<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chapter 7 / Uninstalling SAS Viya</strong></td>
<td>53</td>
</tr>
<tr>
<td>Prepare to Uninstall</td>
<td>53</td>
</tr>
<tr>
<td>Uninstall from a Single Machine</td>
<td>53</td>
</tr>
<tr>
<td>Uninstall from Multiple Machines</td>
<td>54</td>
</tr>
<tr>
<td><strong>Chapter 8 / Next Steps</strong></td>
<td>55</td>
</tr>
<tr>
<td>User Spreadsheet</td>
<td>55</td>
</tr>
<tr>
<td>Further Documentation</td>
<td>55</td>
</tr>
<tr>
<td><strong>Appendix 1 / Creating High Availability PostgreSQL Clusters</strong></td>
<td>57</td>
</tr>
<tr>
<td>Overview</td>
<td>57</td>
</tr>
<tr>
<td>HA PostgreSQL Topologies</td>
<td>57</td>
</tr>
<tr>
<td>Set Up a Horizontal Cluster</td>
<td>59</td>
</tr>
<tr>
<td>Set Up a Vertical Cluster</td>
<td>61</td>
</tr>
<tr>
<td>Set Up a Hybrid Cluster</td>
<td>62</td>
</tr>
<tr>
<td>Set Up Multiple Clusters</td>
<td>63</td>
</tr>
<tr>
<td>Deployment Logs</td>
<td>66</td>
</tr>
<tr>
<td>Verify the Deployment</td>
<td>67</td>
</tr>
<tr>
<td><strong>Appendix 2 / Creating and Using Mirror Repositories</strong></td>
<td>69</td>
</tr>
<tr>
<td>Overview</td>
<td>69</td>
</tr>
<tr>
<td>Requirements</td>
<td>69</td>
</tr>
<tr>
<td>Use Ansible to Create a Mirror Repository</td>
<td>71</td>
</tr>
<tr>
<td><strong>Appendix 3 / Troubleshooting</strong></td>
<td>75</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>75</td>
</tr>
</tbody>
</table>
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.

What’s New in SAS Deployment

SAS Repositories
To ensure that you deploy the latest software, SAS provides SAS Viya in repository packages that are maintained by SAS. Specifically, the software is packaged in the RPM Package Manager (RPM) format, which simplifies installation, uninstallation, and upgrade tasks. Each time you deploy or update your software, you automatically receive the latest RPM packages that are available.

Note: The RPM-based deployment model works with repositories that are native to your operating system. As a result, a SAS Software Depot is not required in your environment.
Industry Standard Tools

You deploy SAS Viya with tools that are designed for deploying and updating software on Linux operating systems.

- SAS Viya deployment takes advantage of yum, a software package manager for Linux operating systems. Yum commands are used for secure access to RPM packages and for deploying and updating software in your environment.

- Ansible is the preferred tool for deploying SAS Viya. Ansible provides ease and flexibility for deploying to multiple machines. SAS provides an Ansible playbook that is based on your software order, and that can be customized for your environment. When you run the playbook, Ansible automates a series of yum commands that deploy the software.

Note: The SAS Deployment Wizard and the SAS Deployment Manager that support SAS 9.4 are not used to install and configure SAS Viya.

One Deployment Guide

This guide includes all the information that is needed to deploy a working environment: system requirements, pre-installation tasks, installation instructions, and information about post-installation steps. In previous releases, this information was provided in separate documents.

What Gets Deployed

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

- SAS Visual Investigator

- SAS Cloud Analytic Services (CAS), which is the analytics and license server for SAS Viya. CAS Server Monitor is the web application that provides the graphical user interface to SAS Cloud Analytic Services.

- Elasticsearch, which provides search capabilities for SAS Visual Investigator, is used to generate data for visualizations.

Note: The software that you can deploy is based on your order.

Deployment Scenarios

Advice about the Scenarios

- Ansible is used to deploy software. Ansible is shown as installed on a separate machine, called the Ansible controller.

- 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.

If you purchased one or more data connectors, they 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.

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: Elasticsearch Cluster**

In this scenario, Elasticsearch is deployed across multiple nodes in a clustered environment. An advantage of this scenario is that optimal processing can be achieved through spreading the Elasticsearch queries across multiple nodes. During deployment, the Elasticsearch master node and client node are deployed on separate machines. Data nodes are also on their own machines.


**Scenario 2: Elasticsearch in High-Availability Mode**

In this scenario, Elasticsearch is deployed in high-availability mode with multiple master nodes and client nodes.

- Elasticsearch recommends at least three master nodes and four client nodes.
- The data nodes are deployed on individual machines, and replication of the data can ensure against failure of a single data node.
- 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.

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 for SAS Visual Investigator

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

SAS strongly recommends consulting with a sizing expert at SAS 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 host 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.

The following table contains minimum recommendations for a single-machine deployment:

**Requirements for Single-Machine Deployment**

<table>
<thead>
<tr>
<th>Item</th>
<th>Minimum Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Xeon CPU with 4 cores</td>
</tr>
<tr>
<td></td>
<td>x86 architecture with a minimum speed of 2.6 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>42 GB of RAM</td>
</tr>
<tr>
<td></td>
<td>Memory clock speed of 1600 MHz</td>
</tr>
<tr>
<td>Disk Space and Speed</td>
<td>2 x 300 GB</td>
</tr>
<tr>
<td></td>
<td>10,000 RPM</td>
</tr>
</tbody>
</table>

In a multi-machine deployment, follow similar minimum guidelines for each target machine.

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.

---

**Operating System Requirements**

**Supported Operating Systems**

For the full list of supported operating systems, see the following website:

**Linux Prerequisites**

SAS Viya deployment requires the operating system to be registered with the Red Hat Network. 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.
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 14.

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 operating-system package:

$ 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 Prerequisites for SAS Visual Investigator**

For SAS Visual Investigator, the following additional libraries are required:
- gettext-0.17-18
- jq-1.3-2
- unzip-6.0-5

**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](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.

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.

