

SAS[®] Visual Investigator 10.3.1 on Linux: Deployment Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS® Visual Investigator 10.3.1 on Linux: Deployment Guide*. Cary, NC: SAS Institute Inc.

SAS® Visual Investigator 10.3.1 on Linux: Deployment Guide

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

June 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

10.3.1-P1:dplyvi0phy0lax

Contents

Chapter 1 / Introduction	1
About This Guide	1
How Deployment Works	2
Deployment Examples and Guidance	2
Contact SAS Technical Support	4
Chapter 2 / System Requirements	5
Hardware Requirements	6
Operating System Requirements	8
Server Software Requirements	10
Data Source and Storage Requirements	11
User and Group Requirements	13
Security Requirements	15
Client Requirements	16
Deployment Tools	16
Chapter 3 / Pre-Installation Tasks	19
Make Sure That You Have the Required Files	19
Configure SELinux	20
Enable Required Ports	20
Firewall Considerations	22
Configure Use of a Proxy Server	24
Enable the Yum Cache	24
Enable a Shared File System	25
Install Ansible	25
(Optional) Enable Key-Based SSH Authentication	27
Perform Linux Tuning	27
Verify Wildcard DNS Alias	30
Install the Database Client	30
Chapter 4 / Installing SAS Viya with Ansible	31
Modify the Initial Deployment	32
Use a Mirror Repository	32
Edit the Inventory File	32
Modify the vars.yml File	37
Create the sitedefault.yml File	50
Revise Elasticsearch File	53
Deploy the Software	54
Install with SAS 9.4 Software	55
Deployment Logs	56
Chapter 5 / Post-Installation Tasks	57
Set the Password for the CAS Administrator or Another Administrative Account	57
Reset the PostgreSQL Passwords	58
Change the Administrative User Password for SAS Message Broker	58
Configure SAS/ACCESS Interface to DB2	59
Configure ODBC	59
Configure SAS/ACCESS Interface to Oracle	60
Configure SAS Data Connector to PostgreSQL	60
Configure SAS/ACCESS Interface to Teradata	61

Configure CAS For SAS Visual Investigator	62
Restart the SAS Visual Investigator Applications	62
Connect to PostgreSQL	63
Configure a Standard or Multi-tenant Deployment	63
Configure the First System User	66
Chapter 6 / Validating the Deployment	69
Perform Installation Qualification on RPM Packages	69
Access CAS Server Monitor	71
Verify RabbitMQ	71
Verify PostgreSQL	72
Verify SAS/ACCESS Interface to DB2	72
Verify SAS/ACCESS Interface to ODBC	73
Verify SAS/ACCESS Interface to Oracle	74
Verify SAS/ACCESS Interface to PostgreSQL	76
Verify SAS/ACCESS Interface to Teradata	77
Validate Elasticsearch	78
Validate SAS Visual Investigator	79
Chapter 7 / Uninstalling SAS Viya	81
What deploy-cleanup Does	81
Uninstall Command	81
Chapter 8 / Completing the Deployment	83
Save Snapshot Directory Content	83
Further Documentation	83
Chapter 9 / Managing Your Software	85
Overview	85
Updating Your SAS Viya Software	86
Add SAS Viya Software	89
Migrating the Configuration Information	91
Upgrading Your SAS Viya Software	93
Generate a New Ansible Playbook	95
Appendix 1 / Creating High Availability PostgreSQL Clusters	99
Overview	99
HA PostgreSQL Topologies	99
Set Up a Horizontal Cluster	101
Set Up a Vertical Cluster	103
Set Up a Hybrid Cluster	104
Set Up Multiple Clusters	105
Deployment Logs	108
Verify the Deployment	109
Appendix 2 / Creating and Using Mirror Repositories	111
Overview	111
Requirements	111
Use Ansible to Create a Mirror Repository	113
Uninstalling SAS Viya from Mirrored Repositories	116
Appendix 3 / Troubleshooting	119
Troubleshooting SAS Viya	119

Introduction

About This Guide	1
Get the Latest Guide	1
SAS Products and Supporting Components	1
Audience	2
How Deployment Works	2
Deployment Examples and Guidance	2
About the Deployment Examples	2
Elasticsearch Cluster	3
Elasticsearch in High-Availability Mode	3
Contact SAS Technical Support	4

About This Guide

Get the Latest Guide

Make sure that you have the latest version of this guide, which is available at the following site:

[SAS Viya Deployment Guides](#)

Note: The contents of this guide are subject to continual updates. If you are viewing a saved copy of the PDF version of this guide, the content might be outdated.

SAS Products and Supporting Components

This guide provides information for deploying software that is listed in your Software Order Email (SOE), which can include the following products:

SAS Visual Investigator
 SAS Cloud Analytic Services (CAS)
 Elasticsearch
 SAS/ACCESS Interface to DB2 (on SAS Viya)
 SAS/ACCESS Interface to ODBC
 SAS/ACCESS Interface to Oracle (on SAS Viya)
 SAS/ACCESS Interface to PostgreSQL (on SAS Viya)
 SAS/ACCESS Interface to Teradata (on SAS Viya)

Note: Unless another situation is specifically cited, the information in this guide pertains to the software that you ordered.

Audience

This guide is written for administrators who install and configure software for your company or organization. To perform the steps in this guide, you should have a working knowledge of Ansible, which is the preferred tool for deploying and updating SAS Viya. Also, you should have a working knowledge of the Linux operating system and basic commands.

How Deployment Works

You use Ansible to deploy SAS Viya to one or multiple machines.

- Ansible is a software orchestration tool that provides a straightforward approach to deploying SAS Viya. To deploy using Ansible, you customize files for your environment, and then you run a command to deploy software according to the values in those files. The set of files, known collectively as “the playbook,” provides the instructions about what software is deployed on which machines. In this guide, “run the playbook” means to deploy or update SAS Viya.
- Before you can run the playbook, you must create one that is customized for your order. To do that, you use the SAS Orchestration CLI. The Software Order Email (SOE) that SAS sends to your business or organization contains a link to instructions on how to use the SAS Orchestration CLI.
- Each time you run the playbook, Ansible automates a series of yum commands that securely access the latest SAS Viya software to which you are entitled. The software is downloaded from repositories that are maintained by SAS or from a local mirror repository that you create and maintain at your own site.

Note: Yum is a software-package manager for Linux operating systems. SAS Viya is packaged in the RPM Package Manager (RPM) format, which simplifies installation and upgrade tasks.

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

Deployment Examples and Guidance

About the Deployment Examples

- Ansible is used to deploy the software. Ansible is shown as installed on a separate machine, called the Ansible controller.
- Elasticsearch, which provides search capabilities for SAS Visual Investigator, is used to generate data for visualizations.
- The SAS Cloud Analytic Services (CAS) server provides the run-time environment where data management and analytics take place.
 - 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: primary CAS controller, secondary CAS controller, or CAS worker. If you deploy the CAS server to a single machine, the primary CAS

controller role is assigned. For a distributed CAS server, the controller roles and worker roles are assigned.

- If you purchased one or more data connectors, they must be deployed to one or more machines on which CAS is running.

Note: Data connectors vary according to the order.

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

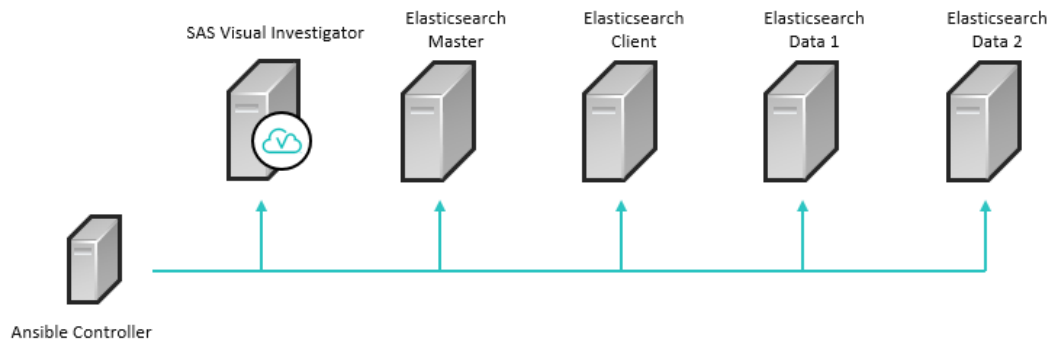
Elasticsearch Cluster

In this example, Elasticsearch is deployed across multiple nodes in a clustered environment. An advantage of this deployment is that optimal processing can be achieved through spreading the Elasticsearch queries across multiple nodes. During deployment, the Elasticsearch master 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 the following site:

<https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html>

Figure 1.1 Elasticsearch Cluster



Elasticsearch in High-Availability Mode

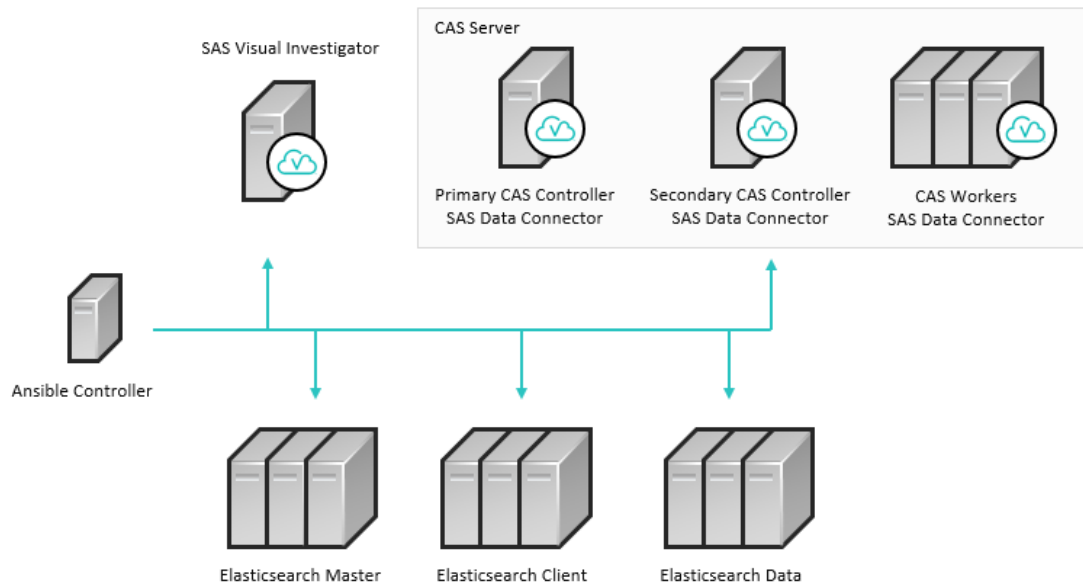
In this example, Elasticsearch is deployed in high-availability mode with multiple master nodes and client nodes.

- 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.
- The CAS server is deployed across multiple nodes in a clustered environment. An advantage of this deployment is that optimal processing can be achieved through massively parallel processing (MPP) for multiple users. During deployment, the CAS roles are assigned to the nodes and a data connector is configured.

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

<https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html>

Figure 1.2 *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	6
Host Requirements	6
Hardware Requirements for SAS Visual Investigator	6
File Space Requirements	7
Operating System Requirements	8
Supported Operating Systems	8
Linux Requirements	8
Additional Linux Requirements for SAS Visual Investigator	10
SAS Support for Alternative Operating Systems	10
Server Software Requirements	10
Java	10
Apache httpd	11
Data Source and Storage Requirements	11
Supported Data Sources	11
Data Encoding Requirement	12
Requirements to Transfer Data from SAS 9.4	12
Requirements for SAS/ACCESS Interface to DB2	12
Requirements for SAS/ACCESS Interface to ODBC	12
Requirements for SAS/ACCESS Interface to Oracle	12
Requirements for SAS/ACCESS Interface to PostgreSQL	12
Requirements for SAS/ACCESS Interface to Teradata	12
User and Group Requirements	13
Set Up the Account that Deploys the Software	13
Additional User Account Requirements	13
Security Requirements	15
LDAP Requirements	15
Enhance Default Security Settings	15
Transport Layer Security Requirements for the SAS Embedded Process	16
Client Requirements	16
Web Browser Requirements	16
Database Drivers	16
Deployment Tools	16
Ansible Controller Requirements	16

Hardware Requirements

Host Requirements

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

- A static IP address

The Consul component binds to a single private IP address per machine. If any of your intended hosts has multiple network interface cards (NICs), verify whether multiple NICs have assigned IP addresses, including private IP addresses. To avoid an error during the deployment, you must edit the inventory file to add a `consul_bind_adapter` parameter for these hosts. For more information, see [“Single-Machine Deployment” on page 33](#).

- A static host name

Some networking environments, such as Dynamic Host Configuration Protocol (DHCP), and some cloud providers use dynamic host names or IP address assignments by default. Although it is possible to deploy the software successfully in these environments, any future change to either IP addresses or host names might result in an inoperative deployment. Therefore, SAS recommends that before you start the installation, you work with your network administrator to ensure that IP addresses and host names are static.

- A fully qualified domain name that is 64 characters or fewer in length. This requirement is included in prerequisite checking.

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

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

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

To increase available disk space for the installation, SAS recommends that you mount additional volumes at `/opt/sas` instead of to a subdirectory of `/opt/sas`. Mounting a volume in the installation directories increases the difficulty of uninstalling the SAS Viya volume or moving the volume to another location at a later time.

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 Viya components can be installed on a single machine or on multiple machines. However, SAS does not recommend installation on a single machine if your environment will include large volumes of data or high levels of user concurrency. In these cases, SAS recommends using 3 - 5 separate machines for the Elasticsearch component.

- ☐ (Optional) Obtain an official hardware recommendation that is based on your estimated SAS workload and number of users. The sizing information provided here is not intended as a substitution for expert advice.

To request sizing expertise, contact your SAS account representative. If you need assistance in determining your SAS account representative, send an email to contactcenter@sas.com.

- ☐ Verify that at least 10 GB of disk space are available for your SAS Viya installation. The installation files are automatically downloaded to the `/var/cache/yum` directory.

- ☐ Verify that additional space for logs is available in `/opt/sas/viya`. For more information, see [“File Space Requirements” on page 7](#).

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

Table 2.1 Requirements for Single-Machine Deployment

Item	Minimum Level
CPU	Intel or AMD CPU with 4 cores x86_64 architecture with a minimum speed of 2.6 GHz 32-bit chipsets are not supported.
Memory	64 GB of RAM
Disk Space and Speed	40 GB 10,000 RPM

In a multi-machine deployment, follow similar minimum guidelines for each target machine. However, additional memory should be allocated to machines where multiple services will be deployed. And the machine where Elasticsearch is installed also has different requirements:

Table 2.2 Requirements for the Elasticsearch Machine

Item	Minimum Level	Recommended Level
CPU	Intel or AMD CPU with 2 cores, x86_64 architecture	Additional cores are preferred over more powerful CPUs. 32-bit chipsets are not supported.
Memory	16 GB of RAM	64 GB of RAM This is the Elasticsearch recommended maximum level. The HeapSize parameter should be set to about one-half the RAM capacity of the machine. You can set this during playbook configuration. For more information, see the comments in the <code>inventory.ini</code> file.
Disk Space and Speed	2 x 300 GB 10,000 RPM	SSDs are preferred over spinning media. If you use spinning media, high-performance server disks, such as those found in 15k RPM drives, are recommended.

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.

File Space Requirements

SAS Viya software is installed in the `/opt` directory on each target machine. In many cases, this directory is on a file system with 50 GB of disk space or fewer.

Disk space requirements depend on many factors related to usage scenarios. However, several predefined system caslibs and the Public caslib all have a default location for persistent storage: `/opt/sas/viya/`

`config/data/cas/default/public`. If you anticipate that many users will use browsers to access the user interfaces and import data from files, additional space for this file system will be required.

The CAS controller also uses additional disk space for the CAS cache directory. You can configure the location of this cache in the playbook. For more information, see [Set Up the CAS Cache Directory](#).

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

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

Additional space for logs is required in `/opt/sas/viya`. The amount that is required depends on the logging level that you have set. However, the minimum amount of disk space that is required for the installation and for logging is 40 GB.

Regularly monitoring disk space usage for caslibs and logs is a critical CAS administrative task. For more information, see [Monitor Disk Activity](#) in *SAS Viya Administration: Monitoring*.

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

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

As part of your log management strategy, create symbolic links at the `/opt/sas` level in order to capture all logging activity from SAS Viya components.

The HTTPD component of the Apache HTTP server logs to `/var/log/httpd`. The logs in this directory can grow very large. In addition to using symbolic links to change the log location, you should also implement a log rollover strategy. See the Apache documentation for guidance about log rotation.

Operating System Requirements

Supported Operating Systems

For the full list of supported operating systems, see

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-operating-systems.html>

Linux Requirements

A SAS Viya deployment requires the operating system to be registered with the Red Hat Network or Oracle Unbreakable Linux Network (ULN). Registration enables you to receive periodic software updates. For a SAS software deployment, registration also enables yum to download software from SAS repositories. Verify that the machine where you perform the deployment (typically, the Ansible controller) is registered and that your subscription has been activated. To use Ansible for the deployment, the Ansible controller machine must be connected to the Red Hat Network with a Server-Optional subscription in addition to the Base (operating-system) subscription. The managed nodes must also be registered to the Red Hat Network, but a Base subscription is sufficient.

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

```
subscription-manager version
```

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

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

A list of active subscriptions is returned.

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

```
uln_register
```

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

If you have enabled Security-Enhanced Linux (SELinux) in your environment, you must enable permissive mode on all of the target machines in your deployment. For more information, see [“Configure SELinux” on page 20](#).

