SAS® Viya™ 3.2: Deployment Guide

Introduction

About This Guide

Use this guide to deploy SAS Viya in your environment.

- The contents of this document are subject to continual updates. Make sure that you have the latest version of this document, which is available from the SAS Viya Install Center.
- To use this guide successfully, you should have a working knowledge of the Linux operating system and basic commands.
- Unless another situation is specifically cited, the information in this guide pertains to the software that you ordered.

How Deployment Works

Using Ansible — The Basics

To have the most control over your deployment and to simplify the deployment tasks, use Ansible.

- Ansible is a software orchestration tool 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 for how and where SAS Viya is deployed. In this guide, “to run the playbook” means to deploy SAS Viya.
- Your playbook is included in the Software Order Email (SOE) that SAS sends to your business or organization. The playbook contains other required files, such as your license file, which are customized for your order.
- Each time you run the playbook, Ansible automates a series of yum commands that securely access the latest SAS Viya software to which you are entitled. The software is downloaded from repositories that are maintained by SAS or from local repositories that you choose to mirror from the SAS repositories.

  Note: Yum is a software-package manager for Linux operating systems. SAS Viya is packaged in the RPM Package Manager (RPM) format, which simplifies installation, uninstallation, and upgrade tasks.
- To use Ansible, you must install it first. The machine on which you install Ansible is called the Ansible controller, and it must have access to the machines on which you plan to deploy SAS Viya. Instructions about how to install Ansible are provided in this guide.

Differences between a Deployment of SAS Viya and of SAS 9

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 RPM-based deployment model works with repositories that are native to your operating system, a SAS Software Depot is no longer required.

What Gets Deployed

Products and Supporting Components

This guide provides information for deploying the following products and supporting components:

- SAS Visual Data Mining and Machine Learning
- SAS Visual Analytics
- SAS Visual Statistics
- SAS Cloud Analytic Services (CAS), which is the analytics and license server for SAS Viya. CAS is licensed with most SAS Viya products.
- Data connectors, which enable you to configure connections to data sources such as existing SAS data, Oracle databases, and Hive data in Hadoop. Data connectors vary by the order.
- (Optional) SAS Event Stream Processing for CAS. The components are always deployed, but a separate license is required in order to enable them.
- (Optional) SAS Event Stream Manager, a separately licensed component to help manage multiple SAS Event Stream Processing environments.
- (Optional) SAS Event Stream Processing Analytics, a separately licensed component to enable advanced analytics and machine learning techniques in event stream processing models.

Deployment and User Interface Options

By default, a deployment using Ansible includes the installation of 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.” To accomplish a full deployment, you run the playbook without an installation option.

Although SAS recommends a full deployment, you can deploy a subset of user interfaces and functionality to meet the needs of your users.

- To support the data scientists and programmers who prefer to use SAS Studio only, you can run the playbook with a “programming-only” option. Understand that this type of deployment does not include SAS Home, SAS Environment Manager, and the complete suite of services that are included with a full deployment. And a programming-only deployment does not support the optional SAS Event Stream Manager product. Therefore, make sure that you are providing your users with the features that they require.

  Note: You can also deploy a programming-only environment on a single machine by deploying with yum. This type of deployment does not use Ansible.

- To support business and data analysts who prefer point-and-click user interfaces for SAS Viya, you can run the playbook with a “visual-only” option. This type of deployment does not include SAS Studio and SAS Foundation, which provide typical capabilities for running SAS code.

Deployment Scenarios

Advice about the Scenarios

- In most scenarios, it is assumed that Ansible is used to deploy software. Ansible is shown as installed on a separate machine, called the Ansible controller. For instructions about installing Ansible, see Install Ansible on page 32.
- Deploying the CAS server to a dedicated machine, or in a distributed method across multiple machines, might improve analytics-processing performance for users.

- When you deploy the CAS server, a role is assigned to each machine: CAS controller or CAS worker. If you deploy the CAS server to a single machine, the controller role is assigned. For a distributed CAS server, both roles are assigned.
- To specify the target machines that are shown in the multi-machine deployments, you edit the `hosts` file that is associated with the playbook.
- Data connectors 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.

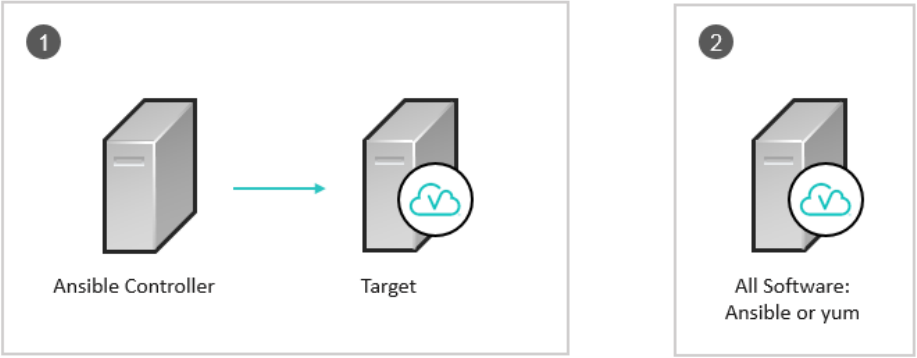
  Note: Data connectors vary according to the order.
- If you purchased SAS Event Stream Processing for CAS (optional), you must have an existing deployment of SAS Event Stream Processing 3.2 or a later version in your environment in order to provide data to the CAS actions that support streaming data. This independent installation of SAS Event Stream Processing must be running on a separate machine on which no CAS components are installed. The independent installation automatically enables SAS Event Stream Processing Studio.
- If you purchased SAS Event Stream Manager, installation of SAS Event Stream Processing on a separate machine is required.
- For deployments that use Hadoop, additional configuration is required to enable access to data in Hive or SASHDAT on Hadoop Distributed File System (HDFS). Additional configuration occurs after you deploy SAS software and the CAS controller and workers using Ansible.

Scenario 1: Single Machine

In this scenario, you can use Ansible or yum to deploy all SAS software to a single machine.

The following figure shows two options for deployment:

Two Examples of a Single-Machine Deployment



1	The Ansible machine, which is called the Ansible controller, deploys all SAS software to a different machine, the target node.
2	All SAS software is deployed to a single machine by using Ansible or yum. If Ansible is used, the target node is the same machine as the one where Ansible is installed and running.

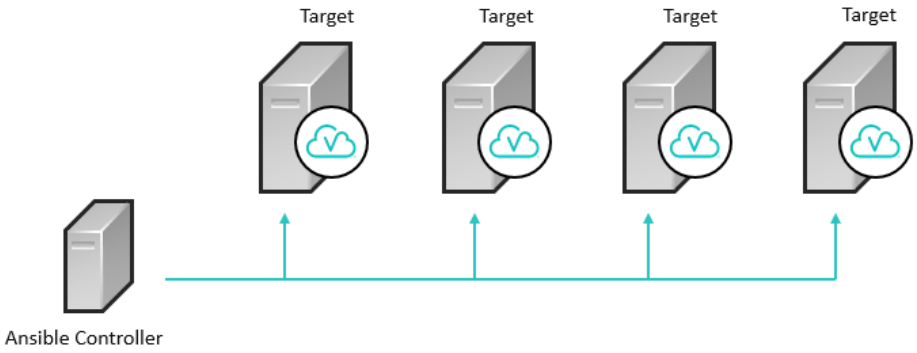
Scenario 2: Single Machine, Repeated

This scenario is useful for deploying SAS software in the following environments:

- development, testing, staging, and production environments
- the same deployment for different groups of users

The following figure shows an Ansible controller that is used to deploy the same SAS software to multiple target nodes:

Single Machine, Repeated

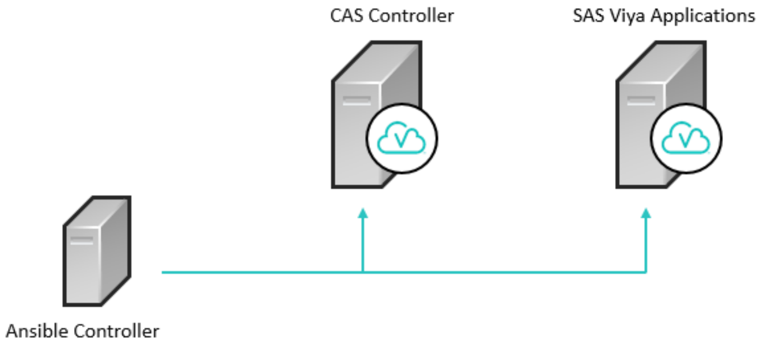


Scenario 3: Dedicated Machine for the CAS Server

The CAS server is the analytics server that the SAS procedures use for analytics processing. In this scenario, the CAS controller is deployed to a target node that is separate from other SAS software. During deployment, the CAS controller role is assigned to the target node.

> **TIP** Deploying the CAS server to a dedicated machine might improve analytics-processing performance over a deployment in which all SAS software, including the CAS server, is installed on the same machine.

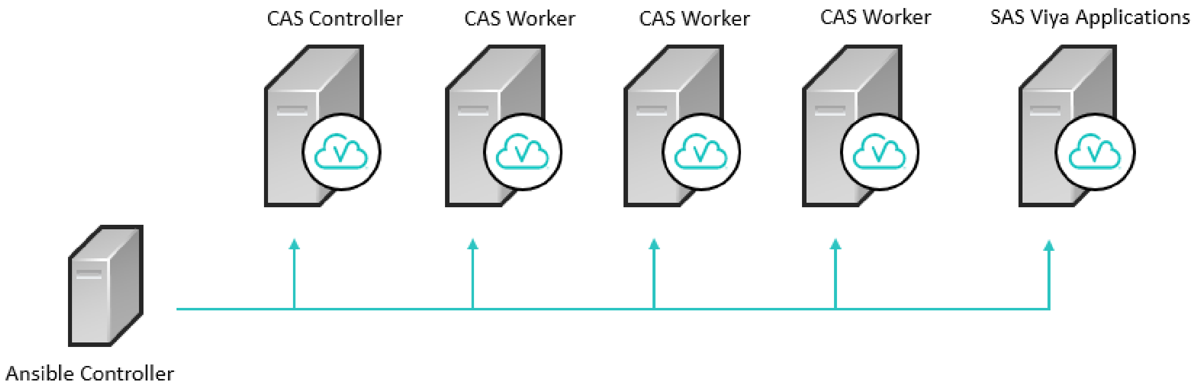
Dedicated Machine for the CAS Server



Scenario 4: Distributed CAS Server

In this scenario, CAS is deployed across two or more nodes in a clustered environment. An advantage of this scenario is that optimal processing can be achieved through massively parallel processing (MPP) for multiple users. During deployment, the CAS controller and CAS worker roles are assigned to the nodes.

Distributed CAS Server

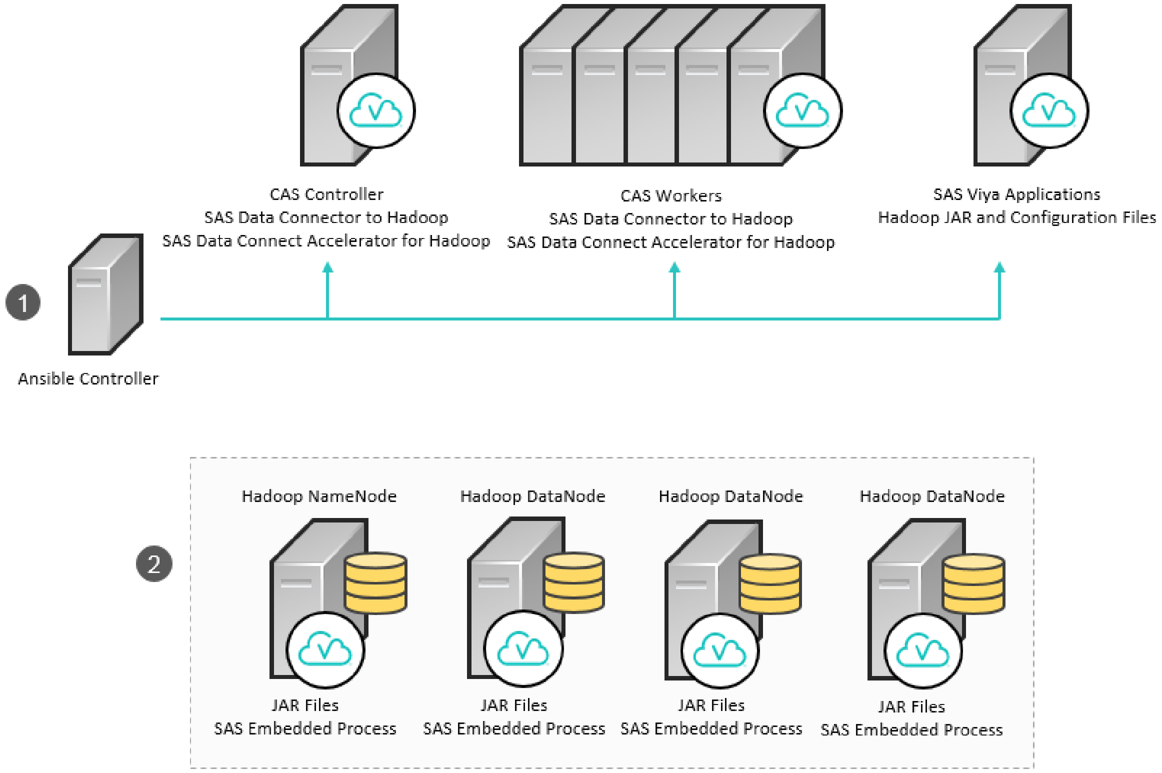


Hadoop Scenarios

Hadoop Scenario 1: Access to Data in Hive

The following scenario provides guidance for deploying SAS software to support access to data in Hive.

Distributed CAS Server with Access to Data in Hive



Overview of Deployment Steps

❶	Ansible is used to deploy SAS software: SAS Viya applications on one machine, and the CAS controller and worker nodes on the same machines as the data connectors. SAS Data Connector to Hadoop enables serial processing. SAS Data Connect Accelerator for Hadoop enables parallel processing between the CAS server and Hadoop. For details about installing with Ansible, see Installing SAS Viya with Ansible on page 35.
❷	Hadoop JAR files are installed. To enable parallel processing, the SAS Embedded Process is deployed to the Hadoop nodes. The installation of the JAR files and the deployment of the SAS Embedded Process are not performed using Ansible. For more information about installing and configuring this software, see Appendix D: Hadoop Deployment: Configuring SAS Access to Hadoop, SAS Data Connector to Hadoop and Optionally, the SAS Data Connect Accelerator for Hadoop on page 105.

Note: This deployment model provides parallel access to data in Hive through the use of the SAS Embedded Process and SAS Data Connect Accelerator for Hadoop. However, this type of deployment does not provide the ability to save data from CAS back to a Hive table. To save data from CAS, path-based caslibs must be used. A caslib is an in-memory space that is used to hold tables, to access control lists, and to provide data source information. All data that is available to CAS through caslibs and all operations in CAS that use data are performed with a caslib in place.

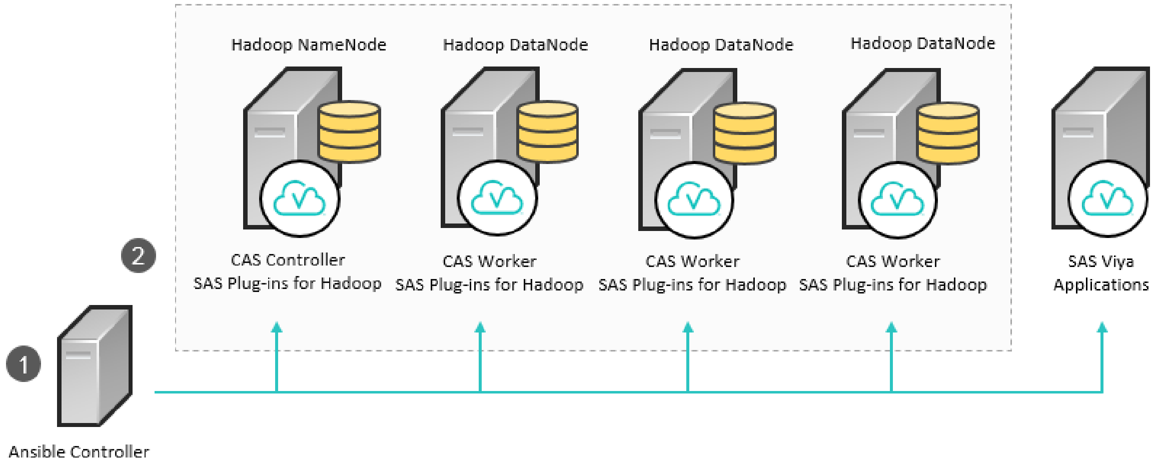
Hadoop Scenario 2: CAS SASHDAT Access to HDFS

This section describes the implementations that support loading and saving data in HDFS as SASHDAT tables. The SASHDAT file format supports SAS formats, is memory-efficient, and is optimal for use with in-memory processing that is provided by CAS. To enable users to save and load HDFS data, the SAS Plug-ins for Hadoop are configured on the Hadoop nodes. The SAS Plug-ins for Hadoop are required for this scenario.

CAS Deployed on Hadoop Nodes

In this implementation, the CAS controller is deployed to the NameNode, and the CAS workers are deployed to all DataNodes or to a subset of the DataNodes. An advantage of deploying to all DataNodes is that users do not need passwordless SSH to access the HDFS data store. If the resource demands for your Hadoop cluster leave capacity for SAS software, consider deploying CAS on Hadoop nodes.

CAS Deployed on All Hadoop Nodes



❶	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. For details about installing with Ansible, see Installing SAS Viya with Ansible on page 35.
❷	The Hadoop cluster is configured, and the SAS Plug-ins for Hadoop are configured on each Hadoop node on which CAS is deployed. For more information, see Appendix E: Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS on page 129.

Note: Consider the following information when deciding where to deploy the CAS controller 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 to load SASHDAT tables.
- If the CAS workers are deployed to a subset of the DataNodes, consider the following information:
  - Passwordless SSH is required to load SASHDAT tables.
  - When you save a SASHDAT table, data is still written locally, so 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.
- If you move the CAS controller to a DataNode, SAS recommends that you define the **`HADOOP_NAMENODE`** environment variable.

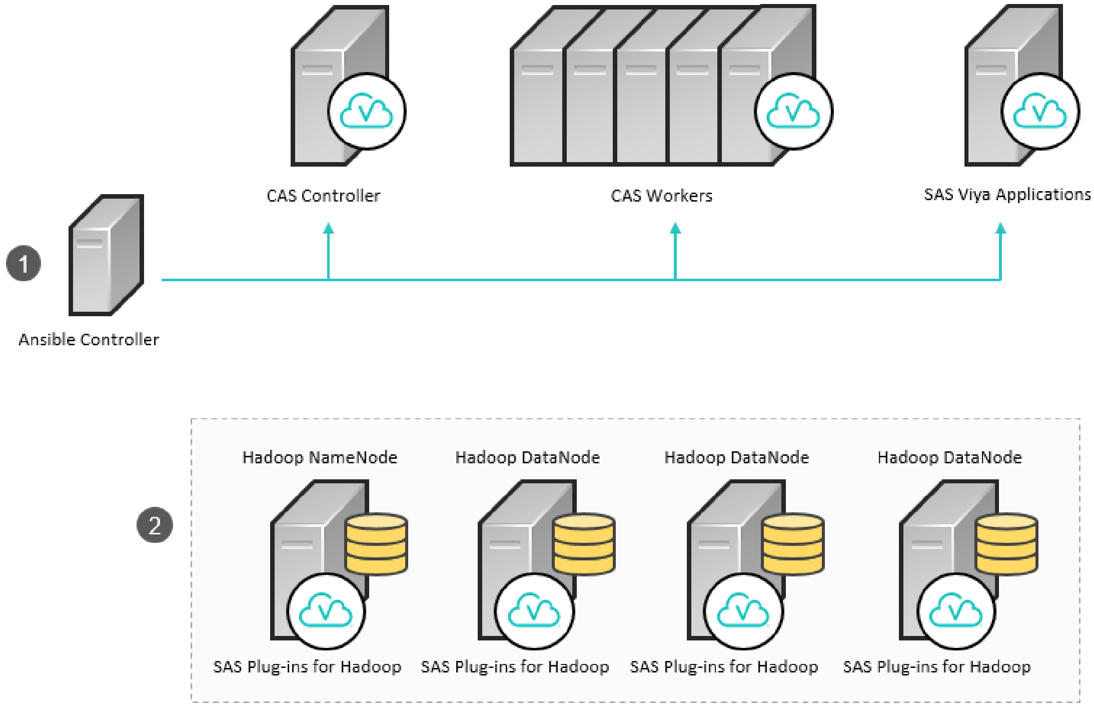
For information about CAS environment variables, see *SAS Viya Administration: SAS Cloud Analytic Services*.

Remote Access to HDFS

In this implementation, the CAS controller and the CAS workers are deployed to machines that are not part of the Hadoop cluster. This implementation enables a remote, parallel connection between a Hadoop cluster and a set of machines that is dedicated to CAS.

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

Remote Access to HDFS



Overview of Deployment Steps

1	Ansible is used to deploy SAS software: SAS Viya applications on one machine, and the CAS controller and the CAS workers on other machines. For details about installing with Ansible, see Installing SAS Viya with Ansible on page 35.
2	The Hadoop cluster is configured, and the SAS Plug-ins for Hadoop are configured on each Hadoop node. For more information, see Appendix E: Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS on page 129.

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

Contact SAS Technical Support

Technical support is available to all customers who license SAS software. However, we encourage you 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 call, 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

Each target machine in your SAS Viya deployment must have both of the following attributes:

- Static IP address

  The Consul 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 adapters have assigned private IP addresses. You will also see an error during the deployment if the machine has only a public IP address. For more information, see Appendix G: Deployment Troubleshooting on page 145.

- 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 SAS Viya 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.

If you plan to deploy SAS Viya on multiple machines, make sure that the clock time is synchronized across all of them.

Hardware Requirements for SAS Visual Analytics and SAS Visual Statistics

Use the guidelines in this section to select machine targets for your SAS Viya deployment that includes SAS Visual Analytics and SAS Visual Statistics.

SAS strongly recommends consulting with a sizing expert to obtain an official hardware recommendation that is based on your estimated SAS workload and number of users. The sizing information provided here is not intended as a substitution for expert advice. 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.

SAS Viya components can be installed on a single machine or on multiple machines. Verify that the fully qualified domain name on each machine in your deployment is 64 characters or fewer in length. This requirement is included in prerequisite checking. The installation files are automatically downloaded to the `/var/cache/yum` directory. This directory therefore requires sufficient available disk space to accommodate the installation packages. Verify that at least 10 GB of disk space are available for SAS Viya installation.

Additional space for logs is also required in `/opt/sas/viya`. For more information, see Log File Space Requirements on page 17.

The following table contains minimum recommendations for a single-machine deployment, based on the deployment type (full or programming-only). For a visual-only deployment, SAS recommends following the guidance for a full deployment because these deployments require similar resources:

Requirements for Single-Machine Deployment, Full (All User Interfaces)

Item	Minimum Level
CPU	Intel Xeon CPU with 4 cores x86 architecture with a minimum speed of 2.6 GHz
Memory	64 GB of RAM Memory clock speed of 1600 MHz
Disk Space and Speed	2 x 300 GB 10,000 RPM

Requirements for Single-Machine Deployment, Programming Interface Only

Item	Minimum Level
CPU	Intel Xeon CPU with 4 cores x86 architecture with a minimum speed of 2.6 GHz
Memory	32 GB of RAM Memory clock speed of 1600 MHz
Disk Space and Speed	2 x 300 GB 10,000 RPM

The general rule to follow for a multi-machine deployment is to apply similar minimum guidelines for each target machine. However, if you distribute components across multiple machines and isolate selected components, such as separating the Report Services server from the Core Services server, you can typically reduce the RAM on each of those machines to 32 GB.

Note: The CAS server can be installed on a single machine, or the two CAS machine roles can be installed on separate machines (“distributed CAS”). In a distributed CAS deployment, hardware specifications for the CAS controller and CAS workers are the same as those for the CAS server.

An additional machine can be used as a “thin client” from which end users can access the product user interface. This machine requires minimal processing power and storage space and can run on Windows or UNIX.

Hardware Requirements for SAS Visual Data Mining and Machine Learning

Use the guidelines in this section to select machine targets for your SAS Viya deployment that includes SAS Visual Data Mining and Machine Learning, SAS Visual Analytics, and SAS Visual Statistics.

SAS strongly recommends consulting with a sizing expert to obtain an official hardware recommendation that is based on your estimated SAS workload and number of users. The sizing information provided here is not intended as a substitution for expert advice. 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.

SAS Viya components can be installed on a single machine or on multiple machines. Verify that the fully qualified domain name on each machine in your deployment is 64 characters or fewer in length. This requirement is included in prerequisite checking.

The installation files are automatically downloaded to the `/var/cache/yum` directory. This directory therefore requires sufficient available disk space to accommodate the installation packages. Verify that at least 10 GB of disk space are available for SAS Viya installation.

Additional space for logs is also required in `/opt/sas/viya`. For more information, see Log File Space Requirements on page 17.

The following table contains minimum recommendations for a single-machine deployment, based on the deployment type (full or programming-only). For a visual-only deployment, SAS recommends following the guidance for a full deployment because these deployments require similar resources:

Requirements for Single-machine Deployment, Full (All User Interfaces)

Item	Minimum Level
CPU	Intel Xeon CPU with 16 cores x86 architecture, with a minimum speed of 2.6 GHz Hyper-threading is recommended.
Memory	64 GB of RAM Memory clock speed of 1600 MHz
Disk Space	2 x 600 GB, 10K RPM
Adapter and Network	Dual 10 Gb NIC 10 Gb Ethernet
I/O Throughput	400 MB/sec

Requirements for Single-machine Deployment, Programming Interface Only

Item	Minimum Level
CPU	Intel Xeon CPU with 16 cores x86 architecture, with a minimum speed of 2.6 GHz Hyper-threading is recommended.
Memory	32 GB of RAM Memory clock speed of 1600 MHz
Disk Space	2 x 600 GB, 10K RPM
Adapter and Network	Dual 10 Gb NIC 10 Gb Ethernet
I/O Throughput	400 MB/sec

The general rule to follow for a multi-machine deployment is to apply similar minimum guidelines for each target machine. However, if you distribute components across a multiple machines and isolate selected components, such as separating the Report Services server from the Core Services server, you can typically reduce the RAM on each of those machines to 32 GB.

