SAS® Visual Investigator 10.1.2: Deployment Guide

Introduction

About This Guide

Use this guide to deploy SAS Viya in your environment.

- Make sure that you have the latest version of this deployment guide. The contents of this document are subject to continual updates.
- 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. You are notified if offering-specific content is required.

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, and 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.

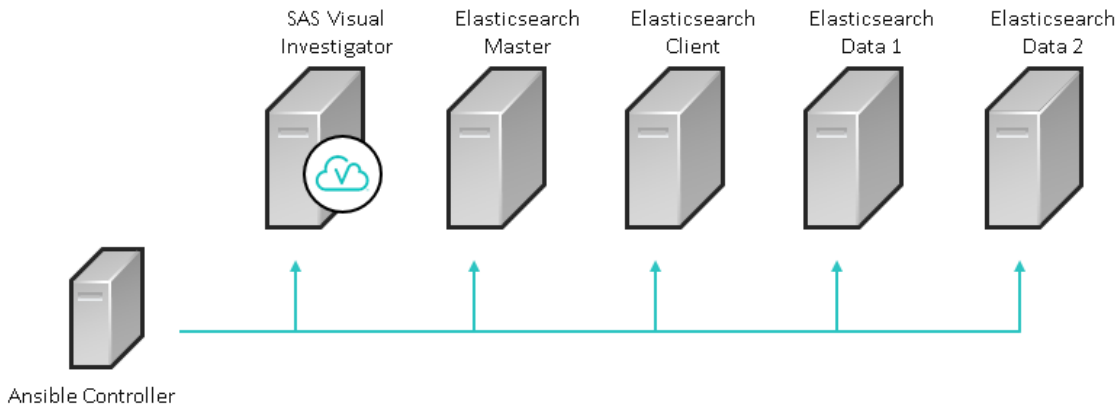
  Note: Data connectors vary according to the order.
- If you purchased SAS Event Stream Processing for SAS Viya (optional), the playbook automatically installs it on every CAS controller and worker machine. You must have an existing deployment of SAS Event Stream Processing 3.2 or a later version in your environment to provide data to the CAS actions that support streaming data. This independent installation of SAS Event Stream Processing must be running on a separate machine on which no CAS components are installed. The independent installation also enables SAS Event Stream Processing Studio.
- 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.

For more information about Elasticsearch, see the documentation at https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html.

Elasticsearch Cluster



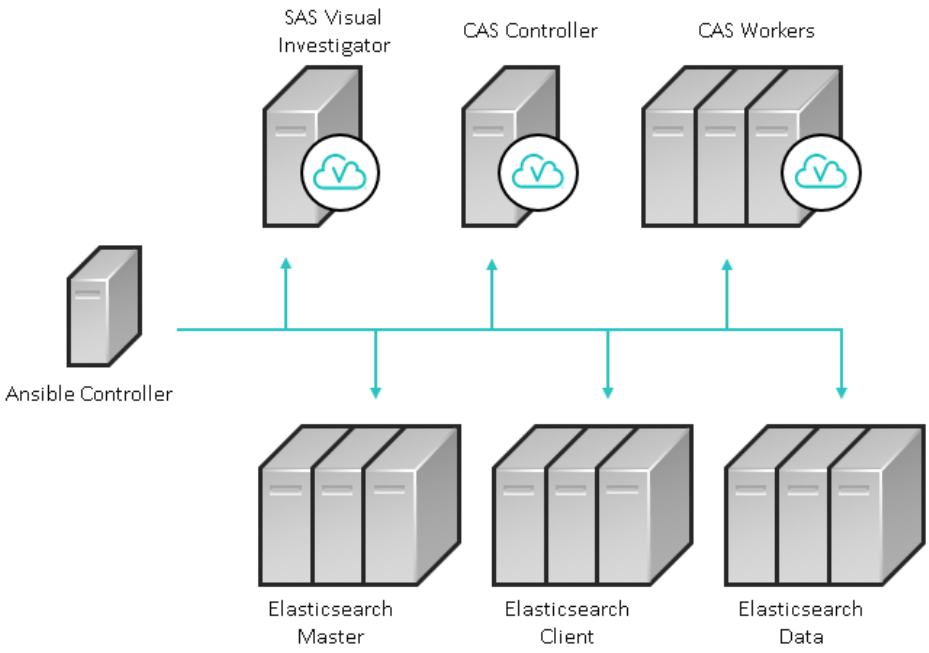
Scenario 2: Elasticsearch in High-Availability Mode

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.

For more information about Elasticsearch, see the documentation at https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html.

Elasticsearch in High-Availability Mode



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 SAS Sizing Expert to obtain an official hardware recommendation that is based on your estimated SAS workload and number of users. The sizing information provided here is not intended as a substitution for expert advice. To request sizing expertise, 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`. For more information, see Log File Space Requirements on page 7.

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

Requirements for Single-Machine Deployment

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

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.