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

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

The following libraries are required:

- glibc-2.12-1.166.el6 and later (on Red Hat Enterprise Linux 6.x or the equivalent). Refer to [RHBA-2015:1465](#) to obtain the latest updated package list.
- glibc-2.17-107.el7 and later (on Red Hat Enterprise Linux 7.x or the equivalent). Refer to [RHSA-2016:2573](#) to obtain the latest updated package list.
- libpng (on Red Hat Enterprise Linux 6.x or the equivalent)
- libpng12 (on Red Hat Enterprise Linux 7.x or the equivalent)
- libXp
- libXmu
- net-tools
- the numactl package
- the X11/Xmotif (GUI) packages
- xterm

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

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

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

```
$ yum update systemd
```

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

Be sure to follow the steps that are described in [“Perform Linux Tuning” on page 27](#) before starting the deployment process.

Additional Linux Requirements for SAS Visual Investigator

For SAS Visual Investigator, the following additional utilities are required:

- gettext-0.17-18
- bc 1.06.95

Note: These first two utilities must be installed on each machine where your PostgreSQL database is installed. These machines are members of the [sasdatasvc] group in your inventory file.

To install them, run the following commands:

```
sudo yum install -y gettext
sudo yum install -y bc
```

- jq-1.5

Install this utility on each machine in the [sviconfig] and [elasticsearch] host groups in your inventory file.

Note: You can run the following command to install the jq utility on Red Hat Enterprise Linux 7.x:

```
sudo yum install -y jq
```

However, on Red Hat Enterprise Linux 6.x, this command installs an unsupported version of jq. Run the following commands instead to obtain a more recent version of jq:

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

SAS Support for Alternative Operating Systems

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

Server Software Requirements

Java

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

For a list of supported JRE distributions and other requirements, see:

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-jre.html>

SAS Viya supports alternative distributions of the JRE, such as Azul Zulu. However, IBM SDK, Java Technology Edition is not supported. In some cases, running `sudo yum install java` to install Java can result in the unintentional installation of the IBM JRE, which causes failures with an installation utility.

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

```
java -version
```

Apache httpd

The deployment process can automatically install Apache httpd if it is not detected on the machines that you designate as targets for the HTTP reverse proxy installation. Apache httpd is required in order to create the Apache HTTP Server, which provides security and load balancing for multiple SAS Viya components.

SAS recommends that you install Apache httpd before you start the deployment process. You can then update the Apache HTTP Server to use certificates that comply with the security policies at your enterprise, and the playbook can distribute them automatically. For more information, see [“Enhance Default Security Settings” on page 15](#).

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

To install redundant instances and to specify the machine target or targets for the Apache HTTP Server, use the [httpproxy] host group in the inventory file. For more information, see [“Assign the Target Machines to Host Groups” on page 35](#). If you install Apache httpd before starting the deployment process, specify any machines where you have installed it for the [httpproxy] host group so that the deployment can add required software to them. However, because the Apache HTTP Server is required for internal communications among SAS Viya components, do not replace the Apache components that are installed by the playbook.

The Apache HTTP Server (the reverse proxy server) must be dedicated to a single SAS Viya deployment.

Data Source and Storage Requirements

Supported Data Sources

SAS Visual Investigator supports the following external data sources:

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

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

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

Data Encoding Requirement

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

You can also use SAS/CONNECT to transfer and automatically convert data from a non-UTF-8 encoded SAS session to the UTF-8 encoded SAS Viya environment. For information about how to convert data from non-UTF-8 to UTF-8, see [Access to SAS 9.4 Data](#).

Requirements to Transfer Data from SAS 9.4

SAS/CONNECT is required in the environment in order to transfer data from other SAS deployments and operating systems to SAS Viya. SAS/CONNECT can convert data from a non-UTF-8 encoded SAS session to the UTF-8 format that SAS Viya requires.

SAS/CONNECT is not included with a standard SAS Viya order, and must be separately licensed.

Requirements for SAS/ACCESS Interface to DB2

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

For information about supported IBM DB2 versions and requirements, see:

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html>.

Requirements for SAS/ACCESS Interface to ODBC

SAS/ACCESS Interface to ODBC (on SAS Viya) is used by SAS Visual Investigator to enable access to Microsoft SQL Server and MySQL data sources by means of a generic ODBC driver. SAS/ACCESS Interface to ODBC includes SAS Data Connector to ODBC.

For information about ODBC support for Microsoft SQL Server and MySQL, see:

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html>.

Requirements for SAS/ACCESS Interface to Oracle

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

You must install the Oracle client on the CAS controller.

For information about supported Oracle versions and requirements, see:

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html>.

Requirements for SAS/ACCESS Interface to PostgreSQL

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

For information about supported PostgreSQL versions and requirements, see:

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html>.

Requirements for SAS/ACCESS Interface to Teradata

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

For information about supported Teradata versions and requirements, see:

<https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html>.

User and Group Requirements

Set Up the Account that Deploys the Software

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

- Super user (sudo) or root access.

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

```
sudo -v
```

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

```
sudo -l
```

- (Optional) Root privileges, to run the tenant onboarding scripts. If you plan to deploy multiple tenants, contact your IT department to discuss the requirement to complete some deployment steps as the root user.
- Appropriate permissions to create subdirectories in the `/sas/install/sas viya playbook` directory path. Instructions for creating this directory are provided.
- A home directory that is readable by the user accounts (cas and sas) that are required for the deployment. Instructions for configuring the cas account are provided. The sas account is created automatically. If you have more than one machine target for SAS Infrastructure Data Server, make sure that all these machines have the same home directory for the installation user account.
- A subdirectory for the installation user's SSH keys: `~/ .ssh`.

Additional User Account Requirements

Perform the following steps before you start the deployment:

- ☐ Create a user group named `sas`. This group is required for the deployment.
- ☐ Create user accounts as described in the following table. The administrator account is referred to as `viadmin` in this document.
- ☐ Verify that all user accounts use the same UID, and all groups use the same GID on all machines in your deployment. If you need to modify the UIDs of any mismatched user accounts, use the `usermod` command. For groups with mismatched GIDs, use the `groupmod` command.
- ☐ 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 following table.

The following table identifies and describes the required user accounts. Because these accounts are used for the installation and for running services during the product's normal operation, do not delete them or change their names after they have been created. These user accounts do not require root or sudo privileges.

Table 2.3 Required User Accounts

Account Name and Group	Description	Purpose
sas; member of sas group	<p>A non-logon service account without user restrictions.</p> <p>No password. You can add a password after installation, if necessary, but make sure that it does not expire.</p> <p>The default user name is required.</p> <p>The sas group is an administration group, not a general user group.</p>	<p>Required for the installation, and created automatically.</p> <p>The installation process sets user and group ownership permissions on all the installation files. This user must exist to enable ownership.</p> <p>After the installation has completed, this user account enables the required components to run.</p> <p>The sas group is intended to allow access to administrative features, such as logs and backup. It is the group owner of many files on disk. Restrict membership in this group to administrators.</p>
cas; member of sas group	<p>The process owner of CAS processes. No default password is assigned, but a password is required if you plan to use this account as the CAS administrator. If you are using both local and LDAP accounts in your deployment, user credentials must match.</p> <p>Must be able to connect from the CAS controller 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.</p> <p>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.</p>	<p>Required for managing and enabling CAS. Create this user account and add it to the sas group before you start the deployment. Verify that the sas group is this user's primary group.</p> <p>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 Admin user in the vars.yml file before running the playbook.</p>
viadmin	<p>Administrative user account. Requires a valid LDAP user ID.</p> <p>You can assign another user name to this account, if desired.</p> <p>You must configure a user name and password for this user ID in the sitedefault.yml file. For more information, see “Create the sitedefault.yml File” on page 50.</p> <p>If you want to use this same user account as the CAS Admin user, configure viadmin for the casenv_admin_user in vars.yml. Both of these files are discussed in depth in the “Installing SAS Viya with Ansible” chapter.</p>	<p>The first user account that signs on to SAS Visual Investigator after the deployment has completed.</p>

To ensure access to all SAS Visual Investigator user interfaces, create the casenv_admin_user as a local operating-system account, and assign the same password as the LDAP password.

When the deployment has completed, the administrator creates a group for users and a group for administrators. The administrator must add the viadmin account to the sviadms group. If you have set up a superuser account for the CAS Admin user, you might want to add that user to the sviadms group as well.

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

- apache
- postgres

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

Security Requirements

LDAP Requirements

SAS Visual Investigator on Linux supports LDAP for user authentication. Microsoft Active Directory and OpenLDAP are supported LDAP implementations. In addition, the CAS server uses OAuth tokens for all clients in your deployment. LDAPS is supported, but the required certificates are not configured automatically by the deployment process.

Before SAS Visual Investigator is deployed, do the following:

- Set up tenants in LDAP. Use the documentation that is appropriate for your LDAP implementation.
Note: If you plan to deploy multiple tenants, all tenants must be contained within the same LDAP server. Multiple LDAP servers are not supported.
- Edit the `sitedefault.yml` file to enable the playbook to automatically configure the LDAP identity provider for OAuth to use. For more information, see [“Create the `sitedefault.yml` File” on page 50](#).

The deployment process configures some LDAP settings automatically. After the deployment has completed, 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, and can then set up user accounts for valid LDAP users. For more information, see [“Completing the Deployment” on page 83](#).

Enhance Default Security Settings

By default, SAS is deployed with secure HTTP (HTTPS) for many network connections. These connections are secured by a Public Key Infrastructure (PKI) based on Hashicorp Vault, which is configured by SAS. The certificates are all signed by a Vault-generated root CA and intermediate.

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

SAS recommends that you update the Apache `httpd` instances in your environment to use certificates that comply with the security policies at your enterprise before you start the deployment process. If you then specify the location of the intermediate certificates and the root CA, the playbook can distribute your certificates to all appropriate machines in your environment. For more information, see [“Specify the Path to Certificates” on page 39](#).

You also have the option to let the playbook install Apache `httpd` and to set up your Apache HTTP Server automatically, with default security settings and self-signed certificates. In that case, you can add your own certificates after the deployment has completed, which will require a brief outage.

If you instead keep the default security settings and certificates, end-users will see a standard web browser warning message. In this case, SAS recommends instructing end-users to click through this warning message until you have replaced the certificates.

For more information about setting up the Apache HTTP Server and configuring additional security settings, see [Encryption in SAS Viya: Data in Motion](#).

You can also take the following optional actions after the playbook has been run:

- Block external connections to port 80.
- Use HTTPS for access to SAS Viya user interfaces from a web browser.
- Add custom certificates to the self-signed certificates that the deployment provides on all machines.
- Upgrade the security protocol and ciphers that are enabled by default using the `sas-ssl.conf` file.

Transport Layer Security Requirements for the SAS Embedded Process

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

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

Client Requirements

Web Browser 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.

For information about supported web browsers to access SAS Visual Investigator user interfaces, see <https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-web-browsers.html>.

Database Drivers

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

Deployment Tools

Ansible Controller Requirements

Ansible is required to install SAS Visual Investigator.

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

A typical Ansible deployment consists of at least one control machine (the Ansible controller) and multiple Ansible managed nodes (the machines where SAS software is installed). In a single-machine deployment, Ansible and all SAS software are installed on the Ansible controller. For more information, see [“Install Ansible” on page 25](#).

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 22](#).

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

Pre-Installation Tasks

<i>Make Sure That You Have the Required Files</i>	19
<i>Configure SELinux</i>	20
<i>Enable Required Ports</i>	20
<i>Firewall Considerations</i>	22
<i>Configure Use of a Proxy Server</i>	24
Overview	24
Using curl	24
Using yum	24
<i>Enable the Yum Cache</i>	24
<i>Enable a Shared File System</i>	25
<i>Install Ansible</i>	25
Standard Ansible Installation	25
Streamlined Ansible Installation	25
Test Your Ansible Installation	26
<i>(Optional) Enable Key-Based SSH Authentication</i>	27
<i>Perform Linux Tuning</i>	27
Set the MaxStartups Value	27
Set the ulimit Values	28
Set the Semaphore Values	29
Change the Default Time-outs	29
Tune Memory Capacity	30
<i>Verify Wildcard DNS Alias</i>	30
<i>Install the Database Client</i>	30

Make Sure That You Have the Required Files

- 1 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 also directs you to create a playbook with the SAS Orchestration CLI. If you have not already created a playbook, create it now using the readme file from the download site.
- 2 The playbook that you created, `SAS_Viya_playbook.tgz`, should be placed in a directory on your Ansible controller that is able to be read by other users. The recommended location is `/sas/install`. If you have not already saved this file to such a location, save it now.
- 3 In the same directory where you have saved the playbook, uncompress it.

```
tar xf SAS_Viya_playbook.tgz
```

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	Requires Allowed Inbound Traffic From	Notes
CAS Communicator Port	0	SAS Viya Servers only	
httpd	80 (internal) 443 (external)	anywhere (SAS Viya servers, workstation)	
SAS Event Stream Manager agent	2552	ESP servers only	Required only if your order included SAS Event Stream Manager.
default Erlang Port Mapper Daemon (epmd) port	4369	SAS Viya Servers only	
SAS Infrastructure Data Server	5430–5439	SAS Viya Servers only	For a single server deployment with no failover, ports 5430-5432 must be opened. Additional standby nodes each get the next available port number sequentially up to 5439.
CAS Server Starting Port	5570	SAS Viya Servers and workstations	Used by clients to make binary connections to CAS.
default SAS Messaging Broker AMQP client access port	5672	SAS Viya Servers only	

Process	Required Port	Requires Allowed Inbound Traffic From	Notes
SAS Studio	7080 (if you are performing a full deployment, the deployment will use ephemeral ports, so no port needs to be reserved)	SAS Viya Servers only	Not required for SAS Visual Investigator.
Vault	8200	SAS Viya Servers only	
SAS Configuration Server	8300–8309, 8500, 8501	SAS Viya Servers only	SAS uses HashiCorp Consul as its configuration server. Ports should be open to both UDP and TCP traffic.
Object Spawner	8591		
CAS Server Monitor	8777	anywhere (SAS Viya servers)	Used by clients to make REST HTTP calls to CAS, as with the Python REST interface.
default SAS Messaging Broker management web console port	15672	SAS Viya Servers only	
SAS/CONNECT Spawner management	17541	anywhere (SAS Viya servers, SAS 9.X servers, workstation)	
SAS/CONNECT Spawner	17551	anywhere (SAS Viya servers, SAS 9.X servers, workstation)	
SAS Model Manager launcher context	18201–18250	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
SAS Job Execution launcher context	18501–18600	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
SAS Visual Forecasting launcher context	18601–19000	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
SAS Cloud Analytic Services Server	19990-19999	SAS Viya Servers only	

Process	Required Port	Requires Allowed Inbound Traffic From	Notes
default SAS Messaging Broker clustering port	25672	SAS Viya Servers only	

If your order included SAS Event Stream Processing, any ports that will be used for ESP servers must be open to HTTP traffic. For more information, see [Using the ESP Server](#).

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 in the table from the ports that can be allocated from within the ephemeral port range.

- 1 To determine the active ephemeral port range, run the following command on your host:

```
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 from the table that are included in your ephemeral port range to the Linux system reserved ports list. Add ports to the reserved list as comma-separated values or as a range within quotation marks:

```
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>

SAS recommends that you confirm access by running the curl command on the machine to which you will be downloading software. Run the following command from the playbook subdirectory (`/sas/install/sas_viya_playbook` if you used the recommended location for uncompressing your playbook).

```
curl -Lv --cert ./entitlement_certificate.pem --cacert ./SAS_CA_Certificate.pem
--head yum-repository-URL
```

Note: The preceding list contains values for *yum-repository-URL*.

Your system administrator can help you determine whether the command result indicates that you can successfully reach the repositories.