Note: The CAS server can be installed on a single machine, or the two CAS machine roles can be installed on separate machines (“distributed CAS”). In a distributed CAS deployment, hardware specifications for the CAS controller and CAS workers are the same as those for the CAS server.

An additional machine can be used as a “thin client” from which end users can access the product user interface. This machine requires minimal processing power and storage space and can run on Windows or UNIX.

Log File Space Requirements

SAS Viya software is installed in the `/opt` directory on each target machine. Additional space for logs is also 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 40 GB. SAS recommends using monitoring tools to ensure that none of the locations used by the deployment fills up without warning.

If disk space is limited, SAS recommends creating symbolic links from the installation or log directories to the partitions where plenty of disk space is available (at least 40 GB). For example, you can create a symbolic link from the SAS Viya log space (`/var/log`) to a directory that has additional free space:

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

The 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. The Apache documentation provides guidance:

- Apache 2.2 log rotation
- Apache 2.4 log rotation

(Optional) High-Availability Requirements

Some SAS Viya software components, including SAS Infrastructure Data Server and the SAS Studio user interface, can be deployed with multiple instances to support high availability.

If you have more than one machine target for SAS Infrastructure Data Server (the required PostgreSQL database), all of these machines must have the same home directory for the sas user account. The requirement is relevant during security configuration as part of the installation. Therefore, if you use an account other than sas during playbook configuration, the requirement for consistent home directories applies to that user account instead.

SAS Studio users can set preferences and store projects, and 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 in a high-availability configuration:

- Set up a shared file system and configure SAS Studio to use a shared drive in that file system.
- Enable file sharing for Home directories on all hosts where SAS Studio is installed.

Operating System Requirements

Supported Operating Systems

For the full list of supported operating systems, see the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-operating-systems.html

Linux Prerequisites

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. To use Ansible for the deployment, the Ansible controller machine 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/.

If you have enabled Security-Enhanced Linux (SELinux) in your environment, you must enable permissive mode on all of the target machines in your deployment. For more information, see Configure SELinux on page 29.

The typical Linux installation includes all 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 are required:

- glibc 2.12
- libpng (on Red Hat Enterprise Linux 6.*x* or the equivalent)

  libpng12 (on Red Hat Enterprise Linux 7.*x* or the equivalent)
- libXp
- libXmu
- net-tools
- the numactl package
- the X11/Xmotif (GUI) packages
- xterm

On Linux 7.*x*, verify that the systemd package on each machine is at version 219-30 or later. Run the following command:

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

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

```
$ yum update systemd
```

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 Linux Requirements for SAS Event Stream Processing for CAS

The information in this topic is relevant for users of SAS Event Stream Processing for CAS. The SAS Event Stream Processing Engine 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 are compiled using the following compiler options:

```
-D_REENTRANT
```

```
-D_THREAD_SAFE
```

All of 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 4.x 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 Requirements

The Java Runtime Environment (JRE) must be installed on every machine in your deployment. Only the JRE is required. The full JDK is not required. For a list of supported JRE distributions, see the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-jre.html.

SAS Viya supports alternative distributions of the JRE, such as Azul Zulu or IBM SDK, Java Technology Edition. For those vendors, the supported versions are the same.

The playbook checks for a preinstalled version of Java that meets or exceeds the requirements. If it 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 your vars.yml file before you run your playbook.

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 moving to OpenJDK or Oracle JRE as a remedy. You can determine the current Java version on a Linux machine by running the following command:

```
java -version
```

Apache httpd

The deployment automatically installs Apache httpd from Linux repositories if it is not detected on the machines that you designate as targets for the HTTP Proxy installation. Apache httpd is required to create the Apache HTTP Server, which provides security and load balancing for multiple SAS Viya components.

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 the httpd components on multiple servers. A load balancer is then required to provide high availability for the Apache HTTP Server. Otherwise, you risk a SAS Viya outage if one httpd instance becomes unavailable.

To install redundant instances and to specify the machine target for the Apache HTTP Server, use the [httpproxy] host group in the Ansible inventory file. For more information, see Assign the Target Machines to Host Groups on page 36. If you have your own installation of Apache httpd, specify that machine for the [httpproxy] host group so that the deployment can add required software to it. However, the Apache httpd component is required for internal communications among SAS Viya components. Therefore, replacing the Apache components that are installed during the SAS deployment is not supported.

In addition, the Apache HTTP Server must be dedicated to a single SAS Viya deployment.

Data Source and Storage Requirements

Overview of Data Warehouse and Storage Requirements

If you purchased SAS Event Stream Processing for CAS, a separately licensed instance of SAS Event Stream Processing 3.2 or a later version is a required data source. If you do not already have an installation of SAS Event Stream Processing, purchase the product separately and install it on a separate machine in order to obtain a valid license. The separate license also enables SAS Event Stream Processing Studio, which is the product user interface.

You can install software to enable data retrieval from a Hadoop data store and from various data storage appliances. Depending on one or more of your data sources, you might also install one or more data connectors and a data connect accelerator on your CAS controller and CAS workers.

Depending on your data source, you might be required to install the following additional software on your CAS machines:

- The database client for your associated database software. You might need to install the database client on the CAS controller.
- Drivers or other requirements for the data connector to be used with your data source. The appropriate data connector is installed by the SAS deployment onto the CAS controller and all CAS workers. You must install any drivers or other required software on the CAS controller.

Refer to the section that corresponds to your data connector or data connect accelerator for additional system requirements that apply to the CAS controller and CAS workers.

Data Encoding Requirement

UTF-8 is the only SAS session encoding supported by SAS Viya. If your DBMS encoding is non-UTF-8, the SAS software typically converts the data to UTF-8 to work with CAS processes. Additional settings, such as changes to environment variables, might be required if you are attempting to use a database with non-UTF-8 encoding.

You can also use SAS/CONNECT to transfer and automatically convert data from a non-UTF-8 encoded SAS session to the UTF-8 encoded SAS Viya environment. For information about how to convert data from non-UTF-8 to UTF-8, see Migrating Data to UTF-8.

Supported Data Sources

SAS Viya supports the following data sources:

- Hadoop
- Impala
- Data sources accessible with an ODBC driver
- Oracle
- PC files, which support the following file extensions:
  - `.jmp`

  - .spss
  - .stata
  - .xlsx or .xls
- PostgreSQL
- Teradata

Note: Each data source also requires a data connector and possibly a SAS data connect accelerator. In some cases, a data connector might have individual system requirements.

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

SAS Viya also supports the following data sources, which use SAS data connectors that are automatically included with CAS and are not separately licensed or configured:

- SASHDAT on HDFS
- LASR Analytic Server (SAS 9.4)
- SAS data sets

SAS Viya also supports CSV files, which do not require a data connector. CSV files can be accessed directly.

Hadoop Requirements

Supported Releases of Hadoop Distributions

SAS Viya supports multiple third-party distributions of Hadoop.

Note: If you upgrade your Hadoop version and have already deployed SAS Viya with SASHDAT, then 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 the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/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 data connector setup is required.
- For Hive, SAS Data Connector to Hadoop and possibly SAS Data Connect Accelerator for Hadoop are required. The SAS Data Connectors do have individual system requirements, which are discussed in later sections.

Note: Apache Hadoop 0.23, 2.4.0, and 2.7.1 and later versions are supported only for HDFS caslibs. These are caslibs that are used when saving SASHDAT files to an HDFS path.

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: http://support.sas.com/resources/thirdpartysupport/v94/hadoop/alternative-hadoop-distributions.html. The same policy that applies to SAS 9.4 also applies to SAS Viya.

Requirements to Import Data from SAS 9.4

SAS/CONNECT is required in the environment to move data from other SAS deployments and operating systems into 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. You must order it separately. If you order SAS/CONNECT, the required commands to install it are automatically included in your playbook.

Requirements for SAS Data Connector to Hadoop

SAS Data Connector to Hadoop is included in SAS/ACCESS Interface to Hadoop (on SAS Viya).

In addition to a supported Hadoop distribution, SAS Data Connector to Hadoop also requires the following:

- Hive

  In addition, Hive (specifically, the hadoop_extract.sh script) requires the following software:

  - Oracle JRE version 1.8 or a later version
  - Python, strace, and wget (which are included with Linux)
- MapReduce
- YARN
- HCatalog for processing non-delimited Hive file types.

Requirements for SAS Data Connect Accelerator for Hadoop

SAS Data Connect Accelerator for Hadoop is included in SAS In-Database Technologies for Hadoop (on SAS Viya).

SAS Data Connect Accelerator for Hadoop requires the following:

- Hive
- MapReduce
- YARN
- HCatalog for processing non-delimited Hive file types.
- SAS Embedded Process for Hadoop that is deployed. For more information, see Deploy the SAS Embedded Process for Hadoop for Parallel Processing on page 111.

  Note: In order to load data in parallel with the SAS Embedded Process, the CAS controller and each CAS worker must have an IP address that can be routed to externally from the SAS Embedded Process nodes.

Requirements for SAS Data Connector to Impala

SAS Data Connector to Impala is included in SAS/ACCESS Interface to Impala (on SAS Viya).

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

Requirements for SAS Data Connector to ODBC

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

For information about ODBC support, see the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html.

Requirements for SAS Data Connector to Oracle

SAS Data Connector to Oracle is included in SAS/ACCESS Interface to Oracle (on SAS Viya).

You must install the Oracle client on the CAS controller.

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

Requirements for SAS Data Connector to PostgreSQL

SAS Data Connector to PostgreSQL is included in SAS/ACCESS Interface to PostgreSQL (on SAS Viya).

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

Requirements for SAS Data Connector to Teradata

SAS Data Connector to Teradata is included in SAS/ACCESS Interface to Teradata (on SAS Viya).

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

Requirements for SAS Data Connect Accelerator for Teradata

SAS Data Connect Accelerator for Teradata is included in SAS In-Database Technologies for Teradata (on SAS Viya).

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

Note: SAS Embedded Process to Teradata is required. In order to load data in parallel with the SAS Embedded Process, the CAS controller and each CAS worker must have an IP address that can be routed to externally from the SAS Embedded Process nodes.

User and Group Requirements

Set Up the User Account that Deploys the Software

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

- Administrator privileges for the Linux machine where the deployment is launched.
- A home directory that is readable by the user accounts that are required for the deployment (cas and sas).

  If you have more than one machine target for SAS Infrastructure Data Server, make sure that all of these machines have the same home directory for the installation user account.
- Super-user (sudo) access.

  Run the following command to verify that your user ID is included in the sudoers file:

  ```
  sudo -v
  ```

  As an alternative, verify your sudoers privileges with the following command:

  ```
  sudo -l
  ```

  Note: The ability to start a shell (via the `!SHELL` entry in some sudoers files) as root is not required.

Set Up the cas Account

A user account and a group are required for the deployment. As part of the prerequisite check, 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. Because the cas user account has the Super User role in the visual administration interface (SAS Environment Manager), it also has unrestricted access to CAS. It functions as a back-end service account.

- If you plan to use the customized script to perform a yum deployment, the required user account and the group are created automatically.
- If the policies in your environment do not allow for the creation of a cas user and a sas group, identify alternative but equivalent values for casenv_user and casenv_group in the vars.yml file before you run the playbook. For more information, see Accept the Passwordless SSH Default on page 41.

Perform the following steps to set up the required group and user accounts:

1. Create the **sas** group, or its equivalent, if one does not exist.
2. Create the user account. The recommended user name is `cas`. Assign the user account to the sas group or an equivalent group.
3. Generate an SSH public key for the cas user in the **`$HOME/.ssh`** directory. For more information, see Define the Machines in the Deployment on page 36.
4. Verify that the user account is present on each node where a CAS component is running. Also, verify that the account has a consistent UID and GID on all machines 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.

When the deployment has completed, the cas user (or the equivalent user that you configured in vars.yml) might not have logon access to SAS Studio or to CAS Server Monitor. Use an LDAP account for this purpose. For more information, see Set Up Administrative Users on page 58.

Set Up Additional User Accounts

A few other user accounts are required in order to configure and run the software after the deployment has completed. For a complete list and description of these accounts, see User Accounts (Reference) on page 25. The topics in this section provide an overview of the requirements that apply to user accounts for a programming-only and visual-only deployment. However, a full deployment is recommended. In a full deployment, prepare accounts for both programmers and non-programmers in order to access all user interfaces.

Set Up Accounts for Non-Programmers

Note: This section does not apply to a programming-only deployment.

Non-programmers will not log on to SAS Studio. Instead, they will use SAS Environment Manager. The following requirements apply to these user accounts:

- Each user must be able to authenticate to your LDAP provider.
- If you plan to configure front-end single sign-on (SSO), make sure that each user can authenticate to the associated provider. This is an additional requirement rather than a replacement for the preceding requirement. For more information, see SAS Viya Administration: Authentication.
- Any user of the visual interface who also wants to authenticate to SAS Studio must also have a valid host account on the SAS Studio host. Also, the passwords for these accounts must match.

Set Up Accounts for Programmers

Note: This section does not apply to a visual-only deployment.

Programmers will not log on to SAS Environment Manager. Instead, they will log on to SAS Studio. Account requirements for programmers resemble the requirements for the cas account. However, the following factors apply to these users:

- The accounts that exist only on your LDAP server cannot log on to SAS Studio by default.
- Each SAS Studio user must have a valid host (operating system) account on the machine on which the SAS Studio web application runs.
- SAS Studio users also require an LDAP account in order to access CAS. The passwords must match.
- Each user must log on with an account that has a home directory.

Prepare to Bind to LDAP

LDAP is required for the visual interface. It is not required in a programming-only deployment.

To support the visual interface, SAS Viya must have Read access to your LDAP provider. To bind to the LDAP server, SAS Viya requires a userDN and password. LDAP anonymous binding is not supported.

LDAPS is supported, but the required certificates are not configured automatically by the deployment.

Configuring SAS Viya to access your LDAP provider is a post-deployment task. For more information, see Configure Your Environment with SAS Environment Manager on page 57.

User Accounts (Reference)

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

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:

Account Name and Group	Parameters	Purpose
sas; member of sas group	Non-logon service account without user restrictions. No password. You can add a password after installation, if necessary. The default user name is required. The sas group is an administration group, not a general user group.	Required for the installation, and created automatically. The installation process sets user and group ownership permissions on all of the installation files. This user must exist to enable ownership. After the installation has completed, this user account enables required components to run, including the web application server for SAS Environment Manager. 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.

Account Name and Group	Parameters	Purpose
cas; member of sas group	Process owner of CAS processes. 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. The cas user must be able to connect from the CAS controller machine to each CAS worker without providing a password. If the CAS server is running in an analytic cluster environment (with multiple CAS workers), passwordless SSH can be configured by the Ansible playbook. For more information, see Set Up Passwordless SSH for CAS on page 41. The “cas” 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.	Required for managing and enabling Cloud Analytic Services (CAS). If you are using Ansible to deploy SAS Viya, create this user account and add it to the sas group before you start the deployment. (If you are using the customized script that is described in Appendix B: Deploying with Yum, the deployment creates the user account for you.) This user corresponds to the CAS (Superuser) role in the CAS administration interface, CAS Server Monitor.
sasboot	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 that; however, it provides a “backdoor” option in the event that your identity provider becomes unavailable. For more information, see *SAS Viya Administration: Identity Management*.	Administrator account that is used for preliminary logon to the visual administration interface. **Note:** This account is not recognized by SAS Studio, the programming interface.
user account for SAS Studio access	A user account that is defined on the operating system of the machine where SAS Studio will be installed.	User account that is used for preliminary logon to the programming administration interface.

Note: In addition to the cas account that will own CAS processes after the deployment has completed, a user account named cas is created automatically by the deployment. This user is the file owner of many of the files that are copied to the machine by the installation RPMs.

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.

Security Requirements

Requirements to Enable Encryption

The deployment provides a default level of encryption for data at rest (stored data) and data in motion (transmitted data). You can “harden” the deployment by blocking external connections to port 80, by adding custom certificates to the self-signed certificates that the deployment provides on all machines, and by upgrading the security protocol and ciphers that are enabled by default. Also, you can configure TLS encrypted connections between CAS workers, and you can use HTTPS for access to SAS Viya user interfaces from a web browser.

For more information about configuring additional security settings after the deployment has completed, see Encryption in SAS Viya.

Transport Layer Security (TLS) Software Requirements for the SAS Embedded Process

If you are using the SAS Embedded Process, you can secure data transfer between your cluster and CAS. To use 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.

Client Requirements

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

Web Browsers for SAS Viya User Interfaces

The SAS Studio and CAS Server Monitor user interfaces have identical web browser requirements. However, the visual user interfaces for SAS Visual Analytics, SAS Visual Statistics, and SAS Visual Data Mining and Machine Learning 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 the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-web-browsers.html.

Mobile Platform Support

Support for mobile devices is not yet available for all SAS Viya user interfaces. For information about mobile device support, see the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/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.

Ansible Controller Requirements

Deployment using Ansible is optional. However, Ansible is recommended for multi-machine deployments.

For information about supported Ansible versions, see the following website: https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-operating-systems.html
.

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 that uses Ansible rather than yum, Ansible and all SAS software are installed on the Ansible controller. For more information, see Install Ansible on page 32.

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 between all machines in the deployment and from the controller to the SAS yum repositories. For more information, see Firewall Considerations on page 31.

The Ansible controller must be connected to the Red Hat Network. Oracle Linux machines require an Oracle Linux support subscription. With Oracle Linux 6 Update 5 or later, the ULN registration procedure automatically registers each host with the latest channels for the base repository and the Unbreakable Enterprise Kernel Release 3 (UEK R3). Oracle Ksplice, which provides automatic security updates, is supported, but is not required for SAS Viya.

Pre-installation Tasks

Make Sure That You Have the Required Files

When you order SAS software, SAS sends a Software Order Email (SOE) to your business or organization that includes information about the software order. The SOE directs you to save its attached .tgz file and the license file to a directory on your Ansible controller. The recommended location is **/sas/install**. If you have not already done so, you must save those files before performing any of the steps in this section.

In the same directory where you have saved the .tgz file, uncompress it.

```
tar xf SAS_Viya_playbook.tgz
```

A sas_viya_playbook subdirectory is added, containing the following files:

- a second copy of the license file
- the entitlement_certificate.pem and SAS_CA_Certificate.pem files
- the customized_deployment_script.sh file
- the files that make up the SAS Viya playbook, referred to in the rest of this guide as “the playbook”

Configure SELinux

If you have enabled Security-Enhanced Linux (SELinux) in your environment, you must enable permissive mode on all of the target machines in your deployment. You can run the following command to check whether SELinux is enabled on an individual system:

```
sudo sestatus
```

For all Linux distributions, if a mode that is not permissive is returned, run the following commands:

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

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.

Process	Required Port	Notes
HTTPD	80 (internal) 443 (external) See note below.	
default Erlang Port Mapper Daemon (epmd) port	4369	
SAS Infrastructure Data Server	5430–5439	For a single server deployment with no failover, ports 5430-5432 must be opened. Additional standby nodes each get the next available port number sequentially up to 5439.

Process	Required Port	Notes
CAS Server Starting Port	5570	Used by clients to make binary connections to CAS.
CAS Communicator Port	5580	
default SAS Messaging Broker AMQP client access port	5672	
SAS Studio	7080 (if you are performing a visual-only or full deployment, the deployment will use ephemeral ports, so no port needs to be reserved)	Not required for SAS Visual Investigator.
SAS Configuration Server	8300–8309, 8500, 8501	SAS uses HashiCorp Consul as its configuration server.
Object Spawner	8591	
CAS Server Monitor	8777	Used by clients to make REST HTTP calls to CAS, as with the Python REST interface.
Elasticsearch	9200	Required only for SAS Visual Investigator.
default SAS Messaging Broker management web console port	15672	
SAS/CONNECT Spawner	17551	
SAS Cloud Analytic Services Server	19990-19999	
default SAS Messaging Broker clustering port	25672	

Note: In order for the machines in your deployment to communicate appropriately, port 80 on the machine where HTTPD is installed must be reachable by any machine on which SAS software is deployed. HTTPD for SAS Viya must not be installed on a machine where port 80 is used by other software. You can assign HTTPD to a different machine using the [httpproxy] host group as described at Assign the Target Machines to Host Groups on page 36.

In order to secure web access to your SAS Viya software, only port 443 (HTTPS) should be open externally on the machine where HTTPD is installed.

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 and are allocated automatically from within this range. 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 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:

```
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:

```
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 that are included in the 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:

```
sudo sysctl -w net.ipv4.ip_local_reserved_ports="ports-or-port-range"
```

Here is an example:

```
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.

4 Add an entry to the /etc/sysctl.conf file to make your changes permanent. Here is an example:

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

Firewall Considerations

The following steps should be performed on each machine in the deployment.

1 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 from 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/

  This website provides instructions for registering with the Oracle ULN.

2 Ensure that the firewall allows access to the following yum repositories that are hosted by SAS so that content can be delivered for deployment:

- https://ses.sas.download/
- https://bwp1.ses.sas.download/
- https://bwp2.ses.sas.download/
- https://sesbw.sas.download

3 Determine if iptables or firewalld are running:

```
sudo service --status-all
```

If you are using a version of Red Hat Enterprise Linux, Oracle Linux, or CentOS that is earlier than version 7.1, look for the status of iptables. If you are using any other version of Linux, including versions of Red Hat Enterprise Linux, Oracle Linux, or CentOS that are later than version 7.1, look for the status of firewalld.

If iptables or firewalld is running, go to step 4.

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 users should see the **`/etc/centos-release`** file.

4 To stop iptables, perform the following commands:

```
sudo service iptables stop
sudo chkconfig iptables off
sudo service ip6tables stop
sudo chkconfig ip6tables off
```

To stop firewalld, perform the following commands:

```
sudo service firewalld stop
sudo chkconfig firewalld off
```

Note: For more information about the service utility, look at the administration documentation available at the Red Hat Customer Portal (https://access.redhat.com/).

Configure the Use of a Proxy Server

If your organization uses a proxy server as an intermediary for Internet access, you should configure yum to use it. The steps to configure the **/etc/yum.conf** file vary by operating system. Refer to your vendor documentation for details.

Enable the Yum Cache

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**.

```
keepcache = 1
```

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

Install Ansible

Ansible is third-party software that provides automation and flexibility for deploying software to multiple machines. If you decide to use Ansible to deploy your software, use the information in this section to install and configure Ansible.

Installation Steps

Follow these steps to install Ansible on a Linux machine that runs SAS Viya. These steps assume that you have sudo access to the machine where you are installing Ansible.

1 Run the following commands:

Note: For improved readability, the third command occupies two lines. However, make sure that you enter the command on a single line.

```
sudo yum install -y epel-release
sudo yum install -y gcc automake openssl-devel python-devel libffi-devel
sudo yum install -y python-crypto python-paramiko python-keyczar python-setuptools python-pip
     python-six python-pip
sudo yum install -y python-virtualenv
mkdir work && cd work
virtualenv deployment
source deployment/bin/activate
pip install ansible==2.2.1
```

2 Confirm that the correct version of Ansible is installed.

```
ansible --version
```

Test Your Ansible Installation

To test that Ansible has been installed correctly, run the following command:

```
ansible localhost -m ping
```

If the command runs successfully, Ansible is ready for use.

Perform Linux Tuning

Set the MaxStartups Variable

The MaxStartups variable specifies the maximum number of concurrent connections available to the machine. If you expect a large number of users, you should edit the **/etc/ssh/sshd_config** file on each SAS Cloud Analytics Server (CAS) machine (controller and any workers) and update the value for MaxStartups to 100.

Permit the Creation of Temporary Files

All CAS processes must have Read, Write, and Execute permissions to the **/tmp** directory on each CAS machine. Therefore, you must grant these permissions to the server's ID and also to the IDs of any CAS users who will create sessions under their own identity. Creating sessions under the user’s identity is typical for connections that are made through programming interfaces such as SAS and Python.

Set the ulimits

The Linux operating system provides controls that enable you to limit the maximum number of open file descriptors and the maximum number of processes that a user ID can use. The sas account, cas account, and any other account that will be used to run a CAS session require nofiles at 20480 or above and nproc at 65536 or above.

Perform the following steps as the root user ID to ensure that the limits are high enough for each machine in your deployment to function correctly. For distributed CAS server installations, you can edit the files on one machine and copy the files to the other machines.

1 If all the accounts running CAS sessions are members of the sas group, then we recommend using a group definition to define the limits.

  To set the maximum number of open file descriptors for each machine in your deployment, open the **/etc/security/limits.conf** file on each machine. Add the following line or verify that it already exists:

  ```
  *     -     nofile     20480
  ```

  If you are setting nofiles for an account, replace the asterisk (*) with the user ID for that account. Repeat the line for each user.

  ```
  cas     -     nofile     20480
  ```

  If you are setting nofiles for a group, replace the asterisk with the @ symbol followed by the group name.

  ```
  @sas     -     nofile     20480
  ```

  You can also set nofiles for all users, regardless of group, by leaving the asterisk in place.

2 If all the accounts running CAS sessions are members of the sas group, then we recommend using a group definition to define the ulimits.

  For each machine in your deployment, edit the appropriate ***-nproc.conf** file and change the value for nproc from the default value of 1024 to 65536.

```
*      -     nproc      65536
```

If you are setting nproc for an account, replace the asterisk (*) with the user ID for that account. Repeat the line for each user.

```
cas      -     nproc      65536
```

If you are setting nproc for a group, replace the asterisk with the @ symbol followed by the group name.