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 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
```

Data Source and Storage Requirements

Data Encoding Requirement

UTF-8 is the only SAS session encoding that is 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 Visual Investigator supports the following data sources:

- PostgreSQL
- Oracle 12c

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.

# Security Requirements

## User Account Requirements

Perform the following steps before you start the deployment:

- Verify that your user account has Administrator privileges for the Linux machine where you are launching the SAS software deployment.
- Verify that the user account that you are using for the deployment has super user (sudo) access. Run the following command to verify that the user ID is included in the sudoers file:
  
  ```bash
  sudo -v
  ```

  As an alternative, verify your sudoers privileges with the following command:
  
  ```bash
  sudo -l
  ```

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

- Create a user account named cas and an administrator account (named viadmin as an example in this document). Add them both to a group named sas.

- Verify that the cas account has a consistent UID and GID on all machines in your deployment.

  If you are deploying with custom user accounts, you might have to use the `usermod` command to modify the UIDs of any mismatched user accounts to make them consistent. For groups with mismatched GIDs, use the `groupmod` command.

- Verify that all users who launch CAS sessions have a consistent UID and GID on all machines in your deployment.

- Save the `id_rsa` key for the cas user in the `$HOME/.ssh` directory.

- Add the cas and viadmin users to an LDAP server to ensure that these credentials are shared throughout your deployment.

- Verify that the cas account is present on every node where a CAS component is running.

  Additional requirements for this user account are described in the table that follows.

The table identifies and describes the required 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 once they have been created. The accounts are not run as root. If you must log on to any of these accounts, use sudo to access them:
<table>
<thead>
<tr>
<th>Account Name and Group</th>
<th>Parameters</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas; member of sas group</td>
<td>UID: 1002, GID: 1001</td>
<td>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.</td>
</tr>
<tr>
<td>sas; member of sas group</td>
<td>UID: 1001, GID: 1001</td>
<td>Typical user account, subject to user restrictions. 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 local and LDAP accounts in your deployment, user credentials must match. 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 a clustered environment (with multiple CAS workers), passwordless SSH can be configured by the deployment process. Requires an RSA key in the $HOME/.ssh directory. The “cas” user name is strongly recommended. Assigning this user name enables the deployment to assign SSH keys. To assign a different user name, modify the casenv_user parameter in the vars.yml file.</td>
</tr>
<tr>
<td>viadmin</td>
<td>Administrative user account. In addition to Administrator privileges, it requires a valid LDAP user ID. You can assign another user name to this account, if desired. This is the user ID that you must enter as the ADMIN_ID in the script to set up TLS (the svi-visual-investigator.sh script) and in the perms.xml file. For more information, see “Configure SAS Visual Investigator” on page 41.</td>
<td>The first user account that signs on to SAS Visual Investigator after the deployment has completed.</td>
</tr>
</tbody>
</table>

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.
Authentication

SAS Visual Investigator on Linux supports LDAP for user authentication. Microsoft Active Directory and OpenLDAP are supported LDAP implementations. In addition, the CAS server uses OAuth tokens for all clients in your deployment.

Before SAS Visual Investigator is deployed, do the following:

- Set up tenants in LDAP. Use the documentation that is appropriate for your LDAP implementation.
- (Optional) Edit the sitedefault.yml file to enable the playbook to automatically configure the LDAP identity provider for OAuth to use. For more information, see “Create the sitedefault.yml File” on page 33.

The deployment process configures some LDAP settings automatically. After your SAS Viya deployment completes, the designated SAS Visual Investigator administrator must log on to the administration application as the viadmin user. The administrator is prompted to set up a more secure user account. Then the administrator can import valid LDAP user accounts from a spreadsheet to the SAS Visual Investigator environment. For more information, see “Next Steps” on page 55.

Client Requirements

Overview of 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 Visual Investigator or CAS Server Monitor

End users can access the SAS Visual Investigator user interface from a web browser running on a desktop machine. For information about supported web browsers and the corresponding platforms to access SAS Viya user interfaces, see https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-web-browsers.html.

Browsers on tablets and other mobile devices are not supported for displaying SAS Visual Investigator. However, you can access CAS Server Monitor from an Apple iPad.

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 17.

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 16.

The Ansible controller must be connected to the Red Hat Network.
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_Visual_Investigator_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 files that make up the SAS Visual Investigator 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.

<table>
<thead>
<tr>
<th>Process</th>
<th>Required Port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPD</td>
<td>80 (internal)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>443 (external)</td>
<td>See note below.</td>
</tr>
<tr>
<td>default Erlang Port Mapper Daemon (epmd) port</td>
<td>4369</td>
<td></td>
</tr>
<tr>
<td>SAS Infrastructure Data Server</td>
<td>5430–5439</td>
<td>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.</td>
</tr>
<tr>
<td>CAS Server Starting Port</td>
<td>5570</td>
<td>Used by clients to make binary connections to CAS.</td>
</tr>
<tr>
<td>CAS Communicator Port</td>
<td>5580</td>
<td></td>
</tr>
<tr>
<td>default SAS Messaging Broker AMQP client access port</td>
<td>5672</td>
<td></td>
</tr>
<tr>
<td>SAS Studio</td>
<td>7080 (if you are performing a visual-only or full deployment, the deployment will use ephemeral ports, so no port needs to be reserved)</td>
<td>Not required for SAS Visual Investigator.</td>
</tr>
<tr>
<td>SAS Configuration Server</td>
<td>8500</td>
<td>SAS uses HashiCorp Consul as its configuration server.</td>
</tr>
<tr>
<td>Object Spawner</td>
<td>8591</td>
<td></td>
</tr>
</tbody>
</table>
### Process | Required Port | Notes
---|---|---
CAS Server Monitor | 8777 | Used by clients to make REST HTTP calls to CAS, as with the Python REST interface.
Elasticsearch | 9200 | Only required 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. However, in order to secure web access to your SAS Viya software, only port 443 (HTTPS) should be open externally.

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:
   ```bash
   sudo sysctl net.ipv4.ip_local_port_range
   ```
   The results contain two numbers:
   ```bash
   net.ipv4.ip_local_port_range = inclusive-lower-limit inclusive-upper-limit
   ```

2. To list any existing reserved ports, run the following command:
   ```bash
   sudo sysctl net.ipv4.ip_local_reserved_ports
   ```
   Here is an example of the results:
   ```bash
   net.ipv4.ip_local_reserved_ports = 23, 25, 53
   ```
   If no ports are reserved, no ports are listed in the results:
   ```bash
   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:
   ```bash
   sudo sysctl -w net.ipv4.ip_local_reserved_ports="ports-or-port-range"
   ```
   Here is an example:
   ```bash
   sudo sysctl -w net.ipv4.ip_local_reserved_ports="5672,15672,25672,4369,16060-16069,9200"
   ```
   **Note:** The `sysctl` command numerically sorts the port numbers regardless of the order that you specify.

4. Add an entry to the `/etc/sysctl.conf` file to make your changes permanent. Here is an example:
   ```bash
   net.ipv4.ip_local_reserved_ports = 4369,5672,9200,15672,16060-16069,25672
   ```
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.

```plaintext
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:

   ```shell
   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.