- 3 Determine whether the iptables or firewalld program is running.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl list-unit-files
```

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

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

Note: To identify the version of Linux that you are using, Red Hat Enterprise Linux and Oracle Linux users should see the `/etc/redhat-release` file. CentOS users should see the `/etc/centos-release` file.

- 4 To stop iptables, run the following commands.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl stop iptables.service
sudo systemctl disable iptables.service
sudo systemctl stop ip6tables.service
sudo systemctl disable ip6tables.service
```

To stop firewalld, run the following commands.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl stop firewalld.service
sudo systemctl disable firewalld.service
```

Configure Use of a Proxy Server

Overview

The SAS Viya deployment process uses both yum and curl to download RPM packages from SAS repositories. If your organization uses a forward HTTP proxy server, configure each target deployment machine to use the proxy server as appropriate. Refer to the man pages for yum.conf and curl for more information about proxy settings.

In addition, ensure Viya-to-Viya HTTP requests are not routed through the proxy during deployment by adding the IP addresses, hostnames, or domains for the Viya machines to the no_proxy variable. For example, if the Viya machines are using IP addresses 10.255.47.131 and 10.255.47.132 and hostnames machine1.example.com and machine2.example.com, you can configure no_proxy with the following command:

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

Using curl

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

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

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

Using yum

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

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

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.

Enable a Shared File System

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

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

Note: Multiple lines are used to improve readability. However, in your environment, make sure that you enter the command on a single line.

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

The shared file system is mounted for the CAS server and for the tenant in a single-tenant deployment or for the provider in a multi-tenant deployment. For more information about adding tenants, see [Multi-tenancy: Initial Tasks](#).

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, you must install a supported version of Ansible.

Standard Ansible Installation

The Ansible installation process is documented at http://docs.ansible.com/ansible/latest/intro_installation.html. You should always follow the Ansible documentation and choose the installation method that works best for your IT environment.

Streamlined Ansible Installation

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

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

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

```
## find out which release (6 or 7)
if grep -q -i "release 6" /etc/redhat-release ; then
    majversion=6
elif grep -q -i "release 7" /etc/redhat-release ; then
    majversion=7
else
    echo "Apparently, running neither release 6.x nor 7.x "
fi
## Attach EPEL
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-$majversion.noarch.rpm
# Display the available repositories
sudo yum repolist
```

- 2 To Install Python PIP and related packages:

```
sudo yum install -y python python-setuptools python-devel openssl-devel
sudo yum install -y python-pip gcc wget automake libffi-devel python-six
```

- 3 Since EPEL will no longer be needed, you can remove it with the following command:

```
sudo yum remove -y epel-release
```

- 4 Upgrade PIP and setuptools with the following command based on the version of Linux you are running.

For Red Hat Enterprise Linux 6.7 (and later within 6.x) or an equivalent distribution:

```
sudo pip install --upgrade pip
```

For Red Hat Enterprise Linux 7.1 (and later within 7.x) or an equivalent distribution:

```
sudo pip install --upgrade pip setuptools
```

- 5 To install a specific version of Ansible through PIP:

```
sudo pip install ansible==2.3.2
```

Test Your Ansible Installation

- 1 To test the Ansible version:

```
ansible --version
```

Here is an example of successful output:

```
ansible 2.3.2.0
config file =
configured module search path = Default w/o overrides
python version = 2.7.5 (default, May 3 2017, 07:55:04) [GCC 4.8.5 20150623 (Red Hat 4.8.5-14)]
```

- 2 To perform a basic ping test:

```
ansible localhost -m ping
```

Here is an example of successful output:

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

(Optional) Enable Key-Based SSH Authentication

Note: Even though key-based SSH authentication is optional, it is recommended.

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

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

Note: These steps assume that the `PasswordAuthentication` keyword has been enabled in the SSH daemon configuration file. It is also assumed that the user has a password that can be used for `ssh-copy-id` authentication.

- 1 Create an SSH key pair without a passphrase. The following example specifies the RSA key type. However, you can specify any key type that is supported by your SSH installation. Refer to the `ssh-keygen` man page for more information.

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

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

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

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

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

Perform Linux Tuning

Set the MaxStartups Value

The `MaxStartups` variable specifies the maximum number of concurrent connections that are available to the machine. If you expect a large number of users, you should increase the `MaxStartups` value on each SAS Cloud Analytics Server (CAS) machine (controller and any workers) as follows.

- 1 Open the `/etc/ssh/sshd_config` file.
- 2 Ensure that the value for the `MaxStartups` variable is 100.

```
MaxStartups 10:30:100
```

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

Set the ulimit Values

Overview

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

- open file descriptors
- stack size
- processes available to a user ID

Each resource type with limits is stored in the appropriate file on each machine in your deployment.

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

```
*      -      nofile      value
```

The asterisk (*) indicates all user accounts.

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

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

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

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

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

Set the Maximum Number of Open File Descriptors, Stack Size, and Address Space Limit

For each machine in your deployment:

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

2 Set the limit for open file descriptors as follows:

- If PostgreSQL will be deployed on the machine, set the limit (using the nofile item) to 150000 for the sas user.

```
sas      -      nofile      150000
```

- If the machine is running Elasticsearch, set the limit to at least 65536 for the sas user.

```
sas      -      nofile      65536
```

- For all other machines in the deployment, set the limit for the sas account, the cas account, and any other account that will be used to run a CAS session, including the root user, to at least 48000.

```
*      -      nofile      48000
```

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

```
*      -      nofile      150000
```

3 Set the address space limit as unlimited:

```
*      -      as      unlimited
```

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


```
sas      -      stack      10240
```

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

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

Set the Maximum Number of Processes Available

For each machine in your deployment:

- 1 Open the appropriate `*-nproc.conf` file. 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`

- 2 Set the limit for the number of processes as follows:

- If PostgreSQL will be deployed on the machine, set the limit (using the `nproc` item) to 100000 for the `sas` user.

```
sas      -      nproc      100000
```

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

```
*        -      nproc      65536
```

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

```
*        -      nproc      100000
```

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

Set the Semaphore Values

Repeat these steps for each machine in your deployment.

- 1 Open the `/etc/sysctl.conf` file.
- 2 Add the following lines or modify existing values as follows:

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

- 3 To enable `httpd` to start in multi-tenant deployments, add the following line at the end of the file or modify existing values as follows:

```
kernel.msgmni=value-greater-than-or-equal-to-1024
```

- 4 Save and close the `/etc/sysctl.conf` file.
- 5 Refresh the revised settings from the `/etc/sysctl.conf` file:

```
sudo sysctl -p
```

Change the Default Time-outs

Note: The information in this section applies only to systems running Red Hat Enterprise Linux 7.1 or later or equivalent distributions. If you are using a Linux distribution earlier than Red Hat Enterprise Linux 7.1, you should skip this section.

To change the default time-out values:

- 1 Open the `/etc/systemd/system.conf` file.
- 2 Find the two variables that control time-outs: `DefaultTimeoutStartSec` and `DefaultTimeoutStopSec`.
- 3 If the lines that contain these variables are not already uncommented, uncomment each line by removing the number sign (#).
- 4 Assign both the `DefaultTimeoutStartSec` and `DefaultTimeoutStopSec` variables a value of `1800s`.

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

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

Tune Memory Capacity

This section is applicable only if you are using virtual machines (VMs) in your deployment. If you are not using VMs, skip this section.

On each VM that will be running Elasticsearch:

- 1 Open the `/etc/sysctl.conf` file.
- 2 Add the following lines to increase the maximum virtual memory:

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

- 3 Save and close the `/etc/sysctl.conf` file.
- 4 Refresh the settings from the `/etc/sysctl.conf` file.

```
sudo sysctl -p
```

Verify Wildcard DNS Alias

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

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

Install the Database Client

The PostgreSQL database requires client software.

Install the client named `psql` by running the following command:

```
sudo yum install postgresql
```

Note: In a multi-machine environment, run the command on each machine where PostgreSQL is installed.

Installing SAS Viya with Ansible

Modify the Initial Deployment	32
Use a Mirror Repository	32
Edit the Inventory File	32
Overview	32
Single-Machine Deployment	33
Multiple-Machine Deployment	33
Modify the vars.yml File	37
Set the Deployment Label	37
Set the Pre-deployment Validation Parameters	37
Specify Security Settings	39
Specify the Path to Certificates	39
(Optional) Specify JRE	40
Set Up Passwordless SSH for CAS	40
Install Consul on CAS Hosts	42
Define the CAS User Group	42
Set Up the CAS Admin User	43
Add Data Source Information	43
Modify the LD_LIBRARY_PATH	47
Set the CAS Cache Directory	47
Set Up HDFS and Co-location	49
Create the sitedefault.yml File	50
Revise Elasticsearch File	53
Determine If You Are Affected	53
Modify the start.yml File	54
Deploy the Software	54
Assessment Test	54
Commands	54
Run from a Directory Other than the Default	55
Successful Playbook Execution	55
Retry a Failed Deployment	55
Install with SAS 9.4 Software	55
Deployment Logs	56

Modify the Initial Deployment

This chapter describes the initial deployment of your SAS Viya software only. For information about modifying an existing deployment with updated software or adding new software to an existing deployment, see [“Managing Your Software” on page 85](#).

Use a Mirror Repository

By default, SAS downloads and installs the latest software available from the software repositories. If your deployment does not have access to the internet or if you must always deploy the same version of software (such as for regulatory reasons), you should create and use a mirror repository for deployment. For details about creating and using mirror repositories, see [“Creating and Using Mirror Repositories” on page 111](#).

Edit the Inventory File

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

Overview

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

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

Each inventory file consists of two parts:

deployment target definition

A specification of each machine on which SAS Viya software will be deployed.

host group assignment list

A mapping of the installable groups of software and the machines on which they will be deployed. SAS Viya software is deployed as host groups, which are identified by square brackets ([]) in the inventory file. Each host group is preceded by comments that describe the purpose of the software in the host group. The user specifies the machines on which a host group will be deployed by listing them under the host group name. A machine can have more than one host group deployed on it.

Here is an example of a host group assignment list:

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

deployTarget2

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

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

Single-Machine Deployment

This section is applicable only if you are performing a single-machine deployment. If you are performing a multi-machine deployment, you should skip this section and go to [“Multiple-Machine Deployment” on page 33](#).

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

```
cp sample-inventories/inventory_local.ini inventory.ini
```

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

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

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

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

For a local machine:

```
deployTarget ansible_connection=local consul_bind_adapter=eth0
```

For a remote machine:

```
deployTarget ansible_host=host1.example.com consul_bind_adapter=eth0
```

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

Multiple-Machine Deployment

Specify the Machines in the Deployment

If you are performing a single-machine deployment, you should skip this section and go to [“Single-Machine Deployment” on page 33](#).

The first section in the `inventory.ini` 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 is located at the beginning of the `inventory.ini` file.

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

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

Table 4.1 *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_host	the IP address of the remote machine.
ansible_user	the user ID that is used by Ansible to connect to each of the remote machines and to run the deployment.
ansible_ssh_private_key_file	the private key file that corresponds to the public key that was previously installed on each of the remote machines. This file typically resides in your <code>~/ .ssh</code> directory.

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 multi-machine deployment:

```
main ansible_host=host1.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
controller ansible_host=host2.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
worker-1 ansible_host=host3.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
worker-2 ansible_host=host4.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
postgres ansible_host=host5.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
sas-elastic-master-1 ansible_host=host6.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
sas-elastic-master-2 ansible_host=host7.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
sas-elastic-client-1 ansible_host=host8.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
sas-elastic-client-2 ansible_host=host9.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
sas-elastic-data-1 ansible_host=host10.example.com ansible_user=user1
    ansible_ssh_private_key_file=~/.ssh/id_rsa
sas-elastic-data-2 ansible_host=host11.example.com ansible_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.

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

```
main ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa consul_bind_adapter=eth0
```

Assign the Target Machines to Host Groups

The second section in the inventory file is used to assign deployment targets to each host group. Under each group, assign machines to the group by using the appropriate alias. Here is a typical assignment that uses the machines from the preceding example.

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

```
[CoreServices]
main

[configuratr]
main

[consul]
sas-elastic-client-1
sas-elastic-client-2

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

[httpproxy]
main

[pgpoolc]
postgres

[rabbitmq]
main

[sasdatasvrc]
postgres

[sviconfig]
sas-elastic-client-1

[tenant]
main

[sas-casserver-primary]
controller

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

```

worker-2

[viprCommon]
main

[viprESM]
main

[viprEntity]
main

[viprSand]
main

[viprVi]
main

[viprVsd]
main

[sas-all:children]
CoreServices
configuratr
consul
elasticsearch
httpproxy
pgpoolc
rabbitmq
sasdatasvrc
sviconfig
tenant
sas-casserver-primary
sas-casserver-worker
viprCommon
viprESM
viprEntity
viprSand
viprVi
viprVsd

```

Consider the following issues when editing the inventory file:

- SAS recommends that you do not remove any host groups from the list or any entries from the [sas-all:children] list unless you are an experienced Ansible user. A host group can have no entries under it, but the host group should not be removed even if it is empty. Removing a host group that contains targeted machines from the [sas-all:children] list can result in critical tasks not being executed on those targeted machines.
- Note that the entry for [elasticsearch] contains information not included with the other host groups. For more information, see the comment preceding [elasticsearch] in the inventory file.
- If you are installing Elasticsearch on a single machine, that machine must be included in the [elasticsearch], [sviconfig], and [consul] host groups. If you are installing Elasticsearch on more than one machine, each Elasticsearch client node must be included in the [consul] host group, and at least one of the client nodes must be included in the [sviconfig] host group.
- SAS recommends that you do not put Elasticsearch, CAS, and PostgreSQL on the same machine.

- If you are using HDFS that is co-located with CAS, then [sas-casserver-primary] and [sas-casserver-worker] should be assigned to machines in the Hadoop cluster.
- [sas-casserver-primary] supports a single instance of the CAS controller node (sometimes referred to as the CAS server) per tenant. The first host in the list is used by the tenant in a single-tenant deployment or the provider in a multi-tenant deployment. If you change the first host, you change the tenant in a single-tenant deployment or the provider in a multi-tenant deployment. Any additional hosts in the list are used in a multi-tenant environment. The *tenant-vars.yml* file determines the hosts' configuration for the tenants (primary, secondary, or worker).

Note: For more information about the *tenant-vars.yml* file, see [Multi-tenancy: Initial Tasks](#).

- [sas-casserver-secondary] supports a single instance of the CAS backup controller for the tenant in a single-tenant deployment or the provider in a multi-tenant deployment. This host group is used only by the tenant in a single-tenant deployment or the provider in a multi-tenant deployment.
- [sas-casserver-worker] provides worker nodes for the tenant in a single-tenant deployment or the provider in a multi-tenant deployment. This host group is used only by the tenant in a single-tenant deployment or the provider in a multi-tenant deployment.
- If the machines that you specify for [pgpoolc] or [sasdatasvrc] do not have an alias of deployTarget in the deployment target reference, you must open the *sas_viya_playbook/vars.yml* file and replace the instance of deployTarget under INVOCATION VARIABLES with the alias that you used in the deployment target reference:

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

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

Note: By default, your deployment includes a single-machine, single-node instance of HA PostgreSQL, used as the SAS Infrastructure Data Server. To deploy HA PostgreSQL with multiple nodes, see [“Creating High Availability PostgreSQL Clusters” on page 99](#).

Modify the vars.yml File

As its name suggests, the vars.yml file contains deployment variables that enable you to customize your deployment to meet your needs.

Set the Deployment Label

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

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

```
DEPLOYMENT_LABEL: "vi-04april2018"
```

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 16](#).

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 Verify that the `DEPLOYMENT_LABEL` variable has content and contains only lowercase alphabetic characters, numbers, and hyphens.

Note: For more information about the `DEPLOYMENT_LABEL` variable, see [“Set the Deployment Label” on page 37](#).

- 2 Verify that a CAS controller host is defined.

Note: For information about assigning software to machines, see [“Assign the Target Machines to Host Groups” on page 35](#).

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

- 4 Verify that each machine in the inventory file can successfully connect to every other machine in the inventory file.

Note: For more information about modifying the inventory file, see [“Specify the Machines in the Deployment” on page 33](#).

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

- 6 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 42](#).

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

Create and Verify sas User and sas Group

If the sas user and sas group do not already exist, the playbook creates the sas user and places it in the sas group. If 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 20](#).

- 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 Host Groups” on page 35](#).

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

Note: For more information about setting limits, see [“Set the ulimit Values” on page 28](#).

Specify Security Settings

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

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

By default, `SECURE_CONSUL` is set to `true` and `DISABLE_CONSUL_HTTP_PORT` is set to `false`. Both HTTP and HTTPS ports will be available after the software is deployed.

Specify the Path to Certificates

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

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

To set `HTTPD_CERT_PATH`:

- 1 Open the `vars.yml` file.
 - 2 Set the value of `HTTPD_CERT_PATH` based on the following conditions. Ensure that any value you use is enclosed in single quotation marks (').
- If your `SSLCertificateChainFile` contains the root certificate authority (CA) and all intermediate certificates, remove the existing value for `HTTPD_CERT_PATH`. Ensure that all browsers and clients have the root CA in their truststore.

Here is an example of the modified variable:

```
HTTPD_CERT_PATH:
```

- If your `SSLCertificateChainFile` contains the intermediate links but not the root CA, `HTTPD_CERT_PATH` should be the path to the file on the machine in the `[httpproxy]` host group in the inventory file that contains the root CA.
- If your `SSLCertificateChainFile` contains no certificates and no root CA, `HTTPD_CERT_PATH` should be the path to the file on the machine in the `[httpproxy]` host group in the inventory file that contains the

intermediate certificates and the root CA. Ensure that all the intermediate certificates are in the truststore of all browsers and clients.

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

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

- 3 Save and close the vars.yml file.

(Optional) Specify JRE

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

- 1 With a text editor, open the vars.yml file.
- 2 Set the value of `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” on page 10](#).

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 40](#) for more information about the default process.
- Use your own passwordless SSH. See [“Use Your Own Passwordless SSH” on page 41](#).
- Use the deployment process to create a customized passwordless SSH. Customization can include users other than the default. See [“Create Customized Passwordless SSH” on page 41](#).

Accept the Passwordless SSH Default

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

```
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` block.
- 3 The private and public keys are copied to each host that the playbook runs against.
- 4 The `ssh-keyscan` utility is run from each host to every other host in the CAS cluster.
- 5 The user's public key is added to the `~/.ssh/authorized_keys` file.

Use Your Own Passwordless SSH

If you choose to use your own passwordless SSH, you must set the `cas` user to be a user that you have already configured for passwordless SSH. For details, see [“Set Up the CAS Admin User” on page 43](#).

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

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

```
setup_sas_users: false
```

- 3 Save and close the `vars.yml` file.

Create Customized Passwordless SSH

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

Note: Use these steps to create new users. If you list an existing user in this section and use different values than what that user already has, the user will be modified to match the values in the `vars.yml` file.

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

Note: Comments have been removed from the following example.

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

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

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

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

- `setup_home` – uses the value of `true` or `false`. Determines whether the shell and home values should be used by the deployment. To accept the default, use a value of `false`.
 - `shell` – the location of the shell for the user account to use. It can be used only if `setup_home` is set to `true`.
 - `home` – the location of the user directory to be created. It can be used only if `setup_home` is set to `true`.
- c As an option, to install any packages to be defined under `extra_packages`, set `setup_packages` to `true`.
- d Under `extra_packages`, specify one or more names of any additional packages to install along with a comment that describes its purpose. The administrator typically uses this field to specify additional packages for the deployment (such as Firefox or Git) as a convenience. The field is ignored if `setup_packages` is set to `false`.

3 Save and close the `vars.yml` file.

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

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

Install Consul on CAS Hosts

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

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

Note: For more information about modifying the inventory file, see [“Edit the Inventory File” on page 32](#).