Log File Space Requirements

SAS Viya software is installed in the `/opt` directory on each target machine. Additional space for logs is also required in `/opt/sas/viya`. The amount that is required depends on the logging level that you have set. However, the minimum amount of disk space required for the installation and for logging is 40 GB. We recommend using monitoring tools to ensure that none of the locations used by the deployment fills up without warning.

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

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

The HTTPD component of the Apache web server logs to **/var/log/httpd**. The logs in this directory can grow very large. In addition to using symlinks to change the log location, you should also implement a log rollover strategy. The Apache documentation provides guidance:

- Apache 2.2 log rotation
- Apache 2.4 log rotation

Operating System Requirements

Supported Operating Systems

The following operating systems are supported:

- Red Hat Enterprise Linux 6.7 (64-bit) and later within 6.x
- Red Hat Enterprise Linux 7.1 and later within 7.x

In a multi-machine deployment, we recommend that all server machines have the same version of Linux, including the same patch level. A mixture of operating system levels is not advisable. We also strongly recommend installing identical operating system versions and patch levels on groups of server machines that perform similar roles. For example, use identical operating systems for all CAS machines, or for all machines that host microservices.

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 --installed
```

If the subscription has been activated, the following message is returned: `"subscribed to Red Hat Enterprise Linux Server"`.

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 15.

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:

- libXp
- libXmu

- glibc 2.12
- the numactl package
- the X11/Xmotif (GUI) packages
- libpng (on Red Hat Enterprise Linux 6.*x* or the equivalent)

  libpng12 (on Red Hat Enterprise Linux 7.*x* or the equivalent)
- xterm

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

The setuid mount option must be enabled for the file systems in which SAS software is installed. The processes — sasauth, sasperm, and elssrv — must be able to access these file systems at SAS run time.

Additional Linux Requirements for SAS Event Stream Processing in SAS Cloud Analytic Services

The information in this topic is relevant for users of SAS Event Stream Processing in SAS Cloud Analytic Services. The SAS Event Stream Processing Engine libraries were built using gcc-4.4.7-16 and the Boost library 1.58. The Boost library 1.58 is automatically installed with SAS Event Stream Processing. The libraries are compiled using the following compiler options:

```
-D_REENTRANT
```

```
-D_THREAD_SAFE
```

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

The SAS Event Stream Processing 4.x libraries have been built using gcc-4.4.7-16 on Red Hat Enterprise Linux Server 6.7 using libc-2.12.so, libstdc++.so.6.0.13 and libgcc_s-4.4.7-20120601.so.1

SAS Support for Alternative Operating Systems

Some variants of operating systems are alternatives to the list of officially supported environments. These variants are sometimes derived from a supported distribution's source code that might become part of a future release of a supported distribution.

SAS support for an alternative operating system distribution is limited to installation or functional issues. In addition, SAS software uses technologies from various third-party vendors, which might not support these alternative operating systems at the same level as SAS software. Any attempt to re-create a customer's scenario at SAS is done on an officially supported operating system distribution and third-party vendor software stack. If SAS is unable to reproduce the problem, customers must perform further diagnostics on their own in order to isolate the problem, up to and including reproducing the problem on a supported operating system distribution and third-party vendor software stack.

If you use an alternative operating system, you must have the appropriate skills to resolve differences between the supported operating system and the alternative operating system. By using an alternative operating system, you acknowledge that you can resolve the differences inherent in that alternative system. These restrictions do not apply to virtual applications supplied by SAS.

Server Software Requirements

Java Requirements

The Java Runtime Environment (JRE) must be installed on the machine where you install software to run on SAS Viya. Only the JRE is required, not the full JDK. The following versions are supported:

- Oracle JRE SE version 1.8.0_92 or a later release

- OpenJDK version 1.8.x

  Note: This open-source version of Java is included with Linux.

Past releases of SAS have provided a private JRE in the deployment package. In contrast, SAS Viya supports alternative versions of the JRE or JDK, such as Azul Zulu or IBM SDK, Java Technology Edition.

The deployment 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.

To avoid a problem with rendering that affects a selected set of fonts in Traditional Chinese, Korean, and Japanese locales, use Oracle JRE instead of OpenJDK. As a workaround, you can make a change to the registry to support an alternate set of fonts with OpenJDK. You must also download and install the alternative fonts. For more information, refer to SAS Note 58855.

Data Source and Storage Requirements

Data Encoding Requirement

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

You can also use SAS/CONNECT to transfer and automatically convert data from a non-UTF-8 encoded SAS session to the UTF-8 encoded SAS Viya environment. For information about how to convert data from non-UTF-8 to UTF-8, see "Migrating your Data to UTF-8" in the *SAS® Viya National Language Support (NLS): Reference Guide*.

Supported Data Sources

SAS Viya 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:

```
sudo -v
```

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

```
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:

Account Name and Group	Parameters	Purpose
sas; member of sas group	UID: 1002 GID: 1001 Non-login service account without user restrictions. No password. You can add a password after installation, if necessary. The password does not expire. The default user name is required.	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.

