-1.8.0`. The default should be used unless you edit the playbook to specify a different location for the installation of the JRE.

To edit the vars.yml file:

1. Open the vars.yml file.

2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #FOUNDATION_CONFIGURATION:
   1: JAVA_HOME=location-of-your-Java-8-JRE
   2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server

3. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4. Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   #CAS_SETTINGS:
   1: JAVA_HOME=location-of-your-Java-8-JRE
   2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server

5. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6. Save and close the vars.yml file.
SAS/ACCESS Interface to Teradata and SAS In-Database Technologies for Teradata

1. Install all the packages from the Teradata Tools and Utilities (TTU) Base distribution on the SAS client node. In addition, you will need to install TTU on the following CAS nodes:
   - For serial data access only, install TTU on all CAS controller nodes.
   - For multi-node data access, install TTU on all CAS controller and CAS worker nodes.
   - For parallel data access, install TTU on all CAS controller nodes.
   
   Note: Parallel data access requires SAS Data Connect Accelerator for Teradata.

2. If you are setting the encoding on the SAS client node to UTF-8, modify the clispb.dat file for Teradata. This should be done on each CAS node where TTU was installed in Step 1. If you are not using encoding, you can skip this step.

   Locate the clispb.dat file in the directory where TTU was installed (for example, /opt/teradata/client/16.20/etc/clispb.dat). The path varies depending on the TTU version and where it is installed.

   Ensure that the following two lines are in the clispb.dat file.

   ```
   charset_type=N
   charset_id=UTF8
   ```

   SAS recommends keeping the clispb.dat file in the default location. Relocating the file can cause issues.

3. Open the vars.yml file.

4. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes.

   ```
   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   ```

5. Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

6. Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.

   ```
   #CAS_SETTINGS:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   ```

7. Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

8. Save and close the vars.yml file.

SAS/ACCESS Interface to Vertica

For the following steps, depending on how you have configured your VerticaODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

To edit the vars.yml file:

1. Open the vars.yml file.

2. Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

   ```
   #FOUNDATION_CONFIGURATION:
   1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
   ```
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Vertica-client
4: VERTICAINI=path-to-the-Vertica-configuration-files

3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4 Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.

#CAS_SETTINGS:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Vertica-client
4: VERTICAINI=path-to-the-Vertica-configuration-files

5 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

6 Save and close the vars.yml file.

Specify Multiple Data Connectors

Note: When adding multiple SAS/ACCESS products, make sure that the lines that you add are in the same block and are numbered consecutively from first to last. Even though the lines for the SAS/ACCESS products can be mixed in the block, ensure that the lines for each product remain in the order that was provided in the preceding sections.

Because the LD_LIBRARY_PATH variable is included for each SAS/ACCESS product, if you have more than one data connector, use as many lines as you have data connectors.

Here is an example of a block for both the DataDirect version of SAS/ACCESS Interface to ODBC and for SAS/ACCESS Interface to Oracle.

1 Open the vars.yml file.

2 Under FOUNDATION_CONFIGURATION, add the appropriate lines as shown, including the indentions, spaces, and numerical prefixes:

#FOUNDATION_CONFIGURATION:
1: ODBCHOME=ODBC-home-directory
2: ODBCINI=location-of-your-odbc.ini-file-including-file-name
3: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
4: ORACLE_HOME=Oracle-home-directory
5: TWO_TASK=ORACLE_SID
6: ORABNV_ASK=NO
7: SASORA=V9
8: PATH=$PATH:$ORACLE_HOME:bin
9: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
10: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH

3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

4 Under CAS_SETTINGS, add the appropriate lines as shown, including the indentions, spaces, and numerical prefixes:

#CAS_SETTINGS:
1: ODBCHOME=ODBC-home-directory
2: ODBCINI=location-of-your-odbc.ini-file-including-file-name
3: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
4: ORACLE_HOME=Oracle-home-directory
5: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).

Save and close the vars.yml file.

### Set the Java Location

If you have not already identified a `JAVA_HOME` while adding data source information (see "Add Data Source Information" on page 87), you should add the location of your Java to the vars.yml file.

1. Open the vars.yml file.
2. Under CAS_SETTINGS, add a line for `JAVA_HOME`, numbered in sequence with any variables already present:

   ```yaml
   #CAS_SETTINGS:
   ...next-number-in-sequence-of-variables: JAVA_HOME=location-of-your-Java-8-JRE
   ```
3. Under the `JAVA_HOME` line, add another line to include the `JAVA_HOME` in the `LD_LIBRARY_PATH` variable.

   ```yaml
   #CAS_SETTINGS:
   ...next-number-in-sequence-of-variables: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
   ```
4. If you have not already done so, uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).
5. Save and close the vars.yml file.

Here is an example:

```yaml
CAS_SETTINGS:
  1: ODBCHOME=ODBC-home-directory
  2: ODBCINI=location-of-your-odbc.ini-file-including-file-name
  3: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
  4: ORACLE_HOME=Oracle-home-directory
  5: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
  6: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
  7: JAVA_HOME=/usr/lib/jvm/jre-1.8.0
  8: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

### Set Up the CAS Cache Directory

**Change the CAS Cache Directory**

SAS Cloud Analytics Services (CAS) is the in-memory analytic server for SAS Viya. As a memory efficiency, CAS organizes in-memory data in blocks and memory maps the blocks. The blocks are stored as temporary files in directories on the host. The controller also uses the cache directory temporarily to store uploaded files.

By default, only the `/tmp` directory is used as the cache directory. This is sufficient for demonstration purposes, but not for production use of the server.

For a production-use server, set the cache to use a series of directories. The size required differs for each deployment, but can run from gigabytes to terabytes. When you specify a series of directories, each time the server needs to use disk, it uses the next path in the list. This strategy is used to distribute the load across disk volumes.
SAS supports only local file systems for the CAS cache such as EXT4 and XFS. It does not support network file systems such as GPFS.

To change the CAS cache:

1. Open the vars.yml file.

2. In the CAS CONFIGURATION section, uncomment the line that contains the CAS_DISK_CACHE variable. To uncomment, remove the number sign (#).

3. Remove the /tmp value from the variable and replace it with the directory that you want to use as the CAS cache. If you want to use more than one directory, list them all with colons separating the directories. For example:

   ```yaml
   CAS_CONFIGURATION:
     env:
       CAS_DISK_CACHE: /disk1:/disk2:/disk3
   ```

   **Note:** To avoid the potential for problems caused by the CAS disk cache that can fill up the root file system, do not specify any directory that is on the same partition or logical volume as the root file system.

4. Save and close the vars.yml file.

SAS recommends that you create directories dedicated to caching that are owned by the ID that executes the CAS server (cas by default). Each directory should be set up identically on each CAS node. All CAS processes must have Read, Write, and Execute permissions for these directories. Therefore, permissions must be granted to the server's ID and the ID of any CAS user that connects through programming interfaces like SAS and Python.

The directory structure must be identical on each controller and worker, but the controller host does not require the same volume of space as each worker. To conserve disk space on the controller, the directories that are specified in CAS_DISK_CACHE can occupy as little as one partition or one logical volume. Likewise, the directories can be specified as file system links to a single directory. In all cases, make sure that the directories do not use the same disk or the same volume as the root file system.

**Tune the CAS Cache Directory**

Here are some tuning tips for the CAS cache directory:

- Configure each disk device as a separate file system. For hosts with eight or more disk devices, dedicate one device for file system journals. When you create the file systems, specify the dedicated device as the external journal.

  **Note:** If you can predict that the total size of the tables in CAS_DISK_CACHE is less than the available RAM, you can set CAS_DISK_CACHE to /dev/shm rather than to a disk file system.

- The noatime and nodiratime mount options are applicable if no other data on the file system prevents the use of these mount options. If appropriate for your power supply, the nobarrier mount option might be applicable. Increasing the read-ahead value might improve performance. Refer to the Linux documentation for more information about these mount options.

- Reducing the aggressiveness to swap memory pages can improve performance:

  ```bash
  sudo sysctl -w vm.swappiness=1
  ```

**Set the CAS Data Directory**

**Note:** If you are performing an upgrade, skip this section.

By default, product caslibs are written to /opt/sas/viya/config/data/cas/default, which is often hosted on a single hard disk drive with limited storage. To ensure proper performance of your SAS solutions, SAS recommends that the CASDATADIR option be configured to point to a high-performance storage platform.
Examples of high-performance storage platforms include SAN, NVMe, and multiple drive disk arrays. You can also use NFS, but it must be mounted with the `NO_ROOT_SQUASH` option. Whichever storage platform you use must be owned by the cas user.

Changing the CAS data directory is especially useful for solutions that can be resource-intensive, such as SAS Visual Forecasting, SAS Visual Data Mining and Machine Learning, SAS Visual Text Analytics, and SAS Analytics for IoT.

Perform these steps to define a different CAS data directory:

1. Open the vars.yml file.
2. In the CAS_CONFIGURATION block, add the following line for CASDATADIR after the env line.
   ```yaml
   env:
     CASDATADIR: path-to-CAS-data-directory
   ``
   Note: The CAS data directory must be accessible from the primary and secondary CAS controllers.
3. Save and close the vars.yml file.

**Set Up HDFS and Co-location**

Default settings for the CAS_CONFIGURATION section of the vars.yml file appear as follows:

```yaml
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
  cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    #colocation: 'none'
```

Note: For descriptions of HDFS and co-location, see “CAS Server Co-located with Hadoop” on page 14.

If you include a machine in the host group [sas_casserver_worker] in the inventory file, the playbook assumes that you are performing a massively parallel processing (MPP) deployment. This means that your CAS deployment includes a controller and at least one worker. When the playbook runs, it removes the number sign (#) from the colocation variable and adds a mode variable that is set to `mpp`. You must continue to edit the CAS_CONFIGURATION section as follows:

1. Open the vars.yml file.
2. If you are deploying some or all of your CAS machines on the same machines where HDFS is running, revise the variables’ values as follows:

   ```yaml
   CAS_CONFIGURATION:
     env:
       #CAS_DISK_CACHE: /tmp
       HADOOP_NAMENODE: namenode-host-name
       HADOOP_HOME: location-of-your-Hadoop-home-directory-on-the-HDFS-server
     cfg:
       #gcport: 0
       #httpport: 8777
       #port: 5570
       colocation: 'hdfs'
       mode: 'mpp'
   ```
Note: HADOOP_NAMENODE can be up to two host names, the primary and standby namenodes, separated by a colon. For example:

HADOOP_NAMENODE=namenode1:namenode2

Note: If you intend to use remote HDFS, ensure that the path used for HADOOP_HOME includes /lib/hadoop. For example: /opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib/hadoop.

3 If you are deploying CAS on machines completely separate from the HDFS machines, revise the variables’ values as follows:

CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    HADOOP_NAMENODE: namenode-host-name
    HADOOP_HOME: location-of-your-Hadoop-home-directory-on-the-HDFS-server
    CAS_ENABLE_REMOTE_SAVE: 1
    CAS_REMOTE_HADOOP_PATH: 'SASHDAT-executables-directory-on-the-HDFS-server'
  cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    colocation: 'hdfs'
    mode: 'mpp'

By default, the deployment searches HADOOP_HOME and /opt/sas/HDATHome/bin for the HDAT plug-ins. You should supply a value for CAS_REMOTE_HADOOP_PATH only if you are using a location for the HDAT plug-ins other than HADOOP_HOME or /opt/sas/HDATHome/bin.

Note: HADOOP_NAMENODE can be up to two host names, the primary and standby namenodes, separated by a colon. For example:

HADOOP_NAMENODE=namenode1:namenode2

Note: If you intend to use remote HDFS, ensure that the path used for HADOOP_HOME includes /lib/hadoop. For example: /opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib/hadoop.

4 Save and close the vars.yml file.

Note: For more information about CAS environment variables, see CAS Environment Variables in SAS Viya Administration: SAS Cloud Analytic Services.

Change the Load Balancer Settings

If your deployment uses a load balancer to increase capacity and reliability, you should set its location in the vars.yml file.

1 Open the vars.yml file if it is not already open.

2 Locate the CAS_CONFIGURATION block of variables.

CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    #CAS_VIRTUAL_HOST: 'loadbalancer.company.com'
    #CAS_VIRTUAL_PROTO: 'https'
    #CAS_VIRTUAL_PORT: 443
  cfg:
    #gcport: 0
    #httpport: 8777
3 Remove the number sign (#) from the beginning of each of these lines: CAS_VIRTUAL_HOST, CAS_VIRTUAL_PROTO, CAS_VIRTUAL_PORT, and SERVICESBASEURL.

Note: You must remove the number sign from each of these lines regardless of whether the default values are correct for your deployment.

CAS_CONFIGURATION:
env:
    #CAS_DISK_CACHE: /tmp
    CAS_VIRTUAL_HOST: 'loadbalancer.company.com'
    CAS_VIRTUAL_PROTO: 'https'
    CAS_VIRTUAL_PORT: 443
cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    #colocation: 'none'
    SERVICESBASEURL: 'https://loadbalancer.company.com'

4 Remove the example values and replace them with appropriate values for the load balancer.

Note: The values for the CAS_VIRTUAL_HOST and SERVICESBASEURL variables must be changed since the default values for both are not true FQDNs. You are not required to change the values for the CAS_VIRTUAL_PROTO and CAS_VIRTUAL_PORT variables if the defaults are accurate for your deployment.

CAS_CONFIGURATION:
env:
    #CAS_DISK_CACHE: /tmp
    CAS_VIRTUAL_HOST: 'fully-qualified-domain-name-of-the-load-balancer'
    CAS_VIRTUAL_PROTO: 'protocol-portion-of-the-URL-for-the-reverse-proxy'
    CAS_VIRTUAL_PORT: port-where-the-load-balancer-listens-for-incoming-connections
cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    #colocation: 'none'
    SERVICESBASEURL: 'https://fully-qualified-domain-name-of-the-load-balancer'

5 Save and close the vars.yml file.

(Optional) Set the CAS Network Port

GCPORT is a named port that can be used by a distributed CAS server for communication between the controller and its worker nodes. You can specify almost any regular port number for this variable. If you do not specify a port number for GCPORT, an ephemeral port is used for each node of the CAS cluster.

To set GCPORT:

1 Open the vars.yml file if it is not already open.

2 Locate the CAS_CONFIGURATION block of variables.

CAS_CONFIGURATION:
env:
    #CAS_DISK_CACHE: /tmp
3 Remove the number sign (#) from the beginning of the gcport line.

```
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    #CAS_VIRTUAL_HOST: 'loadbalancer.company.com'
    #CAS_VIRTUAL_PROTO: 'https'
    #CAS_VIRTUAL_PORT: 443
  cfg:
    gcport: 0
    #httpport: 8777
    #port: 5570
    #colocation: 'none'
    SERVICESBASEURL: 'https://loadbalancer.company.com'
```

4 Remove the sample value for the gcport variable and replace it with a preferred value. Here is an example:

```
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    #CAS_VIRTUAL_HOST: 'loadbalancer.company.com'
    #CAS_VIRTUAL_PROTO: 'https'
    #CAS_VIRTUAL_PORT: 443
  cfg:
    gcport: 5580
    #httpport: 8777
    #port: 5570
    #colocation: 'none'
    SERVICESBASEURL: 'https://loadbalancer.company.com'
```

5 Save and close the vars.yml file.

### Declare User Name Case Insensitivity for Back-End Authentication

If you are using Microsoft Active Directory for back-end user authentication and your user names contain uppercase or mixed case letters, to ensure a successful authentication in a SAS Viya deployment, you must configure SAS Viya to ignore the casing of user names.

To configure your SAS Viya deployment to ignore the casing of user names:

1 Open the vars.yml file.

2 In the CAS_CONFIGURATION section, add two new variables to the end of the env block.

```
CAS_CONFIGURATION:
  env:
    ...
    CASUSERLOWERCASE: 'on'
```
3 Save and close the vars.yml file.

Change the CAS Host and Port

Changing the CAS host or port is especially useful if you connect to a CAS environment that is different from the one that is being deployed. To change either the CAS host or the port, or both, to be used by SAS Studio:

1 Open the vars.yml file if it is not already open.

2 Locate the sasenv_cas_host and sasenv_cas_port variables that immediately follow the Foundation Configuration header.

   # Optional: Will use the CAS controller host as defined in the inventory file. If one is not defined it will default to localhost.
   # If you know the host of the controller you want to connect to, provide that here
   #sasenv_cas_host:

   # Optional: If a value is not provided, the system will use the CAS port as defined for the CAS controller.
   # If you know the port of the grid you want to connect to, provide that here.
   #sasenv_cas_port:

3 Remove the number sign (#) from the beginning of the line that contains the sasenv_cas_host variable or the sasenv_cas_port variable. You can also remove # from both lines.

   # Optional: Will use the CAS controller host as defined in the inventory file. If one is not defined it will default to localhost.
   # If you know the host of the controller you want to connect to, provide that here
   sasenv_cas_host:

   # Optional: If a value is not provided, the system will use the CAS port as defined for the CAS controller.
   # If you know the port of the grid you want to connect to, provide that here.
   sasenv_cas_port:

4 Add the value for the CAS host or port, as appropriate.

   # Optional: Will use the CAS controller host as defined in the inventory file. If one is not defined it will default to localhost.
   # If you know the host of the controller you want to connect to, provide that here
   sasenv_cas_host: fully-qualified-domain-name-of-CAS-controller

   # Optional: If a value is not provided, the system will use the CAS port as defined for the CAS controller.
   # If you know the port of the grid you want to connect to, provide that here.
   sasenv_cas_port: port-for-SAS-Studio-to-connect-to-CAS-controller

5 Save and close the vars.yml file.
Change the SAS/CONNECT Ports

The SAS/CONNECT Spawner management port is used for monitoring the health of the SAS/CONNECT Spawner. To change the SAS/CONNECT Spawner management port or the SAS/CONNECT Spawner port:

1. Open the vars.yml file if it is not already open.
2. Locate the block of variables that begins with Set the ports that SAS/CONNECT will listen on.
   
   # Set the ports that SAS/CONNECT will listen on
   #sasenv_connect_port: 17551
   #sasenv_connect_mgmt_port: 17541

3. Remove the number sign (#) from the beginning of the line that contains the sasenv_connect_port variable or the sasenv_connect_mgmt_port variable. You can also remove # from both lines.
   
   # Set the ports that SAS/CONNECT will listen on
   sasenv_connect_port: 17551
   sasenv_connect_mgmt_port: 17541

4. Change the port number, as appropriate.
   
   # Set the ports that SAS/CONNECT will listen on
   sasenv_connect_port: 17552
   sasenv_connect_mgmt_port: 17542

5. Save and close the vars.yml file.

Change SAS Studio Configuration Properties

Note: If you are performing an upgrade, skip this section.

To change SAS Studio configuration properties:

1. Open the vars.yml file if it is not already open.
2. Locate the STUDIO.Configuration block of variables.
   
   # Updates the init_deployment.properties and appserver_deployment.sh
   STUDIO_CONFIGURATION:
   init:
   #sastudio.appserver.port_comment: '# Port that Studio is listening on'
   #sastudio.appserver.port: 7080
   #sastudio.appserver.https.port: 7443
   #webdms.workspaceServer.hostName: localhost
   #webdms.workspaceServer.port: 8591
   appserver:
   #1: '# Comment about KEY'
   #2: KEY="value with spaces"

3. To change any default properties for SAS Studio:
   
   a. In the init subsection, to indicate a variable to change, uncomment its line by removing the number sign (#) from the beginning of the line. Here are the definitions of the variables that can be changed:
Table 4.2 Descriptions of SAS Studio Variables in the vars.yml File

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sasstudio.appserver.port_comment</td>
<td>Adds a comment before the port entry in the init_deployment.properties that are created by the deployment process.</td>
</tr>
<tr>
<td>sasstudio.appserver.port</td>
<td>Specifies the port to use for HTTP. Should be modified only for programming-only deployments.</td>
</tr>
<tr>
<td>sasstudio.appserver.https.port</td>
<td>Specifies the port to use for HTTPS. Should be modified only for programming-only deployments.</td>
</tr>
<tr>
<td>webdms.workspaceServer.hostName</td>
<td>Specifies the host to use to connect to the workspace server.</td>
</tr>
<tr>
<td>webdms.workspaceServer.port</td>
<td>Specifies the port to use to connect to the workspace server.</td>
</tr>
</tbody>
</table>

Here is an example of the SAS Studio ports to be changed:

```yml
# Updates the init_deployment.properties and appserver_deployment.sh
STUDIO_CONFIGURATION:
  init:
    #sasstudio.appserver.port_comment: '# Port that Studio is listening on'
    sasstudio.appserver.port: 7080
    sasstudio.appserver.https.port: 7443
    #webdms.workspaceServer.hostname: localhost
    #webdms.workspaceServer.port: 8591
```

**b** Change the values, as appropriate. Here is an example:

```yml
# Updates the init_deployment.properties and appserver_deployment.sh
STUDIO_CONFIGURATION:
  init:
    #sasstudio.appserver.port_comment: '# Port that Studio is listening on'
    sasstudio.appserver.port: 7081
    sasstudio.appserver.https.port: 7444
    #webdms.workspaceServer.hostname: localhost
    #webdms.workspaceServer.port: 8591
```

4 To add environment variables for SAS Studio:

**a** Uncomment the lines in the appserver subsection by removing the number sign (#) from the beginning of the lines. Here is an example of a commented line:

```yml
appserver:
  1: '# Comment about KEY'
  2: KEY="value with spaces"
```

**b** Add the environment variables, as appropriate, and ensure that the line numbers are incremented by one for each line that you add. Use the indentation that is already in the vars.yml file. Ensure that the value in each line is enclosed in single quotation marks. Comments must include the comment character, #, within the quotation marks.

5 After you have made the appropriate changes, save and close the vars.yml file.
**Change the Object Spawner Port**

To change the Object Spawner default port:

**Note:** If you change the port value for the Object Spawner, you must also change the value of `webdms.workspaceServer.port` in the `STUDIO_CONFIGURATION` block to match the port number that you specify in the `SPAWNER_CONFIGURATION` block. See “Change SAS Studio Configuration Properties” on page 107 for more information about the `STUDIO_CONFIGURATION` block.

1. Open the vars.yml file if it is not already open.

2. Locate the `SPAWNER_CONFIGURATION` block of variables.

   ```yaml
   # Updates spawner.cfg
   SPAWNER_CONFIGURATION:
     #sasPort: 8591
   ``

3. Uncomment the third line by removing the number sign (#) from the beginning of the line.

   ```yaml
   # Updates spawner.cfg
   SPAWNER_CONFIGURATION:
     sasPort: 8591
   ``

4. Change the port number to the one that you want to use.

   ```yaml
   # Updates spawner.cfg
   SPAWNER_CONFIGURATION:
     sasPort: 8592
   ``

5. Save and close the vars.yml file.

**Create the autoexec_deployment.sas File**

**Overview**

SAS Viya uses the autoexec_deployment.sas file to set environment variables. To create that file at deployment, add values to the vars.yml file before running the playbook.

**autoexec_deployment.sas**

1. Open the vars.yml file if it is not already open.

2. Locate the `WORKSPACESERVER_CONFIGURATION` block of variables.

   ```yaml
   # Updates the workspaceserver autoexec_deployment.sas
   #WORKSPACESERVER_CONFIGURATION:
     #1: /* Comment about key */
     #2: key=value;
   ``

3. Uncomment the second, third, and fourth lines by removing the number sign (#) from the beginning of the line.

   ```yaml
   # Updates the workspaceserver autoexec_deployment.sas
   WORKSPACESERVER_CONFIGURATION:
     1: /* Comment about key */
     2: key=value;
   ``

4. Add the environment variables, as appropriate, and ensure that the line numbers are incremented by one for each line that you add to the `WORKSPACESERVER_CONFIGURATION` block. Use the indentation that is already
in the vars.yml file. Ensure that the value in each line is enclosed in single quotation marks. Comments must include the comments set of characters, /* and */, within the quotation marks.

5 After you have entered all the settings that you want to use, save and close the vars.yml file.

**Change the autoexec_deployment.sas File after Deployment**

After your software has been deployed, you might want to change some settings that were created in the previous section. You could edit the autoexec_deployment.sas file directly, but you would lose any customizations if the playbook were run again.

To prevent the loss of your customizations, SAS recommends that you create a separate file (referred to as a usermods file) to be consumed by subsequent deployments. For the changes to the autoexec_deployment.sas file, create a usermods file named sasv9_usermods.cfg by copying the sasenv_deployment file and renaming it. Make your changes to the sasv9_usermods.cfg file. It overrides the autoexec_deployment.sas file in subsequent deployments.

**Create the sasenv_deployment File**

**Overview**

SAS Viya uses the sasenv_deployment file to set environment variables. To create that file at deployment, add values to the vars.yml file before running the playbook.

**sasenv_deployment**

1 Open the vars.yml file if it is not already open.

2 Locate the FOUNDATION_CONFIGURATION block of variables.

   # Creates a workspaceserver sasenv_deployment file
   #FOUNDATION_CONFIGURATION:
   #1: '# Comment about KEY'
   #2: KEY=value

3 Uncomment the second, third, and fourth lines by removing the number sign (#) from the beginning of the line.

   # Creates a workspaceserver sasenv_deployment file
   FOUNDATION_CONFIGURATION:
   1: '# Comment about KEY'
   2: KEY=value

4 Add the environment variables, as appropriate, and ensure that the line numbers are incremented by one for each line that you add to the FOUNDATION_CONFIGURATION block. Use the indentation that is already in the vars.yml file. Ensure that the value in each line is enclosed in single quotation marks. Comments must include the comment character, #, within the quotation marks.

5 After you have entered all the settings that you want to use, save and close the vars.yml file.

**Change the sasenv_deployment File after Deployment**

After your software has been deployed, you might want to change some settings that were created in the previous section. You could edit the sasenv_deployment file directly, but you would lose any customizations if the playbook were run again.

To prevent the loss of your customizations, SAS recommends that you create a separate file (referred to as a usermods file) to be consumed by subsequent deployments. For the changes to the sasenv_deployment file,
create a usermods file named /opt/sas/viya/config/etc/server-type/default/sasenv_local by copying the sasenv_deployment file and renaming it. Make your changes to the sasenv_local file. It overrides the sasenv_deployment file in subsequent deployments.

Create the sasv9_deployment.cfg File

Overview

SAS Viya uses the sasv9_deployment.cfg file to set system options. To create that file at deployment, add values to the vars.yml file before running the playbook.

sasv9_deployment.cfg

1. Open the vars.yml file if it is not already open.

2. Locate the SASV9_CONFIGURATION block of variables.

   # Creates a workspaceserver sasv9_deployment.cfg file
   #SASV9_CONFIGURATION:
   #1: '/* Comment about OPTION */'
   #2: 'OPTION value'

3. Uncomment the second, third, and fourth lines by removing the number sign (#) from the beginning of the line.

   # Creates a workspaceserver sasv9_deployment.cfg file
   SASV9_CONFIGURATION:
   1: '/* Comment about OPTION */'
   2: 'OPTION value'

4. Add the system options and comments, as appropriate, and ensure that the line numbers are incremented by one for each line that you add to the SASV9_CONFIGURATION block. Here is an example:

   # Creates a workspaceserver sasv9_deployment.cfg file
   SASV9_CONFIGURATION:
   1: '/* aligning output with page boundary */'
   2: 'ALIGNSASIOFILES'
   3: 'BYLINE'
   4: '/* setting the paper size */'
   5: 'PAPERSIZE=LETTER'

   Note: For the list of available system options, see SAS System Options: Reference: System Options.

   Use the indentation that is already in the vars.yml file. Ensure that the value in each line is enclosed in single quotation marks. Comments must include the comments set of characters, /* and */, within the quotation marks.

5. After you have entered all the settings that you want to use, save and close the vars.yml file.

Modify the sasv9_deployment.cfg File after Deployment

After your software has been deployed, you might want to change the configuration settings that were created in the previous section. You could modify the sasv9_deployment.cfg file directly, but you would lose any customizations if the playbook were run again.

To prevent the loss of your customizations, SAS recommends that you create a separate file (referred to as a usermods file) to be consumed by subsequent deployments. For the changes to the sasv9_deployment.cfg file, create a usermods file named sasv9_usermods.cfg by copying the sasv9_deployment.cfg file and renaming it.
Make your changes to the sasv9_usermods.cfg file. It overrides the sasv9_deployment.cfg file in subsequent deployments.

(Optional) Disable SELinux Verification

If SELinux is active in your environment, and you want to deploy SAS Viya with SELinux enabled, add a key-value pair.

1 Open the vars.yml file if it is not already open.
2 Add the following statement at the bottom of the file:
   
   VERIFY_SELINUX: false

3 Save and close the vars.yml file.

For more information about SELinux, see “Configure SELinux” on page 63.

Modify the sitedefault.yml File

The sitedefault.yml file, which is in the /roles/consul/files directory in the playbook, is used primarily for the bulk loading of configuration values to multiple machines. After the initial deployment, you cannot simply modify sitedefault.yml to change an existing value and deploy the software again. You can modify sitedefault.yml only to set property values that have not already been set. Therefore, SAS recommends that you do not use sitedefault.yml for the initial deployment of your SAS Viya software, except where specifically described in this document.

For more information about using sitedefault.yml, see Bulk Loading of Configuration Values (sitedefault.yml) in SAS Viya Administration: Configuration Properties.

Properties for Backup and Restore

The sitedefault.yml file can be used to create a backup configuration for your deployment before the software is deployed. For the steps to create the backup configuration, see “Create a Backup Configuration Using the sitedefault.yml File” in Create and Edit a Backup Configuration in SAS Viya Administration: Backup and Restore.

SAS Viya and Multi-tenancy

Note: If you are performing an upgrade, skip this section, unless you are upgrading a multi-tenant deployment of SAS Visual Investigator or SAS Intelligence and Investigation Management. If you are upgrading a multi-tenant deployment of SAS Visual Investigator or SAS Intelligence and Investigation Management, see specifically Step 3b in “Enable Multi-tenancy” on page 113.

Overview of Multi-tenancy

The term multi-tenant identifies software that allows several groups of users to interact with a single instance of the software. Each group is called a tenant. This section describes how to ensure that your SAS Viya software deployment supports multi-tenancy.

If you anticipate having multiple tenants, even if you have only one tenant initially, you must enable multi-tenancy. Changing from a single-tenant deployment to a multi-tenant deployment requires that you re-deploy your software and make the modifications described in this section. If you do not want your SAS Viya deployment to support multi-tenancy, skip this section.
Considerations for Multi-tenancy and LDAP

SAS recommends that you enter the LDAP information about your multi-tenant deployment with SAS Environment Manager after installing your software (see “Configure the Connection to Your Identity Provider” on page 122). If you use SAS Environment Manager, you must remove the sas.identities.providers.ldap.connection block from your sitedefault.yml file. You can remove the block while performing the steps in the next section. See step 3.

Instead of using SAS Environment Manager, administrators with experience deploying SAS Viya might prefer to complete the sas.identities.providers.ldap.connection block of the sitedefault.yml file. SAS recommends this approach only to automate a deployment. For more information about the variables in the sas.identities.providers.ldap.connection block, see “sas.identities.providers.ldap.connection” in SAS Viya Administration: Configuration Properties.

Enable Multi-tenancy

To enable a multi-tenant deployment:

1. To prevent tenants from being able to read SAS data sets that they do not own, add a value to the vars.yml file for multi-tenancy:
   a. Open vars.yml.
   b. In the STUDIO_CONFIGURATION block, under the init section, add the following variable and value.
   ```
   STUDIO_CONFIGURATION:
   init:
   #sasstudio.appserver.port_comment: '# Port that Studio is listening on'
   #sasstudio.appserver.port: 7080
   #sasstudio.appserver.https.port: 7443
   #webdms.workspaceServer.hostName: localhost
   #webdms.workspaceServer.port: 8591
   webdms.showSystemRoot: false
   ```
   c. Save and close the vars.yml file.

2. If you have not already done so, make a copy of sitedefault_sample.yml and name the copied file sitedefault.yml. The file sitedefault_sample.yml is in the directory where you uncompressed your playbook. If you used the recommended location when you uncompressed the playbook, the location is /sas/install/sas_viya_playbook/roles/consul/files/.

3. Edit sitedefault.yml.

   Note: It is critical that you edit the sitedefault.yml file carefully and check your work thoroughly before saving the file and performing the deployment. This includes using the right values for the variables and maintaining the indentation and spacing of the file. After you set a value with sitedefault.yml, you cannot re-run sitedefault.yml to change that value. You will have to use other tools, such as SAS Environment Manager, to make post-deployment changes to the sitedefault.yml file.
   a. At the beginning of the application block and at the same level as sas.identities.providers.ldap.connection, add the following line:
   ```
   config:
   application:
   sas.multi.tenancy.enabled: true
   ```
   b. On the next line, identify the selected database option:
   ```
   sas.multi.tenancy.db.mode: database-option
   ```
Here are the valid values:

- **schemaPerApplicationTenant** specifies that a single database is shared by all tenants, but each tenant is partitioned into separate schemas based on the application, or microservice. The schema’s name is generated based on the application and tenant name (such as identities_acme) to avoid the possibility of name collisions with multiple applications containing database tables with the same name. This mode is useful when database connection resources are limited because this mode uses a single connection pool for all tenants. A disadvantage is that data for all tenants is secured by a single credential. In addition, connections for all tenants come from a single connection pool, which means that a single tenant can consume all connection resources and deprive other tenants of resources.

  **Note:** The default value is **schemaPerApplicationTenant**. To accept the default, do not add the `sas.multi.tenancy.db.mode` variable.

- **databasePerTenant** specifies that a separate database instance is used for each tenant. This mode is useful for users who prefer maximum isolation of tenant data. In this mode, unique database credentials per tenant enhance security and isolation. However, a disadvantage is that each tenant must have a unique database connection pool, which can significantly increase the total connection count that the back-end database server must support. Additional tuning is required for this mode.

  **Note:** Ensure that you are selecting the database option you truly wish to use. After your software has been deployed, you cannot change the database option without uninstalling and reinstalling your software.

  Here is an example of the code that specifies the non-default value:

  ```
  config:
    application:
      sas.multi.tenancy.enabled: true
      sas.multi.tenancy.db.mode: databasePerTenant
  ```

  **Note:** If you are installing or upgrading a multi-tenant deployment of SAS Visual Investigator or SAS Intelligence and Investigation Management, ensure that **databasePerTenant** is the selected value.

  c If you specified **databasePerTenant** for `sas.multi.tenancy.db.mode` in Step 3b on page 113, you must add parameters for the connection pool.

  ```
  config:
    application:
      sas.multi.tenancy.enabled: true
      sas.multi.tenancy.db.mode: databasePerTenant
    spring:
      datasource.initialSize: ${spring.datasource.tomcat.initialSize}
      datasource.maxIdle: ${spring.datasource.tomcat.maxIdle}
      datasource.maxActive: ${spring.datasource.tomcat.maxActive}
      datasource.minEvictableIdleTimeMillis: ${spring.datasource.tomcat.minEvictableTimeMillis}
      datasource.minIdle: ${spring.datasource.tomcat.minIdle}
      datasource.tomcat.initialSize: 0
      datasource.tomcat.maxIdle: 0
      datasource.tomcat.minEvictableTimeMillis: 30000
      datasource.tomcat.minIdle: 0
      datasource.tomcat.maxActive: 100
  ```

d On the next two lines, list the internal host names that are used to access the provider zone or that are used in a subdomain to access other zones.

  Here is an example:

  ```
  config:
    application:
      sas.multi.tenancy.enabled: true
```
The comma-separated list of internal host names should specify any hosts that will be used to access the provider and any domain from which tenant subdomains will potentially be built. Machines that are included in the [httpproxy] host group in the inventory.ini file should be included in this list. Machines that are included in the [CoreServices] host group in the inventory.ini file should be included in this list if there are multiple domains and they are a subset of each other. Do not use wildcard characters in this list.

A host name from this list with a prepended tenant name will be used as the URL to reach each tenant after each one has been onboarded.

- If you decide to use SAS Environment Manager to set up your LDAP identities, locate the sas.identities.providers.ldap.connection block, and delete it. Otherwise, skip this step. For more information, see “Considerations for Multi-tenancy and LDAP” on page 113.

- Save and close the sitedefault.yml file.

4 Add wildcard versions of the host names to your DNS using the method that your administrator recommends. For example, if you added hostname1.company.com and hostname2.company.com to the internal.hostnames variable in your sitedefault.yml file, you would add *.hostname1.company.com and *.hostname2.company.com to your DNS that point to the host name or IP address of the HTTP server of your SAS installation. The DNS record type is irrelevant. Therefore, you can use A record, AAA record, or CNAME record as long as all tenant-specific subdomains will resolve to your SAS installation.

5 SAS Viya will replace the default self-signed certificate that is installed with the Apache HTTP Server with a certificate that includes a wildcard of subdomains from the fully qualified domain name of the machine. If you update the Apache HTTP Server to use your own custom certificate, make sure it contains subject alternate names for each tenant or uses a wildcard for the subdomain. See Replace Self-Signed Certificates with Custom Certificates (Linux Pre-Deployment) in Encryption in SAS Viya: Data in Motion for more information on how to install a new certificate in the Apache HTTP Server.

After you run the playbook and perform post-installation tasks, your deployment will have a single tenant, which is referred to as the provider. The provider is intended for administration of the multi-tenant environment.

To administer multi-tenancy, including adding more tenants, see Overview in SAS Viya Administration: Multi-tenancy.

---

### Deploy the Software

#### Assessment Test

Before you deploy the software, SAS recommends that you run the following command to assess the readiness of your system for deployment. Before running the command, ensure that you are at the top level of the playbook in the `sas_viya_playbook` directory.

**Note:** For information about what the assessment test validates, see “Set the Pre-deployment Validation Parameters” on page 81.

```bash
ansible-playbook system-assessment.yml
```

Add an option based on the password requirements for the user ID that performs the command:
Table 4.3 Command Options Based on Password Requirements

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td>use the command as written</td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td>--ask-become-pass</td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td>--ask-pass</td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td>--ask-pass --ask-become-pass</td>
</tr>
</tbody>
</table>

If you receive an unexpected error, run the following command to ensure that you are using a supported version of Ansible.