```
@sas      -     nproc      65536
```

You can also set nproc for all users, regardless of group, by leaving the asterisk in place.

Note: In the filename *-nproc.conf, * is a wildcard that refers to a unique prefix to the nproc.conf filename that varies according to the version of Linux that is used. For Red Hat Enterprise Linux 6.7 or an equivalent distribution, the file location is **`/etc/security/limits.d/90-nproc.conf`**. For Red Hat Enterprise Linux 7.1 or an equivalent distribution, the file is **`/etc/security/limits.d/20-nproc.conf`**.

Installing SAS Viya with Ansible

This chapter describes the first installation of your SAS Viya software with the Ansible playbook included in your Software Order Email. For information about installing your software on a single machine with the yum utility, see Appendix B: Deploying with Yum on page 86.

Modify the Initial Deployment

This chapter describes the initial deployment of your SAS Viya software only. For information about modifying an existing deployment with updated software or adding new software to an existing deployment, see SAS Viya 3.2 Administration: Software Updates.

Use a Mirror Repository

By default, SAS downloads and installs the latest software available from the software repositories. If your deployment does not have access to the Internet, you should create a mirror repository. Mirror repositories also allow you to store and use the same version of software for a number of reasons:

- meeting regulatory restrictions
- creating a dev/test/prod environment
- adding Cloud Analytic Services (CAS) workers after initial deployment
- moving from symmetric multi-processing (SMP) to massively parallel processing (MPP) mode after initial deployment

For details about creating and using mirror repositories, see Appendix C: Creating and Using Mirror Repositories on page 94.

Edit the Inventory File

Ansible uses an inventory file to define the machines to be included in a deployment and the software to be installed on them. For multi-machine deployments, the **`sas_viya_playbook/hosts`** 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/hosts`**. The **`sas_viya_playbook/host_local`** file is used for a single-machine deployment.

Note about Sharing the hosts and the host_local File

The hosts and host_local files are generated for a specific software order. Do not copy these files from one playbook and attempt to use them in another playbook.

Edit the host_local File

If you are performing a multi-machine deployment, skip this section and go to Define the Machines in the Deployment on page 36.

The first line of the host_local file is a reference to a deployment target 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 use the host_local file without modification.

If you are using Ansible remotely, you should modify the first line in the host_local file to include the location of the machine where SAS Viya is being deployed. Here is an example:

```
deployTarget ansible_ssh_host=host1.example.com
```

Save the host_local file.

Define the Machines in the Deployment

If you are performing a single-machine deployment, skip this section and go to Assign the Target Machines to Host Groups on page 36.

The first section in the hosts 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 hosts file.

```
deployTarget ansible_ssh_host=<machine address> ansible_ssh_user=<userid> ansible_ssh_private_key_file=
<keyfile>
```

The following table describes the components of the deployment target reference:

Component of the Deployment Target Reference	Description
deployTarget	specifies the alias that is used by Ansible to refer to the physical machine definition. The default alias is **deployTarget**. 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 **ansible-controller**.
ansible_ssh_host	specifies any resolvable address for the target host, such as the IP address or fully qualified domain name.
ansible_ssh_user	specifies the user ID that is used by Ansible to connect to each of the remote machines and to run the deployment.
ansible_ssh_private_key_file	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 **~/.ssh** directory.

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

```
deployTarget ansible_connection=local
```

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

```
sas-stateful ansible_ssh_host=host1.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
sas-programming ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
sas-visual ansible_ssh_host=host3.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
controller-1 ansible_ssh_host=host4.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
worker-1 ansible_ssh_host=host5.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
worker-2 ansible_ssh_host=host6.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
worker-3 ansible_ssh_host=host7.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
```

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. Here is a typical assignment that uses the machines from the preceding example. Single-machine deployments using host_local as their inventory file should have to make few, if any, changes to the inventory file.

Add more than one host to a host group to achieve high availability (HA) for the software represented by the host group. Any caveats to this policy are described in the comments in the inventory file. If you plan to use high availability, you must plan for it in your initial deployment. You cannot change your deployment to add high availability without uninstalling your SAS Viya software and re-installing.

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

```
[AdminServices]
sas-visual

[CASServices]
sas-visual

[CoreServices]
sas-visual

[DataServices]
sas-visual

[HomeServices]
sas-visual

[ReportServices]
sas-visual

[ReportViewerServices]
sas-visual

[ThemeServices]
sas-visual

[programming]
sas-programming

[configuratn]
sas-visual

[consul]
sas-stateful

[httpproxy]
sas-stateful

[pgpoolc]
sas-stateful

[rabbitmq]
sas-stateful

[sasdatasvrc]
sas-stateful

[sas-casserver-primary]
controller-1

[sas-casserver-worker]
```

```
worker-1
worker-2
worker-3

[sas-all:children]
AdminServices
CASServices
CoreServices
DataServices
HomeServices
ReportServices
ReportViewerServices
ThemeServices
programming
configuratn
consul
httpproxy
pgpoolc
rabbitmq
sasdatasvrc
sas-casserver-primary
sas-casserver-worker
```

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.
- 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:

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

- SAS Event Stream Processing for CAS is automatically installed on all machines where CAS components are installed. These components enable two additional CAS action sets. Using these action sets is optional and requires a separate license. SAS Event Stream Processing 3.2 or later must be installed as a separate component with a separate license because it is required as a data source.
- If you purchased SAS Event Stream Manager (optional), the machine where you intend to install it must be specified in both the [viprESM] and [consul] host groups. In addition, do not install SAS Event Stream Manager and SAS Event Stream Processing on the same machine.

After you have completed your edits, save and close the hosts file.

Note: By default, your deployment includes a single-machine, single-node instance of HA PostgreSQL, used as the SAS Infrastructure Data Server. To deploy HA PostgreSQL with multiple nodes, refer to Appendix A: Creating High Availability PostgreSQL Clusters with Multiple Nodes on page 76.

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.

Set the Deployment Label

The DEPLOYMENT_LABEL 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, numeric characters, and hyphens. Nonalphanumeric characters, including a space, are not allowed. Here is an example of a valid name:

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

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. Use the following command to run the validation check without running the entire playbook: `ansible-playbook -i` *`inventory-file-name`* `system-assessment.yml`.

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 with a message.

Note: For information about supported Ansible versions, see Ansible Controller Requirements on page 27.

Verify Machine Properties

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

1. With a text editor, open the **`sas_viya_playbook/vars.yml`** file. If you used the recommended location at which to uncompress your playbook, the file is located at **`/sas/install/sas_viya_playbook/vars.yml`**.

2. Verify that the DEPLOYMENT_LABEL variable has content and contains only alphanumeric characters or the hyphen character.

   Note: For more information about the DEPLOYMENT_LABEL variable, see Set the Deployment Label on page 39.

3. Verify that a CAS controller host is defined.

   Note: For information about assigning software to machines, see Define the Machines in the Deployment on page 36.

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

5. 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 Define the Machines in the Deployment on page 36.

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

7. If the sas_consul_on_cas_hosts variable is set to **`false`**, verify that consul and localconsul are not placed on CAS primary nodes or worker nodes.

Note: For more information about sas_consul_on_cas_hosts, see Install Consul on CAS Hosts on page 43.

8. 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 given and the playbook stops.

Verify System Requirements

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

1. Verify that each machine's SELinux mode is either disabled or enabled but is set to *permissive*.

   Note: For more information about setting the SELinux mode, see Configure SELinux on page 29.

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 Define the Machines in the Deployment on page 36.

4. For each machine, verify that the install user has the following ulimits:

   - nofile is set to 20480 or higher.
   - nproc is set to 65536 or higher.

   Note: For more information about setting ulimits, see Set the ulimits on page 33.

Define Multiple Invocations

The INVOCATION_VARIABLES block is used to set the parameters of a High Availability (HA) PostgreSQL cluster of more than one machine. For details, see Appendix A: Creating High Availability PostgreSQL Clusters with Multiple Nodes on page 76.

Specify JRE (Optional)

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 sas_install_java to **`false`**. For example:

   ```
   sas_install_java: false
   ```

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

   ```
   sasenv_java_home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.101-3.b13.el6_8.x86_64/jre
   ```

4. Save and close the vars.yml file.

For a list of supported versions of Java, see Java Requirements on page 19.

Set Up Passwordless SSH for CAS

Manage Passwordless SSH

If 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 Ansible 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 41 for more information about the default process.
- Use your own passwordless SSH. See Use Your Own Passwordless SSH on page 41.
- 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 42.

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:

```
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 43.

To prevent the deployment process from setting up passwordless SSH, perform the following steps.

1. Open the vars.yml file.
2. Set the setup_sas_users field to **`false`**. Here is an example:

   ```
   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, perform the following steps:

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

Note: Comments have been removed from the following example.

```
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 setup_sas_packages 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.

Install Consul on CAS Hosts

SAS Viya uses HashiCorp Consul to discover other machines in the deployment. The Consul agent is normally deployed on all machines in a deployment, but it can be omitted from a machine that hosts only a CAS server. Omit the Consul agent only if you intend to share the CAS server machine across multiple SAS Viya deployments. Set the sas_consul_on_cas_hosts variable to **`false`** to disable deployment of the Consul agent on CAS server machines. If a value is not specified, **`true`** is used, by default.

- To deploy Consul on the CAS machines, set the sas_consul_on_cas_hosts variable to **`true`**. The default for sas_consul_on_cas_hosts is **`true`**.
- If you set the sas_consul_on_cas_hosts variable to **`false`**, and you assign the same machine to the [programming] host group and either the [sas-casserver-primary] host group or the [sas-casserver-worker] host group, the requirements check fails.

Note: For more information about modifying the inventory file, see Define the Machines in the Deployment on page 36.

Define the CAS User Group

Ensure that the user group for your CAS user account is correct.

1 Open the vars.yml file.

2 In the casenv_group field, insert the user group name.

3 Save and close the vars.yml file.

Set Up the CAS Admin User

To designate an LDAP user to be the CAS Admin user, perform the following steps:

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 valid LDAP user that exists and that can log on:

```
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.

Note: This user must have a single set of credentials that are valid for all applicable authentication providers. In a full deployment, dual authentication occurs for logon to CAS Server Monitor and access to CAS from SAS Studio. In a visual-only deployment, dual authentication occurs for logon to CAS Server Monitor. For more information, see Security in SAS Viya.

Change the CAS Instance Name

Changing this variable is not supported in SAS Viya 3.2.

Change the Tenant Name

By default, the ID for the tenant being deployed is **shared**. This ID is used in the CAS name for SAS Environment Manager (if you deploy all of your software or the visual interface only). To change the name, assign a different value to the casenv_tenant variable:

1 Open the vars.yml file.

2 Uncomment the line that contains the casenv_tenant variable. To uncomment, remove the number sign (#).

3 In the casenv_tenant field, insert the ID for the tenant:

```
casenv_tenant: tenant-ID
```

4 Save and close the vars.yml file.

Add Data Source Information

Overview of the Data Sources

If your order includes one or more data connectors, you must edit the vars.yml file to include information that is needed to install and configure the specific data connector. If you intend to use HDFS, you must also edit the vars.yml file.

The vars.yml file contains an example of a typical CAS_SETTINGS block that is commented out with number signs (#). The following sections contain examples of CAS_SETTINGS blocks that are appropriate for the specific connector. To customize the file, either uncomment the lines and edit the existing CAS_SETTINGS block or create a new CAS_SETTINGS block 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.

After you save the file, the Ansible script is run in order to update the cas.settings file.

SAS Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop

Follow these steps to edit the vars.yml file.

1 Open the vars.yml file.

2 Uncomment the CAS_SETTINGS line. To uncomment, remove the number sign (#).

3 Under CAS_SETTINGS:, add the following lines, including the spaces and numerical prefixes:

```
   1: JAVA_HOME=location-of-your-Java-8-JRE
   2: 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 playbook to specify a different location for the installation of the JRE.

4 If you are using MapR, add the following line:

```
3: MAPR_HOME=location-of-MapR-file
```

5 Save and close the vars.yml file.

SAS Data Connector to Impala

Follow these steps to edit the vars.yml file.

1. Open the vars.yml file.

2. Uncomment the `CAS_SETTINGS` line. To uncomment, remove the number sign (#).

3. Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes. 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 example includes both files:

   ```
   1: ODBCINI=location-of-your-odbc.ini-file
   2: ODBCINST=location-of-your-odbinstc.ini-file
   3: CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
   4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
   /opt/cloudera/impalaodbc/lib/64
   ```

4. Save and close the vars.yml file.

SAS Data Connector to ODBC

Follow these steps to edit the vars.yml file.

1. Open the vars.yml file.

2. Uncomment the `CAS_SETTINGS` line. To uncomment, remove the number sign (#).

3. Under CAS_SETTINGS, add the following lines (including the spaces and numerical prefixes), depending on the version of ODBC that you are using.

   For DataDirect:

   ```
   1: ODBCHOME=ODBC-home-directory
   2: ODBCINST=location-of-your-odbc.ini-file-including-file-name
   3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```

   For iODBC:

   ```
   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=$LD_LIBRARY_PATH:$ODBCHOME/lib
   ```

   For unixODBC:

   ```
   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, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

4. Save and close the vars.yml file.

SAS Data Connector to Oracle

Follow these steps to edit the vars.yml file.

1. Open the vars.yml file.

2. Uncomment the `CAS_SETTINGS` line. To uncomment, remove the number sign (#).

3 Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

```
1: ORACLE_HOME=ORACLE-home-directory
2: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

4 Save and close the vars.yml file.

SAS Data Connector to PostgreSQL

Follow these steps to edit the vars.yml file.

1 Before modifying the vars.yml file for PostgreSQL, you must create the odbcinst.ini file if it does not already exist.

  a At the top level of the sas_viya_playbook directory, run the following command to create and open a new file.

  ```
  sudo vi odbcinst.ini
  ```

  b Add the following lines to the odbcinst.ini file:

  ```
  [PostgreSQL]
  Description=ODBC for PostgreSQL
  Driver=/opt/sas/viya/home/lib64/psqlodbcw.so
  ```

  c Save and close the odbcinst.ini file.

2 Open the vars.yml file.

3 Uncomment the `CAS_SETTINGS` line. To uncomment, remove the number sign (#).

4 Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

```
1: ODBCINSTINI=location-of-your-odbcinst.ini-file
2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-PostgreSQL-ODBC-driver:
/opt/sas/viya/home/lib64/lib
```

5 Save and close the vars.yml file.

SAS Data Connector to Teradata

Follow these steps to edit the vars.yml file.

1 Locate the clispb.dat file, which is your Teradata client configuration file.

2 Ensure that the following two lines are in the clispb.dat file.

```
charset_type=N
charset_id=UTF8
```

3 Open the vars.yml file.

4 Uncomment the `CAS_SETTINGS` line. To uncomment, remove the number sign (#).

5 Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

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.

```
1: COPERR=location-of-Teradata-install/lib
2: COPLIB=directory-that-contains-clispb.dat
3: NLSPATH=Teradata-TTU-installation-path-including-msg-directory:$NLSPATH
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:Teradata-TTU-installation-path-including-lib64-directory:
$LD_LIBRARY_PATH
```

An example of the TTU Default LD_LIBRARY_PATH is

```
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/teradata/client/15.10/lib64:/opt/teradata/client/15.10
/tbuild/lib64
```

6 Save and close the vars.yml file.

Specifying Multiple Data Connectors

Note: When adding multiple data connectors, 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 data connectors can be mixed in the block, ensure that the lines for each data connector remain in the order provided in the preceding sections.

Because the LD_LIBRARY_PATH variable is included for each data connector, if you have more than one data connector, use as many lines as you have data connectors.

1 Open the vars.yml file.

2 Uncomment the `CAS_SETTINGS` line. To uncomment, remove the number sign (#).

3 Under CAS_SETTINGS, add the appropriate lines. Here is an example of a block for both the DataDirect version of SAS Data Connector to ODBC and for SAS Data Connector to Oracle:

```
1: ODBCHOME=/dbi/odbc/dd7.1.4
2: ODBCINI=/r/ge.unx.sas.com/vol/vol310/u31/fedadmin/ODBC/odbc_714_MASTER.ini
3: ORACLE_HOME=/dbi/oracle/12c
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
5: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Note: In the fourth line, each variable is separated with a colon.

4 Save and close the vars.yml file.

Set 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.

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. For more information about tuning, see “Memory” in SAS Cloud Analytic Services: Fundamentals.

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:

```
CAS_CONFIGURATION:
   env:
     CAS_DISK_CACHE: /var/tmp:/var/tmp2:/var/tmp3
```

It is recommended 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.

4 Save and close the vars.yml file.

Set Up HDFS and Co-location

Note: If your software order does not include SAS Access to Hadoop, SAS Data Connector to Hadoop, or the SAS Data Connect Accelerator for Hadoop, you should skip this section.

Default settings for the CAS_CONFIGURATION section of the vars.yml file appear as follows:

```
CAS_CONFIGURATION:
   env:
     #CAS_DISK_CACHE: /tmp
   cfg:
     #gcport: 5580
     #httpport: 8777
     #port: 5570
     #colocation: 'none'
```

Note: For descriptions of HDFS and co-location, see Hadoop Scenario 2: CAS SASHDAT Access to HDFS on page 11.

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:

```
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: 5580
     #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_NAME` 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: 5580
     #httpport: 8777
     #port: 5570
     colocation: 'hdfs'
     mode: 'mpp'
```

By default, CAS_REMOTE_HADOOP_PATH is set to **`$HADOOP_HOME/bin`**. You should supply a value for CAS_REMOTE_HADOOP_PATH only if you are using a location for the HDAT plug-ins in a different location than the default.

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_NAME` 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 *SAS Viya Administration: SAS Cloud Analytic Services*.

Establish TLS for the CAS Servers

Default settings for the CAS_CONFIGURATION section of the vars.yml file appear as follows:

```
CAS_CONFIGURATION:
   env:
     #CAS_DISK_CACHE: /tmp
   cfg:
     #gcport: 5580
     #httpport: 8777
     #port: 5570
     #colocation: 'none'
```

In order for the CAS machines in your deployment to have TLS security, modify the CAS_CONFIGURATION section as follows:

1 Open the vars.yml file.

2 Add the following highlighted variables and their respective values:

```
CAS_CONFIGURATION:
   env:
     #CAS_DISK_CACHE: /tmp
     CAS_CLIENT_SSL_REQUIRED: 'true'
     CAS_CALISTLOC: path-to-CA-chain-used-for-Apache-HTTP-Server-certificate
   cfg:
     #gcport: 5580
     #httpport: 8777
     #port: 5570
```

```
          #colocation: 'none'
          servicesbaseurl: 'https://http-proxy-host-name'
```

3 If you want to change the certificates that CAS Client TLS is using from certificates that are generated by SAS to site-signed or third-party certificates, add the following lines to the vars.yml file, in the same CAS_CONFIGURATION block:

```
CAS_CONFIGURATION:
   env:
     #CAS_DISK_CACHE: /tmp
     CAS_CLIENT_SSL_CERT:
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tls/certs/customer-server-certificate
   CAS_CLIENT_SSL_KEY:
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/private/custom-encrypted-server-key
```

4 Save and close the vars.yml file.

For more information about the TLS variables, see "CAS TLS Environment Variables" in Encryption in SAS Viya: Data in Motion.

Revise sitedefault.yml (Optional)

The sitedefault.yml file (in the **\roles\consul\files** directory in the playbook) is an advanced feature, used most often for provisioning a number of systems to be deployed in the same manner. It is not recommended that you use sitedefault.yml for the first deployment of your SAS Viya software, but instead for subsequent deployments.

For more information about filling out the sitedefault.yml file, see "Automate Configuration Properties During Deployment (Ansible)" in SAS Viya Administration: Configuration Properties.

Deploy the Software

Command Line

You deploy the software by running the playbook. Here is the basic syntax for the command to run the playbook:

```
command [ option ]
```

The command that you select is determined by your deployment and password requirements. See Commands on page 50.

You can select an option to specify the interface to the software to be installed in your environment. You can also specify the level of installation or configuration to perform. See Options on page 51.

Commands

Ensure that you are at the top level of the playbook in the **sas_viya_playbook** directory.

Use the appropriate command to run the playbook, according to the password requirements for the user ID that performs the deployment:

Note: The commands should be run as a root or sudoer user. Do not run these commands as a sas or cas user.

For a multi-machine deployment:

Does not require passwords	**ansible-playbook -i hosts site.yml**
Requires a sudo password only	**ansible-playbook -i hosts site.yml --ask-become-pass**

Requires an SSH password only	**ansible-playbook -i hosts site.yml --ask-pass**
Requires both a sudo and an SSH password	**ansible-playbook -i hosts site.yml --ask-pass --ask-become-pass**

All software (including Ansible) is on a single machine:

Does not require a sudo password	**ansible-playbook -i host_local site.yml**
Requires a sudo password	**ansible-playbook -i host_local site.yml --ask-become-pass**

The Ansible controller is separate from the single machine on which the software is to be deployed:

Does not require passwords	**ansible-playbook -i host_local site.yml**
Requires a sudo password only	**ansible-playbook -i host_local site.yml --ask-become-pass**
Requires an SSH password only	**ansible-playbook -i host_local site.yml --ask-pass**
Requires both a sudo and an SSH password	**ansible-playbook -i host_local site.yml --ask-pass --ask-become-pass**

Options

The following options can be specified in the command line:

-e "sas_install_type=visual"
deploys only the visual interface, including SAS Environment Manager and CAS.

-e "sas_install_type=programming"
deploys only the programming interface, including CAS, SAS Foundation, and SAS Studio.

Note: If you are installing SAS Event Stream Manager, do not use the programming-only option. SAS Event Stream Manager must be deployed with a full deployment (no command lines) or with the visual-only option.

--tags install
only installs the software, but does not configure or start it.

--tags config
configures and starts the software that was installed using the install-only option described above.

For example, if you wanted to deploy only the programming interface for multiple machines that do not require extra passwords, the entire command would be

```
ansible-playbook -i hosts site.yml -e "sas_install_type=programming"
```

If you wanted to install the software on only a single machine that does not include Ansible but also requires SSH passwords, the entire command would be

```
ansible-playbook -i host_local site.yml --ask-pass --tags install
```

Run from a Directory Other than the Default

The Ansible 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 SAS Viya 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 *********************************************************************
deployTarget               : ok=81    changed=65    unreachable=0    failed=0
```

The most important indicator of success from this message is failed=0.

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 Commands on page 50 and any appropriate options.

Apply the SAS Event Stream Manager License

Note: If your order does not contain SAS Event Stream Manager, skip this section.

1 In the directory where you uncompressed the playbook, copy the license file to a new file named license.txt.

```
sudo cp license-file-name.txt license.txt
```

For *license-file-name*, substitute the filename of the license file that you saved to your computer from the SOE.

2 Copy the license file to the license directory on the machine where SAS Event Stream Manager is deployed, the machine that you assigned to the [viprESM] host group.

If SAS Event Stream Manager and the playbook are located on the same machine, go to the directory where the playbook was uncompressed, and run the following command:

```
sudo cp license.txt /opt/sas/viya/config/etc/sysconfig/sas-esm-service/default/license.txt
```

3 Restart the SAS Event Stream Manager service by running the following command:

```
sudo service sas-viya-esm-service-default restart
```

Log On to Your SAS Viya Software

If you deployed your SAS Viya software with both interfaces or the programming interface only, perform the following steps to log on:

1 Open SAS Studio from a URL with this format:

```
http://webserver-host-name/SASStudio
```

Use the host name from the machine that you assigned to the [httpproxy] host group in the inventory file. For more information assigning machines, see Assign the Target Machines to Host Groups on page 36.

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 Administrators on page 74.

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.

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

  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 specified in the SPAWNER_CONFIGURATION block.

The port numbers listed in those blocks are the defaults. For example

```
SPAWNER_CONFIGURATION:
  #sasPort: 8591
```

To change the value:

1 Remove the number sign from the beginning of the variable for the port number that you want to change.

2 Change the port value to the one that you want to use.

3 Save and close the vars.yml file.

Here is the earlier example revised in this way:

```
SPAWNER_CONFIGURATION:
  sasPort: 8592
```

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
```

Manual Configuration Tasks

Configure Machine

Configure High Availability in SAS Studio

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

Note: To use high availability in SAS Studio, a shared file system is required. For details, see (Optional) High-Availability Requirements on page 17.

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.

2 For each programming host, do the following:

   a Stop SAS Studio.

   - For RHEL 6.7:

   ```
   sudo service sas-viya-sasstudio-default stop
   ```

   - For RHEL 7.x:

   ```
   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-xxx). 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>
   ```

3 Start SAS Studio.

   - For RHEL 6.7:

   ```
   sudo service sas-viya-sasstudio-default start
   ```

   - For RHEL 7.x:

   ```
   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 **`/etc/httpd/conf.d/proxy.conf`** file.

   b Locate and remove (or comment out) any existing lines that contain `ProxyPass` or `ProxyPassReverse`:

```
# ProxyPass /SASStudio http://SAS-Studio-host:7080
# ProxyPassReverse  /SASStudio http:// SAS-Studio-host:7080
```

   c Add the following lines to map 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
  BalancerMember http://SAS-Studio-host-2:SAS-Studio-port route=sasstudio-SAS-Studio-host-2-IP
  ProxySet scolonpathdelim=on stickysession=JSESSIONID
</Proxy>
ProxyPass /SASStudio balancer://SASStudio-cluster
ProxyPassReverse /SASStudio balancer://SASStudio-cluster
```

   Here is an example:

```
<Proxy balancer://SASStudio-cluster>
  BalancerMember http://hosta.company.com:7080 route=sasstudio-100-10-0-1
  BalancerMember http://hosta.company.com:7080 route=sasstudio-100-10-0-2
  ProxySet scolonpathdelim=on stickysession=JSESSIONID
</Proxy>
ProxyPass /SASStudio balancer://SASStudio-cluster
ProxyPassReverse /SASStudio balancer://SASStudio-cluster
```

   d Add the following lines to map remote servers:

```
ProxyPass /SASStudio/privatehost-instance-1 http://SAS-Studio-host-1-instance:SAS-Studio-port
ProxyPassReverse /SASStudio/privatehost-instance-1 http://SAS-Studio-host-1-instance:SAS-Studio-port
ProxyPass /SASStudio/privatehost-instance-2 http://SAS-Studio-host-2-instance:SAS-Studio-port
ProxyPassReverse /SASStudio/privatehost-instance-2 http://SAS-Studio-host-2-instance:SAS-Studio-port
```

   Here is an example:

```
ProxyPass /SASStudio/privatehosta http://hosta.company.com:7080
ProxyPassReverse /SASStudio/privatehosta http://hosta.company.com:7080
ProxyPass /SASStudio/privatehostb http://hostb.company.com:7080
ProxyPassReverse /SASStudio/privatehostb http://hostb.company.com:7080
```

   e 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 RHEL 6.7:

```
sudo service httpd restart
```

- For RHEL 7.x:

```
sudo systemctl restart httpd
```

7 On each programming machine, open SAS Studio from a URL with this format:

```
http://webserver-host-name/SASStudio
```

Configure Security

Set the Password for the CAS Administrator or Another Administrative Account

SAS recommends using an LDAP user as the CAS administrator. However, you can enable the cas user account to be the CAS administrator by adding a password to the cas user account on the CAS controller and all CAS worker nodes. To assign a password, use the following command:

```
sudo passwd cas
```

You must also create an LDAP account with an identical password for this user.