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 cas to be the CAS Admin user, perform the following steps:

- 1 Open the vars.yml file.
- 2 Uncomment the `casenv_admin_user` line. To uncomment, remove the number sign (#).
- 3 In that same field, insert the name of a user that exists and that can log on:

```
casenv_admin_user: valid-user
```

Note: The value for `casenv_admin_user` must be the same value that is used for the cas username in the `sitedefault.yml` file. For more information, see [Step 8 on page 52](#).

- 4 Save and close the vars.yml file.

When the deployment is complete, you should use this user to log on to CAS Server Monitor.

Note: This user must have a single set of credentials that are valid for all applicable authentication providers. In a full deployment, dual authentication occurs for logon to CAS Server Monitor and access to CAS from SAS Visual Investigator. For more information, see [SAS Viya Administration: Security](#).

Add Data Source Information

Overview of the Data Sources

If your software order includes one or more SAS/ACCESS products, you must edit the vars.yml file to include information that is needed to configure those products during deployment. Also, if you plan to use Hadoop Distributed File System (HDFS), you must also edit the vars.yml file.

SAS Viya uses the `sasenv_deployment` and `cas_settings` files to configure environment variables for the data sources. To create those files at deployment, add values to the `FOUNDATION_CONFIGURATION` and `CAS_SETTINGS` blocks of the vars.yml file before you run the playbook. The vars.yml file contains typical examples of these blocks, which are commented out with number signs (#). The following sections contain examples of these blocks that are appropriate for the specific SAS/ACCESS products. To customize the file, either uncomment the lines and edit the existing blocks or create new blocks using the example's format.

Note: If you start a new block, ensure that each line in the block begins with three spaces and a number. Each numbered line should reflect its numerical order within the block.

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

SAS/ACCESS Interface to DB2

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Uncomment the `FOUNDATION_CONFIGURATION` line. To uncomment, remove the number sign (#).
- 3 Under `FOUNDATION_CONFIGURATION`, add the following lines, including the spaces and the numerical prefixes.

```
1: CLASSPATH=$CLASSPATH:DB2-related-classpath
2: DB2INSTANCE=DB2-instance
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-DB2-install
```

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

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

```
1: DB2INSTANCE=DB2-instance
2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-DB2-install
```

- 6 Save and close the vars.yml file.

SAS/ACCESS Interface to ODBC

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.

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

For DataDirect:

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

For iODBC:

```
#FOUNDATION_CONFIGURATION:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINSTINI=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

For unixODBC:

```
#FOUNDATION_CONFIGURATION:
1: ODBC_SYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
2: ODBCINI=name-of-your-odbc.ini-file
3: ODBCINSTINI=name-of-your-odbcinst.ini-file
4: LD_LIBRARY_PATH=location-of-ODBC-driver-manager-library:$LD_LIBRARY_PATH
```

Note: For unixODBC, if ODBC_SYSINI is not set in your environment, ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 4 Under CAS_SETTINGS, add the following lines (including the indentions, spaces, and numerical prefixes), depending on the version of ODBC that you are using:

For DataDirect:

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

For iODBC:

```
#CAS_SETTINGS:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINSTINI=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

For unixODBC:


```
#CAS_SETTINGS:
1: ODBCYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
2: ODBCINI=name-of-your-odbc.ini-file
3: ODBCINSTINI=name-of-your-odbcinst.ini-file
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

Note: For unixODBC, if ODBCYSINI is not set in your environment, ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

- 5 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).
- 6 Save and close the vars.yml file.

SAS/ACCESS Interface to Oracle

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 3 Under FOUNDATION_CONFIGURATION, add the following lines (including the spaces and the numerical prefixes).

```
1: ORACLE_HOME=Oracle-home-directory
2: TWO_TASK=ORACLE_SID
3: ORAENV_ASK=NO
4: SASORA=V9
5: PATH=$PATH:$ORACLE_HOME/bin
6: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

- 4 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).
- 5 Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.

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

- 6 Save and close the vars.yml file.

SAS/ACCESS Interface to PostgreSQL

For the following steps, depending on how you have configured your PostgreSQL ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

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

On the SAS Client node:

- a In the directory where you want to create the odbcinst.ini file, run the following command to create and open a new file.

```
sudo vi odbcinst.ini
```

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

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

- c Save and close the odbcinst.ini file.

On the CAS nodes:

- a In the directory where you want to create the odbcinst.ini file, run the following command to create and open a new file.

```
sudo vi odbcinst.ini
```

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

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

- c Save and close the odbcinst.ini file.

- 2 Open the vars.yml file.

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

- 4 Under FOUNDATION_CONFIGURATION, add the following lines (including the spaces and the numerical prefixes).

```
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: PGCLIENTENCODING=encoding-for-the-PostgreSQL-client
4: PATH=$PATH:path-to-PostgreSQL-bulk-loading
5: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-PostgreSQL-client
```

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

- 6 Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.

```
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: PGCLIENTENCODING=encoding-for-the-PostgreSQL-client
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-PostgreSQL-client
```

- 7 Save and close the vars.yml file.

SAS/ACCESS Interface to Teradata and SAS In-Database Technologies for Teradata

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

- 2 On the CAS nodes, and the SAS client node (if you set the encoding to UTF-8), ensure that the following two lines are in the clispb.dat file.

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

- 3 Open the vars.yml file.

- 4 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).

- 5 Under FOUNDATION_CONFIGURATION, add the following lines, including the spaces and the numerical prefixes.

Note: Multiple lines are used for LD_LIBRARY_PATH to improve readability. However, in your environment, make sure that you enter the command on a single line.

```

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

```

Here is an example of the TTU Default LD_LIBRARY_PATH:

```

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

```

- 6 Uncomment the CAS_SETTINGS line. To uncomment the line, remove the number sign (#).
- 7 Under CAS_SETTINGS, add the following lines, including the spaces and the numerical prefixes.

Note: Multiple lines are used for LD_LIBRARY_PATH to improve readability. However, in your environment, make sure that you enter the command on a single line.

```

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

```

Here is an example of the TTU Default LD_LIBRARY_PATH:

```

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

```

- 8 Save and close the vars.yml file.

Modify the LD_LIBRARY_PATH

Note: If you have performed the steps in “SAS/ACCESS Interface to Oracle” on page 45, you should skip this section.

- 1 Open the vars.yml file.
- 2 In the CAS Specific section, uncomment the CAS_SETTINGS line. To uncomment, remove the number sign (#).
- 3 Locate the following line:

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

- 4 Replace that line with the following:

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

- 5 Save and close the vars.yml file.

Set the CAS Cache Directory

Change the CAS Cache Directory

SAS Cloud Analytics Services (CAS) is the in-memory analytic server for SAS Viya. As a memory efficiency, CAS organizes in-memory data in blocks and memory maps the blocks. The blocks are stored as temporary files in directories on the host.

By default, only the /tmp directory is used as the cache directory. This is sufficient for demonstration purposes, but not for production use of the server.

For a production-use server, set the cache to use a series of directories. The size required differs for each deployment, but can run from gigabytes to terabytes. When you specify a series of directories, each time the server needs to use disk, it uses the next path in the list. This strategy is used to distribute the load across disk volumes.

To change the CAS cache:

- 1 Open the vars.yml file.
- 2 In the CAS_CONFIGURATION section, uncomment the line that contains the CAS_DISK_CACHE variable. To uncomment, remove the number sign (#).
- 3 Remove the /tmp value from the variable and replace it with the directory that you want to use as the CAS cache. If you want to use more than one directory, list them all with colons separating the directories. For example:

```
CAS_CONFIGURATION:
  env:
    CAS_DISK_CACHE: /disk1:/disk2:/disk3
```

Note: To avoid the potential for problems caused by the CAS disk cache that can fill up the root file system, do not specify any directory that is on the same partition or logical volume as the root file system.

SAS recommends that you create directories dedicated to caching that are owned by the ID that executes the CAS server (`cas` by default). Each directory should be set up identically on each CAS node. All CAS processes must have Read, Write, and Execute permissions for these directories. Therefore, permissions must be granted to the server's ID and the ID of any CAS user that connects through programming interfaces like SAS and Python.

- 4 Save and close the vars.yml file.

Change the CAS Cache Directory

Here are some tuning tips for the CAS cache directory:

- Use the EXT4 file system or the XFS file system, and configure each disk device as a separate file system. For hosts with eight or more disk devices, dedicate one device for file system journals. When you create the file systems, specify the dedicated device as the external journal.
- **Note:** If you can predict that the total size of the tables in CAS_DISK_CACHE is less than the available RAM, you can set CAS_DISK_CACHE to `/dev/shm` rather than to a disk file system.
- The `noatime` and `nodiratime` mount options are applicable if no other data on the file system prevents the use of these mount options. If appropriate for your power supply, the `nobarrier` mount option might be applicable. Increasing the read-ahead value might improve performance. Refer to the Linux documentation for more information about these mount options.
- Reducing the aggressiveness to swap memory pages can improve performance:


```
sudo sysctl -w vm.swappiness=1
```
- In addition to creating multiple file systems, you should create each file system with multiple directories to avoid contention by multiple threads. The total number of directories that is assigned to CAS_DISK_CACHE should be at least two times the number of CPUs on the host that are licensed for CAS. Include additional CPUs in the total CPU count to accommodate the Intel Hyper-Threading Technology that is used to support multiple threads. Also, try to distribute the directories across the file systems.

Table 4.2 Sample Configurations

CPU and Disk Count	Suggested Configuration
32 CPUs, 16 disks	<ul style="list-style-type: none"> ■ Use one disk device for the file system journal. ■ Create 15 file systems, specifying the dedicated device for the journal. ■ 32 CPUs × 2 = 64 directories. 64 ÷ 15 file systems rounds up to 5 directories on each file system.
32 CPUs, 24 disks	<ul style="list-style-type: none"> ■ Use one disk device for the file system journal. ■ Create 23 file systems, specifying the dedicated device for the journal. ■ 32 CPUs × 2 = 64 directories. 64 ÷ 23 file systems rounds up to 3 directories on each file system.
48 CPUs, 16 disks	<ul style="list-style-type: none"> ■ Use one disk device for the file system journal. ■ Create 15 file systems, specifying the dedicated device for the journal. ■ 48 CPUs × 2 = 96 directories. 96 ÷ 15 file systems rounds up to 7 directories on each file system.

Set Up HDFS and Co-location

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

```
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    #HADOOP_NAMENODE: 127.0.0.1
    #HADOOP_HOME: /opt/vendor
  cfg:
    #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 Uncomment the HADOOP_NAMENODE and HADOOP_HOME lines. To uncomment, remove the number sign (#).
- 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 co-located 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 Host Groups” on page 35](#).

Create the sitedefault.yml File

To configure LDAP/Active Directory to be used by OAuth, you must create and modify the sitedefault.yml file located at `sas_viya_playbook/roles/consul/files`.

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

```

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

```

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

- 2 Open the sitedefault.yml file:

```

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

```

It contains the following fields:

Note: Comments have been removed to improve readability.

```

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

```

```

    baseDN:
  sas.identities:
    administrator:

```

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

- 4 Specify site-specific default values, as appropriate:

- The comments in the file provide guidance about the values to specify.
- Enclose each value in single quotation marks.
- Specify the highlighted line exactly as shown.
- Maintain the indentation as shown.

```

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

```

- 5 To override the default setting from the identities service, enter the appropriate value for your LDAP server. For example, following the `sas.identities.providers.ldap.profile` block, using the same indentation level, add a new block for `sas.identities.providers.ldap.tenancy`.

```

config:
  application:
    ...
    sas.identities.providers.ldap.profile:
      file: 'ldap/ldap-search-and-bind.xml'
    sas.identities.providers.ldap.tenancy:
      userRdn: 'ou=users'
      groupRdn: 'ou=groups'

```

To clear the default and have no relative distinguished name settings, use this block in the same location.

```

config:
  application:
    ...
    sas.identities.providers.ldap.profile:
      file: 'ldap/ldap-search-and-bind.xml'
    sas.identities.providers.ldap.tenancy:
      userRdn: ''
      groupRdn: ''

```

Consult with your LDAP administrator to determine what values you should use. For more information about LDAP distinguished names, see [LDAP DNs and RDNs](#).

- 6 Following the `sas.identities.providers.ldap.tenancy` block, using the same indentation level, add a new block for `sas`.

```
config:
  application:
    sas:
      multi:
        tenancy:
          enabled: 'true|false'
```

If you are performing a multi-tenant deployment, select the value `true` for `enabled`. Otherwise, select `false`.

- 7 Following the `sas` block, using the same indentation level, add a new block for `deployment`.

```
config:
  application:
    sas:
      ...
      deployment:
        highAvailabilityEnabled: 'true|false'
```

If you are performing a high availability (HA) deployment, select the value `true` for `highAvailabilityEnabled`. Otherwise, select `false`.

- 8 Following the `deployment` block, using the same indentation level, add a new block for `cas`.

```
config:
  application:
    deployment:
      ...
      cas:
        username: "CAS-administrator-user-ID"
        password: "CAS-administrator-password"
```

Note: The CAS administrator user ID that is used as the value for `username` must be the same user ID that is used for the `casenv_admin_user` in the `vars.yml` file. See [“Set Up the CAS Admin User” on page 43](#) for more information.

- 9 Following the `application` block, using the same indentation level, add a new block for `postgres`.

```
config:
  application:
    ...
    postgres:
      sas.dataserver.conf:
        common:
          max_connections: 'value-based-on-deployment-type'
      sas.dataserver.pool:
        common:
          num_init_children: 'pgpool-server-instances'
          max_pool: 'number-of-connections-to-each-server-instance'
```

The value for `max_connections` depends on which type of deployment you are performing. For a single-machine deployment, SAS recommends a value of 1000. For a multi-machine deployment, SAS recommends a value of 3000. You might need to increase these values over the lifetime of the deployment depending on performance.

Note: The value for `max_connections` must be greater than or equal to the value of `num_init_children` times the value of `max_pool`.

SAS recommends a value of 2 or 3 for `max_pool`.

Therefore, the value of `num_init_children` can be calculated as less than or equal to `max_connections` divided by `max_pool`.

10 (Optional) By default, TLS is enabled in a SAS Visual Investigator deployment. To disable TLS:

a Following the `cas` block, using the same indentation level, add a new block for `sas.security`:

```
config:
  application:
    cas:
    ...
    sas.security:
      network.web.enabled: 'false'
      network.sasData.enabled: 'false'
      network.databaseTraffic.enabled: 'false'
      network.serverControl.enabled: 'false'
```

b In the `vars.yml` file, set the `SECURE_CONSUL` variable to `false`. See [“Specify Security Settings” on page 39](#) for information about the `SECURE_CONSUL` variable.

11 Save and close the `sitedefault.yml` file.

Revise Elasticsearch File

Determine If You Are Affected

If you are deploying on Red Hat Enterprise Linux 6.x or an equivalent distribution, such as CentOS 6.x, use the instructions in [“Modify the start.yml File” on page 54](#) to modify the `start.yml` file.

If you are deploying on Oracle Linux 6.x, perform the following tests to determine whether you need to modify the `start.yml` file.

1 Verify the Linux kernel version by running the following command:

```
uname -r
```

2 The output will be the version of your Linux kernel. If the version is `3.5.x` or greater, continue to step 3. If the version is less than `3.5.x`, you must modify the `start.yml` file, detailed in [“Modify the start.yml File” on page 54](#).

3 Verify the presence of required kernel parameters by running the following command:

```
cat /boot/config-`uname -r` | grep SECCOMP
```

4 Successful output will look like this:

```
CONFIG_HAVE_ARCH_SECCOMP_FILTER=y
CONFIG_SECCOMP_FILTER=y
CONFIG_SECCOMP=y
```

If you see output like this, you do not need to revise the `start.yml` file. However, if you see no output or a message like the following:

```
# CONFIG_SECCOMP is not set
```

you should go to [“Modify the start.yml File” on page 54](#) to revise the start.yml file.

Modify the start.yml File

- 1 In the directory where you uncompressed your playbook, find and open the `/roles/elasticsearch/task/start.yml` file. If you used the recommended directory, the file will be in `/sas/install/roles/elasticsearch/task/start.yml`.
- 2 Between the `Execute elasticsearch_move_config_files` section and the `Execute elasticsearch_change_ownership` section, add the following code.

Note: Because of page width restrictions, the line in this code that begins with `command:` has been broken into two lines. When you insert the code into the file, it should be added as a single line.

```
- name: Execute elasticsearch_move_config_files
  ...

- name: Execute elasticsearch_add_bootstrap_disable
  become: 'yes'
  become_user: root
  command: "bash -c 'echo \"bootstrap.system_call_filter: false\" >>
  {{ SAS_CONFIG_ROOT }}/etc/elasticsearch/default/elasticsearch.yml'"
  tags:
  - config

- name: Execute elasticsearch_change_ownership
```

- 3 Save and close the start.yml file.

Deploy the Software

Assessment Test

Before you deploy the software, SAS recommends run the following command to assess the readiness of your system for deployment.