```
ansible-playbook --version
```

Note: For information about supported Ansible versions, see “Ansible Controller Requirements” on page 53.

If you are using a supported version of Ansible and still receive errors from the system assessment, fix those errors before you run the deployment command.

**Deployment Command**

Ensure that you are at the top level of the playbook in the `sas_viya_playbook` directory. Here is the basic syntax for the command to run the playbook and deploy the software:

```
ansible-playbook site.yml [ option ]
```

Add an option based on the password requirements for the user ID that performs the command, using Table 4.3 on page 116. To specify if you want to perform only an installation or configuration, see “Options” on page 116.

In addition, SAS recommends adding a `-vvv` option to enable verbose logging. This option will assist SAS Technical Support in diagnosing any issues you might need to contact them about.

**Options**

To install, but not configure the software, use the same command that is described in “Deployment Command” on page 116, but replace `site.yml` with `install-only.yml`. Here is an example:

```
ansible-playbook install-only.yml --ask-pass --ask-become-pass -vvv
```

To configure software that has been installed only, use the full command that is described in "Deployment Command" on page 116.

**Run from a Directory Other Than the Default**

The playbook runs the commands from the top-level `sas_viya_playbook` directory, by default. If you want to run the playbook from another directory, modify the `ansible.cfg` configuration file with the appropriate configuration options. Refer to the Ansible documentation to find the appropriate `ansible.cfg` file and add those options.

**Successful Playbook Execution**

Here is an example of the output from a successful playbook execution:

```
PLAY RECAP ********************************************************************
```

116 Chapter 4 / Installation
deployTarget : ok=81  changed=65  unreachable=0  failed=0

The most important indicator of success from this message is failed=0.

If the deployment is successful, the software is deployed to the /opt/sas directory.

**Retry a Failed Deployment**

If your deployment fails, and you are able to respond to the error message and can recover from the error, you must restart the deployment using the appropriate deployment commands described in "Assessment Test" on page 115 and any appropriate options.

Failures can occur if there are port conflicts. See "Install with SAS 9.4 Software" on page 117 for a potential source of port conflicts.

**Install with SAS 9.4 Software**

SAS Viya software can be installed on the same machines as an existing SAS 9.4 deployment. No special steps need to be taken at deployment time.

During the deployment, the playbook might halt with an error indicating the ports that SAS Viya needs are in use by the SAS 9.4 deployment. If you receive that error, you should open the vars.yml file in a text editor and search for the variables for the ports that SAS Viya uses. The ports can be found in the following sections of the vars.yml file:

- For SAS/CONNECT, the sasenv_connect_port variable
- For SAS Studio, the sasstudio.appserver.port in the STUDIO_CONFIGURATION block
- For the object spawner, the sasPort in the SPAWNER_CONFIGURATION block

**Deployment Logs**

Logs for Ansible deployments are stored in sas_viya_playbook/deployment.log. If you used the recommended location for uncompressing your playbook, the file is located at /sas/install/sas_viya_playbook/deployment.log.

To view the logs from the yum installation commands that are used in your deployment, run the following commands:

```
sudo yum history
sudo less /var/log/yum.log
```
# Post-installation Tasks

**Configure Security**
- Change the Administrative User Password for SAS Message Broker ........................................ 120
- Configure Your Environment with SAS Environment Manager .................................................. 121

**Configure the Machines**
- Enable Directory Creation for Users of the Compute Server ..................................................... 125
- Configure High Availability in SAS Studio ..................................................................................... 126
- Create a Backup Configuration ....................................................................................................... 128
- (Optional) Create a Local Copy of Documentation ........................................................................ 128
- Configure Locale and Encoding ...................................................................................................... 129

**Configure Data Access**
- SAS Data Preparation and Data Sources ....................................................................................... 131
- Configure SAS/ACCESS Interface to Amazon Redshift ................................................................. 131
- Configure SAS/ACCESS Interface to DB2 ...................................................................................... 132
- Configure SAS/ACCESS Interface to Google BigQuery ............................................................... 132
- Configure SAS/ACCESS Interface to Greenplum .......................................................................... 133
- Configure SAS Data Connector to Hadoop .................................................................................... 134
- Configure SAS/ACCESS Interface to HAWQ ............................................................................... 135
- Configure SAS/ACCESS Interface to Impala ............................................................................... 136
- Configure SAS/ACCESS Interface to JDBC .................................................................................. 137
- Configure SAS/ACCESS Interface to Microsoft SQL Server ......................................................... 138
- Configure SAS/ACCESS Interface to MySQL ............................................................................... 138
- Configure SAS/ACCESS Interface to Netezza ............................................................................. 139
- Configure SAS/ACCESS Interface to ODBC ............................................................................... 140
- Configure SAS/ACCESS Interface to Oracle ................................................................................ 141
- Configure SAS/ACCESS Interface to PostgreSQL ....................................................................... 142
- Configure SAS/ACCESS Interface to SAP ASE .......................................................................... 143
- Configure SAS/ACCESS Interface to SAP HANA ....................................................................... 143
- Configure SAS/ACCESS Interface to SAP R/3 ............................................................................ 144
- Configure SAS/ACCESS Interface to Snowflake ......................................................................... 144
- Configure SAS Data Connector to Spark ...................................................................................... 145
- Configure SAS/ACCESS Interface to Teradata ............................................................................ 146
- Configure SAS/ACCESS Interface to Vertica ............................................................................... 146
- (Optional) Configure Java for the ACCELWHERE Option .............................................................. 146

**Configure Model Access**
- Configure Access to Models .......................................................................................................... 147
- Configure Access to Analytic Store Model Files .......................................................................... 147

**Configure SAS Analytics for IoT**
- Offerings Included with SAS Analytics for IoT ............................................................................ 147

**Configure SAS Data Preparation and SAS Data Agent**
- SAS Data Agent Deployment Scenarios ........................................................................................ 148
Configure SAS Data Quality

Configure the Quality Knowledge Base

Configure SAS Data Science

Software Order Associated with SAS Data Science

Configure SAS Decision Manager

Configure Access to Analytic Store Model Files

Configure SAS Event Stream Processing

Complete SAS Event Stream Processing and SAS Analytics for IoT Setup

Complete SAS Event Stream Manager Setup

Configure SAS Intelligent Decisioning

Configure Analytic Store File System Paths

Configure SAS Model Manager

Configure the Workflow Client User

Configure Access to Analytic Store Model Files

Configure Security

Change the Administrative User Password for SAS Message Broker

Note: The tasks in this section are applicable if you deployed all your software. If you deployed the programming interface only, skip this section.

You must change the administrative user password for SAS Message Broker as soon as possible after you have deployed SAS Viya.

1. Locate a machine that you have previously assigned to the [rabbitmq] host group in the inventory file. This machine is the message broker machine.

2. Sign on to the message broker machine with sudo privileges.

3. Change to this directory:

   /opt/sas/viya/home/bin

4. Run the message broker account tool with these arguments:

   sudo ./sas-rabbitmq-acc-admin change_passwd -t account-type -u user-ID --promptpw

   -t account-type specifies the account user type, which is always the client type. The client user has full administrative rights. These rights can change in future releases.

   -u user-ID identifies the client user ID for SAS Message Broker. By default, the user ID out of the box is "sasclient".

   --promptpw prompts for the new password for the client user ID for SAS Message Broker. The password that you enter is hidden, by default.

Here is an example that changes the password for the default administrative user:

   sudo ./sas-rabbitmq-acc-admin change_passwd -t client -u sasclient --promptpw
5 Restart all SAS Viya services. Restarting the SAS Viya services activates the changes to the credentials for SAS Message Broker. For more information, refer to SAS Viya Administration: General Servers and Services.

**Configure Your Environment with SAS Environment Manager**

Note: The tasks in this section are applicable if you deployed all your software. If you deployed the programming interface only, skip this section.

**Sign In as the sasboot User**

Your SAS environment is deployed with an initial administrator account that is named sasboot. The password for this account has expired by default, so you must reset the password before you can sign in.

To reset the password:

1. Locate the most recent log for the SAS Logon service in `/var/log/sas/viya/saslogon/default`.
   
   **Note:** SAS Logon is installed on one or more machines to which you are assigned in the CoreServices host group in the inventory file. For information about the inventory file, see “Edit the Inventory File” on page 74.

2. Search the log for the characters, `sasboot`.
   
   ```bash
grep 'sasboot' /var/log/sas/viya/saslogon/default/sas-saslogon_date-and-time-stamp.log
   ```
   
   Here is a typical message:
   
   Reset password for initial user sasboot using link: /SASLogon/reset_password?code=xxxxxx

3. Sign in from a URL with this format:
   
   ```bash
   https://reverse-proxy-server/SASLogon/reset_password?code=password
   ```
   
   **Note:** Use the host name from the machine that you assigned to the [httpproxy] host group in the inventory file. For information about the inventory file, see “Edit the Inventory File” on page 74.

   Make a note of this URL to share with any other users of your SAS Viya software, as described in “Share Important Deployment Information with the Administrators” on page 177.

4. Follow the instructions on the displayed web page to reset the password.
   
   **Note:** If the URL has expired, run the following command:
   
   - For Red Hat Enterprise Linux 6.x or an equivalent distribution:
     ```bash
     sudo service sas-viya-saslogon-default stop
     ```
   - For Red Hat Enterprise Linux 7.x or an equivalent distribution:
     ```bash
     sudo systemctl stop sas-viya-saslogon-default
     ```
     
   Then go to the log and obtain the new URL. The URL expires 24 hours after the SAS Logon service restarts. For security purposes, the URL that is specified in a browser or in a text editor also expires, even if the password is not reset.

   After you reset the password, SAS Environment Manager automatically opens in your browser.

5. Click **Yes** for all of the assumable groups so that you have the permissions to perform subsequent tasks.
Configure the Connection to Your Identity Provider

After completing the installation of SAS Viya, you must configure the connection to your identity provider before your users can access SAS Environment Manager and SAS Visual Analytics.

While signed in as sasboot, configure the connection to your identity provider:

Note: Only LDAP-based identity providers are supported. You need to have basic familiarity with LDAP administration. For more information about the properties that are relevant for this procedure, see “sas.identities.providers.ldap” in SAS Viya Administration: Configuration Properties.

1. Select the $\text{Identity Provider}$ from the side menu to open the Configuration page.

2. On the Configuration page, select Basic Services from the list, and then select the Identities service from the list of services.

3. To configure user properties, in the sas.identities.providers.ldap.user section, click New Configuration:

   a. Specify a value for the baseDN required field. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

      Note: When using the LDAP protocol, passwords are transmitted over the network as clear-text. To secure the deployment, SAS recommends that you configure encrypted LDAP connections. For more information, see Encrypt LDAP Connections in Encryption in SAS Viya: Data in Motion.

      For each property that represents a user-level field in SAS, specify a corresponding property in the LDAP server software.

      TIP Consider specifying a custom filter to limit the user accounts that SAS Viya returns from your LDAP server.

   b. If you are not performing a multi-tenant deployment, skip this step.

      If you are using a custom LDAP structure, select the Apply configuration only to this tenant (provider) option.

      If you are using the fixed LDAP structure that is described in “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46, do not select this option.

   c. Click Save.

4. To configure group properties, in the sas.identities.providers.ldap.group section, click New Configuration. In the New Configuration window:

   a. Specify a value for the baseDN required field. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

      For each property that represents a group-level field in SAS, specify a corresponding property in the LDAP server software.

      TIP Consider specifying a custom filter to limit the accounts that SAS Viya returns from your LDAP server.

   b. If you are not performing a multi-tenant deployment, skip this step.

      If you are using a custom LDAP structure, select the Apply configuration only to this tenant (provider) option.

      If you are using the fixed LDAP structure that is described in “Set Up Accounts for Multi-tenant Deployments: Single LDAP Server for All Tenants” on page 46, do not select this option.
c Click **Save**.

5 To configure connection properties, in the `sas.identities.providers.ldap.connection` section, click **New Configuration**. In the New Configuration window:

a Specify values for the following required fields: `host`, `password`, `port`, `url`, and `userDN`. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

b If you are not performing a multi-tenant deployment, skip this step.

   If you are using an LDAP server per tenant, select the **Apply configuration only to this tenant (provider)** option.

   If you are using a single LDAP server for all tenants, or a fixed directory structure that applies to all tenants, do not select this option.

c Click **Save**.

6 To verify user and group information, from the SAS Environment Manager side menu, select **Users** to open the Users page.

   On the Users page, select **Users** from the list in the toolbar. Your users should appear after a few minutes. It is not necessary to restart any servers or services. Then select **Groups** from the list to display your groups.

7 Verify that user and group information is displayed correctly. If not, make any necessary changes to the identities service properties, then restart the Identities and SAS Logon Manager services:

- For Red Hat Enterprise Linux 6.7:

  ```bash
  sudo service sas-viya-identities-default restart
  sudo service sas-viya-saslogon-default restart
  ```

- For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

  ```bash
  sudo systemctl restart sas-viya-identities-default
  sudo systemctl restart sas-viya-saslogon-default
  ```

---

**Set Up Administrative Users**

While you are signed on to SAS Environment Manager as the sasboot user, set up at least one SAS Administrator user, as follows:

1 On the Users page in SAS Environment Manager, select **Custom Groups** from the list in the toolbar.

2 In the left pane, click **SAS Administrators**.

3 In the **Members** section of the right pane, click **`,`,** and add one or more members to the group (including your own account, if applicable).

4 Sign out from SAS Environment Manager so that you are no longer signed in as the sasboot user.

5 If you added your own account to the SAS Administrators group, you can sign on again to SAS Environment Manager using that account.

   Open SAS Environment Manager from a URL with the following format:

   ```plaintext
   https://reverse-proxy-server/SASEnvironmentManager
   ```

**TIP** Since SAS Administrators is an assumable group, the following prompt is displayed: **Do you want to opt in to all of your assumable groups?**. Select **Yes** if you want the extra
permissions that are associated with the SAS Administrators group. The selection remains in effect until you sign out.

**Sign In Using LDAP Credentials**

Open SAS Environment Manager from a URL with the following format:

https://reverse-proxy-server/SASEnvironmentManager

Sign in as one of the SAS Administrators that you set up in “Set Up Administrative Users” on page 123.

**Configure the Connection to the Mail Service**

After installing a new SAS Viya deployment, you must configure the connection to your mail service. Complete these steps while you are signed in as one of the SAS Administrators.

1. Select the ➕ from the side menu to open the Configuration page.

2. On the Configuration page, select **Basic Services** from the list, and then select **Mail service** from the list of services.

3. In the **sas.mail** section, click ✉️. In the Edit Configuration window, follow these steps:
   a. Specify a value for the following required fields: **host** and **port**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.
   b. Click **Save**.

4. (Optional) To enable the health check for the mail service, perform the following steps.
   a. Select the ➕ from the side menu to open the Configuration page.
   b. On the Configuration page, select **Basic Services** from the list, and then select **Mail service** from the list of services.
   c. In the **management.health.mail** section, click ✉️.
   d. Turn the **enabled** toggle to **on**.
   e. Click **Save**.

When this toggle is set, health checks will be enabled after the mail service is restarted. If the mail host is not configured or is configured incorrectly, or if it cannot connect to the SMTP mail server, the mail service will indicate it is in a failed state.

- For Red Hat Enterprise Linux 6.7:
  ```bash
  sudo service sas-viya-mail-default restart
  ```

- For Red Hat Enterprise Linux 7.x or later and SUSE Linux:
  ```bash
  sudo systemctl restart sas-viya-mail-default
  ```

**Disable the Password Reset Feature and Reset the sasboot Password**

When you are finished setting up LDAP and the initial administrative users, you should reset the password for the sasboot user. For additional security, you can then disable the password reset feature. This action prevents password reset links from being written to the log each time the SASLogon service is restarted.
1. Sign in to SAS Environment Manager as an administrative user and select Configuration from the side menu to open the Configuration page.

2. On the Configuration page, select Definitions from the drop-down list.

3. In the left pane, select sas.logon.initial. Then click New Configuration at the top of the right pane. If a definition already exists, you can select to edit the existing definition.

4. In the New sas.logon.initial Configuration window or the Edit sas.logon.initial Configuration window, set reset.enabled to off.

5. Click Save.

6. Restart the SAS Logon Manager services:
   - For Red Hat Enterprise Linux 6.7:
     
     ```
     sudo service sas-viya-saslogon-default restart
     ```
   - For Red Hat Enterprise Linux 7.x or later and SUSE Linux:
     
     ```
     sudo systemctl restart sas-viya-saslogon-default
     ```

   For more information, see General Servers and Services: Operate in SAS Viya Administration: Servers and Services.

   Note: After you disable this feature, you can still change the sasboot password if the existing password is known. Enter the URL for SAS Viya with the path /SASLogon/change_password. If you are already signed in as another user, first sign out and then sign back in as sasboot using the current password. You can then complete the steps to change the password.

**Review the Default Blacklist Paths**

When SAS Viya is deployed, the default blacklist is installed to protect sensitive deployment objects from unauthorized access. Review these default blacklist paths. For details, see Manage Path Lists (Whitelists and Blacklists) in SAS Viya Administration: SAS Cloud Analytic Services.

**Configure SAS Viya to Encrypt the LDAP Connection**

SAS Viya supports encrypted connections between the LDAP client and server. To configure a secure LDAP connection, see Encrypt LDAP Connections in Encryption in SAS Viya: Data in Motion.

---

Configure the Machines

**Enable Directory Creation for Users of the Compute Server**

The SAS compute server and compute service enable users to submit SAS programs and jobs for processing. Several SAS Viya products use the compute service to process programs and statements that were written in the SAS language, including SAS Analytics for IoT, SAS Decision Manager, SAS Intelligent Decisioning, SAS Model Manager, SAS Studio 5.X, SAS Visual Data Mining and Machine Learning, and SAS Visual Forecasting.

Users of these products require a home directory that can be accessed by the compute server each time that a process starts. For each user, the home directory location should be defined in /etc/passwd or in UNIX Attributes in Active Directory. If home directories will be local to the machine target for the compute server installation, you can enable the automatic creation of these directories. When automatic creation of directories is
enabled, SAS Viya checks for the existence of a user’s home directory whenever that user’s compute server session starts. If the home directory is not found, it is created.

Follow these steps to enable this feature:

1. Configure the SASMAKEHOMEDIR environment variable. Run the following commands on the machine where the SAS Configuration Server is installed:

    Note: Enter the second command on a single line. Multiple lines are used here to improve readability.

    ```
    . /opt/sas/viya/config/consul.conf
    /opt/sas/viya/home/bin/sas-bootstrap-config --token-file /opt/sas/viya/config/etc
    /SASSecurityCertificateFramework/tokens/consul/default/client.token kv write
    config/launcher-server/global/environment/SASMAKEHOMEDIR 1
    ```

2. The default directory permissions are set to 0700, but you can change them if necessary. On the same machine, run the following commands:

    Note: Enter the second command on a single line. Multiple lines are used to improve readability.

    ```
    . /opt/sas/viya/config/consul.conf
    /opt/sas/viya/home/bin/sas-bootstrap-config --token-file /opt/sas/viya/config/etc
    /SASSecurityCertificateFramework/tokens/consul/default/client.token kv write
    config/launcher-server/global/environment/SASHOMEDIRPERMS permissions
    ```

    For permissions, specify the permissions to apply to the local home directories that are created. Specify permissions in octal format. The default setting provides Read/Write/Execute permissions to the user only. Supported values are between 0700 and 0777.

3. Restart the runlauncher service. Run the following commands on Red Hat Enterprise Linux 6.x:

    ```
    cd /etc/init.d/
    sudo sas-viya-runlauncher-default restart
    ```

    Run the following command on Red Hat Enterprise Linux 7.x or SUSE Linux:

    ```
    sudo systemctl restart sas-viya-runlauncher-default
    ```

**Configure High Availability in SAS Studio**

Note: The tasks in this section are applicable only if you deployed the programming-only interface.

Note: A shared file system is required. For details, see “High-Availability Requirements” on page 28.

1. Identify your programming hosts. The programming proxy hosts are the hosts that have been listed in the [programming] host group in the inventory file. For details, see “Assign the Target Machines to Host Groups” on page 77.

2. For each programming host, do the following:
   a. Stop SAS Studio.
      - For Red Hat Enterprise Linux 6.7:
        ```
        sudo service sas-viya-sasstudio-default stop
        ```
      - For Red Hat Enterprise Linux 7.x and SUSE Linux:
        ```
        sudo systemctl stop sas-viya-sasstudio-default
        ```
   b. Determine the unique IP address of each SAS Studio instance.
      ```
      hostname -i
      ```
c. Locate and edit the SAS Studio configuration file `SASCONFIG/etc/sasstudio/default/init_usermods.properties`.

d. Change the following line and add the unique IP address for that SAS Studio instance.

```
sasstudio.appserver.instanceid=sasstudio-<IP-Address>
```

Note: When you enter the IP address, replace the periods (xxx.xxx.xxx.xxx) with hyphens (xxx-xxx-xxx-xx). An example is 123-123-123-123.

e. Save and close the SAS Studio configuration file.

After you have completed the preceding steps on all programming hosts, you have a list of IP addresses for each programming host.

```
<IP address of first SAS Studio host>
<IP address of second SAS Studio host>
<IP address of third SAS Studio host>
```


   - For Red Hat Enterprise Linux 6.7:
     ```
sudo service sas-viya-sasstudio-default start
     ```
   
   - For Red Hat Enterprise Linux 7.x or later and SUSE Linux:
     ```
sudo systemctl start sas-viya-sasstudio-default
     ```

4. Identify your http proxy hosts. The http proxy hosts are the hosts that have been listed in the `[httpproxy] host` group in the inventory file.

5. On each http proxy host, do the following:
   a. Locate and edit the `proxy.conf` file.
      - For Red Hat Enterprise Linux, the `proxy.conf` file is in the `/etc/httpd/conf.d` directory.
      - For Linux SUSE, the `proxy.conf` file is in the `/etc/apache2/conf.d` directory.
   b. Locate and remove (or comment out) any existing lines that contain `ProxyPass` or `ProxyPassReverse`:
      ```
      # ProxyPass /SASStudio http://SAS-Studio-host:7080/SASStudio
      # ProxyPassReverse  /SASStudio http:// SAS-Studio-host:7080/SASStudio
      ```
   c. Add the following lines to map the balancers. Substitute the appropriate hosts, ports, and IP addresses.

      Note: When you enter the IP address, replace the periods (xxx.xxx.xxx.xxx) with hyphens (xxx-xxx-xxx-xxx). An example is 123-123-123-123.

      ```
      <Proxy balancer://SASStudio-cluster>
      BalancerMember http://SAS-Studio-host-1:SAS-Studio-port/ route=sasstudio-SAS-Studio-host-1-IP
      ProxySet scolonpathdelim=on stickysession=JSESSIONID
      </Proxy>
      ProxyPass /SASStudio balancer://SASStudio-cluster/SASStudio
      ProxyPassReverse /SASStudio balancer://SASStudio-cluster/SASStudio
      ```

      Here is an example:

      ```
      <Proxy balancer://SASStudio-cluster>
      BalancerMember http://hosta.company.com:7080/ route=sasstudio-100-10-0-1
      BalancerMember http://hostb.company.com:7080/ route=sasstudio-100-10-0-2
      ProxySet scolonpathdelim=on stickysession=JSESSIONID
      </Proxy>
      ProxyPass /SASStudio balancer://SASStudio-cluster/SASStudio
      ```
ProxyPassReverse /SASStudio balancer://SASStudio-cluster/SASStudio

d Save and close the proxy.conf file.

Ensure that you modified the proxy.conf file for each http proxy host.

6 On each http proxy host, start httpd:
   - For Red Hat Enterprise Linux 6.7:
     sudo service httpd restart
   - For Red Hat Enterprise Linux 7.x or later:
     sudo systemctl restart httpd
   - For SUSE Linux:
     sudo systemctl restart apache2

7 On each programming machine, open SAS Studio from a URL with this format:
   http://reverse-proxy-server/SASStudio

Create a Backup Configuration

The tasks in this section are applicable only if you deployed all your software. If you deployed the programming interface only, skip this section. For information about backing up a programming-only deployment, see Backing Up and Restoring Programming-Only Deployments in SAS Viya Administration: Backup and Restore.

1 In SAS Environment Manager, confirm that the DEFAULT_BACKUP_SCHEDULE has been created. For details, see Initial Tasks in SAS Viya Administration: Backup and Restore.

2 Check the logs at /opt/sas/viya/config/var/log/deploymentBackup/default and /opt/sas/viya/config/var/log/backup-agent/default. If the following message is in the deploymentBackup log, restart the deploymentBackup service.

   ServiceSchedule] c.sas.backup.util.BackupScheduleManager : service [BACKUP_SCHEDULE_ERROR] Cannot schedule backup since maximum retry attempt is reached and one of the dependent services is still not running

   Restart the deploymentBackup service. Confirm that the following message is now in the log:

   ServiceSchedule] c.sas.backup.util.BackupScheduleManager : service Default schedule created for BackupService to run backup job every Sunday 1AM

3 Set the sharedVault location and ensure that the permissions on the designated location are set. For details, see Initial Tasks in SAS Viya Administration: Backup and Restore.

4 In SAS Environment Manager, click Jobs.

5 Right-click the DEFAULT_BACKUP_SCHEDULE and then select Run from the pop-up menu to immediately run the backup.

6 To confirm that the backup ran successfully, in the Jobs list, click the Monitoring tab.

7 On the Monitoring tab of the Jobs page, ensure that the jobs are running without any warnings and errors.

(Optional) Create a Local Copy of Documentation

You can configure your software to give your users access to local documentation. Here are two instances where access to local documentation would be useful:
You have customized your documentation.

Your SAS system is highly secure, and it does not have access to the internet. Because the SAS documentation is cloud-hosted, it cannot be reached without internet access.

**Note:** The cloud-hosted SAS documentation is frequently updated. The SAS administrator should refresh the local copy on a regular basis to ensure that your users have up-to-date information.

You can download PDF versions of the documentation, or you can create customized versions of the documentation. Create an HTML page with links to all documents that make up your local documentation collection, and create a link to this page.

To configure local documentation:

1. Access SAS Environment Manager.
2. Select **Configuration** from the left navigation bar.
3. Under the **View** menu, select **Definitions**.
4. Select the sas.htmlcommons definition.
5. Click **New Configuration**.
6. On the New sas.htmlcommons Configuration pane, click **Add Property** to add the following two properties:
   - `additionalHelpMenuUrl` — Specify the path to the HTML page that contains links to your local documentation.
   - `additionalHelpMenuLabel` — Provide a meaningful label for the link that your users can access.
     If you do not provide this parameter, a default label of **Additional Help** is used.
7. Click **Save** on the New htmlcommons Configuration pane.

Users see a new item in the **Help Menu** list, between the **Help Center** and **About** entries. Clicking this link opens the specified HTML page.

### Configure Locale and Encoding

SAS Viya supports all the SAS session encodings that are available in SAS 9.4. By default, SAS Viya uses an encoding of UTF-8 and a locale of en_US. You can change the SAS LOCALE option, the SAS ENCODING option, or both options.

#### Defaults for Locale and Encoding

The SAS LOCALE is set to `en_US` for the SAS programming run-time environment in `nls/u8/sasv9.cfg`.

The SAS ENCODING is set to `UTF-8` for the SAS programming run-time environment in `/opt/sas/spre/home/SASFoundation/nls/u8/sasv9.cfg`.

The `sas` command in `/opt/sas/spre/home/SASFoundation` is a symbolic link to `/opt/sas/spre/home/SASFoundation/bin/sas_u8`, which uses the encoding value set in `/opt/sas/spre/home/SASFoundation/nls/u8/sasv9.cfg`.

#### Configure the SAS LOCALE or ENCODING Value

If the defaults for LOCALE and ENCODING for the SAS programming run-time environment do not meet your needs, you can change these values using the post-installation task named `build_default_locale.task`.

**Note:** You must run `build_default_locale.task` as root or as a user with sudo privileges.
When run, build_default_locale.task updates the appropriate sasv9.cfg files. If you change the SAS ENCODING value, the task also updates the sas symbolic link.

The task supports these options:

locale
  is a valid locale name.

encoding
  is a valid encoding name.

-h
  displays usage information for the command.

? 
  displays usage information for the command.

The locale and encoding values that you specify must match those values for the operating system. The spelling and punctuation might be different from the SAS locale or encoding name. For example, on CentOS 7.2, the SAS ENCODING, latin9, is specified as iso885915@euro. For valid values, see Encodings and Their Aliases in SAS 9.4 and SAS Viya Programming Documentation National Language Support (NLS).

Configure the SAS LOCALE Option for SAS Running with UTF-8

To change the default SAS LOCALE option, run build_locale_default.task and specify the -locale option.

For example, if you want to change the default locale for the SAS UTF-8 server from en_US to fr_FR, run this command from your Linux shell:

```
/opt/sas/spre/home/SASFoundation/install/install.d/build_default_locale.task -locale fr_FR
```

Consider these tips about using the LOCALE option:

- You can override the LOCALE option setting for your SAS session by setting the LOCALE option on the command line.
- You can change the LOCALE option during your SAS session by setting the LOCALE option in the OPTIONS statement.

Note: SAS Workspace Server automatically uses the locale that matches the locale that is sent by the client. The LOCALE option value from sasv9.cfg and sasv9_local.cfg does not affect a SAS Studio session. If a locale is set in the sasv9_local.cfg file, that locale is set for SAS programs that are run in batch mode.

Configure the SAS LOCALE and ENCODING Options

To change the SAS ENCODING and LOCALE options, run build_locale_default.task and specify both -locale and -encoding on the command line.

For example, to configure SAS with the regional encoding for Simplified Chinese on CentOS 7.2, run build_default_locale.task with these options:

```
/opt/sas/spre/home/SASFoundation/install/install.d/build_default_locale.task -locale zh_CN -encoding gbk
```

When the task executes, the following code is displayed in the Linux shell:

```
- LOCALE   = zh_CN
- ENCODING = euc-cn
SAS executable used     : zh
SAS 2-byte Language code: zh
Modifying sas link file = bin/sas_zh
Build default locale task completed.
```
When the task runs, the `sas` symbolic link is modified to point to the `bin/sas_zh` script. The `sas_zh` script concatenates the `nls/zh/sasv9.cfg` file that sets the `LOCALE` and `ENCODING` options as follows:

```
-LOCALE zh_CN
-ENCODING euc-cn
```

Note: The `LANG` environment variable setting in your Linux shell must match the locale and encoding that you plan to select for SAS Foundation and SAS Workspace Server.

Here is an example:

```
LANG=ja_JP.eucjp; export LANG
```

---

**Configure Data Access**

**SAS Data Preparation and Data Sources**

If you are deploying only SAS Data Preparation in the cloud and it will only access databases in a SAS Data Agent on-premises deployment, you have no data sources to configure and should skip this section.

**Configure SAS/ACCESS Interface to Amazon Redshift**

Note: This information is applicable only if you ordered SAS/ACCESS Interface to Amazon Redshift (on SAS Viya).

If SAS/ACCESS Interface to Amazon Redshift was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Amazon Redshift without having to run the playbook again.

1. To reference a Data Source Name (DSN) in your connection, add the DSN to the `odbc.ini` file.
   a. On the SAS client node, edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition.
   b. On the CAS node, edit the `/opt/sas/viya/home/lib64/accessclients/odbc.ini` and add your DSN definition.

   For an example DSN definition, see the [Amazon RedShift Wire Protocol] template in the `odbc.ini` file.

2. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

3. Add the following lines to each file:
   ```bash
   export ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   export ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   export ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```

4. Save and close the `workspaceserver_usermods.sh` and `sas-compsrv` files.

5. Using a text editor, open the `cas_usermods.settings` file.
   ```bash
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

6. Add the following lines:
   ```bash
   export ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
   ```
export ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
export ODBCHOME=/opt/sas/viya/home/lib64/accessclients
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib

7 Save and close the cas_usermods.settings file.

8 Edit the vars.yml file to include the configuration settings as documented in “SAS/ACCESS Interface to SAP ASE” on page 95.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to DB2

   Note: This information is applicable only if you ordered SAS/ACCESS Interface to DB2 (on SAS Viya).

If SAS/ACCESS Interface to DB2 was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to DB2 without having to run the playbook again.

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2 Add the following lines to each file:
   ```
   export CLASSPATH=$CLASSPATH:DB2-related-classpath
   export DB2INSTANCE=DB2-instance
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-DB2-installation
   ```

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 Using a text editor, open the cas_usermods.settings file.
   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

5 Add the following lines:
   ```
   export DB2INSTANCE=DB2-instance
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-DB2-installation
   ```

6 Save and close the cas_usermods.settings file.

7 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to DB2” on page 88.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to Google BigQuery

There are no additional configuration steps required for SAS/ACCESS to Google BigQuery.

Configure SAS/ACCESS Interface to Greenplum

   Note: This information is applicable only if you ordered SAS/ACCESS Interface to Greenplum (on SAS Viya).

If SAS/ACCESS Interface to Greenplum was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Greenplum without having to run the playbook again.

1 To reference a Data Source Name (DSN) in your connection, add the DSN to the odbc.ini file.
a On the SAS client node, edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition.

b On the CAS node, edit the `/opt/sas/viya/home/lib64/accessclients/odbc.ini` and add your DSN definition.

For an example DSN definition, see the [Greenplum Wire Protocol] template in the odbc.ini file.

2 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

3 Add the following lines to each file:

   **Note:** Depending on how you have configured your ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

   ```bash
   export ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   export ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   export ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```

   For bulk loading, add the following lines.

   ```bash
   export GPHOME_LOADERS=Greenplum-bulk-loader-installation-location
   export GPLOAD_HOME=Greenplum-installation-location
   export GPLOAD_PORT=Greenplum-bulk-load-port
   ```

4 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

5 Edit the vars.yml file to include the configuration settings as documented in “SAS/ACCESS Interface to Greenplum” on page 89.

   **Note:** The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

---

**Configure SAS Data Connector to Hadoop**

**Note:** The information in this section is applicable only if you ordered SAS/ACCESS Interface to Hadoop (on SAS Viya).

If SAS/ACCESS Interface to Hadoop was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Hadoop without having to run the playbook again.

To manually configure the variables:

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

2 Add the following lines to each file:

   ```bash
   export JAVA_HOME=location-of-your-Java-8-JRE
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
   ```

   If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is `/usr/lib/jvm/jre-1.8.0`. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.
If you are using MapR, add the following line:
```
export MAPR_HOME=/opt/mapr
```

Save and close the workspaceserver_usermods.sh and sas-compsrv files.

On each CAS node, use a text editor to edit the cas_usermods.settings file:
```
sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
```

Add the following lines:
```
export JAVA_HOME=location-of-your-Java-8-JRE
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

If you are using MapR, add the following line:
```
export MAPR_HOME=/opt/mapr
```

Save and close the cas_usermods.settings file.

Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop” on page 89.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to HAWQ

Note: This information is applicable only if you ordered SAS/ACCESS Interface to HAWQ (on SAS Viya).

If SAS/ACCESS Interface to HAWQ was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to HAWQ without having to run the playbook again.

To reference a Data Source Name (DSN) in your connection, add the DSN to the odbc.ini file.

- On the SAS client node, edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition.

- On the CAS node, edit the `/opt/sas/viya/home/lib64/accessclients/odbc.ini` and add your DSN definition.

For an example DSN definition, see the [Greenplum Wire Protocol] template in the odbc.ini file.

For each host that is specified in the [programming] host group, use a text editor to open the following files:

- `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
- `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

Add the following lines to each file:

Note: Depending on how you have configured your ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

```
export ODBCINI="/opt/sas/spre/home/lib64/accessclients/odbc.ini"
export ODBCINST="/opt/sas/spre/home/lib64/accessclients/odbcinst.ini"
export ODBCHOME="/opt/sas/spre/home/lib64/accessclients"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```
Save and close the workspaceserver_usermods.sh and sas-compsrv files.

Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to HAWQ” on page 90.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to Impala

Note: This information is applicable only if you ordered SAS/ACCESS Interface to Impala (on SAS Viya).

If SAS/ACCESS Interface to Impala was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Impala without having to run the playbook again.

1. Install a third-party ODBC Driver Manager. The Impala ODBC driver is an ODBC API-compliant shared library. In addition, the Impala ODBC driver requires that you also install a third-party ODBC Driver Manager. A version of the unixODBC Driver Manager is available for download from the unixODBC website http://www.unixodbc.org/.

2. To enable the Impala driver to be loaded dynamically at run time, include the full pathname of the shared library in the shared library path.

3. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

4. Add the following lines to each file:

   Note: Multiple lines are used for LD_LIBRARY_PATH to improve readability. However, in your environment, make sure that you enter the command on a single line.

   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
   export EASYSOFT_UNICODE=YES
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
   location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:opt/cloudera/impalaodbc/lib/64
   ```

   Note: The EASYSOFT_UNICODE variable should only be added if you want to set the encoding for the SAS client to UTF-8.

5. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

6. Using a text editor, open the cas_usermods.settings file.

   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

7. Add the following lines:

   Note: Multiple lines are used for LD_LIBRARY_PATH to improve readability. However, in your environment, make sure that you enter the command on a single line.

   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
   location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:opt/cloudera/impalaodbc/lib/64
   ```

8. Save and close the cas_usermods.settings file.
To use an Impala ODBC driver from a different vendor than SAS/ACCESS Interface to Impala on SAS Viya, set either the SAS_IMPALA_DRIVER_VENDOR environment variable or the DRIVER_VENDOR connection option. Here are some examples:

- Set the environment variable to use the MapR Impala ODBC driver:

  ```
  SAS_IMPALA_DRIVER_VENDOR=MAPR
  export SAS_IMPALA_DRIVER_VENDOR
  ```

- When defining the caslib, set the DRIVER_VENDOR variable to use the Progress DataDirect Impala ODBC driver:

  ```
  action addCaslib lib="datalib" datasource={srctype="impala", server="impserver", schema="default", DRIVER_VENDOR="DATADIRECT"} ; run
  ```

Currently, the only valid values for the driver vendor are DATADIRECT and MAPR.

Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Impala” on page 90.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to JDBC

Note: This information is applicable only if you ordered SAS/ACCESS Interface to JDBC (on SAS Viya).

If SAS/ACCESS Interface to JDBC was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to JDBC without having to run the playbook again.

1. For each host that is specified in the [programming] host group, use a text editor to open the following files:

   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

2. Add the following lines to each file:

   ```
   export JAVA_HOME=location-of-your-Java-8-JRE
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
   ```

   If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

3. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4. On each CAS node, use a text editor to edit the cas_usermods.settings file:

   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

5. Add the following lines:

   ```
   export JAVA_HOME=location-of-your-Java-8-JRE
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
   ```

   If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

6. Save and close the cas_usermods.settings file.
Edit the vars.yml file to include the configuration settings. For information, see "SAS/ACCESS Interface to JDBC" on page 91.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to Microsoft SQL Server

Note: This information is applicable only if you ordered SAS/ACCESS Interface to Microsoft SQL Server (on SAS Viya).

If SAS/ACCESS Interface to Microsoft SQL Server was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Microsoft SQL Server without having to run the playbook again.

1. To reference a Data Source Name (DSN) in your connection, add the DSN to the odbc.ini file.
   a. On the SAS client node, edit the /opt/sas/spre/home/lib64/accessclients/odbc.ini file and add your DSN definition.
   b. On the CAS node, edit the /opt/sas/viya/home/lib64/accessclients/odbc.ini and add your DSN definition.

For an example DSN definition, see the [SQL Server Wire Protocol] template in the odbc.ini file.

Note: If your deployment will have encryption enabled, perform the following steps on the client machine and the CAS nodes:
   - Add the SSLLibName connection option to the DSN and set it to the absolute path for the OpenSSL SSL library file. Here is an example:
     SSLLibName=/usr/lib64/libssl.so
   - Add the CryptoLibName connection option to the DSN and set it to the absolute path for the OpenSSL cryptographic library file. Here is an example
     CryptoLibName=/usr/lib64/libcrypto.so

2. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

3. Add the following lines to each file:
   ```
   export ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
   export ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
   export ODBCHOME=/opt/sas/spre/home/lib64/accessclients
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```

4. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

5. On each CAS node, use a text editor to edit the cas_usermods.settings file:
   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

6. Add the following lines:
   ```
   export ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
   export ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
   export ODBCHOME=/opt/sas/viya/home/lib64/accessclients
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```
Save and close the cas_usermods.settings file.

Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Microsoft SQL Server” on page 92.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to MySQL

Note: This information is applicable only if you ordered SAS/ACCESS Interface to MySQL (on SAS Viya).

If SAS/ACCESS Interface to MySQL was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to MySQL without having to run the playbook again.

1. On the CAS machine(s), use a text editor to edit the cas_usermods.settings file:
   
   `sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings`

2. Add the following line:
   
   `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-MySQL-client-library`

3. Save and close the cas_usermods.settings file.

4. Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to MySQL” on page 92.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to Netezza

Note: This information is applicable only if you ordered SAS/ACCESS Interface to Netezza on SAS Viya).

If SAS/ACCESS Interface to Netezza was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Netezza without having to run the playbook again.

1. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   
   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

2. Add the following lines to each file:
   
   ```bash
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbc.ini-file-including-file-name
   export NZ_ODBC_INI_PATH=path-to-the-Netezza-configuration-files
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Netezza-client
   ```

3. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4. Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Netezza” on page 93.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.
Configure SAS/ACCESS Interface to ODBC

Note: This information is applicable only if you ordered SAS/ACCESS Interface to ODBC (on SAS Viya).

If SAS/ACCESS Interface to ODBC was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to ODBC without having to run the playbook again.

1. Using a text editor, open the odbc.ini file in your home directory in order to configure data sources.

   Some vendors of ODBC drivers might provide support for system administrators to maintain a centralized copy of the odbc.ini file via the environment variable ODBCINI. Refer to your ODBC driver's vendor documentation for more specific information.

   Add the location of the shared libraries to one of the system environment variables in order to enable the ODBC drivers to be loaded dynamically at run time. The ODBC drivers are ODBC API-compliant shared libraries, which are referred to as shared objects in UNIX.

2. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

3. Add the following lines to each file, using the content that is appropriate for the version of ODBC that you are using.
   For DataDirect:
   ```
   export ODBCHOME=ODBC-home-directory
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```
   For iODBC:
   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
   ```
   For unixODBC:
   ```
   export ODBC_SYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
   export ODBCINI=name-of-your-odbc.ini-file
   export ODBCINSTINI=name-of-your-odbcinst.ini-file
   export LD_LIBRARY_PATH=location-of-ODBC-driver-manager-library:$LD_LIBRARY_PATH
   ```
   Note: For unixODBC, if ODBC_SYSINI is not set in your environment, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

4. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

5. On each CAS node, use a text editor to edit the cas_usermods.settings file:
   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

6. Add the following lines, depending on the version of ODBC that you are using.
   For DataDirect:
   ```
   export ODBCHOME=ODBC-home-directory
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```
   For iODBC:
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINSTINI=location-of-your-odbcinst.ini-file-including-file-name
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library

For unixODBC:

export ODBCSYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
export ODBCINI=name-of-your-odbc.ini-file
export ODBCINSTINI=name-of-your-odbcinst.ini-file
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library

Note: For unixODBC, if ODBCSYSINI is not set in your environment, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

7 Save and close the cas_usermods.settings file.

8 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to ODBC” on page 93.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to Oracle

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Oracle (on SAS Viya).

If SAS/ACCESS Interface to Oracle was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Oracle without having to run the playbook again.

To manually configure the variables:

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2 Add the following lines to each file:

   export ORACLE_HOME=Oracle-home-directory
   export TWO_TASK=ORACLE_SID
   export ORAENV_ASK=NO
   export SASORA=V9
   export PATH=$PATH:$ORACLE_HOME/bin
   export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 On each CAS node, use a text editor to edit the cas_usermods.settings file:

   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings

5 Add the following lines:

   export ORACLE_HOME=Oracle-home-directory
   export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH

6 Save and close the cas_usermods.settings file.

7 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Oracle” on page 94.
Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to PostgreSQL

Note: This information is applicable only if you ordered SAS/ACCESS to PostgreSQL (on SAS Viya).

A file that contains information about the database connection is required. You have two options for providing connection information:

Note: Create the file in the /opt/sas/viya/home directory.

- Reference a Data Source Name (DSN).

Create an odbc.ini file. Here is an example of an odbc.ini file that supports DSN:

```
[postgresql_data_source_name]
Driver=/opt/sas/viya/home/lib64/psqlodbcw.so
ServerName=localhost or hostname or ip>
username=user name
password=password
database=
database
port=5432
```

- Specify connection information in your code.

Create and configure the odbcinst.ini file. Here is an example:

```
[ODBC Drivers]
PostgreSQL=Installed
[PostgreSQL]
Description=ODBC for PostgreSQL
Driver=/opt/sas/viya/home/lib64/psqlodbcw.so
```

Note: During installation, you should also have set the ODBCINI environment variable.

If SAS/ACCESS Interface to PostgreSQL was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to PostgreSQL without having to run the playbook again.

1. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2. Add the following lines to each file:

   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export PGCLIENTENCODING=encoding-for-the-PostgreSQL-client
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/sas/spre/home/lib64
   Export PATH=$PATH:/opt/sas/spre/home/lib64/psql
   ```

   For bulk loading, add the following line:

   ```
   export PATH=$PATH:/opt/sas/spre/home/lib64/psql
   ```

3. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4. On each CAS node, use a text editor to edit the cas_usermods.settings file:

   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

5. Add the following lines:

   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   ```

```
export PGCLIENTENCODING=UTF-8
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/sas/spre/home/lib64

6 Save and close the cas_usermods.settings file.

7 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to PostgreSQL” on page 95.
    Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to SAP ASE

Note: This information is applicable only if you ordered SAS/ACCESS Interface to SAP ASE (on SAS Viya).

If SAS/ACCESS Interface to SAP ASE was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to SAP ASE without being required to run the playbook.

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2 Add the following lines to each file:
   ```
   export SYBASE=path-to-the-SAP-ASE-Open-Client-Home
   export LD_LIBRARY_PATH= $LD_LIBRARY_PATH: $SYBASE/lib
   ```

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 Edit the vars.yml file to include the configuration settings that are described in “SAS/ACCESS Interface to SAP ASE” on page 95.
    Note: Adding the configuration settings to the vars.yml file ensures that they are maintained each time that the software is deployed.

Configure SAS/ACCESS Interface to SAP HANA

Note: This information is applicable only if you ordered SAS/ACCESS Interface to SAP HANA (on SAS Viya).

If SAS/ACCESS Interface to SAP HANA was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to SAP HANA without having to run the playbook again.

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2 Add the following lines to each file:
   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client
   ```

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 On each CAS node, use a text editor to edit the cas_usermods.settings file:
   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

5 Add the following lines:
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client

6 Save and close the cas_usermods.settings file.

7 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to SAP HANA” on page 96.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS/ACCESS Interface to SAP R/3

Note: This information is applicable only if you ordered SAS/ACCESS Interface to SAP R/3 (on SAS Viya).

If SAS/ACCESS Interface to SAP R/3 was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to SAP R/3 without having to run the playbook again.

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2 Add the following lines to each file:
   - export RFC_INI=path-to-the-SAP-R/3.ini-file
   - export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-SAP-R/3-client

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to SAP R/3” on page 96.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Additional required post-installation tasks are described in Post-Installation Instructions for SAS/ACCESS 9.4 Interface to R/3.

Configure SAS/ACCESS Interface to Snowflake

Note: This information is applicable only if you ordered SAS/ACCESS Interface to Snowflake (on SAS Viya).

If SAS/ACCESS Interface to Snowflake was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Snowflake without having to run the playbook again.

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2 Add the following lines to each file:
   - export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   - export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   - export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-odbc-driver-manager:location-of-your-Snowflake-odbc-driver

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 On each CAS node, use a text editor to edit the cas_usermods.settings file:
Chapter 5 / Post-installation Tasks

5 Add the following lines:

- `export ODBCINI=location-of-your-odbc.ini-file-including-file-name`
- `export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name`
- `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-odbc-driver-manager:location-of-your-Snowflake-odbc-driver`

6 Save and close the cas_usermods.settings file.

7 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Snowflake” on page 96.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

Configure SAS Data Connector to Spark

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Hadoop (on SAS Viya).

If SAS/ACCESS Interface to Spark was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Spark without having to run the playbook again.

To manually configure the variables:

1 For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

2 Add the following lines to each file:

   - `export JAVA_HOME=location-of-your-Java-8-JRE`
   - `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server`

   If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

3 Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4 On each CAS node, use a text editor to edit the cas_usermods.settings file:
   - `sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings`

5 Add the following lines:

   - `export JAVA_HOME=location-of-your-Java-8-JRE`
   - `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server`

   If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

6 Save and close the cas_usermods.settings file.

7 Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Spark and SAS In-Database Technologies for Spark” on page 97.

Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.
Configure SAS/ACCESS Interface to Teradata

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Teradata (on SAS Viya).

If SAS/ACCESS Interface to Teradata was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Teradata without having to run the playbook again.

To manually configure the variables:

1. Install all the packages from the Teradata Tools and Utilities (TTU) Base distribution on the SAS client node. Additionally, you will need to install TTU on the following CAS nodes:
   - For serial data access only, install TTU on all CAS controller nodes.
   - For multi-node data access, install TTU on all CAS controller and CAS worker nodes.
   - For parallel data access, install TTU on all CAS controller nodes.
     
     Note: Parallel data access requires SAS Data Connect Accelerator for Teradata.

2. If you are setting the encoding on the SAS client node to UTF-8, modify the clispb.dat file for Teradata. This should be done on each CAS node where TTU was installed in Step 1. If you are not using encoding, you can skip this step.

   Locate the clispb.dat file in the directory where TTU was installed (for example `/opt/teradata/client/16.20/etc/clispb.dat`). The path will vary depending on the TTU version and where it is installed.

   Ensure that the following two lines are in the clispb.dat file.

   ```
   charset_type=N
   charset_id=UTF8
   ```

   SAS recommends keeping the clispb.dat file in the default location. Relocating the file can cause issues.

3. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - `/opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh`
   - `/opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv`

4. Add the following lines to each file:

   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   ```

5. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

6. On each CAS node, use a text editor to edit the cas_usermods.settings file:

   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

7. Add the following lines:

   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
   ```

8. Save and close the cas_usermods.settings file.

9. Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Teradata and SAS In-Database Technologies for Teradata” on page 98.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.
Configure SAS/ACCESS Interface to Vertica

Note: This information is applicable only if you ordered SAS/ACCESS Interface to Vertica (on SAS Viya).

If SAS/ACCESS Interface to Vertica was not configured during deployment, perform the following steps to configure SAS/ACCESS Interface to Vertica without having to run the playbook again.

1. For each host that is specified in the [programming] host group, use a text editor to open the following files:
   - /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
   - /opt/sas/viya/config/etc/sysconfig/compsrv/default/sas-compsrv

2. Add the following lines to each file:
   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbc.ini-file-including-file-name
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Vertica-client
   export VERTICAINI=path-to-the-Vertica-configuration-files
   ```

   If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

3. Save and close the workspaceserver_usermods.sh and sas-compsrv files.

4. On each CAS node, use a text editor to edit the cas_usermods.settings file:
   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   ```

5. Add the following lines:
   ```
   export ODBCINI=location-of-your-odbc.ini-file-including-file-name
   export ODBCINST=location-of-your-odbc.ini-file-including-file-name
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Vertica-client
   export VERTICAINI=path-to-the-Vertica-configuration-files
   ```

6. Save and close the cas_usermods.settings file.

7. Edit the vars.yml file to include the configuration settings. For information, see “SAS/ACCESS Interface to Vertica” on page 98.

   Note: The addition of the configuration settings to the vars.yml file ensures that they persist each time the software is deployed.

(Optional) Configure Java for the ACCELWHERE Option

Important: If you want to configure your Java installation to improve Hadoop cluster processing speed, perform these tasks.

The ACCELWHERE option that is available with SAS Scalable Performance Data Engine leverages MapReduce when interacting with HDFS. It improves processing speed by optimizing data subsetting that occurs in the Hadoop cluster. For more information about the ACCELWHERE option, see WHERE Processing Optimization with MapReduce in SAS 9.4 and SAS Viya 3.4 Programming Documentation SAS SPD Engine: Storing Data in the Hadoop Distributed File System.

To enable the ACCELWHERE option for your Java installation:

1. Go to the location where Java is installed. If you are unsure of the location, the value for the JAVA_HOME environment variable is the fully qualified pathname of the top level of the Java directory tree.
2 Determine whether the `JAVA_HOME/lib/tools.jar` file exists. If it exists, then the ACCELWHERE option is already enabled and no further action is required. If it does not exist, continue with these steps to copy a `tools.jar` file to the appropriate location.

3 At the top level of the `JAVA_HOME` directory, create a subdirectory named `lib`.
   
   ```bash
   mkdir lib
   ```

4 Find out the version of Java that you are using. Most deployments of Java have a version number in the name of the directory that is used as `JAVA_HOME`. For example, if the directory is named `jre1.8.0_144`, you are using Java version 1.8.0_144. Similarly, if the directory is named `jdk1.8.0_161`, you are using Java version 1.8.0_161. If the directory name does not contain a version number, you should consult with your system administrator to find out the version of Java that is installed.

5 Locate a copy of the `tools.jar` file from the version of the JDK that matches the version of Java that is installed. Here are two methods for locating a copy of the `tools.jar` file:
   - Download the correct version of the JDK from the Oracle web site and unpack the downloaded file to find the `tools.jar` file.
   - Look in an existing Java installation for the correct version of the JDK.

6 Copy the located `tools.jar` file to the `lib` directory that you created.

7 Repeat these steps for each machine that runs SAS software.

---

**Configure Model Access**

**Configure Access to Models**

In order to import models into SAS Model Manager and register models from Model Studio, SAS Visual Analytics, and SAS Studio into the common model repository, as well as add a model from the common model repository into a decision flow, users must have the appropriate access permissions. For more information, see `Access to Models` in *SAS Viya Administration: Models*.

**Configure Access to Analytic Store Model Files**

In order to publish analytic store models from SAS Model Manager, Model Studio, SAS Decision Manager, and SAS Intelligent Decisioning to the SAS Micro Analytic Service destination, see `Analytic Store Model Files` in *SAS Viya Administration: Models*.

**Configure SAS Analytics for IoT**

**Offerings Included with SAS Analytics for IoT**

Some offerings included with SAS Analytics for IoT, SAS Decision Manager, SAS Event Stream Processing, and SAS Model Manager, require configuration tasks.

- To configure SAS Decision Manager, see “Configure SAS Decision Manager” on page 149.
- To configure SAS Event Stream Processing, see “Configure SAS Event Stream Processing” on page 149.
- To configure SAS Model Manager, see “Configure SAS Model Manager” on page 155.
Configure SAS Data Preparation and SAS Data Agent

SAS Data Agent Deployment Scenarios

Here are the scenarios for deploying SAS Data Agent and SAS Data Preparation:

- If your SAS Data Preparation deployment is in the cloud and your SAS Data Agent is on-premises, you have separate orders for SAS Data Preparation and SAS Data Agent. To deploy SAS Data Agent, see SAS Data Agent for Linux: Deployment Guide.

- If both your SAS Data Preparation deployment and SAS Data Agent deployment are on-premises, you have one order for both products. Deploy SAS Data Agent and SAS Data Preparation on separate machines. To deploy SAS Data Agent, see SAS Data Agent for Linux: Deployment Guide.

**CAUTION!** Do not run the on-premises deployment of SAS Data Agent on any machine that contains a previous SAS Viya deployment.

Configure SAS Data Quality

Configure the Quality Knowledge Base

**Note:** This information is applicable only if your order contains SAS Data Quality.


Your deployment includes the latest version of the QKB for Contact Information. If your deployment is a new deployment, this QKB is imported onto your CAS server and SAS Data Quality is configured to use this QKB by default. If your deployment is an upgrade from a previous version of SAS Viya, you have a new version of the QKB for Contact Information available with your upgraded deployment. However, SAS Data Quality is still configured to use the QKB you used as a default prior to your upgrade deployment. If you wish to configure SAS Data Quality to use the new version of the QKB for Contact Information by default, you must first import the new QKB onto your CAS server. For instructions on importing your QKB, see Import a QKB in SAS Data Quality 3.4: QKB Management for more information.

After successfully importing the new QKB onto your CAS server, you can configure SAS Data Quality to use it as your default QKB. Refer to Set the Default QKB and the Default Locale in SAS Data Quality 3.4: QKB Management for more information.

You can also configure SAS Data Quality to use a different QKB, such as one you have customized for use with other SAS Data Quality products. Refer to Set the Default QKB and the Default Locale in SAS Data Quality 3.4: QKB Management for more information.

SAS Data Quality also requires a QKB locale setting. After you deploy your CAS server, it is configured to use the English, United States locale, by default. If you use SAS Data Quality DATA step functions in CAS, or if you invoke Data Quality transformations in a job in the SAS Data Preparation application suite, you will use the English, United States locale setting, by default.

You can use the QARC command-line utility to archives and compress SAS Quality Knowledge Bases into the QARC format so they can be imported and used with SAS Cloud Analytic Services. For more information on a
Configure SAS Data Science

Software Order Associated with SAS Data Science

One other software order is associated with deploying SAS Data Science. The software order contains SAS Embedded Process for Hadoop and SAS Embedded Process for Teradata on Linux. The Software Order Email (SOE) specifies the location of the deployment documentation: SAS Embedded Process: Deployment Guide. You should deploy this order only if you are using SAS in-Database Technologies for Hadoop or SAS in-Database Technologies for Teradata.

Note: The SOE for the associated order does not refer to SAS Data Science explicitly.

Configure SAS Decision Manager

Configure Access to Analytic Store Model Files

For SAS Decision Manager, in order to publish decisions that use analytic store models to the SAS Micro Analytic Service destination, see Access to Analytic Store Model Files in SAS Viya Administration: Models.

Configure SAS Event Stream Processing

Complete SAS Event Stream Processing and SAS Analytics for IoT Setup

If your order included SAS Event Stream Processing or SAS Analytics for IoT, take a few steps to complete the deployment. If you did not purchase SAS Event Stream Processing, or if you do not plan to use the streaming analytics features of SAS Analytics for IoT, you can skip this section.

Set Environment Variables for SAS Event Stream Processing and SAS Analytics for IoT

If your order included SAS Event Stream Processing, you must set several environment variables to enable it to run. You should also set these variables if you plan to use the streaming analytics features of SAS Analytics for IoT.

At a shell that is used to invoke only SAS Event Stream Processing, export the three environment variables:

```bash
export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/version
export LD_LIBRARY_PATH=$DFESP_HOME/lib:/opt/sas/viya/home/SASFoundation/sasexe
export PATH=$PATH:$DFESP_HOME/bin
```

For version, specify the appropriate version of the SAS Event Stream Processing software, such as 5.2 or 6.1.

If you need to maintain your LD_LIBRARY_PATH setting for another SAS product, use the following command instead:
SAS Event Stream Processing includes the internal component SAS Micro Analytic Service. To use the Anaconda Python support in SAS Micro Analytic Service, you need to set additional variables for your version of Python. For instructions, see SAS Micro Analytic Service: Programming and Administration Guide, which is available on the SAS Event Stream Processing product page.

Depending on the shell environment that you use, you can also add these export commands to your .bashrc file or .profile file to update the settings automatically. Another option is to create a configuration shell script and copy it to your /etc/profile.d directory.

Enable Metering for ESP Servers

If your order included SAS Event Stream Processing, you must take additional steps to enable the product license.

Note: These steps are not required for SAS Analytics for IoT, which uses a different license.

The playbook applies the product license on each machine where you have deployed SAS Event Stream Processing. However, you must set up and run at least one metering server to track the number of incoming events and to maintain event counts on your ESP servers.

The metering server aggregates counts that are based on the license, the source window, and the hour of day. It stores aggregated results so that a client can query and track the total volume of messages that are processed. Enabling the metering server ensures that your ESP server is in compliance with the terms of its license. Event metering is not required on development servers because they do not contribute to the event volume that is assigned to a license.

Run the following command to start the metering server using its default port (31001):

```
dfesp_metering -d
```

The -d argument creates a log file in the configuration directory.

For more information about the metering server, see Setting Up and Using the Metering Server in the SAS Event Stream Processing user documentation.

Start the ESP Server

When the playbook has completed, the SAS Event Stream Processing Studio, Streamviewer, and SAS Event Stream Manager processes are already running. Before you can open or create a model in SAS Event Stream Processing Studio, you must start the ESP server.

If you have deployed SAS Event Stream Manager, you can take some additional steps to set up a connection between the ESP server and SAS Event Stream Manager. For more information, see “Configure the ESP Server for SAS Event Stream Manager” on page 153.

To start an ESP server:

1. Change directories:

```
cd /opt/sas/viya/home/SASEventStreamProcessingEngine/version/bin
```

   For version, specify the version of the SAS Event Stream Processing software.

2. Start the ESP server. Here is an example of the command:

```
./dfesp_xml_server -pubsub n -http port &
```

   The ampersand (&) enables additional commands to be entered in the same window that started the server. Other values that provide server start-up instructions are defined in the esp-properties.yml configuration file. For more information, see Server Configuration Properties.

   You also have the option to configure the ESP server to start and run as a system daemon. You can then use init.d scripts to start and stop it. For more information, see Setting Up the ESP Server as a Daemon.
Start SAS Event Stream Processing Studio

Additional steps might be required to use SAS Event Stream Processing Studio, which provides a user interface for creating models. It is automatically started during the installation. However, you can start it manually if you find that the service is not running.

1. SAS Event Stream Processing Studio requires Java 1.8. If Java 1.8 is not the default version of Java on your system, update the following script to set the $SAS_JAVA_HOME$ environment variable:

   ```bash
   /opt/sas/viya/config/etc/sysconfig/sas-javaesntl/sas-java
   ```

   Here is an example:

   ```bash
   $SAS_JAVA_HOME=/usr/java/jdk1.8.0_101/jre
   ```

   Or supply the location of the JDK, if applicable. Here is an example:

   ```bash
   $SAS_JAVA_HOME=/usr/java/jdk1.8.0_101
   ```

   **Note:** Do not include the `/bin/java` portion of the path for the definition of $SAS_JAVA_HOME$.

2. (Optional) If you plan to import models from SAS Model Manager, increase the default memory allocation for the Java Virtual Machine (JVM). Configuring the JVM is a post-deployment task. For more information, see Configuration Properties: Java Virtual Machine (JVM) in the SAS Viya Administration documentation.

3. Verify that you have set the required environment variables. For more information, see “Set Environment Variables for SAS Event Stream Processing and SAS Analytics for IoT” on page 149.

4. SAS Event Stream Processing Studio should be running when the playbook completes. Check the status of the espvm process:
   - For Red Hat Enterprise Linux 6.x:
     ```bash
     sudo service sas-viya-espvm-default status
     ```
   - For Red Hat Enterprise Linux 7.x or SUSE Linux:
     ```bash
     sudo systemctl status sas-viya-espvm-default
     ```

5. If the espvm service is reported to be down, to start it:
   - For Red Hat Enterprise Linux 6.x:
     ```bash
     sudo service sas-viya-espvm-default start
     ```
   - For Red Hat Enterprise Linux 7.x or SUSE Linux:
     ```bash
     sudo systemctl start sas-viya-espvm-default
     ```

6. When the service is running, you can access SAS Event Stream Processing Studio using a web browser on Windows or Linux. Use the following URL:

   ```bash
   https://reverse-proxy-server/SASEventStreamProcessingStudio
   ```

   For `reverse-proxy-server`, use the host name from the machine that you assigned to the [httpproxy] host group in the inventory file.

7. Before you can open or create a model in SAS Event Stream Processing Studio, you must start the ESP server. For instructions, see “Start the ESP Server” on page 150
Log on to Streamviewer

SAS Event Stream Processing Streamviewer is a web-based client that visualizes events that stream through event stream processing models. Streamviewer is installed automatically along with SAS Event Stream Processing.

When the deployment process has completed, take the following steps to access Streamviewer:

1. The Streamviewer process is started automatically by the playbook. To check the status of the Streamviewer process:
   - For Red Hat Enterprise Linux 6.x:
     ```
     sudo service sas-viya-espstrmvwr-default status
     ```
   - For Red Hat Enterprise Linux 7.x or SUSE Linux:
     ```
     sudo systemctl status sas-viya-espstrmvwr-default
     ```

2. If it is not running, start the process:
   - For Red Hat Enterprise Linux 6.x:
     ```
     sudo service sas-viya-espstrmvwr-default start
     ```
   - For Red Hat Enterprise Linux 7.x or SUSE Linux:
     ```
     sudo systemctl start sas-viya-espstrmvwr-default
     ```

3. Open the following URL:
   ```
   http://ESP-server-host-name/SASEventStreamProcessingStreamviewer
   ```
   For `ESP-server-host-name`, substitute the host name of the machine where Streamviewer is installed.

4. Enter your user ID and password and click Sign in.

When you successfully log on to Streamviewer, the home page appears.

Encryption and Authentication Options

SAS Event Stream Processing provides optional encryption and authentication features. You can enable encryption on TCP/IP connections within an event stream processing engine. You can also configure ESP servers to require client authentication for SAS TCP/IP clients.

To enable encryption, the OpenSSL libraries must be installed on all computer systems that run the ESP server and clients. Version 1.0.2 or later of the Transport Layer Security (TLS) Protocol is required in order to take advantage of ECDH support for encryption ciphers used in encrypted connections.

Authentication and encryption apply to the following ESP server APIs:

- The ESP Server (XML Server) HTTPS API
  - Connections that are created by a client to communicate with an ESP server
  - Connections that are created by a file and socket connector or adapter that acts as a socket client or server
  - Connections that are created by the Streamviewer component (streamviewer.html) to communicate with the ESP server using the HTTPS protocol

- C, Java, or Python Publish/Subscribe API
  - Connections that are created by a client that uses the C, Java, or Python Publish/Subscribe API to communicate with an ESP server
  - Connections that are created by an adapter to communicate with an ESP server
Configuration of these security options has been greatly simplified in recent releases of SAS Event Stream Processing. For more information about enabling security for an ESP server or for Streamviewer, see SAS Event Stream Processing: Security.

Configure Database Connectivity for SAS Event Stream Processing

Database connections are optional for SAS Event Stream Processing. If your order included SAS Event Stream Processing, you can enable database connectivity by performing some configuration tasks after the installation has completed. For a full discussion of database connections for SAS Event Stream Processing, see Using the Database Connector and Adapter.

Complete SAS Event Stream Manager Setup

If your order included SAS Event Stream Manager, you must complete setup after the installation has completed. Otherwise, skip this section.

(Optional) Modify JVM Settings for SAS Event Stream Manager

If you plan to import models from SAS Model Manager, you should increase the default memory allocation for the Java Virtual Machine (JVM).

Configuring the JVM is a post-deployment task. You can use SAS Environment Manager to change the default memory settings. SAS recommends that you initially change the setting to 512 MB. For more information, see Configuration Properties: Java Virtual Machine (JVM) in the SAS Viya Administration documentation.

Log On to SAS Event Stream Manager

If your order included SAS Event Stream Manager, it is installed in your environment by the playbook. SAS Event Stream Manager uses SAS Logon Manager for logon functionality. SAS Logon Manager requires LDAP for user authentication.

1. Open SAS Event Stream Manager from a URL with the following format:
   
   https://reverse-proxy-server/SASEventStreamManager

   For reverse-proxy-server, use the host name of the machine where you installed SAS Viya.

   The Sign In to SAS window is displayed.

2. Enter your user ID and password, and click Sign In.

Successful logon to the SAS Event Stream Manager user interface indicates that the software has been installed correctly.

Configure the ESP Server for SAS Event Stream Manager

Note: If you have not deployed SAS Event Stream Manager, you can skip this step.

To manage SAS Event Stream Processing instances with SAS Event Stream Manager, you must set up connections to ESP servers. You can use the SAS Event Stream Manager user interface to manually locate and define ESP servers that are running in your environment. However, you can also start your ESP servers with some additional instructions that enable secure, persistent sockets between SAS Event Stream Manager and ESP servers. SAS Event Stream Manager can then locate and manage ESP servers automatically.

To start an ESP server:

1. Provide SAS Logon Manager with a client ID and client secret for SAS Event Stream Manager. First, obtain the value of the SAS Configuration Server (Consul) token for your environment:

   Note: Specify the command on a single line. Multiple lines are used here for improved readability.
sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token

2 Run a curl command to request a registration token for a new client. In this example, the client is named app:

```bash
curl -X POST "http://localhost/SASLogon/oauth/clients/consul?callback=false&serviceId=app" -H "X-Consul-Token: X-Consul-Token-value"
```

For `X-Consul-Token-value`, substitute the value for the Consul token, which you obtained from the previous step.

Note: Specify the command on a single line. This request must pass a `callback=false` query string parameter and authenticate directly by passing a Consul token. If the Consul token that you specified in the command is valid, SAS Logon Manager returns the OAuth access token for registration in the response.

3 Use the registration token to register the client ID. This step establishes the ESP server as a new client of SAS Logon Manager. Run the following curl command:

```bash
  "client_id": "client-id",
  "client_secret": "client-secret",
  "scope": ["openid", "+"],
  "resource_ids": "none",
  "authorities": ["uaa.none"],
  "authorized_grant_types": ["password"]
}
```

Note: You can find more information about the required steps to configure a new client for SAS Logon Manager in Obtain an Access Token Using Password Credentials in SAS Viya Administration: Authentication.

4 Create an XML file with filename esm.xml. Make sure that it uses the required syntax.

Here is an example:

```xml
<esm>
  <server name="SAS-Event-Stream-Manager-host">
    <url>http://fully-qualified-host-name</url>
    <auth>
      <clientId>client-ID</clientId>
      <clientSecret>client-secret</clientSecret>
      <user>user-name</user>
      <password>password</password>
    </auth>
  </server>
</esm>
```

a For `SAS-Event-Stream-Manager-host`, substitute the host name of the machine where SAS Event Stream Manager is running.

b For `fully-qualified-host-name`, substitute the fully qualified domain name of the machine where the SAS Event Stream Manager server is running.

c For `client-ID`, substitute the client ID that you provided to SAS Logon Manager for the SAS Event Stream Manager instance.

d For `client-secret`, substitute the client secret that you provided to SAS Logon Manager for SAS Event Stream Manager.
For user-name, substitute a user name for an LDAP user account that is valid for use with SAS Logon Manager.

For password, substitute the password that corresponds to the user account that you specified.

Repeat the <server></server> section of the file as many times as required to accommodate all SAS Event Stream Manager servers.

5. Save the file in a network-accessible directory.

6. Change directories:
   
   cd /opt/sas/viya/home/SASEventStreamProcessingEngine/version/bin

   For version, specify the version of the SAS Event Stream Processing software.

7. Start the ESP server. Here is an example of the command:
   
   ./dfesp_xml_server -esm file://full-path-to-file/esm.xml -pubsub n -http port &

   The -esm file://esm.xml argument instructs the ESP server to use the values in the file that you created previously. Other values that provide server start-up instructions are defined in the esp-properties.yml configuration file. For more information, see Server Configuration Properties.

   When it is started with the optional -esm file://esm.xml argument, the ESP server automatically registers with SAS Event Stream Manager, which can then manage it. The esm.xml file instructs the ESP server where to locate SAS Event Stream Manager. The ESP server registers itself with SAS Logon Manager as a new client with a new secret. SAS Logon Manager can then provide a token that enables the ESP server to set up a persistent web socket for secure communications with SAS Event Stream Manager.

**Configure SAS Intelligent Decisioning**

**Configure Analytic Store File System Paths**

**Note:** In May 2019, SAS Decision Manager was renamed SAS Intelligent Decisioning.

For SAS Intelligent Decisioning, in order to publish decisions that use analytic store models to the SAS Micro Analytic Service destination, see Access to Analytic Store Model Files in SAS Viya Administration: Models.

**Configure SAS Model Manager**

**Configure the Workflow Client User**

A user account must be configured for a workflow client in order for it to make service task calls.

To configure this user:

1. In SAS Environment Manager, select the $ from the side menu to open the Configuration page.

2. On the Configuration page, select All Services from the View menu, and then select Workflow service from the list of services.

3. Enter sas.workflow.client.sasmodelmanager in the Filter field.

   a. If sas.workflow.client.sasmodelmanager does not exist, click New Configuration to create this item.
If `sas.workflow.client.sasmodelmanager` still does not exist, stop and start the service with the following commands:

```
sudo /etc/init.d/sas-viya-modelmanagement-default stop
sudo /etc/init.d/sas-viya-modelmanagement-default start
```

It might take a few minutes for the service to start again. After the service has started, click 🔄 to see the property listed.

4 Enter the ID of an authorized user in the `defaultServiceAccount` field. This user ID will be used to execute all service tasks for any workflow process that has been assigned the client identifier `SAS Model Manager`. For more information about the required user account, see "User Accounts (Reference)" on page 49.

For more information about client identifiers, see Specify a Client Identifier in SAS Workflow Manager: User’s Guide.

5 Click Save.

6 Log in to SAS Viya using the user you entered in the `defaultServiceAccount` field.

**Configure Access to Analytic Store Model Files**

For SAS Model Manager, in order to publish analytic store models to the SAS Micro Analytic Service destination, see Access to Analytic Store Model Files in the SAS Viya Administration: Models.
Verify SAS/ACCESS Interface to SAP R/3 ............................. 173
Verify SAS/ACCESS Interface to Snowflake .............................. 174
Verify SAS/ACCESS Interface to Teradata .................................. 174
Verify SAS/ACCESS Interface to Vertica .................................. 175

Verify Access to SAS Logon and SAS Drive

Note: The following information applies only if you are performing a full deployment. If you are performing a programming-only deployment, skip this section.

To verify that you can access SAS Logon and SAS Drive:

1 To verify SAS Logon, open it using a URL with this format:
   http://reverse-proxy-server/SASLogon
2 To verify SAS Drive, open it using a URL with this format:
   http://reverse-proxy-server/SASDrive

For more information about SAS Drive, see Introduction to SAS Drive in SAS Drive.

Log On to Your Version of SAS Studio

The version of SAS Studio that you are using depends on which type of deployment you performed:

- If you deployed a programming-only environment, then your environment contains SAS Studio 4.4.
- If you deployed a full environment, then your environment contains both SAS Studio 4.4 and SAS Studio 5.1.
   By default, you will log on to SAS Studio 5.1 to perform the deployment tasks.

   Note: To use certain features for administration, you might want to use SAS Studio 4.4. For guidance on which release to use for ongoing administration, see Choosing between SAS Studio 5.1 and 4.4 in SAS Viya Administration: Configuration Properties.

To ensure that your default version of SAS Studio has been deployed correctly and is working, log on to it:

1 Open SAS Studio from a URL with this format:
   - For SAS Studio 4.4:
     http://reverse-proxy-server/SASstudio
   - For SAS Studio 5.1:
     http://reverse-proxy-server/SASStudioV

   Use the host name of the machine that you assigned to the [httpproxy] host group in the inventory file. For more information about assigning machines, see “Assign the Target Machines to Host Groups” on page 77.

   Make a note of this URL to share with any other users of your SAS Viya software, as described in “Share Important Deployment Information with the Administrators” on page 177.

2 Log on using the credentials for your operating system account.

Note: To log off from SAS Studio, click Sign Out on the toolbar. Do not use the Back button on your web browser.
SAS Viya File Locations

After the deployment process has completed, default directories will be populated with installation and configuration files. The following table lists default directory and file locations that you can verify as part of deployment validation. The table also describes recommended locations for installation files to assist you in locating them later:

Table 6.1  Default Directory and File Locations

<table>
<thead>
<tr>
<th>Directory Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/opt/sas</td>
<td>SAS Viya root path.</td>
</tr>
<tr>
<td>/opt/sas/viya/home</td>
<td>SAS Viya home directory.</td>
</tr>
<tr>
<td>/opt/sas/spre/home</td>
<td>Home directory for the SAS Foundation programming environment and files to support SAS 9.4 integration.</td>
</tr>
<tr>
<td>/opt/sas/tenant-ID</td>
<td>Location of tenant-specific subdirectories in a multi-tenant deployment. Tenants cannot access files in other tenant ID directory paths.</td>
</tr>
<tr>
<td>$USERHOME/sas_repos</td>
<td>Default location of the SAS software repositories that are created and populated, based on your order, when you create a mirror repository. $USERHOME corresponds to the installation user’s home directory.</td>
</tr>
<tr>
<td>/opt/sas/viya/config</td>
<td>Location where configuration files are stored after the deployment process has completed.</td>
</tr>
<tr>
<td>/opt/sas/viya/config/data/cas/instance-name/name-of-Public-caslib</td>
<td>Default location for persistent storage for the predefined system caslibs and the Public caslib.</td>
</tr>
<tr>
<td>/sas/install/sas_viya_playbook</td>
<td>Recommended location for all files that are required for the deployment. Individual files that provide configuration options are stored in subdirectories of /sas_viya_playbook. For more information, see &quot;Files Used for Deployment&quot; on page 4.</td>
</tr>
<tr>
<td>/sas/install/sas_viya_playbook/deployment.log</td>
<td>Default location for Ansible deployment logs.</td>
</tr>
<tr>
<td>/opt/sas/viya/config/var/log</td>
<td>Top-level directory for application logs.</td>
</tr>
<tr>
<td>/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts</td>
<td>Default location of SAS-provided TLS certificates to enable encryption of data in motion.</td>
</tr>
<tr>
<td>Directory Path</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>/opt/sas/viya/config/backup</td>
<td>Also referred to as “local vault.” The location where the backup files for data sources are created. The local vault location cannot be changed. As a last step in the backup process, the contents of the local vault are moved to the shared vault. The SAS Viya administrator sets the location of the sharedvault parameter in SAS Environment Manager as a post-deployment step. For more information, see Backup and Restore: Concepts in SAS Viya Administration.</td>
</tr>
<tr>
<td>/opt/sas/viya/config/var/log/deploymentBackup/default</td>
<td>Locations for backup activity logs.</td>
</tr>
<tr>
<td>/opt/sas/viya/config/var/log/backup-agent/default</td>
<td>Location of SAS Viya init scripts, which are used to start daemons and long-running processes.</td>
</tr>
<tr>
<td>/etc/init.d/</td>
<td>Location of SAS Viya init scripts, which are used to start daemons and long-running processes.</td>
</tr>
<tr>
<td>/tmp</td>
<td>Default location for the CAS disk cache. This location is not recommended for use in a production environment. Change the corresponding variable in vars.yml during playbook configuration. For more information, see “Set Up the CAS Cache Directory” on page 100.</td>
</tr>
</tbody>
</table>

**Perform Installation Qualification on RPM Packages**

Some of your SAS software is collected in RPM (Red Hat Package Manager) packages.

1. To qualify the installation of your RPM packages, run the basic RPM commands:

   ```
   rpm -Vv package-name
   ```

   **Note:** Using the `-Vv` option provides a status for all the files in the package. If you want a list of the failures only, use the `-V` option.

   For example, to verify the contents of the `sas-certframe` package, use the following command:

   ```
   sudo rpm -Vv sas-certframe
   ```

   The output is as follows:

   ```
   ........ /opt/sas/viya/home/SASFoundation
   ........ /opt/sas/viya/home/SASFoundation/sasexe
   ........ /opt/sas/viya/home/SASFoundation/sasexe/fsnetiom.so
   ........ /opt/sas/viya/home/SASFoundation/utilities
   ........ /opt/sas/viya/home/SASFoundation/utilities/bin
   ........ /opt/sas/viya/home/SASFoundation/utilities/bin/tktsql
   ```

2. To verify SAS Event Stream Processing deployment, run the following command to obtain a list of the relevant RPM packages that are deployed on your system:

   ```
   rpm -qa sas-esp*
   ```

   The output is as follows:
Create a for loop command for verifying multiple packages that share a common naming convention. For example, to verify all packages whose names begin with `sas-`, use the following query:

```bash
for i in $(rpm -qg "SAS");do sudo rpm -Vv $i;done
```

A successful verification shows the list of files that make up the RPM but with no error indicators, as follows:

```
# rpm -Vv sas-certframe
........... /opt/sas/viya/home/lib/sas-certframe/sas-init-functions
#
```

An unsuccessful verification provides error indicators next to the filename. Here is an example:

```
# rpm -Vv sas-certframe
S.5....T. /opt/sas/viya/home/lib/sas-certframe/sas-init-functions
#
```

The error indicators are shown in the following format:

```
SM5DLUGT c
```

In addition, if a file is missing, the error message contains the phrase “missing”:

```
missing  /opt/sas/viya/home/lib/sas-certframe/sas-init-functions
```

The meaning of each error indicator is described as follows:

- **S** File size. RPM keeps track of file sizes. A difference of even one byte triggers a verification error.

- **M** File mode. The permissions mode is a set of bits that specifies access for the file's owner, group members, and others. Even more important are two additional bits that determine whether a user's group or user ID should be changed if they execute the program that is contained in the file. Since these bits permit any user to become root for the duration of the program, you must be cautious with a file's permissions.

- **D** Major and minor numbers. Device character and block files contain a major number. The major number is used to communicate information to the device driver that is associated with the special file. For example, under Linux, the special files for SCSI disk drives should have a major number of 8, and the major number for an IDE disk drive's special file should be 3. Any change to a file's major number could produce disastrous effects. RPM tracks such changes. A file's minor number is similar to the major number, but conveys different information to the device driver. For disk drives, this information can consist of a unit identifier.

- **L** Symbolic link. If a file is a symbolic link, RPM checks the text string that contains the name of the symbolically linked file.

- **U** File owner. Most operating systems keep track of each file's creator, primarily for resource accounting. Linux and UNIX also use file ownership to help determine access rights to the file. In
addition, some files, when executed by a user, can temporarily change the user's ID, normally to a more privileged ID. Therefore, any change of file ownership might have significant effects on data security and system availability.

G File group. Similar to file ownership, a group specification is attached to each file. Primarily used for determining access rights, a file's group specification can also become a user's group ID if that user executes the file's contents. Therefore, any changes in a file's group specification are important and should be monitored.

T Modification time. Most operating systems keep track of the date and time that a file was last modified. RPM keeps modification times in its database.

c Configuration file. This is useful for quickly identifying configuration files, since they are likely to change and therefore are unlikely to verify successfully. You could also get a d in this slot, indicating that the file is for documentation, which is also likely to change often.

Verification failures are expected for files that contain frequently changing content, such as environment-specific Java paths, newly generated TLS certificates, SAS license information, and CAS customizations. Such verification failures for these types of files usually do not indicate any errors in the files.

Note: In SAS Viya 3.4, the following files are renamed during the deployment process. If you perform a verification and receive "missing" indications for the following files, they can be safely ignored. The following are the default pathnames.

- /opt/sas/viya/config/etc/evmcltsvcs/sas-ops-agent-update.sh
- /opt/sas/viya/config/etc/evmsvrops/sas-ops-agentsrv-update.sh

Access CAS Server Monitor

Note: This section is applicable only if you have a programming-only deployment. If you have a full deployment, skip this section.

To verify that CAS Server Monitor has been successfully deployed, access it by opening a web browser and entering the URL in the address field in the following format:

- For single tenant:
  
  http://reverse-proxy-server/cas-shared-default-http

  Here is an example:

  http://host1.sas.com/cas-shared-default-http

- For multi-tenant:
  
  http://tenant.reverse-proxy-server/cas-tenant-instance-http

  Here is an example:

  http://acme.host1.sas.com/cas-acme-default-http

Log on as a SAS Administrator user.

If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the "Your connection is not private" message. SAS recommends replacing the certificates before giving end-users access to SAS Viya. For information, see “Transport Layer Security” on page 41.
Access SAS Environment Manager

Note: This section is applicable only if you have a full deployment. If you have a programming-only deployment, skip this section.

1. Go to the machine you assigned to the [AdminServices] host group.

2. Open SAS Environment Manager from a URL with the following format:

   https://reverse-proxy-server/SASEnvironmentManager

   Note: If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the 'Your connection is not private' message. SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see "Transport Layer Security" on page 41.

3. Sign on as one of the SAS Administrators that you set up in "Set Up Administrative Users" on page 123.

Verify SAS Message Broker

Note: This section is applicable only if you have a full deployment. If you have a programming-only deployment, skip this section.

1. To verify that SAS Message Broker has been deployed correctly, go to the machine that you assigned to the [rabbitmq] host group.

2. Open a browser and go to the following address:

   - If HTTPS is enabled:
     https://reverse-proxy-server:15672/#/

     Note: If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the 'Your connection is not private' message. SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see Access SAS Message Broker via HTTPS and HTTP (Linux Full Deployment) for full deployments in Encryption in SAS Viya: Data in Motion. If you have a programming-only deployment, see Access SAS Studio via HTTPS in the same document.

   - If HTTP is enabled:
     http://reverse-proxy-server:15672/#/

     If the RabbitMQ logon window appears, then SAS Message Broker is functioning as expected.

Verify SAS Infrastructure Data Server

Note: This section is applicable only if you have a full deployment. If you have a programming-only deployment, skip this section.

Use these steps to verify that SAS Infrastructure Data Server has been deployed correctly.
1 On the machine that you assigned to the [pgpoolc] host group, to check status:

- On Red Hat Enterprise Linux 6.x and Linux 7.x:
  
  ```
  sudo service sas-viya-sasdatasvrc-postgres status
  ```

- For SUSE Linux:
  
  ```
  sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status
  ```

2 If SAS Infrastructure Data Server is running appropriately, you should receive a response like this:

PGPool is running with PID=11445

Checking Postgresql nodes status...

<table>
<thead>
<tr>
<th>node_id</th>
<th>hostname</th>
<th>port</th>
<th>status</th>
<th>lb_weight</th>
<th>role</th>
<th>select_cnt</th>
<th>load_balance_node</th>
<th>replication_delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>machine1</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>primary</td>
<td>1</td>
<td>true</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>machine2</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>standby</td>
<td>0</td>
<td>false</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>machine3</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>standby</td>
<td>0</td>
<td>false</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>machine4</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>standby</td>
<td>0</td>
<td>false</td>
<td>0</td>
</tr>
</tbody>
</table>

(4 rows)

A status of `up` for a node indicates the node is running.

---

### Verify Elasticsearch

**Note:** If you are not deploying SAS Visual Investigator or SAS Investigation and Intelligence Management, skip this section.

1 At any machine in the Elasticsearch cluster, enter each of the following commands as the sas user to set the location of the security files:

**Note:** Machines in the Elasticsearch cluster are listed in the [elasticsearch] host group in your inventory.ini file. For more information about the inventory.ini file, see “Edit the Inventory File” on page 74.

```bash
vault_token=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/searchguard/default/vault.token
cert_file=/opt/sas/viya/config/etc/elasticsearch/default/certs/searchguard/sgadmincert.pem
ca_cert=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem
key_file=/opt/sas/viya/config/etc/elasticsearch/default/keys/searchguard/sgadminkey.pem
```

2 Obtain the vault address.

**Note:** Enter the second command on a single line. Multiple lines are used here to improve readability.

```bash
source /opt/sas/viya/config/consul.conf
curl --cacert ${ca_cert} --key ${key_file} --cert ${cert_file} https://IP-address-for-Elasticsearch-master-node:9200/_cluster/health?pretty=true
```
Here is typical output for the command:

```json
{
    "cluster_name" : "testcluster",
    "status" : "green",
    "timed_out" : false,
    "number_of_nodes" : 2,
    "number_of_data_nodes" : 3,
    ...
}
```

If the value of status is **green**, the cluster is fully functional. For additional information about Elasticsearch cluster health, refer to [https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html).

**Note:** For deployments that have only one data node, the status value should be **yellow** to indicate that the cluster is functional.

---

### Verify SAS Event Stream Manager Status

To verify that a deployment of SAS Event Stream Manager has completed successfully, check that the required SAS services are available. You can check the status of all the SAS Event Stream Manager services by running the following commands on Red Hat Enterprise Linux 6.x:

- `sudo service sas-viya-esm-service-default status`
- `sudo service sas-viya-esm-webui-default status`

Run the following commands on Red Hat Enterprise Linux 7.x or SUSE Linux:

- `sudo systemctl status sas-viya-esm-service-default`
- `sudo systemctl status sas-viya-esm-webui-default`

Here is typical command output from Red Hat Enterprise Linux 6.7 to indicate that the software is running normally:

```
sas-viya-esm-service-default is running
sas-viya-esm-webui-default is running
```

The output is different on Linux 7.x or SUSE Linux, but it reports that each service is running.

---

### Overview of Data Access Verification

**Overview**

Your version of SAS Studio depends on which type of deployment you performed:

- If you deployed a programming-only environment, then your environment contains SAS Studio 4.4.
- If you deployed a full environment, then your environment contains both SAS Studio 4.4 and SAS Studio 5.1.

By default, you will log into SAS Studio 5.1.

After starting a CAS session, run the SAS code as specified in the verification section for your data connector(s). If any of the verification steps for data access return an error, perform the appropriate configuration steps again.
Verify with SAS Studio 4.4

To verify the SAS Data Connectors:

1. Open SAS Studio from a URL with this format:
   http://hostname/SASStudio

2. Log on using the credentials for your operating system account.

3. Start a new CAS session.
   a. In the navigation pane, open the Snippets section.
   c. Right-click Create CAS Connection and select Open. The snippet opens in the code editor.
      In the code, specify the cashost and the casport. (Default is casport of 5570).
   d. In the toolbar, click \( \text{Run} \) to run the new CAS session code.
   e. Right-click New CAS Session and select Open. The snippet opens in the code editor.
   f. In the toolbar, click \( \text{Run} \) to run the new CAS session code.

Run the SAS code as specified in the verification section for your data connector(s). If any of the verification steps for data access return an error, perform the appropriate configuration steps again.

Verify with SAS Studio 5.1

1. Open SAS Studio from a URL with this format:
   http://hostname/SASStudioV

2. Log on using the credentials for your operating system account.

3. Start a new CAS session.
   a. In the left navigation pane, click on the Snippets icon.
   c. Right-click New CAS Session and select Open. The snippet opens in the code editor.
   d. In the toolbar, click \( \text{Run} \) to run the new CAS session code.

Run the SAS code as specified in the verification section for your data connector(s). If any of the verification steps for data access return an error, perform the appropriate configuration steps again.

Verify SAS/ACCESS Interface to Amazon Redshift

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Amazon Redshift (on SAS Viya).
To verify that SAS/ACCESS Interface to Amazon Redshift was successfully deployed:

1 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Amazon Redshift LIBNAME statement:

   libname arslib redshift server="Redshift-host-name" database="Redshift-database-name" user="user-ID" password="user-password";

   If SAS/ACCESS to Amazon Redshift was successfully deployed, the execution of the LIBNAME statement returns results without error.

2 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Amazon Redshift:

   caslib rslib datasource=(srctype="redshift", username="user-ID", password="password", server="Redshift-host-name", database="Redshift-database-name");

   proc casutil;
   list files incaslib="rslib";
   run;

   If the data connector was successfully deployed, the results are the names of the tables in Amazon Redshift.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

---

**Verify SAS/ACCESS Interface to DB2**

**Note:** The information in this section is applicable only if you ordered SAS/ACCESS Interface to DB2 (on SAS Viya).

To verify that SAS/ACCESS Interface to DB2 was successfully deployed:

1 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to DB2 LIBNAME statement:

   libname db2lib db2 database="DB2-database-name" user="user-ID" password="user-password";

   If SAS/ACCESS to DB2 was successfully deployed, the execution of the LIBNAME statement returns results without error.

2 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to DB2:

   caslib db2clib datasource=(srctype="db2", username="user-ID", password="password", database="DB2-database-name");

   proc casutil;
   list files incaslib="db2clib";
   run;

   If the data connector was successfully deployed, the results are the names of the tables in DB2.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.
Verify SAS/ACCESS Interface to Google BigQuery

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Google BigQuery (on SAS Viya).

1 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Google BigQuery LIBNAME statement:

   libname gbqlib bigquery project="Google-cloud-project" schema="dataset-ID"
   credfile="location-of-Google-application-credentials-file";

   If SAS/ACCESS to Google BigQuery was successfully deployed, the execution of the LIBNAME statement returns results without error.

2 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Google BigQuery:

   caslib gbqCLib datasource=(srctype="bigquery", project="Google-cloud-project", schema="dataset-ID",
   credfile="location-of-Google-application-credentials-file");

   proc casutil; list files incaslib="gbqCLib"; run;

   If the data connector was successfully deployed, the results are the names of the tables in Google BigQuery.

Verify SAS/ACCESS Interface to Greenplum

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Greenplum (on SAS Viya).

From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Greenplum LIBNAME statement:

   libname glib greenplum server="greenplum-host-name" database="greenplum-database-name" user="user-ID"
   password="user-password";

   If SAS/ACCESS to Greenplum was successfully deployed, the execution of the LIBNAME statement returns results without error.

   If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to HAWQ

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to HAWQ (on SAS Viya).

From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to HAWQ LIBNAME statement:

   libname hawqlib hawq server="hawq-host-name" database="hawq-database-name" user="user-ID"
   password="user-password";
If SAS/ACCESS to HAWQ was successfully deployed, the execution of the LIBNAME statement returns results without error.

If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

**Verify SAS/ACCESS Interface to Impala**

*Note:* The information in this section is applicable only if you ordered SAS/ACCESS Interface to Impala (on SAS Viya).

To verify that SAS/ACCESS Interface to Impala was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Impala LIBNAME statement:

   ```sas
   libname imp impala server="Impala-host-name" database="Impala-database-or-schema-name" user="user-ID" password="user-password";
   
   If SAS/ACCESS to Impala was successfully deployed, the execution of the LIBNAME statement returns results without error.
   
   2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Impala:

      ```sas
      caslib implib datasource=(srctype="impala", username="user-ID", password="user-password", server="Impala-host-name", database="Impala-database-or-schema-name");
      
      proc casutil;
      list files incaslib="implib";
      run;
      
      If the data connector was successfully deployed, the results are the names of the tables in Impala.
      
      If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.
      
**Verify SAS/ACCESS Interface to JDBC**

*Note:* The information in this section is applicable only if you ordered SAS/ACCESS Interface to JDBC (on SAS Viya).

To verify that SAS/ACCESS Interface to JDBC was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to JDBC LIBNAME statement:

   ```sas
   libname jdbc driver="full-jdbc-driver-name" URL="jdbc-connection-url" username="user-ID" password="user-password";
   
   If SAS/ACCESS to JDBC was successfully deployed, the execution of the LIBNAME statement returns results without error.
   
   2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to JDBC:

      ```sas
      caslib jdbclib datasource={srctype="jdbc", username="user-ID", password="user-password",}
If the data connector was successfully deployed, the results are the names of the tables in JDBC.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

---

### Verify SAS/ACCESS Interface to Microsoft SQL Server

**Note:** The information in this section is applicable only if you ordered SAS/ACCESS Interface to Microsoft SQL Server (on SAS Viya).