   ```shell
   ansible --version
   ```

Test Your Ansible Installation

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

```shell
ansible all -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.

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 ulimits 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 ulimits.

   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. Also, in the `/etc/security/limits.conf` file on each machine running Elasticsearch, set nofiles to no lower than 65536 for the sas user.

   ```
   sas     -     nofile     65536
   ```

3. On each machine running Elasticsearch, run the following command:

   ```
   sudo sysctl -w vm.max_map_count=262144
   ```

4. 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.
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.

---

**Verify Wildcard DNS Aliasing**

If you use multi-tenancy with SAS Visual Investigator, your site must have a wildcard DNS alias. For example, if your domain is acme.com, you would need an alias for *.acme.com so that the tenant name can be prepended to the domain name for resolution. The wildcard DNS alias enables each tenant to have a unique URL without having to create a DNS entry for each tenant.

Work with your system administrator to ensure that the wildcard DNS alias has been implemented.
Installing SAS Viya with Ansible

Edit the Inventory File ............................................................... 21
  Note about Sharing the hosts and the host_local File ..................... 22
  Edit the host_local File .......................................................... 22
  Define the Machines in the Deployment .................................... 22
  Assign the Target Machines to Ansible Groups ............................ 23

Modify the vars.yml File ............................................................ 25
  Set the Deployment Label ......................................................... 25
  Set the Pre-deployment Validation Parameters .............................. 25
  Specify JRE (Optional) ............................................................. 27
  Set Up Passwordless SSH for CAS ............................................. 27
  Install Consul on CAS Hosts .................................................... 29
  Define the CAS User Group ...................................................... 29
  Set Up the CAS Admin User ...................................................... 29
  Install Consul on CAS Hosts .................................................... 30
  Define the CAS User Group ...................................................... 30
  Set Up the CAS Admin User ...................................................... 30
  Change the Tenant Name ........................................................ 31
  Add Data Source Information ................................................... 31
  Modify the LD_LIBRARY_PATH .................................................... 31
  Set the CAS Cache Directory ................................................... 32
  Set Up HDFS and Co-location ................................................... 32

Create the sitedefault.yml File .................................................... 33

Deploy the Software .................................................................... 35
  Commands ................................................................................. 35
  Run from a Directory Other than the Default ............................... 35
  Successful Playbook Execution ................................................... 36
  Retry a Failed Deployment ....................................................... 36

Install with SAS 9.4 Software ...................................................... 36

Deployment Logs ......................................................................... 36

**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, you should skip this section and go to “Define the Machines in the Deployment” on page 22.

The first line of the host_local file is a deployment target reference, defining 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 using the following format:

```
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, you should skip this section and go to “Assign the Target Machines to Ansible Groups” on page 23.

The first section in the hosts file declares a deployment target reference for each target machine. It also specifies the connection information that is needed by Ansible to connect to that machine. The following line is an example of the format of the deployment target reference. It can also be found 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 declarations:

<table>
<thead>
<tr>
<th>Component of the Deployment Target Declaration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployTarget</td>
<td>the alias that is used by Ansible to refer to the physical machine definition. Choose a meaningful alias such as ansible-controller.</td>
</tr>
<tr>
<td>ansible_ssh_host</td>
<td>the IP address of the remote machine.</td>
</tr>
<tr>
<td>ansible_ssh_user</td>
<td>the user ID that is used by Ansible to connect to each of the remote machines and to run the deployment.</td>
</tr>
<tr>
<td>ansible_ssh_private_key_file</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

```
deployTarget ansible_connection=local
```

The following example lists the deployment targets for a multiple machine deployment:

```
main ansible_ssh_host=host1.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/id_rsa
controller ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file= 
Worker 1: ansible_ssh_host=host3.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Worker 2: ansible_ssh_host=host4.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Sas-ELastic-Master-1: ansible_ssh_host=host5.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Sas-ELastic-Master-2: ansible_ssh_host=host6.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Sas-ELastic-Client-1: ansible_ssh_host=host7.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Sas-ELastic-Client-2: ansible_ssh_host=host8.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Sas-ELastic-Data-1: ansible_ssh_host=host9.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
Sas-ELastic-Data-2: ansible_ssh_host=host10.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa

Note that each machine is listed only once. That is, no machine should be given more than one alias.

**Assign the Target Machines to Ansible Groups**

The second section in the inventory file is used to assign deployment targets to each Ansible 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 changes to the inventory file.

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

```
[CoreServices]
main

[configuratn]
main

[consul]
main

[elasticsearch]
sas-elastic-master-1 ElasticSearch_HostType=master ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
sas-elastic-master-2 ElasticSearch_HostType=master ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
sas-elastic-client-1 ElasticSearch_HostType=client ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
sas-elastic-client-2 ElasticSearch_HostType=client ElasticSearch_HeapSize=8g ElasticSearch_QueueSize=10000
sas-elastic-data-1 ElasticSearch_HostType=data ElasticSearch_HeapSize=16g ElasticSearch_QueueSize=10000
sas-elastic-data-2 ElasticSearch_HostType=data ElasticSearch_HeapSize=16g ElasticSearch_QueueSize=10000

[httpproxy]
main

[pgpoolc]
main

[rabbitmq]
main

[sasdatasvrc]
main
```
Consider the following issues when editing the inventory file:

- It is strongly recommended 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.

- The Ansible group [consul] and all other groups that begin with "vipr*" must have the same target machine or machines.

- Note that the entry for [elasticsearch] contains information not included with the other Ansible groups. For more information, see the comment preceding [elasticsearch] in the inventory file.

- If you are using HDFS, [sas-casserver-primary] and [sas-casserver-worker] should be assigned to machines in the Hadoop cluster.

- You must specify the same machine for [pgpoolc] and [sasdatasvrc]. In addition, if the machine that you specify does not have an alias of deployTarget in the deployment target reference, you must open the sas_viya_playbook/vars.yml file and replace the instance of deployTarget under INVOCATION VARIABLES with the alias that you used in the deployment target reference:

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

After you have completed your edits, save and close the 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, see "Creating High Availability PostgreSQL Clusters" on page 57.

---

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

```
DEPLOYMENT_LABEL: "VA04april2017"
```

**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 11.
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 25.

3 Verify that a CAS controller host is defined.
   
   Note: For information about assigning software to machines, see “Assign the Target Machines to Ansible Groups” on page 23.

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 22.

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 29.

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 you have already created a different install user and install group, the playbook verifies that the install user is in the install group and that the user can log on. If any part of 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 requirements 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 14.

2 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 “Assign the Target Machines to Ansible Groups” on page 23.
3 For each machine, verify that the install user has the following limits:
   - nofile is set to 20480 or higher.
   - nproc is set to 65536 or higher.

   Note: For more information about setting limits, see “Set the ulimits” on page 18.

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 preinstalled 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:
   ```yaml
   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:
   ```yaml
   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 8.

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 27 for more information about the default process.
- Use your own passwordless SSH. See “Use Your Own Passwordless SSH” on page 28.
- 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 28.

Accept the Passwordless SSH Default

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

```yaml
sas_users:
  cas:
    group: sas
    password: ''
    setup_home: false
    shell:
    home:
```
The casenv_user variable must also be set to the CAS account ID.
If you accept the default, the deployment process occurs as follows:

1. SSH keys are set up for the CAS user account.
2. A set of keys is created for any other user that is defined in the sas_users field.
3. The private and public keys are copied to each host that the playbook runs against.
4. The ssh-keyscan utility is run from each host to every other host in the CAS cluster.
5. The user’s public key is added to the \texttt{~/.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 29.

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:
   ```yaml
   setup_sas_users: false
   ```
3. Save and close the vars.yml file.

**Create Customized Passwordless SSH**

To use the playbook to set up passwordless SSH, 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.

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

2. Edit the fields as follows:
   a. Ensure that the setup_sas_users variable is set to true.
   b. Create a list of user accounts and attributes under sas_users.

      Here are the attributes:

      - group – the group to which the user belongs. If the group does not exist, it is created when the playbook runs.
      - password – the encoded password for the user account. If you do not want to assign a password to the user account, use quotation marks ("") that indicate that no password is assigned.
Note: The comments in the vars.yml file explain how to create an encrypted password.

- setup_home – uses the value of true or false. Determines whether the shell and home values should be used by the deployment. To accept the default, use a value of false.
- shell – the location of the shell for the user account to use. It can be used only if setup_home is set to true.
- home – the location of the user directory to be created. It can be used only if setup_home is set to true.

As an option, to install any packages to be defined under extra_packages, set setup_packages to true.

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 "Edit the Inventory File" on page 21.

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

If you want a user other than the cas user to be the CAS Admin user, perform the following steps:
1. Open the vars.yml file.
2. Remove the number sign (#) from the #casenv_admin_user field.
3. In that same field, insert the name of a user that exists and that can log on:
   ```yml
   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 [SAS Viya Administration: Security](#).