To enable any other user account as a CAS administrator, you must add a password to that account on the CAS controller and all CAS worker nodes.

Note: To access CAS Server Monitor, you must set the password for the CAS Administrator or another administrative account.

Change the Administrative User Password for SAS Message Broker

Note: The tasks in this section are applicable if you deployed all of your software or only the visual interface. 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.

   --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, see “All Servers and Services” in SAS Viya Administration: General Servers and Services.

Configure Your Environment with SAS Environment Manager

The tasks in this section are applicable if you deployed all of your software or only the visual interface. 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. Follow these steps:

1 Locate the 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 35.

2 Search the log for the characters, **`sasboot`**, by using the following command:

   ```
   grep 'sasboot' sas-saslogon_date-and-time-stamp.log
   ```

   A message similar to the following is displayed

   ```
   Reset password for initial user sasboot using link: /SASLogon/reset_password?
   code=xxxxxx
   ```

3 Sign in from a URL with this format:

   ```
   http://http-proxy-host-name/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 35.

   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 Administrators on page 74.

4 Follow the instructions on the displayed web page to reset the password.

   Note: If the URL has expired, go to **`/etc/init.d`** and run the following command:

   ```
   sudo ./sas-viya-saslogon-default restart
   ```

   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. Opt in to all of the assumable groups so that you have the permissions to perform subsequent tasks.

Configure the Connection to Your Identity Provider

After installing a new SAS Viya deployment, you must configure the connection to your identity provider before your users can access SAS Environment Manager and SAS Visual Analytics. Complete these steps while you are signed in as the sasboot user.

Note: Only LDAP-based identity providers are supported. These instructions assume that you have basic familiarity with LDAP administration. For details about properties, see "sas.identities.providers.ldap" in SAS Viya Administration: Configuration Properties

1 If the Configuration page of SAS Environment Manager is not already displayed, select **Resources** ⇨ **Configuration** from the side menu ☰.

2 Select **Basic Services** from the list, and then select the **Identities service** from the list of services.

3 In the **sas.identities.providers.ldap.user** section, click ✱. In the New Configuration window, do the following:

   a Specify a value for the following required field: **baseDN**. 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 user-level field in SAS, specify a corresponding property in the LDAP provider software.

   > **TIP** In this step, consider specifying a custom filter to limit the group accounts that SAS Viya returns from your provider.

   b Click **Save**.

4 In the **sas.identities.providers.ldap.group** section, click ✱. In the New Configuration window, do the following:

   a Specify a value for the following required field: **baseDN**. 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 provider software.

   > **TIP** In this step, consider specifying a custom filter to limit the group accounts that SAS Viya returns from your provider.

   b Click **Save**.

5 In the **sas.identities.providers.ldap.connection** section, click ✱. In the New Configuration window, do the following:

   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 Click **Save**.

6 From the SAS Environment Manager side menu, select **Previous** ⇨ **Users**.

   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.

   Verify that user and group information is displayed correctly. If not, make any necessary changes to the identities service properties.

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:

```
http://http-proxy-host-name/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.

6 Restart the SASLogon service. For more information, see General Servers and Services: Operate in *SAS Viya Administration: General Servers and Services*.

Sign in Using LDAP Credentials

Open SAS Environment Manager from a URL with the following format:

```
http://http-proxy-host-name/SASEnvironmentManager
```

Sign in as one of the SAS Administrators that you set up in Set Up Administrative Users on page 58.

Reset the sasboot Password and Disable the Password Reset Feature

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 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 **Resources** ⇨ **Configuration** from the side menu ≡.

2 On the Configuration page, select **Definitions** from the drop-down list.

3 In the left pane, select **sas.logon.initial**. Then select ✱ 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**.

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.

Restrict Folder Creation at the Root Level

SAS administrators are typically the only users who create folders at the root level. To restrict all other users from creating content at the root level, see “Restrict Creation of Top-Level Folders” in SAS Viya Administration: General Authorization.

Configure SAS Viya to Connect to LDAPS Provider

After the deployment is complete, be aware that your system is not yet secured. To configure LDAPS, see “Configure SAS Viya to Connect to LDAPS Provider” in Encryption in SAS Viya: Data in Motion.

Configure Data Access

Configure SAS Data Connector to Hadoop

Note: The information in this section is applicable only if you ordered SAS Data Connector to Hadoop.

During installation, you should have configured the location of the shared libraries and the library path in the vars.yml file. To ensure that any redeployment contains these configuration settings, you must also make these changes in the vars.yml file. For information, see SAS Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop on page 44.

To manually configure the variables:

1 Locate the cas.settings file in the /opt/sas/viya/home/SASFoundation directory on the CAS controller. 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.

2 If you are using MapR, add the following line:

```
export MAPR_HOME=/opt/mapr
```

3 Save and close the cas.settings file.

Configure SAS Data Connector to Impala

Note: This information is applicable only if you ordered SAS Data Connector to Impala.

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/download.html.

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.

Note: During installation, you should have configured the pathname of the shared library and the CLOUDERAIMPALAODBC environment variable in the vars.yml file. If you did not set up the pathname and the CLOUDERAIMPALAODBC environment variable in the vars.yml file, you must configure the LD_LIBRARY_PATH variable and the CLOUDERAIMPALAODBC environment variable manually using the instructions in this step. To ensure that any redeployment contains the configuration settings, you must also make these changes in the vars.yml file. For information, see SAS Data Connector to Impala on page 45.

a Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

b Under CAS_SETTINGS, add the following lines. 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 example includes both files:

```
export ODBCINI=location-of-your-odbc.ini-file
export ODBCINST=location-of-your-odbinstc.ini-file
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
```

c Save and close the cas.settings file.

3 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.

Configure SAS Data Connector to ODBC

Note: This information is applicable only if you ordered SAS Data Connector to ODBC.

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 During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see SAS Data Connector to ODBC on page 45.

Using a text editor, open the cas.settings file.

```
sudo vi /opt/sas/viya/home/SASFoundation/cas.settings
```

3 Add the following lines that are appropriate for the version of ODBC that you are using.

- DataDirect:

  ```
  export ODBCHOME=ODBC-home-directory
  export ODBCINI="location-of-your-odbc.ini-file-including-file-name"
  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
  ```

- 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=$LD_LIBRARY_PATH:$ODBCHOME/lib
  ```

- 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 you have not set up ODBCSYSINI in your environment, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

4 Save and close the cas.settings file.

Configure SAS Data Connector to Oracle

Note: The information in this section is applicable only if you ordered SAS Data Connector to Oracle.

During installation, you should have configured the location of the shared libraries and the library path by editing the vars.yml file. For information about those steps, see SAS Data Connector to Teradata on page 46.

You must also configure the location of the shared libraries and the library path in the cas.settings file.

To manually configure these variables:

1 Locate the cas.settings file in the /opt/sas/viya/home/SASFoundation directory on the CAS controller. Add the following lines:

```
export ORACLE_HOME=ORACLE-home-directory
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

2 Save and close the cas.settings file.

Configure SAS Data Connector to PostgreSQL

Note: This information is applicable only if you ordered SAS Data Connector to PostgreSQL.

By default, the ODBC driver for PostgreSQL is shipped with SAS Data Connector to PostgreSQL. When SAS Data Connector to PostgreSQL code is submitted for execution, this ODBC driver is automatically referenced. As a result, it is unnecessary to set environment variables that point to this ODBC driver.

During installation, you should have configured the location of the shared libraries in the vars.yml file. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see SAS Data Connector to PostgreSQL on page 46.

1 Create and configure the odbcinst.ini file. Here is an example:

```
[PostgreSQL]
Description=ODBC for PostgreSQL
Driver=/opt/sas/viya/home/lib64/psqlodbcw.so
```

2 Locate the cas.settings file in the **`/opt/sas/viya/home/SASFoundation`** directory on the CAS controller.

3 Add the following lines:

```
export ODBCINSTINI="location-of-your-odbcinst.ini-file"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-PostgreSQL-ODBC-driver:
/opt/sas/viya/home/lib64/lib
```

4 Save and close the cas.settings file.

Configure SAS Data Connector to Teradata

Note: The information in this section is applicable only if you ordered SAS Data Connector to Teradata.

During installation, you should have configured the location of the shared libraries and the library path by editing the vars.yml file. For information about those steps, see SAS Data Connector to Teradata on page 46.

You must also configure the location of the shared libraries and the library path in the cas.settings file.

To manually configure these variables:

1 Locate the clispb.dat file, which is your Teradata client configuration file.

2 Ensure that the following two lines are in the clispb.dat file.

```
charset_type=N
charset_id=UTF8
```

3 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

4 Add the following lines:

```
export COPERR=location-of-Teradata-installation/lib
export COPLIB=directory-that-contains-clispb.dat
export NLSPATH=Teradata-TTU-installation-path-including-msg-directory:$NLSPATH
export LD_LIBRARY_PATH=Teradata-TTU-installation-path-including-lib64-directory:$LD_LIBRARY_PATH
```

Here is an example of the TTU default LD_LIBRARY_PATH:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/teradata/client/15.10/lib64:/opt/teradata/client/15.10/tbuild/lib64
```

5 Save and close the cas.settings file.

Validating the Deployment

Perform Installation Qualification on RPM Packages

Some of your SAS software is collected in RPM (Red Hat Package Manager) packages. This section describes how to qualify the installation of your RPM packages.

Here is the basic command to verify RPM packages:

```
rpm -Vv <package name>
```

For example, to verify the contents of the `sas-envesml` package, use the following command:

```
rpm -Vv sas-envesml
```

You can also 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:

```
for i in $(rpm -qa | grep -e "^sas-");do rpm -Vv $i;done
```

A successful verification shows the list of files that make up the RPM but no error indicators, as follows:

```
# rpm -Vv sas-envesml
......... /opt/sas/viya/home/lib/envesml/sas-init-functions
#
```

An unsuccessful verification provides error indicators beside the filename. Here is an example:

```
# rpm -Vv sas-envesml
S.5....T. /opt/sas/viya/home/lib/envesml/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/envesml/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.

- 5 - MD5 checksum

  The MD5 checksum of a file is a 128-bit number that is mathematically derived from the contents of the file. The MD5 checksum conveys no information about the contents of the original file, but, any change to the file results in a change to the MD5 checksum. RPM creates MD5 checksums for all files that it manipulates, and stores the checksums in its database. If one of these files is changed, the MD5 checksum changes and the change is detected by RPM.

- 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.

Access CAS Server Monitor

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:

```
http://controller-machine:8777
```

Here is an example:

```
http://my_controller.com:8777
```

Note: During the initial deployment, CAS Server Monitor is set up for HTTP and HTTPS and can also be accessed using the following URLs:

```
http://http-proxy-host-name/cas-shared-default-http

https://http-proxy-host-name/cas-shared-default-http
```

Log on using one of the SAS Administrator users that you established in Set Up Administrative Users on page 58 .

- In a full deployment, dual authentication occurs for logon to CAS Server Monitor and access to CAS from SAS Studio.
- In a visual-only deployment, dual authentication occurs for logon to CAS Server Monitor.

For more information, see “Security in SAS Viya” in *SAS Viya 3.2: Administration*

If you access CAS Server Monitor from within SAS Studio, the link to CAS Server Monitor uses the https protocol by default. (CAS Server Monitor uses the https protocol even when you log in to SAS Studio with the http protocol). The `Connection Not Secure` message is displayed until you do one of the following:

- Import the appropriate Certificate Authority (CA) certificate into the browser.
- Change the protocol from https to http for the CAS Monitor Server that you access from within SAS Studio.

For information about securing CAS Server Monitor, see the *SAS Viya 3.2 Administration: Encryption Guide*.

Note: To access CAS Server Monitor, the password must be set for the cas user ID or other administrative account. To set the password, see Set the Password for the CAS Administrator or Another Administrative Account on page 56.

Access SAS Environment Manager

Note: This section is applicable only if you deployed the visual interface. If you deployed the programming interface only, 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:

   ```
   http://http-proxy-host-name/SASEnvironmentManager
   ```

3. Sign on as one of the SAS Administrators that you set up in Set Up Administrative Users on page 58.

(Optional) Access SAS Event Stream Manager

If you purchased SAS Event Stream Manager, verify that it has been successfully deployed. Perform these steps to access it:

1. Open a web browser instance and enter the URL in the address field in the following format:

   ```
   http://webserver-host:port/SASEventStreamManager
   ```

   Note: Google Chrome 47 or later is required.

   Use the host name of the machine that you assigned to the [httpproxy] host group in the inventory file. Its default port is 80.

   The **Sign In to SAS** window is displayed.

2. When prompted, enter your user ID and password, and click **Sign In**.

   Successful logon to the user interface indicates that the software was installed correctly.

SAS Event Stream Manager uses SAS Logon Manager for logon functionality. Therefore, you can manage user access in SAS Environment Manager and in LDAP.

Verify SAS Message Broker

Note: This section is applicable only if you have a full deployment or a visual-only deployment. If you have a programming-only deployment, skip this section.

1. To verify that the SAS Message Broker has been deployed correctly, go to the machine you assigned to the [rabbitmq] host group.

2. Open a browser and go to the following address:

   ```
   http://RabbitMQ-IP-address: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 or a visual-only 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. During deployment, SAS Infrastructure Data Server performs a verification check and creates a log of that process. The log is located on the machine that you assigned to the [pgpoolc] host group at

   ```
   /opt/sas/viya/config/var/log/sasdatasvrc/service-name/node-name/sds_status_check_date-time-stamp.log
   ```

   For example, for a single-machine deployment, the location would be

   ```
   /opt/sas/viya/config/var/log/sasdatasvrc/postgres/pgpool0/sds_status_check_date-time-stamp.log
   ```

   Go to this log.

2. For a successful deployment, the log will contain tables indicating attempts at reaching the nodes, like this:

   ```
   node_id |    hostname      | port | status | lb_weight |  role
   ---------+---------------+------+--------+-----------+---------
   0       | wipds1.sas.com | 5432 | 2      | 1.000000  | primary
   (1 row)
   ```

   A log for a failed verification contains error messages or other output that is not the direct output of database queries.

3. On the same machine, run the following command:

   ```
   sudo service sas-viya-sasdatasvrc-postgres status
   ```

4. If SAS Infrastructure Data Server is running appropriately, you should receive a response like this:

   ```
    node_id |      hostname       | port | status | lb_weight |  role
   ---------+-------------------+------+--------+-----------+---------
    0       | wipds1.sas.com    | 9452 | 2      | 0.250000  | primary
    1       | wipds2.sas.com    | 9462 | 2      | 0.250000  | standby
    2       | wipds3.sas.com    | 9472 | 2      | 0.250000  | standby
    3       | wipds4.sas.com    | 9482 | 2      | 0.250000  | standby
   ```

   A status of 2 for a node indicates the node is running.

Verify SAS Data Connector to Impala

Verification Methods

Note: The information in this section is applicable only if you ordered SAS Data Connector to Impala.

Depending on your deployment type, you can verify SAS Data Connector to Impala in either of these ways:

- For a full deployment or a programming-only deployment, use SAS Studio. For details, see Use SAS Studio to Verify SAS Data Connector on page 69.
- For a visual-only deployment, use SAS Environment Manager. For details, see Use SAS Environment Manager to Verify the SAS Data Connector to Your Data Source on page 70.

Use SAS Studio to Verify the SAS Data Connector to Impala

To verify that SAS Data Connector to Impala was successfully deployed:

1 Sign on to SAS Studio:

a Open SAS Studio from a URL with the following format: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code:

```
caslib implib datasource=(srctype="impala", username="user-ID",
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 you do not see table names that you recognize, you should perform the configuration steps again. For details, see Configure SAS Data Connector to Impala on page 60.

Verify SAS Data Connector to ODBC

Note: The information in this section is applicable only if you ordered SAS Data Connector to ODBC.

To verify that SAS Data Connector to ODBC was successfully deployed:

1 Sign on to SAS Studio:

a Open SAS Studio from a URL with the following format: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code:

```
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 you do not see table names that you recognize, you should perform the configuration steps again. For details, see Configure SAS Data Connector to ODBC on page 61.

Verify SAS Data Connector to Oracle

Verification Methods

Note: The information in this section is applicable only if you ordered SAS Data Connector to Oracle.

Depending on your deployment type, you can verify SAS Data Connector to Oracle in either of these ways:

- For a full deployment or a programming-only deployment, use SAS Studio. For details, see Use SAS Studio to Verify SAS Data Connector on page 69.
- For a visual-only deployment, use SAS Environment Manager. For details, see Use SAS Environment Manager to Verify the SAS Data Connector to Your Data Source on page 70.

Use SAS Studio to Verify SAS Data Connector

To verify that SAS Data Connector to Oracle was successfully deployed:

1 Sign on to SAS Studio:

  a Open SAS Studio from a URL with the following format: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code:

```
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 you do not see table names that you recognize, you should perform the configuration steps again. For details, see Configure SAS Data Connector to Oracle on page 62.

Verify SAS Data Connector to PostgreSQL

Verification Methods

Note: The information in this section is applicable only if you ordered SAS Data Connector to PostgreSQL.

Depending on your deployment type, you can verify SAS Data Connector to Oracle in either of these ways:

- For a full deployment or a programming-only deployment, use SAS Studio. For details, see Use SAS Studio to Verify SAS Data Connector on page 69.
- For a visual-only deployment, use SAS Environment Manager. For details, see Use SAS Environment Manager to Verify the SAS Data Connector to Your Data Source on page 70.

Use SAS Studio to Verify SAS Data Connector to PostgreSQL

Note: The information in this section is applicable only if you ordered SAS Data Connector to PostgreSQL.

To verify that SAS Data Connector to PostgreSQL was successfully deployed:

1 Sign on to SAS Studio:

   a Open SAS Studio from a URL with the following format: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code:

```
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 you do not see table names that you recognize, you should perform the configuration steps again. For details, see Configure SAS Data Connector to PostgreSQL on page 62.

Use SAS Environment Manager to Verify the SAS Data Connector to Your Data Source

Note: Your data source must already be set up and must already contain tables.

CAUTION! You cannot validate ODBC with SAS Environment Manager.

1 From the side menu ☰, under SAS Environment Manager, select **Data**.

2 At the top of the Data page, from the **View** drop-down list, select **Libraries**.

3 Right-click the library to which you want to import the files, and then click **Import**.

4 From the Import Data window, expand the **Server** menu on the left side and select the database name from the list. The list includes only the data sources that you are authorized to use and for which you have a license.

5 Select your data source.

6 In the Connection Information window, specify the connection information.

Note: For Hadoop only, the option that you choose in the **Data transfer mode** field might not be saved. The next time you make a connection, check the field to make sure that the correct setting is displayed.

Here are some key points about specifying connection information:

- The **Server** field corresponds to the host name for the server. Some databases connect using a data source name instead of the combination of a server name and a port number.
- Most fields are case sensitive. For example, specifying a value of **products** in the **Database** field might not be the same as specifying **PRODUCTS**. Case sensitivity depends on the database vendor. Furthermore, some databases use schemas. Some databases automatically use the user ID as the schema if a schema is not explicitly specified. Be aware that the **User ID** and **Schema** fields can be case sensitive. Consult with your database administrator if you are unsure.
- For Hadoop and Teradata servers only:
  - The **Data transfer mode** field displays only the options that you are licensed to use.
  - Modify the **Character multiplier option** to increase the number of characters that can fit in each cell so that character data truncation does not occur. The lengths for character variables are increased by multiplying the current length by the value that you specify. You can specify a multiplier value from 1 to 5. The default value is **2**.
- If you close the Connection Information window without entering your connection information, or if you want to change your saved connection information, you can reopen the window by clicking the connection button in the Import Data window.

For more information about connection parameters, refer to *SAS Cloud Analytic Services: Language Reference*.

7 Click **Connect**.

Note: If you connect successfully to the server, but no tables are available, the Import window is empty. To verify the connection by importing a table, your data source must contain tables. Set up tables in your data source and then perform the verification steps again.

8 Select a table to import by selecting the check box next to the input table name.

9 Click **OK**.

> **TIP** If the **OK** button is unavailable, make sure that you have selected the check box next to the table that you want to import. If you changed the name of the output table in the **Table Name** column, click **Enter** or click outside of the box to validate the name before clicking **OK**.

10 If the table is imported, you have successfully validated your connection to your data source. If the table fails to import, an icon appears in the **Status** column. By clicking the icon, you can view additional information, such as log or error messages.

Uninstalling SAS Viya

This section describes how to uninstall SAS Viya software if it was deployed using Ansible. For information about uninstalling yum deployments, see Uninstall SAS Viya with Yum on page 93.

What deploy-cleanup.yml Does

When you use the deploy-cleanup command described in the following sections, it performs these actions:

1. Stops all SAS services.
2. Removes all SAS RPMs.
3. Deletes any remaining SAS .pid files.
4. Deletes the entitlement_certificate.pem and SAS_CA_Certificate.pem files.

After the deploy-cleanup command is run, the only item that remains is the snapshot directory. If you deployed your software using Ansible, the deployment saved valuable deployment information for later use in the sas_deployment.tgz file. This file and the playbook are saved to the same location: the **/snapshot/*epoch*** subdirectory, where **epoch** specifies the UNIX epoch (the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time). 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

In addition, it also renames the **/opt/sas/viya** directory to **/opt/sas/viya_*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.

Uninstall from a Single Machine

To uninstall your SAS Viya software from a single-machine deployment, run the following command:

```
ansible-playbook -i host_local deploy-cleanup.yml
```

If the environment requires one or more passwords, the command must include additional parameters:

Password Requirements	Additional Parameters
Password for sudo only	`--ask-become-pass`
Password for SSH only (applies only if the Ansible controller is on a different machine than your SAS software)	`--ask-pass`
Password for both sudo and SSH (applies only if the Ansible controller is on a different machine than your SAS software)	`--ask-become-pass --ask-pass`

Uninstall from Multiple Machines

To uninstall your SAS Viya software from a deployment with more than one machine, run the following command:

```
ansible-playbook -i hosts deploy-cleanup.yml
```

If the environment requires one or more passwords, the command must include additional parameters:

Password Requirements	Additional Parameters
Password for sudo only	`--ask-become-pass`
Password for SSH only	`--ask-pass`
Password for both sudo and SSH	`--ask-become-pass --ask-pass`

If your software deployment includes SAS Embedded Process, uninstall it using the instructions at Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya on page 127.

If your software deployment includes CAS SASHDAT Access to HDFS, uninstall it using the instructions at Uninstall the SASHDAT Plug-ins on page 140.

Next Steps

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

We recommend that you copy the sas_deployment.tgz file and save it to a separate location, possibly on a 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 Administrators

If other persons are responsible for administering your SAS deployment, we recommend that you share the following important information with them:

- Deployment type: did you deploy only the programming interface or the visual interface, or did you perform a full deployment?

  If you used the customized script from your playbook (described in Appendix B: Deploying with Yum on page 86), you deployed the programming interface only.

  If you used Ansible, the type was determined by the addition of any arguments to the deployment command. See Options on page 51 for information about the arguments and which type they deploy.

- URL to access the software, based on deployment type:
  - If you performed a full deployment or if you deployed only the visual interface, your administrators should use SAS Environment Manager. You should have accessed SAS Environment Manager while working through Configure Your Environment with SAS Environment Manager on page 57. Use the same URL that you used in that section.
  - If you deployed only the programming interface, your administrators should use SAS Studio. You should have accessed SAS Studio while working through Log On to Your SAS Viya Software on page 52. Use the same URL that you used in that section.

Refer to Additional Documentation

After you validate the deployment, you can perform initial administrative tasks. For details, refer to SAS Viya Administration: Orientation.

For usage information, refer to the Help that is available from the SAS Viya product and administrative interfaces.

You can also do the following:

- Refer to the appendixes in this guide for information about additional tasks that you might perform, based on your environment.

- Install SAS Event Stream Processing on a separate machine.

  If you purchased SAS Event Stream Processing for CAS, a freestanding installation of SAS Event Stream Processing is required in order to provide data for the supported CAS actions.

- Read the SAS Event Stream Processing user documentation. A separate deployment guide is provided to help you install the software. Links to all SAS Event Stream Processing documentation are available on the SAS Event Stream Processing product page.

- If you purchased SAS Event Stream Manager, be sure to install the SAS Event Stream Manager agent during the installation of SAS Event Stream Processing. You can access the user documentation for SAS Event Stream Manager from the SAS Event Stream Manager product page.

Appendix A: Creating High Availability PostgreSQL Clusters with Multiple Nodes

SAS Viya uses High Availability (HA) PostgreSQL as the SAS Infrastructure Data Server. By default, when you use the instructions in Installing SAS Viya with Ansible on page 35, 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.

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. If 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 multi-node deployments, PGPool node can be colocated with data nodes or deployed on its own machine. Note that colocating 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 multi-node topologies.

Cluster Configuration	Server	Port	Role
Horizontal	Server 1	5432	Primary
	Server 2	5432	Standby
	Server 3	5432	Standby
	Server 4	5432	Standby
Vertical	Server 1	5532	Primary

Cluster Configuration	Server	Port	Role
	Server 1	5533	Standby
	Server 1	5534	Standby
	Server 1	5535	Standby
Hybrid	Server 1	5632	Primary
	Server 1	5633	Standby
	Server 2	5632	Standby
	Server 2	5633	Standby

The two files in your playbook that must be revised for HA PostgreSQL are the hosts and vars.yml files. The hosts 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 hosts and vars.yml, those edits should be done in tandem to ensure alignment.

Note: You must configure the vars.yml file for your desired cluster configuration before you run the playbook. You will not be able to add nodes to an existing cluster after it has been deployed.

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

sasdatasvrc

- NODE_NUMBER: the 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

Set Up a Horizontal Cluster

Edit the hosts File

Modify the hosts file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at Define the Machines in the Deployment on page 36. Then assign the machines to the host groups as described at Assign the Target Machines to Host Groups on page 36. 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 hosts file that includes the horizontal cluster described in the preceding table. The PGPool is on the same machine as the first HA PostgreSQL node. (The example shows only the entries related to HA PostgreSQL):

```
deployTarget1 ansible_ssh_host=host.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget3 ansible_ssh_host=host3.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget4 ansible_ssh_host=host4.example.comx ansible_ssh_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 preceding table, 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 hosts 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 50 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Vertical Cluster

Edit the hosts File

Modify the hosts file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at Define the Machines in the Deployment on page 36. Then assign the machines to the host groups as described at Assign the Target Machines to Host Groups on page 36. 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 hosts 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_ssh_host=host.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deloyTarget1
'''
[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:

```
# 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 50 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Hybrid Cluster

Edit the hosts File

Modify the hosts file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at Define the Machines in the Deployment on page 36. Then assign the machines to the host groups as described at Assign the Target Machines to Host Groups on page 36. 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 hosts 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):

```
deployTarget1 ansible_ssh_host=host.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[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 vertical 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'
     SERVICE_NAME: postgres
```

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 50 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up Multiple Clusters

Modify hosts 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 hosts file should be modified for this HA PostgreSQL deployment (the entries related to HA PostgreSQL are shown):

```
deployTarget1 ansible_ssh_host=host.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget3 ansible_ssh_host=host3.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget4 ansible_ssh_host=host4.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
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'
      PGPOOL_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
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5462'
      SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
  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
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5452'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_horizontal
  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 PGPools 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. In the location where you uncompressed your playbook, find the sitedefault_sample.yml file. If you used the suggestions in this guide, that location is **`/sas/install/sas_viya_playbook/sitedefault_sample.yml`**.