To verify that SAS Data Connector to Microsoft SQL Server was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Microsoft SQL Server LIBNAME statement:

   ```sas
   libname mmslib sqlsvr dsn="DSN-from-odbc.ini" user="user-ID" password="user-password";
   ```

   If SAS/ACCESS to Microsoft SQL Server was successfully deployed, the execution of the LIBNAME statement returns results without error.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Microsoft SQL Server:

   ```sas
   caslib mmslib datasource=(srctype="sqlserver", username="user-ID", password="password", odbc_dsn="DSN-from-odbc.ini");
   ```

   ```sas
   proc casutil;
   list files incaslib="mmslib";
   run;
   ```

   If the data connector was successfully deployed, the results are the names of the tables in Microsoft SQL Server.

   If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

---

### Verify SAS/ACCESS Interface to MySQL

**Note:** The information in this section is applicable only if you ordered SAS/ACCESS Interface to MySQL (on SAS Viya).

From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to MySQL LIBNAME statement:

```sas
libname mylib mysql server="mysql-host-name" database="mysql-database-name" user="user-ID" password="user-password";
```
If SAS/ACCESS to MySQL was successfully deployed, the execution of the LIBNAME statement returns results without error.

From SAS Studio, edit and run the following SAS code:

```
caslib mylib datasource=(srctype="mysql", username="user-ID", password="user-password", server="mysql-host-name", database="mysql-database-name");
proc casutil; list files incaslib="mylib"; run;
```

If the data connector was successfully deployed, the results are the names of the tables in PostgreSQL.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

---

### Verify SAS/ACCESS Interface to Netezza

**Note:** The information in this section is applicable only if you ordered SAS/ACCESS Interface to Netezza (on SAS Viya).

From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Netezza LIBNAME statement:

```
libname nlib netezza server="netezza-host-name" database="netezza-database-name" user="user-ID" password="user-password";
```

If SAS/ACCESS to Netezza was successfully deployed, the execution of the LIBNAME statement returns results without error.

If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

---

### Verify SAS/ACCESS Interface to ODBC

**Note:** The information in this section is applicable only if you ordered SAS/ACCESS Interface to ODBC (on SAS Viya).

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to ODBC LIBNAME statement:

   ```
   libname olib odbc dsn="DSN-from-odbc.ini" user="user-ID" password="password";
   ```

   If SAS/ACCESS to ODBC was successfully deployed, the execution of the LIBNAME statement returns results without error.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to ODBC:

   ```
caslib odbclib datasource=(srctype="odbc", username="user-ID", password="password", odbc_dsn="DSN-from-odbc.ini");
proc casutil;
list files incaslib="odbclib";
run;
```

   If the data connector was successfully deployed, the results are the names of the tables in ODBC.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.
Verify SAS/ACCESS Interface to Oracle

To verify that SAS/ACCESS Interface to Oracle was successfully deployed:

1 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Oracle LIBNAME statement:

   libname olib oracle path="path-to-database" user="user-ID" password="user-password";

   If SAS/ACCESS to Oracle was successfully deployed, the execution of the LIBNAME statement returns results without error.

2 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Oracle:

   caslib oralib datasource=(srctype="oracle" username="user-ID" password="password"
   path="path-to-database" schema="schema-ID");

   proc casutil;
   list files incaslib="oralib";
   run;

   If the data connector was successfully deployed, the results are the names of the tables in Oracle.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to PostgreSQL

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to PostgreSQL.

To verify that SAS/ACCESS Interface to PostgreSQL was successfully deployed:

1 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to PostgreSQL LIBNAME statement:

   libname plib postgres server="PostgreSQL-host-name" database="PostgreSQL-database-name" user="user-ID"
   password="password";

   If SAS/ACCESS to PostgreSQL was successfully deployed, the execution of the LIBNAME statement returns results without error.

2 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to PostgreSQL:

   caslib pglib datasource=(srctype="postgres", username="user-ID", password="password",
   server="PostgreSQL-host-name", database="PostgreSQL-database-name");

   proc casutil;
   list files incaslib="pglib";
   run;

   If the data connector was successfully deployed, the results are the names of the tables in PostgreSQL.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.
Verify SAS/ACCESS Interface to SAP ASE

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to SAP ASE (on SAS Viya).

To verify that SAS/ACCESS Interface to SAP ASE was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to SAP ASE LIBNAME statement:

   ```sas
   libname ase sapase server="sapase-host-name";
   database="sapase-database-name" user="user-ID" password="user-password";
   ```

   If SAS/ACCESS to SAP ASE was successfully deployed, the execution of the LIBNAME statement returns results without error.

   If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to SAP HANA

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to SAP HANA (on SAS Viya).

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to SAP HANA LIBNAME statement:

   ```sas
   libname sphlib saphana server="saphana-host-name" user="user-ID" password="user-password";
   ```

   If SAS/ACCESS to HANA was successfully deployed, the execution of the LIBNAME statement returns results without error.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to SAP HANA:

   ```sas
   caslib sphclib datasource=(srctype="hana", username="user-ID", password="user-password", server="saphana-host-name", instance="saphana-instance")
   ;
   proc casutil;
   list files incaslib="msclib";
   run;
   ```

   If the data connector was successfully deployed, the results are the names of the tables in SAP HANA.

   If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to SAP R/3

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to R/3 (on SAS Viya).
From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to SAP R/3 LIBNAME statement:

```sas
libname bwides r3 user="user-ID" password="user-password" client=800 ashost="sap-host-name" sysnr=06;
```

If SAS/ACCESS to SAP R/3 was successfully deployed, the execution of the LIBNAME statement returns results without error.

---

**Verify SAS/ACCESS Interface to Snowflake**

*Note:* The information in this section is applicable only if you ordered SAS/ACCESS Interface to Snowflake (on SAS Viya).

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Snowflake LIBNAME statement:

   ```sas
   libname snowlib sasiosnf server="Snowflake-server-address" user="user-ID" password="user-password";
   ```

   If SAS/ACCESS to Snowflake was successfully deployed, the execution of the LIBNAME statement returns results without error. If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Snowflake:

   ```sas
   caslib snowLib datasource=(srctype="Snowflake", username="user-ID", password="user-password", server="Snowflake-server-address",database="database-name");
   proc casutil; list files incaslib="snowLib"; run;
   ```

   If the data connector was successfully deployed, the results are the names of the tables in Snowflake. If no table information was returned for the data connector, you should perform the configuration steps again.

---

**Verify SAS/ACCESS Interface to Teradata**

*Note:* The information in this section is applicable only if you ordered SAS/ACCESS Interface to Teradata (on SAS Viya).

To verify that SAS/ACCESS Interface to Teradata was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Teradata LIBNAME statement:

   ```sas
   libname tlib teradata server="teradata-host-name" database="teradata-database-name" user="user-ID" password="user-Password";
   ```

   If SAS/ACCESS to Teradata was successfully deployed, the execution of the LIBNAME statement returns results without error.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Teradata:

   ```sas
   caslib tdlib datasource=(srctype="teradata", dataTransferMode="auto", username="user-ID", password="user-Password", server="teradata-host-name", database="teradata-database-name");
   proc casutil;
   ```
If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

**Verify SAS/ACCESS Interface to Vertica**

To verify that SAS/ACCESS Interface to Vertica was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Vertica LIBNAME statement:

   ```sas
   libname vert vertica server="Vertica-host-name" database="Vertica-database-or-schema-name" user="user-ID" password="user-password";
   
   If SAS/ACCESS to Vertica was successfully deployed, the execution of the LIBNAME statement returns results without error.
   
2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Vertica:

   ```sas
   caslib vertlib datasource=(srctype="Vertica", username="user-ID", server="Vertica-host-name", password="user-password", database="Vertica-database-or-schema-name");
   proc casutil;
   list files incaslib="vertlib";
   run;
   
   If the data connector was successfully deployed, the results are the names of the tables in Vertica.
   
If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.
Completing the Deployment

Save Snapshot Directory Content

If you successfully deployed your software using Ansible, the process saved valuable information for later use. The information is saved in the sas_deployment.tgz file in the directory in which you saved the playbook, in the /snapshot/epoch subdirectory. The sas_deployment.tgz file includes the following files, among others:

- the inventory file that is used in the deployment
- the vars.yml file that is used in the deployment
- the deployment log

SAS recommends that you copy the sas_deployment.tgz file and save it to a separate location, possibly on another machine. You have a backup of important files that might be required later, such as to update an existing order.

Share Important Deployment Information with the Administrators

If other persons are responsible for administering your SAS deployment, it is recommended that you share the following important information with them:

- The deployment type: Did you deploy the programming interface only or did you perform a full deployment? The type was determined by whether you made a programming-only playbook or not (see “Specify the Deployment Type” on page 59).
- The location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.
- The URL to access the software: What products did you deploy?

  - If you deployed products from the SAS Event Stream Processing Family, share the URL for SAS Event Stream Processing Studio:
    https://reverse-proxy-server/SASEventStreamProcessingStudio
  - If your order also included SAS Event Stream Manager, share the following URL:
If you deployed the programming interface only, your administrators should use SAS Studio. The URL is http://reverse-proxy-server/SASStudio.

If you performed a full deployment, your administrators should use SAS Environment Manager. You used SAS Environment Manager to configure your environment for a full deployment as described in “Configure Your Environment with SAS Environment Manager” on page 121. Use the same URL that you used in that section.

If you enabled multi-tenancy in your deployment, information about the LDAP configuration should be included.

- Did you use a single LDAP server for all tenants or an LDAP server per tenant?
- Did you use a fixed LDAP structure or a custom LDAP structure?

Next Steps for SAS Event Stream Processing

Users

If your order included SAS Event Stream Processing, you might need to consult some additional documentation:

- The Streamviewer component was also installed along with SAS Event Stream Processing. Streamviewer is a graphical user interface that enables you to visualize events as they stream through event stream processing models. Its use is optional.
  
  To access Streamviewer, navigate to the following URL:
  
  https://reverse-proxy-server/SASEventStreamProcessingStreamviewer

  For reverse-proxy-server, substitute the host name from the machine that you assigned to the [httpproxy] host group in the inventory file.

  Log in when prompted.

  For a complete guide to Streamviewer usage, see Visualizing Event Streams with Streamviewer.

- If your order included SAS Event Stream Processing for CAS, you now have the option to use an additional CAS action set, espCluster. A second SAS Event Stream Processing action set, loadStreams, is included with all SAS Viya orders. For more information, see Using SAS Event Stream Processing with SAS Cloud Analytic Services Actions in SAS Event Stream Processing: Advanced Topics.

- Read additional documentation about SAS Event Stream Processing. Links to all SAS Event Stream Processing documentation are available on the SAS Event Stream Processing product page. All product user documentation is also available via single sign-on from the SAS Event Stream Processing user interfaces.

Refer to Additional Documentation

- To perform initial administrative tasks, see SAS Viya Administration: Initial Tasks.

- To locate administration and additional documentation for solutions and offerings, go to the SAS Viya documentation page:

  http://support.sas.com/documentation/onlinedoc/viya/index.html

- To locate usage information, refer to the Help that is available from the SAS Viya product and administrative interfaces.
Managing Your Software

Overview

What Is an Update? 180
What Is an Add-On Product? 180
What Is an Upgrade? 180

Updating Your SAS Viya Software

Overview 181
User Requirements for Performing the Update 182
Synchronize the Mirror Repository 182
(Optional) List the Packages That Are Available for Update 183
Update with Yum 183
Update with Zypper 184
Update with Ansible 186
Post-Update Tasks 187
Update the Type of Deployment with Ansible 188

Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4 191

Overview 191
Identify Shared Projects in Model Studio 191
Shut Down All Tenant Services 192
Prepare to Upgrade SAS Event Stream Processing Software 192
Add SAS Viya Software 193
Apply Guest Authorization Rule Changes 196
User-Related Tasks 197

Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya 197

Overview 197
Identify Model Studio Projects Derived from SAS Visual Analytics Reports with Partitioning 198
Identify Shared Projects in Model Studio 198
Delete the mmLibs Caslib 199
Upgrade Using a Mirror Repository 200
Create the New SAS Viya 3.4 Playbook 200
Merge the User-Modified Files 201
(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook 203
Update SAS Visual Text Analytics with Yum 203
Prepare to Upgrade SAS Viya Software 204
Delete the Default Backup Schedule Job 205
Change the dbmsowner User Password 205
Verify the Health of SAS Infrastructure Data Server 206
Prepare to Upgrade SAS Event Stream Processing Software 207
Back Up Your SAS Visual Investigator or SAS Intelligence and Investigation Management Configuration 208
Prepare to Upgrade SAS Visual Investigator or SAS Intelligence and Investigation Management 208
Shut Down All Tenant Services 209
Overview

What Is an Update?

An update provides modifications for features that are not working as intended or adds minor software enhancements and compatibility. Software updates are released to address security issues when they occur, to address minor bugs discovered in the software, and to improve the operation of hardware or peripherals. These incremental updates improve the operation of your software and are small enough that they do not require a new order. Updated software is intended to be compatible with existing configuration, content, and data. To perform an update, you will run the same tools that were run during the initial deployment. You might determine that your software needs updating or you might be notified by SAS that updates are available.

Note: Converting a single-tenant deployment to a multi-tenant deployment, either through an update or an upgrade, is not supported.

Note: Converting the LDAP configuration of a multi-tenant deployment from a single LDAP server for all tenants to a separate LDAP server for each tenant, either through an upgrade or an update, is not supported.

To update your SAS Viya software, see “Updating Your SAS Viya Software” on page 181.

What Is an Add-On Product?

An add-on product is new software that you can order and then install with your currently deployed software. You will need a new order for an add-on product. Adding new software to your deployment will also update your currently deployed software.

Because an add-on product is added to the currently deployed software in an environment, you might need to expand your environment’s capacity before installing an add-on product.

For information about adding on a product, see “Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4” on page 191.

What Is an Upgrade?

There are two types of upgrade associated with SAS Viya 3.4 software. You can upgrade products in SAS Viya 3.4, such as SAS Econometrics 8.3 to SAS Econometrics 8.4. You can also upgrade to SAS Viya 3.4 from earlier versions of SAS Viya, such as SAS Viya 3.2 or SAS Viya 3.3.
Upgrading Products Within SAS Viya 3.4

An upgrade to SAS Viya products adds significant feature changes or improvements to those products. The changes are important enough to require a new order. To upgrade products, you will perform the same steps that are used for adding on products. You will need a new order, and you must get an updated version of the Orchestration CLI to create a new playbook. Upgrading products will also update your currently deployed software.

If you are not sure if the product that you are upgrading from and the product that you are upgrading to are both in SAS Viya 3.4, go to the Product Compatibility Matrix to determine the SAS Viya version that your products are running on. If the product that you upgrading from is from a version of SAS Viya earlier than 3.4, you will actually be upgrading SAS Viya and should use that process.

For information about upgrading a product within SAS Viya 3.4, see “Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4” on page 191.

Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya

An upgrade for SAS Viya adds significant feature changes or improvements to SAS Viya. To perform an upgrade, you will run the same tools that were run during the initial deployment. You will need a new order to upgrade your deployed software, and you must get an updated version of the Orchestration CLI to create a new playbook. Add-on products present in the order are installed as part of the upgrade process. An upgrade might require changes to the deployed software’s configuration.

You might determine that your software needs upgrading or you might be notified by SAS that upgrades are available. SAS recommends creating a backup of the deployed software environment before performing an upgrade.

Note: Converting a single-tenant deployment to a multi-tenant deployment, either through an update or an upgrade, is not supported.

Note: Converting the LDAP configuration of a multi-tenant deployment from a single LDAP server for all tenants to a separate LDAP server for each tenant, either through an upgrade or an update, is not supported.

For information about upgrading to SAS Viya 3.4 from other versions of SAS Viya, see “Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya” on page 197.

Updating Your SAS Viya Software

Overview

An update replaces some or all of your deployed software with the latest version of the software. You perform the update with the same command that was used to install SAS Viya, and use the same software order.

- To see what updates are available for your deployed software, go to the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.
- Use the same tool (Ansible, yum, or Zypper) to update that you used to install. For example, if you used an Ansible playbook for your initial installation, update with Ansible.
- If you want to update your deployment from programming-only to full, you can use only Ansible.
- SAS Data Science is deployed as programming-only. However, to obtain the visual interface for the components of SAS Data Science, different products must be licensed. The resulting order should be deployed using the instructions in “Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4” on page 191. Contact your SAS account representative to determine the exact products that are required.
SAS might update components of the Ansible playbook that is used to deploy your SAS Viya software. You will need to download the current version of the SAS Orchestration CLI to generate a new Ansible playbook for your deployment, and then run the new Ansible playbook.

Here are other considerations when preparing for an update:

- The update process preserves any user-modified configuration values in the vars.yml file, but changes made to other files in the deployment might be lost. Therefore, SAS recommends that you make changes to vars.yml when possible in order to avoid any loss of customizations that you made to other files.
- You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.
  
  Note: If you have added any CAS servers to your initial deployment, you must update those CAS servers as well as each machine in your initial deployment.
- If you are using a PDF version of this guide, go to the Deployment Guides web page at https://support.sas.com/en/documentation/install-center/viya/deployment-guides.html and verify that you have the latest version of the deployment documentation before you start the update process. The release date of each document is located in the bottom right corner of the front page.
- Updating SAS Viya software requires an outage period because some SAS Viya services are stopped and restarted automatically during the update process. The update process is the same regardless of whether the deployment is single-tenant or multi-tenant.

User Requirements for Performing the Update

To perform the update process, you must have administrator privileges for the machine. In addition, your account must have superuser (sudo) access. To verify sudo user privileges, run the following command:

```
sudo -v or sudo -l
```

Synchronize the Mirror Repository

If you are using a mirror repository:

1. (Optional) To list the packages that are available for the update process, run the following command on the machine where the mirror repository is located:

   ```
   mirrormgr mirror diff --deployment-data path-to-deployment-zip-file-from-SOE
   --path path-to-mirror-destination --latest
   
   Note: The mirrormgr mirror diff command returns the available files for all supported platforms of the products in the deployment. To filter out unwanted content from the output of the command, use a pipe and the grep command. For example, add the following at the end of the preceding command to filter out all file names containing "suse":
   | grep -v "suse"
   ```

2. Synchronize the deployment’s mirror repository with SAS’ mirror repository. Use the same options to update the mirror repository that you used to create the mirror repository. For more information, see “SAS Mirror Manager and the Mirror Repository” on page 56.

   To synchronize, run the following command on the machine where the connected mirror repository is located:

   ```
   mirrormgr mirror --deployment-data path-to-deployment-zip-file-from-SOE
   --path path-to-mirror-destination --latest
   ```
If you are in a deployment without internet access, move the files from the machine where the connected mirror repository is located to the machine where the unconnected mirror repository is located.

(Optional) List the Packages That Are Available for Update

To list the packages that are available for the update process, run the following command:

For Red Hat Enterprise Linux:

```bash
sudo yum check-update "sas-*"
```

For SUSE Linux:

```bash
sudo zypper list-updates | grep "sas-*"
```

Update with Yum

Important: You can use yum to update your software only if your deployment is on Red Hat Enterprise Linux or an equivalent distribution.

To update a SAS Viya deployment using yum:

1. (Optional) Record the existing list of installed software before you begin.
   
   a. On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.
   
   Run the following command to create a text file that lists the RPM packages:
   
   ```bash
   sudo rpm -qg SAS > /sas/install/viya_rpms.txt
   ```
   
   b. On each machine in your deployment, create a file that lists the yum groups that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.
   
   Run the following command to create a text file that lists the yum groups:
   
   ```bash
   sudo yum grouplist "SAS*" > /sas/install/viya_yumgroups.txt
   ```
   
   Note: If you receive a message such as the following, it can be ignored.
   
   Repository repository-name is listed more than once in the configuration
   
2. On each machine in your deployment, stop all SAS Viya services in the correct order. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

3. If you have deployed SAS Event Stream Processing, stop the metering server:

   ```bash
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
   ```
   
   Replace host-name with the host name of the machine where the metering server is running.
   
   Replace http-port with the port number for the metering server. By default, it uses port 31001.

4. On each machine in your deployment, run the following commands to show the available updates for all SAS Viya software on the machine:

   ```bash
   sas_groups=${LANG=en_US sudo yum grouplist -v "sas-*" | awk -F'{()}' \
   
   "/"Installed Groups:/{found=1;next}/"[[:alnum:]]/"{found=0}found \ 
   
   \{print@*$\{NF-1\}"}
   ```
sudo yum update $sas_groups $(rpm -qg "SAS")

You must run these commands to update any external software applications on which the SAS yum groups depend.

5 At the prompt Is this ok, review the available updates and then enter y.

6 If the deployment that is being updated includes SAS Model Manager, perform this step.

On each machine in your deployment, run the following command:

```bash
sudo runuser --shell "/bin/sh" \
   --login sas \ 
   --command "/opt/sas/spre/home/SASFoundation/utilities/bin/post_install build_add_cmacros"
```

7 On each machine in your deployment, start all SAS Viya services in the correct order. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

Note: If the CAS controller fails to start because of a permission denied error, navigate to `/opt/sas/viya/config/data/cas/default/referenceData`, delete the files, and start the CAS controller again.

8 (Optional) Record the new list of installed software.

   a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

       Run the following command to create a text file that lists the RPM packages:

       ```bash
       sudo rpm -qg SAS > /sas/install/new_viya_rpms.txt
       ```

   b On each machine in your deployment, create a file that lists the yum groups that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

       Run the following command to create a text file that lists the yum groups:

       ```bash
       sudo yum grouplist "SAS*" > /sas/install/new_viya_yumgroups.txt
       ```

Note: If you receive a message such as the following, it can be ignored.

Repository repository-name is listed more than once in the configuration

You can see the differences between the previous and current deployments by comparing the lists of installed software that preceded the update (Step 1 on page 183) and that followed the update.

   c To verify that a specific update was applied, compare the contents of the text file created in Step 8a on page 184 to the packages listed for the specific update. The package list for a specific update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

**Update with Zypper**

Important: You can use zypper to update your software only if your deployment is on SUSE Linux or an equivalent distribution.

To update a SAS Viya deployment using zypper, repeat these steps for each machine in the deployment:

1 (Optional) Record the existing list of installed software before you begin.

   a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine...
where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Run the following command to create a text file that lists the RPM packages:

```
sudo rpm -qg SAS > /sas/install/viya_rpms.txt
```

b On each machine in your deployment, create a file that lists the SAS packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Run the following command to create a text file that lists the RPM packages:

```
sudo rpm -qa | grep "sas-" > /sas/install/viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

Repository repository-name is listed more than once in the configuration

2 Stop all the SAS services on the machine:

```
sudo /etc/init.d/sas-viya-all-services stop
```

3 If you have deployed SAS Event Stream Processing, stop the metering server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace `host-name` with the host name of the machine where the metering server is running.

Replace `http-port` with the port number for the metering server. The default port is 31001.

4 To update all SAS Viya software on the machine:

```
sudo zypper update "sas-*"
```

5 At the prompt `Continue? [y/n]`, review the available updates and then enter `y`.

6 Restart the services that are installed on the machine, including the CAS controller, a SAS object spawner, and SAS Studio. You must also start SAS/CONNECT if it is included in your deployment.

To restart all the SAS services on the machine:

```
sudo /etc/init.d/sas-viya-all-services start
```

7 If the CAS controller fails to start because of a permission denied error, navigate to `/opt/sas/viya/config/data/cas/default/referenceData`, delete the files, and start the CAS controller again.

8 (Optional) Record the new list of installed software.

a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Run the following command to create a text file that lists the RPM packages:

```
sudo rpm -qg SAS > /sas/install/new_viya_rpms.txt
```

b On each machine in your deployment, create a file that lists the SAS packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Run the following command to create a text file that lists the RPM packages:

```
sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

Repository repository-name is listed more than once in the configuration.
You can see the differences between the previous and current deployments by comparing the lists of installed software that preceded the update (Step 1 on page 184) and that followed the update.

c To verify that a specific update was applied, compare the contents of the text file created in Step 8a on page 185 to the packages listed for the specific update. The package list for a specific update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

**Update with Ansible**

To update a SAS Viya deployment using Ansible:

1. (Optional) Record the existing list of installed software before you begin.

   a. On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

      Run the following command to create a text file that lists all the RPM packages:

      ```
sudo rpm -qg SAS > /sas/install/viya_rpms.txt
      ```

   b. On each machine in your deployment, create a file that lists the yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

      - Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:

        ```
sudo yum grouplist "SAS*" > /sas/install/viya_yumgroups.txt
        ```

      - Run the following command to create a text file that lists the RPM packages on SUSE Linux:

        ```
sudo rpm -qa | grep "sas-" > /sas/install/viya_packages.txt
        ```

      Note: If you receive a message such as the following, it can be ignored.

      ```
      Repository repository-name is listed more than once in the configuration
      ```

2. Review the *_deployment.* files (for example, casconfig_deployment.lua) in the existing deployment for any user-modified changes. If there are any user-modified changes to the *_deployment.* files, back up the file and update the vars.yml file with the changes before you perform the update. If you have questions, contact SAS Technical Support.

   Note: SAS recommends that you add your customizations to the vars.yml file rather than to a *_deployment.* file in order to preserve your customizations. Otherwise, your customizations would be lost during the update process.

   By default, the update process backs up the following files:

   For CAS:

   ```
   /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
   /opt/sas/viya/config/etc/cas/default/casconfig.lua
   /opt/sas/viya/config/etc/cas/default/cas.hosts
   ```

   For SAS Object Spawner:

   ```
   /opt/sas/viya/config/etc/spawner/default/spawner.cfg
   ```

   For SAS/CONNECT:

   ```
   /opt/sas/viya/config/etc/sysconfig/connect/default/sas-connect
   ```
3 To verify the health of the SAS Infrastructure Data Server before running the playbook, perform the task in "Verify the Health of SAS Infrastructure Data Server" on page 206.

4 To initiate the update:

    ansible-playbook update-only.yml

Be sure to use the same options that you used when you performed the initial deployment. For more information, see "Deploy the Software" on page 115.

If you added additional CAS servers to your deployment, run the command for each CAS server. When you added the CAS servers to your deployment, you created inventory.ini and vars.yml files for each additional CAS server. Those same files must be specified in the command to initiate the update.

To initiate the update on additional CAS servers:

    ansible-playbook -i CAS-server-inventory-file-name update-only.yml -e "@CAS-server-vars-file-name"

Be sure to add the appropriate options that you used for the initial deployment of the CAS servers. Repeat this command for each additional CAS server.

5 (Optional) Record the new list of installed software.

   a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 60.

     Run the following command to create a text file that lists the RPM packages:

     sudo rpm -qg SAS > /sas/install/new_viya_rpms.txt

   b On each machine in your deployment, create a file that lists the yum groups or packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 60.

     - Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:

     sudo yum grouplist "SAS*" > /sas/install/new_viya_yumgroups.txt

     - Run the following command to create a text file that lists the RPM packages on SUSE Linux:

     sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt

     Note: If you receive a message such as the following, it can be ignored.

     Repository repository-name is listed more than once in the configuration

     You can see the differences between the previous and current deployments by comparing the lists of installed software that precedes the update (Step 1 on page 186) and that follows the update.

   c To verify that a specific update was applied, compare the contents of the text file created in Step 5a on page 187 to the packages listed for the specific update. The package list for a specific update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

Post-Update Tasks

After you complete an update, sometimes you must perform additional tasks. This section lists those updates, how to determine whether the update has been applied, and the manual steps required to finish the installation.

SAS Hadoop Plug-ins

Note: If you are not using SASHDAT on HDFS, skip this section.
If you receive an updated RPM for SAS Hadoop Plug-ins, for each Hadoop environment, you must determine whether to install it. Use the newer version of the RPM in each Hadoop environment. Install the updated RPM only if its version is newer than the one that is currently installed.

1. Install the updated RPM for SAS Hadoop Plug-ins on the CAS controller.

2. To find out the version number of the RPM that was just installed:
   
   rpm --queryformat "%{VERSION}" -q sas-hdatplugins

   The version number is displayed in a format of three two-digit numbers separated by periods. Examples are 03.13.00 or 03.04.02. The variable VERSION is used to represent the actual version number in your environment.

3. On each Hadoop name node or data node, use the appropriate method to identify the currently installed version:
   
   - For SAS Hadoop Plug-ins that were installed on the Hadoop cluster with the sashdat-install.sh command:
     
     cat /opt/sas/HDATHome/SAS_VERSION

     The output should include a version number in the VERSION format that is shown in the output of Step 2 on page 188.

   - For SAS Hadoop Plug-ins that were installed on the Hadoop cluster with Cloudera Manager, the parcel name should be SASHDAT-VERSION.p0.1.

   - For SAS Hadoop Plug-ins that were installed on the Hadoop cluster with Ambari, the stack name should be SASHDAT-VERSION.s01.

4. If the installed version that is displayed by Step 3 on page 188 is equal to or greater than the version number of the RPM displayed by the command in Step 2 on page 188, skip Step 5 on page 188.

5. If the installed version that is displayed by Step 3 on page 188 is less than the version number of the RPM that is displayed by the command in Step 2 on page 188, follow the steps described in “Deploying SAS Plug-ins for Hadoop” on page 262 to update your Hadoop environment to the newer version of the SAS Hadoop Plug-ins.

Update the Type of Deployment with Ansible

Overview

Using Ansible, you can modify your deployment from programming-only to full.

Note: If you make changes to the proxy.conf file and then rerun the playbook, those changes are overwritten. A copy of the proxy.conf should be created in the /etc/httpd/conf.d directory when you rerun the playbook. Use this copy, along with the instructions in “Post-installation Tasks” on page 119 to make changes to the updated proxy.conf file.

Update the Type of Deployment

To update a SAS Viya deployment from programming-only to full:

1. Complete the task in “Generate a New Ansible Playbook” on page 189.
   
   Note: Do not select the programming-only option in the SAS Orchestration CLI when generating the new playbook. The default playbook generated is for a full deployment.

2. To change from a programming-only deployment to a full deployment, you must complete the configuration for the visual attributes of the deployment. For more information see “Post-installation Tasks” on page 119.
Note: When the programming-only interface is deployed, SAS Studio is accessible on port 7080. However, when updating from a programming-only deployment to a full deployment, SAS Studio uses a dynamically assigned port.

Generate a New Ansible Playbook

You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

If updates are needed in the Ansible playbook, to generate and apply a new Ansible playbook for your deployment:

1 (Optional) Record the existing list of installed software before you begin.

a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

   Run the following command to create a text file that lists all the RPM packages:
   ```
   sudo rpm -qg SAS > /sas/install/viya_rpms.txt
   
   ```

b On each machine in your deployment, create a file that lists the yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

   Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:
   ```
   sudo yum grouplist "SAS*" > /sas/install/viya_yumgroups.txt
   
   ```

   Run the following command to create a text file that lists the RPM packages on SUSE Linux:
   ```
   sudo rpm -qa | grep "sas-" > /sas/install/viya_packages.txt
   
   ```

   Note: If you receive a message such as the following, it can be ignored.

       Repository repository-name is listed more than once in the configuration

2 Use the Software Order Email (SOE) for your original deployment to download the current version of the SAS Orchestration CLI.

3 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site. For more information, see “Create a Playbook” on page 58.

   Note: To modify the type of deployment from programming-only to full, use the correct --deployment-type command line option.

4 You must extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to `/sas/install/`, you might extract the new playbook to `/sas/upgrade/` instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:

   a To preserve the original vars.yml file and the inventory file.

   b To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is accidentally extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

   To extract the new playbook, use a command that is similar to the following:
   ```
   tar xf SAS_Viya_playbook.tgz -C /sas/upgrade/
   ```
5 Merge the vars.yml file and the inventory file from the previous deployment into the new playbook. If the previous inventory file contains any spaces that are used to indent machine names, do not include the extra spaces.

a Compare the two inventory files to check for additions or changes in the newer set of files. Be sure to evaluate the comments to determine whether the requirements for host groups changed between releases of the software.

```bash
diff /sas/install/sas_viya_playbook/vars.yml /sas/upgrade/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory-file /sas/upgrade/sas_viya_playbook/inventory.ini
```

b If the new files contain new content, then merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.

c If the original vars.yml file from the deployment that is being upgraded contains a value for the casenv_tenant variable, it must be removed. Run the following commands to remove the registered CAS service.

Note: Enter each of the three commands on a single line. Multiple lines are used here to improve readability.

```bash
cd /opt/sas/viya/home/bin
./sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
agent service deregister "cas-{casenv_tenant}-default-http"

./sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
agent service deregister "cas-{casenv_tenant}-default"
```

d If you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.

6 To apply the new Ansible playbook, change to the directory where the new playbook is located:

```bash
cd /sas/upgrade/
```

Run the following command:

```bash
ansible-playbook site.yml
```

7 If you removed the CAS service that is associated with a casenv_tenant variable (described in Step 3), update any bookmarked URLs to remove that value and use cas-shared-default-http instead. For example, if your original deployment contained a casenv_tenant value of viya32, change it from http://host.company.com/cas-viya32-default-http to http://host.company.com/cas-shared-default-http.

Note: Do not include casenv_tenant in your new vars.yml. This variable is no longer used.

8 (Optional) Record the new list of installed software.

a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Run the following command to create a text file that lists the RPM packages:

```bash
sudo rpm -qg SAS > /sas/install/new_viya_rpms.txt
```

b On each machine in your deployment, create a file that lists the SAS yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.
Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:

```
sudo yum grouplist "SAS*" > /sas/install/new_viya_yumgroups.txt
```

Run the following command to create a text file that lists the RPM packages on SUSE Linux:

```
sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

You can see the differences between the previous and current deployments by comparing the lists of installed software that precedes the update (Step 1 on page 189) and that follows the update.

To verify that a specific update was applied, compare the contents of the text file created in Step 8a on page 190 to the packages listed for the specific update. The package list for a specific update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

---

**Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4**

**Overview**

The process for adding SAS Viya software to an existing deployment and for upgrading a product within SAS Viya 3.4 is the same. Adding new software to your deployment or upgrading a product also updates your currently deployed software. You will need a new software order to add SAS Viya software to an existing deployment or to upgrade SAS Viya products in SAS Viya 3.4.

Adding SAS Viya software to an existing deployment or upgrading SAS Viya products requires an outage period. During the process, all SAS Viya services must be stopped and then restarted. For multi-tenant deployments, the provider administrator should communicate the outage to the tenant customers before the Ansible playbook is run or before any services are manually stopped.

This chapter includes all the tasks that are performed in order to add SAS Viya software to your deployment. The tasks apply to all deployments regardless of the version of the source environment, the software installed, or whether the deployment is single-tenant or multi-tenant.

You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Note: Converting a single-tenant deployment to a multi-tenant deployment is not supported.

Note: Converting the LDAP configuration of a multi-tenant deployment from a single LDAP server for all tenants to a separate LDAP server for each tenant is not supported.

Note: To deploy or upgrade SAS Data Agent, follow the instructions in SAS Data Agent on Linux: Deployment Guide.

**Identify Shared Projects in Model Studio**

If the current deployment includes SAS Visual Data Mining and Machine Learning, perform this task.

When Model Studio is upgraded, any shared project becomes a private project. To identify which Model Studio projects are shared before the upgrade is performed:

1. Log on to Model Studio as a user that is a member of the SAS Administrators group.
2 In the Projects pane, select Table view.

3 For each project that is not labeled Private in the Shared With column, notify the project owner designated in the Created By column to re-share the project with the group in the Shared With column after the upgrade.

**Shut Down All Tenant Services**

If this is not a multi-tenant deployment, skip this task.

For a multi-tenant deployment, you must shut down the SAS Viya deployment including all tenants in the correct order. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

**Prepare to Upgrade SAS Event Stream Processing Software**

If you have not purchased a newer version of SAS Event Stream Processing or SAS Event Stream Manager, skip this task.

1 Check for configuration files that have been customized.

The version of the SAS Event Stream Processing software to which you are upgrading uses a new configuration file, esp-properties.yml. Some customizations that you have made to your installed software will not be used after the upgrade process has completed unless you merge them into the new file.

On the machine where SAS Event Stream Processing is installed, compare any modified files with the copies that were included with the older version of the software. Run the following commands:

**Note:** Specify each diff command on a single line. Multiple lines are used here for improved readability.

```bash
diff -u $DFESP_HOME/etc/connectors.excluded /opt/sas/viya/config/etc/
SASEventStreamProcessingEngine/default/connectors.excluded

diff -u $DFESP_HOME/etc/esp-logger.xml /opt/sas/viya/config/etc/
SASEventStreamProcessingEngine/default/esp-logger.xml

diff -u $DFESP_HOME/etc/metatags.conf /opt/sas/viya/config/etc/
SASEventStreamProcessingEngine/default/metatags.conf

diff -u $DFESP_HOME/etc/security-properties.yml /opt/sas/viya/config/etc/
SASEventStreamProcessingEngine/default/security-properties.yml
```

2 Save the results of the diff commands to a file. You will refer to this file when you are ready to merge the configuration changes into esp-properties.yml. For more information, see “Complete SAS Event Stream Processing Upgrade Steps” on page 214.

3 If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace host-name with the host name of the machine where Streamviewer is running.

Replace http-port with the port number that you provided when you started Streamviewer with the start-up script.

4 Stop the metering server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace host-name with the host name of the machine where the metering server is running.

Replace http-port with the port number for the metering server. It uses the default port, 31001.
If you are upgrading from SAS Event Stream Manager 4.3, you must remove a Consul entry that refers to the database schema. Later versions of SAS Event Stream Manager do not create this entry.

Set an environment variable that includes the Consul port, and then run the command to delete the entry. Here is an example:

**Note:** Enter each command on a single line. Some commands occupy multiple lines to improve readability.

```bash
sudo -u sas -i
source /opt/sas/viya/config/consul.conf
export CONSUL_HTTP_TOKEN='cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token'
/opt/sas/viya/home/bin/sas-bootstrap-config kv delete 'config/esm-service/spring/datasource/schema'
```

**Add SAS Viya Software**

To add SAS software and update a SAS Viya deployment:

1. Before you begin, you should review the “Introduction” on page 1, “System Requirements” on page 21, and “Pre-installation Tasks” on page 55 chapters of this guide.

2. (Optional) Record the existing list of installed software before you begin.
   a. On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

   To create a text file that lists all the RPM packages:
   ```bash
   sudo rpm -qg SAS > /sas/install/viya_rpms.txt
   ```

   b. On each machine in your deployment, create a file that lists the yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

   - Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:
     ```bash
     sudo yum grouplist "SAS*" > /sas/install/viya_yumgroups.txt
     ```

   - Run the following command to create a text file that lists the RPM packages on SUSE Linux:
     ```bash
     sudo rpm -qa | grep "sas-" > /sas/install/viya_packages.txt
     ```

   **Note:** If you receive a message such as the following, it can be ignored.

   ```bash
   Repository repository-name is listed more than once in the configuration
   ```

3. If your deployment used a mirror repository, you must download the current version of SAS Mirror Manager. For more information, see “SAS Mirror Manager and the Mirror Repository” on page 56.

4. If you are adding software that is mirrored, you must update the mirror using the ZIP file that is attached to your new Software Order Email (SOE) before performing these steps. For more information, see “Synchronize the Mirror Repository” on page 182. If you are upgrading existing software to a new version, you might want to create a new mirror so that you can delete the old files after the upgrade. For more information, see “SAS Mirror Manager and the Mirror Repository” on page 56.

5. Download the latest SAS Orchestration CLI. For more information, see “Create a Playbook” on page 58.

6. Using the SAS Orchestration CLI that you downloaded, create a new playbook. For more information, see “Create a Playbook” on page 58.
You must extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to `/sas/install/`, you might extract the new playbook to `/sas/addon/` or `/sas/upgrade/` instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:

- To preserve the original `vars.yml` file and the inventory file.
- To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following

```
tar xf SAS_Viya_playbook.tgz -C /sas/new-playbook-directory-name/
```

7 Merge the `vars.yml` file and the inventory file from the previous deployment into the new playbook. If the previous inventory file contains any spaces that are used to indent machine names, do not include the extra spaces.

a Compare the two `vars.yml` files, and compare the two inventory files. Check for additions or changes in the newer set of files. Be sure to evaluate the comments to determine whether the requirements for host groups changed between releases of the software.

**Note:** Enter each command on a single line. Multiple lines are used here to improve readability.

```
diff /sas/install/sas_viya_playbook/vars.yml
    /sas/new-playbook-directory-name/sas_viya_playbook/vars.yml

diff /sas/install/sas_viya_playbook/inventory.ini
    /sas/new-playbook-directory-name/sas_viya_playbook/inventory.ini
```

**Note:** The `[consul]`, `[httpproxy]`, and `[operations]` host groups must be present in `/sas/new-playbook-directory-name/sas_viya_playbook/inventory.ini` and must contain the host entries from the original deployment.

**Note:** The `[consul]` and `[httpproxy]` host groups must be present in each `/sas/new-playbook-directory-name/sas_viya_playbook/CAS-server-inventory-file-name` and must contain the host entries from the original deployment.

b If the new files contain new content, then merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.

**Note:** All host groups that are present in the inventory file from the previous deployment must remain on the same machines in the inventory file for the new deployment. New host groups that were not in the previous deployment should be assigned to machines in the current deployment. Review the comments that precede each host group before assigning host groups to machines. For more information, see “Assign the Target Machines to Host Groups” on page 77.

c If you have questions about whether to add a key/value pair from an older file to the new file, contact SAS Technical Support.

d If you have additional CAS servers in your deployment, perform the following steps:

i When you added the CAS servers to your deployment, you created a `inventory.ini` and `vars.yml` file for each additional CAS server. Copy the `inventory.ini` and `vars.yml` file for each additional CAS server to the same location that you extracted the new playbook to.

```
/sas/addon/sas_viya_playbook/CAS-server-inventory-file-name
/sas/addon/sas_viya_playbook/CAS-server-vars-file-name
```

ii Make the same edits to the `inventory.ini` and `vars.yml` file for each additional CAS server that you made to merge the `vars.yml` file and the inventory file from the previous deployment into the new playbook.
To verify the health of SAS Infrastructure Data Server, perform the task in "Verify SAS Infrastructure Data Server" on page 163.

Complete the tasks in "Disk Space Considerations" on page 26, as appropriate.

It is recommended to add a 404 redirect in your deployment's web server to prevent users from accessing the deployment before the process is completed. See your web server documentation.

Note: As administrators, use a secondary URL to complete the deployment steps.

Install your SAS Viya software beginning with "Modify the vars.yml File" on page 81.

If you added additional CAS servers to your deployment, run the command for each CAS server. When you added the CAS servers to your deployment, you created inventory.ini and vars.yml files for each additional CAS server. Those same files must be specified in the command to deploy the software.