Account Name and Group	Parameters	Purpose
cas; member of sas group	UID: 1001 GID: 1001 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.	Required for managing and enabling Cloud Analytic Services (CAS). Create this user account and add it to the sas group before you start the deployment. The CAS administrator corresponds to the Superuser account. After the installation has completed, use this user account to log on to CAS Server Monitor (the administration user interface) and perform some configuration. Or set another user account as the CAS administrator in the vars.yml file before running the playbook.
viadmin	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 37.	The first user account that signs on to SAS Visual Investigator after the deployment has completed.

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

- apache
- postgres
- rabbitmq

Note: The RabbitMQ component also requires an update to **`/etc/password`**. The deployment process performs this update automatically.

Authentication

SAS Visual Investigator on Linux supports LDAP for user authentication. The deployment process configures some LDAP settings automatically. The SAS Visual Investigator administrator then adds selected LDAP users to the SAS Visual Investigator administration application after the deployment has completed.

Microsoft Active Directory and OpenLDAP are supported LDAP implementations. In addition, the CAS server uses OAuth tokens for all clients in your deployment. Before you start the deployment, you can edit the sitedefault.yml file to have the playbook automatically configure the LDAP identity provider for use by OAuth. For more information, see Edit the sitedefault.yml File on page 31.

When 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. For more information, see Next Steps on page 47.

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

The desktop machine that is used to access the SAS Visual Investigator or CAS Server Monitor user interface requires one of the following web browsers:

- Google Chrome 48 and later
- Microsoft Internet Explorer 11

  Note: Microsoft Edge is not supported.

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, it is recommended for multi-machine deployments.

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.

This type of SAS deployment requires Ansible version 2.1 or a later release. Use the version that is included with the Extra Package for Enterprise Linux (EPEL).

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 machine 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 17.

The Ansible controller machine must be connected to the Red Hat Network.

On Red Hat Enterprise Linux, the Ansible server requires 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. For more information, see Linux Prerequisites on page 8.

The Ansible controller requires Python 2.6 or 2.7. Any nodes that you plan to manage with Ansible require Python 2.4 or a later release. If you are running Python 2.4 or an earlier release on the managed nodes, `python-simplejson` is required. The Ansible controller also requires the following Python modules, which are provided by EPEL if the machine is registered:

- paramiko
- PyYAML
- jinja2

For more information about Ansible installation, see Install Ansible on page 18.

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.

Process	Required Port
default Erlang Port Mapper Daemon (epmd) port	4369
CAS Server Starting Port (used by clients to make binary connections to CAS)	5570
CAS Communicator Port	5580
default Rabbit AMQP client access port	5672
SAS Studio (not required for SAS Visual Investigator)	7080 (if you are performing a visual-only or full deployment, the deployment will use ephemeral ports, so no port needs to be reserved)

Process	Required Port
Object Spawner	8591
CAS Server Monitor (used by clients to make REST HTTP calls to CAS, as with the Python REST interface)	8777
Elasticsearch	9200
default Rabbit management web console port	15672
SAS/CONNECT Spawner	17551
default Rabbit clustering port	25672

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:

```
sudo sysctl net.ipv4.ip_local_port_range
```

The results contain two numbers:

```
net.ipv4.ip_local_port_range = inclusive-lower-limit inclusive-upper-limit
```

2 To list any existing reserved ports, run the following command:

```
sudo sysctl net.ipv4.ip_local_reserved_ports
```

Here is an example of the results:

```
net.ipv4.ip_local_reserved_ports = 23, 25, 53
```

If no ports are reserved, no ports are listed in the results:

```
net.ipv4.ip_local_reserved_ports =
```

3 After you determine the limits of the ephemeral port range, you must add any required ports that are included in the ephemeral port range to the Linux system reserved ports list. Add ports to the reserved list as comma-separated values or as a range within quotation marks:

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

Here is an example:

```
sudo sysctl -w net.ipv4.ip_local_reserved_ports="5672,15672,25672,4369,16060-16069,9200"
```

Note: The sysctl command numerically sorts the port numbers regardless of the order that you specify.

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

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

Firewall Considerations

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

1. Ensure that your firewall is open in order to allow access to the IP address of the content delivery servers that provide updates from Red Hat or from Oracle. The IP addresses for content delivery services vary by region. For more information about the list of IP addresses, see one of the following websites:
   - Public CIDR Lists for Red Hat
   - https://linux.oracle.com/

     This website provides instructions for registering with the Oracle ULN.
2. Ensure that the firewall allows access to the following yum repositories that are hosted by SAS so that content can be delivered for deployment:
   - https://ses.sas.download/
   - https://bwp1.ses.sas.download/
   - https://bwp2.ses.sas.download/
   - https://sesbw.sas.download
3. If you are using a version of Red Hat Enterprise Linux, Oracle Linux, or CentOS that is earlier than version 7.1, run the following commands:

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

   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, run the following commands:

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

   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.