```
ansible-playbook system-assessment.yml
```

Fix any errors the system assessment uncovers before you run the deployment command.

Commands

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

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

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

Password Requirements	Command
Does not require passwords	<code>ansible-playbook site.yml</code>

Password Requirements	Command
Requires a sudo password only	<code>ansible-playbook site.yml --ask-become-pass</code>
Requires an SSH password only	<code>ansible-playbook site.yml --ask-pass</code>
Requires both a sudo and an SSH password	<code>ansible-playbook site.yml --ask-pass --ask-become-pass</code>

Run from a Directory Other than the Default

The Ansible playbook runs the commands from the top-level `sas_viya_playbook` directory, by default. If you want to run the playbook from another directory, modify the `ansible.cfg` configuration file with the appropriate SAS Viya configuration options. Refer to the Ansible documentation to find the appropriate `ansible.cfg` file and add those options.

Successful Playbook Execution

Here is an example of the output from a successful playbook execution:

```
PLAY RECAP *****
deployTarget          : ok=81   changed=65   unreachable=0   failed=0
```

The most important indicator of success from this message is `failed=0`, indicating zero failures.

Retry a Failed Deployment

If your deployment fails, and you are able to respond to the error message and can recover from the error, you must restart the deployment using the appropriate deployment commands and options.

Install with SAS 9.4 Software

SAS Viya software can be installed on the same machines as an existing SAS 9.4 deployment. No special steps need to be taken at deployment time.

During the deployment, the playbook might halt with an error indicating the ports that SAS Viya needs are in use by the SAS 9.4 deployment. If you receive that error, you should open the `vars.yml` file in a text editor and search for the variables for the ports that SAS Viya uses. The ports can be found in the `CAS_CONFIGURATION` section of the `vars.yml` file.

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

```
CAS_CONFIGURATION:
cfg:
  #gcport: 0
  #httpport: 8777
  #port: 5570
```

To change the value:

- 1 Uncomment the line with the port value to change. To uncomment, remove the number sign (#).
- 2 Change the port value to the port that you want to use. Here is an example:

```
CAS_CONFIGURATION:
  cfg:
    #gcport: 0
    httpport: 8778
    #port: 5570
```

- 3 Save and close the vars.yml file.
- 4 Deploy your software by running the Ansible playbook as you did initially.

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
```

Post-Installation Tasks

<i>Set the Password for the CAS Administrator or Another Administrative Account</i>	57
<i>Reset the PostgreSQL Passwords</i>	58
<i>Change the Administrative User Password for SAS Message Broker</i>	58
<i>Configure SAS/ACCESS Interface to DB2</i>	59
<i>Configure ODBC</i>	59
<i>Configure SAS/ACCESS Interface to Oracle</i>	60
<i>Configure SAS Data Connector to PostgreSQL</i>	60
<i>Configure SAS/ACCESS Interface to Teradata</i>	61
<i>Configure CAS For SAS Visual Investigator</i>	62
<i>Restart the SAS Visual Investigator Applications</i>	62
<i>Connect to PostgreSQL</i>	63
<i>Configure a Standard or Multi-tenant Deployment</i>	63
Overview	63
Configure a Standard Deployment	63
Configure a Multi-tenant Deployment	65
<i>Configure the First System User</i>	66

Set the Password for the CAS Administrator or Another Administrative Account

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

```
sudo passwd cas
```

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

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

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

Reset the PostgreSQL Passwords

SAS recommends that you reset the database password before you configure SAS Visual Investigator. The user ID for the database owner is dbmsowner.

- 1 Obtain the initial password by running the following commands:

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv read config/application/sas/database/postgres/password
```

- 2 Run the script in order to change passwords:

Note: You will be prompted for the user ID (dbmsowner) and the initial password.

Note: For multi-machine deployments, run the script on the machine on which you installed pgpool.

```
cd /opt/sas/viya/home/libexec/sasdatasvrc/script/
sudo -u sas ./sds_change_user_pw.sh -config_path
/opt/sas/viya/config/etc/sasdatasvrc/postgres/pgpool0/sds_env_var.sh
```

Change the Administrative User Password for SAS Message Broker

You must change the administrative user password for SAS Message Broker as soon as possible after you have deployed SAS Viya.

- 1 Locate a machine that you have previously assigned to the [rabbitmq] host group in the inventory file. This machine is the message broker machine.
- 2 Sign on to the message broker machine with sudo privileges.
- 3 Change to this directory:

```
/opt/sas/viya/home/bin
```

- 4 Run the message broker account tool with these arguments:

```
sudo ./sas-rabbitmq-acc-admin change_passwd -t account-type -u user-ID --promptpw
```

-t account-type

specifies the account user type, which is always the `client` type. The client user has full administrative rights. These rights can change in future releases.

-u user-ID

identifies the client user ID for SAS Message Broker.

--promptpw

prompts for the new password for the client user ID for SAS Message Broker. The password that you enter is hidden, by default.

Here is an example that changes the password for the default administrative user:

```
sudo ./sas-rabbitmq-acc-admin change_passwd -t client -u sasclient --promptpw
```

- 5 Restart all SAS Viya services. Restarting the SAS Viya services activates the changes to the credentials for SAS Message Broker. For more information, refer to [SAS Viya 3.2 Administration Guide: General Servers and Services](#).

Configure SAS/ACCESS Interface to DB2

Note: This information is applicable only if you ordered SAS/ACCESS Interface to DB2 (on SAS Viya).

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to DB2” on page 43](#).

- 1 On the CAS node(s), use a text editor to edit the cas_usermods.settings file.

```
sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
```

- 2 Add the following lines:

```
export DB2INSTANCE=DB2-instance
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-DB2-installation
```

- 3 Save and close the cas_usermods.settings file.

Configure ODBC

Note: This information is applicable only if you ordered SAS/ACCESS Interface to ODBC (on SAS Viya).

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to ODBC” on page 44](#).

- 1 Using a text editor, open the odbc.ini file in your home directory in order to configure data sources.

Some vendors of ODBC drivers might provide support for system administrators to maintain a centralized copy of the odbc.ini file via the environment variable ODBCINI. Refer to your ODBC driver's vendor documentation for more specific information.

Add the location of the shared libraries to one of the system environment variables in order to enable the ODBC drivers to be loaded dynamically at run time. The ODBC drivers are ODBC API-compliant shared libraries, which are referred to as shared objects in UNIX.

- 2 On the CAS node(s), use a text editor to edit the cas_usermods.settings file:

```
sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
```

- 3 Add the following lines, depending on the version of ODBC that you are using.

For DataDirect:

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

For iODBC:

```
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINSTINI=location-of-your-odbcinst.ini-file-including-file-name
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-library
```

For unixODBC:

```
export ODBCYSINI=location-of-your-odbc.ini-and-odbcinst.ini-file-without-file-name
export ODBCINI=name-of-your-odbc.ini-file
export ODBCINSTINI=name-of-your-odbcinst.ini-file
export LD_LIBRARY_PATH=location-of-ODBC-driver-manager-library:$LD_LIBRARY_PATH
```

Note: For unixODBC, if ODBCYSINI is not set in your environment, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

- 4 Save and close the cas_usermods.settings file.

Configure SAS/ACCESS Interface to Oracle

During installation, you should have configured the location of the shared libraries and the library path in the vars.yml file. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Oracle” on page 45](#).

To manually configure the variables:

- 1 On the CAS node(s), use a text editor to edit the cas_usermods.settings file:

```
sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
```

- 2 Add the following lines:

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

- 3 Save and close the cas_usermods.settings file.

Configure SAS Data Connector to PostgreSQL

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

A file that contains information about the database connection is required. You have two options for providing connection information:

Note: Create the file in the /opt/sas/viya/home directory.

- Reference a Data Source Name (DSN).

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

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

- Specify connection information in your code.

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

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

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

During installation, you should have configured the location of the shared libraries in the `vars.yml` file. If you did not set up the location of the shared libraries in the `vars.yml` file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the `vars.yml` file. For information, see [“SAS/ACCESS Interface to PostgreSQL” on page 45](#).

- 1 On the CAS node(s), use a text editor to edit the `cas_usermods.settings` file:

```
sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
```

- 2 Add the following lines:

```
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
export PGCLIENTENCODING=encoding-for-the-PostgreSQL-client
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-PostgreSQL-client
```

- 3 Save and close the `cas_usermods.settings` file.

Configure SAS/ACCESS Interface to Teradata

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Teradata (on SAS Viya).

During installation, you should have configured the location of the shared libraries and the library path in the `vars.yml` file. To ensure that any redeployment contains the configuration settings, you must also make these changes in the `vars.yml` file. For information, see [“SAS/ACCESS Interface to Teradata and SAS In-Database Technologies for Teradata” on page 46](#).

To manually configure the variables:

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

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

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

- 3 On the CAS node(s), use a text editor to edit the `cas_usermods.settings` file:

```
sudo vi /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
```

- 4 Add the following lines:

Note: Multiple lines are used for `LD_LIBRARY_PATH` to improve readability. However, in your environment, make sure that you enter the command on a single line.

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

An example of the TTU Default LD_LIBRARY_PATH is

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

- 5 Save and close the cas_usermods.settings file.

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.

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 Save and close the file.

- 4 Restart the CAS controller by running the following command as root.

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

Restart the SAS Visual Investigator Applications

- 1 On the [sviconfig] machine, to restart the applications, run the following script:

Note: For a high-availability deployment, copy the restart-vi-apps.sh file to each machine where applications that are related to SAS Visual Investigator are installed.

```
sudo bash /opt/sas/viya/home/share/svi-configuration/restart-vi-apps.sh
```

Note: The restart-vi-apps.sh script does not restart all services. To restart all services in the SAS Viya environment, you would use the sas-viya-all-services script in the /opt/sas/viya/config/etc/init.d directory.

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

- 2 To verify service functionality, run the following command:

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

- 3 Log on to your provider tenant at the following URL:

```
your-hostname.com/SASVisualInvestigator
```

Note: A blank web page should be displayed.

Connect to PostgreSQL

To connect to the PostgreSQL instance:

- 1 Obtain the PostgreSQL password in one of the following ways:

- If you have changed the default password, use your new password for PostgreSQL.
- To obtain the current PostgreSQL password, run the following command:

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

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv read "config/application/postgres/password"
```

The current PostgreSQL password is returned.

- 2 To connect to the PostgreSQL server, run the following command:

```
/opt/sas/viya/home/bin/psql -t -h localhost -p 5432 -U dbmsowner postgres
```

You will be prompted for the password:

```
Password for user dbmsowner:
```

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

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

If results are returned, the database is available. To end a connection, run the `\quit` command.

Configure a Standard or Multi-tenant Deployment

Overview

SAS Visual Investigator can be deployed in either of two modes: standard or multi-tenant.

Configure a Standard Deployment

- 1 On the machine where [sviconfig] is specified in the inventory.ini file, change to the following directory:

```
cd /opt/sas/viya/home/share/svi-configuration/tenant-management
```

- 2 As root, edit the bareos-tenant.sh file. Modify the svi_HOST, svi_SERVICE_HTTP_DOMAIN, and svi_SERVICE_CONSUL_IP values to reflect the values for your cluster.

Note: If performing an unsecure deployment, modify the HTTP_PORT and HTTP_SCHEMA values.

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

```

InstanceId=default
PostgresBinary="/opt/sas/viya/home/bin/psql"
SourceDir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"

SASCACerts="{SASConfig}/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem"

# environment variable to be set so that onboarding scripts can do appropriate action in
# BareOS installations
export _svi_bareos=YES

ConfigurableItems()
{
    # This is a template file for tenant on-boarding.
    # Copy this file to a new filename and change the properties below.

    # The following values are to be filled about by the administrator performing the onboarding task
    #
    # Enter the tenant hostname and the domain of the tenant host:
    # Note: This should be the short host name of the server not the fully qualified name.
    local -x svi_HOST="svi-host"

    # Note: This should be the domain only not the fully qualified name
    local -x svi_SERVICE_HTTP_DOMAIN="service-http-domain"

    # Note: This should be the details to access Apache HTTP server
    #
    # - For SSL secured systems this should be port 443 and https
    # - For unsecured systems this should be port 80 and http
    local -x svi_SERVICE_HTTP_PORT=443
    local -x svi_SERVICE_HTTP_SCHEME=https

    # Enter the IP address, port and HTTP/HTTPS scheme where consul is running
    #
    # - For SSL secured systems this should be port 8501 and https
    # - For unsecured systems this should be port 8500 and http
    local -x svi_SERVICE_CONSUL_IP="service-consul-IP"
    local -x svi_SERVICE_CONSUL_PORT=8501
    local -x svi_SERVICE_CONSUL_SCHEME=https

    # Enter the port and HTTP scheme for RabbitMQ
    # typically port 15672
    local -x svi_SERVICE_RABBITMQ_INTERNAL_PORT="15672"
    local -x svi_SERVICE_RABBITMQ_SCHEME=https

```

- 3** To initialize SAS Visual Investigator in standard mode, run the bareos-tenant.sh script:

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

Note: Once you deploy in standard mode, you cannot convert to multi-tenant mode.

- 4** To apply the hot fix, run the following command:

```

cd /opt/sas/viya/home/share/svi-configuration
sudo bash apply_hotfix.sh -h host-where-httpproxy-is-located -u admin-user

```

Note: For *admin-user*, specify the CAS user that is defined in the sitedefault.yml file.

Configure a Multi-tenant Deployment

Configure Changes for a Multi-tenant Deployment

SAS Visual Investigator 10.3.1 releases are multi-tenant-capable deployments. Each tenant must be contained in the same LDAP server. SAS Viya does not support the concept of separate servers for one or more tenants.

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

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

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

Onboard a Tenant

To onboard a tenant:

- 1 On the machine where `[sviconfig]` is specified in the `inventory.ini` file, change to the following directory:

```
cd /opt/sas/viya/home/share/svi-configuration/tenant-management
```

- 2 As root, edit the `bareos-tenant.sh` file. Modify the `svi_HOST`, `svi_SERVICE_HTTP_DOMAIN`, and `svi_SERVICE_CONSUL_IP` values to reflect the values for your cluster.

Note: If performing an unsecure deployment, modify the `HTTP_PORT` and `HTTP_SCHEMA` values.

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

InstanceId=default
PostgresBinary="/opt/sas/viya/home/bin/psql"
SourceDir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"

SASCACerts="${SASConfig}/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem"

# environment variable to be set so that onboarding scripts can do appropriate action in
# BareOS installations
export _svi_bareos=YES

ConfigurableItems()
```

```
{
# This is a template file for tenant on-boarding.
# Copy this file to a new filename and change the properties below.

# The following values are to be filled about by the administrator performing the onboarding task
#
# Enter the tenant hostname and the domain of the tenant host:
# Note: This should be the short host name of the server not the fully qualified name.
local -x svi_HOST="svi-host"

# Note: This should be the domain only not the fully qualified name
local -x svi_SERVICE_HTTP_DOMAIN="service-http-domain"

# Note: This should be the details to access Apache HTTP server
#
# - For SSL secured systems this should be port 443 and https
# - For unsecured systems this should be port 80 and http
local -x svi_SERVICE_HTTP_PORT=443
local -x svi_SERVICE_HTTP_SCHEME=https

# Enter the IP address, port and HTTP/HTTPS scheme where consul is running
#
# - For SSL secured systems this should be port 8501 and https
# - For unsecured systems this should be port 8500 and http
local -x svi_SERVICE_CONSUL_IP="service-consul-IP"
local -x svi_SERVICE_CONSUL_PORT=8501
local -x svi_SERVICE_CONSUL_SCHEME=https

# Enter the port and HTTP scheme for RabbitMQ
# typically port 15672
local -x svi_SERVICE_RABBITMQ_INTERNAL_PORT="15672"
local -x svi_SERVICE_RABBITMQ_SCHEME=https
}
```

- 3 To initialize the first tenant, run the `bareos-tenant.sh` script:

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

- 4 To onboard a tenant, run the following command:

```
sudo bash bareos-tenant.sh your-tenant-name
```

- 5 For each tenant, apply the hot fix by running the following command:

```
cd /opt/sas/viya/home/share/svi-configuration
sudo bash apply_hotfix.sh -h host-where-http-proxy-is-located -u tenant-admin -t tenant-name
```

Note: For *tenant-admin*, specify the CAS user that is defined in the `sitedefault.yml` file.

Configure the First System User

The first system user (`viadmin`) is not part of the SAS Visual Investigator administrator (`sviadms`) group. Therefore, certain Web pages are not viewable. To enable the first user to load all Web pages:

- 1 Log on as the first system user (for example, `viadmin`) using the tenant URL.
- 2 Navigate to the SAS Visual Investigator: Administration page (`admin.html`).

- 3 Choose **Permissions**.
- 4 Select **sviadms**.
- 5 Select **Add Member**.
- 6 Select **Users**.
- 7 Select the viadmin user ID.
- 8 Click **OK**.

Subsequent logon is required to view all page content. Page access is enabled for future logon sessions.

To set up additional users and groups, refer to [Managing Users](#) in the *SAS Visual Investigator 10.3.1 Administrator's Guide*.

Validating the Deployment

<i>Perform Installation Qualification on RPM Packages</i>	69
<i>Access CAS Server Monitor</i>	71
<i>Verify RabbitMQ</i>	71
<i>Verify PostgreSQL</i>	72
<i>Verify SAS/ACCESS Interface to DB2</i>	72
<i>Verify SAS/ACCESS Interface to ODBC</i>	73
<i>Verify SAS/ACCESS Interface to Oracle</i>	74
<i>Verify SAS/ACCESS Interface to PostgreSQL</i>	76
<i>Verify SAS/ACCESS Interface to Teradata</i>	77
<i>Validate Elasticsearch</i>	78
<i>Validate SAS Visual Investigator</i>	79

Perform Installation Qualification on RPM Packages

Some of your SAS software is collected in RPM (Red Hat Package Manager) packages. To qualify the installation of your RPM packages, run the basic RPM command:

```
rpm -Vv package-name
```

Note: The `-Vv` option provides a status for all files in the package. To list the failures only, use the `-v` option.

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

```
rpm -Vv sas-certframe
```

To verify SAS Event Stream Processing deployment, run the following command to obtain a list of the relevant RPM packages that are deployed on your system:

```
rpm -qa sas-esp*
```

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 -qg "SAS");do sudo rpm -Vv $i;done
```

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

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

#

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

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

The error indicators are shown in the following format:

```
SM5DLUGT c
```

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

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

The meaning of each error indicator is described as follows:

- S File size. RPM keeps track of file sizes. A difference of even one byte triggers a verification error.
- M File mode. The permissions mode is a set of bits that specifies access for the file's owner, group members, and others. Even more important are two additional bits that determine whether a user's group or user ID should be changed if they execute the program that is contained in the file. Since these bits permit any user to become root for the duration of the program, you must be cautious with a file's permissions.
- 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. You could also get a d in this slot, indicating that the file is for documentation, which is also likely to change often.