2. Copy and paste the sitedefault_sample.yml file into the same directory, and rename the file as sitedefault.yml.

3. Open the sitedefault.yml file.

4. At the end of the existing file and at the same indention level as `application`, add the following content:

   ```
   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}
   ```

5. 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/sas/sasdatasvrc/postgres/node0/
```

The log for the HA PostgreSQL nodes is located at

```
/opt/sas/viya/config/var/log/sas/sasdatasvrc/postgres/pool0/
```

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 multi-node 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 B: Deploying with Yum

Use this appendix for instructions to deploy only the elements of the programming interface of your SAS Viya software on a single machine. Because SAS Event Stream Manager cannot be deployed with only the programming interface, if your order includes that product you must deploy using the steps described in Installing SAS Viya with Ansible on page 35.

Run the Deployment Script

1 If you left the certificates in the **sas_viya_playbook** directory, you can skip to the next step.

  If you moved the certificates, open the customized_deployment_script.sh file that was included in the playbook that you saved from the Software Order Email (SOE). Use a text editor to specify the directory path that contains the certificates. Here is an example:

```
CERTDIR=/opt/sas/installfiles
```

2 Save and close the customized_deployment_script.sh file.

3 If you are installing SAS Viya on a machine that is already running SAS 9.4 software, determine whether required ports are available by running the following commands:

- SAS Object Spawner:

```
netstat -an |grep 8591
```

- SAS/CONNECT:

```
netstat -an |grep 17551
```

  If a command does not produce any output, then the port is available for use and no changes are required. If the command produces output, then the port is already being used by a product and is blocked for usage by other products. Make a note of any blocked product for additional steps to be taken after the deployment has been performed.

4 Run the script:

```
sudo ./customized_deployment_script.sh
```

5 If you have any blocked products from step 3, modify the required file described for the product or products as described in this list:

- SAS Object Spawner:

  Open the**/opt/sas/viya/config/etc/spawner/default/spawner.cfg** file. Change the sasPort value to an available port number.

  Also open the**/opt/sas/viya/config/etc/sasstudio/default/init_usermods.properties** file. Change the webdms.workspaceServer.port value to the same port number used in the**/opt/sas/viya/config/etc/spawner/default/spawner.cfg** file.

- SAS/CONNECT:

  Open the **/opt/sas/viya/config/etc/sysconfig/connect/default/sas-connect** file. Change the CONNECT_PORT value to an available port number.

Deploy the HTTP Daemon and the SSL Module

1 Run the command to deploy httpd and mod_ssl:

```
sudo yum install httpd mod_ssl
```

2. Create the proxy.conf file:

   ```
   sudo vi /etc/httpd/conf.d/proxy.conf
   ```

3. Copy and paste the following content into the proxy.conf file, and substitute the appropriate host names:

   ```
   RewriteEngine on
   RewriteRule ^/SASStudio$ /SASStudio/ [R]
   ProxyPass /SASStudio http://SAS-studio-host:7080
   ProxyPassReverse  /SASStudio http://SAS-studio-host:7080
   ProxyPass /cas-shared-default-http http://CAS-controller-host:8777/cas-shared-default-http
   ProxyPassReverse  /cas-shared-default-http http://CAS-controller-host:8777/cas-shared-default-http
   ```

4. Save and close the proxy.conf file.

5. Start the httpd service.

   - For RHEL 6.7:

     ```
     sudo service httpd start
     ```

   - For RHEL 7.x:

     ```
     sudo systemctl restart httpd
     ```

Apply the Licenses for SAS and CAS Software

1. Locate the license file that you previously saved.

2. Run the command to apply the license to your SAS software:

   ```
   sudo su -s "/bin/sh" -c "/opt/sas/viya/home/SASFoundation/utilities/bin/apply_license /opt/sas
   /installfiles/license-file-name" sas
   ```

   You receive a message that your license has been applied.

3. To apply the license to CAS, copy your license file into the CAS default configuration directory. Here is an example:

   ```
   sudo cp /opt/sas/installfiles/license-file-name /opt/sas/viya/config/etc/cas/default/
   ```

4. Open the config.lua file:

   ```
   sudo vi /opt/sas/viya/config/etc/cas/default/casconfig.lua
   ```

5. Locate the env.CAS_LICENSE variable in the casconfig.lua file, and specify the directory path that contains the license. Here is an example:

   ```
   env.CAS_LICENSE = config_loc .. '/SASViyaV0300_09J9P5_Linux_x86-64.txt'
   ```

6. Save and close the casconfig.lua file.

Register Your SAS Software

Perform the following command to register your SAS Viya software:

```
sudo su -s "/bin/sh" -c "/opt/sas/viya/home/SASFoundation/utilities/bin/post_install build_registry" sas
```

You receive a message that the build registry tasks have completed.

Set Up the CAS Administrator

Specify the user account for the CAS Admin user. You can use the cas account that was created during the deployment of CAS. Alternatively, you can specify another account.

1 Open the perms.xml file with the following command:

```
sudo vi /opt/sas/viya/config/etc/cas/default/perms.xml
```

2 Replace each instance of the ${ADMIN_USER} variable with the name of a user that exists and that can log on. Here is an example of two such instances:

```
<Administrator name="${ADMIN_USER}-User-SuperUser" user="${ADMIN_USER}" type="SuperUser"/>
```

Here is an example of the replaced values:

```
<Administrator name="casadmin-User-SuperUser" user="casadmin" type="SuperUser"/>
```

3 Save and close the perms.xml file

4 If you want to use the cas user account to be the CAS Admin user, you must add a password to the cas user account. In order to assign a password, use the following command:

```
sudo passwd cas
```

Set Up the CAS Controller to Run as a Service

In order to ensure that the CAS controller runs as a service, perform these steps:

1 Copy and rename the sas-controller.init file with the following command:

```
sudo cp /opt/sas/viya/home/SASFoundation/utilities/bin/sas-cascontroller.init
/etc/rc.d/init.d/sas-viya-cascontroller-default
```

Note: In the example, for improved readability, the single command occupies two lines.

2 Change ownership of the new file with the following command:

```
sudo chown sas:sas /etc/rc.d/init.d/sas-viya-cascontroller-default
```

3 Add the new service with the following command:

```
sudo /sbin/chkconfig --add /etc/rc.d/init.d/sas-viya-cascontroller-default
```

Start the Services

Start the CAS controller, a SAS object spawner, and SAS Studio.

Note: The following examples include a command to start the SAS/CONNECT spawner, which is applicable only if SAS/CONNECT was included in your software order.

```
sudo service sas-viya-cascontroller-default start
sudo service sas-viya-spawner-default start
sudo service sas-viya-sasstudio-default start
sudo service sas-viya-connect-default start
```

Configure SAS Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop

The information in this section is applicable only if you ordered SAS Data Connector to Hadoop or SAS Data Connect Accelerator for Hadoop.

Follow these steps to configure CAS access to the data source:

1. Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

2. 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`** line. If you are using the JRE that is installed with your SAS software, its default location is **`/usr/lib/jvm/jre-1.8.0`**.

3. If you are using MapR, add the following line:

   ```
   export MAPR_HOME=/opt/mapr
   ```

4. Save and close the cas.settings file.

Configure SAS Data Connector to Impala

The information in this section is applicable only if you ordered SAS Data Connector to Impala.

Follow these steps to configure SAS Data Connector to Impala:

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 SAS Technical Support website support.sas.com.

2. Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

3. Under CAS_SETTINGS, add the following lines. Depending on how your Impala ODBC driver is configured, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files, as appropriate. The following example includes both files

   ```
   export ODBCINI=location-of-your-odbc.ini-file
   export ODBCINST=location-of-your-odbinstc.ini-file
   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
   ```

4. Save and close the cas.settings file.

5. 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.

Configure SAS Data Connector to ODBC

The information in this section is applicable only if you ordered SAS Data Connector to ODBC.

Follow these steps to configure SAS Data Connector to ODBC:

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. Using a text editor, open the cas.settings file.

   ```
   sudo vi /opt/sas/viya/home/SASFoundation/cas.settings
   ```

3. Add the following lines that are appropriate for the version of ODBC that you are using.

   - DataDirect:

     ```
     export ODBCHOME=ODBC-home-directory
     export ODBCINI="location-of-your-odbc.ini-file-including-file-name"
     export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
     ```

   - 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=$LD_LIBRARY_PATH:$ODBCHOME/lib
     ```

   - 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 you have not set up ODBCSYSINI in your environment, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

4. Save and close the cas.settings file.

Configure SAS Data Connector to Oracle

The information in this section is applicable only if you ordered SAS Data Connector to Oracle.

Follow these steps to configure SAS Data Connector to Oracle:

1. Ensure that the variable for the shared library path points to the location of the Oracle shared libraries. The name of this variable is operating system-dependent. This variable setting is required because the SAS Data Connector to Oracle executable must know the location of the Oracle shared libraries in order to use them.

2 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller. Add the following lines:

```
export ORACLE_HOME=ORACLE-home-directory
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

3 Save and close the cas.settings file.

Configure SAS Data Connector to PostgreSQL

The information in this section is applicable only if you ordered SAS Data Connector to PostgreSQL.

Follow these steps to configure SAS Data Connector to PostgreSQL:

By default, the ODBC driver for PostgreSQL is shipped with SAS Data Connectorto PostgreSQL. When SAS Data Connector to PostgreSQL code is submitted for execution, this ODBC driver is automatically referenced. As a result, it is unnecessary to set environment variables that point to this ODBC driver.

1 Create and configure the odbcinst.ini file. Here is an example:

```
[PostgreSQL]
Description=ODBC for PostgreSQL
Driver=/opt/sas/viya/home/lib64/psqlodbcw.so
```

2 Locate the cas.settings file in the **`/opt/sas/viya/home/SASFoundation`** directory on the CAS controller.

3 Add the following lines:

```
export ODBCINSTINI="location-of-your-odbcinst.ini-file"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-PostgreSQL-ODBC-driver:
/opt/sas/viya/home/lib64/lib
```

4 Save and close the cas.settings file.

Configure SAS Data Connector to Teradata

The information in this section is applicable only if you ordered SAS Data Connector toTeradata.

Follow these steps to configure SAS Data Connector toTeradata:

1 Locate the clispb.dat file, which is your Teradata client configuration file.

2 Ensure that the following two lines are in the clispb.dat file.

```
charset_type=N
charset_id=UTF8
```

3 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

4 Add the following lines:

```
export COPERR=location-of-Teradata-installation/lib
export COPLIB=directory-that-contains-clispb.dat
export NLSPATH=Teradata-TTU-installation-path-including-msg-directory:$NLSPATH
export LD_LIBRARY_PATH=Teradata-TTU-installation-path-including-lib64-directory:$LD_LIBRARY_PATH
```

Here is an example of the TTU default LD_LIBRARY_PATH:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/teradata/client/15.10/lib64:/opt/teradata/client/15.10/tbuild/lib64
```

5 Save and close the cas.settings file.

Configure Settings for SAS Event Stream Processing for CAS

The information in this section is applicable only if you ordered SAS Event Stream Processing for CAS.

Follow these steps to configure CAS settings for SAS Event Stream Processing:

1. Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.
2. Use a text editor to edit the file **`/opt/sas/viya/home/SASFoundation/cas.settings`**.
3. Add the following lines to the cas.settings file:

   ```
   export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DFESP_HOME/lib
   export TKPATH=$TKPATH:$DFESP_HOME/lib/tk
   ```
4. Save and close the cas.settings file.

Install Sample SAS Data Sets

The programming documentation includes examples of how the SAS software works. To follow the examples on your own deployment, you need sample SAS data sets. Experienced users might not need the sample data sets since they probably already have data that can be used.

To install the sample SAS data sets, run the following command on the machine on which SAS Viya is installed:

```
sudo yum install sas-samplesml
```

The SAS data sets are installed at **`/opt/sas/viya/home/SASFoundation/sashelp`** and require no configuration. The programming documentation describes how to use the examples.

Log On to SAS Studio

Perform the following steps to log on:

1. Open SAS Studio from a URL with this format:

   ```
   http://webserver-host-name/SASStudio
   ```
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.

View Deployment Logs

To view the logs of your yum deployment, run the following commands:

```
sudo yum history
sudo less /var/log/yum.log
```

Validate the Installation

After you complete the procedures in this appendix, you should validate the installation. For details, see Validating the Deployment on page 64.

Uninstall SAS Viya with Yum

Perform the following steps to uninstall your SAS Viya software with yum:

1 Stop all the services with the following command:

```
sudo service sas-viya-all-services stop
```

2 Remove the cascontroller service with the following commands:

```
sudo /sbin/chkconfig --del /etc/rc.d/init.d/sas-viya-cascontroller-default
sudo rm /etc/rc.d/init.d/sas-viya-cascontroller-default
```

3 Remove the products by following these steps:

a Open the customized_deployment_script.sh file that was included in the playbook, which you saved from the Software Order Email (SOE).

b To obtain the list of products to remove, locate the yum groupinstall command in the shell script file. Here is an example:

```
# Install the software
yum groupinstall "SAS Machine Learning" "SAS CAS for Machine Learning" "SAS CAS for Statistics"
"SAS Statistics" "SAS Foundation" "SAS CAS for Visual Analytics"
```

c Remove the products by using them in the following command. Here is an example:

```
sudo yum groupremove "SAS Machine Learning" "SAS CAS for Machine Learning" "SAS CAS for Statistics"
"SAS Statistics" "SAS Foundation" "SAS CAS for Visual Analytics"
```

4 Remove the repositories by following these steps:

a To obtain the names of the repositories to remove, locate the yum install command in the shell script file. Here is an example:

```
# Install definitions of the specific repositories for the ordered products
yum install "sas-va-8.1.0-rpm-latest" "sas-mchnlrng-8.1.1-rpm-latest" "sas-statviya-8.1.0-rpm-latest"
```

b Remove the repositories by using them in the following command. Here is an example:

```
sudo yum erase "sas-va-8.1.0-rpm-latest" "sas-mchnlrng-8.1.1-rpm-latest" "sas-statviya-8.1.0-rpm-latest"
```

5 Remove the main repository definition with the following command:

```
sudo yum erase sas-meta-repo-1-1
```

6 Remove any remaining components with the following command:

```
sudo rpm -e $(rpm -qg SAS)
```

7 Remove the entitlement certificate with the following command:

```
sudo rm /etc/pki/sas/private/entitlement_certificate.pem
```

8 Rename the **`viya`** directory with the following command:

```
sudo mv /opt/sas/viya/ /opt/sas/viya_$(date +%s)
```

This command assigns a suffix to the directory name that is equal to the UNIX epoch (the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time).

9 Close the customized_deployment_script.sh file.

Appendix C: Creating and Using Mirror Repositories

This appendix describes the steps to create a mirror repository. A mirror repository is a copy of the necessary content from SAS that is located at your own site. Mirror repositories are especially useful for sites that have limited access to the Internet.

Requirements

The instructions in this appendix assume a topology that consists of one or more machines that perform these roles: an Ansible controller, a mirror repository host connected to the Internet, a mirror repository host that is not connected to the Internet, and deployment targets. All machines described in this chapter must meet the operating system requirements described in Operating System Requirements on page 17. The following topics describe each type of machine and additional requirements.

Ansible Controller

The Ansible controller is the machine that runs the reposync.yml Ansible playbook. The SAS_Viya_playbook.tgz file from your Software Order Email (SOE) must be on this machine. In addition, the Ansible controller has the following requirements:

- does not require Internet access.
- requires network connectivity and Ansible accessibility to itself, as well as to the connected repository mirror, the unconnected repository mirror, and the deployment target machines. The Ansible controller should have a public key on the connected and unconnected mirror repositories and a private key on the connected mirror repository with the same deployment user account.
- must have Ansible installed.
- must be capable of controlling itself through Ansible.

Connected Repository Mirror

The connected repository mirror is the machine that uses the Internet to connect to the yum repositories that are hosted by SAS. The private key of the user that will run Ansible (on the Ansible controller machine) must be included in that user's home directory on the connected repository mirror. This requirement is fulfilled by default when the connected repository mirror machine is also the Ansible controller machine. In addition, the connected repository mirror has the following requirements:

- must have Internet access.
- must be capable of control by the Ansible controller.
- has 20 GB of free disk space in **`/tmp/mirror/location`** to hold a temporary archive of the repository mirror files.

Unconnected Repository Mirror

The unconnected repository mirror is the machine that contains the yum repository. It serves files over HTTP, usually via Apache httpd. The reposync.yml playbook installs the httpd package on the unconnected repository mirror machine if the package has not already been installed. In addition, the unconnected repository mirror has the following requirements:

- is reachable from your deployment target machine or machines by HTTP.

- can be controlled by your Ansible controller machine.
- has 20 GB of free disk space in **`/var/www/html/pulp`** to hold the mirror repository files.

Deployment Targets

The deployment targets are the machines to which you deploy SAS Viya software. Software repositories are not deployed on the target machines. The deployment targets do not require access to the Internet. However, for RPM packages that do not originate from SAS, the site.yml Ansible playbook will try to download and install various RPM package files. When the playbook runs, it will default to respect local mirror yum repositories that have been set up by Linux system administrators. If local mirror yum repositories are not in place, then the deployment target machine will try to retrieve yum repositories over the Internet.

Machine Combinations

It is possible to combine roles within a single machine. The following table summarizes the compatibility of roles on a single machine.

Machine Role	Ansible Controller	Connected Repository Mirror	Unconnected Repository Mirror	Deployment Target
Ansible Controller	-	recommended	possible	possible
Connected Repository Mirror	recommended	-	not recommended	possible
Unconnected Repository Mirror	possible	not recommended	-	possible
Deployment Target	possible	possible	possible	-

For example, although it is possible for the roles of the connected repository mirror, the unconnected repository mirror, and a deployment target to occupy the same machine as the Ansible controller role, SAS recommends that only the Ansible controller and the connected repository mirror occupy the same machine.

Use Ansible to Create a Mirror Repository

Confirm that Ansible Is Installed on the Ansible Controller

1 Run the following command on the Ansible controller:

```
ansible --version
```

2 If the command results are similar to the following, then Ansible has been successfully installed on the machine.

```
ansible 2.2.1.0
  config file = /home/centos/sas_viya_playbook/ansible.cfg
  configured module search path = Default w/o overrides
```

3 If your results are different, Ansible has not been installed on the machine. To install Ansible on the machine, see Install Ansible on page 32.

Confirm the Identities of the Hosts

1 On each machine in your topology, run the following command:

```
hostname -I && hostname -f
```

2. The command results should be two lines. The first line is the IP address of the host and the second is its FQDN.

   ```
   10.149.16.32
   machine1.example.com
   ```

   Confirm that the command results include an appropriate IP address and an appropriate fully qualified domain name (FQDN) for each machine. If you see unexpected results, then do not proceed. Consider modifying the **`/etc/hosts`** file on each machine, or adjusting your Domain Name System (DNS) infrastructure as appropriate. Do not proceed until the hostname command produces a valid IP address and FQDN.

Prepare the repohosts Inventory File

1. On the Ansible controller machine, locate the repohosts file in the directory where you uncompressed the SAS_Viya_playbook.tgz file. If you followed the suggestions in this guide, that file is located at **`/sas/install/sas_viya_playbook/repohosts`**.

2. Open the repohosts file.

3. The beginning of the file contains the following lines:

   ```
   lighthost ansible_ssh_host=<machine_address>
   darkhost ansible_ssh_host=<machine_address>
   ```

   Replace `<machine_address>` in the first line with any resolvable address, such as the IP address or the fully qualified domain name, for the machine that is the connected mirror repository. Replace `<machine_address>` in the second line with any resolvable address for the machine that is the unconnected mirror repository. If you add ansible_ssh_user information, ensure that the same user is added to both lines.

   Note: Do not use 127.0.0.1 as an IP address for any machines in the file repohosts. Also do not add `ansible_connection=local` to the repohosts file.

4. Save and close the repohosts file.

Confirm Network Connectivity and Ansible Accessibility

1. On the Ansible controller machine, from the sas_viya_playbook directory, run the following command:

   ```
   ansible -i repohosts -m ping all
   ```

2. Confirm that the command results are similar to the following:

   ```
   darkhost | SUCCESS => {
       "changed": false,
       "ping": "pong"
   }
   lighthost | SUCCESS => {
       "changed": false,
       "ping": "pong"
   }
   ```

   If the results do not include the word `SUCCESS`, then do not proceed with these steps until you can confirm both network connectivity and Ansible accessibility.

Install and Enable Apache httpd (Optional)

The RPM package files in the mirror repository on the unconnected mirror repository machine are typically made available to other machines in your topology through a network connection. The HTTP application protocol is a typical form of network connectivity software. Network connectivity is typically achieved by running web server software (such as Apache httpd or Nginx nginx) on the unconnected mirror repository machine. The reposync.yml Ansible playbook can install and start Apache httpd on your unconnected mirror repository machine.

1. On the Ansible controller machine, locate the repo_vars.yml file in the sas_viya_playbook directory.

2. Run the following command to ensure that the file is writeable:

   ```
   chmod +w repo_vars.yml
   ```

3. Open repo_vars.yml.

4. Locate the following line:

   ```
   # setup_httpd_server: no
   ```

5. Uncomment the line, and replace `no` with `yes`.

   ```
   setup_httpd_server: yes
   ```

6. Save and close the repo_vars.yml file.

7. On the unconnected mirror repository machine, ensure that firewall software is not running. Use the commands in steps 3 and 4 of Firewall Considerations on page 31.

Create the Mirror Repository

1. On the Ansible controller machine, from the sas_viya_playbook directory, run the following command:

   ```
   ansible-playbook -i repohosts reposync.yml
   ```

   This command runs the reposync.yml playbook, which performs the following actions:

   - downloads SAS software RPM package files from entitled yum repositories that are hosted by SAS on the Internet
   - places the downloaded files in a temporary location on the connected mirror repository (**`/tmp/mirror/location`** by default)
   - creates a file named repo_override.txt in the current working directory on the Ansible controller
   - copies the files from the temporary location on the connected mirror repository to an Apache httpd accessible location on the unconnected mirror repository (**`/var/www/html/pulp/repos`** by default)
   - (Option) installs and starts Apache httpd software on the unconnected mirror repository

2. When the reposync.yml Ansible playbook has finished running, the command results should be similar to the following:

   ```
   PLAY RECAP **********************************************************************
   darkhost                   : ok=17   changed=7    unreachable=0    failed=0
   lighthost                  : ok=30   changed=14   unreachable=0    failed=0
   ```

   The most important indicator of success from the command results is `failed=0`.

Confirm HTTP Connectivity to the Mirror Repository

On each deployment target machine, run the following command to confirm that the deployment target machine can access the mirror repository on the unconnected mirror repository.

```
curl -s -o /dev/null -w "%{http_code}\n" http://IP-address-of-dark-host/pulp/repos/
```

If the command does not return the value 200, then do not proceed until you can confirm HTTP connectivity from the deployment targets to the unconnected mirror repository.

Deploy the SAS Viya Software to the Deployment Targets

Before deploying your SAS Viya software, you must complete the steps described in Edit the Inventory File on page 35 and Modify the vars.yml File on page 38. After those sections are completed, perform the following steps:

1 On the Ansible Controller machine, from the sas_viya_playbook directory, run the following command:

```
ansible -i hosts -m ping all
```

2 Confirm that the command results are similar to the following:

```
deployTarget | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

If the results do not include the word SUCCESS, then do not proceed until you can confirm both network connectivity and Ansible accessibility.