Configure the Use of a Proxy Server

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

Enable the Yum Cache

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

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

```
keepcache = 1
```

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

Install Ansible

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

Installation Steps

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

1 Prepare your machine for Ansible by performing one of the following steps, based on your operating system:

- Make sure that the machine is registered to the Red Hat Network, as described in Linux Prerequisites on page 8.
- If you are using the equivalent of Red Hat Enterprise Linux 6.7, use the following command:

```
sudo rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
```

- If you are using the equivalent of Red Hat Enterprise Linux 7, use the following command:

```
sudo rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

2 List the repository contents on the machine with the following command:

```
sudo yum repolist all
```

Scan the list of packages for the Extra Package for Enterprise Linux (EPEL) package, which is named epel.

3 If the epel package is not available, use the following command in order to add it:

```
sudo yum install epel-release
```

4 When the epel package is available, use the following command in order to ensure that the package is enabled and to install Ansible:

```
sudo yum --enablerepo=epel install ansible
```

Known Issue with Ansible 2.2.1

Ansible 2.2.1 has a known issue which adversely affects SAS Viya software using it. Future releases of Ansible will address the issue.

If you are using Ansible 2.2.1, perform the following steps to successfully deploy SAS Viya software:

1 In the directory where you uncompressed the playbook, open the **`sas_viya_playbook/roles/viya-run/tasks/main.yml`** file:

```
sudo vi sas_viya_playbook/roles/viya-run/tasks/main.yml
```

2 Locate the following line:

```
copy: content="{{ hostvars[inventory_hostname] | to_nice_yaml }}" dest="{{
'/tmp/' + inventory_hostname + '.out' }}"
```

3 Remove "`| to_nice_yaml`" from that line so that it reads like this:

```
copy: content="{{ hostvars[inventory_hostname] }}" dest="{{ '/tmp/' + inventory_hostname + '.out' }}"
```

4 Save and close main.yml.

Test Your Ansible Installation

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

```
ansible localhost -m ping
```

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

Perform Linux Tuning

Set the MaxStartups Variable

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

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 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.

   ```
   *          soft     nproc      65536
   ```

   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          soft     nproc      65536
   ```

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

```
@sas          soft    nproc     65536
```

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

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

Installing SAS Viya with Ansible

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 21.

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 22.

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:

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

Component of the Deployment Target Declaration	Description
ansible_ssh_private_key_file	the private key file that corresponds to the public key that was previously installed on each of the remote machines. This file typically resides in your **~/.ssh** directory.

The following 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=~/.ssh/id_rsa
controller ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
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.

```
[AdminServices]
main

[CASServices]
main

[CoreServices]
main

[DataServices]
main

[HomeServices]
main
```

```
[ReportServices]
main

[ReportViewerServices]
main

[ThemeServices]
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

[microservices]
main

[pgpool]
main

[pgpoolc]
main

[platform-apps]
main

[platform-services]
main

[rabbitmq]
main

[sasdatasvrc]
main

[svi-apps]
main

[sas-casserver-primary]
controller

[sas-casserver-worker]
worker-1
worker-2
```

```
[va-apps]
main

[viprCommon]
main

[viprEntity]
main

[viprSand]
main

[viprVi]
main

[sas-all:children]
AdminServices
CASServices
CoreServices
DataServices
HomeServices
ReportServices
ReportViewerServices
ThemeServices
configuratn
consul
elasticsearch
httpproxy
microservices
pgpool
pgpoolc
platform-apps
platform-services
rabbitmq
sasdatasvrc
svi-apps
sas-casserver-primary
sas-casserver-worker
va-apps
viprCommon
viprEntity
viprSand
viprVi
```

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 alias under [pgpoolc] and [sasdatasvrc]. If the machine that you specify does not have an alias of “deployTarget” in the deployment target declaration, 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 declaration.

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

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

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 alphanumeric characters and the hyphen. 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 13.

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 22.

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 21.

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 15.

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 22.

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 19.

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:

```
sas_install_java: false
```

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

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

4 Save and close the vars.yml file.

For a list of supported versions of Java, see Java Requirements on page 9.

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 27.
- 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

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

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

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

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

1 SSH keys are set up for the CAS user account.

2 A set of keys is created for any other user that is defined in the sas_users field.

3 The private and public keys are copied to each host that the playbook runs against.

4 The ssh-keyscan utility is run from each host to every other host in the CAS cluster.

5 The user’s public key is added to the **~/.ssh/authorized_keys** file.

Use Your Own Passwordless SSH

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

```
setup_sas_users: false
```

3 Save and close the vars.yml file.

Create Customized Passwordless SSH

To use the playbook to set up passwordless SSH, perform the following steps:

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

Note: Comments have been removed from the following example.

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

2 Edit the fields as follows:

a Ensure that the setup_sas_users variable is set to true.

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

Here are the attributes:

- group – the group to which the user belongs. If the group does not exist, it is created when the playbook runs.
- password – the encoded password for the user account. If you do not want to assign a password to the user account, use quotation marks (") that indicate that no password is assigned.