Verification failures are expected for files that contain frequently changing content, such as environment-specific Java paths, newly generated TLS certificates, SAS license information, and CAS customizations. Such verification failures for these types of files usually do not indicate any errors in the files.

Note: In SAS Viya 3.4, the following files are modified during the deployment process. If you perform a verification and receive error indications for the following files, they can be safely ignored. The following are the default pathnames.

- `/opt/sas/viya/config/etc/evmcltsvcs/alert-track.json`
- `/opt/sas/viya/config/etc/evmcltsvcs/ops-agent.json`
- `/opt/sas/viya/config/etc/evmcltsvcs/watch-log.json`
- `/opt/sas/viya/config/etc/evmsvrops/ops-agentsrv.json`
- `/opt/sas/viya/config/etc/evmsvrops/stream-evdm.json`

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:

`scheme://reverse-proxy-server/cas-shared-default-http`

Note: In a full deployment, the scheme is https.

Here is an example:

`https://host1.sas.com/cas-shared-default-http`

Log on using one of the SAS Administrator users that you established in [“Set the Password for the CAS Administrator or Another Administrative Account” on page 57](#).

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

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

Note: To access CAS Server Monitor, the password must be set for the cas user ID or other administrative account. To set the password, see [“Set the Password for the CAS Administrator or Another Administrative Account” on page 57](#).

Verify RabbitMQ

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

- If HTTPS is enabled:

`https://RabbitMQ-IP-address:15672/#/`

Note: If you receive the `Your Connection is Not Private` message, click the **Advanced** button and click to proceed until you have replaced the certificates. For information about best practices for certificates, see [“Apache httpd” on page 11](#).

- If HTTP is enabled:

`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.

Verify SAS/ACCESS Interface to DB2

Note: The information in this section is applicable only if you ordered SAS Data Connector to DB2.

To verify that SAS Data Connector to DB2 was successfully deployed:

- 1 Log on to the CAS controller host:

- a Create the .authinfo file in the home directory of the user. Here is an example:

```
/home/user-ID/.authinfo
host localhost port port-number user user-ID password password
```

- b Grant Read and Write access for the owner of .authinfo:

```
chmod 600 .authinfo
```

- 2 Start the CAS shell:

- a Run the following command:

```
java -cp "/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/lib/*"
-Dcom.sas.cas.authinfo.debug=true
-Djavax.net.ssl.trustStore=
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.jks
com.sas.cas.Cash
cash>
```

- b At the shell prompt, specify the host name and port:

```
cash> host=host-name:port-number
```

Here is an example:

```
cash> host=localhost:5570
```

c Run the sessionId action

```
cash> s:session_sessionId{}
```

Here is the output:

```
AUTHINFO: Current user is centos
AUTHINFO: Checking for file /home/centos/.authinfo
AUTHINFO: File /home/centos/.authinfo is owned by centos
AUTHINFO: Permissions for /home/centos/.authinfo: rw-----
AUTHINFO: Parsed 1 entries from /home/centos/.authinfo
AUTHINFO: Matched AuthInfoEntry [defaultHost=false, host=localhost,
user=videmo, password=****, port=5570]

{
  Session:Wed Jan 17 18:47:25 2018="8a865674-16d5-944c-8fad-c433ff6a69e2"
}
[Performance] Elapsed 0.000, CPU 0.000, Nodes 1
[Disposition] (OK)
```

3 Verify the SAS data connector:

```
cash> s:table_loadDataSource{name="db2"}
```

If the data connector was successfully deployed, the following results are returned:

```
[Log] (INFO) NOTE: Cloud Analytic Services added the datasource 'db2'.
{
  "db2"
}
[Performance] Elapsed 0.000, CPU 0.000, Nodes 1
[Disposition] (OK)
```

If an error was returned, perform the configuration steps again.

Verify SAS/ACCESS Interface to ODBC

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

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

1 Log on to the CAS controller host:

a Create the .authinfo file in the home directory of the user. Here is an example:

```
/home/user-ID/.authinfo
host localhost port port-number user user-ID password password
```

b Run the following command:

Grant Read and Write access for the owner of .authinfo:

```
chmod 600 .authinfo
```

2 Start the CAS shell:

```
a java -cp "/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/lib/*"
-Dcom.sas.cas.authinfo.debug=true
-Djavax.net.ssl.trustStore=
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.jks
com.sas.cas.Cash
cash>
```

b At the shell prompt, specify the host name and port:

```
cash> host=host-name:port-number
```

Here is an example:

```
cash> host=localhost:5570
```

c Run the sessionId action

```
cash> s:session_sessionId{}
```

Here is the output:

```
AUTHINFO: Current user is centos
AUTHINFO: Checking for file /home/centos/.authinfo
AUTHINFO: File /home/centos/.authinfo is owned by centos
AUTHINFO: Permissions for /home/centos/.authinfo: rw-----
AUTHINFO: Parsed 1 entries from /home/centos/.authinfo
AUTHINFO: Matched AuthInfoEntry [defaultHost=false, host=localhost,
user=videmo, password=****, port=5570]

{
  Session:Wed Jan 17 18:47:25 2018="8a865674-16d5-944c-8fad-c433ff6a69e2"
}
[Performance] Elapsed 0.000, CPU 0.000, Nodes 1
[Disposition] (OK)
```

3 Verify the SAS data connector:

```
cash> s:table_loadDataSource{name="odbc"}
```

If the data connector was successfully deployed, the following results are returned:

```
[Log] (INFO) NOTE: Cloud Analytic Services added the datasource 'odbc'.

{
  "odbc"
}
[Performance] Elapsed 0.002, CPU 0.001, Nodes 1
[Disposition] (OK)
```

If an error was returned, perform the configuration steps again.

Verify SAS/ACCESS Interface to Oracle

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

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

1 Log on to the CAS controller host:

a Create the .authinfo file in the home directory of the user. Here is an example:

```
/home/user-ID/.authinfo
host localhost port port-number user user-ID password password
```

b Grant Read and Write access for the owner of .authinfo:

```
chmod 600 .authinfo
```

2 Start the CAS shell:

a Run the following command:

```
java -cp "/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/lib/*"
-Dcom.sas.cas.authinfo.debug=true
-Djavax.net.ssl.trustStore=
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.jks
com.sas.cas.Cash
cash>
```

b At the shell prompt, specify the host name and port:

```
cash> host=host-name:port-number
```

Here is an example:

```
cash> host=localhost:5570
```

c Run the sessionId action

```
cash> s:session_sessionId{}
```

Here is the output:

```
AUTHINFO: Current user is centos
AUTHINFO: Checking for file /home/centos/.authinfo
AUTHINFO: File /home/centos/.authinfo is owned by centos
AUTHINFO: Permissions for /home/centos/.authinfo: rw-----
AUTHINFO: Parsed 1 entries from /home/centos/.authinfo
AUTHINFO: Matched AuthInfoEntry [defaultHost=false, host=localhost,
user=videmo, password=****, port=5570]

{
  Session:Wed Jan 17 18:47:25 2018="8a865674-16d5-944c-8fad-c433ff6a69e2"
}
[Performance] Elapsed 0.000, CPU 0.000, Nodes 1
[Disposition] (OK)
```

3 Verify the SAS data connector:

```
cash> s:table_loadDataSource{name="oracle"}
```

If the data connector was successfully deployed, the following results are returned:

```
[Log] (INFO) NOTE: Cloud Analytic Services added the datasource 'oracle'.

{
  "oracle"
}
[Performance] Elapsed 0.002, CPU 0.000, Nodes 1
[Disposition] (OK)
```

If an error was returned, perform the configuration steps again.

Verify SAS/ACCESS Interface to PostgreSQL

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

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

1 Log on to the CAS controller host:

- a** Create the .authinfo file in the home directory of the user. Here is an example:

```
/home/user-ID/.authinfo
host localhost port port-number user user-ID password password
```

- b** Run the following command:

Grant Read and Write access for the owner of .authinfo:

```
chmod 600 .authinfo
```

2 Start the CAS shell:

- a**

```
java -cp "/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/lib/*"
-Dcom.sas.cas.authinfo.debug=true
-Djavax.net.ssl.trustStore=
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.jks
com.sas.cas.Cash
cash>
```

- b** At the shell prompt, specify the host name and port:

```
cash> host=host-name:port-number
```

Here is an example:

```
cash> host=localhost:5570
```

- c** Run the sessionId action

```
cash> s:session_sessionId{}
```

Here is the output:

```
AUTHINFO: Current user is centos
AUTHINFO: Checking for file /home/centos/.authinfo
AUTHINFO: File /home/centos/.authinfo is owned by centos
AUTHINFO: Permissions for /home/centos/.authinfo: rw-----
AUTHINFO: Parsed 1 entries from /home/centos/.authinfo
AUTHINFO: Matched AuthInfoEntry [defaultHost=false, host=localhost,
user=videmo, password=****, port=5570]

{
  Session:Wed Jan 17 18:47:25 2018="8a865674-16d5-944c-8fad-c433ff6a69e2"
}
[Performance] Elapsed 0.000, CPU 0.000, Nodes 1
[Disposition] (OK)
```

3 Verify the SAS data connector:

```
cash> s:table_loadDataSource{name="postgres"}
```


If the data connector was successfully deployed, the following results are returned:

```
[Log] (INFO) NOTE: Cloud Analytic Services added the datasource 'postgres'.

{
  "postgres"
}
[Performance] Elapsed 0.007, CPU 0.002, Nodes 1
[Disposition] (OK)
```

If an error was returned, perform the configuration steps again.

Verify SAS/ACCESS Interface to Teradata

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

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

1 Log on to the CAS controller host:

- a** Create the .authinfo file in the home directory of the user. Here is an example:

```
/home/user-ID/.authinfo
host localhost port port-number user user-ID password password
```

- b** Grant Read and Write access for the owner of .authinfo:

```
chmod 600 .authinfo
```

2 Start the CAS shell:

- a** Run the following command:

```
java -cp "/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/lib/*"
-Dcom.sas.cas.authinfo.debug=true
-Djavax.net.ssl.trustStore=
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.jks
com.sas.cas.Cash
cash>
```

- b** At the shell prompt, specify the host name and port:

```
cash> host=host-name:port-number
```

Here is an example:

```
cash> host=localhost:5570
```

- c** Run the sessionId action

```
cash> s:session_sessionId{}
```

Here is the output:

```
AUTHINFO: Current user is centos
AUTHINFO: Checking for file /home/centos/.authinfo
AUTHINFO: File /home/centos/.authinfo is owned by centos
AUTHINFO: Permissions for /home/centos/.authinfo: rw-----
AUTHINFO: Parsed 1 entries from /home/centos/.authinfo
AUTHINFO: Matched AuthInfoEntry [defaultHost=false, host=localhost,
user=videmo, password=****, port=5570]
```

```
{
  Session:Wed Jan 17 18:47:25 2018="8a865674-16d5-944c-8fad-c433ff6a69e2"
}
[Performance] Elapsed 0.000, CPU 0.000, Nodes 1
[Disposition] (OK)
```

3 Verify the SAS data connector:

```
cash> s:table_loadDataSource{name="teradata"}
```

If the data connector was successfully deployed, the following results are returned:

```
[Log] (INFO) NOTE: Cloud Analytic Services added the datasource 'teradata'.

{
  "teradata"
}
[Performance] Elapsed 0.002, CPU 0.001, Nodes 1
[Disposition] (OK)
```

If an error was returned, perform the configuration steps again.

Validate Elasticsearch

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

```
sudo curl -XGET
'https://IP-address-for-Elasticsearch-master-node:9200/_cluster/health?pretty=true'
--cert /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tls/certs/consul/default/consul.pem
--key /opt/sas/viya/config/etc/SASSecurityCertificateFramework/private/consul/default/
consul.key
```

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>.

Note: For deployments with only one data node, the value of status should be **yellow** to indicate that the cluster is functional.

Validate SAS Visual Investigator

- 1 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`.
- 2 Log on as the user that you identified in step 1.
- 3 When you are prompted for **Assumable Groups**, click **Yes**.

Uninstalling SAS Viya

<i>What deploy-cleanup Does</i>	81
<i>Uninstall Command</i>	81

What deploy-cleanup Does

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

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

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

- the inventory file that is used in the deployment
- the `vars.yml` file that is used in the deployment
- the deployment log

The `deploy-cleanup` command renames the `/opt/sas/viya` directory to `/opt/sas/viya_epoch`.

The uninstallation does not remove any users that have been set up.

Uninstall Command

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

To uninstall your SAS Viya software, run the appropriate command from the following table, based on the password requirements for the user ID that performs the command.

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

Password Requirements	Command
Does not require passwords	<code>ansible-playbook deploy-cleanup.yml</code>
Requires a sudo password only	<code>ansible-playbook deploy-cleanup.yml --ask-become-pass</code>
Requires an SSH password only	<code>ansible-playbook deploy-cleanup.yml --ask-pass</code>
Requires both a sudo and an SSH password	<code>ansible-playbook deploy-cleanup.yml --ask-pass --ask-become-pass</code>

The `deploy-cleanup` command leaves a few running processes that should be removed individually.

- 1 `httpd` remains on your system because other software might be using it. If no other software is using `httpd`, you can stop its processes and remove it by running the following command:

```
yum remove httpd
```

- 2 The `epmd` process remains running on your system as an artifact of SAS Message Broker. To stop the process:

- a List all active processes by running the following command:

```
ps -A
```

- b In the results, find “`epmd`” in the far right column, and then locate its process ID (PID) in the far left column.

- c Remove the `epmd` process by running the following command:

```
kill process-ID-for-epmd
```

- 3 The `sas-configuration-cli` process could remain running on your system. To stop the process:

- a List all active processes by running the following command:

```
ps -A
```

- b In the results, find “`sas-configuration-cli`” in the far right column, and then locate its process ID (PID) in the far left column.

- c Remove the `sas-configuration-cli` process by running the following command:

```
kill process-ID-for-sas-configuration-cli
```

Completing the Deployment

<i>Save Snapshot Directory Content</i>	83
<i>Further Documentation</i>	83

Save Snapshot Directory Content

If you successfully deployed your software, the process saved valuable information for later use. The information is saved in the `sas_deployment.tgz` file in the directory in which you saved the playbook, in the `/snapshot/epoch` subdirectory. The `sas_deployment.tgz` file includes the following files, among others:

- the inventory file that is used in the deployment
- the `vars.yml` file that is used in the deployment
- the deployment log

SAS recommends that you copy the `sas_deployment.tgz` file and save it to a separate location, possibly on a another machine. You have a backup of important files that might be required later, such as to update an existing order.

Further Documentation

You can access *SAS Visual Investigator 10.3.1: 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 10.3.1 documentation, you must have an access key. The documentation page explains how to contact SAS Technical Support to request the access key.

Managing Your Software

Overview	85
What Is an Update?	85
What Is an Upgrade?	86
What Is an Add-On Product?	86
What Is a New Ansible Playbook?	86
Updating Your SAS Viya Software	86
Overview	86
List the Packages That Are Available for Update	87
How to Update Your SAS Viya Software	87
Apply the Hot Fix	88
Add SAS Viya Software	89
Overview	89
How To Add SAS Software	89
Migrating the Configuration Information	91
Migration Process	91
Steps to Migrate the Configuration Information	92
Upgrading Your SAS Viya Software	93
Upgrade	93
Generate a New Ansible Playbook	95

Overview

What Is an Update?

An update replaces some or all of your deployed software with the latest versions of that software. Updated software is intended to be compatible with existing configuration, content, and data. To perform an update, you will run the same tools that were run during the initial deployment. You do not need a new order to perform an update. You might determine that your software needs updating or you might be notified by SAS that updates are available.

Note: Converting a single-tenant deployment to a multi-tenant deployment, either through an update or an upgrade, is not supported.

What Is an Upgrade?

An upgrade adds significant feature changes or improvements to your deployed software. To perform an upgrade, you will run the same tools that were run during the initial deployment. You will need a new order to upgrade your deployed software. An upgrade might require changes to the deployed software's configuration.

You might determine that your software needs upgrading or you might be notified by SAS that upgrades are available. SAS recommends creating a backup of the deployed software environment before performing an upgrade.

Note: Converting a single-tenant deployment to a multi-tenant deployment, either through an update or an upgrade, is not supported.

What Is an Add-On Product?

An add-on product is new software that you can order and then install with your currently deployed software. You will need a new order for an add-on product.

Because an add-on product is added to the currently deployed software in an environment, you might need to expand your environment's capacity before installing an add-on product.

What Is a New Ansible Playbook?

SAS might update components of the Ansible playbook that is used to deploy your SAS software. You will need to download the current version of the SAS Orchestration CLI to create a new Ansible playbook for your deployment, and then run the new Ansible playbook.

Updating Your SAS Viya Software

Overview

You must update your deployed software environment in order to get the environment's software to the latest version.

- If you used an Ansible playbook for your initial installation, you should update with Ansible.
- If you have a mirror repository, in order to get the latest compatible software you must first synchronize your mirror repository with the SAS repositories using the following command:

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

After the mirror repository has been synchronized, you can then update your deployment using the mirror repository.