3 On the Ansible controller machine, from the sas_viya_playbook directory, run the following command:

```
ansible-playbook -i hosts site.yml -e "@repo_override.txt"
```

This command runs the site.yml playbook, which installs the SAS Viya software on the deployment targets.

4 When the site.yml Ansible playbook has finished running, the command results should be similar to the following:

```
PLAY RECAP **********************************************************************
deployTarget                   : ok=17   changed=7    unreachable=0    failed=0
deployTarget2                  : ok=30   changed=14   unreachable=0    failed=0
```

The most important indicator of success from these results is `failed=0`.

Use Yum to Create Mirror Repositories

Perform the following steps to create and use a mirror repository with yum. Note that these steps cannot be used to deploy SAS Visual Investigator.

Prepare the Connected Mirror Repository

1 Copy the SAS_Viya_playbook.tgz file from your Software Order Email (SOE) to the repository mirror host.

2 Extract the files from SAS_Viya_playbook.tgz:

```
tar xf SAS_Viya_playbook.tgz
```

Create and Use the createrepos.sh File

1 Go to the sas_viya_playbook directory on the connected mirror repository.

2 Using a text editor, create a new file named createrepos.sh that contains the following content:

```
#!/bin/bash

sudo yum install yum-utils

cp customized_deployment_script.sh setup_repos.sh

sed -i -e 's/^\s*yum groupinstall/#yum groupinstall/' setup_repos.sh

./setup_repos.sh

MIRRORLOC=/tmp/mirror/location
if [ ! -d ${MIRRORLOC} ]; then
  mkdir -p ${MIRRORLOC}
fi

for f in $(ls /etc/yum.repos.d/sas-*.repo | cut -f4 -d/ | sed s/.repo//g | grep -v sas-meta)
do
  reposync -n -d -m --repoid=${f} --download_path=${MIRRORLOC} --download-metadata
done

cd ${MIRRORLOC}
tar -zcvf repomirror.tar.gz sas-*
```

3 Save and close createrepos.sh.

4 Set the Execute bit for createrepos.sh:

```
sudo chmod +x createrepos.sh
```

5 Run createrepos.sh to extract the contents of the SAS repositories and to create a tar ball:

```
sudo ./createrepos.sh
```

Prepare the Unconnected Mirror Repository

1 Copy the tar file that was created from the repository synchronization and the **`sas_viya_playbook/soe.yml`** from the connected mirror repository to the unconnected mirror repository. Here is an example:

Note: The following command assumes that the sas_viya_playbook directory is the current directory n the connected mirror repository.

```
scp /tmp/mirror/location/repomirror.tar.gz soe.yml user@darkhost:/tmp
```

2 On the unconnected mirror repository, go to the **`tmp`** directory.

3 Use a text editor to create a new file named yumrepocreation.sh that contains the following content:

```
#!/bin/bash

sudo yum install yum-utils createrepo httpd

REPOLOC=/var/www/html/pulp/repos
ORDERABLE=$(grep METAREPO_SOE_ORDERABLE soe.yml | awk -F"'" '{ print $2 }')
# Make the directory that will house the yum repository
```

```
if [ ! -d ${REPOLOC} ]; then
  mkdir -p ${REPOLOC}
fi

echo ""
echo "Unpack the files from repomirror.tar.gz"
tar xf repomirror.tar.gz -C ${REPOLOC}

echo ""
echo "Create the repository"
for repo in ${ORDERABLE}; do
  NAME=$(sed -e 's/^"//' -e 's/"$//' <<<"$repo")
  createrepo -v --update ${REPOLOC}/${NAME}  -g ${REPOLOC}/${NAME}/comps.xml
done
```

4 Save and close the yumrepocreation.sh file.

5 Set the Execute bit for the yumrepocreation.sh file:

```
chmod +x yumrepocreation.sh
```

6 Run the yumrepocreation.sh file.

```
sudo ./yumrepocreation.sh
```

Create the repo.conf File

Create a new file named **`/etc/httpd/conf.d/repo.conf`** that contains the following content:

```
<Directory "/var/www/html/pulp/repos/">
  Options All
  AllowOverride All
  Require all granted
  Satisfy any
</Directory>
Alias "/pulp/repos" "/var/www/html/pulp/repos/"
```

Note: If you are using Red Hat Enterprise Linux 6.7 or an equivalent distribution, remove the line that contains `Require all granted`. However, later distributions require the line.

Restart the httpd Service

Restart or reload the httpd service as needed.

1 Check the status of the httpd service:

```
sudo service httpd status
```

2 If httpd is already running, reload it:

```
sudo service httpd reload
```

3 If httpd is not running, start it:

```
sudo service httpd start
```

Confirm HTTP Connectivity to the Mirror Repository

On each deployment target machine, run the following command to confirm that the deployment target machine can access the unconnected mirror repository:

```
curl -s -o /dev/null -w "%{http_code}\n" http://IP-address-of-unconnected-mirror-repository/pulp/repos/
```

If the command does not return the value 200, then do not proceed until you can confirm HTTP connectivity from the deployment targets to the unconnected mirror repository.

Note: You might need to change your firewall software configuration on the unconnected mirror repository machine in order for the curl command to succeed. Another option is to temporarily stop the firewall software on the unconnected mirror repository machine using the commands in steps 3 and 4 of Firewall Considerations on page 31.

Install from the Repository Using Ansible

Confirm that Ansible Is Installed on the Ansible Controller

1 Run the following command on the Ansible controller:

```
ansible --version
```

2 If the command results are similar to the following, then Ansible has been successfully installed on the machine.

```
ansible 2.2.1.0
  config file = /home/centos/sas_viya_playbook/ansible.cfg
  configured module search path = Default w/o overrides
```

3 If your results are different, Ansible has not been installed on the machine. To install Ansible on the machine, see Install Ansible on page 32.

Confirm the Identities of the Hosts

1 On each machine in your topology, run the following command:

```
hostname -I && hostname -f
```

2 The command results should be two lines. The first line is the IP address of the host and the second is its FQDN.

```
10.149.16.32
machine1.example.com
```

Confirm that the command results include an appropriate IP address and an appropriate FQDN for each machine. If you see unexpected results, then do not proceed. Consider modifying the **/etc/hosts** file on each machine, or adjusting your Domain Name System (DNS) infrastructure as appropriate. Do not proceed until the hostname command produces a valid IP address and FQDN.

Prepare the repohosts Inventory File

1 On the Ansible controller machine, locate the repohosts file in the directory where you uncompressed the SAS_Viya_playbook.tgz file. If you followed the suggestions in this guide, that file is located at **/sas/install/sas_viya_playbook/repohosts**.

2 Open the repohosts file.

3 The beginning of the file contains the following lines:

```
lighthost ansible_ssh_host=<machine_address>
darkhost ansible_ssh_host=<machine_address>
```

Replace `<machine_address>` in the first line with any resolvable address, such as the IP address or the FQDN, for the machine that is the connected mirror repository. Replace `<machine_address>` in the second line with any resolvable address for the machine that is the unconnected mirror repository. If you add ansible_ssh_user information, ensure that the same user is added to both lines.

Note: Do not use 127.0.0.1 as an IP address for any machines in the file repohosts. Also do not add `ansible_connection=local` to the repohosts file.

4. Save and close the repohosts file.

Confirm Network Connectivity and Ansible Accessibility

1. On the Ansible controller machine, from the sas_viya_playbook directory, run the following command:

   ```
   ansible -i repohosts -m ping all
   ```

2. Confirm that the command results are similar to the following:

   ```
   darkhost | SUCCESS => {
       "changed": false,
       "ping": "pong"
   }
   lighthost | SUCCESS => {
       "changed": false,
       "ping": "pong"
   }
   fy
   ```

   If the results do not include the word `SUCCESS`, then do not proceed with these steps until you can confirm both network connectivity and Ansible accessibility.

Install and Enable Apache httpd (Optional)

The RPM package files in the mirror repository on the unconnected mirror repository machine are typically made available to other machines in your topology through a network connection. The HTTP application protocol is a typical form of network connectivity software. Network connectivity is typically achieved by running web server software (such as Apache httpd or Nginx nginx) on the unconnected mirror repository machine. The reposync.yml Ansible playbook can install and start Apache httpd on your unconnected mirror repository machine.

1. On the Ansible controller machine, locate the repo_vars.yml file in the sas_viya_playbook directory.

2. Run the following command to ensure that the file is writeable:

   ```
   chmod +w repo_vars.yml
   ```

3. Open repo_vars.yml.

4. Locate the following line:

   ```
   # setup_httpd_server: no
   ```

5. Uncomment the line, and replace `no` with `yes`.

   ```
   setup_httpd_server: yes
   ```

6. Save and close the repo_vars.yml file.

7. On the unconnected mirror repository machine, ensure that firewall software is not running. Use the commands in steps 3 and 4 of Firewall Considerations on page 31.

Deploy the SAS Viya Software

Before deploying your SAS Viya software, you must complete the steps described in Edit the Inventory File on page 35 and Modify the vars.yml File on page 38. After those sections are completed, perform the following steps:

1 Run the repohosts playbook with a tag that creates an override file to be used in later steps.

```
ansible-playbook -i repohosts reposync.yml -t directory,override --skip-tags repocopy
```

2 Run the main deployment pass with the site.yml playbook.

```
ansible-playbook -i hosts site.yml -e "@repo_override.txt"
```

3 When the site.yml Ansible playbook concludes, the output should look similar to this:

```
PLAY RECAP *********************************************************************
deployTarget                   : ok=17   changed=7    unreachable=0    failed=0
deployTarget2                  : ok=30   changed=14   unreachable=0    failed=0
```

The most important indicator of success from this message is `failed=0`.

Install from the Repository Using Yum

Create the sas-manual.repo File

1 In the sas_viya_playbook directory on the connected mirror repository, create a new file named createrepodefn.sh that contains the following content:

```
#!/bin/bash

REPOURI="http://xxx.xxx.xxx.xxx"
ORDERABLE=$(grep METAREPO_SOE_ORDERABLE soe.yml | awk -F"'" '{ print $2 }')

for repo in ${ORDERABLE}; do
  NAME=$(sed -e 's/^"//' -e 's/"$//' <<<"$repo")
  cat << EOL >> sas-manual.repo
[${NAME}]
name=${NAME}
baseurl=${REPOURI}/pulp/repos/${NAME}/
enabled=1
sslverify=0
sslcacert=
sslclientcert=
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x

EOL
done
```

2 Change the value of REPOURI to the location of the unconnected mirror repository.

```
REPOURI="http://example.company.com"
```

3 Save and close the createrepodefn.sh script.

4 Set the Execute bit for the createrepodefn.sh script.

```
chmod +x createrepodefn.sh
```

5 Run the createrepodefn.sh script.

```
./createrepodefn.sh
```

6. The script creates a file named sas-manual.repo. Copy the sas-manual.repo file to each deployment target.

   ```
   scp sas-manual.repo user@deploytarget:/tmp
   ```

7. On each deployment target, as a user with root privileges, copy sas-manual.repo file to **`/etc/yum.repos.d/`**.

Install on the Target Machines

1. If you are installing SAS Viya on a machine that is already running SAS 9.4 software, determine whether the required ports are available by running the following commands:

   - SAS Object Spawner:

     ```
     netstat -an | grep 8591
     ```

   - SAS/CONNECT:

     ```
     netstat -an | grep 17551
     ```

     Note: If SAS/CONNECT is not included in your software order, skip this command.

   If a command does not produce any results, then the port is available for use and no changes are required. If the command produces results, then the port is already being used by a product and is blocked for other products to use. Note any blocked products in preparation for additional steps to be performed after the installation has completed.

2. Copy the final yum groupinstall command from the customized_deployment_script.sh file on the connected mirror repository. Here is an example:

   ```
   # Install the software
   yum groupinstall "SAS/CONNECT" "SAS Machine Learning" "SAS CAS for Event Stream Processing"
   "SAS CAS for Machine Learning" "SAS Statistics" "SAS CAS for Statistics" "SAS Foundation"
   "SAS CAS for Visual Analytics"
   ```

3. Run the copied yum groupinstall command on each machine on which SAS Viya will be installed.

4. If you have any blocked products from step 1, modify the required file as follows:

   - SAS Object Spawner:

     Open the **`/opt/sas/viya/config/etc/spawner/default/spawner.cfg`** file. Change the value for the sasPort variable to an available port number.

     Also open the**`/opt/sas/viya/config/etc/sasstudio/default/init_usermods.properties`** file. Change the webdms.workspaceServer.port value to the same port number that is used in the**`/opt/sas/viya/config/etc/spawner/default/spawner.cfg`** file.

   - SAS/CONNECT:

     Open the **`/opt/sas/viya/config/etc/sysconfig/connect/default/sas-connect`** file. Change the value for the CONNECT_PORT variable to an available port number.

5. Complete the deployment by following the steps described in Appendix B: Deploying with Yum on page 86, starting with Deploy the HTTP Daemon and the SSL Module on page 86.

Appendix D: Hadoop Deployment: Configuring SAS Access to Hadoop, SAS Data Connector to Hadoop and Optionally, the SAS Data Connect Accelerator for Hadoop

Supported Hadoop Distributions

Before you set up Hadoop, you must make sure that your Hadoop distribution is supported by SAS Viya. For details, see Supported Releases of Hadoop Distributions on page 21.

Deployment Tasks for Hive Access

For Hive access, perform the following tasks:

1. Perform the pre-deployment tasks. For more information, see Pre-deployment Hadoop Tasks for Hive Access on page 105.
2. Deploy the Hadoop JAR files. For more information, see Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop on page 107.
3. If you are using the SAS Data Connect Accelerator for Hadoop, deploy the SAS Embedded Process. For more information, see Deploy the SAS Embedded Process for Hadoop for Parallel Processing on page 111.

Pre-deployment Hadoop Tasks for Hive Access

Pre-deployment Checklist for Hive Access

Before you install SAS Viya software that interacts with Hadoop and Hive, 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 Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop on page 44.
- 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. This user account is used to manually deploy the SAS Embedded Process in the Deploy Manually on page 112 section.
- 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 information about how CAS uses memory, see "Memory" in SAS Cloud Analytic Services: Fundamentals.
- 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
- 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 hadoop_extract.sh 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:

   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.

   ```
   ssh-copy-id Hive-server-machine
   ```

3 Edit the target reference list in the inventory file to define the Hadoop cluster information.

   a If you deployed using the host_local file, edit the host_local file to add the Hadoop cluster machine to the list of target references at the beginning of the hosts file. For more information, see Define the Machines in the Deployment on page 36.

   b If you deployed using the hosts file, edit the hosts file to add the Hadoop cluster machine to the list of target references at the beginning of the hosts file. For more information, see Define the Machines in the Deployment on page 36.

4 In the inventory file, add a machine target for the Hadoop Hive node. Also, beneath the list of target machines, add the [hadoopCluster] group. Add the new Hadoop machine target to the new group.

   ```
   hadoop-cluster ansible_ssh_host=hive.example.com ansible_ssh_user=user1
   ```

```
[hadoopCluster]
hadoop-cluster"
```

For more information see Assign the Target Machines to Host Groups on page 36.

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 Run one of the following commands, depending on whether you are performing a single-machine deployment or a multi-machine deployment:

Ansible will copy files to the Hadoop cluster node and then to the CAS controller and SAS programming nodes.

- single-machine deployment: `ansible-playbook -i host_local hadooptracer-launch.yml`
- multi-machine deployment: `ansible-playbook -i hosts hadooptracer-launch.yml`

9 (Cloudera and Hortonworks distributions only) To verify that the required Hadoop JAR files are successfully collected, run one of the following commands:

- single-machine deployment: `ansible-playbook -i host_local hadooptracer-validation.yml`
- multi-machine deployment: `ansible-playbook -i hosts hadooptracer-validation.yml`

Install the Hadoop JAR Files Manually

The hadoop_extract script and the sas_hadoop_config.properties file are located on the CAS controller machine. The hadoop_extract script collects the Hadoop library JAR files and its configuration files from the Hadoop cluster. It also makes the files available to the SAS Viya products that require access to the Hadoop cluster. The hadoop_extract script uses information from the sas_hadoop_config.properties file.

Note: The hadoop_extract script was formerly known as the Hadoop tracer script.

Note: As an alternative, you can use the `-p` option to specify an alternative properties file.

1 Ensure that Python, strace, and wget have been installed on the Hadoop cluster from the package repositories for your Linux distribution.

2 Ensure that the user who runs the script has a home directory in HDFS that has Read and Write access. For example, the user jsmith who is running the script owns the **`/user/jsmith`** home directory.

3 Locate the sas_hadoop_config.properties file or an alternative properties file on the CAS controller machine in the **`/opt/sas/viya/home/SASFoundation/etc`** directory. Edit the appropriate .properties file and make the following changes:

a Set the distribution name:

```
hadoop.cluster.distribution.name=distribution
```

distribution = **`cloudera`** | **`hortonworks`** | **`mapr`**

b Set the qualified host name of the node on which the Hadoop Hive service is deployed.

```
hadoop.cluster.hivenode.hostname=hostname
```

c Ensure that the following requirements have been met on the machine on which the Hive2 services is running.

- A valid SSH account.
- A home directory for the hadooptracer.log file. The hadooptracer.log file is written to the home directory of the hadoop.cluster.hivenode.ssh.account user.
- If your Hadoop cluster includes Kerberos that has been enabled, the user account should also include a configured Kerberos principal. A valid Kerberos ticket for the same user account must be available on the node on which the Hive2 service is running.

d Set the user name and password for SSH authentication to the machine on which the Hive2 service is running. Instead of entering an SSH password, the password property can be left blank in order to be prompted for the password.

```
hadoop.cluster.hivenode.ssh.account=user-account-name
```

Note: The user account is not required to also be an administrative account. The user account must be a Hadoop user account.

e Set the directories to which the script will store XML and JAR files:

```
hadoop.client.jar.filepath=directory-path
hadoop.client.config.filepath=directory-path
```

Note: Each of these paths for hadoop.client.config.filepath, hadoop.client.jar.filepath, and hadoop.client.configfile.repository must be unique.

On the CAS controller, ensure that Write permission has been granted to the directories that are specified in the hadoop.client.jar.filepath property and the hadoop.client.config.filepath property.

f Set the location to which the JAR files and configuration files are backed up. The script creates a new directory **`hive/`*****`hivenode-name`*****`/`*****`time-stamp`*** under the specified repository.

```
hadoop.client.configfile.repository=directory-path
```

Note: The paths for hadoop.client.config.filepath, hadoop.client.jar.filepath, and hadoop.client.configfile.repository must be different.

g Set the directory part and the first part of the log filename. The script creates a log file and names it using the first part of the log filename with a timestamp. The script creates the file sashadoopconfig_*time-stamp*.log. An example of a filename is sashadoopconfig_2017-04-14-10.16.33.log.

```
hadoop.client.sasconfig.logfile.name=/directory-path/sashadoopconfig
```

h To increase the amount of information that is logged, change the default value of the following properties from 0 to 3:

```
hadoop.client.config.log.level=3
```

Here are the supported values:

1 (default)
adds INFO messages.

2
adds DEBUG messages.

3
adds consoleAappender to the log plus level 1 (HadoopTracer.py output).

i Select the option that specifies how the script should filter the JAR files. Using this option, the script detects any duplicate JAR files (files with the same name) and replaces them with files that are based on the selected option.

  Here are the supported values:

  latest (default)
    Duplicate JAR files are replaced by the latest version.

  none
    JAR files are extracted without filtering.

  When you run the hadoop_extract.sh script, by default, any duplicate names of JAR files that are extracted from the cluster are removed. The latest version of the JAR file with the duplicate name is copied to the specified location. To keep multiple versions of the JAR files, set the hadoop.tracer.filter in the sas_hadoop_config.properties file to **`none`**.The default is **`latest`**.

```
hadoop.tracer.filter=latest
```

4 For MapR, add the JAR filename hadoop-0.20.2-dev-core.jar to the current exclusion list as follows:

```
hadoop.jar.exclusion.list=derby,spark-examples,hadoop-0.20.2-dev-core.jar
```

5 Locate the installation directory on the CAS controller machine, and navigate to the **`/opt/sas/viya/home/SASFoundation/utilities/bin`** directory, which contains the script.

  Note: The user who runs the script must have a PATH that includes the required Java version (1.8 or later release).

  Note: You can specify a different properties file by specifying the `-p` option. Here is an example:

```
./hadoop_extract.sh -p alternative-properties-file
```

  Run the script:

```
./hadoop_extract.sh
```

  You are prompted for the Hive password, which is the password for the SSH user account. The SSH user account connects to the Hadoop cluster that corresponds to the `hadoop.cluster.hivenode.ssh.account` name. The account name is specified in the sas_hadoop_config.properties file.

  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.

6 If your deployment includes MapReduce users from Windows clients, locate the mapred-site.xml file in the hadoop-client.config.filepath directory. Edit the mapred-site.xml file and set the property mapreduce.app-submission.cross-platform equal to **`true`**. Here is an example:

```
<property>
 <name>mapreduce.app-submission.cross-platform</name>
 <value>true</value>
 </property>
```

  Note: Be sure to make this modification after you run the hadoop_extract.sh script.

Verify SAS Data Connector to Hadoop

To verify that the software has been successfully deployed:

1 Sign on to SAS Studio:

  a Open SAS Studio from a URL with the following format: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code:

```
cas mysession;

caslib hivelib datasource=(srctype="hadoop" server="server-name"
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;
list files incaslib="hivelib";
run;
```

If the data connector is successfully deployed, the results are a list of the names of tables in the Hive data source.

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.

Deploy the SAS Embedded Process for Hadoop for Parallel Processing

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.
- Deploying files across all nodes requires passwordless SSH.

Note: If you run the SAS Embedded Process installation script (sasep-admin.sh) with sudo access, the script detects the Hadoop cluster topology and automatically deploys the files across all data nodes. Otherwise, you must specify the hosts on which the SAS Embedded Process for Hadoop is installed when you run the install script.

Note: The passwordless SSH user must also have Read, Write, and Execute permissions on the installation directory. The directory structure of the nodes in must match that of the installation directory.

- Writing the configuration file requires Write permission to HDFS.

  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.

SAS Viya and SAS 9.4 on the Same Machine

- If you are running the fourth maintenance release of SAS 9.4 and SAS Viya on the same machine, you must use the SAS Embedded Process from the fourth maintenance release of SAS 9.4. Do not perform the steps to uninstall or install the SAS Embedded Process.
- If you are running the third maintenance release of SAS 9.4 (or an earlier release of SAS) and SAS Viya on the same machine, perform the steps to uninstall the SAS Embedded Process for SAS 9.4 and install the SAS Embedded Process for SAS Viya.
- If you are not running SAS 9.4 and SAS Viya on the same machine, install the SAS Embedded Process for SAS Viya.

For details about how to uninstall, see Options for Uninstallation on page 127.

Deploy the SAS Embedded Process

Methods to Deploy the SAS Embedded Process

CAUTION! If you are running the SAS Embedded Process for the fourth maintenance release of 9.4 on the same machine as SAS Viya, do not install the SAS Embedded Process for Viya.

You can either deploy manually or deploy automatically by using the cluster manager for your Hadoop distribution:

- To deploy manually, see Deploy Manually on page 112.

  TIP Many options are available for installing the SAS Embedded Process.For more information, see SASEP-ADMIN.SH Script on page 118.

- To deploy with your appropriate cluster manager:
  - To deploy with Cloudera Manager, see Deploy the SAS Embedded Process with Cloudera Manager on page 115.
  - To deploy with Hortonworks Ambari, see Deploy the SAS Embedded Process with Ambari on page 117.

CAUTION! You must uninstall the SAS 9.4 Embedded Process or SAS VIya Embedded Process using the same method that you used to install the SAS 9.4 Embedded Process or SAS Viya Embedded Process. For details about how to uninstall, see Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya on page 127.

Deploy Manually

1 On the Hadoop master node, create a new directory that is not part of an existing directory structure such as **/opt/sasep**.

This path is created on each node in the Hadoop cluster during installation of the SAS Embedded Process. It is recommended that you do not use existing system directories such as **`/usr`**. This new directory is referred to as *EPInstallDir* throughout this section.

2. On the CAS controller node, navigate to the **`/opt/sas/viya/home/share/ep`** directory.

3. Locate the sepcorehadp-11.50000-*n*.sh file, where *n* is a number that indicates the latest version of the file.

4. Copy the sepcorehadp-11.50000-*n*.sh file from the client to *EPInstallDir* on the Hadoop cluster. Here is an example that uses the secure copy command:

   ```
   scp sepcorehadp-11.50000-n.sh username@hdpclus1:/EPInstallDir
   ```

   Note: The location to which you transfer the sepcorehadp-11.50000-*n*.sh file becomes the SAS Embedded Process home and is referred to as *EPInstallDir* throughout this section.

To install the SAS Embedded Process for Hadoop, follow these steps:

Note: Passwordless SSH is required in order to install the SAS Embedded Process for Hadoop. Also, Write permission to HDFS might be required. For more information, see Hadoop Prerequisites on page 111.

1. Navigate to the location on your Hadoop master node to which you copied the sepcorehadp-11.50000-*n*.sh file.

   ```
   cd /EPInstallDir
   ```

2. Use the following command to unpack the sepcorehadp-11.50000-*n*.sh file.

   ```
   ./sepcorehadp-11.50000-n.sh [--verbose]
   ```

   Note: The `--quiet` option is enabled by default. Only error messages are displayed. The `--verbose` option causes all messages to be displayed that are generated during the installation process. Using verbose messaging can increase the time that is required to perform the installation.