  Note: The comments in the vars.yml file explain how to create an encrypted password.
- setup_home – uses the value of **`true`** or **`false`**. Determines whether the shell and home values should be used by the deployment. To accept the default, use a value of **`false`**.
- shell – the location of the shell for the user account to use. It can be used only if setup_home is set to **`true`**.
- home – the location of the user directory to be created. It can be used only if setup_home is set to **`true`**.

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

d Under extra_packages, specify one or more names of any additional packages to install along with a comment that describes its purpose. The administrator typically uses this field to specify additional packages for the deployment (such as Firefox or Git) as a convenience. The field is ignored if setup_packages is set to **`false`**.

3 Save and close the vars.yml file.

After you edit the fields and run the playbook, the following actions occur:

- If setup_sas_packages is set to true, any listed extra packages are installed.
- After CAS is installed, SSH is set up for any users that are specified in sas_users.
- CAS is configured for passwordless SSH. In addition, when the CAS controller is started, the workers also start.

Install Consul on CAS Hosts

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 22.

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:

```
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.

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 Colocation

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:
     #gcport: 5580
     #httpport: 8777
     #port: 5570
     #colocation: 'hdfs'
```

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 the precedes the HADOOP_NAMENODE and HADOOP_HOME variables.

3 Revise the variables’ values as follows:

```
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 colocated environment, change the colocation variable to 'none', including the single quotation marks:

```
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 22.

Edit the common.yml and host-verification.yml Files

In the common.yml and host-verification.yml files, add the following code:

```
hosts: sas-all
serial: 1
```

The added line should use the same indention as the "hosts: sas-all" line before it.

Edit the sitedefault.yml File

To configure LDAP/AD for use by Oauth, you must modify the sitedefault.yml file located in the **`sas_viya_playbook/roles/consul/files`** directory.

1 Open the sitedefault.yml file:

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

It looks like this:

```
# Site-Specific Defaults
#
# config:
#     application:
#         sas.identities.providers.ldap.connection:
#             host:
#             password:
#             port:
#             url:
#             userDN:
#         sas.identities.providers.ldap.group:
#             baseDN:
#         sas.identities.providers.ldap.user:
#             baseDN:
#             searchFilter:
```

2 Remove the number sign (#) from the beginning of every line except the first two lines.

3 Fill in the empty fields as follows. Highlighted lines should be added to the file in their entirety. Notice that each value is enclosed in single quotation marks. Also note that maintaining the indention is important.

```
# 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.providers.ldap.profile:
            file: 'ldap/ldap-search-and-bind.xml'
```

4 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:

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

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

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

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

Does not require passwords	**ansible-playbook -i host_local site.yml**

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

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.

Modify an Existing Deployment

For information about modifying an existing deployment with updated software or adding new software to an existing deployment, see SAS Viya 3.1 Administration:Software Updates at http://support.sas.com/documentation/onlinedoc/viya/index.html.

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:

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

Manual Configuration Tasks

Set the Password for the CAS Administrator or Another Administrative Account

To enable the cas user account to be the CAS administrator, you must add a password to the cas user account on the CAS controller and all CAS worker nodes. To assign a password, use the following command:

```
sudo passwd cas
```

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.

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 and the **/opt/sas/viya/config/etc/cas/default/perms.xml** file.

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 cas.oauthsigningkey, and change the line as follows:

```
cas.oauthsigningkey = 'tokenkey'
```

4 Save and close the file.

5 Enter the following commands from the shell:

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

6 Open and edit the **/opt/sas/viya/config/etc/cas/default/perms.xml** file as root:

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

Note: The specified user for user=cas can be the user ID for the casenv_admin_user. An example is user=casenv_admin_user.

a Locate the following line:

```
<!-- CAS server initial launch identity -->
  <Administrator name="cas-User-SuperUser" user="cas" type="SuperUser"/>
```

Change the line as follows:

```
<!-- CAS server initial launch identity -->
 <Administrator name="FIRSTADMINUSER-User-SuperUser" user="viadmin" type="SuperUser"/>
 <Administrator name="cas-User-SuperUser" user="cas" type="SuperUser"/>
```