### Install Consul on CAS Hosts

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. This variable defaults to true if not specified.

If you set the sas_consul_on_cas_hosts variable to false, and you assign the same machine to the [programming] Ansible group and either the [sas-casserver-primary] Ansible group or the [sas-casserver-worker] Ansible group, the requirements check fail.

**Note:** For more information about assigning packages to machines, see "Assign the Target Machines to Ansible Groups" on page 23.

### 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

If you want a user other than the cas user to be the CAS Admin user, perform the following steps:

1. Open the vars.yml file.
2. Remove the number sign (#) from the #casenv_admin_user field.
3. In that same field, insert the name of a user that exists and that can log on:
   ```yml
   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 the CAS Server Monitor.
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. Remove the number sign (#) from the beginning of the line that contains the casenv_tenant variable.
3. In the casenv_tenant field, insert the ID for the tenant:
   
   ```yaml
   casenv_tenant: tenant-ID
   ```
   Note: Tenant names are limited to lowercase letters and numbers only.
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 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:
   
   ```yaml
   1: ORACLE_HOME=Oracle-home-directory
   2: LD_LIBRARY_PATH=/opt/sas/viya/home/SASFoundation/sasexe:
      /opt/sas/viya/home/lib64:$ORACLE_HOME/lib:$LD_LIBRARY_
   ```
4. Save and close the vars.yml file.

Modify the LD_LIBRARY_PATH

Note: If you have performed the steps in “SAS Data Connector to Oracle” on page 31, you should skip this section.
Open the vars.yml file.

In the CAS Specific section, uncomment (remove the number sign (#) from) the line that contains CAS_SETTINGS.

Locate the following line:

#5: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:$JAVA_HOME/lib/amd64/server:$ODBCHOME/lib

Replace that line with the following:

1: LD_LIBRARY_PATH=/opt/sas/viya/home/SASFoundation/sasexe:/opt/sas/viya/home/lib64:$LD_LIBRARY_PATH

Save and close the vars.yml file.

Set the CAS Cache Directory

SAS Cloud Analytics Services (CAS) is the analytics server for SAS Viya. By default, the /tmp directory is used as the cache directory for temporarily memory mapping tables when the data exceeds the allowed resident memory size. However, if /tmp becomes full, new users are prevented from logging on to the machine.

The cache directory can be changed to one that has more space. If you decide to change the cache, be sure to select a directory that accounts for tables that are loaded from other data sources and tables created as outputs from CAS actions. The size required differs for each user, but can run from gigabytes to terabytes. You can also provide a list of directories to be used as cache. If you use a list, 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.

To change the CAS cache:

1. Open the vars.yml file.
2. In the CAS_CONFIGURATION section, remove the number sign (#) that precedes the CAS_DISK_CACHE variable.
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

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

   CAS_CONFIGURATION:
   env:
   #CAS_DISK_CACHE: /tmp
   #HADOOP_NAMENODE: 127.0.0.1
   #HADOOP_HOME: /opt/vendor
cfg:
If you include a machine in the Ansible 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. Remove the number sign that precedes the HADOOP_NAMENODE and HADOOP_HOME variables.
3. Revise the variables' values as follows:

```yaml
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    HADOOP_NAMENODE: primary-namenode-host-name
    HADOOP_HOME: location-of-your-Hadoop-home-directory
  cfg:
    #gcport: 5580
    #httpport: 8777
    #port: 5570
    colocation: 'hdfs'
    mode: 'mpp'
```

4. In addition, if you are not deploying HDFS in a co-located environment, change the colocation variable to 'none', including the single quotation marks:

```yaml
CAS_CONFIGURATION:
  env:
  cfg:
    colocation: 'none'
```

If you change the colocation variable to 'none', you do not need to change any values that are assigned to the HADOOP_NAMENODE and HADOOP_HOME variables.

5. Save and close the vars.yml file.

Note: For more information about assigning machines to Ansible groups, see “Assign the Target Machines to Ansible Groups” on page 23.

---

### Create the sitedefault.yml File

To configure LDAP/AD to be used by Oauth, you must create and modify the sitedefault.yml file located in the

1. Copy the sitedefault_sample.yml and paste a renamed version of it in the same directory:

   ```bash
   sudo cp sas_viya_playbook/roles/consul/files/sitedefault_sample.yml
   sas_viya_playbook/roles/consul/files/sitedefault.yml
   ```

2. Open the sitedefault.yml file:

   ```bash
   sudo vi sas_viya_playbook/roles/consul/files/sitedefault.yml
   ```

   It contains the following fields:
Note: Comments have been removed to improve readability.

```yaml
config:
  application:
    sas.identities.providers.ldap.connection:
      host:
      password:
      port:
      url:
      userDN:
    sas.identities.providers.ldap.group:
      baseDN:
    sas.identities.providers.ldap.user:
      baseDN:
    sas.identities:
      administrator:

3 Remove the double number signs (#) from the beginning of the searchFilter line under sas.identities.providers.ldap.user and remove the value that is assigned to that variable.

4 Specify site-specific default values, as appropriate:
   - The comments in the file provide guidance about the values to specify.
   - Enclose each value in single quotation marks.
   - Specify the highlighted lines exactly as shown.
   - Maintain the indentation as shown.

   # Site-Specific Defaults
   #
   config:
     application:
       sas.identities.providers.ldap.connection:
         host: 'your-LDAP-host'
         password: 'password-for-the-LDAP-host'
         port: 'your-LDAP-port'
         url: 'ldap://your-LDAP-host:your-LDAP-port'
         userDN: 'CN=your-user-CN,OU=your-user-OU,DC=your-user-DC'
       sas.identities.providers.ldap.group:
         baseDN: 'OU=Groups,DC=your-DC'
         searchFilter: '(&(member={0}))'
       sas.identities.providers.ldap.user:
         baseDN: 'DC=your-DC'
         searchFilter: 'sAMAccountName={0}'
         customFilter: 'optional-custom-filter'
       sas.identities:
         administrator: 'administrator-user-ID'
     sas.identities.providers.ldap.profile:
       file: 'ldap/ldap-search-and-bind.xml'