   After this script has completed its execution and the files are unpacked, the following directory structure is created:

   ```
   EPInstallDir/SASEPHome
   EPInstallDir/sepcorehadp-11.50000-n.sh
   ```

   Note: During the installation process, the sepcorehadp-11.50000-*n*.sh is copied to all data nodes. Do not remove or move this file from the **`EPInstallDir/SASEPHome`** directory.

   The SASEPHome directory should have the following structure:

   ```
   EPInstallDir/SASEPHome/bin
   EPInstallDir/SASEPHome/jars
   EPInstallDir/SASEPHome/misc
   EPInstallDir/SASEPHome/sasexe
   EPInstallDir/SASEPHome/utilities
   ```

   The **`EPInstallDir/SASEPHome/jars`** directory contains the SAS Embedded Process JAR files:

   ```
   EPInstallDir/SASEPHome/jars/sasephdp0-*.jar
   EPInstallDir/SASEPHome/jars/sasephdp1-*.jar
   EPInstallDir/SASEPHome/jars/sasephdp2-*.jar
   ```

   The **`EPInstallDir/SASEPHome/bin`** directory should contain the following script:

   ```
   EPInstallDir/SASEPHome/bin/sasep-admin.sh
   ```

3. If your Hadoop cluster is secured with Kerberos and you have sudo access, the HDFS user must have a valid Kerberos ticket in order to access HDFS. You can obtain a valid Kerberos ticket with the **kinit** command.

   ```
   sudo su - root
   su - hdfs | hdfs-userid
   ```

```
kinit -kt location-of-keytab-file-user-for-which-you-are-requesting-a-ticket principal-name
exit
```

Note: The default HDFS user is **hdfs**. You can specify a different user ID with the -hdfsuser argument when you run the `sasep-admin.sh -add` script. If you use a different hdfs superuser, ensure that the user has a home directory in HDFS before you run the sasep-admin.sh -add command. For example, if the hdfs superuser is prodhdfs, ensure that the **/user/prodhdfs** directory exists in HDFS.

To check the status of your Kerberos ticket on the server, as the hdfs user, run the **klist** command. Here is an example of the command and its output:

```
klist
Ticket cache: FILE/tmp/krb5cc_493
Default principal: hdfs@HOST.COMPANY.COM

Valid starting    Expires           Service principal
06/20/15 09:51:26 06/27/15 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
     renew until 06/22/15 09:51:26
```

4 Run the sasep-admin.sh script depending on whether you have sudo access.

If you have sudo access, complete the following steps to deploy the SAS Embedded Process on all nodes. Review all of the information in this step and the script syntax before you run the script.

a Run the sasep-admin.sh script as follows.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -add
```

b The sepcorehadp-11.00000-n.sh file is copied to all data nodes.

Note: If you have sudo access, the SAS Embedded Process installation script (sasep-admin.sh) detects the Hadoop cluster topology and installs the SAS Embedded Process on all DataNode nodes. The install script also installs the SAS Embedded Process on the host node from which you run the script (the Hadoop master NameNode). The SAS Embedded Process is installed even if a DataNode is not present. To add the SAS Embedded Process to new nodes at a later time, you should run the sasep-admin.sh script with the **-host** <*hosts*> option. In addition, a configuration file, ep-config.xml, is automatically created and written to the ***EPInstallDir*/SASEPHome/conf** directory and to the HDFS file system in the **/sas/ep/config** directory.

If you do not have sudo access, complete the following steps to deploy the SAS Embedded Process installation across all nodes. Review all of the information in this step and the script syntax before you run the script.

Note: If you do not have sudo access, the passwordless SSH user must have Read, Write, and Execute permissions on the *EPInstallDir* directory.

a Run the sasep-admin.sh script as follows:

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -x -add -hostfile host-list-filename | -host <">host-list<">
```

Note: If you do not have sudo access, you must use the **-x** option and specify the hosts on which the SAS Embedded Process is deployed with either the **-hostfile** or **-host** option. Automatic detection of the Hadoop cluster topology is not available when you run the installation script with the **-x** option.

CAUTION! The SAS Embedded Process must be installed on all nodes that are capable of running a MapReduce job. The SAS Embedded Process must also be installed on the host node from which you run the script (the Hadoop master NameNode). Otherwise, the SAS Embedded Process does not function properly.

The sepcorehadp-11.50000-*n*.sh file is copied to all nodes that you specify. The configuration file, ep-config.xml, is created and written to the ***EPInstallDir*/SASEPHome/conf** directory.

b Manually copy the ep-config.xml configuration file to HDFS.

Note: This step must be performed by a user that has Write permission to the HDFS root folder /. If your Hadoop cluster is secured with Kerberos, the user who copies the configuration file to HDFS must have a valid Kerberos ticket.

i Log on as your HDFS user or as the user that you use to access HDFS.

ii Create the **/sas/ep/config** directory for the configuration file.

```
hadoop fs -mkdir -p /sas/ep/config
```

iii Navigate to the ***EPInstallDir*/SASEPHome/conf** directory.

iv Use the Hadoop **copyFromLocal** command to copy the ep-config.xml file to HDFS.

```
hadoop fs -copyFromLocal ep-config.xml /sas/ep/config/ep-config.xml
```

5 Verify that the SAS Embedded Process was successfully installed by running the sasep-admin.sh script with the -check option.

If you ran the sasep-admin.sh script with sudo access, run the following command. By default, this command verifies that the SAS Embedded Process was installed on all nodes.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -check
```

If you ran the sasep-admin.sh script with the -x argument, run the following command. This command verifies that the SAS Embedded Process was installed on the hosts that you specified.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -x -check -hostfile host-list-filename | -host <">host-list<">
```

6 Verify that the configuration file, ep-config.xml, was written to the HDFS file system.

```
hadoop fs -ls /sas/ep/config/ep-config.xml
hadoop fs -cat /sas/ep/config/ep-config.xml
```

Note: If your Hadoop cluster is secured with Kerberos, you must have a valid Kerberos ticket in order to access HDFS. Otherwise, you can use the WebHDFS browser.

Note: The **/sas/ep/config** directory is created automatically when you run the installation script with sudo access. If you used the -genconfig option to specify a non-default location, use that location to locate the ep-config.xml file.

Deploy the SAS Embedded Process with Cloudera Manager

The following deployment steps assume either of these scenarios: the SASEP rpm package has been installed directly on the Cloudera Manager server, or the SASEP rpm package has been installed on a network location that is accessible to the Cloudera Manager server.

To deploy the SAS Embedded Process:

1 On the CAS controller machine, create a temporary directory on the file system of the host on which Cloudera Manager is installed.

```
mkdir -p /tmp/sasep
```

2 Navigate to the **/opt/sas/viya/home/share/ep/parcel** directory.

```
cd /opt/sas/viya/home/share/ep/parcel
```

3 Copy the parcel directory to the new directory under **tmp**:

```
cp -r * /tmp/sasep
```

4 Grant permission to the user account that you use to run the install_parcel.sh script. The user account must have super user (sudo) access or root access and must have Execute permission on the script. Here is an example:

5 From the `tmp` directory, run the following command:

```
./install_parcel.sh  -v distro
```

The `tmp` directory is the location to which you copied the parcel directory from the SAS Viya installation. The variable *distro* represents one of the following Linux distributions: redhat5, redhat6, redhat7, suse11x, ubuntu10, ubuntu12, ubuntu14, debian6, or debian7. Select the appropriate value.

Here is an example:

```
./install_parcel.sh  -v redhat6
```

6 When prompted to restart Cloudera Manager, select **y**.

7 Log on to Cloudera Manager.

8 Activate the SASEP parcel:

a From the Menu bar, select **Hosts** ⇨ **Parcels**.

Note: If the SASEP parcel is missing, run **Check for new parcel**.

b On the row for the SAS EP parcel, click **Distribute** to copy the parcel to all nodes.

c Click **Activate**. Answer OK to the Activation prompt. You might be prompted to either restart the cluster or to close the window.

CAUTION! Do not restart the cluster.

d If prompted, click **Close**.

9 Add the SASEP service to create the SASEP configuration file in HDFS.

a Navigate to Cloudera Manager Home.

b In Cloudera Manager, select the ▼ next to the name of the cluster, and then select **Add a Service**. The Add Service Wizard appears.

c Select the SASEP service and click **Continue**.

d On the **Add Service Wizard** ⇨ **Select the set of dependencies for your new service** page, select the dependencies for the service. Click **Continue**.

Note: The dependencies are automatically selected for this service.

e On the **Add Service Wizard** ⇨ **Customize Role Assignments** page, select a node for the service. In the next step you must have a valid Kerberos keytab, so be sure to select a node that has a Kerberos keytab for the hdfs user on that node. Choose any single node. Click **OK**, and then click **Continue**.

f Enter your hdfs user name. The default user name is hdfs. If your cluster is Kerberos enabled, a valid Kerberos keytab for your hdfs user name must be available on the node that was selected for the SAS Embedded Process service.

g Click **Continue**, and then click **Finish**.

A file is added to HDFS for each of the services as follows:

SASEP: **`/sas/ep/config/ep-config.xml`**

Note: If the service that you have just deployed is started, navigate to Cloudera Manager Home and stop the services.

Deploy the SAS Embedded Process with Ambari

1 To launch the script:

a On the CAS controller machine, navigate to the **`/opt/sas/viya/home/share/ep/stack`** directory. Copy the entire stack directory to a temporary directory (/tmp) on your Hadoop Ambari Cluster Manager machine.

b Navigate to the**`/tmp/stack`** directory and run the following command with sudo or as root:

```
./install_sasepstack.sh ambariAdminUsernam
```

Note: To complete the installation process, the Ambari server must be restarted.

- To restart the Ambari server with the script, enter **y**.
- ITo manually restart the script at a later time, press **n**.

After the script finishes, the following message is displayed:

```
You can install the SASEP stack now from Ambari Cluster Manager.
```

2 On the Ambari server, log on to Ambari and deploy the services:

a Click **Actions** and select **+ Add Service**.

The **Add Service Wizard** page and the **Choose Services** panel appear.

b In the **Choose Services** panel, select the **SASEP** service. Click **Next**.

The **Assign Slaves and Clients** panel appears.

c In the **Assign Slaves and Clients** panel, ensure that the NameNode, HDFS_CLIENT, and HCAT_CLIENT are selected where you want the stack to be deployed. By default, the three clients are selected. SAS recommends that you select all clients.

Note: On the **Assign Slaves and Clients** panel, hover over the host name to view the details for NameNode, HDFS_CLIENT, and HCAT_CLIENT.

d Click **Next**. The **Customize Services** panel appears.

The SASEP service stacks are listed.

e Do not change any settings on the **Customize Services** panel. Click **Next**.

Note: By default, Ambari will not retain the credentials that you provide unless you have configured encrypted passwords for storage in Ambari. If you have not configured Ambari for password encryption, you will be prompted to provide credentials whenever cluster changes are made.

If your cluster is secured with Kerberos, the **Configure Identities** panel appears. Enter your Kerberos credentials in the **admin_principal** and **admin_password** text boxes. Click **Next**.

The **Review** panel appears.

f Review the information about the panel. If the information is correct, click **Deploy**.

The **Install, Start, and Test** panel appears. After the stack is installed on all nodes, click **Next**.

The **Summary** panel appears.

g Click **Complete**. The stacks are now installed on all nodes of the cluster.

The SASEP service is displayed on the Ambari dashboard.

h After you deploy all of the services, verify that the following file exists in the Hadoop file system:

SASEP: **`/sas/ep/config/ep-config.xml`**

SASEP-ADMIN.SH Script

Overview of the SASEP-ADMIN.SH Script

The sasep-admin.sh script enables you to perform the following actions:

- Install or uninstall the SAS Embedded Process for Hadoop on a single node or a group of nodes.
- Generate a SAS Embedded Process configuration file and write the file to an HDFS location.
- Install a hot fix to the SAS Embedded Process.
- Check whether the SAS Embedded Process is installed correctly.
- Display all live data nodes on the Hadoop cluster.
- Display the Hadoop configuration environment.
- Display the Hadoop version information for the Hadoop cluster.
- Display the version of the SAS Embedded Process that is installed.
- Deploy the security settings for SAS Data Connect Accelerator for Hadoop across all nodes in the cluster.

Note: The installation of the SAS Embedded Process for Hadoop involves writing a configuration file to HDFS and deploying files on all data nodes. These two tasks can occur automatically, depending on your Hadoop and HDFS permissions.

If you run the SAS Embedded Process install script (sasep-admin.sh) with sudo access, the script detects the Hadoop cluster topology and installs the SAS Embedded Process on all DataNode nodes. The install script also installs the SAS Embedded Process on the host node on which you run the script (the Hadoop master NameNode). In addition, a configuration file, ep-config.xml, is created and written to the HDFS file system.

If you do not have sudo access, you must specify the hosts on which the SAS Embedded Process is installed. In addition, you must manually copy the ep-config.xml configuration file to the HDFS file system.

SASEP-ADMIN.SH Syntax

Action options syntax:

sasep-admin.sh

<-x> -add < -hostfile *host-list-filename* | -host <">*host-list*<"> >
<-maxscp *number-of-copies* > <-hdfsuser *user-ID* >

sasep-admin.sh

<-x> -genconfig < *HDFS-filename* > <-force>

sasep-admin.sh

<-x > -hotfix *hotfix-filename* < -hostfile *host-list-filename* | -host <">*host-list*<"> >
<-maxscp *number-of-copies* > <-hdfsuser *user-ID* >

sasep-admin.sh

<-x > -remove < -hostfile *host-list-filename* | -host <">*host-list*<"> >
<-hdfsuser *user-ID* >

sasep-admin.sh

<-x > -security deploy | reset < -hostfile *host-list-filename* | -host <">*host-list*<"> >
<-force>

Informational options syntax:

sasep-admin.sh <-x > <-check < -hostfile *host-list-filename* | -host <">*host-list*<"> > <-hdfsuser *user-ID* > >

sasep-admin.sh <-env>

sasep-admin.sh <-hadoopversion >

sasep-admin.sh <-nodelist>

sasep-admin.sh <-version >

Action Arguments

-add

installs the SAS Embedded Process.

Requirement If you have sudo access, the script automatically retrieves the list of data nodes from the Hadoop configuration. If you do not have sudo access, you must use the -x argument and either the -hostfile or -host argument.

Tip If you add nodes to the Hadoop cluster, you can specify the hosts on which to install the SAS Embedded Process by using the -hostfile or -host option. The -hostfile option and the -host option are mutually exclusive.

See -hostfile on page 121 and -host on page 121

-genconfig <*HDFS-filename*> <-force>

generates the SAS Embedded Process configuration file in the ***EPInstallDir*/SASEPHome/conf** directory of the local file system.

Requirement If you do not have sudo access, you must use the -x argument.

Interactions When used without the -x argument, the script creates the ep-config.xml configuration file and writes the file to both the *EPInstallDir*/SASEPHome/conf directory on the local file system and the /sas/ep/config/ directory on HDFS. You can change the filename and the HDFS location by using the *HDFS-filename* argument. *HDFS-filename* must be the fully qualified HDFS pathname that points to the location of the configuration file.

When used with the -x argument, the script does not write the configuration file to the HDFS. You must manually copy the file to the HDFS. Deploy Manually on page 112

Note The -genconfig argument creates two identical configuration files under *EPInstallDir*/ SASEPHome/conf/ on the local file system: ep-config.xml and sasep-site.xml. The sasep-site.xml file might be copied to the client side under a folder that is in the classpath. When the sasep-site.xml file is loaded from the classpath, the configuration file on the HDFS location is not used. However, if sasep-site.xml is not found in the classpath, a configuration file must exist on the HDFS. The configuration file must exist either on the default HDFS location /sas/ep/config/ep-config.xml or in the location that is set in the sas.ep.config.file property.

Tips Use the -genconfig argument to generate a new SAS Embedded Process configuration file when you upgrade to a new version of your Hadoop distribution.

Use the HDFS-filename argument to specify another location and configuration filename. If you decide to generate the configuration file in a non-default HDFS location, you must set the sas.ep.config.file property in the mapred-site.xml to the value that you specify in the -genconfig option.

This argument generates an updated ep-config.xml file. Use the -force argument to overwrite the existing configuration file.

Examples The following example generates the configuration files under ***EPInstallDir*/SASEPHome/conf** on the local file system and the ep-config.xml configuration file under **/sas/ep/config** on the HDFS:

```
./sasep-admin.sh -genconfig
```

The following example overwrites the configuration files under ***`EPInstallDir`***`/SASEPHome/conf` on the local file system and under `/sas/ep/config` on the HDFS, if it already exists:

```
./sasep-admin.sh -genconfig -force
```

The following example generates the configuration files under ***`EPInstallDir`***`/SASEPHome/conf` on the local file system and under `/home/hadoop/` on the HDFS:

```
./sasep-admin.sh -genconfig /home/hadoop/ep-config.xml
```

The following example generates the configuration files under ***`EPInstallDir`***`/SASEPHome/conf` on the local file system only:

```
./sasep-admin.sh -x -genconfig
```

The following example overwrites the configuration files under ***`EPInstallDir`***`/SASEPHome/conf` on the local file system only:

```
./sasep-admin.sh -x -genconfig -force
```

-hotfix *hotfix-filename*

distributes a hot fix package.

Requirements Hot fixes must be installed using the same user ID that performed the initial software installation.

Hot fixes should be installed following the installation instructions provided by SAS Technical Support.

-remove

removes the SAS Embedded Process.

Requirement If you do not have sudo access, you must use the -x argument and either the -hostfile or -host argument. The -hostfile option and the -host option are mutually exclusive.

Interactions When used without the -x argument and you have sudo access, the script automatically retrieves the list of data nodes from the Hadoop configuration. In addition, the script automatically removes the epconfig.xml file from the HDFS.

When used with -x argument, the SAS Embedded Process is removed from all hosts that you specify. However, the ep-config.xml file must be removed manually from the HDFS.

See -hostfile on page 121 and -host on page 121

- security deploy | reset <-force>

deploys or resets security settings across all nodes in the cluster.

Requirement If you do not have sudo access, you must use the -x argument.

Note To overwrite security settings without a prompt, use the -force argument.

Tip You can specify one or more hosts for which you want to check the SAS Embedded Process by using the -hostfile or -host option. The -hostfile option and the -host option are mutually exclusive.

See -hostfile on page 121 and -host on page 121

-x on page 121

“Encrypt Data Transfer when Using the SAS Data Connect Accelerator” in *Encryption in SAS Viya 3.2*

Informational Arguments

-check

checks whether the SAS Embedded Process is installed correctly on all data nodes.

Tip You can specify the hosts for which you want to check the SAS Embedded Process by using the -hostfile or -host option. The -hostfile option and the -host option are mutually exclusive..

See -hostfile on page 121 and -host on page 121

-env

displays the SAS Embedded Process install script and the Hadoop configuration environment.

-hadoopversion

displays the Hadoop version information for the Hadoop cluster.

-nodelist

displays all live DataNodes on the Hadoop cluster.

Requirement sudo access is required.

-version

displays the version of the SAS Embedded Process that is installed.

Parameters for Action and Informational Arguments

-x

if you do not have sudo access, runs the script solely under the current user’s credential.

Requirements This option must be the first argument passed to the script.

A list of hosts must be provided with either the -hostfile or -host argument.

If you do not have sudo access, you must use the -x argument.

Interaction If you use the -x argument to install the SAS Embedded Process, that is, with the -add argument, you must use the -x argument in any other sasep-admin.sh script action that supports it.

See -hostfile on page 121 and -host on page 121

-hostfile *host-list-filename*

specifies the full path of a file that contains the list of hosts on which the SAS Embedded Process is installed or removed.

Requirement The -hostfile or -host argument is required if you do not have sudo access.

Interaction Use the -hostfile argument in conjunction with the -add, -hotfix, or -remove arguments.

See -hdfs on page 122

Example

```
-hostfile /opt/sasep/ep.hosts
```

-host <"> *host-list* <">

specifies the target host or host list on which the SAS Embedded Process is installed or removed.

Requirements If you specify more than one host, the hosts must be enclosed in double quotation marks and separated by spaces or commas.

The -host or -hostfile argument is required if you do not have sudo access.

Interaction	Use the -host argument in conjunction with the -add, -hotfix, or -remove arguments.
See	-hdfs on page 122

Example	`-host "server1 server2 server3"`
	`-host bluesvr`
	`-host "blue1, blue2, blue3"`

-maxscp *number-of-copies*
specifies the maximum number of parallel copies between the master and data nodes.

Default	10
Interaction	Use this argument in conjunction with the -add or -hotfix argument.

-hdfsuser *user-ID*
specifies the user ID that has Write access to the HDFS root directory.

Note: The hdfs folder **`/users/user-id`** must exist. Otherwise, the command fails.

Default	hdfs
Interactions	This argument has no affect if you use the -x argument.
	Use the -hdfsuser argument in conjunction with the -add, -check, or -remove argument in order to change, check, or remove the HDFS user ID.
Note	The user ID is used to copy the SAS Embedded Process configuration files to the HDFS.

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 Sign on to SAS Studio:

   a Open SAS Studio from a URL with the following format: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code:

```
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="Hive table to load" casout="CAS table name"
    incaslib="hivelib";
    run;
```

The SAS code loads the table from Hive into CAS. You can 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.

Additional Configuration for HCatalog File Formats

Overview of HCatalog File Types

HCatalog is a table management layer that presents a relational view of data in the HDFS to applications within Hadoop. With HCatalog, data structures that are registered in the Hive metastore, including SAS data, can be accessed through standard MapReduce code and Apache Pig. HCatalog is included in Apache Hive.

The SAS Embedded Process for Hadoop uses HCatalog to process the following complex, non-delimited Apache file formats: Avro, ORC, Parquet, and RCFile.

Prerequisites for HCatalog Support

Here are additional prerequisites for accessing complex, non-delimited file types such as Avro or Parquet:

- Hive and HCatalog must be installed on all nodes of the Hadoop cluster.
- HCatalog support depends on the version of Hive that is running on your Hadoop distribution. See the following table for more information.

  Note: For MapR distributions, Hive 0.13.0 build: 1501 or later must be installed for access to any HCatalog file type.

File Type	Required Hive Version
Avro	0.14
ORC	0.11
Parquet	0.13
RCFile	0.6

SAS Client Configuration

Note: If you used the hadoop_extract.sh script to install the Hadoop JAR files, the configuration tasks in this section are unnecessary. SAS client configuration was completed using the script. For more information, see Install the Hadoop JAR Files on the CAS Controller on page 107.

The following additional configuration tasks must be performed:

- The hive-site.xml configuration file must be included in the hadoopConfigDir path.

- The following Hive or HCatalog JAR files must be included in the hadoopJarPath path.
  - hive-hcatalog-core-*.jar
  - hive-webhcat-java-client-*.jar
  - jdo-api*.jar
- If you are using MapR, the following Hive or HCatalog JAR files must be included in the SAS_HADOOP_JAR_PATH.
  - hive-hcatalog-hbase-storage-handler-0.13.0-mapr-1408.jar
  - hive-hcatalog-server-extensions-0.13.0-mapr-1408.jar
  - hive-hcatalog-pig-adapter-0.13.0-mapr-1408.jar
  - datanucleus-api-jdo-3.2.6.jar
  - datanucleus-core-3.2.10.jar
  - datanucleus-rdbms-3.2.9.jar

For more information about the hadoopConfigDir path and the hadoopJarPath path, see the CASLIB statement in the *SAS Viya Cloud Analytic Services: Language Reference*.

SAS Server-Side Configuration

The SAS Embedded Process deployment automatically sets the HCatalog CLASSPATH in the ep-config.xml file. You could also manually append the HCatalog CLASSPATH to the MapReduce configuration property mapreduce.application.classpath in the mapred-site.xml file on the client side.

Here is an example of an HCatalog CLASSPATH for a Cloudera distribution:

```
/opt/cloudera/parcels/CDH-version/bin/../lib/hive/lib/*,
   /opt/cloudera/parcels/CDH-version/lib/hive-hcatalog/libexec/../share/hcatalog/*
```

Here is an example of an HCatalog CLASSPATH for a Hortonworks distribution:

```
/usr/hdp/version/hive-hcatalog/libexec/../share/hcatalog/*,/usr/hdp/2.4.2.0-258/hive/lib/*
```

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_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/*
```

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 125.

- the number of the SAS Embedded Process MapReduce tasks per node

  For more information, see Specify the Number of MapReduce Tasks on page 126.

- 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 126.

- the buffers for input data

  For more information, see Specify the Number of Input Buffers and an Optimal Buffer Size on page 126.

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).

```
<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.

```
<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 is allowed to use. 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.

```
<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>
```

Add the SAS Embedded Process to Nodes after the Initial Deployment

After the initial deployment of the SAS Embedded Process, you might add more nodes to your Hadoop cluster. Also, you might replace selected nodes. In these instances, you can install the SAS Embedded Process on the new nodes.

Run the sasep-admin.sh script and specify the nodes on which to install the SAS Embedded Process. For more information, see the -add argument in SASEP-ADMIN.SH Syntax on page 118.

Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya

Options for Uninstallation

CAUTION! If you are running the SAS Embedded Process for the fourth maintenance release of SAS 9.4, do not uninstall it.

CAUTION! You must uninstall the SAS 9.4 Embedded Process or the SAS Viya Embedded Process using the same method that you used to install the SAS 9.4 Embedded Process or the SAS Viya Embedded Process.

- To uninstall manually, see Uninstall Manually on page 127.
- To uninstall with Cloudera Manager, see Uninstall with Cloudera Manager on page 127.
- To uninstall with Ambari, see Uninstall with Ambari on page 127.

Uninstall Manually

To uninstall manually, run the following commands:

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -remove
```

Uninstall with Cloudera Manager

1 Log on to Cloudera Manager.

2 Stop the SAS EP service if it is running.

3 From the **Menu** bar, select **Hosts** ⇨ **Parcels**.

4 Select the SASEP parcel.

5 Deactivate the SASEP parcel.

6 Remove the SASEP parcel.

7 Delete the SASEP parcel.

8 When prompted, click **Close**.

   CAUTION! Do not restart the cluster.