b Locate the **Manage Global Caslibs** section and change all instances of the SAS Visual Investigator administrator user ID to the appropriate value as follows:

```
 <!-- Manage Global Caslibs -->
  <CASLib name='_GLOBAL' desc='Permission for global caslib creation'>
       <!-- Global caslib creation is open to administrators -->
      <Grant group='SASAdministrators' perm='ManageAccess'/>
      <Grant user='viadmin' perm='ManageAccess'/>
      <Grant user='cas' perm='ManageAccess'/>
  </CASLib>
      <CASLib name='Formats'
     desc='CAS Formats caslib'
      path='/opt/sas/viya/config/data/cas/default/formats'
      uuid='36ea7916-86af-4547-9180-9876598765cc'
      subdirs='TRUE' type='PATH'>

     <Grant group='SASAdministrators' perm='ManageAccess'/>
     <Grant user='viadmin' perm='ManageAccess'/>
     <Grant user='cas' perm='ManageAccess'/>

     <Grant group='SASAdministrators' perm='AlterCaslib'/>
     <Grant user='viadmin' perm='AlterCaslib'/>
     <Grant user='cas' perm='AlterCaslib'/>

     <Grant group='SASAdministrators' perm='CreateTable'/>
     <Grant user='viadmin' perm='CreateTable'/>
     <Grant user='cas' perm='CreateTable'/>

     <Grant group='SASAdministrators' perm='DropTable'/>
     <Grant user='viadmin' perm='DropTable'/>
     <Grant user='cas' perm='DropTable'/>

     <Grant group='SASAdministrators' perm='AlterTable'/>
     <Grant user='viadmin' perm='AlterTable'/>
     <Grant user='cas' perm='AlterTable'/>

     <Grant group='SASAdministrators' perm='Promote'/>
     <Grant user='viadmin' perm='Promote'/>
     <Grant user='cas' perm='Promote'/>

     <Grant group='SASAdministrators' perm='LimitedPromote'/>
     <Grant user='viadmin' perm='LimitedPromote'/>
     <Grant user='cas' perm='LimitedPromote'/>
```

```
    <Grant group='SASAdministrators' perm='DeleteSource'/>
    <Grant user='viadmin' perm='DeleteSource'/>
    <Grant user='cas' perm='DeleteSource'/>

    <Grant group='SASAdministrators' perm='Insert'/>
    <Grant user='viadmin' perm='Insert'/>
    <Grant user='cas' perm='Insert'/>

    <Grant group='SASAdministrators' perm='Delete'/>
    <Grant user='viadmin' perm='Delete'/>
    <Grant user='cas' perm='Delete'/>

    <Grant group='*' perm='ReadInfo'/>
    <Grant group='*' perm='Select'/>

</CASLib>
```

c Save and close the file.

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

```
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:

```
# useHostToken
```

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

```
# externalIdent
```

c Save and close the file.

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

```
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 using the following command:

```
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.

```
#  Enter the TLS  (https) domain in the TLSDOMAIN variable
TLSDOMAIN=
```

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.

```
#  Enter the Internal  (http) domain in the INTERNALDOMAIN variable standard http address
for your deployment it is called internalvdomain as that is where the internal communications between the
pieces will be occurring.
INTERNALDOMAIN=http-domain
```

5 Locate the line that contains SASHOME. Enter the **/opt/sas/viya/home** path to the SAS Home directory.

```
# Enter the path to the sas/viya/home directory
SASHome=/opt/sas/viya/home
```

6 Locate the line that contains ADMIN_ID. Enter the same user ID that you specified for the SAS Visual Investigator administrator user ID in the perms.xml file.

```
#  Enter the username that will be the admin user for Visual Investigator.
#  This must be the user that signs on to Visual Investigator first.
ADMIN_ID=viadmin
```

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

```
#  Enter the token string that will be used to obtain an access token from
#  SASLogon for client registration
ADMIN_TOKEN=admin-token-string
#  Enter the token string that will be used by other apps if it is set to
#  verify that they can acquire a token from an admin application.
CLIENT_TOKEN=client-token-string
```

Note: Generate a separate hash for the ADMIN_TOKEN and CLIENT_TOKEN values using this command: `echo -n Welcome | md5sum`.

Before you enter the hash for ADMIN_TOKEN or the hash for CLIENT_TOKEN, remove the trailing `-`.

8 Locate the line that contains SIGNING_KEY. Enter the token string with the value of tokenkey:

```
#  Enter the token string that will be used by apps and users by CAS to allow
#  them to connect remotely to CAS.
SIGNING_KEY=tokenkey
```

9 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 `-`.

10 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
```

11 Locate the line that contains $SASHome/bin/sas-bootstrap-config kv, and add the --force option:

```
$SASHome/bin/sas-bootstrap-config kv write --force "$Key" "$Value"
```

12 Save and close the file.

13 On the Consul machine, run the following script:

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

This script loads the required Consul KV pairs into Consul.

Configure the Data Hub

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

2 On the Consul machine, run the following script:

```
/opt/sas/viya/config/share/svi-datahub/datahub_configdb.sh
```

Configure the Script to Locate the Elasticsearch Service

On each machine where Elasticsearch is installed, complete the following steps:

Note: Elasticsearch is installed on one or more machines with the [elasticsearch] target.