```bash
ansible-playbook -i CAS-server-inventory-file-name site.yml -e "@CAS-server-vars-file-name"
```

Be sure to add the appropriate options that you used for the initial deployment of the CAS servers. Repeat this command for each additional CAS server.

After you install the software, you must complete the post-installation tasks that are appropriate for your deployment.

- If SAS Event Stream Processing was added, perform the steps in "Complete SAS Event Stream Processing Upgrade Steps" on page 214.
- If SAS Event Stream Processing or SAS Analytics for IoT was added, perform the steps in "Complete SAS Event Stream Processing and SAS Analytics for IoT Setup" on page 149 to complete the setup process.
- If SAS Event Stream Manager was added, complete SAS Event Stream Manager setup on page 153.
- Configure high availability in SAS Studio on page 126.
- If you added any SAS/ACCESS software to your deployment, configure data access on page 131 as appropriate.
- Configure any products that you have added to your deployment and that have specific steps listed in "Post-installation Tasks" on page 119.
- If you are upgrading SAS Visual Investigator 10.3.x or 10.4.x, or SAS Intelligence and Investigation Management 1.1 or 1.2, perform this step.

With access to SAS Visual Investigator or SAS Intelligence and Investigation Management, update the authorization rules for data objects that were previously added within SAS Visual Investigator 10.3.x or 10.4.x, or SAS Intelligence and Investigation Management 1.1 or 1.2 application. For more information, see Alert Entity-Level Security in SAS Visual Investigator 10.5: Administrator's Guide. Without performing this action, SAS Visual Investigator and SAS Intelligence and Investigation Management users and administrators have limited capability to manage data objects and alerts.

- Validate the deployment on page 157.

- If this is a multi-tenant deployment, for each tenant you must merge the relevant content of the original tenantID_vars.yml file that was used for the original deployment into the format of the tenantID_vars.yml file for the new deployment. For each tenant:
  - Compare the original tenantID_vars.yml file that was used to onboard the tenant with the sample_tenant_vars.yml file in sas_viya_playbook/samples. Perform one of the two following actions:
    - If the two files contain the same properties and structure, copy the original tenantID_vars.yml file into the sas_viya_playbook directory.
If one file contains content (such as new properties or a new structure) that is different from the other file, create a new tenantID_vars.yml file, using the sample_tenant_vars.yml file in sas_viya_playbook/samples. Then replace the new tenantID_vars.yml file properties’ values with the equivalent values from the original tenantID_vars.yml file. For more information, see Configure the vars.yml for the Tenant in SAS Viya Administration: Multi-tenancy.

- Run the multi-tenancy playbook to onboard the tenant using the tenantID_vars.yml file from Step 13i on page 195. For more information, see Run the Playbook in SAS Viya Administration: Multi-tenancy.

- Validate the tenant. For more information, see Validate the Tenant in SAS Viya Administration: Multi-tenancy, and also see Validate the Deployment on page 157.

- Complete the deployment on page 177.

14 If you created a 404 redirect for your deployment’s web server in Step 10 on page 195, remove the 404 redirect. See your web server documentation.

15 To access the software changes that were added to your deployment, users must log back in to any running SAS Studio session or CAS session. If the data mining service is deployed, restart all data mining service instances.

16 (Optional) Record the new list of installed software.

- a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

  Run the following command to create a text file that lists the RPM packages:

  ```bash
  sudo rpm -qg SAS > /sas/install/new_viya_rpms.txt
  ```

- b On each machine in your deployment, create a file that lists the SAS yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

  - Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:
    ```bash
    sudo yum grouplist "SAS*" > /sas/install/new_viya_yumgroups.txt
    ```

  - Run the following command to create a text file that lists the RPM packages on SUSE Linux:
    ```bash
    sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt
    ```

    **Note:** If you receive a message such as the following, it can be ignored.

    ```bash
    Repository repository-name is listed more than once in the configuration
    ```

  You can see the differences between the previous and current deployments by comparing the lists of installed software that precedes the update (Step 2 on page 193) and that follows the update.

- c To verify that a specific update was applied, compare the contents of the text file created in Step 16a on page 196 to the packages listed for the specific update. The package list for a specific update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

### Apply Guest Authorization Rule Changes

**Note:** If guest access is enabled on your system and your deployment has upgraded SAS Visual Analytics, perform these steps.

1 As the sas user, log on to any machine in the [CommandLine] host group of the inventory.ini file.
2 Change to the executable directory:

```
cd /opt/sas/viya/home/bin/
```

3 To prepare to apply the guest authorization rule changes, set the SSL_CERT_FILE environment variable and log on using a SAS command-line interface profile. For more information, see *Command-Line Interface: Preliminary Instructions* in *SAS Viya Administration: Command-Line Interfaces*.

**Note:** When you are prompted to log on, log on using a user account that is a member of the SAS Administrators group.

4 To apply the guest authorization rule changes:

```
sas-admin authorization facilitate-guest
```

**User-Related Tasks**

After the SAS Viya deployment has been upgraded, instruct your users to perform additional tasks to make their resources work properly. To perform user-related tasks:

1 Inform users that they must perform the following actions:
   - Clear web browser caches before using the upgraded deployment.
   - Change any entries in web browser bookmarks from SASHome to SASDrive.

2 If you have upgraded SAS Visual Data Mining and Machine Learning, SAS Visual Forecasting, or SAS Visual Text Analytics, perform this step.
   - Inform Model Studio users that they can upgrade their individual projects. Projects need to be upgraded only once.
     - To upgrade SAS Visual Data Mining and Machine Learning projects, refer the users to *Upgrade Considerations* in *SAS Visual Data Mining and Machine Learning*.
     - To upgrade SAS Visual Forecasting projects, refer the users to *Upgrade Considerations* in *SAS Visual Forecasting*.
     - To upgrade SAS Visual Text Analytics projects, refer the users to *Upgrade Considerations* in *Model Studio: SAS Visual Text Analytics*.

3 Inform tenant administrators that their tenants are accessible. Tenant administrators should provide their users with the instructions in Step 1 on page 197 and Step 2 on page 197.

---

**Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya**

**Overview**

An upgrade adds significant feature changes or improvements to your deployed software. To perform an upgrade, you will run the same tools that were run during the initial deployment. You will need a new software order to upgrade your deployed software. An upgrade might require changes to the deployed software’s configuration.

You might determine that your software needs to be upgraded or you might be notified by SAS that upgrades are available.
Add-on products that are present in the order are installed as part of the upgrade process.

Upgrading SAS Viya software requires an outage period. During the upgrade process, all SAS Viya services must be stopped and then restarted. For multi-tenant deployments, the provider administrator should communicate the outage to the tenant customers before the Ansible playbook is run or before any services are manually stopped.

This chapter includes all the steps that are required for the upgrade process, regardless of the version of the source environment, the software installed, and whether the deployment is single-tenant or multi-tenant.

You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Before you start the upgrade, it is recommended that you review all the steps to determine the tasks that are applicable to your deployed software. During your review, identify the tasks that can be performed before a scheduled outage and those that must be performed during a scheduled outage.

If during or after the upgrade process you encounter issues, see “Troubleshooting” on page 285.

Note: Converting a single-tenant deployment to a multi-tenant deployment is not supported.

Note: Converting the LDAP configuration of a multi-tenant deployment from a single LDAP server for all tenants to a separate LDAP server for each tenant, either through an upgrade or an update, is not supported.

Note: Upgrading a SAS Viya deployment on SUSE Linux is not supported in this release because SUSE Linux is new in SAS Viya 3.4.

Note: To deploy or upgrade SAS Data Agent, follow the instructions in SAS Data Agent on Linux: Deployment Guide.

Note: In your SAS Viya deployment, if Kerberos has been enabled for outbound authentication to CAS, you need a new Kerberos Service Principal Name (SPN) and the associated keytab for the SAS Compute Server. See “Configure Kerberos Properties of the SAS Compute Server” on page 209.

Identify Model Studio Projects Derived from SAS Visual Analytics Reports with Partitioning

If the deployment that is being upgraded to SAS Viya 3.4 includes SAS Visual Data Mining and Machine Learning 8.2, perform this task.

Note: If the deployment that is being upgraded to SAS Viya 3.4 is a multi-tenant deployment, inform the tenant administrators to identify the affected projects before the upgrade is performed and that they must notify the project owners in their tenants about the upgrade.

1 Log on to Model Studio as a user that is a member of the SAS Administrators group.

2 Select Table view to view projects. The data source caslib and table names are displayed for each project in the format CASLIB_NAME.TABLE_NAME.

3 Identify each project in the caslib CASUSER that has a table name that begins _VA_.

4 For each project identified, inform the project owner who is indicated in the Created By column to perform the tasks in Upgrading Model Studio 8.2 Projects Derived from SAS Visual Analytics Reports with Partitioning in SAS Visual Data Mining and Machine Learning: Advanced Topics.

Note: You must complete this task for all identified projects before upgrading SAS Viya.

Identify Shared Projects in Model Studio

If the current deployment includes SAS Visual Data Mining and Machine Learning, perform this task.
Note: If the deployment that is being upgraded to SAS Viya 3.4 is a multi-tenant deployment, inform the tenant administrators to identify the affected projects before the upgrade is performed and that they must notify the project owners in their tenants about the upgrade.

When Model Studio is upgraded, any shared project becomes a private project. Before you perform the upgrade, identify which Model Studio projects are shared:

1. Log on to Model Studio as a user who is a member of the SAS Administrators group.
2. In the Projects pane, select Table view.
3. After the upgrade, for each project that is not labeled Private in the Shared With column, notify the project owner who is identified in the Created By column to share the project again with the group in the Shared With column.

Delete the mmLibs Caslib

If the [ModelServices] host group is present in the inventory file from the previous deployment, perform this task.

Before you delete the mmLibs caslib, you must stop the sas-viya-modelrepository-default service. Run one of the following commands, as appropriate, on each machine in the [ModelServices] host group of the inventory.ini file:

- For Red Hat Enterprise Linux 6.x or an equivalent distribution:
  ```
  sudo service sas-viya-modelrepository-default stop
  ```
- For Red Hat Enterprise Linux 7.x or an equivalent distribution:
  ```
  sudo systemctl stop sas-viya-modelrepository-default
  ```

For more information, see Start and Stop Servers and Services in SAS Viya Administration: General Servers and Services.

Note: Do not restart the sas-viya-modelrepository-default service before you perform the upgrade. The sas-viya-modelrepository-default service is restarted during the execution of the Ansible playbook.

To delete the mmLibs caslib, for each CAS server in the deployment:

1. As the sas user, log on to on any machine in the [CommandLine] host group of the inventory.ini file.
2. Change to the executable directory:
   ```
   cd /opt/sas/viya/home/bin/
   ```
3. To set the SSL_CERT_FILE environment variable before you log on to the SAS Administration CLI, run the following command:
   ```
   export SSL_CERT_FILE=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem
   ```
4. To log on to the SAS Administration CLI:
   ```
   sas-admin --sas-endpoint tenant-URL:port auth login
   ```
   Note: Provide a URL and port number that is appropriate for the tenant that you are performing the deployment on.
   When you are prompted to log on, you must log on using a user account that is a member of the SAS Administrators group.
5. To delete the mmLibs caslib, run the following command:
   ```
   Note: Enter the command on a single line. Multiple lines are used here to improve readability.
   ```
Chapter 8 / Managing Your Software

To log off from the SAS Administration CLI, run the following command:

```
sas-admin --sas-endpoint tenant-URL:port auth logout
```

Note: Provide a URL and port number appropriate for the tenant that you are performing the deployment on.

7 If multiple CAS servers have been created, repeat these steps for each instance in the environment.

Note: For each instance, change `instance` in Step 5 on page 199 to the instance name.

8 If you are upgrading a multi-tenant deployment, repeat these steps for each tenant.

Note: For each tenant, change `name` in Step 5 on page 199 to the tenant ID.

Upgrade Using a Mirror Repository

If you are upgrading a SAS Viya 3.2 deployment or a SAS Viya 3.3 deployment that used a mirror repository and you want to use a mirror repository again, you must create a SAS Viya 3.4 mirror repository in order to upgrade to SAS Viya 3.4. For more information, see “Create a Mirror Repository” on page 56.

Create the New SAS Viya 3.4 Playbook

To create the new playbook that you will use to upgrade your SAS Viya deployment to SAS Viya 3.4:

1 When you upgrade SAS Viya, you receive a new Software Order Email (SOE) from SAS. Use your SOE to download the SAS Orchestration CLI.

2 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site. For more information, see “Create a Playbook” on page 58.

3 You must extract the new playbook to a location that is different from the location used in your original playbook. For example, if you extracted your original playbook to `/sas/install/`, you might extract the new playbook to `/sas/upgrade/` instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:
   - To preserve the original `vars.yml` file and the inventory file.
   - To ensure that the directory that contains the playbook correctly reflects what is delivered. If a new playbook is accidentally extracted over an existing playbook, the files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following:

```
tar xf SAS_Viya_playbook.tgz -C /sas/upgrade/
```
**Merge the User-Modified Files**

**Note:** (Optional) You can run an Ansible playbook to automatically perform most of the steps in this task. If the SAS Viya deployment is multi-tenant, you can use the SAS Viya Administration Resource Kit (ARK) merge playbook to merge the vars.yml files of the tenants.

For more information, see “(Optional) Using the SAS Viya Administration Resource Kit” on page 5 and the SAS Viya ARK merge playbook page on GitHub at https://github.com/sassoftware/viya-ark/blob/master/playbooks/merge-playbook/README.md. After you run the playbook, view the output files. To find out the location of the output files, see the README for the merge playbook.

After you run the merge playbook, you can skip all steps except Step 10 on page 203.

Merging user-modified files includes the following actions:

- Compare the existing deployment’s vars.yml, inventory, and ansible.cfg files with the new files for the SAS Viya 3.4 upgrade.
- Find any post-deployment edits in the existing deployment’s files.
- Update the new files with any post-deployment edits in the existing deployment’s files.

You will find or create original, unedited versions of the vars.yml file from the original SAS Viya 3.2 deployment or the SAS Viya 3.3 deployment. You will compare the three vars.yml files described here and edit the vars.yml for the SAS Viya 3.4 upgrade. You will also compare the current inventory file and the new inventory file, the current ansible.cfg file and the new ansible.cfg file, and update the new files with needed changes.

This guide refers to the three types of vars.yml files as follows:

- vars_original.yml — the vars.yml file for the original SAS Viya 3.2 deployment or the SAS Viya 3.3 deployment as it was received from SAS or created by SAS tools.
- vars_current.yml — the vars.yml file for your current SAS Viya 3.2 deployment or the SAS Viya 3.3 deployment that might contain post-deployment edits.
- vars.yml — the vars.yml file for the SAS Viya 3.4 deployment as it was received from SAS or created by SAS tools.

To merge the user-modified files:

1. Locate the existing vars.yml for your current SAS Viya 3.2 or SAS Viya 3.3 deployment that might contain post-deployment edits, and save a copy of the file by renaming the file as vars_current.yml.

2. If you are upgrading a SAS Viya 3.2 deployment to a SAS Viya 3.4 deployment, perform the following steps.
   - Find the SOE that you received for your original SAS Viya 3.2 deployment.
   - Save the vars.yml from the SOE for your SAS Viya 3.2 deployment to a new location. Name it vars_original.yml.

3. If you are upgrading a SAS Viya 3.3 deployment to a SAS Viya 3.4 deployment, perform one of the two following steps.
   - If you have the original and unedited vars.yml that was generated by the SAS Viya 3.3 Orchestration CLI during the original deployment, copy that unedited vars.yml file and save it with the name vars_original.yml.
   - Otherwise, run the SAS Viya 3.3 Orchestration CLI to create a new and unedited SAS Viya 3.3 playbook using the original SOE attachments from the SAS Viya 3.3. Extract the vars.yml from the newly created playbook. Name it vars_original.yml.

4. Compare the file that are currently in use, vars_current.yml, to the file from the SAS Viya 3.4 playbook, vars.yml.
diff vars_current.yml vars.yml

5 Make a list of any variables that are present in vars_current.yml that are not present in vars.yml.

6 Compare the list of variables that you made to vars_original.yml.
   - Any variable in the list that is not present in vars_original.yml is probably a customization that you want to retain. Add the variable to vars.yml.
   - Any variable in the list that is present in vars_original.yml represents a deprecated variable. Do not add these variables to vars.yml.

7 If the original vars.yml file from the deployment that is being upgraded contains a value for the casenv_tenant variable, it must be removed. To remove the registered CAS service, run the following commands:
   
   Note: Enter each command on a single line. Multiple lines are used here to improve readability.
   ```
   cd /opt/sas/viya/home/bin
   ./sas-bootstrap-config --token-file /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
   agent service deregister "cas-{casenv_tenant}-default-http"
   ./sas-bootstrap-config --token-file /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
   agent service deregister "cas-{casenv_tenant}-default"
   
   Note: Do not include casenv_tenant in your new vars.yml. This variable is no longer used.
   ```

8 Merge the inventory file from the previous deployment into the new playbook. If the previous inventory file contains any spaces that are used to indent machine names, do not include the extra spaces.
   
   a Compare the two inventory files to check for additions or changes in the newer set of files. Be sure to evaluate the comments to determine whether the requirements for host groups changed between releases of the software.
   ```
   diff /sas/install/sas_viya_playbook/inventory-file /sas/upgrade/sas_viya_playbook/inventory.ini
   ```
   
   b If the new files contain new content, merge your customized edits from the two original files into the new file. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.
   
   Note: Because of a change in the structure of the host groups between releases, ensure that the inventory.ini file for the upgrade lists the machine in the [Operations] host group in the [CommandLine] host group.
   
   Note: All host groups that are present in the inventory file from the previous deployment must remain on the same machines in the inventory file for the upgrade. New host groups that were not in the previous deployment should be assigned to machines in the current deployment. Review the comments that precede each host group before assigning host groups to machines. For more information, see "Assign the Target Machines to Host Groups" on page 77.
   
   c If you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.

9 Merge the ansible.cfg file from the previous deployment into the new ansible.cfg file.
   
   a Compare the two files to check for additions or changes in the newer file.
   ```
   diff /sas/install/sas_viya_playbook/ansible.cfg /sas/upgrade/sas_viya_playbook/ansible.cfg
   ```
   
   b If the new file contains new content, merge your customized edits from the original file into the new file. If you have any questions, contact SAS Technical Support.
If you created a sitedefault.yml in the previous deployment, copy it to sitedefault_original.yml to use as reference for any future deployments.

Note: Do not edit sitedefault.yml or sitedefault_original.yml.

(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook

You can run an Ansible playbook to automate some of the pre-upgrade steps.

For more information, see "(Optional) Using the SAS Viya Administration Resource Kit" on page 5 and the SAS Viya ARK upgrade playbook page on GitHub at https://github.com/sassoftware/viya-ark/blob/master/playbooks/viya-upgrade/README.md. After you run the playbook, view the summary report. To find out the location of the summary report, see the README for the pre-upgrade playbook.

Update SAS Visual Text Analytics with Yum

If your SAS Viya deployment contains SAS Visual Text Analytics, you must update SAS Visual Text Analytics before you upgrade SAS Viya.

Note: If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading Update SAS Visual Text Analytics in the Results section, skip this section. For more information see "(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook" on page 203.

Here are the requirements for the update:

- SAS Visual Text Analytics 8.2 is included in your SAS Viya deployment.
- SAS Viya is deployed in an environment that uses Red Hat Enterprise Linux or an equivalent distribution.
- You must use yum to perform the update.
  
  Note: Ansible cannot be used to perform the update.

- You must have administrator privileges for the machine on which the update is performed. In addition, your account must have superuser (sudo) access. To verify sudo user privileges, run the command sudo -v or sudo -l.

- If your software is mirrored, you must update the mirror before performing these steps. See “Synchronize the Mirror Repository” on page 182 for more information.

On each machine that is defined in the [VisualTextAnalytics] host group in the inventory file, perform the following steps.

1. To identify the release year for sas-text-gateway, run the following command.

   ```bash
   yum info sas-text-gateway
   ```

2. Locate sas-text-gateway in the results of the command that you ran in Step 1 on page 203.
   
   - If 2017 is not present in the Release field for sas-text-gateway, skip the rest of this section.
   
   - If 2017 is present in the Release field for sas-text-gateway, perform the remaining steps for the current machine.

3. To stop, update, and then start sas-text-gateway, perform one of the following actions, as appropriate:
   
   - For Red Hat Enterprise Linux 6.x or an equivalent distribution, run the following commands:

     ```bash
     sudo service sas-viya-text-gateway-default stop
     sudo yum update sas-text-gateway
     sudo service sas-viya-text-gateway-default start
     ```

   - For Red Hat Enterprise Linux 7.x or an equivalent distribution, run the following commands:

     ```bash
     sudo systemctl stop sas-viya-text-gateway-default
     ```
sudo yum update sas-text-gateway
sudo systemctl start sas-viya-text-gateway-default

For more information, see Start and Stop Servers and Services in SAS Viya Administration: General Servers and Services.

4 To complete the sas-text-gateway update, you must opt in to the assumable groups. As a member of the SAS Administrators group, log on to SAS Model Studio.

http://host-name/ModelStudio

Use the host name of the machine that you assigned to the [httpproxy] host group in the inventory file. For more information about assigning machines, see “Assign the Target Machines to Host Groups” on page 77.

5 At the prompt Do you want to opt in to all of your assumable groups?, select Yes. The display of the project listing indicates the completion of the Visual Text Analytics update.

6 Log off from SAS Model Studio.

Prepare to Upgrade SAS Viya Software

SAS recommends that you create a backup of the deployed software environment before performing an upgrade.

To prepare to upgrade a SAS Viya deployment:

1 (Optional) Record the existing list of installed software before you begin.

Note: If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading Run Deployment Report in the Results section, skip this section. For more information see “(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook” on page 203.

a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

Run the following command to create a text file that lists all the RPM packages:

```
sudo rpm -qg SAS > /sas/install/viya_rpms.txt
```

b On each machine in your deployment, create a file that lists the yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

- Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:

```
sudo yum grouplist "SAS*" > /sas/install/viya_yumgroups.txt
```

- Run the following command to create a text file that lists the RPM packages on SUSE Linux:

```
sudo rpm -qa | grep "sas-" > /sas/install/viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

Repository repository-name is listed more than once in the configuration

2 In the deployment that you are upgrading, find the following line in the original vars.yml that was used for the deployment:

```
casenv_user: casuser
```

Note: If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading Update Custom CAS User in the Results section, skip Step 2 on page 204 and Step 3 on page 205. For more information see “(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook” on page 203.
3 If `casuser` is anything other than the default value `cas`, perform this step. Otherwise, skip this step.

On each machine listed in these three host groups in the inventory file of the deployment that is to be upgraded:
- `sas_caserver_primary`
- `sas_caserver_secondary`
- `sas_caserver_worker`

run the following command.

```
sudo usermod -G sas casuser
```

where `casuser` is the user account name specified in `vars.yml`.

4 Follow the steps that are described in “Perform Linux Tuning” on page 66 on the target machine before starting the upgrade process.

5 System requirements for RAM, CPU, and disk space are likely to change with each SAS Viya release. Verify that your environment meets the requirements that are listed in “System Requirements” on page 21.

### Delete the Default Backup Schedule Job

**Note:** If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading **Delete the Default Backup Schedule Job** in the **Results** section, skip this section. For more information see “(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook” on page 203.

Before performing an upgrade from SAS Viya 3.3 to SAS Viya 3.4, you must delete the existing default backup schedule job. As part of the upgrade, a new default backup schedule job is created.

To delete the default backup schedule job:

1 Log on to SAS Environment Manager as a user that is a member of the SAS Administrators group.

2 Select **Scheduling**.

3 Select the **DEFAULT_BACKUP_SCHEDULE** job.

4 Delete the **DEFAULT_BACKUP_SCHEDULE** job.

### Change the dbmsowner User Password

**Note:** If you are upgrading a programming-only environment, skip this task. Changing the dbmsowner user password is not required when upgrading programming-only environments.

**Note:** If you are upgrading a SAS Viya 3.3 or later deployment, skip this task.

Although SAS Viya 3.2 permitted the use of nonalphanumeric characters in passwords, SAS Viya 3.4 does not support nonalphanumeric characters in passwords. To remove any nonalphanumeric characters that might exist in the dbmsowner account password, you must change the dbmsowner account password before you upgrade your deployment from SAS Viya 3.2 to SAS Viya 3.4.

The script `sds_change_user_pw.sh` changes either the dbmsowner account password or the sas account password for SAS Infrastructure Data Server. It also synchronizes the new password with configuration files and SAS Configuration Server, which is based on Consul.

**Note:** To change the password, you must know the current password. For more information, see **Get Current Passwords** in SAS Viya 3.3 Administration: Infrastructure Servers.

1 Log on to the machine as the SAS install user (sas).
Locate the data server environment variables file, sds_env_var.sh, and record its location.

By default, sds_env_var.sh resides in `/opt/sas/viya/config/etc/sasdatasvrc/postgres/pgpool0`.

Collect the following information, which you will specify when prompted by the sds_env_var.sh script. You will run the script in a later step.

- database user name
- current database password
- new database password

Note: Your password must conform to the password policy. For more information, see Password Policy in SAS Viya 3.3 Administration: Infrastructure Servers.

Using the location of sds_env_var.sh that was noted in step 2, run the script:

Note: Enter the command on a single line. Multiple lines are used here to improve readability.

```bash
/opt/sas/viya/home/libexec/sasdatasvrc/script/sds_change_user_pw.sh -config_path /opt/sas/viya/config/etc/sasdatasvrc/postgres/pgpool0/sds_env_var.sh
```

TIP If you run the script from the directory where it resides, you might see several `cannot open [No such file or directory]` messages. This is a known issue, and you can safely ignore these messages.

Enter the information that you collected in step 3 in response to the script’s prompts.

After you provide values for the prompts, the script connects to SAS Configuration Server and sets all instances of the database user password that it finds. Changes made in the configuration server are synchronized with the proper SAS Infrastructure Data Server configuration files. Finally, the script runs the necessary SQL commands in the data server to set the permissions for the database user.

To verify that your password was successfully changed, connect to the data server’s first database, `postgres`, using the PostgreSQL interactive terminal, `psql`:

```
/opt/sas/viya/home/bin/psql -h data-server-machine-name -U dbmsowner postgres
```

This command connects you to PostgreSQL as the dbmsowner.

When prompted, enter the new password for dbmsowner.

Type `\q` to exit the `psql` interface.

Stop all SAS Viya services in the correct order. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

Start all SAS Viya services in the correct order. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

Verify the Health of SAS Infrastructure Data Server

Note: If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading Verify the Health of the SAS Infrastructure Data Server in the Results section, skip this section. For more information see "(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook" on page 203.

Note: Do not change the PostgreSQL cluster topology during an upgrade.

Perform the following steps:
1 To verify the health of SAS Infrastructure Data Server, perform the task in “Verify SAS Infrastructure Data Server” on page 163.

2 If you are upgrading SAS Viya 3.2, run the following command:

   `sudo /opt/sas/viya/home/bin/psql --version`

   Record the output of this command.

---

**Prepare to Upgrade SAS Event Stream Processing Software**

If you have not purchased a newer version of SAS Event Stream Processing or SAS Event Stream Manager, skip this task.

1 Check for configuration files that have been customized.

   The version of the SAS Event Stream Processing software to which you are upgrading uses a new configuration file, esp-properties.yml. Some customizations that you have made to your installed software will not be used after the upgrade process has completed unless you merge them into the new file.

   On the machine where SAS Event Stream Processing is installed, compare any modified files with the copies that were included with the older version of the software. Run the following commands:

   **Note:** Specify each diff command on a single line. Multiple lines are used here for improved readability.

   ```
   diff -u $DFESP_HOME/etc/connectors.excluded /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/connectors.excluded
   diff -u $DFESP_HOME/etc/esp-logger.xml /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/esp-logger.xml
   diff -u $DFESP_HOME/etc/metatags.conf /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/metatags.conf
   diff -u $DFESP_HOME/etc/security-properties.yml /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/security-properties.yml
   ```

2 Save the results of the diff commands to a file. You will refer to this file when you are ready to merge the configuration changes into esp-properties.yml. For more information, see “Complete SAS Event Stream Processing Upgrade Steps” on page 214.

3 If you installed Streamviewer, stop the Streamviewer process:

   ```
   $DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
   ```

   Replace `host-name` with the host name of the machine where Streamviewer is running.

   Replace `http-port` with the port number that you provided when you started Streamviewer with the start-up script.

4 Stop the metering server:

   ```
   dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
   ```

   Replace `host-name` with the host name of the machine where the metering server is running.

   Replace `http-port` with the port number for the metering server. It uses the default port, 31001.

5 If you are upgrading from SAS Event Stream Manager 4.3, you must remove a Consul entry that refers to the database schema. Later versions of SAS Event Stream Manager do not create this entry.

   Set an environment variable that includes the Consul port, and then run the command to delete the entry. Here is an example:
Note: Enter each command on a single line. Some commands occupy multiple lines to improve readability.

```bash
sudo -u sas -i
source /opt/sas/viya/config/consul.conf
export CONSUL_HTTP_TOKEN='cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token'
/opt/sas/viya/home/bin/sas-bootstrap-config kv delete 'config/esm-service/spring/datasource/schema'
```

### Back Up Your SAS Visual Investigator or SAS Intelligence and Investigation Management Configuration

Note: If your order does not include SAS Visual Investigator or SAS Intelligence and Investigation Management, skip this task.

### Build

This step involves building the deliverables that can execute the remaining upgrade functions. Two types of detection are used to determine the files that are needed for the upgrade process:

- **Automatic Detection**
  - Automatically detects changes in files (both in the deployment code and the configuration data files).

- **Case-Specific Detection**
  - Specifically identifies some set of changes without performing any analysis.

### Back Up

Back up the following services:


- Make a backup of all Consul key/value pairs. See Back Up the SAS Infrastructure Data Server in SAS Visual Investigator 10.5: Administrator’s Guide for more information.

- Back up the entire `/opt/sas/viya` directory structure.

Back up any manual changes that have been made to the deployment environment.

### Process

Stop and then restart Consul before starting any applications that depend on Consul data. In such a case, Consul is started and Consul data is updated to include additions, changes, and deletions.

### Prepare to Upgrade SAS Visual Investigator or SAS Intelligence and Investigation Management

Note: If your order does not include SAS Visual Investigator or SAS Intelligence and Investigation Management, skip this task.

1. On the existing SAS Visual Investigator 10.3.x or later system, or SAS Intelligence and Investigation Management 1.x or later system, stop all services. For more information, see Start and Stop Servers and Services in SAS Viya Administration.
2 If you are upgrading a SAS Visual Investigator 10.3.x or later system, or SAS Intelligence and Investigation Management 1.x or later system, on the CAS controller, perform the following steps:

   a. Open and edit the `/opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig` file as root.
   
   b. Locate the line that contains `useHostToken` and remove the number sign `#` character at the beginning of the line.
   
   c. Locate the line that contains `externalIdent` and remove the number sign `#` character at the beginning of the line.

3 On machines hosting Elasticsearch, back up existing library and configuration components. Run the following commands:

   ```bash
   sudo mv /opt/sas/viya/home/lib/elasticsearch /opt/sas/viya/home/lib/elasticsearch-5.1.1
   sudo mv /opt/sas/viya/config/etc/elasticsearch /opt/sas/viya/config/etc/elasticsearch-5.1.1
   ```

4 Create a new playbook for SAS Visual Investigator 10.5 or SAS Intelligence and Investigation Management 1.3. See “Prepare to Upgrade SAS Viya Software” on page 204.

5 Edit the inventory file. See “Merge the User-Modified Files” on page 201.

   Note: If you are upgrading SAS Visual Investigator or SAS Intelligence and Investigation Management, notice that the format of the list in the `[elasticsearch]` group has changed for SAS Visual Investigator 10.4 and later and SAS Intelligence and Investigation Management 1.2 and later. See “Assign the Target Machines to Host Groups” on page 77.

6 Modify the `vars.yml` file. See “Merge the User-Modified Files” on page 201.

7 Modify the `sitedefault.yml` file. See “Merge the User-Modified Files” on page 201.

**Shut Down All Tenant Services**

Note: If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading **Shut Down All Tenant Services** in the **Results** section, skip this section. For more information see “(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook” on page 203.

If this is a multi-tenant deployment, perform this task.

Shut down the SAS Viya deployment including all tenants in the correct order. For more information, see **Start and Stop Servers and Services** in **SAS Viya Administration**.

**Configure Kerberos Properties of the SAS Compute Server**

If Kerberos has been enabled for outbound authentication to CAS, perform this task. SAS Viya 3.4 requires that the Kerberos Service Principal Name (SPN) and the associated keytab must be configured for the SAS Compute Server. See “Kerberos Requirements” on page 259.

**Upgrade SAS Viya Software**

To upgrade a SAS Viya deployment:

1 Complete the tasks in “**Disk Space Considerations**” on page 26, as appropriate.

   Note: If you ran the SAS Viya ARK pre-upgrade playbook, and the report contains the heading **Disk Space Considerations** in the **Results** section, skip this step. For more information see “(Optional) Run the SAS Viya ARK Pre-Upgrade Playbook” on page 203.
2 It is recommended that you add a 404 redirect in your deployment's web server to prevent users from accessing the deployment before the process is completed. See your web server documentation.

Note: As administrators, use a secondary URL to complete the deployment steps.

3 Install your SAS Viya software using the steps in the installation chapter beginning with "Modify the vars.yml File" on page 81.

4 Run the following command on all machines in the deployment to restore /var/log/sas symlinks:

   sudo yum reinstall -y sas-envesntl sas-envesnt1

5 If you are upgrading SAS Visual Investigator or SAS Intelligence and Investigation Management, skip this step.

   If you removed the CAS service that is associated with the casenv_tenant variable (described in Step 7 of "Merge the User-Modified Files" on page 201), update any bookmarked URLs to remove that value and use cas-shared-default-http instead.


6 To preserve any existing custom themes in SAS Theme Designer, perform the following steps:

   a As a member of either the Application Administrators group or the SAS Administrators group, log on to SAS Theme Designer. See SAS Theme Designer: User's Guide.

   b If there are any themes in the main table of SAS Theme Designer without the SAS logo next to their name, the deployment has custom themes.

   c If the deployment has no custom themes, log off from SAS Theme Designer and skip to Step 7 on page 210.

   d Select the custom theme.

   e Click Unpublish.

   f Click Publish.

   g If the custom theme was the default theme, when the custom theme's status changes to Published, set the custom theme to be the default.

   h Log off from SAS Theme Designer.

7 After the software has been installed, configure these features, as appropriate:

   a Configure the connection to the mail service on page 124.

   b Configure SAS Viya to encrypt the LDAP connection on page 125.

   c Configure high availability in SAS Studio on page 126.

   d If SAS Embedded Process is part of your deployment, upgrade SAS Embedded Process. Details vary according to the data source. For upgrade instructions by data source, see SAS Embedded Process: Deployment Guide.

   e If SAS Hadoop plug-ins are part of your deployment, update SAS Hadoop plug-ins. See “SAS Hadoop Plug-ins” on page 187.

   f If you have added any SAS/ACCESS software to your deployment as part of upgrading to SAS Viya 3.4, see the appropriate topics in "Configure Data Access" on page 131 for instructions to configure the new software.
g Configure any products that you have added to your deployment and that have specific steps in "Post-installation Tasks" on page 119.

h Validate the deployment on page 157.

i If this is a multi-tenant deployment, for each tenant, you must merge the relevant content of the original tenantID_vars.yml file that was used for the original deployment into the format of the tenantID_vars.yml file for the new deployment. For each tenant:

1. Compare the original tenantID_vars.yml file that was used to onboard the tenant with the sample_tenant_vars.yml file in sas_viya_playbook/samples. Perform one of the two following actions:

   Note: If you ran the SAS Viya ARK merge playbook, which was explained in the note at the beginning of "Merge the User-Modified Files" on page 201, go to Step 7i.ii on page 211 to run the multi-tenancy playbook.

   - If the two files contain the same properties and structure, copy the original tenantID_vars.yml file into the sas_viya_playbook directory.
   - If one file contains content (such as new properties or a new structure) that is different from the other file, create a new tenantID_vars.yml file, using the sample_tenant_vars.yml file in sas_viya_playbook/samples. Then replace the new tenantID_vars.yml file properties' values with the equivalent values from the original tenantID_vars.yml file. For more information, see Configure the vars.yml for the Tenant in SAS Viya Administration: Multi-tenancy.

   Note: Do not copy the following properties and their values from the original tenantID_vars.yml file to the new tenantID_vars.yml file. These property names are exact names. If these exact property names are not present in the original tenantID_vars.yml file that was used to deploy SAS Viya, when you have completed merging the contents of the original tenantID_vars.yml file that was used to deploy SAS Viya with the new tenantID_vars.yml file, go to Step 7i.ii on page 211 to run the multi-tenancy playbook.

   - DEPLOYMENT_ID
   - tenant_instance
   - INSTALL_USER
   - INSTALL_GROUP
   - provider_endpoint_scheme
   - provider_endpoint_port

2. Run the multi-tenancy playbook to onboard the tenant using the tenantID_vars.yml file from Step 7i on page 211. For more information, see Run the Playbook in SAS Viya Administration: Multi-tenancy.

3. Validate the tenant. For more information, see Validate the Tenant in SAS Viya Administration: Multi-tenancy, and also see Validate the Deployment on page 157.

j Complete the deployment on page 177.

k See any appendixes that contain information that is relevant to your deployment.

8 If your deployment includes SAS Model Manager, to create the SAS Model Manager configuration definitions, perform the steps in Installation Note 64011.

9 If you are upgrading SAS Viya 3.2, perform this step.

   If the output of the command in Step 2 on page 207 was 9.4.9, perform the steps in Apply the CVE-2017-7547 Security Patch in SAS Viya 3.3 Administration: Infrastructure Servers.
(Optional) Run the SAS Viya ARK Post-Upgrade Playbook

You can run an Ansible playbook to automate some of the post-upgrade steps.

For more information, see “(Optional) Using the SAS Viya Administration Resource Kit” on page 5 and the SAS Viya ARK upgrade playbook page on GitHub at https://github.com/sassoftware/viya-ark/blob/master/playbooks/viya-upgrade/README.md. After you run the playbook, view the summary report. The location of the summary report can be found in the README for the post-upgrade playbook.

Update CAS Server User Formats

Note: If you ran the SAS Viya ARK post-upgrade playbook, and the report contains the heading Adding default user formats in the Results section, skip this section. For more information see “(Optional) Run the SAS Viya ARK Post-Upgrade Playbook” on page 212.

1 If you are upgrading SAS Visual Investigator or SAS Intelligence and Investigation Management, skip this step.
   
   If your SAS Viya deployment is single-tenant, skip this step.
   
   If your SAS Viya deployment is multi-tenant, perform the following steps for each tenant:
   
   a To list the files, run the following command:
      
      ```
      ls /opt/sas/tenant/config/etc/cas/default/casstartup.lua_*
      ```
      
      You specify tenant as the tenant name.
   
   b If a file named casstartup.lua_epoch is listed, where epoch is the UNIX epoch, copy any existing entries in the Default User formats section of casstartup.lua_epoch into /opt/sas/tenant/config/etc/cas/default/casstartup_usermods.lua.
   
   c If no changes were made to /opt/sas/tenant/config/etc/cas/default/casstartup_usermods.lua in Step 1b on page 212, skip this step.
      
      On the machine where the CAS controller is located, run one of the following pairs of commands:
      
      - For Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:
        
        ```
        sudo systemctl stop sas-tenant-cascontroller-default
        sudo systemctl start sas-tenant-cascontroller-default
        ```
        
        where tenant is the tenant name.
      
      - For Red Hat Enterprise Linux 6.x (or an equivalent distribution):
        
        ```
        sudo service sas-tenant-cascontroller-default stop
        sudo service sas-tenant-cascontroller-default start
        ```
        
        You specify tenant as the tenant name.
   
2 On the machine where the CAS controller is located, list the files at the specified location:
   
   ```
   ls /opt/sas/viya/config/etc/cas/default/casstartup.lua_*
   ```
   
3 If a file named casstartup.lua_epoch is listed, where epoch is the UNIX epoch, copy any existing entries in the Default User formats section of casstartup.lua_epoch into /opt/sas/viya/config/etc/cas/default/casstartup_usermods.lua.
   
4 If no changes were made to /opt/sas/viya/config/etc/cas/default/casstartup_usermods.lua in Step 3 on page 212, skip this step.
On the machine where the CAS controller is located, run one of the following pairs of commands:

- For Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:
  ```
  sudo systemctl stop sas-viya-cascontroller-default
  sudo systemctl start sas-viya-cascontroller-default
  ```

- For Red Hat Enterprise Linux 6.x (or an equivalent distribution):
  ```
  sudo service sas-viya-cascontroller-default stop
  sudo service sas-viya-cascontroller-default start
  ```

Add SAS Caslib ACLs

**Note:** If you ran the SAS Viya ARK post-upgrade playbook, and the report contains the heading *SAS-Created caslib controls upgraded* in the *Results* section, skip this section. For more information see "(Optional) Run the SAS Viya ARK Post-Upgrade Playbook" on page 212.

To add SAS caslib ACLs:

1. As the sas user, log on to any machine in the [CommandLine] host group of the inventory.ini file.
2. To set the new required ACLs on SAS caslibs across all CAS servers, run the `add_new_caslib_controls.sh` script with the following command:
   ```
   Note: Enter the command on a single line. Multiple lines are used here to improve readability.
   
   /opt/sas/viya/home/share/deployment/add_new_caslib Controls.sh
   --sas-endpoint "tenant-URL:port"
   
   Note: Provide a URL and port number appropriate for the tenant that you are performing the deployment on.
   ```
3. The script interactively prompts you to log on. You must log on using a user account that is a member of the SAS Administrators group. After you log on, the script runs against all CAS servers in the environment, sets the new ACLs, and then exits.
4. If this is a multi-tenant deployment, repeat these steps for each tenant.
   ```
   Note: Enter the command on a single line. Multiple lines are used here to improve readability.
   
   /opt/sas/viya/home/share/deployment/add_new_caslib Controls.sh
   --sas-endpoint "tenant-URL:port"
   
   Provide a URL and port number that are appropriate for the tenant that you are performing the deployment on.
   ```

Apply Guest Authorization Rule Changes

**Note:** If guest access is enabled on your system and SAS Visual Analytics has been upgraded, perform this task.

**Note:** If you ran the SAS Viya ARK post-upgrade playbook, and the report contains the heading *Enabling Guest Access* in the *Results* section, skip this section. For more information see "(Optional) Run the SAS Viya ARK Post-Upgrade Playbook" on page 212.

1. As the sas user, log on to any machine that is listed in the [CommandLine] host group of the inventory.ini file.
2. Change to the executable directory:
   ```
   cd /opt/sas/viya/home/bin/
   ```
3 To prepare to apply the guest authorization rule changes, set the SSL_CERT_FILE environment variable and log on using a SAS command-line interface profile. For more information, see Command-Line Interface: Preliminary Instructions in SAS Viya Administration: Command-Line Interfaces.

Note: When you are prompted to log on, log on using a user account that is a member of the SAS Administrators group.

4 To apply the guest authorization rule changes:

```
sas-admin authorization facilitate-guest
```

Note: When you are prompted to log on, you must log on using a user account that is a member of the SAS Administrators group.

**Complete SAS Event Stream Processing Upgrade Steps**

Earlier, you were instructed to compare versions of SAS Event Stream Processing configuration files and save a file reflecting the differences between them. Now merge any differences that you found into the new configuration file for SAS Event Stream Processing 6.1, named esp-properties.yml.

1 Open the file that contains the results of the diff commands that you saved in “Prepare to Upgrade SAS Event Stream Processing Software” on page 207.

2 Use your preferred text editor to modify the following file: `/opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/esp-properties.yml`.

3 For the customizations that you found in `connectors.excluded`, locate the `connectors:` section of `esp-properties.yml`. For each connector that you excluded, set the connector’s value to false.

4 For the customizations that you found in `esp-logger.xml`, copy the changes from your diff file to the `logging:` section of `esp-properties.yml`.

5 For the customizations that you found in `metatags.conf`, copy the changes from your diff file to the `meta:` section of `esp-properties.yml`.

6 For the customizations that you found in `security-properties.yml`, copy the changes from your diff file to the `security:` section of `esp-properties.yml`.

7 If your merged changes included enabling SSL but did not include updating the `server_cert_file` parameter, specify `server.pem` for the `server_cert_file` name in that same section of the file.

8 When you have completed the modifications, save `esp-properties.yml`.

9 Clean up unnecessary files in order to avoid file conflicts. Change directories:

```
cd /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default
```

10 Delete the following files:

- `connectors.excluded.rpmsave`
- `esp-logger.xml.rpmsave`
- `metatags.conf`
- `metatags.conf.rpmnew` (an empty file that was created in order to preserve the contents of the actual file)
- `security-properties.yml`
- `security-properties.yml.rpmnew` (an empty file that was created in order to preserve the contents of the actual file)

Note: Backup files are created only during an upgrade if customizations are detected.
11 Complete the tasks that are described in Complete SAS Event Stream Processing Setup on page 149.

Upgrade SAS Visual Investigator or SAS Intelligence and Investigation Management

Note: If your order does not include SAS Visual Investigator or SAS Intelligence and Investigation Management, skip this task.

To complete the upgrade of SAS Visual Investigator or SAS Intelligence and Investigation Management:

1 On a machine that was included in the [consul] host group in the inventory.ini file, remove the key config/application/sas/datasource/initializer/enabled with the following command.

   Note: Enter the command on a single line. Multiple lines are used here to improve readability.

   ```
   sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token kv delete 'config/application/sas/datasource/initializer/enabled'
   ```

2 For multi-tenant environments using custom web server certificates, distribute the /etc/pki/tls/certs/customer.crt and /etc/pki/tls/private/customer.key files from the machine specified in [httpproxy] host group in your inventory.ini file to the equivalent locations on the per-tenant CAS machines. For more information about enabling certificate management, see Manage Certificates in SAS Viya Administration: Data in Motion for more information about managing certificates.