9 Run the following command to remove the **/sas/ep** directory.

```
hadoop fs -rm -r -f /sas/ep
```

Uninstall with Ambari

Note: Root or passwordless sudo access is required in order to remove the stack.

1 On the CAS controller machine, navigate to the **/opt/sas/viya/home/share/ep/stack** directory.

2 Copy the entire stack directory to a temporary directory (/tmp) on your Hadoop Ambari Cluster Manager machine.

3 Navigate to the **/tmp/stack** directory and run the following command to delete the stack:

```
./delete_stack.sh Ambari-Admin-User-Name
```

4 Enter the Ambari administrator password at the prompt. A message appears that offers options for removal. Enter one of the options, as appropriate:

- Enter 1 to remove the SASEP config file only.
- Enter 2 to remove a specific version of the SASEP service.
- Enter 3 to remove all versions of the SASEP service.

5 You are prompted to restart the Ambari server in order to complete the removal of the SASEP service. To remove the SAS Embedded Process, you must restart the Ambari server.

6 Enter **y** to restart the Ambari server. The SASEP service is no longer listed on the Ambari dashboard user interface.

Appendix E: Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS

Supported Hadoop Distributions

Before you set up Hadoop, ensure that your Hadoop distribution is supported by SAS Viya. For more information, see Supported Releases of Hadoop Distributions on page 21.

Note: When an existing Hadoop cluster is shared between SAS 9.4 and SAS Viya, you must deploy the SAS Plug-ins for Hadoop that are delivered with SAS Viya. For details, refer to sections in this appendix. The SAS Viya HDAT Plug-ins are backward compatible with all SAS LASR Analytic Server versions. However, the SAS Plug-ins for Hadoop deployment from SAS LASR is not compatible with the CAS Server for Viya.

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 Scenario 2: CAS SASHDAT Access to HDFS on page 11.
- 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 12.

To configure your existing Hadoop cluster:

1. Perform the Hadoop pre-deployment tasks. Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters on page 129.
2. Configure your implementation of Hadoop:
   - For Apache, see Configure the Existing Apache Hadoop Cluster to Interoperate with the CAS Server on page 131.
   - For Cloudera, see Configure the Existing Cloudera Hadoop Cluster to Interoperate with the CAS Server on page 132.
   - For Hortonworks, see Configure the Existing Hortonworks Data Platform Hadoop Cluster to Interoperate with the CAS Server on page 137.
3. Verify CAS SASHDAT Access to HDFS. For more information, see Verify CAS SASHDAT Access to HDFS on page 139.

Pre-deployment Tasks for HDFS

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/home/SASFoundation/cas.settings** file and the **/opt/sas/viya/config/etc/cas/default/casconfig.lua** file.

- □ For more information about updating CAS environment variables, see *SAS Viya Administration: SAS Cloud Analytic Services*.

- Each machine in the cluster must be able to resolve the host name of all the other machines in the cluster.
- 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
```

- 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 Kerberos and Secure Shell (SSH), review the requirements and perform the appropriate tasks in Kerberos Requirements on page 130 and Review the Passwordless Secure Shell Requirements on page 130.

Kerberos Requirements

SAS Viya operates with Kerberos as follows:

- 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.
- 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 Java is set up appropriately.

- 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.
- If you are using the Oracle JRE or the IBM JRE, use the two JAR files in place of the existing local_policy.jar file and the US_export_policy.jar file. These files are located in your JRE location. This location is typically the **JAVA_HOME/jre/lib/security/** directory. These files can be obtained from the IBM or Oracle website.
- 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, the files do not need to be replaced.

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
```

Review the Passwordless Secure Shell Requirements

Here are the passswordless 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 on page 131.

- 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.

Configure Passwordless SSH to Use Kerberos

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
```

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.

> **TIP** Even though you can specify **host ***, use of the wildcard is not recommended because it would allow GSSAPI Authentication with any host name.

Configure the Existing Apache Hadoop Cluster to Interoperate with the CAS Server

1 Locate your SAS Viya installation directory on the CAS controller, and then locate the following files:

```
/opt/sas/viya/home/SASFoundation/hdatplugins/SAS_VERSION
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.cas.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.grid.provider.yarn.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.lasr.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sascasfd
/opt/sas/viya/home/SASFoundation/hdatplugins/sashdfsfd
/opt/sas/viya/home/SASFoundation/hdatplugins/start-namenode-cas-hadoop.sh
/opt/sas/viya/home/SASFoundation/hdatplugins/start-datanode-cas-hadoop.sh
```

2 If you cannot locate the **/opt/sas/viya/home/SASFoundation/hdatplugins** directory, then install the RPM package sas-hdatplugins-timestamp.x86_64.rpm.

```
sudo rpm -i /sas-hdatplugins-03.00.00-20160315.083831547133.x86_64.rpm
```

3 Change directories to the location of the full path that corresponds to **`$HADOOP_HOME`**.

Note: `$HADOOP_PREFIX` has been deprecated.

4 Create a new subdirectory **`sas`** under the **`$HADOOP_HOME/share/hadoop/`** directory.

5 Locate and propagate to the following three JAR files to the **`$HADOOP_HOME/share/hadoop/sas`** directory on each machine in the Apache Hadoop cluster:

```
sas.cas.hadoop.jar
sas.lasr.hadoop.jar
sas.grid.provider.yarn.jar
```

6 Locate the sashdfsfd file, the sascasfd file, the start-namenode-cas-hadoop.sh file, and the start-datanode-cas-hadoop.sh file and propagate them to the **`$HADOOP_HOME/bin`** directory on each machine in the Apache Hadoop cluster.

7 Locate the SAS_VERSION file and propagate it to the **`$HADOOP_HOME`** directory on each machine in the Apache Hadoop cluster.

8 On the machine to which you initially installed Apache Hadoop, add the following SAS properties for the HDFS configuration to the **`$HADOOP_HOME/etc/hadoop/hdfs-site.xml`** file:

Note: Adjust values appropriately for your deployment. The port numbers should be valid port numbers.

```
<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.service.allow.put</name>
<value>true</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>
<property>
<name> dfs.namenode.fs-limits.min-block-size</name>
<value>0</value>
</property>
```

Configure the Existing Cloudera Hadoop Cluster to Interoperate with the CAS Server

Overview of Deployment Methods

You can either deploy manually or deploy automatically by using the cluster manager for your Hadoop distribution:

- To deploy manually, see Deploy Manually on page 133.
- To deploy with Cloudera Manager, see Deploy with Cloudera Manager on page 135.

Deploy Manually

Use Cloudera Manager to configure your existing Cloudera Hadoop (CDH 5) deployment to interoperate with the CAS server.

1 Locate your SAS Viya installation directory on the CAS controller, and then locate the following files:

```
/opt/sas/viya/home/SASFoundation/hdatplugins/SAS_VERSION
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.cas.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.grid.provider.yarn.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.lasr.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sascasfd
/opt/sas/viya/home/SASFoundation/hdatplugins/sashdfsfd
/opt/sas/viya/home/SASFoundation/hdatplugins/start-namenode-cas-hadoop.sh
/opt/sas/viya/home/SASFoundation/hdatplugins/start-datanode-cas-hadoop.sh
```

2 If you cannot locate the **/opt/sas/viya/home/SASFoundation/hdatplugins** directory, then you must install the RPM package sas-hdatplugins-timestamp.x86_64.rpm.

```
sudo rpm -i /sas-hdatplugins-03.00.00-20160315.083831547133.x86_64.rpm
```

3 Locate the full path of the cluster. The default HADOOP_HOME location is the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop** directory.

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

4 Locate and propagate the following three JAR files to the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop/lib** directory on each machine in the cluster.

```
sas.cas.hadoop.jar
sas.lasr.hadoop.jar
sas.grid.provider.yarn.jar
```

5 Locate the sashdfsfd file, the sascasfd file, the start-namenode-cas-hadoop.sh file, and the start-datanode-cas-hadoop.sh file and propagate them to the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop/bin** directory of each machine in the CDH cluster. Here is an example:

```
/opt/cloudera/parcels/CDH-version/lib/hadoop/bin
```

6 Locate the SAS_VERSION file and propagate it to the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop/** directory on each machine in the CDH cluster.

7 Log on to Cloudera Manager as an administrator.

8 From Cloudera Manager Home, select the HDFS service. Within the HDFS service, select **Configuration** to edit the HDFS configuration properties.

Note: In the following steps, you must edit specific HDFS configuration properties. Locate the property to edit by specifying its name in the search bar.

a In the `dfs.namenode.plugins` property, add the following line to the plug-in configuration for the NameNode:

```
com.sas.cas.hadoop.NameNodeService
```

b In the `dfs.datanode.plugins` property, add the following line to the plug-in configuration for the DataNode:

```
com.sas.cas.hadoop.DataNodeService
```

9 Add the following lines to the advanced configuration for service-wide configuration. 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>com.sas.cas.service.allow.put</name>
<value>true</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>
```

10 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.service.allow.put</name>
<value>true</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>
```

Note: When Cloudera Manager prioritizes the HDFS client configuration, the client safety valve is used. When Cloudera Manager prioritizes anything else (such as YARN), the service safety valve is used. A best practice is to update the values of the safety valves. For more information, see the Cloudera documentation.

11 If you are not using the version of Java that is supplied with Cloudera, add the location of the JAVA_HOME variable. Navigate to the **Gateway Default Group**. Under **Advanced**, add the location of the JAVA_HOME variable to the **HDFS Client Environment Advanced Configuration Snippet for hadoop-env.sh (Safety Valve)** property. Here is an example:

```
JAVA_HOME=/usr/lib/java/jdk1.7.0_07
```

12 Save your changes and deploy the client configuration to each machine in the Hadoop cluster.

13 In Cloudera Manager, restart the HDFS service and any dependencies.

14 Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see Verify CAS SASHDAT Access to HDFS on page 139.

Deploy with Cloudera Manager

The following deployment steps assume that the hdatplugins rpm package has been installed directly on one of the following machines:

CAUTION! When the Cloudera Hadoop parcel is upgraded, the HDATPlugins parcel must be deactivated and then reactivated.

- the Cloudera Manager server
- on a machine in the network that is accessible to the Cloudera Manager server

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.

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 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.

  a Click **Distribute** to copy the parcel to all nodes.

  b Click **Activate**. You are prompted to restart the cluster or to close the window.

  c When prompted, click **Close**.

  CAUTION! Do not restart the cluster.

6 From Cloudera Manager Home, select the HDFS service. Within the HDFS service, select **Configuration** to edit the HDFS configuration properties.

  Note: In the following steps, you must edit specific HDFS configuration properties. Locate the property to edit by specifying its name in the search bar.

  a In the `dfs.namenode.plugins` property, add the following line to the plug-in configuration for the NameNode:

```
com.sas.cas.hadoop.NameNodeService
```

  b In the `dfs.datanode.plugins` property, add the following line to the plug-in configuration for the DataNode:

```
com.sas.cas.hadoop.DataNodeService
```

7 Add the following lines to the advanced configuration for service-wide configuration. 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>com.sas.cas.service.allow.put</name>
```

```
<value>true</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>
```

8 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.service.allow.put</name>
<value>true</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>
```

Note: When Cloudera Manager prioritizes the HDFS client configuration, the client safety valve is used. When Cloudera Manager prioritizes anything else (such as YARN), the service safety valve is used. A best practice is to update the values of the safety valves. For more information, see the Cloudera documentation.

9 Navigate to the **HDFS** tab. In the search field, enter **HDFS Service Environment Advanced Configuration Snippet (Safety Valve)**. Add the following property to the **HDFS (Service-wide)** field:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/cloudera/parcels/SASHDAT/*
```

10 If you are not using the version of Java that is supplied with Cloudera, add the location of the JAVA_HOME variable. Navigate to the **Gateway Default Group**. Under **Advanced**, add the location of the JAVA_HOME variable to the **HDFS Client Environment Advanced Configuration Snippet for hadoop-env.sh (Safety Valve)** property. Here is an example:

```
JAVA_HOME=/usr/lib/java/jdk1.7.0_07
```

11 Save your changes and deploy the client configuration to each machine in the Hadoop cluster.

12 In Cloudera Manager, restart the HDFS service and any dependencies.

13 Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see Verify CAS SASHDAT Access to HDFS on page 139.

Configure the Existing Hortonworks Data Platform Hadoop Cluster to Interoperate with the CAS Server

Overview of Deployment Methods

- To deploy manually, see Deploy Manually on page 137.
- To deploy with Ambari, see Deploy with Ambari on page 138.

Deploy Manually

Use the Ambari interface to configure your existing Hortonworks Data Platform (HDP) deployment to interoperate with the CAS server.

1 Locate your SAS Viya installation directory on the CAS controller, and then locate the following files:

```
/opt/sas/viya/home/SASFoundation/hdatplugins/SAS_VERSION
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.cas.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.grid.provider.yarn.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.lasr.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sascasfd
/opt/sas/viya/home/SASFoundation/hdatplugins/sashdfsfd
/opt/sas/viya/home/SASFoundation/hdatplugins/start-namenode-cas-hadoop.sh
/opt/sas/viya/home/SASFoundation/hdatplugins/start-datanode-cas-hadoop.sh
```

2 If you cannot locate the **/opt/sas/viya/home/SASFoundation/hdatplugins** directory, then you must install the RPM package sas-hdatplugins-timestamp.x86_64.rpm.

```
sudo rpm -i /sas-hdatplugins-03.00.00-20160315.083831547133.x86_64.rpm
```

3 Locate the full path of the HDP cluster. The default location is the **/usr/hdp/*version*/hadoop** directory.

4 Locate and propagate the following three JAR files to the **/usr/hdp/*version*/hadoop/lib** directory on each machine in the HDP Hadoop cluster:

```
sas.cas.hadoop.jar
sas.lasr.hadoop.jar
sas.grid.provider.yarn.jar
```

5 Locate the sashdfsfd file, the sascasfd file, the start-namenode-cas-hadoop.sh file, and the start-datanode-cas-hadoop.sh file and propagate them to the **bin** directory on each machine in the HDP cluster:

Note: All files in the **/bin** directory must be executable with file permissions of 755.

```
/usr/hdp/version/hadoop/bin
```

6 Locate the SAS_VERSION file and propagate it to the following directory on each machine in the HDP cluster:

```
/usr/hdp/version/hadoop/
```

7 In the Ambari interface, create a custom hdfs-site.xml file and add the following properties:

a Click **HDFS Service**.

b Choose **Config Section**.

c Click **Advanced**.

d 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.service.allow.put
`true`

com.sas.cas.hadoop.service.namenode.port
`15452`

com.sas.cas.hadoop.service.datanode.port
`15453`

dfs.namenode.fs-limits.min-block-size
`0`

8. Save the properties and restart all 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 Verify CAS SASHDAT Access to HDFS on page 139.

Deploy with Ambari

The following deployment steps assume that the hdatplugins rpm package is installed directly on one of the following machines:

CAUTION! When the Hortonworks Hadoop parcel is upgraded, the HDATPlugins parcel 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 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.

In the Ambari interface, create a custom hdfs-site.xml file and add the following properties:

a Click **HDFS Service**.

b Choose **Config Section**.

c Click **Advanced**.

d 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.service.allow.put
`true`

com.sas.cas.hadoop.service.namenode.port
`15452`

com.sas.cas.hadoop.service.datanode.port
`15453`

dfs.namenode.fs-limits.min-block-size
`0`

e Save the properties and restart all HDP services and MapReduce services.

f Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see Verify CAS SASHDAT Access to HDFS on page 139.

Verify 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.

```
hadoop fs -mkdir /test
hadoop fs -chmod 777 /test
```

2 To verify that the software has been successfully deployed, run the following SAS code:

```
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.
```

Uninstall the SASHDAT Plug-ins

Uninstall with Cloudera

1. From the Menu bar, select **Hosts** ⇨ **Parcels**.
2. Select the SASHDAT parcel.
3. Deactivate the SASHDAT parcel.
4. Remove the SASHDAT parcel.
5. Delete the SASHDAT parcel.
6. When prompted, click **Close**.

Uninstall with Ambari

Note: To remove the stack, root or passwordless sudo access is required.

1. 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
   ```

2. At the prompt, enter the Ambari administrator password. A message appears that offers options for removal.
3. Enter one of the following options:
   - Enter 1 to only remove the SASHDAT service.
   - Enter 2 to remove a specific version of the SASHDAT service.
   - Enter 3 to remove all versions of the SASHDAT service.

   To complete the removal of the SASHDAT service, you are prompted to restart the Ambari server.
4. Enter **y** to restart the Ambari server.

The SASHDAT service is no longer listed on the Ambari dashboard.

Appendix F: SAS In-Database Deployment: Configuring SAS Viya to Access Teradata

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.

Overview of the In-Database Deployment Package for Teradata

SAS In-Database Technologies Teradata for SAS Viya includes SAS Data Connect Accelerator for Teradata and the SAS Embedded Process for Teradata, as well as a security configuration file. This section describes how to install and configure the in-database deployment package for Teradata.

The SAS Embedded Process is a SAS server process that runs within Teradata to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that are installed on your Teradata system.

If you are using SAS Data Connect Accelerator for Teradata and you want to secure data transfer between your Teradata cluster and CAS, use the security configuration file.

Note: If you are adding additional nodes, the version of the SAS Embedded Process must be the same for the existing and new nodes.

Note: In addition to installing the in-database deployment package for Teradata, you must also install a set of SAS Embedded Process functions in the Teradata database. The functions package for the SAS Embedded Process is downloadable from Teradata.For more information, see Install the Support Functions for the SAS Embedded Process on page 144.

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

Teradata Installation and Configuration

To install and configure the SAS In-Database Technologies for Teradata:

1. Install the in-database deployment package. For more information, see Installing the SAS In-Database Deployment Package for Teradata on page 142.
2. Install the support functions for the SAS Embedded Process. For more information, see Install the Support Functions for the SAS Embedded Process on page 144.
3. (Optional) If you are using SAS Data Connect Accelerator, and you want to secure the data transfer between your Teradata or Hive cluster and CAS, you must enable security. For more information, see *SAS Viya Administration: Encryption*.

Installing the SAS In-Database Deployment Package for Teradata

Copy the SAS In-Database Deployment Packages for Teradata to the Server Machine

1. Locate the SAS in-database deployment package file, sepcoretera-11.50000-*n*.x86_64.rpm. *n* is a number that indicates the latest version of the file.
2. Navigate to the **/opt/sas/viya/home/share/ep** directory. This directory was created when you installed SAS Viya.
3. Locate the sepcoretera-11.50000-*n*.x86_64.rpm file. *n* is a number that indicates the latest version of the file.
4. Copy this file to a temporary directory on the Teradata machine. Make sure that you copy the file to the server machine according to the procedures that are used at your site. Here is an example of a secure copy command.

   ```
   scp sepcoretera-11.50000-n.x86_64.rpm  root@teramach1:/temporary-dir
   ```

   This package file is readable by the Teradata Parallel Upgrade Tool.

Install the SAS In-Database Deployment Package 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 knowledge about 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 included in the Teradata Online Publications site at 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.

Follow these steps to use the Teradata Parallel Upgrade Tool to install the SAS in-database deployment package.

Note: The Teradata Parallel Upgrade Tool prompts are subject to change as Teradata enhances its software.

1 Locate the in-database deployment packages on your server machine. The location must be accessible from at least one of the Teradata nodes. For more information, see Copy the SAS In-Database Deployment Packages for Teradata to the Server Machine on page 142.

2 Start the Teradata Parallel Upgrade Tool.

3 Be sure to select all Teradata TPA nodes for installation, including Hot Stand-By Nodes.

4 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.

5 If the installation is successful, sepcoretera-11.50000-*n*.x86_64 is displayed. *n* is a number that indicates the latest version of the file.

   Alternatively, you can manually verify that the installation is successful by running these commands from the shell prompt.

```
psh "rpm -q -a" | grep sepcoretera
```

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: http://*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 🏃 to run the new CAS session code.

3 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Teradata:

```
caslib teralib datasource=(srctype="teradata", dataTransferMode="serial" username="<user ID>",
password="<password>", server="<Teradata host name>", database="<Teradata database name>");

proc casutil;
  list files incaslib="teralib";
run;
```

4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connect Accelerator for Teradata:

```
caslib teraplib datasource=(srctype="teradata", dataTransferMode="parallel" username="<user ID>",
password="<password>", server="<Teradata host name>", database="<Teradata database name>");

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.

CAUTION! If you are using Teradata 15, you must drop the SAS_SYSFNLIB.SASEP_VERSION function to disable the Teradata Table Operator (SASTblOp). Otherwise, your output can contain missing rows or incorrect results. To drop the function, enter the following command:

```
drop function SAS_SYSFNLIB.SASEP_VERSION
```

This issue is fixed in theTeradata maintenance release 15.00.04.

Appendix G: Deployment Troubleshooting

Issue	Explanation	Resolution
SAS Viya services do not start.	In Consul is deployed, one cause might be that certain SAS Configuration Server (Consul) files are corrupted.	1 Stop all services. 2 Delete the /opt/sas/viya/config/data/consul/checks/. directory. 3 Restart all services.
After removing the software and attempting to reinstall the software: `Error: Nothing to do`	The directories containing the software were deleted. However, the yum remove command was never run. In the /var/log/yum.log, the last entry for the rpm is `Installed`.	Clean up the yum repository by running the following command: `yum remove` *`packagename`* You can then reinstall the software.

After running SAS code:
`ERROR: Procedure PCA not found`
`ERROR: Procedure KCLUS not found.`
	The installation was attempted on a system that was not completely cleaned up from a previous installation.	Uninstall SAS/CONNECT by running the following command:
		`yum groups mark remove "SAS/CONNECT"`
		Reinstall SAS/CONNECT by running the following command:
		`sudo yum groupinstall "SAS/CONNECT"`

When running the deployment:
`TimeoutError(error_message)\nTimeoutError:`
` Timer expired\n", "rc": 257}  13:15:37 |`
`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`
	The system failed to gather mount information.	Do one of the following:
		■ Set **`/etc/mtab`** as a link to **`/proc/mounts`** by running the following command:
		`sudo ln -s /proc/mounts /etc/mtab`
		■ Edit the ansible.cfg file and add or change the time-out value for Ansible as follows:
		`timeout=`*`number of seconds`*
		Deploy your software by running the Ansible playbook again.

Issue	Explanation	Resolution

When running the deployment:
`TASK [casserver-validate :`
` Verify that cas admin user is defined]`
` **************`
 	
`[WARNING]: when statements should not`
`include jinja2 templating delimiters such`
`as {{ }} or {% %}. Found: (casadminuser.rc`
`is defined and casadminuser.rc == 1) or`
`({{ cas_admin_user }} != {{ casenv_user }})`
 	
`fatal: [deployTarget]: FAILED! =>`
`{"failed": true, "msg": "The conditional`
`check '(casadminuser.rc is defined and`
`casadminuser.rc == 1) or ({{ cas_admin_user }}`
`!= {{ casenv_user }})' failed. The error`
`was: error while evaluating conditional`
`((casadminuser.rc is defined and`
`casadminuser.rc == 1) or ({{ cas_admin_user }}`
`!= {{ casenv_user }})): 'cas' is undefined"}`
	An unsupported version of Ansible is being used.	Run the following command:
		`ansible --version`
		If the version number that is returned is not 2.2.1, install Ansible using the instructions in Install Ansible on page 32.

When attempting to access SAS Viya software from Google Chrome, the following message is displayed: `Your connection is not private.`	If you have previously accessed a website using https, when you access the website again, Google Chrome automatically redirects to https.	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 If the machine is known to the browser, click **Query** to determine if 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.

Issue	Explanation	Resolution

When running the deployment:
`TASK [casserver-validate :`
` Verify that cas admin user is defined]`
` **************`
 	
`[WARNING]: when statements should not`
`include jinja2 templating delimiters such`
`as {{ }} or {% %}. Found: (casadminuser.rc`
`is defined and casadminuser.rc == 1) or`
`({{ cas_admin_user }} != {{ casenv_user }})`
 	
`fatal: [deployTarget]: FAILED! =>`
`{"failed": true, "msg": "The conditional`
`check '(casadminuser.rc is defined and`
`casadminuser.rc == 1) or ({{ cas_admin_user }}`
`!= {{ casenv_user }})' failed. The error`
`was: error while evaluating conditional`
`((casadminuser.rc is defined and`
`casadminuser.rc == 1) or ({{ cas_admin_user }}`
`!= {{ casenv_user }})): 'cas' is undefined"}`
	An unsupported version of Ansible is being used.	Run the following command:
		`ansible --version`
		If the version number that is returned is not 2.2.1, install Ansible using the instructions in Install Ansible on page 32.

When running the deployment:
`fatal: [deployTarget2]: FAILED! =>`
`{"changed": false, "failed": true, "msg":`
`"Get http://localhost:8500/v1/kv/config/`
`application/rabbitmq/username: dial tcp`
`[::1]:8500: getsockopt: connection`
`refused\n\`
`ERROR: Unable to read a key\nGet`
`http://localhost:8500/v1/kv/config/`
`application/rabbitmq/password: dial tcp`
`[::1]:8500: getsockopt:connection`
`refused\n\`
`ERROR: Unable to read a key\n"}`
	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 private IP addresses assigned.
	- no private IP address.
		To confirm the cause of the failure, check in 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 the following file, which is installed by the Ansible playbook: **`/etc/sysconfig/sas/sas-viya-consul-default`**
		Locate the following section of the file:
		`# Consul option: -bind`
		`# Holds the desired name of a network interface`
		`or IPv4 address.`
		`# Please do not edit this. Instead, set it via`
		`your Ansible run.`
		`export CONSUL_BIND_EXTERNAL=adapter-name`
		For *adapter-name*, supply the name of the adapter that Consul should use to locate the machine.

Issue	Explanation	Resolution
When attempting to access SAS Viya software from Google Chrome, the following message is displayed: `Your connection is not private.`	If you have previously accessed a website using https, when you access the website again, Google Chrome automatically redirects to https.	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 if 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.