1 Locate and edit the sas-viya-svi-elasticsearch-default file in the **/etc/init.d** directory.

2 Locate the following lines:

```
#!/bin/sh
#
# elasticsearch-server Elasticsearch broker
#
# chkconfig: - 80 05
# description: Enable AMQP service provided by Elasticsearch
#

### BEGIN INIT INFO
# Provides:          elasticsearch-server
```

3 Insert two comment lines so that the line that contains elasticsearch-server is on the fifth line:

```
#!/bin/sh
#
#
#
# elasticsearch-server Elasticsearch broker
#
# chkconfig: - 80 05
# description: Enable AMQP service provided by Elasticsearch
#

### BEGIN INIT INFO
# Provides:          elasticsearch-server
```

4 Save and close the file.

5 Locate and edit the sas-viya-all-services file in the **/etc/init.d** directory.

6 Locate the line that contains backup-agent httpproxy and change the line as follows:

```
backup-agent httpproxy rabbitmq-server sasdatasvrc elasticsearch)
```

7 Save and close the file.

8 For Red Hat Enterprise Linux 7.x only, to reload the file, enter the following command:

```
sudo systemctl daemon-reload
```

Restart the Applications

On the Consul machine, to restart applications, run the following script:

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

Note: The restart_apps.sh script does not restart all services. To manage the entire environment, use the sas-viya-all-services script. The sas-viya-all-services script does not manage the Elasticsearch status. Use the following command for the Elasticsearch status:

```
sudo service sas-viya-svi-elasticsearch-default status
```

When the script completes, you should have a working SAS Visual Investigator installation.

Validating the Deployment

Perform Installation Qualification on RPM Packages

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

Here is the basic command to verify RPM packages:

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

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

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

You can also create a for loop command for verifying multiple packages that share a common naming convention. For example, to verify all packages whose names begin with `sas-`, use the following query:

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

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

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

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

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

The error indicators are shown in the following format:

```
SM5DLUGT c
```

In addition, if a file is missing, the error message contains the phrase “missing”:

```
missing  /opt/sas/viya/home/lib/envesml/sas-init-functions
```

The meaning of each error indicator is described as follows:

- S - file size

  RPM keeps track of file sizes. A difference of even one byte triggers a verification error.

- M - file mode

  The permissions mode is a set of bits that specifies access for the file's owner, group members, and others. Even more important are two additional bits that determine whether a user's group or user ID should be changed if they execute the program that is contained in the file. Since these bits permit any user to become root for the duration of the program, you must be cautious with a file's permissions.

- 5 - MD5 checksum

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

- D - major and minor numbers

Device character and block files contain a major number. The major number is used to communicate information to the device driver that is associated with the special file. For example, under Linux, the special files for SCSI disk drives should have a major number of 8, and the major number for an IDE disk drive's special file should be 3. Any change to a file's major number could produce disastrous effects. RPM tracks such changes.

A file's minor number is similar to the major number, but conveys different information to the device driver. For disk drives, this information can consist of a unit identifier.

- L - symbolic link

  If a file is a symbolic link, RPM checks the text string that contains the name of the symbolically linked file.

- U - file owner

  Most operating systems keep track of each file's creator, primarily for resource accounting. Linux and UNIX also use file ownership to help determine access rights to the file. In addition, some files, when executed by a user, can temporarily change the user's ID, normally to a more privileged ID. Therefore, any change of file ownership might have significant effects on data security and system availability.

- G - file group

  Similar to file ownership, a group specification is attached to each file. Primarily used for determining access rights, a file's group specification can also become a user's group ID if that user executes the file's contents. Therefore, any changes in a file's group specification are important and should be monitored.

- T - modification time

  Most operating systems keep track of the date and time that a file was last modified. RPM keeps modification times in its database.

- c - configuration file

  This is useful for quickly identifying configuration files, since they are likely to change and therefore are unlikely to verify successfully.

Access CAS Server Monitor

To verify that CAS Server Monitor has been successfully deployed, access it by opening a web browser and entering the URL in the address field in the following format:

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

Here is an example:

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

Note: During the initial deployment, CAS Access Monitor is set up for HTTP only. Additional manual steps are required to configure the CAS Server Monitor for HTTPS. For information about securing CAS Server Monitor, see “Configure CAS Server Monitor for HTTPS” in *Encryption in SAS Viya*.

Log on using the cas user ID and the cas password. If you changed the default cas user, use the user account information that you set up.

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 35.

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.

Validate SAS Visual Investigator

1 Find the user name that was specified as the ADMIN_ID in the script that is located at **/opt/sas/viya/config/share/svi-visual-investigator/svi-visual-investigator.sh**.

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

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

4 When you are prompted for **Assumable Groups**, click **Yes**.

5 When the SAS Visual Investigator window opens, the user name appears in the upper right corner. Click that name.

6 If the options for **SAS Visual Investigator Admin** and **Sign out** have been specified, then SAS Visual Investigator has been deployed appropriately.

Uninstalling SAS Viya

Prepare to Uninstall

1 Open the **/etc/rc.d/init.d/sas-viya-audit-default** file.

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

2 Find the line that reads as follows:

```
# chkconfig: -97 03
```

3 Change that line to read as follows. Note the addition of a space before “97”.

```
# chkconfig: - 97 03
```

4 Save and close the **/etc/rc.d/init.d/sas-viya-audit-default** file.

Uninstall from a Single Machine

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

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

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

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

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:

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

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

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 10.1: Administrator's Guide*.

      Note: All users must be valid LDAP users.