5 Save and close the sitedefault.yml file.
```
# Deploy the Software

## 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:

### For a multi-machine deployment:

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td><code>ansible-playbook -i hosts site.yml</code></td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td><code>ansible-playbook -i hosts site.yml --ask-become-pass</code></td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td><code>ansible-playbook -i hosts site.yml --ask-pass</code></td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td><code>ansible-playbook -i hosts site.yml --ask-pass --ask-become-pass</code></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require a sudo password</td>
<td><code>ansible-playbook -i host_local site.yml</code></td>
</tr>
<tr>
<td>Requires a sudo password</td>
<td><code>ansible-playbook -i host_local site.yml --ask-become-pass</code></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td><code>ansible-playbook -i host_local site.yml</code></td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td><code>ansible-playbook -i host_local site.yml --ask-become-pass</code></td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td><code>ansible-playbook -i host_local site.yml --ask-pass</code></td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td><code>ansible-playbook -i host_local site.yml --ask-pass --ask-become-pass</code></td>
</tr>
</tbody>
</table>

## 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, indicating zero failures.

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 and options.

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:

- CAS_CONFIGURATION
- STUDIO_CONFIGURATION
- SPAWNER_CONFIGURATION

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. Here is the earlier example revised in this way:

```
SPAWNER_CONFIGURATION:
  sasPort: 8592
```

3. Save and close the vars.yml file.

4. Deploy your software by running the Ansible playbook as you did initially.

Note: If you change the port value for the object spawner, after installing your software, you must change the value of webdms.workspaceServer.port in the /opt/sas/viya/config/etc/sasstudio/default/init_usermods.properties file to match the port number that you specified in the vars.yml file.

Deployment Logs

Logs for Ansible deployments are stored in sas_viya_playbook/deployment.log.
To view the logs from the yum installation commands that are used in your deployment, run the following commands:

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

```bash
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

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

To activate the changed password, a restart services is required. The restart will occur at a later point in the manual configuration steps.

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 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 Oracle” on page 31.

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

2. Save and close the cas.settings file.
Configure the SAS Data Connector to PostgreSQL

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

Provide connection specifics in one of the following ways:

- reference a Data Source Name (DSN)

  Create an odbc.ini file. Here is an example of an odbc.ini file that supports DSN:

  [postgresql_data_source_name]
  Driver=/opt/sas/viya/home/lib64/psqlodbcw.so
  ServerName=localhost or hostname or ip>
  username=user name
  password=password
  database=database
  port=5432

- in your code

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

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

Note: During installation, you should have set the ODBCINI environment variable.

Configure SAS Visual Investigator

Configure CAS For SAS Visual Investigator

1. Stop the CAS controller by running the following command as root:

   sudo service sas-viya-cascontroller-default stop

2. Identify the user ID to be used as the SAS Visual Investigator administrator user ID. The user ID must be:
   - a valid LDAP or Active Directory user ID.
   - for the user that signs on to SAS Visual Investigator first.
   - the user ID that you will enter as the ADMIN_ID in the /opt/sas/viya/config/share/svi-visual-investigator/svi-visual-investigator.sh script.

   In the following steps, the SAS Visual Investigator administrator user ID is viadmin.

3. On the CAS controller, edit the /opt/sas/viya/config/etc/cas/default/casconfig.lua file as root.

   a. Locate the line that contains cas.provlist, and change the line as follows:

      cas.provlist = 'oauth'
b Locate the line that contains `--- default CAS super user used in perms.xml` and change the line as follows:

```plaintext
env.ADMIN_USER = 'user-from-the-vars.yml-file-deault-is-cas'
```

```plaintext
env.ADMIN_USER = VI-Admin-User
```

4 Save and close the file.

5 Enter the following commands from the shell:

```bash
cd /opt/sas/viya/config/etc/cas/default
sudo mv permstore permstore.0
```

6 On the CAS controller, open and edit the `/opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig` file as root by running the following command:

```bash
sudo vi /opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig
```

   a Locate the line that contains `useHostToken` and comment it out (add a number sign (#) to the beginning of the line), as follows:

```plaintext
# useHostToken
```

   b Locate and comment out the line that contains `externalIdent` as follows:

```plaintext
# externalIdent
```

c Save and close the file.

7 Restart the CAS controller by running the following command as root.

```bash
sudo service sas-viya-cascontroller-default start
```

---

**Configure SAS Visual Investigator**

1 On the machine where the host target [viprVi] is installed, locate the `svi-visual-investigator.sh` file in the `/opt/sas/viya/config/share/svi-visual-investigator` directory.

2 Open and edit the `svi-visual-investigator.sh` script as root by running the following command:

```bash
sudo vi /opt/sas/viya/config/share/svi-visual-investigator/svi-visual-investigator.sh
```

3 Locate the line that contains `TLSDOMAIN`. Enter the TLS domain. If you do not have a TLS domain, leave the value blank.

```plaintext
# Enter the TLS (https) domain in the TLSDOMAIN variable
# Example: acme.ssl.com
TLSDOMAIN=TLS-Domain
```

4 Locate the line that contains `INTERNALDOMAIN`. Enter the standard HTTP address for your deployment. The internal domain is where the internal communications between the components occur.

```plaintext
# Enter the Internal (http) domain in the INTERNALDOMAIN variable
# Example: acme.services.com
INTERNALDOMAIN=http-domain
```

5 Locate the lines that contain `ADMIN_TOKEN`. Enter a string to be used by all applications for interprocess communications:

```plaintext
# Enter the token string that will be used to obtain an access token from SASLogon for client registration
ADMIN_TOKEN=admin-token-string
```
Note: Generate a hash for the ADMIN_TOKEN value using this command: `echo -n Welcome | md5sum`. Before you enter the hash for ADMIN_TOKEN, remove the trailing `-.

6 Locate the lines that contain INITIAL_USER_PASSWORD. Enter the initial user password.

   INITIAL_USER_PASSWORD=initial-user-password
   Note: This is not an LDAP user. This is only used by the onboarding scripts not for interacting with the user interface.

7 Locate the line that contains DATAHUB_METADATA_USER_PW. Enter the password:

   # Enter the password that will be used for the datahub metadata database
   # This will be associated with the fdhadmin postgres internal user
   DATAHUB_METADATA_USER_PW=password
   Note: Generate the hash for DATAHUB_METADATA_USER_PW using this command: `echo -n Welcome | md5sum`.
   Before you enter the hash for DATAHUB_METADATA_USER_PW, remove the trailing `-.

8 Locate the line that contains POSTGRESQL_MAX_CONNECTION. Enter the maximum number of connections:

   - For deployments that are not high-availability, use 1000 for the number of maximum connections.
   - For high-availability deployments, use 3000 for the number of maximum connections.

   # Enter the maximum number of postgres connections
   # This will be used to configure kernel parameters to ensure the best functionality of postgres
   POSTGRESQL_MAX_CONNECTIONS=PostgreSQL-maximum-connections

9 Locate the line that contains ELASTICSEARCH_CLIENT_HOST. Enter the Elasticsearch host name or IP address:

   # Enter the Elasticsearch client node hostname or IP address and elastic search port. This is used if elasticsearch's client node is placed on a different server than the main consul server
   ELASTICSEARCH_CLIENT_HOST=client-node-hostname-or-IP-address

10 Save and close the file.

11 On the [viprVi] machine, run the following script:

   `sudo /opt/sas/viya/config/share/svi-visual-investigator/svi-visual-investigator.sh`

   The script loads some of the required key-value pairs for the SAS Visual Investigator application.