Updating SAS Viya software requires an outage period because all SAS Viya services must be stopped before the update process and restarted after the update process has completed. The update process is the same regardless of whether the deployment is for a single-tenant or a multi-tenant.

Note: An update process preserves any user-modified configuration values in the vars.yml file, but changes made to other files in the deployment might be lost. Therefore, SAS recommends that you make changes to vars.yml when possible in order to streamline the update process.

List the Packages That Are Available for Update

To list the packages that are available for an update process, run the following command:

```
sudo yum check-update "sas-*
```

How to Update Your SAS Viya Software

Overview

An update replaces some or all of your deployed software with the latest versions of that software. An update process is performed with the same command that was used for installing SAS Viya, using the same software order and the same playbook.

Note: If you have a mirror repository, in order to get the latest compatible software you must first synchronize your mirror repository with the SAS repositories, then update your deployment using the mirror repository.

To perform an update process, you must have administrator privileges for the machine. In addition, your account must have superuser (sudo) access. To verify sudo user privileges, run the command: `sudo -v` or `sudo -l`.

Update with Ansible

To update a SAS Viya deployment using Ansible:

- 1 (Optional) Record the existing list of installed software before you begin.

On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

```
sudo rpm -qg SAS > /sas/install/viya_rpms.txt
```

On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. For example, you can run the following command to create a text file that lists all the SAS host groups:

```
sudo yum grouplist "SAS*" > /sas/install/viya_hostgroups.txt
```

Note: If messages appear that are similar to the following example, they can be safely ignored.

```
Repository repositoryname is listed more than once in the configuration
```

- 2 Review the `*_deployment.*` files (for example, `casconfig_deployment.lua`) in the existing deployment for any user-modified changes. If there are any user-modified changes to the `*_deployment.*` files, back up the file and update the `vars.yml` file with the changes before you run the upgrade process. If there is any question, contact SAS Technical Support.

Note: SAS recommends that you add your customizations to the `vars.yml` file rather than to a `*_deployment.*` file in order to preserve your customizations. Otherwise, your customizations would be lost during the update process.

When you use the Ansible playbook to update your environment, the update process backs up the following files:

```
CAS
- /opt/sas/viya/config/etc/cas/default/cas_usermods.settings
- /opt/sas/viya/config/etc/cas/default/casconfig.lua
- /opt/sas/viya/config/etc/cas/default/cas.hosts
```

Object Spawner

```
- /opt/sas/viya/config/etc/spawner/default/spawner.cfg
```

SAS/CONNECT

```
- /opt/sas/viya/config/etc/sysconfig/connect/default/sas-connect
```

- 3 To perform an update process, run the same command and options that you ran when you performed the initial deployment. For more information, see [“Deploy the Software” on page 54](#).
- 4 (Optional) After the update process has completed, record the new list of installed software.

On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

```
sudo rpm -qq SAS > /sas/install/new_viya_rpms.txt
```

On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. For example, you can run the following command to create a text file that lists all the SAS host groups:

```
sudo yum grouplist "SAS*" > /sas/install/new_viya_hostgroups.txt
```

Note: If messages appear that are similar to the following example, they can be safely ignored.

```
Repository repositoryname is listed more than once in the configuration
```

You can compare the results of these files with the files that were created before performing this task to see the differences between the previous and current deployments.

Apply the Hot Fix

After updating your SAS software:

- 1 To ensure that all services are running, run the following commands:

```
/opt/sas/viya/home/share/svi-configuration/  
bash restart-vi-apps.sh  
sudo service sas-viya-all-services status
```

- 2 Apply the hot fix that is based on your type of deployment:

- For a standard deployment, run the following commands:

```
cd /opt/sas/viya/home/share/svi-configuration  
sudo bash apply_hotfix.sh -h host-where-http-proxy-is-located -u admin-user
```

Note: For *admin-user*, specify the CAS user that is defined in the `sitedefault.yml` file.

- For a multi-tenant deployment, for each tenant, run the following commands:

```
cd /opt/sas/viya/home/share/svi-configuration  
sudo bash apply_hotfix.sh -h host-where-http-proxy-is-located -u tenant-admin -t  
tenant-name
```

Note: For *tenant-admin*, specify the CAS user that is defined in the `sitedefault.yml` file.

Add SAS Viya Software

Overview

Here are some of the most common scenarios for adding SAS Viya products:

- Adding new products from your initial SAS Viya order.
For one reason or another, you ordered software and did not install all of it. Or you installed the software, but chose not to configure it.
- Deploying additional products from a new SAS Viya order.
The additional products are not a part of your original SAS Viya order. So you made another order and now have to download and deploy the new order.
- Re-installing and reconfiguring a SAS product.
You want to move a SAS product to a new machine.
- Applying updates (maintenance) to a SAS product that requires also updating its configuration.
For some reason, you were unable to finish applying the updates and you need to rerun the playbook to complete the updates to your configuration.

Adding SAS Viya software to an existing deployment requires an outage period because all SAS Viya services must be stopped before the process begins and they must be restarted after the process has completed. The process is the same regardless of whether the deployment is single-tenant or multi-tenant.

Before you begin, you should review the [“Introduction” on page 1](#), [“System Requirements” on page 5](#), and [“Pre-Installation Tasks” on page 19](#) chapters of this document.

How To Add SAS Software

To add SAS Software and update a SAS Viya deployment:

- 1 (Optional) Record what is installed before you begin.
On each machine in your deployment, create a file that lists the names and versions of all of the RPM packages of the SAS Viya software that are installed. For example, you can run the following command to create a text file that lists all of the SAS RPM packages:

```
sudo rpm -qq SAS > /sas/install/viya_rpms.txt
```


On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. For example, you can run the following command to create a text file that lists all of the SAS host groups:

```
sudo yum grouplist "SAS*" > /sas/install/viya_hostgroups.txt
```
- 2 When you purchase additional products, you receive a new Software Order Email (SOE) from SAS. Use your SOE to download the SAS Orchestration CLI.
- 3 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the download site.
- 4 Extract the new playbook to a separate location from your original playbook. For instance, if you extracted your original playbook to `/sas/install/`, then you might extract the new playbook to `/sas/addon_sas/` instead. You need to extract the new playbook to a different location than you used for your deployment in order to:

- Preserve the original inventory and vars.yml files.
- Make sure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing one, files that were removed in the newer playbook would still be available and could impact researching and resolving deployment issues.

Here is an example of a command that can be used to extract the new playbook.

```
tar xf SAS_Viya_playbook.tgz -C /sas/addon_sas/
```

5 Merge the vars.yml and inventory file from the previous deployment into the new playbook.

- a SAS recommends that you compare the two files since there could be additions or changes in the newer set of files.

```
** compare original to new, then merge vars.yml as needed **
diff /sas/install/sas_viya_playbook/vars.yml /sas/addon_sas/sas_viya_playbook/vars.yml
** compare original to new, then merge inventory files as needed **
diff /sas/install/sas_viya_playbook/inventory.ini /sas/addon_sas/sas_viya_playbook/inventory.ini
```

- b If there is new content in the new vars.yml, then merge your original edits into the new vars.yml. If a key/value pair that was edited in the older file is not in the new file, then there is a good chance that it is not needed in the new file.
- c If the vars.yml file from the deployment being upgraded contains a value for the casenv_tenant variable, use the following commands to remove the registered CAS service associated with that value from Consul.

```
cd /opt/sas/viya/home/bin
./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token \
agent service deregister \
"cas-{casenv_tenant}-default-http"

./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token \
agent service deregister \
"cas-{casenv_tenant}-default"
```

- d Review the inventory file from the original deployment. Review [“Edit the Inventory File” on page 32](#) for details about planning the target machine list for the add-on deployment.

Merge the target machine list from your previous deployment's inventory file into the new playbook's inventory.ini file, and then assign the target machines to the host groups.

- e If there is any question about a key/value pair from an older file needing to be in the new file, contact SAS Technical Support.

6 Install your SAS Viya software.

Follow the remaining steps in the installation chapter, beginning with [“Modify the vars.yml File” on page 37](#).

- 7 If you removed the CAS service associated with a casenv_tenant value described in Step 5, ensure that any bookmarked URLs are updated to remove that value and use "cas-shared-default-http" instead. For example, if your original deployment had a casenv_tenant value of "viya32", then a URL of

```
http://host.company.com/cas-viya32-default-http
```

should be changed to

`http://host.company.com/cas-shared-default-http`

- 8 After the software is installed, complete the tasks described in the subsequent chapters of this deployment guide.
- 9 (Optional) Record the new list of the installed software.

On each machine in your deployment, create a file that lists the names and versions of all of the RPM packages of the SAS Viya software that are installed. For example, you can run the following command to create a text file that lists all of the SAS RPM packages:

```
sudo rpm -qq SAS > /sas/install/new_viya_rpms.txt
```

On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. For example, you can run the following command to create a text file that lists all of the SAS host groups:

```
sudo yum grouplist "SAS*" > /sas/install/new_viya_hostgroups.txt
```

Note: If messages appear that are similar to the following example

```
Repository repositoryname is listed more than once in the configuration
```

These messages can be safely ignored.

You can compare the results of these files with the files that were created prior to performing this task to see the differences between the previous and current deployments.

Migrating the Configuration Information

Important: This section contains several references to *SAS Visual Investigator: Administrator's Guide*. To access the secure SAS Visual Investigator documentation, you must have an access key. A message on the SAS Visual Investigator documentation page explains that licensed customers can contact SAS Technical Support to request the access key. SAS recommends that you obtain the access key before completing the tasks in this section.

Migration Process

SAS Visual Investigator configuration information can be migrated from all SAS Visual Investigator 10.2.x and later versions to SAS Visual Investigator 10.3.x. Because the newer version of SAS Visual Investigator contains enhanced features that were not available in the previous version, additional product configuration might be required.

Here is an overview of the migration process:

- 1 The configuration is exported from SAS Visual Investigator 10.2.x.
 - a Modified templates are saved.
 - b The configuration is exported.
 - c Data source information is obtained.
- 2 SAS Visual Investigator 10.3.x is installed separately.
- 3 The configuration is imported to SAS Visual Investigator 10.3.x.
 - a SAS Visual Investigator 10.3.x is deployed.

- b** A tenant is defined.
- c** Data source information is configured.
- d** A selective configuration import is performed.
- e** For each tenant, steps c and d are repeated.

In order to move the deployment configuration from the previous environment to the new environment, you must import the configuration to the new environment. This is done by using the import process of the import and export configuration procedure.

See [Importing and Exporting Product Configuration Information](#) in *SAS Visual Investigator 10.3: Administrator's Guide*.

Note: If you are exporting from an earlier version of SAS Visual Investigator and importing to 10.3.x, the existing properties in the earlier version (set on the **Properties** page) are ignored. Therefore, you must reconfigure them on the target system.

Steps to Migrate the Configuration Information

- 1** From the existing SAS Visual Investigator 10.2.x version, obtain the following information:
 - **Modified templates** — If you have customized the alert-detail-default, alert-inspector-default, or alert-summary-default templates, access the page in the designer and select **Save As**. Assign new filenames to retain the modifications.

The default versions (recognized by the product by filename only) of these templates are not imported into the new environment.
 - **Configuration details** — Use the export feature to retrieve the configuration information that is associated with the deployment. A ZIP file is obtained. See [Exporting the Deployment Configuration](#) in *SAS Visual Investigator 10.3: Administrator's Guide*.
 - **Data source information** — From the administration interface, obtain the information about all the data sources in this deployment that will be needed in the new deployment. Later, you will manually enter the data store information into the new SAS Visual Investigator 10.3.x system. In order for the import to be successful, the data store names must match.
- 2** Define each tenant that is required in the new SAS Visual Investigator deployment. See [Multiple Tenant Overview](#) and [Managing Tenants — Linux Operating System](#) in *SAS Visual Investigator 10.3.1: Administrator's Guide*.
- 3** Using the information obtained in Step 1 and adding any required data source information, configure the data sources for SAS Visual Investigator 10.3.x.

Note: Failure to configure the data and the data sources before running the import and export procedure results in a configuration import and export failure error. Before you can import a configuration to the target host, you must re-create all data stores that are defined for the source host on the target host. For example, re-create data stores that are used by external entities on the target host. Failure to perform this task results in a configuration import and export failure.
 - a** Obtain the list of data stores in the SAS Visual Investigator 10.2.x system and define them in the 10.3.x system.
 - b** If you uploaded files to your SAS Visual Investigator 10.2.x system, find the same files and upload them to your 10.3.x system using the **Import** toolbar.
- 4** (Optional) If you imported data from CSV files in SAS Visual Investigator 10.2.x through the **Import** tab, you must perform the following steps:

- a Before importing any configuration information into SAS Visual Investigator 10.3.x, you must first import the data from the original CSV files to SAS Visual Investigator 10.3.x through the **Import** tab.

Note: If the original CSV files are no longer available, you can export the tables from the aiuserdata schema. In the SAS Visual Investigator 10.2.x environment, the tables are exported in CSV format. Import these new CSV files into the SAS Visual Investigator 10.3.x environment through the **Import** tab. To ensure consistency between the SAS Visual Investigator 10.2.x and 10.3.x environments, name each CSV file after the table whose data it contains. For example, a table named customers should be added to a CSV file named customers.zip.

Note: You must select the same primary keys and column selections that were used in SAS Visual Investigator 10.2.x.

- b When importing the SAS Visual Investigator 10.2.x ZIP file, deselect all documents and templates that were created in the previous step.

5 Import the configuration information.

TIP During the import, if you receive an error that references a missing object, you can resolve the issue by deselecting the referenced item and then re-importing.

- 6 (Optional) Change the imported queues to use the new default disposition actions. See [Queues](#) in SAS Visual Investigator 10.3.1: Administrator's Guide.
- 7 Click **Re-index and resolve all entities** to run a full re-index to complete the process.
- 8 In multi-tenant systems, repeat the steps for each tenant.

Upgrading Your SAS Viya Software

Upgrade

About Upgrade

Upgrade is designed to take a deployment environment and existing deployed components of SAS Visual Investigator and to transform them, in-place, to a newer version.

Note: Upgrade is available only for upgrading SAS Visual Investigator from 10.2.x to 10.3.x.

Backing Up Your Configuration

Build

This step involves building the deliverables that can execute the remaining Upgrade functions. Two types of detection are used to determine the files needed for the Upgrade process:

Automatic Detection

Automatically detect changes in files (both in the deployment code and the configuration data files).

Case-Specific Detection

Specifically identify some set of changes without performing any analysis.

Back Up

Back up the following services:

- Back up Postgres. See [Back Up the SAS Infrastructure Data Server](#) for more information.
- Consul — Make a backup of all key/value pairs. See [Back up the SAS Configuration Server](#) for more information.
- Back up the entire `/opt/sas/viya` directory structure.

Back up any manual changes that have been made to the deployment environment.

Process

Stop and then restart Consul before starting any applications that depend on Consul data. In such a case, Consul is started and Consul data is updated to include additions, changes, and deletions.

Roll Back

If Upgrade fails for any reason, you can roll back to the backups that you made earlier.

Restore the services to their prior state. All applications should be stopped before performing this procedure.

- Consul — Restore key/value pairs.
- PostgreSQL — Connect to the restored database.
- Search — Restore the search indexes.

Performing an Upgrade

Once you have completed the backup process, perform the following:

- 1 On the existing SAS Visual Investigator 10.2.x system, perform the following steps:
 - a To stop all services, run the following command:


```
sudo service sas-viya-all-services stop
```
 - b To ensure that services are stopped, run the following command:


```
sas-viya-all-services status
```
 - c To reattempt to stop services that are still identified as 'up', run the following command for each service that was displayed in the output of the `sas-viya-all-services status` command:


```
sudo service servicename stop
```

For *servicename* use the name of the service from the output of the `sas-viya-all-services status` command.
- 2 On the CAS controller, perform the following steps:
 - a Open and edit the `/opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig` file as root.
 - b Locate the line that contains `useHostToken` and remove the number sign # character at the beginning of the line.
 - c Locate the line that contains `externalIdent` and remove the number sign # character at the beginning of the line.
- 3 On machines hosting Elasticsearch, back up existing library and configuration components. Run the following commands:

```
sudo mv /opt/sas/viya/home/lib/elasticsearch /opt/sas/viya/home/lib/elasticsearch-5.1.1
sudo mv /opt/sas/viya/config/etc/elasticsearch /opt/sas/viya/config/etc/elasticsearch-5.1.1
```

- 4 Create a new playbook for the SAS Visual Investigator 10.3.x system.
- 5 Create the sitedefault.yml file, as indicated in the installation instructions.
- 6 Edit the vars.yml file, as indicated in the installation instructions.
- 7 Modify your inventory file to reflect your architecture.
- 8 Run the playbook.

```
ansible-playbook site.yml
```

- 9 Stop and restart the system.

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

- 10 To deregister deprecated services from Consul, run the following command:

```
for sname in SASScenarioAdministrator audit; do
    # determine service ID
    sid=$(/opt/sas/viya/home/bin/sas-bootstrap-config catalog service $sname | jq -r '.items[]
| .serviceID')
    # deregister
    /opt/sas/viya/home/bin/sas-bootstrap-config agent service deregister ${sid}
done
```

- 11 To configure the Search Guard plugin for Elasticsearch to ensure secure communication:

- a Copy the update-searchguard-jwtKey.sh script from /opt/sas/viya/home/share/svi-configuration/update-searchguard-jwtKey.sh to each server running Elasticsearch. The script is available on the server where svi-configuration is installed (the [sviconfig] host group in Ansible).
- b Run the script as root:

```
sudo bash update-searchguard-jwtKey.sh
```

- c Ensure that the signing key is included in the Search Guard configuration file:

```
cat /opt/sas/viya/home/lib/elasticsearch/plugins/search-guard-5/sgconfig/sg_config.yml
```

- 12 After installation, follow the instructions for manual configuration.