3 Stop and restart all services. For more information, see Start and Stop Servers and Services in SAS Viya Administration: General Servers and Services.

4 If you are upgrading a SAS Visual Investigator 10.3.x or later deployment, skip this step.

   To deregister deprecated services from Consul, run the following command:

   ```
   for sname in SASScenarioAdministrator audit; do
       # determine service ID
       sid=$(/opt/sas/viya/home/bin/sas-bootstrap-config catalog service $sname | jq -r '.items[].serviceID')
       # deregister
       /opt/sas/viya/home/bin/sas-bootstrap-config agent service deregister $sid
   done
   ```

5 Stop and restart all services. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

6 To view the status of the SAS services, run the following command:

   ```
   sudo service sas-viya-all-services status
   ```

   Repeat the command until the Status of all services is Up, continue to the next step.

7 Follow the instructions for manual configuration. See “Post-installation Tasks” on page 119.


9 After onboarding all existing tenants, restart all services. For more information, see Start and Stop Servers and Services in SAS Viya Administration.

10 For each tenant, manually re-index entities. See Re-indexing and Resolving Entities in SAS Visual Investigator 10.5: Administrator’s Guide.

11 With access to the SAS Visual Investigator or SAS Intelligence and Investigation Management application, update the authorization rules for data objects that were previously added within SAS Visual Investigator.
10.3.x or 10.4.x, or SAS Intelligence and Investigation Management 1.1 or 1.2 application. See Alert Entity-Level Security in SAS Visual Investigator 10.5: Administrator’s Guide for additional process details. Without performing this action, SAS Visual Investigator and SAS Intelligence and Investigation Management users and administrators have limited capability to manage data objects and alerts.

After the completion of these steps, you should have a system that has migrated all your SAS Visual Investigator or SAS Intelligence and Investigation Management pre-existing tenants, data, and indexes to the new system.

**Deregister the Launcher Service and Backup Agent Service for the Previous Version**

Note: If you are upgrading SAS Visual Investigator, skip this section.

The upgrade process replaced the launcher service and the backup-agent service from your previous SAS Viya 3.2 or SAS Viya 3.3 environment with the latest release of the software. However, the obsolete services might not be fully deregistered from Consul. To resolve this issue, you must identify and deregister the previous release of the launcher service and the backup-agent services.

**Set Up the Bootstrap Service**

When asked to do so in the sections below, set up the sas-bootstrap-config script by running the following command to retrieve and add the configuration values:

```
source /opt/sas/viya/config/consul.conf
```

**Determine the ID and Service Address for the Launcher Service**

To determine the ID and service address for the launcher service:

1. From any machine in the deployment that is assigned to the [CommandLine] host group, to set up the sas-bootstrap-config script, run the following command:

   ```bash
   source /opt/sas/viya/config/consul.conf
   ```

2. Set the following environment variable to provide Consul token access:

   ```bash
   export CONSUL_HTTP_TOKEN=$(sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token)
   ```

3. Change to the executable directory:

   ```bash
   cd /opt/sas/viya/home/bin/
   ```

4. Determine the ID for the service by running the following command:

   ```bash
   ./sas-bootstrap-config catalog service launcher-server
   ```

   In the output that is returned, there is a value for the serviceID. Identify any serviceID values with a UUID value. An example is "serviceID": "19bf95b6-891c-11e8-a4fb-003ee1cd7c54".

5. For each serviceID instance with a UUID value, also identify the associated serviceAddress. An example is "serviceAddress": "10.111.11.111". Then follow the steps in “Deregister the Launcher Service” on page 216.

   For any other serviceID instances, skip the steps in “Deregister the Launcher Service” on page 216.

**Deregister the Launcher Service**

To deregister the launcher service for each of the IP instances with a UUID value that are shown by the catalog service command above:
1. From the machine identified by the serviceAddress, to set up the sas-bootstrap-config script, run the following command:

```bash
source /opt/sas/viya/config/consul.conf
```

2. Set the following environment variable to provide Consul token access:

```
export CONSUL_HTTP_TOKEN=$(sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens /consul/default/client.token)
```

3. Change to the executable directory:

```bash
cd /opt/sas/viya/home/bin/
```

4. To deregister the service on each IP instance with a UUID value that are shown by the catalog service command, use the output from the serviceID in the preceding command:

```bash
./sas-bootstrap-config agent service deregister UUID
```

5. Repeat steps 1–4 for each of the IP instances with a UUID value.

### Deregister the Backup-Agent Service on CAS Worker Nodes

**Note:** If you are upgrading a single-machine deployment, skip this task.

To identify the CAS worker nodes on which to deregister the service:

1. In the inventory file that was used to perform the upgrade, find the [sas_casserver_worker] host group.

2. Confirm that no machine in the [sas_casserver_worker] host group is present in any other host group in the inventory file that was used to perform the upgrade.

To deregister the backup-agent service, on each CAS worker machine identified in **Step 2 on page 217**, perform the following steps:

1. To set up the sas-bootstrap-config script, run the following command:

```bash
source /opt/sas/viya/config/consul.conf
```

2. Set the following environment variable to provide Consul token access:

```
export CONSUL_HTTP_TOKEN=$(sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens /consul/default/client.token)
```

3. Change to the executable directory:

```bash
cd /opt/sas/viya/home/bin/
```

4. Determine the ID for the service by running the following command:

```bash
./sas-bootstrap-config agent service list | grep -i backup-agent
```

In the output that is returned, there is a value for the ID. An example is "ID": "backup-agent-10-104-21-66".

5. To deregister the IP instance that is shown by the agent service list command, use the output from the ID in the preceding command to deregister the service:

```bash
./sas-bootstrap-config agent service deregister backup-agent-ID
```

Here is an example:

```bash
./sas-bootstrap-config agent service deregister backup-agent-10-104-23-112
```

6. Edit the `/opt/sas/viya/config/etc/viya-svc-mgr/svc-ignore` file to add the following line:
7  Run one of the following commands, as appropriate:
   - For Red Hat Enterprise Linux 6.x or an equivalent distribution:
     ```
     sudo service sas-viya-backup-agent-default stop
     ```
   - For Red Hat Enterprise Linux 7.x or an equivalent distribution:
     ```
     sudo systemctl stop sas-viya-backup-agent-default
     ```

8  Repeat steps 1-7 for each CAS worker machine.

**Deregister Obsolete Services**

*Note:* If you are upgrading SAS Visual Investigator, skip this section.

**Obsolete Services**

As part of the upgrade process, several services from your SAS Viya 3.2 or SAS Viya 3.3 environment were removed or replaced with other services. These obsolete services might not be fully deregistered from Consul, which causes Consul to log repeated messages about these failing services. To resolve this issue, you must deregister the services.

**CAUTION!** Deregister only the following services for SAS Viya 3.2 or SAS Viya 3.3. Removing other services causes other failures.

SAS Viya 3.2 services that are no longer supported:

- SASHome (service name=sashome)
- home (service name=home)
- recipeExecutionProvider (service name=recipeexecutionprovider)
- SASVisualDataBuilder (service name=sasvisualdatabuilder)
- data-preparation-plans (service name=data-preparation-plans)

SAS Viya 3.3 services that are no longer supported:

- SASHome (service name=sashome)
- home (service name=home)
- SASJobMonitor (service name=datamanagementjobmonitor)
- SASProjects (service name=datamanagementprojects)

**Set Up the Bootstrap Service**

To set up the sas-bootstrap-config script, run the following command on every machine in the deployment to retrieve and add the configuration values:

```
source /opt/sas/viya/config/consul.conf
```

**Deregister Health Checks and Services**

To deregister the health checks and services, on every machine, perform the following steps:

1  Set the following environment variable to provide Consul token access:
   ```
   Note: Enter the command on a single line. Multiple lines are used here to improve readability.
   ```
export CONSUL_HTTP_TOKEN=$(sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token)

2 Change to the executable directory:

cd /opt/sas/viya/home/bin/

3 To deregister the health check, determine the checkID for the service by running the following command:

   ./sas-bootstrap-config agent check list | grep -i service-name

   In the output that is returned, there is a value for the checkID. Here is an example: "checkID": "service:service-name-10-123-4-56".

   Repeat this step for every service listed for your source environment, in the preceding Caution. If the noted services are not found, the services have already been deregistered and you can skip the rest of this section.

   Note: If you have multiple services running on multiple machines, more than one entry is returned from the preceding command. Each checkID corresponds to the IP address of the machine where the service is running. Each checkID value should be deregistered.

4 To deregister each of the health checkID IP instances that are shown by the agent check list command, use the information from the checkID value in the preceding command output:

   ./sas-bootstrap-config agent check deregister --id service:service-name-10-123-4-56

5 To deregister the service, find out the ID for the service by running the following command:

   ./sas-bootstrap-config agent service list | grep -i service-name

   In the output that is returned, there is a value for the ID. Here is an example: "ID": "service-name-10-123-4-56".

   Repeat this step for every service listed for your source environment in the Caution note above.

   Note: If you have multiple services running on multiple machines, more than one entry is returned from the preceding command. Each ID corresponds to the IP address of the machine where the service is running. Each ID value should be deregistered.

6 To remove each of the IP instances that are shown by the agent service list command, use the output from the ID in the preceding command to deregister the service:

   ./sas-bootstrap-config agent service deregister service-name-10-123-4-56

Example: Remove Data Preparation Plans Services

For example, to remove data-preparation-plans:

1 To list the data-preparation-plans services in Consul that are required to be deregistered:

   ./sas-bootstrap-config agent check list | grep -i data-preparation-plans

   "checkID": "service:data-preparation-plans-10-123-4-56",
   "serviceID": "data-preparation-plans-10-123-4-56",

2 To deregister the health check, use the output from the checkID value in the preceding command:

   ./sas-bootstrap-config agent check deregister --id service:data-preparation-plans-10-123-4-56

3 To list the data-preparation-plans services in Consul that are required to be deregistered:

   ./sas-bootstrap-config agent service list | grep -i data-preparation-plans

   "data-preparation-plans-10-123-4-56": {
   "ID": "data-preparation-plans-10-123-4-56",

4 To deregister the service, use the output from the ID in the preceding command:
Viewing Percentage-Based Metrics in SAS Data Profile

Note: If your deployment does not include SAS Data Preparation, skip this section.

If you are using SAS Data Preparation, percentage-based metrics in SAS Data Profile reports, such as Percent Null and Percent Blank, are not available in SAS Viya 3.4 after upgrading from SAS Viya 3.3. To view these metrics, rerun SAS Data Profile on your data set after upgrading to SAS Viya 3.4.

Import an Existing QKB

Note: If your order does not include SAS Data Quality, skip these steps. This information is applicable only if your order contains SAS Data Quality.

Your upgrade includes the latest version of the QKB for Contact Information.

- If a QKB was already on your CAS server, it is still in place. However, to use the latest version of the QKB for Contact Information, you can import it to your CAS server by following the instructions at Import a QKB in SAS Viya Administration.
- If a QKB was not previously installed on your CAS system, this latest version has been imported for you and is ready to use.

Back Up and Make a List of the New Deployment

To back up and make a list of the new deployment:

1. If you created a 404 redirect in your deployment’s web server in Step 2 on page 210, remove the 404 redirect. See your web server documentation.

2. Immediately after the upgrade, perform a backup of your deployment for possible recovery purposes.
   - Do not wait for the scheduled backup to run.
   - Do not use the backups of SAS Viya that were performed before this upgrade to use for recovery.

3. (Optional) Record the new list of installed software.
   
   Note: If you ran the SAS Viya ARK post-upgrade playbook, and the report contains the heading Run Deployment Report in the Results section, skip this Step 3 on page 220. For more information see “(Optional) Run the SAS Viya ARK Post-Upgrade Playbook” on page 212.

   a. On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

      Run the following command to create a text file that lists the RPM packages:

      ```
sudo rpm -qg SAS > /sas/install/new_viya_rpms.txt
      ```

   b. On each machine in your deployment, create a file that lists the SAS yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 60.

      To create a text file that lists the yum groups on Red Hat Enterprise Linux:

      ```
sudo yum grouplist "SAS*" > /sas/install/new_viya_yumgroups.txt
      ```

      To create a text file that lists the RPM packages on SUSE Linux:
sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt

Note: If you receive a message such as the following, it can be ignored.

Repository repository-name is listed more than once in the configuration

You can see the differences between the previous and current deployments by comparing the lists of
installed software that precedes the upgrade (Step 1 on page 204) and that follows the upgrade.

To verify that a specific update was applied, compare the contents of the text file that was created in Step
3a on page 220 to the packages that are listed for the specific update. The package list for a specific
update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at

Preserve Access Controls for Database Caslibs

Note: If you are performing an upgrade of SAS Viya 3.3, skip this section.

This section is needed only if you are upgrading SAS Viya 3.2.

SAS Viya 3.3 includes changes to how access controls for database caslibs are stored. Path-based caslibs that
use a directory as a data source are unaffected.

**TIP** If you have database caslibs in a SAS Viya 3.2 deployment, but you do not have access controls or
you do not need to preserve the access controls, you can drop the caslib and add an identical caslib with
the same name. Using the same name enables SAS Visual Analytics reports to remain valid. Perform the
drop and add after upgrading SAS Viya.

Immediately after the upgrade, the change to the access controls prevents data access with caslibs that use
databases. Specifically, the server does not perform the loadTable, save, columnInfo, and fileInfo actions. Here
is the error message that is displayed:

```
Caslib caslib-name is from an old release and cannot be used. Create a new caslib and copy
the access controls to it.
```

To preserve existing access controls, after SAS Viya has been upgraded and after SAS Cloud Analytic Services
(CAS) has been started with the new release, perform the following steps for each database caslib:

1. Temporarily add a caslib with the same data source as the original caslib. Use a temporary name such as
   OdbclibNew. Use the same database server, port, schema, and so on, as applicable for the data source.

2. From a SAS session that is running the upgraded deployment of SAS Viya or SAS 9.4M5, run the
   copyObjects action. In this example, odbclib is the existing caslib and temp-odbcllib is the new temporary
   caslib.

   **CAUTION!** The copyObjects action is a restricted action that is designed for this specific purpose. Do not attempt
to use it for any other purpose elsewhere.

   ```
   cas;
   proc cas;
   accessControl.assumeRole / adminRole="data";
   run;

   accessControl.copyObjects /
   fromObjectSelector={caslib="odbclib" objType="caslib"}
   toObjectSelector={caslib="temp-odbcllib" objType="caslib"};
   run;
   ```

3. Drop the original caslib. In the example, odbclib is the caslib.
Add a caslib that is identical to the original caslib. Use the same name as the original caslib and the same data source information.

5 Restore the access controls from the temporary caslib to the newly created caslib:

```sas
    cas;
    proc cas;
    accessControl.copyObjects /
      fromObjectSelector={caslib="temp-odbclib" objType="caslib"}
      toObjectSelector={caslib="odbclib" objType="caslib"};
    run;

    accessControl.dropRole / adminRole="data";
    run;
```

6 Drop the temporary caslib.

User-Related Tasks

After the SAS Viya deployment has been upgraded, instruct your users to perform additional tasks to make their resources work properly. To perform user-related tasks:

1 Inform users that they must perform the following actions:
   - Clear web browser caches before using the upgraded deployment.
   - Change any entries in web browser bookmarks from SASHome to SASDrive.

2 If you have upgraded SAS Visual Data Mining and Machine Learning, SAS Visual Forecasting, or SAS Visual Text Analytics, perform this step.
   Inform Model Studio users that they can upgrade their individual projects. Projects must be upgraded only once.
   - To upgrade SAS Visual Data Mining and Machine Learning projects, refer the users to Upgrade Considerations in SAS Visual Data Mining and Machine Learning.
   - To upgrade SAS Visual Forecasting projects, refer the users to Upgrade Considerations in SAS Visual Forecasting.
   - To upgrade SAS Visual Text Analytics projects, refer the users to Upgrade Considerations in Model Studio: SAS Visual Text Analytics.

3 Inform tenant administrators that their tenants are accessible. Tenant administrators should provide their users with the instructions in Step 1 on page 197 and Step 2 on page 197.
Uninstalling SAS Viya

What deploy-cleanup Does

When you use the deploy-cleanup command described in the following sections, it performs these actions:

1. Stop all SAS services.
2. Remove all SAS RPMs.
3. Delete any remaining SAS .pid files.
4. Delete the entitlement_certificate.pem and SAS_CA_Certificate.pem files.

The deploy-cleanup command renames the /opt/sas/viya directory to /opt/sas/viya_epoch. Also, the /opt/sas/spre directory is renamed as /opt/sas/spre_epoch.

The uninstallation does not remove the customized script that you received with your SOE, and it does not remove any users that have been set up.

Use deploy-cleanup

1. Offboard and delete any tenants that have been added to your deployment. See Offboard a Tenant in SAS Viya Administration: Multi-Tenancy.
2. Ensure that you are at the top level of the playbook in the sas_viya_playbook directory.
3. Here is the basic syntax for the command to run the playbook and deploy the software:
   
   ```
   ansible-playbook deploy-cleanup.yml
   ```
   
   Add an option based on the password requirements for the user ID that performs the command, using Table 9.1 on page 224.
Table 9.1  Command Options Based on Password Requirements

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td>use the command as written</td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td><code>--ask-become-pass</code></td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td><code>--ask-pass</code></td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td><code>--ask-pass --ask-become-pass</code></td>
</tr>
</tbody>
</table>

Here is an example of the deploy command that requires both sudo and SSH passwords:

```bash
ansible-playbook deploy-cleanup.yml --ask-pass --ask-become-pass
```

4 If you have added CAS servers to your deployment, you will need to run a similar command for each CAS server:

```bash
ansible-playbook -i CAS-server-inventory-file-name deploy-cleanup.yml -e "@CAS-server-vars-file-name"
```

Add an option based on the password requirements for the user ID that performs the command, using Table 9.1 on page 224.

The deploy-cleanup command leaves a few running processes that should be removed individually.

1 httpd remains on your system because other software might be using it. If no other software is using httpd, you can stop its processes and remove it by running the following command:

```bash
yum remove httpd
```

2 The epmd process remains running on your system as an artifact of SAS Message Broker. To stop the process:

   a List all active processes by running the following command:
      ```bash
      ps -A
      ```
   b In the results, find “epmd” in the far right column, and then locate its process ID (PID) in the far left column.
   c Remove the epmd process by running the following command:
      ```bash
      kill process-ID-for-epmd
      ```

3 The sas-configuration-cli process could remain running on your system. To stop the process perform the following steps on every machine in your deployment:

   a List all active processes by running the following command:
      ```bash
      ps -A
      ```
   b In the results, find “sas-configuration-cli” in the far right column, and then locate its process ID (PID) in the far left column. If “sas-configuration-cli” is not listed, then you can move on to the next machine.
   c Remove the sas-configuration-cli process by running the following command:
      ```bash
      kill process-ID-for-sas-configuration-cli
      ```

4 Depending on how you deployed your software, the deployment process may have altered or replaced your certificate. Use the following steps to determine if you want to replace the certificate as part of the process of uninstalling your software.
a Ensure that the system refers to the appropriate SSL configuration file by performing this command:

```
set OPENSSL_CONF=location-of-openssl.conf-file
```

b Run the following command to determine the distinguishing characteristics of the certificate being used:

```
openssl x509 -noout -in "location-of-certificate-file" -subject -startdate -issuer
```

The output will look like this:

```
subject= /C=US/O=Self-Signed Certificate/CN=mymachinename.mycompany.com
notBefore=Jul  5 19:30:10 2018 GMT
issuer= /C=US/O=Self-Signed Certificate/CN=mymachinename.mycompany.com
```

c Determine if the following conditions are met in the output:

- The value for O= (the organization) in the subject= line is Self-Signed Certificate.
- The values in the subject = line and the issuer= line are identical.
- The value in the notBefore= line is the time and date when you deployed your software.

If all of these conditions are met, then the certificate was deployed during the deployment of your SAS Viya software, and the certificate should be removed to return the system to its pre-deployment condition.

---

### Uninstall SAS Embedded Process

If your software deployment includes SAS Embedded Process, uninstall it using the instructions in SAS Embedded Process: Deployment Guide.

---

### Uninstall SASHDAT Plug-ins

If your software deployment includes CAS SASHDAT Access to HDFS, uninstall it using the instructions at “Uninstalling SAS Plug-ins for Hadoop” on page 268.

---

### Uninstall a Mirror Repository

If your deployment includes a mirror repository and you want to remove it as well, you can run a basic Linux command to do so. Because all the files of the mirror repository are contained in a single directory, use the following command to remove the mirror repository:

```
sudo rm -rf path-to-mirror-repository
```

If you did not change the default location of the SAS Mirror Manager log when you deployed your software, you should also remove the log from /local/share/mirrormgr in the home directory of the install user.
Overview

SAS Viya uses High Availability (HA) PostgreSQL as the SAS Infrastructure Data Server. By default, when you use the instructions in “Installation” on page 73, Ansible deploys HA PostgreSQL as a single node on a single machine. However, HA PostgreSQL supports other topologies. This appendix describes those topologies and explains how to use Ansible to deploy them.

CPS PostgreSQL

If your software order contains SAS Assortment Planning, SAS Demand Planning, SAS Financial Planning, or SAS Markdown Optimization, your deployment includes a single-machine, single-node instance of CPS PostgreSQL. However, you can deploy CPS PostgreSQL on multiple nodes. If you modify the default CPS PostgreSQL variables, you need to set up the HA PostgreSQL variables in the inventory.ini file. You can then deploy the HA PostgreSQL instances using Ansible.
PostgreSQL deployment, be aware that variable names and default port ranges for CPS PostgreSQL are different from those used by HA PostgreSQL. Here is a comparison of the variable names and port values:

<table>
<thead>
<tr>
<th>HA PostgreSQL Variable</th>
<th>CPS PostgreSQL Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgpoolc</td>
<td>cpspgpoolc</td>
</tr>
<tr>
<td>sasdatasvrc</td>
<td>cpsdatasvrc</td>
</tr>
<tr>
<td>postgres</td>
<td>cpspostgres</td>
</tr>
<tr>
<td>Default ports 5430–5439</td>
<td>Default ports 5440–5449</td>
</tr>
</tbody>
</table>

Note: HA PostgreSQL and CPS PostgreSQL clusters can be co-located or distributed.

### HA PostgreSQL Topologies

The standard PostgreSQL deployment with SAS Viya consists of one PGPool and one PostgreSQL data node. All data connection and database requests are routed through PGPool. You connect to PGPool just as you would connect to PostgreSQL, using standard database connectors. With SAS Viya we also have the ability to deploy High Availability PostgreSQL, a clustered database containing one PGPool and one or more data nodes. One data node is designated as a primary and all others are standby nodes. Replication happens in real time to keep the data nodes in sync. All write requests are routed to the primary data node by PGPool; read requests can be distributed across all data nodes, allowing for higher performance. In the event that the primary data node is lost, PGPool will automatically promote a standby node to primary and reestablish replication from the new primary to the remaining standby data nodes.

The PostgreSQL deployment for Viya also supports the ability to deploy multiple database clusters as part of a single deployment. For example, you might want to put your microservices on one cluster while having dedicated clusters for your server. Each cluster is considered a service and each member of that cluster (PGPool and data nodes) is considered a node within that service. A cluster can be deployed on the same machines as other clusters or on their own machines.

A cluster can be deployed in four possible configurations:

- **Single Node** - One PGPool and one data node on the same machine. This is the default deployment for SAS Viya.
- **Horizontal** - Each data node on a separate machine.
- **Vertical** - All data nodes on a single machine.
- **Hybrid** - A combination of horizontal and vertical where there are at least two machines within the cluster and there is more than one data node on a machine within the cluster.

For multinode deployments, PGPool node can be co-located with data nodes or deployed on its own machine. Note that co-locating nodes on a machine provides increased read throughput but also increases the risk of node loss should that machine become unavailable.

The following table demonstrates how nodes can be distributed in the multinode topologies.
HA PostgreSQL Topologies

<table>
<thead>
<tr>
<th>Cluster Configuration</th>
<th>Server</th>
<th>Port</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>Server 1</td>
<td>5432</td>
<td>Primary</td>
</tr>
<tr>
<td></td>
<td>Server 2</td>
<td>5432</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 3</td>
<td>5432</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 4</td>
<td>5432</td>
<td>Standby</td>
</tr>
<tr>
<td>Vertical</td>
<td>Server 1</td>
<td>5532</td>
<td>Primary</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5533</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5534</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5535</td>
<td>Standby</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Server 1</td>
<td>5632</td>
<td>Primary</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5633</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 2</td>
<td>5632</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 2</td>
<td>5633</td>
<td>Standby</td>
</tr>
</tbody>
</table>

The two files in your playbook that must be revised for HA PostgreSQL are the inventory.ini and vars.yml files. The inventory.ini file (the inventory) identifies roles that will be placed on each machine. The vars.yml file specifies the settings for pgpoolc and sasdatasvrc that are used to define the HA PostgreSQL instance or instances desired on each of those machines. Because the definitions for HA PostgreSQL come from synchronized edits of inventory.ini and vars.yml, those edits should be done in tandem to ensure alignment.

When you revise the vars.yml file for your cluster, the following variables under INVOCATION_VARIABLES should be modified:

**pgpoolc**
- PCP_PORT: the PCP port for the PGPool instance
- PGPOOL_PORT: the PGPool port. This is the primary port that all database connections will go to.
- SANMOUNT: the location where the data files will be placed
- SERVICE_NAME: the unique name that you assign to your cluster. This value must match the SERVICE_NAME of all sasdatasvrc nodes that will attach to this pgpoolc in the cluster.

**sasdatasvrc**
- NODE_NUMBER: the sequential node identifier starting at 0
- NODE_TYPE: P for primary or S for standby. There can be only one primary per cluster.
- PG_PORT: The PostgreSQL database port. PGPool talks to the database on this port. Clients use the PGPOOL_PORT.
- SANMOUNT: the location where the data files will be placed
- SERVICE_NAME: the unique name that you assign to your cluster. This value must match the SERVICE_NAME of the pgpoolc node that this data node will attach to in the cluster.
Set Up a Horizontal Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Specify the Machines in the Deployment” on page 75. Then assign the machines to the host groups as described at “Assign the Target Machines to Host Groups” on page 77. Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

This is an example of a completed inventory.ini file that includes the horizontal cluster described in the table above, with PGPool being on the same machine as the first HA PostgreSQL node. (The example shows only the entries related to HA PostgreSQL):

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file= ~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file= ~/.ssh/id_rsa
deployTarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file= ~/.ssh/id_rsa
deployTarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file= ~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1

[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
```

Edit the vars.yml File

Open the vars.yml file in the playbook. In the INVOCATION_VARIABLES section, fill in the variables appropriate for your deployment. Using the horizontal cluster example from the table above, this section would describe four machines, one of which would have a subsection for pgpoolc and all having subsections for sasdatasvrc. This is what that section would look like when filled out for our example:

```
INVOCATION_VARIABLES:
deployTarget1:
  pgpoolc:
    - PCP_PORT: '5431'
    - PGPOOL_PORT: '5430'
    - SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
    - SERVICE_NAME: postgres
  sasdatasvrc:
    - NODE_NUMBER: '0'
    - NODE_TYPE: P
    - PG_PORT: '5432'
    - SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
    - SERVICE_NAME: postgres
```
deployTarget2:
sasdatasvrc:
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5432'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres
deployTarget3:
sasdatasvrc:
  - NODE_NUMBER: '2'
    NODE_TYPE: S
    PG_PORT: '5432'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres
deployTarget4:
sasdatasvrc:
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5432'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres

Note that the machine listed under [pgpoolc] in the inventory.ini file is the only one that has pgpoolc variables in the vars.yml file. Because all four machines will have HA PostgreSQL nodes on them, all four machines have sasdatasvrc variables in the vars.yml file. The nodes are numbered from 0 to 3, and node 0, on the deployTarget1 machine, is the primary node. The entry for SANMOUNT: will read the deployment and use the location of the SAS_CONFIG_ROOT directory and append the directory name.

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 115 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Vertical Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Specify the Machines in the Deployment” on page 75. Then assign the machines to the host groups as described at “Assign the Target Machines to Host Groups” on page 77. Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

This is an example of a completed inventory.ini file that includes the vertical cluster described in the table above, with PGPool being on the same machine as the HA PostgreSQL nodes. (The example shows only the entries related to HA PostgreSQL):

deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file= ~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
...
[sasdatasvrc]
deployTarget1
...
Edit the vars.yml File

Open the vars.yml file in the playbook. In the INVOCATION_VARIABLES section, fill in the variables appropriate for your deployment. Using the vertical cluster example from the table above, this section would describe a single machine, with a subsection for pgpoolc and four subsections for the sasdatasvrc nodes. This is what that section would look like when filled out for our example:

```yaml
# Multiple invocation definitions
INVOCATION_VARIABLES:
deployTarget1:
  pgpoolc:
    - PCP_PORT: '5531'
    - PGPOOL_PORT: '5530'
    - SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    - SERVICE_NAME: postgres
  sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5532'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5533'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5534'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '3'
      NODE_TYPE: S
      PG_PORT: '5535'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
```

Note that the machine is described with a single pgpoolc entry and four sasdatasvrc entries. The nodes are numbered from 0 to 3, and node 0 is the primary node. The PORT entries all show a different port in order to avoid any conflict. The entry for SANMOUNT: will read the deployment and use the location of the SAS_CONFIG_ROOT directory and append the directory name.

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 115 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Hybrid Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Specify the Machines in the Deployment” on page 75. Then assign the machines to the host groups as described at “Assign the Target Machines to Host Groups” on page
Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

This is an example of a completed inventory.ini file that includes the hybrid cluster described in the table above, with PGPool being on the same machine as two of the HA PostgreSQL nodes. (The example shows only the entries related to HA PostgreSQL):

```
[pgpoolc]
deployTarget1
...

[sasdatasvrc]
deployTarget1
deployTarget2
...
```

**Edit the vars.yml File**

Open the vars.yml file in the playbook. In the INVOCATION_VARIABLES section, fill in the variables appropriate for your deployment. Using the hybrid cluster example from the table above, this section would describe a two machines, with a subsection for pgpoolc on the same machine as two of the sasdatasvrc nodes. This is what that section would look like when filled out for our example:

```
# Multiple invocation definitions
INVOCATION_VARIABLES:
deployTarget1:
  pgpoolc:
    - PCP_PORT: '5631'
      PGPOOL_PORT: '5630'
    SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
    SERVICE_NAME: postgres
  sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5632'
      SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5633'
      SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
      SERVICE_NAME: postgres

deployTarget2:
  sasdatasvrc:
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5632'
      SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '3'
      NODE_TYPE: S
      PG_PORT: '5633'
      SANMOUNT: '{SAS_CONFIG_ROOT}/data/sasdatasvrc'
```
Note that the first machine has a single pgpoolc entry and two sasdatasvrc entries. The nodes are numbered from 0 to 3, and node 0 is the primary node. The PORT entries for either machine show a different port in order to avoid any conflict. The entry for SANMOUNT: will read the deployment and use the location of the SAS_CONFIG_ROOT directory and append the directory name.

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 115 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up Multiple Clusters

Modify inventory.ini and vars.yml Files

This example consists of four machines and has the following clusters:

- a single-node cluster with pgpoolc and sasdataservc on a machine named deployTarget1
- a horizontal cluster with pgpoolc on deployTarget1 and a sasdatasrvc node on each machine
- a vertical cluster with pgpoolc on deployTarget3 and all the sasdatasrvc nodes on deployTarget4
- a hybrid cluster with pgpoolc on deployTarget1, two sasdatasrvc nodes on deployTarget2, and two more sasdatasrvc nodes on deployTarget3

This is how the inventory.ini file should be modified for this HA PostgreSQL deployment (the entries related to HA PostgreSQL are shown):

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
```

...  

```
[pgregpool]
deployTarget1
deployTarget3
deployTarget4
...  

[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
...  
```

This is how the INVOCATION_VARIABLES section of the vars.yml file would be filled out:

```
# Multiple invocation definitions
INVOCATION_VARIABLES:
deployTarget1:
  pgpoolc:
    - PCP_PORT: '5431'
PPOOL_PORT: '5430'
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
```
SERVICE_NAME: postgres_hybrid
- PCP_PORT: '5461'
PGPOOL_PORT: '5460'
SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres

sasdatasvrc:
- NODE_NUMBER: '0'
  NODE_TYPE: P
  PG_PORT: '5452'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
  SERVICE_NAME: postgres_horizontal

deployTarget2:
sasdatasvrc:
- NODE_NUMBER: '0'
  NODE_TYPE: P
  PG_PORT: '5432'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
  SERVICE_NAME: postgres_hybrid
- NODE_NUMBER: '2'
  NODE_TYPE: S
  PG_PORT: '5433'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
  SERVICE_NAME: postgres_hybrid

deployTarget3:
pgpoolc:
- PCP_PORT: '5441'
PGPOOL_PORT: '5440'
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres_vertical

sasdatasvrc:
- NODE_NUMBER: '1'
  NODE_TYPE: S
  PG_PORT: '5432'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
  SERVICE_NAME: postgres_hybrid
- NODE_NUMBER: '3'
  NODE_TYPE: S
  PG_PORT: '5433'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
  SERVICE_NAME: postgres_hybrid
- NODE_NUMBER: '2'
  NODE_TYPE: S
  PG_PORT: '5452'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
  SERVICE_NAME: postgres_horizontal

deployTarget4:
pgpoolc:
  - PCP_PORT: '5451'
  - PGPOOL_PORT: '5450'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal
sasdatasvrc:
  - NODE_NUMBER: '0'
    NODE_TYPE: P
    PG_PORT: '5442'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5443'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '2'
    NODE_TYPE: S
    PG_PORT: '5444'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5445'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5452'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal

Note: If you are deploying multiple clusters, one of the PGPoools must be named postgres, and each PGPool name must be unique across clusters. In addition, each cluster must contain one sasdatasvrc node with a NODE_TYPE of P.

Configure Services to the Clusters

By default, all microservices connect to the HA Postgres cluster that is named postgres. You can configure individual services to use additional HA Postgres clusters (if they exist) by adding service-specific sections to the sitedefault.yml file.

1 If you have not already copied and renamed the sitedefault.yml file, locate the sitedefault_sample.yml file on the Ansible controller machine. If you used the recommended location for uncompressing your playbook, the file is located at /sas/install/sas_viya_playbook/roles/consul/files/
   sitedefault_sample.yml. Make a copy of sitedefault_sample.yml and name the copy sitedefault.yml.

2 Open the sitedefault.yml file.

3 At the end of the existing file and at the same indentation level as application, add the following content:

```yaml
config:
  application:
    ...
    service-name
    sas:
      database:
```
databaseServerName: cluster-name
spring.datasource.password: ${sas.database.cluster-name.password}

The value for cluster-name must exactly match the SERVICE_NAME value for the cluster in the INVOCATION_VARIABLES section in the vars.yml file.

The following example shows the addition of the authorization service that uses an HA Postgres cluster named postgres-horizontal:

    config:
      application:
        ...
        authorization:
          sas:
            database:
              databaseServerName: postgres-horizontal
              spring.datasource.password: ${sas.database.postgres-horizontal.password}

4 Save and close the sitedefault.yml file.

Deployment Logs

Each PGPool node and HA PostgreSQL data node has its own set of directories for logging. The logs for PGPool are located at

    /opt/sas/viya/config/var/log/sasdatasvrc/postgres/pgpool0/

The log for the HA PostgreSQL nodes is located at

    /opt/sas/viya/config/var/log/sasdatasvrc/postgres/node0/

Verify the Deployment

The deployment performs a verification of the HA PostgreSQL cluster before it completes. This verification first confirms that connections can be made to PGPool and to all data nodes, and then runs queries on all of the nodes. The verification also performs write and delete operations to ensure that values that are written to or removed from the primary data node are replicated to all of the standby nodes in a multinode deployment.

The verification log is called sds_status_check_date-timestamp.log. It can be found in the pgpool log folder of each cluster. The fastest way to determine whether your HA PostgreSQL deployment was successful is to read the verification log.
Appendix 2

Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop

Supported Hadoop Distributions ................................................................. 239
Deployment Tasks for Hive and Spark Access ............................................. 239
Pre-deployment Hadoop and Spark Tasks .................................................. 240
  Pre-deployment Checklist ........................................................................ 240
  Security .................................................................................................... 241
Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop .... 241
  Requirements to Deploy JAR Files on the CAS Controller ...................... 241
  Install the Hadoop JAR Files on the CAS Controller ............................. 242
  Set up Hive LLAP Support on a Hortonworks Hadoop 3 Cluster ............... 244
  Set Up Multiple Hadoop Versions for Multiple Hadoop Servers ............. 246
  Verify SAS Data Connector to Hadoop ................................................... 246
Verify SAS/ACCESS Interface to Spark ..................................................... 247

Supported Hadoop Distributions

Before you set up Hadoop, you must make sure that your Hadoop distribution is supported by SAS Viya. For details, see “Hadoop Requirements” on page 35.

Deployment Tasks for Hive and Spark Access

For Hive access, perform the following tasks:

1. Perform the pre-deployment tasks. For more information, see “Pre-deployment Hadoop and Spark Tasks” on page 240.

2. Deploy the Hadoop JAR files. For more information, see “Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop” on page 241.

3. If you are using the SAS Data Connect Accelerator for Hadoop, deploy the SAS Embedded Process. For more information, see “Overview of the In-Database Deployment Package for Hadoop” on page 250.
Pre-deployment Checklist

Before you install SAS Viya software that interacts with Hadoop and Hive and Spark, it is recommended that you verify your Hadoop environment. Use the following checklist:

- Ensure that you have configured SAS Data Connector to Hadoop and, if required, SAS Data Connect Accelerator for Hadoop. For details, see “SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop” on page 89.

- Understand and verify your Hadoop user authentication.

- Have sudo access on the NameNode.

- Enable the HDFS user with Write permission to the root of HDFS.
  The HDFS user home directory, /user/user-account, must exist and must have drwxrwxrwx permissions for the HDFS user directory. If you deploy the SAS Embedded Process, this user account is used to manually deploy in the SAS Embedded Process: Deployment Guide.

- Verify that the Hadoop master node can connect to the Hadoop slave nodes using passwordless SSH. For more information, see the Linux manual pages about ssh-keygen and ssh-copy-id.

- Understand and verify your security setup.
  - Verify that you can use your defined security protocol to connect from your client machine, which is outside of the SAS Viya environment) to your Hadoop cluster.
  - It is highly recommended that you enable Kerberos or another security protocol for data security. If your cluster is secured with Kerberos, you must obtain a Kerberos ticket. You also must have knowledge of any additional security policies.
  - For clusters that have Kerberos security enabled, verify that you have a valid ticket on the node on which the Hive2 service is running.

- Gain working knowledge about the Hadoop distribution that you are using (for example, Cloudera or Hortonworks).
  You also need working knowledge about the HDFS, MapReduce 2, YARN, and Hive services. For more information, see the Apache website or the vendor’s website.
  For MapR, you must install the MapR client. The installed MapR client version must match the version of the MapR cluster that SAS Viya connects to. For more information, see the MapR documentation.

- Verify that the HCatalog, HDFS or Hive, MapReduce, and YARN services are running on the Hadoop cluster. SAS Viya software uses these various services, which ensure that the appropriate JAR files are located during the configuration.

- For the Hive server:
  - Identify the machine on which the Hive server is running. If the Hive server is not running on the same machine as the NameNode, note the server and port number of the Hive server for future configuration.
  - Know the host name of the Hive server and the host name of the NameNode.

- For MapReduce:
  - Know the location of the MapReduce home directory.
  - Request permission to restart the MapReduce service.
  - Verify that you can run a MapReduce job successfully.
Security

Kerberos Security

SAS Data Connector to Hadoop can be configured for a Kerberos ticket cache-based logon authentication by using MIT Kerberos 5 Version 1.9.

Note: SAS Viya must be configured for pluggable authentication module (PAM) support.

If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you must manually add the Java Cryptography Extension local_policy.jar file to each instance of JAVA_HOME on the Hadoop cluster. If you are located outside the United States, you must also manually add the US_export_policy.jar file. The addition of these files is governed by the United States import control restrictions.

If you are using the Oracle JRE or the IBM JRE, the appropriate JAR file must also replace the existing local_policy.jar file and the US_export_policy.jar file in your JRE location. This location is typically the JAVA_HOME/jre/lib/security/ directory. You can obtain the appropriate file from the Oracle website or the IBM website.

It is recommended that you back up the existing local_policy.jar file and the US_export_policy.jar file first in case they need to be restored.

If you are using the OpenJDK, the files do not need to be replaced.

JDBC Read Security for Hive

SAS Data Connector to Hadoop can access Hadoop data through a JDBC connection to Hive. Depending on your release of Hive, Hive might not implement Read security. A successful connection from SAS Viya can allow Read access to all data that is accessible to the identity that is used to access the Hive server. Hive can be secured with Kerberos. SAS Data Connector to Hadoop supports Kerberos 5 Version 1.9 or a later release.

Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop

Requirements to Deploy JAR Files on the CAS Controller

- Hadoop cluster manager:
  - host name and port number
  - credentials (account name and password)
- Hive service host name
- For Ansible installation, SSH credentials of the Linux account that has access to the machine on which the Hive service has been installed and is running.
- If your deployment includes MapReduce users from Windows clients, after you run the hadooptracer.py script, you must follow the instruction to edit the mapred-site.xml file and set the mapreduce.app-submission.cross-platform property to true.
Install the Hadoop JAR Files on the CAS Controller

Overview of Installing the Hadoop JAR Files
You can install the Hadoop JAR files by using either of the following methods:
- Ansible
- Manual steps

Install the Hadoop JAR Files with Ansible
1. Ensure that Python, strace, and wget have been installed on the Hadoop cluster from the package repositories for your Linux distribution.
2. On the Ansible controller, run the following command in order to enable passwordless SSH:
   
   ```
   ssh-copy-id Hive-server-machine
   
   Note: If the SAS install user is different from the user that is set up on the Hadoop cluster, you might want to specify the ssh-copy-id specifically for that user for the Hadoop cluster.
   ```
3. Edit the inventory.ini file to add the Hadoop cluster machine to the list of target references at the beginning of the file. For more information, see “Specify the Machines in the Deployment” on page 75.
4. In the inventory.ini file, add a machine target for the Hadoop Hive node. Also, beneath the list of target machines, add the [hadooptracer1 ] group. Add the new Hadoop machine target to the new group.
   ```
   hadoop-cluster ansible_host=ansible-host ansible_ssh_user=user
   [hadooptracer1]
   hadoop-cluster
   
   For more information see “Assign the Target Machines to Host Groups” on page 77.
   ```
5. Open the all file that is located in the directory where you unpacked the Ansible playbook:
   ```
   sudo vi /sas/install/group_vars/all
   ```
6. Modify the following variables using the descriptions in the comments in the all file:
   ```
   Note: If the directory does not exist, it is automatically created when you run the Ansible playbook.
   ```
   ```
   hadoop_conf_home: /opt/sas/viya/config/data/hadoop
   lib_folder_name: lib
   conf_folder_name: conf
   
   Note: These directories correspond to a JAR file path of /opt/sas/viya/config/data/hadoop/lib
   and to a configuration file path of /opt/sas/viya/config/data/hadoop/conf.
   ```
7. Save and close the all file.
8. To launch the playbook, run the following command:
   ```
   ansible-playbook utility/hadooptracer-launch.yml
   ```
   Ansible will copy files to the Hadoop cluster node and then to the CAS controller and SAS programming nodes. If the Ansible playbook is run with a cluster where Spark2 is installed, the Spark2 JAR files are placed in the /opt/sas/viya/config/data/hadoop/lib/spark directory and the Spark2 configuration files are placed in the /opt/sas/viya/config/data/hadoop/conf directory.
(Cloudera and Hortonworks distributions only) Verify that the required Hadoop JAR files are successfully collected:

`ansible-playbook utility/hadooptracer-validation.yml`

**Note:** When the Ansible playbook is run with a cluster that does not contain a Spark2 installation, a warning that some tasks were skipped might be displayed.

### Obtain and Install the Hadoop Tracer Script Manually

The script is organized into two parts. The first part is a list of the Hadoop services that the script is tracing to a new `driver.json` file. The second part is the remainder of the code. This division allows the list of Hadoop services and additional JAR files to be modified manually without having to make Python code changes.

If the `driver.json` file is moved or removed from its default location, the Hadoop tracer script will fail with an error message indicating that the file is missing.