**Configure the Data Hub**

1 On the machine where the host target [viprCommon] was installed, verify the location of the datahub_configdb.sh file in the `/opt/sas/viya/config/share/svi-datahub` directory.

2 On the viprCommon machine, run the following script:

   `sudo /opt/sas/viya/config/share/svi-datahub/datahub_configdb.sh`
Run the Script to Change Consul

Run the following command:

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --force -- "config/audit/sas.audit/archive/schedule/cron" "0 0 0 1 * *"
```

Add the Key-Value Pairs to Consul

To add the key-value pairs to Consul, run each of the following commands sequentially.

Note: For each command, be sure to insert a space between -- and config. Here is an example:

```
-- config
```

However, do not insert a space here: --force.,

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --force -- config/geodeserver/sas.cache.config.distributedCache true
```

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --force -- config/geodeserver/sas.cache.default.mode local
```

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --force -- config/geodeserver/sas.verify.resource.database false
```

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --force -- config/geodelocator/sas.verify.resource.database false
```

Restart the Applications

On the [viprVi] machine, to restart the applications, run the following script:

```
sudo bash /opt/sas/viya/config/share/svi-visual-investigator/restart-vi-apps.sh
```

Note: The restart-vi-apps.sh script does not restart all services. To restart all services in the SAS Viya environment, you would use the sas-viya-all-services script instead.

When the script completes, you should have a SAS Visual Investigator installation that is ready for tenant onboarding.

Connect to PostgreSQL

To connect to the PostgreSQL instance:

1. Obtain the PostgreSQL password in one of the following ways:
   - If you have changed the default password, use your new password for PostgreSQL.
   - To obtain the current PostgreSQL password, run the following command:
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token

kv read "config/application/postgres/password"

The current PostgreSQL password is returned.

2 To connect to the PostgreSQL server, run each of the following commands sequentially.

```bash
export PGPASSWORD="postgresql-password"
/opt/sas/viya/home/bin/psql -t -h localhost -p 5432 -U dbmsowner postgres
```

3 To test whether the SAS Visual Investigator database is available for you to run queries on, run the following command:

```sql
SELECT count(datname) FROM pg_database WHERE datistemplate=false AND datname='SharedServices';
```

If results are returned, the database is available.

---

**Configure and Onboard Tenants for SAS Visual Investigator**

**Set up LDAP to Onboard a Tenant**

SAS Visual Investigator 10.2 releases are multi-tenant deployments. All tenants must be contained within the same LDAP server. SAS Viya does not yet support the concept of separate servers for one or more tenants.

When you set up the LDAP server, follow these guidelines:

- Create a tenant named **provider** on the LDAP server.
- Create the group and user definitions under a single tenant branch (or OU).
- Ensure that the baseDN properties that you specify for groups and for users are identical.
- Structure the LDAP environment according to the requirements of the SASLogon and Identities services.

Here is an example of the resulting LDAP environment:

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

**Onboard a Tenant**

To onboard a tenant:

1 Change to the following directory:

```bash
cd /opt/sas/viya/home/share/svi-datahub
```
Extract the archive called tm.tgz and make sure that you have appropriate permissions by running the following commands:

```bash
sudo unzip tm-10.2.1.5.zip
sudo chown -R sas:sas *
```

As root, edit the bareos-tenant.sh file. Modify these values to reflect the values for your cluster.

```bash
# SAS default installation locations, change if needed
SASHome=/opt/sas/viya/home
SASConfig=/opt/sas/viya/config
InstanceId=default

ConfigurableItems()
{
  # This is a template file for tenant on-boarding.
  # Copy this file to a new filename and change the properties below.
  
  # The following values are to be filled about by the administrator performing the onboarding task
  #
  # Enter the tenant hostname and the non-TLS domain of the tenant host:
  local -x svi_HOST="tenant-host"
  # Note: This should be the short host name of the server not the fully qualified name.
  
  # This is the internal non-TLS domain that resolves directly to the Cloud Foundry HAProxy.
  local -x svi_CF_NONTLS_DOMAIN="non-TLS-domain"
  # Note: This should be the domain only not the fully qualified name
  
  # This is the first user added for the new tenant.
  # The tenant first user is a username that can be authenticated using the configured LDAP type.
  # This username may be different for each deployed tenant.
  local -x svi_TENANT_USERNAME="tenant-username"
  # This is an LDAP user ID
  
  # Enter the credentials for the initial onboarding account
  local -x svi_APP_LOGON_INITIAL_USERNAME="sasboot"
  # This is not an LDAP user
  local -x svi_APP_LOGON_INITIAL_PASSWORD="password from svi-visual-investigator.sh for INITIAL_USER_PASSWORD"

  # Enter the credentials for the tenant admin account
  local -x svi_APP_LOGON_TENANT_ADMIN_USERNAME="sas.tenant_admin"
  # tenant admin user the first user to login for the tenant
  local -x svi_APP_LOGON_TENANT_ADMIN_PASSWORD="tenant-admin-password"

  # Enter the IP address where consul is running
  local -x svi_SERVICE_CONSUL_IP="service-consul-IP"
  # Enter the port for RabbitMQ
  local -x svi_SERVICE_RABBITMQ_INTERNAL_PORT="service_rabbitMQ-port"
  # typically port 15672
}
```

Run the bareos-tenant.sh script as follows:

```bash
sudo bash bareos-tenant.sh -i
```

To onboard a tenant, run the following command:

```bash
sudo bash bareos-tenant.sh your-tenant-name
```
Onboard Subsequent Tenants

Before adding a subsequent tenant, you must re-initialize the script for the next tenant:

1. Remove the previous tenant initialization as follows:
   
   ```bash
   cd /opt/sas/viya/home/share/svi-datahub
   sudo rm .initialized
   ```

2. Edit the bareos-tenant.sh file in the `/opt/sas/viya/home/share/svi-datahub` directory.

3. Locate the line that contains `svi_TENANT_USERNAME`. Change the line as follows:
   
   ```bash
   svi_TENANT_USERNAME=next_tenant_username
   ```

4. Save the bareos-tenant.sh file.

5. Run the following commands:
   
   ```bash
   sudo bash bareos-tenant.sh -i
   sudo bash bareos-tenant.sh next-tenant-username
   ```
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:
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 the Password for the CAS Administrator or Another Administrative Account” on page 39.

- 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 into 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 39.

---

**Verify RabbitMQ**

To verify that RabbitMQ has been deployed correctly, open a browser and go to the following address:

http://RabbitMQ-IP-address:15672/#/

If the RabbitMQ logon window appears, then RabbitMQ is functioning as expected.

---

**Verify PostgreSQL**

Note: This section is applicable only if your order contains PostgreSQL. If it does not, skip this section.

1 Run the following command:

```
/opt/sas/viya/home/bin/sas-bootstrap-config kv read "config/application/postgres/password"
```

2 Note the output of the command. It is the password for the dbmsowner.

3 Connect to the database:

```
/opt/sas/viya/home/bin/psql -h IP-address-for-PostgreSQL-database -U dbmsowner postgres
```

4 When prompted, enter the password that you noted in step 2:
Password for user dbmsowner:

5 If PostgreSQL is deployed appropriately, you should receive a response like this:

```
psql (9.4.9)
Type *help* for help
postgres=#
```

6 To exit the prompt, type `\q` and press Enter.

---

**Validate Elasticsearch**

To determine the health of the deployed Elasticsearch, use the following command:

```
curl -XGET 'http://IP-address-for-Elasticsearch-master-node:9200/_cluster/health?pretty=true'
```

Typical output follows:

```
{
  "cluster_name" : "testcluster",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 2,
  "number_of_data_nodes" : 3,
...
}
```

If the value of status is **green**, the cluster is fully functional. For additional information about Elasticsearch cluster health, refer to [https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html).

---

**Validate SAS Visual Investigator**

1 Go to the SAS Visual Investigator URL on the machine for the `http_proxy` target. For example, if the `http_proxy` target machine is test.acme.com, then you would go to `http://test.acme.com/SASVisualInvestigator`.

2 Log on as the user that you identified in step 1.

3 When you are prompted for Assumable Groups, click Yes.
Uninstalling SAS Viya

Prepare to Uninstall

   
   ```sh
   sudo vi /etc/rc.d/init.d/sas-viya-audit-default
   ```

2. Find the line that reads as follows:
   
   ```sh
   # chkconfig: -97 03
   ```

3. Change that line to read as follows. Note the addition of a space before “97”.
   
   ```sh
   # chkconfig: - 97 03
   ```


Uninstall from a Single Machine

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

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

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

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Additional Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password for sudo only</td>
<td><code>--ask-become-pass</code></td>
</tr>
<tr>
<td>Password for SSH only (applies only if the Ansible controller is on a different machine than your SAS software)</td>
<td><code>--ask-pass</code></td>
</tr>
<tr>
<td>Password for both sudo and SSH (applies only if the Ansible controller is on a different machine than your SAS software)</td>
<td><code>--ask-become-pass</code> <code>--ask-pass</code></td>
</tr>
</tbody>
</table>
When the appropriate command is executed, Ansible performs a group uninstallation, which removes your SAS Viya software, including both certificates. It also renames the `/opt/sas/viya` directory to `/opt/sas/viya_<epoch>`, where `<epoch>` specifies the UNIX epoch (the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970). 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 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`

To uninstall the software from the SAS Viya machine only, run the following command:

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

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

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Additional Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password for sudo only</td>
<td><code>--ask-become-pass</code></td>
</tr>
<tr>
<td>Password for SSH only</td>
<td><code>--ask-pass</code></td>
</tr>
<tr>
<td>Password for both sudo and SSH</td>
<td><code>--ask-become-pass --ask-pass</code></td>
</tr>
</tbody>
</table>

To uninstall individual CAS workers, first stop the CAS controller or remove the worker from the cluster via the CAS Server Monitor. Then uninstall the worker host and restart the CAS controller, if it was stopped.

Repeat this step to uninstall each CAS worker.

For more information about options that Ansible offers when working with specific hosts, see the Ansible documentation.
Next Steps

User Spreadsheet

1. Locate the svi-user-management.xls spreadsheet in the /opt/sas/viya/home/share/svi-visual-investigator directory.
2. Copy the svi-user-management.xls spreadsheet to a location where it can be opened in Microsoft Excel.
3. Provide the location of the svi-user-management.xls spreadsheet and the initial SAS Visual Investigator user ID (viadmin) and password to the designated SAS Visual Investigator administrator.
   a. The administrator logs on to the administration application and sets a new password for the account.
   b. The administrator uses the svi-user-management.xls spreadsheet to import users, as explained in the topic about managing groups and users in the SAS Visual Investigator 10.2.1: Administrator’s Guide.

Note: All users must be valid LDAP users.

Further Documentation

You can access SAS Visual Investigator 10.2.1: Administrator’s Guide from within the SAS Visual Investigator application or from the SAS Visual Investigator documentation page. To access the secure SAS Visual Investigator 10.2.1 documentation, you must have an access key. The documentation page explains how to contact SAS Technical Support to request the access key.
Creating High Availability PostgreSQL Clusters

Overview
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 21, 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. In the event that the primary data node is lost, PGPool will automatically promote a standby node to primary and reestablish replication from the new primary to the remaining standby data nodes.

The PostgreSQL deployment for Viya also supports the ability to deploy multiple database clusters as part of a single deployment. For example, you might want to put your microservices on one cluster while having dedicated clusters for your server. Each cluster is considered a service and each member of that cluster (PGPool and data nodes) is considered a node within that service. A cluster can be deployed on the same machines as other clusters or on their own machines.

A cluster can be deployed in four possible configurations:

- **Single Node** - One PGPool and one data node on the same machine. This is the default deployment for SAS Viya.
- **Horizontal** - Each data node on a separate machine.
- **Vertical** - All data nodes on a single machine.
- **Hybrid** - A combination of horizontal and vertical where there are at least two machines within the cluster and there is more than one data node on a machine within the cluster.

For 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.

<table>
<thead>
<tr>
<th>Cluster Configuration</th>
<th>Server</th>
<th>Port</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>Server 1</td>
<td>5432</td>
<td>Primary</td>
</tr>
<tr>
<td></td>
<td>Server 2</td>
<td>5432</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 3</td>
<td>5432</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 4</td>
<td>5432</td>
<td>Standby</td>
</tr>
<tr>
<td>Vertical</td>
<td>Server 1</td>
<td>5532</td>
<td>Primary</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5533</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5534</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5535</td>
<td>Standby</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Server 1</td>
<td>5632</td>
<td>Primary</td>
</tr>
<tr>
<td></td>
<td>Server 1</td>
<td>5633</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 2</td>
<td>5632</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td>Server 2</td>
<td>5633</td>
<td>Standby</td>
</tr>
</tbody>
</table>

The two files in your playbook that must be revised for HA PostgreSQL are the inventory.ini and vars.yml files. The inventory.ini file (the inventory) identifies roles that will be placed on each machine. The vars.yml file specifies the settings for pgpoolc and sasdatasvrc that are used to define the HA PostgreSQL instance or instances desired on each of those machines. Because the definitions for HA PostgreSQL come from synchronized edits of inventory.ini and vars.yml, those edits should be done in tandem to ensure alignment.
**Set Up a Horizontal Cluster**

**Edit the inventory.ini File**

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Define the Machines in the Deployment” on page 22. Then assign the machines to the host groups as described at “Assign the Target Machines to Ansible Groups” on page 23. Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

This is an example of a completed inventory.ini file that includes the horizontal cluster described in the table above, with PGPool being on the same machine as the first HA PostgreSQL node. (The example shows only the entries related to HA PostgreSQL):