Note: You can access *SAS Visual Investigator 10.1: 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 10.1 documentation, you must have an access key. The documentation page explains how to contact SAS Technical Support to request the access key.

Appendix A: Creating and Using Mirror Repositories

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

General Requirements

The environment must meet the following requirements.

The Ansible controller must have the following:

- access to the Internet
- the ability to connect to the mirror repository host
- the ability to connect to the machines on which the software will be deployed
- 15 GB of free space in **`/tmp`** to hold the necessary files

The mirror repository host must have the following:

- the ability to connect to the Internet
- 15 GB of free space in **`/var/www/html/pulp`** to hold the mirror repository files
- 15 GB of free space in **`/tmp`** to hold the temporary archive of the repository mirror files

Use Ansible to Create a Mirror Repository

These instructions assume that Ansible has been installed on a separate machine from the machine on which the software will be deployed.

Note: For more information about installing Ansible, see Install Ansible on page 18.

Perform the following steps to create and use a mirror repository using Ansible:

1. Go to the directory that was created when you uncompressed your playbook from the Software Order Email (SOE). For more information, see Make Sure That You Have the Required Files on page 15.
2. Using a text editor, create a file named repohost.ini that contains the following content:

   ```
   machine-name ansible_ssh_host=IP-address-for-machine-name

   [repohost]
   machine-name
   ```

   machine-name is the name of the mirror repository host.
3. Save the repohost.ini file.
4. If you have already installed the httpd software, run the following command on the appropriate host:

   ```
   ansible-playbook -i repohost.ini reposync.yml
   ```

   The results of running this command follow:

   - The host *machine-name* begins running as an Apache httpd server from which the deployment process obtains files.

- The file **/var/www/html/pulp/repos/repo.override.txt** is created.
- A new customized_deployment_script.sh file is created, and is located in**/tmp/mirror/location/**. It will be modified to use the mirror repository.
- The Ansible controller downloads all the RPM files.
- The RPMs are copied to *machine-name*.

If httpd is not installed, run the following command:

```
ansible-playbook -i repohost.ini reposync.yml -e "setup_httpd_for_sync=true"
```

The results of running this command follow:

- httpd is installed.
- SELinux is disabled.
- Firewalls are disabled for the machine.
- httpd is configured.

5 If you install SAS Viya on the same host that is also the mirror of the SAS repository, run the following command:

```
sudo chown sas:sas ~sas/.ssh/authorized_keys
```

6 The URL in the repo.override.txt file is based on the value for the ansible_ssh_host variable. If the default URL is incorrect or if you need to provide a customized port value, edit the sasenv_mirror.yml file and set the repomirror_url variable to the appropriate value.

7 Perform the steps that are described in Installing SAS Viya with Ansible on page 21 up to Deploy the Software on page 32. Do not run the command in that step. Instead, run the following command:

```
ansible-playbook -i hosts site.yml -e "@/tmp/mirror/location/repo.override.txt"
```

Depending on your system setup, the user ID might require passwords for sudo or SSH. Add the following arguments based on the password requirement, if any, for the user ID that performs the deployment.

User requires a sudo password only.	**--ask-become-pass**
User requires an SSH password only.	**--ask-pass**
User requires both a sudo password and an SSH password.	**--ask-pass --ask-become-pass**

8 After the deployment has run, continue with the deployment instructions at Manual Configuration Tasks on page 35.

Appendix B: Troubleshooting

Error	Explanation	Resolution
After removing software and attempting to reinstall software: `Error: Nothing to do`	The directories containing the software were deleted. However, the yum remove command was never run. In the /var/log/yum.log, the last entry for the rpm is `Installed`.	Clean up the yum repository by running the following command": `yum remove packagename` You can then reinstall the software.

After running SAS code:
`ERROR: Procedure PCA not found`
`ERROR: Procedure KCLUS not found.`
	The installation was attempted on a system that was not completely cleaned up from a previous installation.	Uninstall SAS/CONNECT by running the following command:
		`yum groups mark remove "SAS/CONNECT"`
		Reinstall SAS/CONNECT by running the following command:
		`sudo yum groupinstall "SAS/CONNECT"`

When running the deployment:
`TimeoutError(error_message)\nTimeoutError:`
`Timer expired\n", "rc": 257}  13:15:37 |`
`INFO: | * 13:15:37 |`
`WARNING: | Execution return code '2'`
`is not the expected value '0' 13:15:37 |`
`INFO: | * 13:15:37 |`
`INFO: | Updating deployment times data`
`for step deploy_time with value 19 13:15:37 |`
`INFO: | * 13:15:37 |`
`WARNING: | Ansible execution`
`encountered failures`
	The system failed to gather mount information.	Do one of the following:
		■ Set **`/etc/mtab`** as a link to **`/proc/mounts`** by running the following command:
		`sudo ln -s /proc/mounts /etc/mtab`
		■ Edit the ansible.cfg file and add or change the time-out value for Ansible as follows:
		`timeout=number of seconds`
		Deploy your software by running the Ansible playbook again.