To run the Hadoop tracer script successfully:

- For Ansible, ensure that the user running the script has passwordless SSH access to all of the Hadoop services.
- Ensure that Python 2.6 or later and strace are installed. Contact your system administrator if these packages are not installed on the system.
- Ensure that the user running the script has authorization to issue HDFS and Hive commands.
- If Hadoop is secured with Kerberos, obtain a Kerberos ticket for the user before running the script.
- If you want to pull one or more JAR files that are not included in the output of the hadoop tracer script, modify the `driver.json` file by adding the JAR files to the `ExtraJarFiles` property at the bottom of the file. Here is an example of the addition of a new JAR file named `xyz-service.jar` to the `ExtraJarFiles` property:

```json
"ExtraJarFiles": ["jline-*","jruby-complete-*","hive-warehouse-connector-assembly-*","xyz-service-*"]
```

To obtain and run the Hadoop tracer script:

1. On the Hadoop server, create a temporary directory to hold a ZIP file that you download later. An example would be `/tmp/sas/hadooptracer_19w34`.
2. Download the hadooptracer.zip file from the following FTP site to the directory that you created in step 1. `ftp.sas.com/techsup/download/blind/access/hadooptracer.zip`
3. Using a method of your choice (for example, PSFTP, SFTP, SCP, or FTP), transfer the ZIP file to the Hive node on your Hadoop cluster.
4. Unzip the file. The `hadooptracer.py` and `driver.json` files are included in this ZIP file.
5. Change permissions on the `hadooptracer.py` file to include the Execute permission.
   ```bash
   chmod 755 ./hadooptracer.py
   ```
6. Run the tracer script.
   ```bash
   python ./hadooptracer.py --filterby=latest --postprocess --jsonfile ./driver.json
   ```

**Note:** The `filterby=latest` option ensures that if duplicate JAR or configuration files exist, the latest version is selected. If you want to pull the necessary JAR files without filtering, use `filterby=none`, or omit the `filterby=` option.

**TIP** The `postprocess` option ensures that the `hdp.version` tokens are replaced. It is highly recommended that you run the tracer script with this option. This option is ignored for Cloudera clusters.
This tracer script performs the following tasks:

- Pulls the necessary Hadoop JAR and configuration files and places them in the `/tmp/jars` directory and the `/tmp/sitexmls` directory, respectively.
- Creates a `hadooptracer.json` file in the `/tmp` directory. If you need a custom path for the JSON output file, use this command instead:

  ```python
  python ./hadooptracer.py -f /your-path/hadooptracer.json
  ```

- Creates a log in the `/tmp/hadooptracer.log` directory.

  **Note:** Some error messages in the console output for `hadooptracer.py` are normal and do not necessarily indicate a problem with the JAR and configuration file collection process. However, if the files are not collected as expected or if you experience problems connecting to Hadoop with the collected files, contact SAS Technical Support and include the `hadooptracer.log` file.

7. On the SAS client machine, create two directories to hold the JAR and configuration files. An example would be the `/opt/sas/hadoopfiles/lib` and `/opt/sas/hadoopfiles/conf` directories.

8. Using a method of your choice (for example, PSFTP, SFTP, SCP, or FTP), copy the files in the `/tmp/jars` and `/tmp/sitexmls` directories on the Hadoop server to the directories on the SAS client machine that you created in step 7.

  **Note:** If you connect to the Hadoop server with an HTTP REST API, you do not need the Hadoop JAR files on the SAS client machine.

### Set up Hive LLAP Support on a Hortonworks Hadoop 3 Cluster

To set up LLAP (Hive Live Long and Process) support on a Hortonwork Hadoop 3 Cluster:

1. Ensure that you have completed the update process. For details, see "Updating Your SAS Viya Software" on page 181.

2. Ensure that you have run the hadooptracer script. For details, see "Overview of Installing the Hadoop JAR Files" on page 242.

3. Locate the `hive-site.xml` file:

   - If you ran the Ansible playbook, check the `/opt/sas/viya/config/data/hadoop/conf` directory.
   - If you ran the script manually, check the `/tmp/sitexmls` directory.

4. To modify the `hive-site.xml` file, you must have this information:

   - The name of the host at which the `hive.llap.daemon.service.hosts` file resides
   - The value of the `llap.if.hs2.connection` property

   Obtain the information from your Hadoop administrator or by accessing the Ambari Server of the Hortonworks Hadoop 3 cluster.

5. You must update two properties. These properties are available on the Ambari Server client. You might need to obtain this information from your Hadoop administrator.

   - Enter the Advanced hive-interactive-site (hive.llap.daemon.service.hosts) property:
6  Locate the following section in the hive-site.xml file:

```xml
<property>
  <name>llap.if.hs2.connection</name>
  <value></value>
</property>
```

Remove the comment tags from the beginning and end of the hive-site.xml file:

```
<-- and -->
```

7  Add the Advanced hive-interactive-site value from Step 5 from Ambari to the hive.llap.daemon.service.hosts property. Here is an example:

```xml
<property>
  <name>hive.llap.daemon.service.hosts</name>
  <value>@llap0</value>
</property>
```

8  Add the HIVESERVER2 INTERACTIVE JDBC URL value from Ambari to the llap.if.hs2.connection property. Here is an example:

```xml
<property>
  <name>llap.if.hs2.connection</name>
  <value>jdbc:hive2://
  </value>
</property>
```
Save the file.

If your SAS Viya 3.4 playbook was updated during the hot fix update process, a message will be printed to the console as a reminder that the Hive service is llap-enabled.

**Set Up Multiple Hadoop Versions for Multiple Hadoop Servers**

If you have multiple Hadoop servers that run different Hadoop versions:

- The version of the JAR files in the hadoopJarPath directory on the CAS server must match the version of the JAR files on the Hadoop server to which CAS connects.
- Each CAS session can connect only to Hadoop clusters of one configured hadoopJarPath version.
- Separate concurrent CAS sessions can independently connect to different versions of Hadoop clusters.

To support multiple Hadoop versions:

1. Create and populate separate directories with version-specific Hadoop JAR files for each Hadoop version.
2. Start separate CAS sessions, and point each separate CAS session to one of the hadoopJarPath versions.

Upgrading your Hadoop server version might involve multiple active Hadoop versions. The same multi-version instructions apply.

**Verify SAS Data Connector to Hadoop**

To verify that the software has been successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Hadoop LIBNAME statement:

   ```
   option set=SAS_HADOOP_CONFIG_PATH="path-to-config-files";
   option set=SAS_HADOOP_JAR_PATH="path-to-jar-files";
   libname hdplib hadoop server="hive-hadoop-host-name" user=user-ID password=user-password;
   ```

   **Note:** Do not use the USER= argument or PASSWORD=argument if your Hadoop cluster is secured by Kerberos.

   For more information, see [LIBNAME Statement for the Hadoop Engine](#).

   If SAS/ACCESS to Hadoop was successfully deployed, the execution of the LIBNAME statement will return results without error.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Hadoop:

   ```
   caslib hdlib datasource=(srctype="hadoop", dataTransferMode="serial", username="user-ID", server="hadoop-host-name", hadoopjarpath="path-to-jar-files", hadoopconfigdir="path-to-config-files", schema="Hadoop-schema-name");
   proc casutil;
   list files incaslib="hdlib";
   run;
   ```

   If the data connector was successfully deployed, the results are the names of the tables in Hive.
If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to Spark

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Spark (on SAS Viya).

To verify that SAS Data Connector to Spark was successfully deployed:

1. From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Spark LIBNAME statement:

   ```sas
   option set=SAS_HADOOP_CONFIG_PATH="path-to-config-files";
   option set=SAS_HADOOP_JAR_PATH="path-to-jar-files";
   libname spklib sasiosk server="spark-thrift-server-host-name" user=user-ID password=user-password;
   ```

   **Note:** Do not use the USER= argument or PASSWORD=argument if your Hadoop cluster is secured by Kerberos.

   For more information, see [LIBNAME Statement for the Spark Engine](#).

   If SAS/ACCESS to Spark was successfully deployed, the execution of the LIBNAME statement returns results without error.

2. From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Spark:

   ```sas
   caslib hdlib datasource=(srctype="spark", dataTransferMode="serial", username="user-ID", server="spark-thrift-server-host-name", hadoopjarpath="path-to-jar-files", hadoopconfigdir="path-to-config-files", schema="Hadoop-schema-name");
   proc casutil;
   list files incaslib="hdlib";
   run;
   ```

   **Note:** For the hadoopjarpath variable, the Spark JAR files must be specified as follows:

   - For Hortonworks, the location of the Spark JAR files must be added to the end of the path:
     ```bash
     /opt/sas/viya/config/data/hadoop/lib:/opt/sas/viya/config/data/hadoop/lib/spark
     ```
   - For Cloudera, the location of the Spark JAR files must be added to the beginning of the path:
     ```bash
     /opt/sas/viya/config/data/hadoop/lib/spark:/opt/sas/viya/config/data/hadoop/lib
     ```

   If the data connector was successfully deployed, the results are the names of the tables in the Spark Thrift server.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.
Appendix 3

Hadoop Deployment: Configuring SAS In-Database Technologies

Hadoop Prerequisites

The SAS In-Database Technologies for Hadoop on SAS Viya includes SAS Data Connect Accelerator for Hadoop and the SAS Embedded Process for Hadoop. The installation of the in-database deployment package for Hadoop involves writing a configuration file to HDFS and deploying files on all the data nodes. The following tasks can occur automatically, depending on your Hadoop and HDFS permissions.

- The CAS controller and each CAS worker node must have an IP address that can be routed to externally from the SAS Embedded Process nodes.
- The master node requires passwordless SSH for the slave nodes.
- Writing the configuration file requires Write permission to HDFS.

Note: The SAS Embedded Process is installed on all YARN Node Manager nodes. The installation script detects the Hadoop cluster topology and installs the SAS Embedded Process on all YARN Node Manager nodes. The script also installs the SAS Embedded Process on the host node on which you run the script. You can choose any node on the cluster to be the SAS Embedded Process master node, as long as the node from which you run the script has passwordless SSH for the slave nodes.

Note: The passwordless SSH user must also have Read, Write, and Execute permissions on the installation directory. The directory structure of the nodes must match that of the installation directory.
Note: The SAS Embedded Process installation script creates the configuration file on the local file system in the `EPInstallDir/conf` folder. If you run the SAS Embedded Process installation script with `sudo` access, the script automatically creates and writes the configuration file to HDFS during the initial deployment. If you do not run the script with `sudo` access, you must manually copy the configuration file to HDFS.

The parallel loading of data using SAS Data Connect Accelerator for Hadoop requires a massively parallel processing (MPP) system.

Overview of the In-Database Deployment Package for Hadoop

SAS In-Database Technologies for Hadoop (on SAS Viya) includes SAS Data Connect Accelerator and SAS Embedded Process for Hadoop.

Note: SAS In-Database Technologies for Hadoop also includes SAS In-Database Technologies for Spark. SAS In-Database Technologies for Spark contains the Data Connect Accelerator and SAS Embedded Process for Spark. These Spark components are automatically deployed when you deploy SAS Embedded Process for Hadoop.

SAS Embedded Process is a SAS server process that is deployed in and runs within your data source to read and write data. SAS Embedded Process contains macros, run-time libraries, and other software that is deployed on your data source. SAS Embedded Process is delivered with any SAS 9.4 software or SAS Viya software that requires it.

Only one version of SAS Embedded Process is required for SAS 9.4, SAS Viya, or both. Only one version should be active and you should always use the latest version that is available.

SAS Embedded Process for SAS 9.4: Install and Deploy on SAS Viya

Note: Starting with SAS 9.4M6, the delivery, installation, and deployment process for SAS Embedded Process has changed. The SAS Embedded Process deployment instructions are in SAS Embedded Process: Deployment Guide.

You should always install the latest release of SAS Embedded Process.

SAS Embedded Process is now being delivered via a link in your Software Order Email (SOE), and the deployment process is identical for both SAS 9.4 and SAS Viya.

When you license a SAS product that requires SAS Embedded Process, you will receive two SOEs:

- One SOE is for your SAS 9.4 or SAS Viya software offering.
- Another SOE is for SAS Embedded Process.

(Optional) Deploy TLS Certificates

If you are using a SAS Data Connect Accelerator, the data that is transferred between the data provider and the CAS server is not encrypted by default. However, SAS Viya supports TLS encryption between the data provider and the CAS server. When SAS Viya 3.4 is deployed, TLS is enabled and configured on the CAS server (server side). The deployment process provides a default level of encryption for data in motion. Options are set in the...
vars.yml file and are defined in the casconfig_deployment.lua file. These settings enable data connector encryption and specify the location of the TLS private key and the password.

However, you must take additional steps to enable encryption on the data provider. The prerequisites and the process for enabling TLS encryption on the data provider are different for each data provider. The first step is to deploy the TLS certificates across all nodes in the cluster.

1. On the CAS controller machine, locate the TLS certificates in the trustedcerts.pem file in the `/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/` directory.

2. Copy the trustedcerts.pem file to the `security/certs` directory on the Hadoop master node.

3. To complete the deployment of TLS encryption, you also must update a `dcsecurity.properties` file. Copy both the .pem file and the `dcsecurity.properties` file to all nodes on the CAS server. For more information about how to complete the deployment, see Encrypt Data Transfer When Using the SAS Data Connect Accelerator (Linux Full Deployment) in Encryption in SAS Viya: Data in Motion.

Verify SAS Data Connect Accelerator for Hadoop

The information in this section is applicable only if you ordered SAS Data Connect Accelerator for Hadoop.

To verify that the software has been successfully deployed:

1. From SAS Studio, edit and run the following SAS code:
   ```sas
   cas mysession;
   caslib hivelib datasource=(srctype="hadoop" server="server name"
   dataTransferMode="parallel"
   hadoopconfigdir="path-to-directory-containing-Hadoop-config-files-collected-with-hadoop_extract.sh"
   hadoopjarpath="path-to-directory-containing-Hadoop-JAR-files-collected-with hadoop_extract.sh")
   proc casutil;
   load casdata=hivelib.casdata="Hive table to load" casout="CAS table name"
   incaslib="hivelib";
   run;
   ```

   The SAS code loads the table from Hive into CAS.

2. Check the log to verify that the load was successful. As an option, to view the data, run the following code to assign a libref to the caslib and view the table with PROC PRINT:
   ```sas
   libname caslib cas caslib=hivelib;
   proc print data=caslib.CAS table name; run;
   ```

   If SAS Data Connect Accelerator and the SAS Embedded Process have been successfully deployed, the results show the appearance of data in the table. If you do not see the data, you should perform the configuration steps again.

Verify SAS Data Connect Accelerator for Spark

The information in this section is applicable only if you ordered SAS Data Connect Accelerator for Spark.

To verify that the software has been successfully deployed:

1. From SAS Studio, edit and run the following SAS code:
cas mysession;
caslib sparklib datasource=(srctype="spark" server="server name"
dataTransferMode="parallel"
hadoopconfigdir="path-to-directory-containing-Hadoop-config-files-collected-with-hadoop_extract.sh"
hadoopjarpath="path-to-directory-containing-Hadoop-JAR-files-collected-with_hadoop_extract.sh");
proc casutil;
load casdata="Spark table to load" casout="CAS table name" incaslib="sparklib";
run;

The SAS code loads the table from Spark into CAS.

2 Check the log to verify that the load was successful. As an option, to view the data, run the following code to assign a libref to the caslib and view the table with PROC PRINT:

    libname caslib cas caslib=hivelib;
    proc print data=caslib.CAS table name; run;

If SAS Data Connect Accelerator and the SAS Embedded Process have been successfully deployed, the results show the appearance of data in the table. If you do not see the data, you should perform the configuration steps again.

Add the YARN Application CLASSPATH for MapR

Two configuration properties specify the YARN application CLASSPATH: yarn.application.classpath and MapReduce.application.classpath. If you do not specify the YARN application CLASSPATH, MapR uses the default CLASSPATH. However, if you specify the MapReduce application CLASSPATH, the YARN application CLASSPATH is ignored. The SAS Embedded Process for Hadoop requires both the YARN application CLASSPATH and the MapReduce application CLASSPATH.

To ensure that the YARN application CLASSPATH exists, you must manually add the YARN application CLASSPATH to the yarn-site.xml file. Without the manual definition in the configuration file, the MapReduce application master fails to start a YARN container.

Here is the default YARN application CLASSPATH for Linux:

    $HADOOP_CONF_DIR,
    $HADOOP_COMMON_HOME/share/hadoop/common/*,
    $HADOOP_HDFS_HOME/share/hadoop/hdfs/*,
    $HADOOP_YARN_HOME/share/hadoop/yarn/*,
    $HADOOP_YARN_HOME/share/hadoop/yarn/lib/*

Here is the default YARN application CLASSPATH for Windows:

    %HADOOP_CONF_DIR%,
    %HADOOP_COMMON_HOME%/share/hadoop/common/*,
    %HADOOP_COMMON_HOME%/share/hadoop/common/lib/*,
    %HADOOP_HDFS_HOME%/share/hadoop/hdfs/*,
    %HADOOP_HDFS_HOME%/share/hadoop/hdfs/lib/*,
    %HADOOP_YARN_HOME%/share/hadoop/yarn/*,
    %HADOOP_YARN_HOME%/share/hadoop/yarn/lib/*

Note: On MapR, the YARN application CLASSPATH does not resolve the symbols or variables that are included in pathnames such as $HADOOP_HDFS_HOME.
Performance Tuning for the SAS Embedded Process

Overview of Performance Tuning Properties

You can tune the SAS Embedded Process by editing certain properties in the ep-config.xml file or the mapred-site.xml file, as appropriate.

The ep-config.xml file is created when you install the SAS Embedded Process. By default, the file is located in the /sas/ep/config/ep-config.xml directory.

The mapred-site.xml file is copied to the client machine when the hadoop_extract.sh script was run. By default, the file is located in the directory that you specified for the hadoop.client.config.filepath variable.

You can change the values of the following properties:

- **trace levels**
  For more information, see “Change the Trace Level” on page 254.

- **the number of the SAS Embedded Process MapReduce tasks per node**
  For more information, see “Specify the Number of MapReduce Tasks” on page 254.

- **the maximum amount of memory in bytes that the SAS Embedded Process is allowed to use**
  For more information, see “Specify the Amount of Memory That the SAS Embedded Process Uses” on page 254.

- **the buffers for input data**
  For more information, see “Specify the Number of Input Buffers and an Optimal Buffer Size” on page 255.

- **the number of concurrent input reader threads**
  Each reader thread takes a file split from the input splits queue, opens the file, positions itself at the beginning of the split, and starts reading the records. Each record is stored in a native buffer that is shared with the DS2 container. When the native buffer is full, it is pushed to the DS2 container for processing. When a reader thread finishes reading a file split, it takes another file split from the input splits queue. The default number of input threads is 3.

  ```xml
  <property>
    <name>sas.ep.input.threads</name>
    <value>number-of-input-threads</value>
  </property>
  ```

- **the number of output writer threads**
  The sas.ep.output.threads property is used to change the number of output writer threads. The SAS Embedded Process super writer technology improves performance by writing output data in parallel and producing multiple parts of the output file per mapper task. Each writer thread is responsible for writing one part of the output file. The default number of output threads is 2.

  ```xml
  <property>
    <name>sas.ep.output.threads</name>
    <value>number-of-output-threads</value>
  </property>
  ```

**TIP** To apply any change that you make to the cluster, you must restart the node managers.
Change the Trace Level

You can modify the level of tracing by changing the value of the sas.ep.server.trace.level property in the ep-config.xml file. The default value is 4 (TRACE_NOTE).

```xml
<property>
    <name>sas.ep.server.trace.level</name>
    <value>trace-level</value>
</property>
```

The `trace-level` represents the level of trace that is produced by the SAS Embedded Process. Here are the `trace-level` values:

**Note:** Trace options can produce a significant volume of output. If trace options are not required for troubleshooting or monitoring, set the `trace-level` value to 0.

0  TRACe_OFF
1  TRACe_FATAL
2  TRACe_ERROR
3  TRACe_WARN
4  TRACe_NOTE
5  TRACe_INFO
10 TRACe_ALL

Specify the Number of MapReduce Tasks

You can specify the number of the SAS Embedded Process MapReduce Tasks per node by changing the sas.ep.superreader.tasks.per.node property in the ep-config.xml file. The default number of tasks is 6.

```xml
<property>
    <name>sas.ep.superreader.tasks.per.node</name>
    <value>number-of-tasks</value>
</property>
```

Specify the Amount of Memory That the SAS Embedded Process Uses

The SAS Embedded Process is managed by the Hadoop MapReduce framework. Load balancing and resource allocation are managed by YARN. Adjust the YARN container limits to change the amount of memory that the SAS Embedded Process. For information about how CAS uses memory, see “Memory” in SAS Cloud Analytic Services: Fundamentals.
Specify the Number of Input Buffers and an Optimal Buffer Size

You can specify the number of buffers in which to store input data and the optimal size of one input buffer. You specify this information by changing the `sas.ep.input.buffers` and `sas.ep.optimal.input.buffer.size` properties in the `mapred-site.xml` file.

The default value of the `sas.ep.input.buffer` property is 4 buffers. The default value of the `sas.ep.optimal.input.buffer.size` property is 1MB.

```xml
<property>
  <name>sas.ep.input.buffers</name>
  <value>number-of-buffers</value>
</property>

<property>
  <name>sas.ep.optimal.input.buffer.size.mb</name>
  <value>buffer-size-in-MB</value>
</property>
```
About CAS SASHDAT Access to HDFS

Some SAS Viya applications that rely on Hadoop will also require SAS Plug-ins for Hadoop. Supported Hadoop distributions that are modified with SAS Plug-ins for Hadoop enable CAS to write SASHDAT file blocks evenly.
across the HDFS file system. This even distribution provides a balanced workload across the machines in the cluster and enables SAS Viya analytic processes to read SASHDAT tables very quickly.

**Supported Hadoop Distributions**

Before you set up Hadoop, ensure that your Hadoop distribution is supported by SAS Viya. For more information, see Support for Databases in SAS Viya 3.4.

**Overview of Deployment Tasks for HDFS for Existing Hadoop Clusters**

During the SAS Viya installation, your CAS software was deployed in one of the following ways:

- the CAS controller and workers were deployed to the nodes on your Hadoop cluster. For an overview of this deployment scenario, see “Hadoop Integration: CAS SASHDAT Access to HDFS” on page 14.
- the CAS controller and the CAS workers were deployed to nodes that are not part of the Hadoop cluster. For an overview of this deployment scenario, see “Remote Access to HDFS” on page 16.

To configure your existing Hadoop cluster:

1. Perform the Hadoop pre-deployment tasks. For more information, see “Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters” on page 258.
2. Deploy SAS Plug-ins for Hadoop on the Hadoop NameNode and to each DataNode. For more information, see “Deploying SAS Plug-ins for Hadoop” on page 262.
3. Verify CAS SASHDAT Access to HDFS. For more information, see “Verifying CAS SASHDAT Access to HDFS” on page 267.

**Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters**

Here are the requirements for existing Hadoop clusters that are configured for use with the CAS server.

- Verify that the following CAS environment variables are set correctly for your Hadoop environment: cas.colocation, HADOOP_NAMENODE, and HADOOP_HOME.
  - During the SAS Viya installation, values for the CAS environment variables are set in the vars.yml file before you run the playbook. After you run the playbook, the settings for the CAS environment variables are stored in the /opt/sas/viya/config/etc/cas/default/cas.settings file and the /opt/sas/viya/config/etc/cas/default/casconfig_deployment.lua file.
  - For more information about updating CAS environment variables, see SAS Cloud Analytic Services: Overview in SAS Viya Administration.
- Each machine in the cluster must be able to resolve the host name of all the other machines in the cluster.
- The time must be synchronized across all machines in the cluster.
- For Cloudera 5 only, all machines that are configured for the CAS server must be in the same role group.
For Secure Shell (SSH), review the requirements and perform the appropriate tasks in “Kerberos Requirements” on page 259.

For Kerberos, review the requirements and perform the appropriate tasks in “Kerberos Requirements” on page 259.

---

### Review the Passwordless Secure Shell Requirements

Here are the passwordless Secure Shell (SSH) requirements:

- To support Kerberos, enable the GSSAPI authentication methods in your implementation of SSH.
  
  **Note:** If you are using Kerberos, see “Configure Passwordless SSH to Use Kerberos and GSSAPI” on page 261.

- Passwordless SSH is required for connections from all CAS machines to all machines in the Hadoop cluster. Passwordless SSH is required for the user account that runs the CAS server and for the user accounts that run CAS sessions. By default, the user account that runs the CAS server and CAS sessions is the cas user. Also, passwordless SSH is set up by default.

- If you are running a co-located deployment and use a subset of the machines, passwordless SSH is required for the user account that runs the CAS session. By default, the user account is the cas user, and all CAS nodes are set up with passwordless SSH. Passwordless SSH is also required for the user account that is used to start the CAS server.

- Passwordless SSH is required when a block of data exists on a Hadoop node that exists outside of the Hadoop nodes in the CAS session.

---

### Kerberos Requirements

#### Overview

SAS Viya does not directly interact with Kerberos. Instead, SAS Viya relies on the underlying operating system and the APIs to handle the requests for tickets, the management of ticket caches, and the authentication of users. For an overview of Kerberos and SAS Viya, refer to Kerberos Authentication in SAS Viya Administration: SAS Logon Manager.

#### Host Requirements

For Kerberos, on the CAS server, the `/etc/hosts` file contains the host names of the machines in the cluster. Each host name is specified in this format:

`short-name fully-qualified-domain-name`

Here is an example:

```
abchost abchost.abcdomain
```

#### User Account Requirements for Authentication

Here are the user account requirements for authentication:
SAS Viya must be configured for pluggable authentication module (PAM) support.

The default administrative user for the CAS server deployments is the cas local user account. It is recommended that you change this account to a network account so that the local cas user does not generate a credentials cache.

Ensure that the network user account has generated a credentials cache in the location that is defined in your krb5.conf file or in the /tmp/ directory:

1. Log on to CAS Server Monitor as the user. Verify the time at which you logged on.
2. Verify that the file has a timestamp that is equal to the time that you logged on to CAS Server Monitor. Here is an example:

   `/tmp/krb5cc_53736`

### Java Requirements

Ensure that Java is set up appropriately.

For the Hadoop cluster:

- The SAS script that is deployed to the Hadoop nodes) requires the JAVA_HOME environment variable to be correctly set for the Hadoop cluster. The SAS script will also be called by the remote connection from the SAS Cloud Analytic Services hosts. The SAS script will run `$HADOOP_HOME/bin/hadoop`. Most Hadoop distributions will define the JAVA_HOME environment variable in `$HADOOP_HOME/libexec/hadoop-config.sh`. Therefore, it is important that this script is validated to ensure that the value of JAVA_HOME is correctly set.

- If you are using Advanced Encryption Standard (AES) encryption with Kerberos, manually add the Java Cryptography Extension local_policy.jar file in each place that JAVA_HOME resides in the Hadoop cluster. If you are located outside the United States, you must also manually add the US_export_policy.jar file. The addition of these files is governed by the United States import control restrictions.

For the SAS Cloud Analytic Services (CAS) hosts:

- If you are using the Oracle JRE or the IBM JRE, use the two JAR files rather than the existing local_policy.jar file and the US_export_policy.jar file. These files are in your JRE location, which is typically the `JAVA_HOME/jre/lib/security/` directory. These files can be obtained from the IBM website or the Oracle website.

- If you are using the Oracle JRE or the IBM JRE, it is recommended that you back up the existing local_policy.jar file and the US_export_policy.jar file in case they ever need to be restored. If you are using the OpenJDK, you do not need to back up these files because they will unlikely need to be restored.

### Kerberos Service and Service Principal Name Requirements

A Kerberos principal is a service or user that is known to the Kerberos system. A Kerberos principal is required for SAS Cloud Analytic Services.

This principal must be in the format of a Service Principal Name (SPN). The SPN includes a realm name, which is the capitalization of the domain name.

**Note:** In the SPN, REALM must be specified in uppercase characters.

```
Service Class/Fully Qualified Hostname@REALM
```

The default Service Class is “sascas”. A different Service Class can be used if required.

Here is an example:
With an Active Directory Kerberos Key Distribution Center (KDC), the User Principal Name (UPN) for the user must be the same as the SPN because Active Directory allows the initialization of a Kerberos Ticket-Granting Ticket only for a UPN. This restriction does not apply to other Kerberos distributions because the other Kerberos distributions do not distinguish between a UPN and an SPN.

**Kerberos Keytab Requirements**

- In order for a principal to be used by SAS Cloud Analytic Service, Kerberos Keytab is required.
  - The default Kerberos Keytab location and filename is `/etc/sascas.keytab`. The location can be changed.
  - The keytab file contains the long-term keys for the principal.
  - The keytab file must be available on the CAS Controller for the operating system account running the CAS Controller process. By default, this is the cas account.

- To use the GSSAPI for SSH, the end user must be able to obtain a service ticket to connect to the remote machine. The SSH client will request a service ticket for:

  ```
  host/<Fully Qualified Hostname>
  ```

- The host Service Principal Name must be registered in the Key Distribution Center (KDC). The SSH Daemon will check the default keytab of the operating system for a long-term key associated with the host principal. The default keytab on Linux is `/etc/krb5.keytab`. Therefore, this Kerberos Keytab must contain the long-term keys for the Host principal.

**Configure Passwordless SSH to Use Kerberos and GSSAPI**

Traditionally, public key authentication in SSH is used in order to meet the passwordless access requirement. For Secure Mode Hadoop, GSSAPI with Kerberos is used as the passwordless SSH mechanism. GSSAPI with Kerberos meets the passwordless SSH requirements and also supplies Hadoop with the credentials that are required for users in order to perform operations in HDFS with SASHDAT files. Certain options must be specified in the SSH daemon configuration file and the SSH client configuration files to support a default configuration of the SSH Daemon (SSHD).

1. In the `sshd_config` file, specify the `GSSAPIAuthentication` option:

   ```
   GSSAPIAuthentication yes
   ```

   **Note:** By default, the SSH Daemon will validate Service Tickets where the SPN matches only the host's current host name. In multi-homed or systems using a DNS alias, the SSH connection will fail. The SSH Daemon can be configured to validate the Service Ticket using any value within the default Kerberos Keytab. To enable the SSH Daemon to use any value in the Kerberos Keytab, the property `GSSAPIStrictAcceptorCheck` must be set to no.

   ```
   GSSAPIStrictAcceptorCheck no
   ```

2. In the `ssh_config` file, specify these options:

   ```
   Host *.domain.net
   GSSAPIAuthentication yes
   GSSAPIDelegateCredentials yes
   ```

   where `domain.net` is the domain name that is used by the machine in the Hadoop cluster.
**Deploying SAS Plug-ins for Hadoop**

**Overview of Deploying SAS Plug-ins for Hadoop**

To deploy SAS Plug-ins for Hadoop:

1. **Copy** the SAS Plug-ins for Hadoop files to the Hadoop cluster.

2. **Install** SAS Plug-ins for Hadoop to the Hadoop NameNode and to each DataNode by using one of the following methods:
   - the `sashdat-install.sh` script (supplied by SAS)
     - Note: The `sashdat-install.sh` script installs SAS Plug-ins for Hadoop on all supported Hadoop distributions.
   - parcel with Cloudera Manager
   - stack with Ambari

3. **Configure** HDFS Service properties.

**Copying the Plug-ins Files to the Hadoop Cluster**

Follow these steps to copy SAS Plug-ins for Hadoop files to the Hadoop cluster:

1. On the CAS controller, change to the `/opt/sas/viya/home/SASFoundation/hdatplugins` directory.
   The directory contains the following files:
   - `sashdat-install.sh`
   - `sashdat-03.04.gz`

2. Copy the `sashdat-install.sh` and `sashdat-03.04.gz` files from the CAS controller to the `/tmp` directory of the NameNode host of your Hadoop cluster.
   - Note: Ensure that the file permissions are set to 0755.

3. Go to one of the following sections:
   - “`sashdat-install.sh`” on page 263
   - “Cloudera Manager” on page 263
   - “Ambari” on page 264

**Installing SAS Plug-ins for Hadoop**

Depending on your Hadoop distribution, you can install SAS Plug-ins for Hadoop by using `sashdat-install.sh`, Cloudera Manager, or Ambari.
sashdat-install.sh
You can use the sashdat-install.sh script supplied by SAS to install SAS Plug-ins for Hadoop on all supported Hadoop distributions.

1 Make sure that have reviewed “Overview and Requirements” on page 271.

2 Log on to the Hadoop NameNode machine (blade 0) with a UNIX account that has sudo privileges and passwordless SSH access to every machine in the Hadoop cluster.

3 Change to the directory that was specified in Step 2, and run the sashdat-install.sh script using one of the following commands:
   - Deploy with the ‘hdfs’ account querying the hdfs service for the list of machines:
     ```
     sashdat-install.sh -add
     ```
     Here is an example:
     ```
     ./sashdat-install.sh -add
     ```
   - Deploy supplying your own list of machines:
     ```
     sashdat-install.sh -add -hostfile host-list-filename
     ```
     Here is an example:
     ```
     ./sashdat-install.sh -add -hostfile /tmp/my_hosts
     ```
   - Deploy specifying a different parent installation path:
     ```
     ./sashdat-install.sh -add -hdathome /opt/my_path/
     ```
     For more information, see “sashdat-install.sh Reference”.

Cloudera Manager
You can use Cloudera Manager with parcel to install SAS Plug-ins for Hadoop on all supported Cloudera Hadoop distributions.

1 On the CAS controller machine, navigate to the `/opt/sas/viya/home/SASFoundation/hdatplugins/parcel/` directory. Copy the parcel directory to the `tmp` directory of the file system of the host where Cloudera Manager is installed.
   Note: Ensure the files in the parcel directory have executable permissions.

2 From the `tmp` directory, run the following script:
   Note: The user account that you use to run the script must have super user (sudo) or root access.
   ```
   ./install_parcel.sh -v distro
   ```
   where `tmp` directory is the file system location where you copied from the SAS Viya installation and `distro` is the following Linux distribution: redhat6.
   Here is an example:
   ```
   install_parcel.sh -v redhat6
   ```

3 Select Y when asked to restart the Cloudera Manager server.

4 Log on to Cloudera Manager as administrator.

5 Activate the parcel.
   - Click Distribute to copy the parcel to all nodes.
Click **Activate**. You are prompted to restart the cluster or to close the window.

When prompted, click **Close**.  

**CAUTION!** Do not restart the cluster.

---

**Ambari**

You can use Ambari with stack to install SAS Plug-ins for Hadoop on all supported Cloudera Hadoop distributions.

*Note:* The following deployment steps assume that the hdatplugins rpm package is installed directly on one of the following machines:

**CAUTION!** When the Hortonworks Hadoop stack is upgraded, the HDATPlugins stack must be deactivated and then reactivated. If the Hortonworks Hadoop level is upgraded in **Express** mode on Ambari, the HDATPlugins stack must be restarted. If the Hortonworks Hadoop level is upgraded in **Rolling** mode, a restart of the HDATPlugins stack is not required.

- the Ambari server
- a machine in the network that is accessible to the Ambari server

1. To launch the script, on the CAS controller machine, navigate to the `/opt/sas/viya/home/SASFoundation/hdatplugins/stack/` directory. Copy the `/stack` directory to the `/tmp` directory of the host where the Ambari Server is installed, and run the following command:

   ```
   ./install_hdatplugins.sh Ambari-admin-username
   ```

   After the script finishes running, this message is displayed: You can install the HDATPLUGINS stack now from Ambari Cluster Manager.

2. Log on to Ambari. On the Ambari server, deploy the services.
   
   a. Click **Actions** and select **+ Add Service**. The Add Service Wizard page and the Choose Services panel open.

   b. In the Choose Services panel, select **SASHDAT**. Click **Next**. The Assign Slaves and Clients panel opens.

   c. In the Assign Slaves and Clients panel under **Client**, select all data nodes and all name nodes where you want the stack to be deployed. The Customize Services panel opens. The SASHDAT stack is listed.

   d. Do not change any settings on the Customize Services panel. Click **Next**.

      *Note:* If your cluster is secured with Kerberos, the Configure Identities panel opens. Enter your Kerberos credentials in the **admin_principal** text box and the **admin_password** text box. Click **Next**. The Review panel opens.

   e. Review the information in the panel. If the values are correct, click **Deploy**. The Install, Start, and Test panel opens. After the stack is installed on all nodes, click **Next**. The Summary panel opens.

   f. Click **Complete**. The stacks are now installed on all nodes of the cluster. SASHDAT is displayed on the Ambari dashboard.

   g. On every node, all files in the `/usr/hdp/Hadoop-version/hadoop/bin` directory must be executable with file permissions of 755.

---

**Configuring HDFS Service Properties**

Configure HDFS service properties for SAS Plug-ins for Hadoop based on your Hadoop distribution.
Cloudera Hadoop

Note: If Cloudera Manager provides a choice between classic and new layouts, use classic layout.

To use Cloudera Manager to configure HDFS service properties for SAS Plug-ins for Hadoop:

1. Log on to Cloudera Manager as an administrator.

2. Navigate to the Service-Wide group. Under Advanced, add the following lines to the HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property:

   `<property><name>dfs.namenode.plugins</name> <value>com.sas.cas.hadoop.NameNodeService</value></property>`
   `<property><name>dfs.datanode.plugins</name> <value>com.sas.cas.hadoop.DataNodeService</value></property>`
   `<property><name>com.sas.cas.hadoop.service.namenode.port</name> <value>15452</value></property>`
   `<property><name>com.sas.cas.hadoop.service.datanode.port</name> <value>15453</value></property>`
   `<property><name>dfs.namenode.fs-limits.min-block-size</name> <value>0</value></property>`
   `<property><name>com.sas.cas.hadoop.short.circuit.command</name> <value>/opt/sas/HDATHome/bin/sascasfd</value></property>`

   Note: You can change the port for the SAS name node and data node plug-ins. This example shows the default ports (15452 and 15453, respectively).

   Note: The SAS Plug-ins for Hadoop installation directory, HDATHome, is deployed under /opt/sas/, by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

3. Navigate to the Gateway Default Group. Under Advanced, add the following lines to the HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property:

   `<property><name>com.sas.cas.hadoop.service.namenode.port</name> <value>15452</value></property>`
   `<property><name>com.sas.cas.hadoop.service.datanode.port</name> <value>15453</value></property>`
   `<property><name>dfs.namenode.fs-limits.min-block-size</name> <value>0</value></property>`
   `<property><name>com.sas.cas.hadoop.short.circuit.command</name> <value>/opt/sas/HDATHome/bin/sascasfd</value></property>`

4. Navigate to the Service-Wide group. Under Advanced, add the following line to the HDFS Service Environment Advanced Configuration Snippet (Safety Valve) property:

   `HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*`

5. Navigate to the Gateway Default Group. Under Advanced, add the following property in HDFS Client Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh:

   `HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*`

6. Click Cloudera Manager Home, and then select the Yarn service. Within the Yarn service, navigate to the Gateway Default Group by clicking Configuration and Gateway Default Group ⇒ Advanced. Add the following properties in Gateway Client Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh:

   `JAVA_HOME=Java-home-path
   HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*`

   Note: The value of the JAVA_HOME variable should be a valid path to the Java installation used by the Cloudera Hadoop system on all Hadoop nodes. For example:

   `JAVA_HOME=/usr/java/jdk1.7.0_67-cloudera`

7. Save changes.
From the Cloudera Manager home, select the drop-down list for your cluster and select **Deploy Client Configuration**. In the dialog box, select **Deploy Client Configuration**, and then click **Close**.

Restart the HDFS service and any dependencies in Cloudera Manager.

Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see “Verifying CAS SASHDAT Access to HDFS” on page 267.

**Hortonworks Data Platform Hadoop**

To use Ambari to configure Hortonworks HDFS service properties for SAS Plug-ins for Hadoop:

1. Log on to Ambari.
2. Click **HDFS Service**.
3. Choose **Config Section**.
4. Click **Advanced**.
5. Select **Custom hdfs-site** and add the following properties:
   - `dfs.namenode.plugins`
     - `com.sas.cas.hadoop.NameNodeService`
   - `dfs.datanode.plugins`
     - `com.sas.cas.hadoop.DataNodeService`
   - `com.sas.cas.hadoop.service.namenode.port`
     - `15452`
     - *Note:* You can change the port for the SAS name node and data node plug-ins. This example shows the default ports (15452 and 15453, respectively).
   - `com.sas.cas.hadoop.service.datanode.port`
     - `15453`
   - `dfs.namenode.fs-limits.min-block-size`
     - `0`
   - `com.sas.cas.hadoop.short.circuit.command`
     - `/opt/sas/HDATHome/bin/sascasfd`
     - *Note:* The SAS Plug-ins for Hadoop installation directory, **HDATHome**, is deployed under `/opt/sas/`, by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.
6. Save the properties.
7. Add the following statement to the **hadoop-env template** of HDFS on the **Advanced hadoop-env** tab, in the section, **# Set Hadoop-specific environment variables here**:
   ```
   export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
   ```
   *Note:* Ensure that the export command occupies a single line.
8. Restart all Hortonworks Data Platform (HDP) services and MapReduce services.
9. Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see “Verifying CAS SASHDAT Access to HDFS” on page 267.
Apache Hadoop

To configure Apache Hadoop HDFS service properties for SAS Plug-ins for Hadoop:

1. Define the following properties in `$HADOOP_HOME/etc/hadoop/hdfs-site.xml` and propagate the changes across all nodes in your Hadoop cluster:

   - **Note:** The SAS Plug-ins for Hadoop installation directory, HDATHome, is deployed under `/opt/sas/` by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

   - **Note:** Adjust values appropriately for your deployment. The port numbers should be valid port numbers.

   ```xml
   <property>
   <name>dfs.namenode.plugins</name>
   <value>com.sas.cas.hadoop.NameNodeService</value>
   </property>
   <property>
   <name>dfs.datanode.plugins</name>
   <value>com.sas.cas.hadoop.DataNodeService</value>
   </property>
   <property>
   <name>com.sas.cas.hadoop.service.namenode.port</name>
   <value>15452</value>
   </property>
   <property>
   <name>com.sas.cas.hadoop.service.datanode.port</name>
   <value>15453</value>
   </property>
   <property>
   <name>dfs.namenode.fs-limits.min-block-size</name>
   <value>0</value>
   </property>
   <property>
   <name>com.sas.cas.hadoop.short.circuit.command</name>
   <value>/opt/sas/HDATHome/bin/sascasfdc</value>
   </property>
   ```

2. On every machine in the cluster, in `/etc/hadoop/hadoop-env.sh`, in the section, `# Set Hadoop-specific environment variables here`, set `HADOOP_CLASSPATH` to the following value:

   ```bash
   export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
   ```

   - **Note:** Ensure that the export command occupies a single line.

3. Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS,

   ```sas
   cas mysession;
   caslib testhdat datasource=(srctype="hdfs") path="/test";
   ```

Verifying CAS SASHDAT Access to HDFS

1. To create the `/test` directory in HDFS, run the following commands as the hdfs user. The `/test` directory is used for testing the Hadoop cluster with SAS test jobs.

   ```bash
   hadoop fs -mkdir /test
   hadoop fs -chmod 777 /test
   ```

2. To verify that the software has been successfully deployed, run the following SAS code:

   ```sas
   cas mysession;
   caslib testhdat datasource=(srctype="hdfs") path="/test";
   ```
proc casutil;
  load data=sashelp.zipcode;
  save casdata="zipcode" replace;
run;

3 If you have successfully saved the data in CAS to the SASHDAT format in HDFS, the following message appears in the log output:

NOTE: Cloud Analytic Services saved the file zipcode.sashdat to HDFS in caslib TESTHDAT.

### Uninstalling SAS Plug-ins for Hadoop

Depending on your Hadoop distribution, you can uninstall SAS Plug-ins for Hadoop by using sashdat-install.sh, Cloudera Manager, or Ambari.

**sashdat-install.sh**

You can use the sashdat-install.sh script that is supplied by SAS to uninstall SAS Plug-ins for Hadoop on all supported Hadoop distributions.

1 Make sure that have reviewed “Overview and Requirements” on page 271.

2 Log on to the Hadoop NameNode machine (blade 0) with a UNIX account that has sudo privileges and passwordless SSH access to every machine in the Hadoop cluster.