```ini
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
'''
[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
```
Edit the vars.yml File

Open the vars.yml file in the playbook. In the INVOCATION_VARIABLES section, fill in the variables appropriate for your deployment. Using the horizontal cluster example from the table above, this section would describe four machines, one of which would have a subsection for pgpoolc and all having subsections for sasdatasvrc. This is what that section would look like when filled out for our example:

```
INVOCATION_VARIABLES:
  deployTarget1:
    pgpoolc:
      - PCP_PORT: '5431'
        PGPPOOL_PORT: '5430'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
    sasdatasvrc:
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5432'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
  deployTarget2:
    sasdatasvrc:
      - NODE_NUMBER: '1'
        NODE_TYPE: S
        PG_PORT: '5432'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
  deployTarget3:
    sasdatasvrc:
      - NODE_NUMBER: '2'
        NODE_TYPE: S
        PG_PORT: '5432'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
  deployTarget4:
    sasdatasvrc:
      - NODE_NUMBER: '3'
        NODE_TYPE: S
        PG_PORT: '5432'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
```

Note that the machine listed under [pgpoolc] in the inventory.ini file is the only one that has pgpoolc variables in the vars.yml file. Because all four machines will have HA PostgreSQL nodes on them, all four machines have sasdatasvrc variables in the vars.yml file. The nodes are numbered from 0 to 3, and node 0, on the deployTarget1 machine, is the primary node. The entry for SANMOUNT: will read the deployment and use the location of the SAS_CONFIG_ROOT directory and append the directory name.

After you save the vars.yml file and you complete the other deployment steps, use the commands described at "Deploy the Software" on page 35 to deploy your SAS Viya software, including HA PostgreSQL.
Set Up a Vertical Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Define the Machines in the Deployment” on page 22. Then assign the machines to the host groups as described at “Assign the Target Machines to Ansible Groups” on page 23. Make sure that the machine that you want to use for PGPoll is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

This is an example of a completed inventory.ini file that includes the vertical cluster described in the table above, with PGPoll being on the same machine as the HA PostgreSQL nodes. (The example shows only the entries related to HA PostgreSQL):

```yaml
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
'''
[sasdatasvrc]
deployTarget1
```

Edit the vars.yml File

Open the vars.yml file in the playbook. In the INVOCATION_VARIABLES section, fill in the variables appropriate for your deployment. Using the vertical cluster example from the table above, this section would describe a single machine, with a subsection for pgpoolc and four subsections for the sasdatasvrc nodes. This is what that section would look like when filled out for our example:

```yaml
# Multiple invocation definitions
INVOCATION_VARIABLES:
deployTarget1:
  pgpoolc:
    - PCP_PORT: '5531'
      PGPOOL_PORT: '5530'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
  sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5532'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5533'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5534'
```
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres
- NODE_NUMBER: '3'
NODE_TYPE: S
PG_PORT: '5535'
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres

Note that the machine is described with a single pgpoolc entry and four sasdatasvrc entries. The nodes are numbered from 0 to 3, and node 0 is the primary node. The PORT entries all show a different port in order to avoid any conflict. The entry for SANMOUNT will read the deployment and use the location of the SAS_CONFIG_ROOT directory and append the directory name.

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 35 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Hybrid Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Define the Machines in the Deployment” on page 22. Then assign the machines to the host groups as described at “Assign the Target Machines to Ansible Groups” on page 23. Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

This is an example of a completed inventory.ini file that includes the hybrid cluster described in the table above, with PGPool being on the same machine as two of the HA PostgreSQL nodes. (The example shows only the entries related to HA PostgreSQL):

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
...
[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 35 to deploy your SAS Viya software, including HA PostgreSQL.

---

Set Up Multiple Clusters

Modify inventory.ini and vars.yml Files

This example consists of four machines and has the following clusters:

- a single-node cluster with pgpoolc and sasdataservc on a machine named deployTarget1
- a horizontal cluster with pgpoolc on deployTarget1 and a sasdatasrvc node on each machine
- a vertical cluster with pgpoolc on deployTarget3 and all the sasdatasrvc nodes on deployTarget4
- a hybrid cluster with pgpoolc on deployTarget1, two sasdatasrvc nodes on deployTarget2, and two more sasdatasrvc nodes on deployTarget3

This is how the inventory.ini file should be modified for this HA PostgreSQL deployment (the entries related to HA PostgreSQL are shown):

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=~/./ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=~/./ssh/id_rsa
```
deploytarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
deploytarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
...
[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: '5432'
      SANMOUNT: '{% SAS_CONFIG_ROOT %}/data/sasdatasvrc'
      SERVICE_NAME: postgres_hybrid
    - 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 indentation level as application, add the following content:

```yaml
config:
  application:
    ...
  service-name
  sas:
    database:
      databaseServerName: cluster-name
      spring.datasource.password: ${sas.database.cluster-name.password}
```

The value for cluster-name must exactly match the SERVICE_NAME value for the cluster in the INVOCATION_VARIABLES section in the vars.yml file.

The following example shows the addition of the authorization service that uses an HA Postgres cluster named postgres-horizontal:

```yaml
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.
Overview

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 6. 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_Visual_Investigator_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.
- 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.

<table>
<thead>
<tr>
<th>Machine Role</th>
<th>Ansible Controller</th>
<th>Connected Repository Mirror</th>
<th>Unconnected Repository Mirror</th>
<th>Deployment Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible Controller</td>
<td>-</td>
<td>recommended</td>
<td>possible</td>
<td>possible</td>
</tr>
<tr>
<td>Connected Repository Mirror</td>
<td>recommended</td>
<td>-</td>
<td>not recommended</td>
<td>possible</td>
</tr>
</tbody>
</table>
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:
   ```bash
   ansible --version
   ```

2. If the command results are similar to the following, then Ansible has been successfully installed on the machine.
   ```bash
   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 17.

#### Confirm the Identities of the Hosts

1. On each machine in your topology, run the following command:
   ```bash
   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.
   ```text
   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_host=<machine_address>
   darkhost ansible_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_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 16.

---

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 21 and “Modify the vars.yml File” on page 25. 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:
   ```bash
   ansible -i hosts -m ping all
   ```

2. Confirm that the command results are similar to the following:
   ```json
   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:
   ```bash
   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:
   ```bash
   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.
## Troubleshooting

<table>
<thead>
<tr>
<th>Error</th>
<th>Explanation</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>After removing software and attempting to reinstall software: Error: Nothing to do</td>
<td>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.</td>
<td>Clean up the yum repository by running the following command: yum remove packagename. You can then reinstall the software.</td>
</tr>
<tr>
<td>After running SAS code: ERROR: Procedure PCA not found ERROR: Procedure KCLUS not found.</td>
<td>The installation was attempted on a system that was not completely cleaned up from a previous installation.</td>
<td>Uninstall SAS/CONNECT by running the following command: yum groups mark remove &quot;SAS/CONNECT&quot;. Reinstall SAS/CONNECT by running the following command: sudo yum groupinstall &quot;SAS/CONNECT&quot;</td>
</tr>
</tbody>
</table>
| When running the deployment: The system failed to gather mount information. | TimeoutError(error_message)\nTimeoutError: Timer expired\n"rc": 257} 13:15:37 | 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. |