3 Remove or disable these properties in `$HADOOP_HOME/etc/hadoop/hdfs-site.xml`:

- `<name>dfs.namenode.plugins</name>`
  - `<value>com.sas.cas.hadoop.NameNodeService</value>`
- `<name>dfs.datanode.plugins</name>`
  - `<value>com.sas.cas.hadoop.DataNodeService</value>`
- `<name>com.sas.cas.hadoop.service.namenode.port</name>`
  - `<value>15452</value>`
- `<name>com.sas.cas.hadoop.service.datanode.port</name>`
  - `<value>15453</value>`
- `<name>com.sas.cas.service.allow.put</name>`
  - `<value>true</value>`
- `<name>dfs.namenode.fs-limits.min-block-size</name>`
  - `<value>0</value>`
- `<name>com.sas.cas.hadoop.short.circuit.command</name>`
  - `<value>/opt/sas/HDATHome/bin/sascasfd</value>`

4 Change to the `/opt/sas/viya/home/SASFoundation/hdatplugins` directory. Run the sashdat-install.sh script using one of the following commands:

Note: To run the script, use a user account that is allowed to run the hdfs Hadoop command. If an appropriate user account does not exist, run the script with either the `-host myhost.domain.com` or `-hostfile /tmp/myhostfile.txt` option.
- Uninstall by querying the hdfs service for the list of machines:
  
  `sashdat-install.sh -remove`

  Here is an example:
  
  `./sashdat-install.sh -remove`

- Uninstall supplying your own list of machines:
  
  `sashdat-install.sh -remove -hostfile host-list-filename`

  Here is an example:
  
  `./sashdat-install.sh -remove -hostfile /tmp/my_hosts`

- Uninstall specifying a different parent installation path:
  
  `./sashdat-install.sh -remove -hdathome /opt/my_path/`

  For more information, see “sashdat-install.sh Reference”.

### Cloudera Manager

1. Log on to the Cloudera Manager as an administrator.

2. From Cloudera Manager Home, select the HDFS service. Within the HDFS service, select **Configuration** to remove the HDFS configuration properties.

   **Note:** In the following steps, you must remove specific HDFS configuration properties. Locate the property to remove by specifying its name in the search bar.

   a. In the `dfs.namenode.plugins` property, remove the following line from the plug-in configuration for the NameNode:

      `com.sas.cas.hadoop.NameNodeService`

   b. In the `dfs.datanode.plugins` property, remove the following line from the plug-in configuration for the DataNode:

      `com.sas.cas.hadoop.DataNodeService`

3. Navigate to the Service-Wide group. Under Advanced, remove the following lines from the HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property.

   **Note:** The SAS Plug-ins for Hadoop installation directory, **HDATHome**, is deployed under `/opt/sas/` by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

4. Navigate to the Gateway Default Group. Under Advanced, remove the following lines from the HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property.

   ```xml
   <name>com.sas.cas.hadoop.service.namenode.port</name>
   <value>15452</value>

   <name>com.sas.cas.hadoop.service.datanode.port</name>
   <value>15453</value>

   <name>dfs.namenode.fs-limits.min-block-size</name>
   <value>0</value>

   <name>com.sas.cas.hadoop.short.circuit.command</name>
   <value>/opt/sas/HDATHome/bin/sascasfd</value>
   ```

5. Navigate to the Gateway Default Group. Under Advanced, remove the following lines from the HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property.
Navigate to the HDFS Environment Client Safety Valve. Remove the following property from the HDFS Service Environment Advanced Configuration Snippet (Safety Valve) Server-wide:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

Navigate to the Gateway Default Group. Remove the following property from the HDFS Client Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh Gateway Default Group:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

From Cloudera Manager Home, select the YARN service. Within the YARN service, navigate to the Gateway Default Group. Remove the following property from the Client Safety Valve Gateway Client Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh Gateway Default Group:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

From the Menu bar, select **Hosts** 🔄 **Parcels**.

Select the SASHDAT parcel.

Deactivate the SASHDAT parcel.

Remove the SASHDAT parcel.

Delete the SASHDAT parcel.

When prompted, click **Close**.

### Ambari

**Note:** To remove the stack, root or passwordless sudo access is required.

1. Log on to Ambari as an administrator, and stop HDFS, YARN, and MapReduce services.

2. Delete the custom hdfs-site.xml file that contains the SAS Plug-ins for Hadoop properties (such as, the `com.sas.cas.*` properties).

3. Remove the following statement from the **hadoop-env template** of HDFS on the **Advanced hadoop-env** tab, in the section, # Set Hadoop-specific environment variables here:

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

4. On the CAS controller machine, navigate to the `/opt/sas/viya/home/SASFoundation/hdatplugins/stack/` directory and run the following command to delete the stack:

   `/delete_stack.sh Ambari-Admin-User-Name`

5. At the prompt, enter the Ambari administrator password.

6. Enter **y** to restart the Ambari server.

The SASHDAT service is no longer listed on the Ambari dashboard.
sashdat-install.sh Reference

Overview and Requirements

The sashdat-install.sh script enables you to deploy SAS Plug-ins for Hadoop on a Support for Databases in SAS Viya 3.4. The script provides an alternative to Cloudera parcels and Ambari stacks.

The UNIX account with which the script is run requires sudo privileges and passwordless SSH access to every machine in the Hadoop cluster when adding and removing SAS Plug-ins for Hadoop. No sudo access is required when you are checking whether the plug-ins are correctly installed on all data nodes.

When adding or removing SAS Plug-ins for Hadoop, the sashdat-install.sh script attempts to query the Hadoop configuration to automatically discover the machine name for all of the nodes in the cluster. In order to query the hdfs service for machine names, the script assumes that your site uses the default Hadoop user account 'hdfs.' You can provide a different Hadoop account with execution permissions for the hdfs command, or, provide your own list of machine names.

You must provide a list of machine names under these conditions:
- the hdfs service is down.
- you are adding new machines to the cluster.
- you want to use a list of machines that is different from what is in the Hadoop configuration.

Syntax

- Add SAS Plug-ins for Hadoop:

  sashdat-install.sh -add <-hostfile host-list-filename | -host "host-list"> <-hdfsuser user-ID> <-hdathome parent-installation-path>

- Remove SAS Plug-ins for Hadoop:

  sashdat-install.sh -remove <-hostfile host-list-filename | -host host-list> <-hdfsuser user-ID> <-hdathome parent-installation-path>

- Check whether SAS Plug-ins for Hadoop is properly installed:

  sashdat-install.sh -x -check <-hdathome parent-installation-path>

- Query the version of SAS Plug-ins for Hadoop:

  sashdat-install.sh -version <-hdathome parent-installation-path>

Options

-add
  installs SAS Plug-ins for Hadoop on all machines in the cluster, or on a user-supplied the list of machines.

Requirements

The UNIX user account with which you run the sashdat-install.sh script must have sudo permissions and passwordless SSH access to every machine in the Hadoop cluster.

The script assumes that your site uses the default Hadoop user account, 'hdfs,' with which the script automatically retrieves the list of data nodes from the Hadoop configuration. If your site does not use the 'hdfs' user account, then you must use the -hdfsuser user-ID option to
provide a valid Hadoop user account with execution permissions for the hdfs command. Or, you can provide your own list of machines using either the -hostfile or -host option.

-hdathome parent-installation-path  (optional) specifies a custom parent installation path for the plug-ins instead of the default /opt/sas path. The subdirectory HDATHome will be created under the specified -hdathome path.

-hdfsuser user-ID  (optional) specifies the user ID that has execution permissions for hdfs to run the hdfs dfsadmin -report command to retrieve the machine names of the nodes in the Hadoop cluster.

The -hdfsuser option is not required when the default Hadoop account, 'hdfs,' is present, or when you supply your own list of machines in the Hadoop cluster using the -hostfile or -host option.

-hostfile host-list-filename  (optional) specifies the full path of the file that contains the list of machine names for all of the cluster nodes on which the plug-ins are installed or removed.

Requirement The host list file must contain one fully qualified machine name per line.

Example

machine001.example.com
machine002.example.com
machine003.example.com
machine004.example.com

-host "host-list"  (optional) specifies the list of machine names for all of the cluster nodes on which the plug-ins are installed or removed.

Requirement If you specify more than one machine, the names must be separated by spaces or commas. The wildcard character, the asterisk (*), is allowed.

Examples

-host server1 server2 server3
-host blue1,blue2,blue3
-host bluesvr*

-removes the plug-ins on all machines in the cluster, or on a user-supplied the list of machines.

Requirements The UNIX user account with which you run the sashdat-install.sh script must have sudo permissions and passwordless SSH access to every machine in the Hadoop cluster.

The script assumes that your site uses the default Hadoop user account, hdfs, with which the script automatically retrieves the list of data nodes from the Hadoop configuration. If your site does not use the hdfs user account, then you must use the -hdfsuser user-ID option to provide a valid Hadoop user account with execution permissions for the hdfs command. Or, you can provide your own list of machines using either the -hostfile or -host option.

-version <-hdathome parent-installation-path>  displays the version of the plug-ins that are installed.

Example ./sashdat-install.sh -version

-x -check <-hdathome parent-installation-path>  checks whether the plug-ins are installed correctly on all data nodes.
Tip You can specify the hosts for which you want to check the plug-ins by using the -hostfile or -host option.

Example ./sashdat-install.sh -x -check

Add Examples
This section demonstrates various ways to use sashdat-install.sh to add SAS Plug-ins for Hadoop to your supported Hadoop cluster:

Add using the ‘hdfs’ account to query Hadoop for a list of machines:
.
Add using the ‘my-hdfs’ account to query Hadoop for a list of machines:
.
Add specifying a user-supplied list of machines:
.
Add specifying a user-supplied installation path:
.

Remove Examples
This section demonstrates various ways to use sashdat-install.sh to remove SAS Plug-ins for Hadoop to your supported Hadoop cluster:

Remove using the ‘hdfs’ account to query Hadoop for a list of machines:
.
Remove using the ‘my-hdfs’ account to query Hadoop for a list of machines:
.
Remove specifying a user-supplied list of machines:
.
Remove specifying a user-supplied installation path:
Prerequisites

The SAS in-database deployment package requires the following:

- version 15.10 of the Teradata client and server environment.
- the CAS controller and each CAS worker node must have an IP address that can be routed to externally from the SAS Embedded Process nodes.
- approximately 200 MB of disk space in the /opt file system on each Teradata Trusted Parallel Appliance (TPA) node.
Connections from SAS 9.4 Clients

The following SAS 9.4 clients can connect to a Teradata Server that has installed the SAS Viya version of SAS Embedded Process for Teradata:

- SAS Analytics Accelerator for Teradata
- SAS High-Performance Analytics
- SAS In-Database Code Accelerator for Teradata
- SAS LASR
- SAS Scoring Accelerator for Teradata

Installing the SAS In-Database Deployment Package for Teradata

(Optional) Deploy TLS Certificates

If you are using a SAS Data Connect Accelerator, the data that is transferred between the data provider and the CAS server is not encrypted by default. However, SAS Viya supports TLS encryption between the data provider and the CAS server. When SAS Viya 3.4 is deployed, TLS is enabled and configured on the CAS server (server side). The deployment process provides a default level of encryption for data in motion. Options are set in the vars.yml file and are defined in the casconfig_deployment.lua file. These settings enable data connector encryption and specify the location of the TLS private key and the password.

However, you must take additional steps to enable encryption on the data provider. The prerequisites and the process for enabling TLS encryption on the data provider are different for each data provider. The first step is to deploy the TLS certificates across all nodes in the cluster.

1. On the CAS controller machine, locate the TLS certificates in the trustedcerts.pem file in the /opt/sas/viya/config/etc/SASSEcurityCertificateFramework/cacerts/ directory.

2. Copy the trustedcerts.pem file to the SAS Embedded Process security/certs directory on Teradata.

3. To complete the deployment of TLS encryption, you also must update a dcsecurity.properties file. Both the .pem file and the dcsecurity.properties file must then be copied to all nodes on the server. For more information about how to complete the deployment, see Encrypt Data Transfer When Using the SAS Data Connect Accelerator in Encryption in SAS Viya: Data in Motion.

Verify the Connection to Teradata

To verify that SAS Data Connector to Teradata and SAS Data Connect Accelerator for Teradata were successfully deployed:

1. Sign on to SAS Studio:
   a. Open SAS Studio from a URL with the following format: https://http-proxy-host-name/SASStudio
   b. Enter the credentials for your operating system account.
2 Start a CAS session:
   a In the navigation pane, open the **Snippets** section.
   b Select **Snippets ⇒ Cloud Analytic Services**.
   c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
   d In the toolbar, click ![Run](image) to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Teradata LIBNAME:

   ```sas
   libname tlib teradata server="teradata-host-name" database="teradata-database-name" user="user-ID"
   password="user-Password";
   ```

   If SAS/ACCESS to Teradata was successfully deployed, the execution of the LIBNAME statement will return without error.

4 From SAS Studio, edit and run the following SAS code to verify the SAS Data Connector to Teradata:

   ```sas
   caslib tdlib datasource=(srctype="teradata", dataTransferMode="auto", username="user-ID",
   password="user-Password",
   server="teradata-host-name", database="teradata-database-name")
   ```

   ```sas
   proc casutil;
   list files incaslib="tdlib";
   run;
   ```

   If the validation of the libname or data connector failed, error was return on the execution of the libname or no table information was returned for the data connector, you should perform the configuration steps again.

5 From SAS Studio, edit and run the following SAS code to verify the SAS Data Connect Accelerator for Teradata:

   ```sas
   caslib teraplib datasource=(srctype="teradata", dataTransferMode="parallel" username="<user ID>",
   password="<password>", server="<Teradata host name>", database="<Teradata database name>");
   ```

   ```sas
   proc casutil;
   list files incaslib="teraplib";
   run;
   ```

   If the data connector was successfully deployed, the results are the names of the tables in Teradata. If you do not see table names that you recognize, you should perform the configuration steps again.

### Install the Support Functions for the SAS Embedded Process

The support function (sasepfunc) package for the SAS Embedded Process includes stored procedures that generate SQL to interact with the SAS Embedded Process. The support function package also includes functions that load the SAS program and other run-time control information into shared memory. The setup script for the support function package creates the SAS_SYSFNLIB database and the fast path functions in TD_SYSFNLIB.

The support function package is available from the Teradata Software Server. For access to the package that includes the installation instructions, contact your local Teradata account representative or the Teradata consultant that supports your SAS and Teradata integration activities.
Configuring SAS Data Quality Accelerator for Teradata

Overview
You can use SAS Data Quality technology in the Teradata database via the SAS Data Quality Accelerator for Teradata. To use SAS Data Quality Accelerator for Teradata, you must perform the following tasks after deploying the SAS In-Database Technologies for Teradata (SAS Embedded Process):

- install SAS data quality stored procedures in the Teradata database
- configure a SAS Quality Knowledge Base (QKB) in the Teradata database

The SAS In-Database Technologies deployment provides shell scripts that enable you to install and manage the data quality stored procedures within the Teradata database. In addition, it contains a shell script that enables you to package the QKB for deployment inside the Teradata database.

The QKB is a collection of files that store data and logic that support data management operations. SAS software products reference the QKB when performing data management operations on your data.

Each Teradata node needs approximately 8 GB for the QKB.

SAS In-Database Deployment Package for Teradata
The SAS Embedded Process deployment package for Teradata (sepcoretera) installs three scripts in the /opt/SAS/ep/home/install/pgm directory of the Teradata database server:

- a stored procedure creation script named dq_install.sh
- a user authorization script named dq_grant.sh
- a stored procedure removal script named dq_uninstall.sh

Run the dq_install.sh script to create the data quality stored procedures in the Teradata database and the dq_grant.sh script to grant users permission to execute the data quality stored procedures.

The dq_uninstall.sh script is provided to enable you to remove the data quality stored procedures from the database. You must remove any data quality stored procedures that have already been installed from the Teradata database before upgrading or re-installing either SAS Data Quality Accelerator for Teradata or the SAS Embedded Process.

Note: All three scripts must be run as the root user.

Using the dq_install.sh Script
The dq_install.sh script is located in the /opt/SAS/ep/home/install/pgm directory of the Teradata database server.

The dq_install.sh script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_PASS=, DBC_SRVR=, and DBC_USER= variables.

Running dq_install.sh puts the data quality stored procedures into the SAS_SYSFNLIB database and enables the accelerator functionality.

Here is the syntax for executing dq_install.sh:

```
./dq_install.sh <-l log-path>
```
log-path
specifies an alternative name and location for the dq_install.sh log. When this parameter is omitted, the script
creates a file named dq_install.log in the current directory.

The next step in the installation is to grant users permission to execute the stored procedures.

Using the dq_grant.sh Script
The dq_grant.sh shell script is provided to enable the Teradata system administrator to grant users authorization
to the data quality stored procedures. The dq_grant.sh shell script is provided to enable the Teradata system
administrator to grant users authorization to the data quality stored procedures. The dq_grant.sh script is located
in the /opt/SAS/ep/home/install/pgm directory of the Teradata database server.

The dq_grant.sh script requires modification before it can be run. The Teradata administrator must edit the shell
script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_SRVR=,
DBC_USER=, and DBC_PASS= variables.

Here is the syntax for executing dq_grant.sh:

```bash
./dq_grant.sh <-l log-path> user-name
```

log-path
specifies an alternative name and location for the dq_grant.sh log. When this parameter is omitted, the script
creates a file named dq_grant.log in the current directory.

user-name
is the user name to which permission is being granted. The target user account must already exist in the
Teradata database.

The authorizations granted by dq_grant.sh supplement existing authorizations that the target user account
already has in the Teradata database.

You can verify that authorization was granted successfully for a user by logging on to the database as the user
and issuing the following command in a Basic Teradata Query (BTEQ) session:

```sql
call sas_sysfnlib.dq_debug();
```

The command will fail if the user does not have permission. Otherwise, it will have no effect.

The data quality stored procedures are not yet ready to use. A QKB must be installed in the Teradata database
for the data quality stored procedures to be usable.

Locating the QKB
The QKB is located in the following directory:

UNIX: /opt/sas/spre/home/share/refdata/qkb

Packaging the QKB
Before a QKB can be deployed in the Teradata database, you must package it into an .rpm file. An .rpm file is a
file that is suitable for installation on Linux systems that use RPM package management software. SAS Data
Quality Accelerator for Teradata provides the qkb_pack script to package the QKB into an .rpm file.

qkb_pack is created in the following directory during deployment:

UNIX
SASHome/SASDataQualityAcceleratorforTeradata/9.4/install/pgm

You must execute qkb_pack from the SASHome/SASDataQualityAcceleratorforTeradata/9.4/
install/pgm location.
Here is the syntax for executing `qkb_pack`:

**Example Code A5.1**

UNIX:

```
./qkb_pack.sh qkb-dir out-dir
```

`qkb-dir`  
specify the path to the QKB. Use the name of the QKB’s root directory. Typically, the root directory is found at the following directories:


`out-dir`  
specify the directory where you want the package file to be created.

Here is an example of a command that you might execute in order to package a SAS QKB for Contact Information that resides on a UNIX computer.

```
cd /opt/sas/spre/home/SASFoundation/install/pgm  
./qkb_pack.cmd /opt/sas/spre/home/share/refdata/qkb/CI/29 /tmp
```

The package file that is created in /tmp will have a name in the following form:

```
sasqkb_product-version-timestamp.noarch.rpm
```

`product`  
is a two-character product code for the QKB, such as CI (for Contact Information) or PD (for Product Data).

`version`  
is the version number of the QKB.

`timestamp`  
is a UNIX datetime value that indicates when `qkb_pack` was invoked. A UNIX datetime value is stored as the number of seconds since January 1, 1970.

`noarch`  
indicates that the package file is platform-independent.

Here is an example of an output filename representing the QKB for Contact Information 29:

```
sasqkb_ci-29.0-1474057340608.noarch.rpm
```

After running `qkb_pack`, put the sasqkb package file on your Teradata database server in a location where it is available for both reading and writing. The package file must be readable by the Teradata Parallel Upgrade Tool. You need to move this package file to the server machine in accordance with procedures used at your site.

## Installing the QKB Package File with the Teradata Parallel Upgrade Tool

This installation should be performed by a Teradata systems administrator in collaboration with Teradata Customer Services. A Teradata Change Control is required when a package is added to the Teradata server. Teradata Customer Services has developed change control procedures for installing the SAS in-database deployment package.

The steps assume full knowledge of the Teradata Parallel Upgrade Tool and your environment. For more information about using the Teradata Parallel Upgrade Tool, see the Parallel Upgrade Tool (PUT) Reference, which is on the Teradata Online Publications site located at [http://www.info.teradata.com/GenSrch/eOnLine-Srch.cfm](http://www.info.teradata.com/GenSrch/eOnLine-Srch.cfm). On this page, search for “Parallel Upgrade Tool” and download the appropriate document for your system.

The following section explains the basic steps to install the sasqkb package file using the Teradata Parallel Upgrade Tool.
Note: It is not necessary to stop and restart the Teradata database when you install a QKB. However, if the SAS Embedded Process is running, you must stop it and then re-start it after the QKB is installed. It is also necessary to stop and restart the SAS Embedded Process for QKB updates. For information about stopping and restarting the SAS Embedded Process, see Controlling the SAS Embedded Process in SAS 9.4 and SAS Viya 3.4 Programming Documentation / In-Database Products: User’s Guide.

Note: The Teradata Parallel Upgrade Tool is a third-party product. Therefore, its prompts are subject to change.

1. Start the Teradata Parallel Upgrade Tool.
2. Be sure to select all Teradata TPA nodes for installation, including Hot Stand-By nodes.
3. If Teradata Version Migration and Fallback (VM&F) is installed, you might be prompted about whether to use VM&F. If you are prompted, choose Non-VM&F installation.

You can verify that the QKB installation was successful by running the following command from the shell prompt on one of the Teradata nodes.

```
psh "rpm -q -a" | grep sasqkb
```

If the installation was successful, the command returns the version number of the sasqkb package. Failure to return an output indicates that a library of that name could not be found.

The QKB is installed in the `/opt/qkb/default` directory of each Teradata node.

You are now ready to validate the data quality stored procedures for use.

### Validating the Accelerator Installation

Here is a simple BTEQ program that can be used to verify that the SAS Data Quality Accelerator for Teradata is operational.

The code first lists the locales that are installed in the QKB. Then it creates a table named Dqacceltest and executes the DQ_GENDER() stored procedure on the table. Before running the example, substitute a real value for the `output_table_1`, `output_table_2`, and `locale` variables throughout the program. For `locale`, use one of the values returned by the DQ_LIST_LOCALES() stored procedure. This example assumes that the SAS Data Quality Accelerator for Teradata is using the QKB for Contact Information.

The example also sets the SAS Data Quality Accelerator DQ_OVERWRITE_TABLE option to create temporary output tables in the SAS Data Quality Accelerator session. If you run the example again in the same SAS Data Quality Accelerator session, the new output tables overwrite any existing output tables and the output tables are automatically discarded at the end of the session. The DROP TABLE statement removes table Dqacceltest from your database.

```
call sas_sysfnlib.dq_list_locales('mydb.output_table_1');
select * from mydb.output_table_1;

call sas_sysfnlib.dq_set_option('DQ_OVERWRITE_TABLE', '1');

create table mydb.dqacceltest (id_num integer, name varchar(64))
unique primary index(id_num);

insert into mydb.dqacceltest (id_num, name) values (1, 'John Smith');
insert into mydb.dqacceltest (id_num, name) values (2, 'Mary Jones');

call sas_sysfnlib.dq_gender('Name', 'mydb.dqacceltest', 'name', 'id_num',
'mydb.output_table_2', 'locale');

select gender from mydb.output_table_2;
drop table mydb.dqacceltest;
```
If the request was successful, the SELECT statement produces an output table that contains the following:

<table>
<thead>
<tr>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

**Troubleshooting the Accelerator Installation**

Q. I ran the sample code and the output tables were not created in my user schema. What now?

A. The stored procedures can fail if one or more of the following conditions are true:

- The request specifies an output location to which the user does not have Write permission. Verify that you have access to the database that is specified in the `output_table` parameters.

- The data quality stored procedures are not installed correctly. Verify that the stored procedures are in the `SAS_SYSFNLIB` database by executing the following command in BTEQ:

```sql
select TableName from dbc.tables where databasename='SAS_SYSFNLIB' and tablename like 'dq_4';
```

The command should return a list similar to the following:

**Note:** This is an incomplete list.

```
TableName
-----------------------------
dq_set_qkb
dq_match_parsed
dqi_drop_view_if_exists
dqi_get_option_default
dq_debug
dq_propercase
dqi_tbl_dbname
dqi_drop_tbl_if_exists
dq_set_option
dqt_error
dq_standardize
dq_standardize_parsed
dq_debug2
dqi_invoke_table
dq_lowercase
dq_set_locale
dq_extract
dq_uppercase
dq_list_bindings
dqi_replace_tags
dq_list_defns
dqi_call_ep
dqi_get_bool_option
dqi_gen_toktxt
dqt_codegen
dq_match
dq_parse
dqt_trace
dq_pattern
```
The procedures are absent, run the dq_install.sh script again, making sure that you are logged in as Teradata system administrator.

Permission to the data quality stored procedures is not granted correctly. Verify that the target user name submitted to the dq_grant.sh script is a valid user account in the Teradata database. Verify that the database server and granter information in the dq_grant.sh shell script is correct.

The QKB is not in the correct location. Look for subdirectories similar to the following in the /opt/qkb/default directory on the Teradata nodes: chopinfo, grammar, locale, phonetx, regexlib, scheme, and vocab.

Your SQL request does not use the Teradata dialect. The stored procedures are invoked with the CALL keyword from any product that supports the Teradata SQL dialect. When you submit the data quality stored procedures in the SAS SQL procedure using explicit pass-through, the database connection is made in ANSI mode by default. You must specify the MODE= option to switch to Teradata mode. Refer to the SAS/ACCESS Interface to Teradata documentation for more information about the MODE= option. Refer to the appropriate documentation about how to set Teradata mode in other client programs.

### Updating and Customizing a QKB

SAS provides regular updates to the QKB. It is recommended that you update your QKB each time a new one is released. For a listing of the latest enhancements to the QKB, see “What’s New in SAS Quality Knowledge Base.” The What’s New document is available on the Quality Knowledge Base (QKB) for SAS and DataFlux Documentation site on support.sas.com.

Check the What’s New document for each QKB to determine which definitions have been added, modified, or deprecated, and to learn about new locales that might be supported. Contact your SAS account representative to order updated QKBs and locales. To deploy a new QKB, follow the steps in “Packaging the QKB” on page 279 and “Installing the QKB Package File with the Teradata Parallel Upgrade Tool” on page 280. The accelerator supports one QKB in the Teradata database.

The standard definitions in the QKB are sufficient for performing most data quality operations. However, you can use the Customize feature of DataFlux Data Management Studio to modify the QKB definitions to meet specific needs.

If you want to customize your QKB, SAS recommends that you customize your QKB on a local workstation before copying it to the Teradata database for deployment. When updates to the QKB are required, merge your customizations into an updated QKB locally, and copy the updated, customized QKB to the Teradata node. This enables you to deploy a customized QKB to the Teradata database using the same steps that you would use to deploy a standard QKB. Copying your customized QKB from a local workstation into your cluster also means that you will have a backup of the QKB on your local workstation. See the online Help provided with your SAS Quality Knowledge Base for information about how to merge any customizations that you have made into an updated QKB.

### Removing SAS Data Quality Accelerator from the Teradata Database

Before you can upgrade, re-install, or permanently remove SAS Data Quality Accelerator for Teradata or the SAS Embedded Process, you must remove any existing data quality stored procedures from the Teradata
database. The stored procedures are removed from the Teradata database by using the dq_uninstall.sh script. For more information about this script, see “Using the dq_uninstall.sh Script” on page 284.

It is not necessary to remove the QKB when upgrading or re-installing software. QKB deployment steps automatically overwrite an older version of the QKB when you install a new one. For information to replace the QKB, see “Updating and Customizing a QKB” on page 283 and “Locating the QKB” on page 279.

When you are permanently removing SAS Data Quality Accelerator for Teradata from the Teradata database server, follow whatever procedure is appropriate at your site for removing the QKB. The Teradata administrator also needs to remove data quality authorizations from the Teradata database in accordance with site procedures.

**Using the dq_uninstall.sh Script**

Note: To stop the embedded process, see Controlling the SAS Embedded Process in SAS 9.4 and SAS Viya 3.4 Programming Documentation / In-Database Products: User’s Guide. Stopping the SAS Embedded Process ensures that none of the accelerator files are locked when dq_uninstall.sh attempts to remove them.

The accelerator provides the dq_uninstall.sh shell script for removing the data quality stored procedures from the Teradata database. The dq_uninstall.sh script is located in the `/opt/SAS/ep/home/install/pgm` directory of the Teradata database server.

The dq_uninstall.sh script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_PASS=, DBC_SRVR=, and DBC_USER= variables.

Here is the syntax for executing dq_uninstall.sh:

```
./dq_uninstall.sh < -l log-path>
```

**log-path**

specifies an alternative name and location for the dq_uninstall.sh log. When this parameter is omitted, the script creates a file named dq_uninstall.log in the current directory.

Running dq_uninstall.sh disables the SAS Data Quality Accelerator for Teradata functionality and removes the data quality stored procedures from the database.
Nothing to Do Dialog

Error
After removing the software and attempting to re-install the software, this message is displayed:

Error: Nothing to do

Explanation
The directories that contain the software were deleted. However, the yum remove command was never run. In /var/log/yum.log, the last entry for the rpm message is installed.

Resolution
Clean up the yum repository by running the following command.

yum remove packagename
You can then re-install the software.

---

**PCA and KCLUS Procedures Were Not Found**

**Error**

ERROR: Procedure PCA not found

or

ERROR: Procedure KCLUS not found

**Explanation**
The installation was attempted on a system that was not completely cleaned up from a previous installation.

**Resolution**

Uninstall SAS/CONNECT by running the following command:

```bash
yum groups mark remove "SAS/CONNECT"
```

Re-install SAS/CONNECT by running the following command:

```bash
sudo yum groupinstall "SAS/CONNECT"
```

---

**Timeout Dialog**

**Error**

When running the deployment:

```bash
TimeoutError{error_message}
TimeoutError: Timer expired
INFO:  13:15:37 |
WARNING: Execution return code '2' is not the expected value '0' 13:15:37 |
INFO:  13:15:37 |
INFO: Updating deployment times data for step deploy_time with value 19 13:15:37 |
INFO:  13:15:37 |
WARNING: Ansible execution encountered failures
```

**Explanation**
The system failed to gather mount information.

**Resolution**

Perform one of the following actions:

- Set `/etc/mtab` as a link to `/proc/mounts` by running the following command:
  ```bash
  sudo ln -s /proc/mounts /etc/mtab
  ```

- Edit the `ansible.cfg` file and add or change the timeout value for Ansible as follows:
  ```bash
  timeout=number-of-seconds
  ```

  Deploy your software by running the Ansible playbook again.
From Any Browser: Connection Is Not Private

Explanation
The default self-signed certificates are not in the operating system truststore by default. The Apache Web Server is configured to use a certificate that is signed by this Certificate Authority (CA). When you open any SAS URL and navigate to the web server from a machine that does not have this CA in the truststore, you will receive the message Your connection is not private. The message does not indicate that there is any problem with the SAS deployment.

Resolution
SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see the Security section of the System Requirements chapter.

From Google Chrome: Connection Is Not Private

Error
When attempting to access SAS Viya software from Google Chrome, the following message is displayed:

Your connection is not private.

Explanation
If you have previously accessed a website using https, when you access the website again, Google Chrome automatically redirects to https.

Resolution
To reset Google Chrome so that it does not redirect to https:

1 In the Chrome address bar, enter this command:
   chrome://machine-name/#hsts

2 Under Query domain, in the Domain box, enter the name of the machine that was used in the URL that you were attempting to access.

3 Click Query to determine whether the machine is known to the browser.

4 If the machine is known to the browser, under Delete domain, enter that machine name in the Domain box. Click Delete.

The corrected URL should now work with the HTTP protocol.

Unable to Read a Key

Error
When running the deployment, the following message is displayed:

fatal: [deployTarget2]: FAILED! =>{"changed": false, "failed": true, "msg":
ERROR: Unable to read a key
ERROR: Unable to read a key"}
**Explanation**

Consul requires each machine to have a single, private IP address. It does not bind to a public IP address by default. A machine target that is specified in your inventory file has one of the following conditions:

- multiple network adapters that have been assigned private IP addresses.
- no private IP address.

**Resolution**

To confirm the cause of the failure, check the Consul logs for an entry that resembles the following:

Starting Consul agent...==> Error starting agent: Failed to get advertise address:
Multiple private IPs found. Please configure one.

The resolution is to configure an adapter for the Consul bind parameter in `/etc/sysconfig/sas/sas-viya-consul-default`

Note: This file was installed by the Ansible playbook. This problem can be avoided by specifying the consul bind adapter in the inventory file during deployment.

Locate the following section of the file:

# Consul option: -bind
# Specify the desired name of a network interface or IPv4 address.
export CONSUL_BIND_EXTERNAL=adapter-name

For adapter-name, supply the name of the adapter that Consul should use to locate the machine.

---

**After an Upgrade, One or More of the RabbitMQ Nodes Fails to Start**

**Error**

The log file contains the following message:

`=ERROR REPORT==== 16-Nov-2017::16:50:21 ===
Cluster upgrade needed but other disc nodes shut down after this one.

Please first start the last disc node to shut down.

Note: if several disc nodes were shut down simultaneously they may all show this message. In which case, remove the lock file on one of them and start that node. The lock file on this node is:

/opt/sas/viya/config/var/lib/rabbitmq-server/mnesia/rabbit@abc.unx.abc.com/nodes_running_at_shutdown

**Explanation**

The RabbitMQ cluster was not stopped or started in the correct order.

**Resolution**

1. Stop and restart the RabbitMQ nodes.
   a. Manually stop all nodes in the reverse order in which they were started during the upgrade.
   b. Manually restart the nodes in the order that is listed in the inventory file’s [rabbitmq] target group. You should wait for each rabbitMQ node to start completely before advancing to the next node in the list. If all nodes start successfully, skip to step 4.

2. For each rabbitMQ node that failed to start, connect to that target host and remove the lock file. Here is an example:
Manually restart the nodes in the order that is listed in the inventory file’s [rabbitmq] target group. You should wait for each rabbitMQ node to start completely before advancing to the next node in the list. If all nodes start successfully, skip to step 4.

3 On each failed target host, remove the rabbitMQ Mnesia database as follows:
   a On each failed target machine, remove the following directory and all contents:
      targethost$ sudo rm -R /opt/sas/viya/config/var/lib/rabbitmq-server/mnesia
   b On each failed target machine, remove the internal RabbitMQ cluster indicator file for SAS Viya:
      targethost$ sudo rm /opt/sas/viya/config/var/lib/rabbitmq-server/sasrabbitmq/sas.cluster.configured
   c If required, reset the rabbitMQ cluster password. As part of the original deployment, you were instructed to change the default RabbitMQ client password. When you remove the Mnesia database, the password is reset to the system default. To change the password for the cluster, on one of the RabbitMQ target machines, run the following command.
      targethost$ sudo /opt/sas/viya/home/bin

4 Rerun the playbook to continue the upgrade process after all rabbitMQ service failures have been resolved.

CAS Controller Start-Up Fails

Explanation
The administrator has changed the casenv_user, which causes the CAS controller start-up to fail.

Resolution
1 Edit the /opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig_tenant_default file, where tenant is either viya or the tenant name.
2 Change the line with restrictServerLaunch=old user to restrictServerLaunch=new user.
3 Rerun the playbook.

Invalid Value for Internal Host Name

Explanation
You might have an error in sitedefault.yml such as an incorrect value for internal.hostnames. However, you cannot correct the error and rerun the playbook. The sitedefault.yml file is used to set site-based values for properties during an initial deployment. On a subsequent run of the deployment playbook, properties that were previously set are not modified. The sitedefault.yml preserves any customer-based modifications to these values. If you rerun the playbook, only sitedefault.yml properties that have no value in the environment are applied.
Resolution
SAS Environment Manager is the preferred tool to modify the site-based property values. During deployment, you can also use the sas-bootstrap-config command with the --force option before you rerun the playbook. To modify the values, the --force option is required. Here is an example of how to modify the internal host name:

Note: Multiple lines are used for the export CONSUL_HTTP_TOKEN command to improve readability. However, in your environment, make sure that you enter the command on a single line.

```bash
source /opt/sas/viya/config/consul.conf
export CONSUL_HTTP_TOKEN=$(sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token)
cd /opt/sas/viya/home/bin/
./sas-bootstrap-config kv write config/application/zones/internal.hostnames correct-value-for-host-name
```

Project Creation Failed

Error
Project creation failed with: creatingProviderError
An unhandled provider creation error was detected.
Setting project to failed creation state

Explanation
This error is not an issue for the deployment. The configuration bootstrap process will retry until it is successful.

Resolution
Restart the data mining service.

Zypper Run Command Failed: Library Is Locked

Error
Zypper run command failed with return code 7

Explanation
The Zypp system management library is locked because PackageKit is running.

Resolution
Ensure that the PID in the error message has released the lock, and then run the playbook again.

Zypper Run Command Failed: Shared Vault Value Has Not Been Set

Error
The deployment receives error messages such as ERROR 11789 and Zypper run command failed with return code 7.

Explanation
The shared vault was not set before the deployment process was completed. The deployment did not fail and should succeed if there are no other errors.
Deployment of More Than Five Machines Takes More Time

Explanation
By default, Ansible uses five forked processes when deploying software, meaning that it can deploy up to five machines at the same time. Deploying more than five machines increases deployment time.

Resolution
SAS recommends changing the default setting for Ansible so that the number of forks and the number of machines match in the deployment.

1. Open the ansible.cfg file in the \sas_viya_playbook directory on the Ansible controller machine.
2. Under [defaults], add a forks parameter that is set to the number of machines in your deployment. Here is an example:

   [defaults]
   forks = 9

3. Save and close the ansible.cfg file.

For more information about tuning Ansible and forks, see your Ansible documentation.

Connection Reset by Peer Network Problem

Error
Deployments on Red Hat Enterprise Linux might receive a "Connection reset by peer" message when installing or applying updates for SAS Viya. This is usually indicative of networking issues.

Resolution
Deployments on Red Hat Enterprise Linux might receive a "Connection reset by peer" message during deployment or when applying updates for SAS Viya. This is usually indicative of networking issues.

To change the retries and timeout values for yum:

1. Open the /etc/yum.conf file as root or with sudo on the affected machine. Here is an example of a typical /etc/yum.conf file:

   [main]
   cachedir=/var/cache/yum
   keepcache=0
   debuglevel=2
   logfile=/var/log/yum.log
   exactarch=1
   obsoletes=1
   gpgcheck=1
   plugins=1
   metadata_expire=1800
2 If the `retries` and `timeout` variables are present, ensure that they are set to 20 and 120, respectively. If those variables are not present in the file, add them.

```conf
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800
retries=20
timeout=120
...  
```

3 Save and close the `/etc/yum.conf` file.

4 Repeat these steps for every affected machine.

If you continue to get the “Connection reset by peer” message, reopen the `/etc/yum.conf` file and revise these values upward.

---

**Internet Connectivity Problems**

Performing the pre-installation and installation tasks requires connection to the internet and, specifically, SAS repositories online. If you receive errors about connecting to the internet, you should work with your system administrator to correct them. The following steps are provided as guidelines for general areas of connectivity.

1 If you are deploying on Red Hat or an equivalent distribution, ensure that your firewall is open in order to allow access to the IP address of the content delivery servers that provide updates from Red Hat or Oracle. The IP addresses for content delivery services vary by region. For more information about the list of IP addresses, see one of the following websites:
   - Public CIDR Lists for Red Hat
   - [https://linux.oracle.com/](https://linux.oracle.com/)

   This website provides instructions for registering with the Oracle ULN.

2 Ensure that the firewall allows access to the SAS repositories.
   a In the same directory where you have saved the `.tgz` file, uncompress it.
      ```bash
tar xf SAS_Viya_playbook.tgz
      ```
      Be sure to leave a compressed copy of the `.tgz` file in the same location.
   b Run the following command from the playbook subdirectory (`/sas/install/sas_viya_playbook` if you used the recommended location for uncompressing your playbook).
      ```bash
curl -OLv --cert ./entitlement_certificate.pem --cacert ./SAS_CA_Certificate.pem
https://ses.sas.download/ses/repos/meta-repo/bigfile.bin
      ```
      If the firewall is set up correctly, the command successfully transfers the `bigfile.bin` file. If a connection fails, add any failing server to your firewall proxy whitelist and try the command again. Repeat this step until you successfully transfer the `bigfile.bin` file.
3 The firewall service should not be running while you deploy your software.

   a Create a list of the services that are running by performing the appropriate command from the list below.

   For Red Hat Enterprise Linux 6.x:
   ```bash
   sudo service --status-all
   ```

   For Red Hat Enterprise Linux 7.x and SUSE Linux:
   ```bash
   sudo systemctl list-unit-files
   ```

   b Use the following table to identify the name of the service that you should look for in the output from the
   command.

   Note: To identify the version of Linux that you are using, Red Hat Enterprise Linux and Oracle Linux
   users should see the `/etc/redhat-release` file. CentOS Linux users should see the `/etc/centos-release`
   file. SUSE Linux users should see the `/etc/os-release` file.

   **Table A6.1 Firewall Services by Linux Distribution and Version**

<table>
<thead>
<tr>
<th>Linux Version</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux earlier than 7.1</td>
<td>iptables</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 7.1 or later</td>
<td>firewalld</td>
</tr>
<tr>
<td>Oracle Linux earlier than 7.1</td>
<td>iptables</td>
</tr>
<tr>
<td>Oracle Linux 7.1 or later</td>
<td>firewalld</td>
</tr>
<tr>
<td>CentOS Linux earlier than 7.1</td>
<td>iptables</td>
</tr>
<tr>
<td>CentOS Linux 7.1 or later</td>
<td>firewalld</td>
</tr>
<tr>
<td>SUSE Linux</td>
<td>SuSEfirewall2.service</td>
</tr>
</tbody>
</table>

   If the firewall service from the table is listed in the output of the command, then the firewall is running
   and you should continue to the next step. Otherwise, you do not need to take any further actions.

   c To stop iptables, run the following commands:
   ```bash
   sudo service iptables stop
   sudo chkconfig iptables off
   sudo service ip6tables stop
   sudo chkconfig ip6tables off
   ```

   To stop firewalld, run the following commands:
   ```bash
   sudo systemctl stop firewalld.service
   sudo systemctl disable firewalld.service
   ```

   To stop SuSEfirewall2.service, run the following commands:
   ```bash
   sudo systemctl stop SuSEfirewall2.service
   sudo systemctl disable SuSEfirewall2.service
   sudo systemctl stop SuSEfirewall2_init.service
   sudo systemctl disable SuSEfirewall2_init.service
   ```
After an Upgrade, the Default Backup Schedule Is Not Working

Explanation
After performing an upgrade to SAS Viya 3.4, the default backup schedule might not work.

Resolution
Delete the default backup schedule and restart the backup service. For more information about deleting the backup schedule, see Jobs: How To in SAS Viya Administration: Jobs.

Setup Failed Because of Symbolic Link Problem with RabbitMQ

Explanation
Your deployment is using a symbolic link, and there is a symbolic link issue in Erlang 20.2.2. This issue has been fixed in Erlang 20.3.4.

Resolution
1. Edit the rabbitmq-env file in the /opt/sas/viya/home/lib/rabbitmq-server/sbin/ directory.
2. Search for RABBITMQ_HOME and set the value to the path of the RabbitMQ home directory. Here is an example:

RABBITMQ_HOME=/opt/sas/viya/home/lib/rabbitmq-server

After an Upgrade, an Additional Instance of Folder Services Might Exist

Explanation
In releases prior to SAS Viya 3.3, if the [ReportServices] inventory group was placed on a separate machine from the [CoreServices] group that contains the sas-folders microservice, after an upgrade to SAS Viya 3.3 or later, you might have an extra nonfunctional instance of the folders service.

Resolution
1. Uninstall the sas-folders package only from the [ReportServices] machine. Do not uninstall the sas-folders package from the [CoreServices] machine.

   sudo yum remove sas-folders

2. Deregister the sas-folders service only from the [ReportServices] machine. Do not deregister the instance from the [CoreServices] machine. For details about deregistering the service, see “Deregister Obsolete Services” on page 218.

Note: Ensure that you deregister only the old sas-folders service.