Note: Onboarding actions are not required.

- 13 With access to SAS Visual Investigator application, update the authorization rules for data objects that were previously added within the 10.2.x application. See [Alert Entity-Level Security](#) for additional process details. Without performing this action, Visual Investigator users and administrators will have limited capability to manage data objects and alerts.

After the completion of these steps, you should have a system that has migrated all your pre-existing tenants, data, and indexes into the new system.

Generate a New Ansible Playbook

If updates are needed in the Ansible playbook, to generate and apply a new Ansible playbook for your deployment:

1 (Optional) Record the existing list of installed software before you begin.

On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

```
sudo rpm -qq SAS > /sas/install/viya_rpms.txt
```

On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. For example, you can run the following command to create a text file that lists all the SAS host groups:

```
sudo yum grouplist "SAS*" > /sas/install/viya_hostgroups.txt
```

Note: If you receive a message such as the following, it can be ignored.

```
Repository repositoryname is listed more than once in the configuration
```

- 2 Use the Software Order Email (SOE) for your original deployment to download the current version of the SAS Orchestration CLI.
- 3 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site.
- 4 You must extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to `/sas/install/`, you might extract the new playbook to `/sas/upgrade/` instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:

- To preserve the original vars.yml file and the inventory file.
- To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following:

```
tar xf SAS_Viya_playbook.tgz -C /sas/upgrade/
```

5 Merge the vars.yml file and the inventory file from the previous deployment into the new playbook.

- a Compare the two vars.yml files, and compare the two inventory files since there could be additions or changes in the newer set of files.

```
diff /sas/install/sas_viya_playbook/vars.yml /sas/upgrade/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory-file /sas/upgrade/sas_viya_playbook/inventory.ini
```

- b If the new files contain new content, then merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.
- c If the original vars.yml file from the deployment that is being upgraded contains a value for the casenv_tenant variable, it must be removed. Run the following commands to remove the registered CAS service.

```
cd /opt/sas/viya/home/bin
./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/c
lient.token \
agent service deregister \
"cas-{casenv_tenant}-default-http"

./sas-bootstrap-config \
--token-file
```

```
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/c
lient.token \
agent service deregister \
"cas-{casenv_tenant}-default"
```

- d If you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.
- 6 If you have deployed SAS Event Stream Processing or SAS Event Stream Manager, perform the following steps:

- a Stop the SAS Event Stream Processing Studio (esvm) service by running the following command on Red Hat Enterprise Linux 6.x:

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

on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-esvm-default
```

- b If you have installed Streamviewer, find its process ID so that you can kill the Streamviewer service:

```
ps -ef
```

Kill the Streamviewer process, substituting the process ID that was returned in the previous step:

```
kill -9 process-ID
```

- c Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

- 7 To apply the new Ansible playbook, change to the directory where the new playbook is located:

```
cd /sas/upgrade/
```

Run the following command:

```
ansible-playbook site.yml
```

- 8 If you removed the CAS service that is associated with a `casenv_tenant` variable (described in Step 3), ensure that any bookmarked URLs are updated to remove that value and use `cas-shared-default-http` instead. For example, if your original deployment contained a `casenv_tenant` value of `viya32`, you should change it from `http://host.company.com/cas-viya32-default-http` to `http://host.company.com/cas-shared-default-http`.

Note: Do not include `casenv_tenant` in your new `vars.yml`. This variable is no longer used.

- 9 (Optional) After the process has completed, record the new list of installed software.

On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

```
sudo rpm -qq SAS > /sas/install/new_viya_rpms.txt
```

On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. For example, you can run the following command to create a text file that lists all the SAS host groups:

```
sudo yum grouplist "SAS*" > /sas/install/new_viya_hostgroups.txt
```

You can compare the results of these files with the files that were created prior to performing this task to see the differences between the previous and current deployments.

Note: If you receive a message such as the following, it can be ignored.

Repository repositoryname is listed more than once in the configuration

Appendix 1

Creating High Availability PostgreSQL Clusters

Overview	99
HA PostgreSQL Topologies	99
Set Up a Horizontal Cluster	101
Edit the inventory.ini File	101
Edit the vars.yml File	102
Set Up a Vertical Cluster	103
Edit the inventory.ini File	103
Edit the vars.yml File	103
Set Up a Hybrid Cluster	104
Edit the inventory.ini File	104
Edit the vars.yml File	104
Set Up Multiple Clusters	105
Modify inventory.ini and vars.yml Files	105
Configure Services to the Clusters	108
Deployment Logs	108
Verify the Deployment	109

Overview

SAS Viya uses High Availability (HA) PostgreSQL as the SAS Infrastructure Data Server. By default, when you use the instructions in in [“Installing SAS Viya with Ansible” on page 31](#), Ansible deploys HA PostgreSQL as a single node on a single machine. However, HA PostgreSQL supports other topologies. This appendix describes those topologies and explains how to use Ansible to deploy them.

HA PostgreSQL Topologies

The standard PostgreSQL deployment with SAS Viya consists of one PGPool and one PostgreSQL data node. All data connection and database requests are routed through PGPool. You connect to PGPool just as you would connect to PostgreSQL, using standard database connectors. With SAS Viya we also have the ability to deploy High Availability PostgreSQL, a clustered database containing one PGPool and one or more data nodes. One data node is designated as a primary and all others are standby nodes. Replication happens in real time to keep the data nodes in sync. All write requests are routed to the primary data node by PGPool; read requests

can be distributed across all data nodes, allowing for higher performance. In the event that the primary data node is lost, PGPool will automatically promote a standby node to primary and reestablish replication from the new primary to the remaining standby data nodes.

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

A cluster can be deployed in four possible configurations:

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

For multi-node deployments, PGPool node can be colocated with data nodes or deployed on its own machine. Note that colocating nodes on a machine provides increased read throughput but also increases the risk of node loss should that machine become unavailable.

The following table demonstrates how nodes can be distributed in the multi-node topologies.

Cluster Configuration	Server	Port	Role
Horizontal	Server 1	5432	Primary
	Server 2	5432	Standby
	Server 3	5432	Standby
	Server 4	5432	Standby
Vertical	Server 1	5532	Primary
	Server 1	5533	Standby
	Server 1	5534	Standby
	Server 1	5535	Standby
Hybrid	Server 1	5632	Primary
	Server 1	5633	Standby
	Server 2	5632	Standby
	Server 2	5633	Standby

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

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

When you revise the vars.yml file for your cluster, the following variables under INVOCATION_VARIABLES should be modified:

pgpoolc

- PCP_PORT: the PCP port for the PGPool instance
- PGPOOL_PORT: the PGPool port. This is the primary port that all database connections will go to.
- SANMOUNT: the location where the data files will be placed
- SERVICE_NAME: the unique name that you assign to your cluster

sasdatasvrc

- NODE_NUMBER: the sequential node identifier starting at 0
- NODE_TYPE: P for primary or S for standby. There can be only one primary per cluster.>
- PG_PORT: The PostgreSQL database port. PGPool talks to the database on this port. Clients use the PGPOOL_PORT.
- SANMOUNT: the location where the data files will be placed
- SERVICE_NAME: the unique name that you assign to your cluster

Set Up a Horizontal Cluster

Edit the inventory.ini File

Modify the inventory.ini file in order to describe the topology that you are using. First, define all the machines in your deployment as described in [“Specify the Machines in the Deployment” on page 33](#). Then assign the machines to the host groups as described in [“Assign the Target Machines to Host Groups” on page 35](#).

Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

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

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
'''

[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
```

```
deployTarget4
...
```

Edit the vars.yml File

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

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

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

After you save the vars.yml file and you complete the other deployment steps, use the commands described in [“Deploy the Software” on page 54](#) to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Vertical Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at [“Specify the Machines in the Deployment” on page 33](#). Then assign the machines to the host groups as described at [“Assign the Target Machines to Host Groups” on page 35](#). Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

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

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deleyTarget1
'''
[sasdatasvrc]
deployTarget1
...
```

Edit the vars.yml File

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

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

```

SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres
- NODE_NUMBER: '3'
  NODE_TYPE: S
  PG_PORT: '5535'
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres

```

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

After you save the vars.yml file and you complete the other deployment steps, use the commands described at [“Deploy the Software” on page 54](#) to deploy your SAS Viya software, including HA PostgreSQL.

Set Up a Hybrid Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at [“Specify the Machines in the Deployment” on page 33](#). Then assign the machines to the host groups as described in [“Assign the Target Machines to Host Groups” on page 35](#). Make sure that the machine that you want to use for PGPool is listed under [pgpoolc] and that every machine that you want to be a PostgreSQL data node is listed under [sasdatasvrc].

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

```

deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
...
[sasdatasvrc]
deployTarget1
deployTarget2
...

```

Edit the vars.yml File

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

```

# Multiple invocation definitions
INVOCATION_VARIABLES:
  deployTarget1:
    pgpoolc:
      - PCP_PORT: '5631'

```

```

PGPOOL_PORT: '5630'
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
SERVICE_NAME: postgres
sasdatasvc:
- NODE_NUMBER: '0'
  NODE_TYPE: P
  PG_PORT: '5632'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
  SERVICE_NAME: postgres
- NODE_NUMBER: '1'
  NODE_TYPE: S
  PG_PORT: '5633'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
  SERVICE_NAME: postgres
deployTarget2:
sasdatasvc:
- NODE_NUMBER: '2'
  NODE_TYPE: S
  PG_PORT: '5632'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
  SERVICE_NAME: postgres
- NODE_NUMBER: '3'
  NODE_TYPE: S
  PG_PORT: '5633'
  SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
  SERVICE_NAME: postgres

```

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

After you save the vars.yml file and you complete the other deployment steps, use the commands described at [“Deploy the Software” on page 54](#) to deploy your SAS Viya software, including HA PostgreSQL.

Set Up Multiple Clusters

Modify inventory.ini and vars.yml Files

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

- a single-node cluster with pgpoolc and sasdatasvc on a machine named deployTarget1
- a horizontal cluster with pgpoolc on deployTarget1 and a sasdatasvc node on each machine
- a vertical cluster with pgpoolc on deployTarget3 and all the sasdatasvc nodes on deployTarget4
- a hybrid cluster with pgpoolc on deployTarget1, two sasdatasvc nodes on deployTarget2, and two more sasdatasvc nodes on deploytarget3

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

```

deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa

```

```

deploytarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
deployTarget3
deployTarget4
...
[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
...

```

This is how the INVOCATION_VARIABLES section of the vars.yml file would be filled out:

```

# Multiple invocation definitions
INVOCATION_VARIABLES:
  deployTarget1:
    pgpoolc:
      - PCP_PORT: '5431'
        PGPOOL_PORT: '5430'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_hybrid
      - PCP_PORT: '5461'
        PGPOOL_PORT: '5460'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
    sasdatasvrc:
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5452'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_horizontal
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5462'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
  deployTarget2:
    sasdatasvrc:
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5432'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_hybrid
      - NODE_NUMBER: '2'
        NODE_TYPE: S
        PG_PORT: '5433'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_hybrid
      - NODE_NUMBER: '1'
        NODE_TYPE: S
        PG_PORT: '5452'

```

```

    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal
deployTarget3:
  pgpoolc:
    - PCP_PORT: '5441'
      PGPOOL_PORT: '5440'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_vertical
  sasdatasvrc:
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5432'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_hybrid
    - NODE_NUMBER: '3'
      NODE_TYPE: S
      PG_PORT: '5433'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_hybrid
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5452'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_horizontal
deployTarget4:
  pgpoolc:
    - PCP_PORT: '5451'
      PGPOOL_PORT: '5450'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_horizontal
  sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5442'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_vertical
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5443'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_vertical
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5444'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_vertical
    - NODE_NUMBER: '3'
      NODE_TYPE: S
      PG_PORT: '5445'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_vertical
    - NODE_NUMBER: '3'
      NODE_TYPE: S
      PG_PORT: '5452'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_horizontal

```

Note: If you are deploying multiple clusters, one of the PG Pools must be named postgres, and each PG Pool name must be unique across clusters. In addition, each cluster must contain one sasdatasvrc node with a NODE_TYPE of P.

Configure Services to the Clusters

By default, all microservices connect to the HA Postgres cluster that is named postgres. You can configure individual services to use additional HA Postgres clusters (if they exist) by adding service-specific sections to the sitedefault.yml file.

- 1 If you have not already copied and renamed the sitedefault.yml file, locate the sitedefault_sample.yml file on the Ansible controller machine. If you used the recommended location for uncompressing your playbook, the file is located at `/sas/install/sas_viya_playbook/roles/consul/files/sitedefault_sample.yml`. Make a copy of sitedefault_sample.yml and name the copy sitedefault.yml.
- 2 Open the sitedefault.yml file.
- 3 At the end of the existing file and at the same indentation level as `application`, add the following content:

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

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

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

```
config:
  application:
  ...
  authorization:
    sas:
      database:
        databaseServerName: postgres-horizontal
        spring.datasource.password: ${sas.database.postgres-horizontal.password}
```

- 4 Save and close the sitedefault.yml file.

Deployment Logs

Each PG Pool node and HA PostgreSQL data node has its own set of directories for logging. The logs for PG Pool are located at

```
/opt/sas/viya/config/var/log/sas/sasdatasvrc/postgres/node0/
```

The log for the HA PostgreSQL nodes is located at

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

Verify the Deployment

The deployment performs a verification of the HA PostgreSQL cluster before it completes. This verification first confirms that connections can be made to PGPool and to all data nodes, and then runs queries on all of the nodes. The verification also performs write and delete operations to ensure that values that are written to or removed from the primary data node are replicated to all of the standby nodes in a multi-node deployment.

The verification log is called `sds_status_check_date-timestamp.log`. It can be found in the pgpool log folder of each cluster. The fastest way to determine whether your HA PostgreSQL deployment was successful is to read the verification log.

Appendix 2

Creating and Using Mirror Repositories

Overview	111
Requirements	111
Ansible Controller	112
Connected Repository Mirror	112
Unconnected Repository Mirror	112
Deployment Targets	112
Machine Combinations	113
Use Ansible to Create a Mirror Repository	113
Confirm that Ansible Is Installed on the Ansible Controller	113
Confirm the Identities of the Hosts	113
Prepare the repohosts Inventory File	113
Confirm Network Connectivity and Ansible Accessibility	114
(Optional) Install and Enable Apache httpd	114
Create the Mirror Repository	115
Confirm HTTP Connectivity to the Mirror Repository	115
Deploy the SAS Viya Software to the Deployment Targets	116
Uninstalling SAS Viya from Mirrored Repositories	116
Uninstall the Repositories	116
Uninstall from the Deployment Target	117

Overview

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

Requirements

The instructions in this appendix assume a topology that consists of one or more machines that perform these roles: an Ansible controller, a mirror repository host connected to the Internet, a mirror repository host that is not connected to the Internet, and deployment targets. All machines described in this chapter must meet the operating system requirements described in [“System Requirements” on page 5](#). The following topics describe each type of machine and additional requirements.

Ansible Controller

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

- does not require Internet access.
- requires network connectivity and Ansible accessibility to itself, as well as to the connected repository mirror, the unconnected repository mirror, and the deployment target machines.
- must have Ansible installed.
- must be capable of controlling itself through Ansible.

Connected Repository Mirror

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

- must have Internet access.
- must be capable of control by the Ansible controller.
- has 100 GB of free disk space in `/opt/sas/repomirror` to hold a temporary archive of the repository mirror files.

Unconnected Repository Mirror

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

- does not require internet access.
- is reachable from your deployment target machine or machines by HTTP.
- can be controlled by your Ansible controller machine.
- has 100 GB of free disk space in `/var/www/html/pulp` to hold the mirror repository files.

Deployment Targets

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

Machine Combinations

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

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

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

Use Ansible to Create a Mirror Repository

Confirm that Ansible Is Installed on the Ansible Controller

- 1 Run the following command on the Ansible controller:

```
ansible --version
```

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

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

- 3 If your results are different, Ansible has not been installed on the machine. To install Ansible on the machine, see [“Install Ansible” on page 25](#).

Confirm the Identities of the Hosts

Ensure that the output of the `hostname`, `hostname -f`, and the `hostname -s` command prints good and expected output.

Prepare the rephosts Inventory File

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

- 2 Ensure that the repohosts file is writable.

```
chmod +w repohosts
```

- 3 Open the repohosts file.

- 4 The beginning of the file contains the following lines:

```
lighthost ansible_host=<machine_address>
darkhost ansible_host=<machine_address>
```

In the first line, replace `<machine_address>` with any resolvable address, such as the IP address or the fully qualified domain name, for the machine that is the connected mirror repository. In the second line, replace `<machine_address>` with any resolvable address for the machine that is the unconnected mirror repository. If either mirror repository will be running Ansible, replace the target declaration for the appropriate machine with `ansible_connection=local`. Here is an example:

```
lighthost ansible_connection=local
```

Note: Do not use 127.0.0.1 as an IP address for any machines in the file repohosts.

If you add `ansible_user` information, ensure that the same user is added to both lines.

- 5 Save and close the repohosts file.

Confirm Network Connectivity and Ansible Accessibility

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

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

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

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

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

(Optional) Install and Enable Apache httpd

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

- 1 On the Ansible controller machine, locate the `repo_vars.yml` file in the `/sas_viya_playbook/utility` directory.
- 2 Run the following command to ensure that the file is writeable:

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

3 Open `repo_vars.yml`.

4 Locate the following line:

```
# setup_httpd_server: no
```

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

```
setup_httpd_server: yes
```

6 Save and close the `repo_vars.yml` file.

7 On the unconnected mirror repository machine, ensure that firewall software is not running. Use the commands in steps 3 and 4 of [“Firewall Considerations” on page 22](#).

Create the Mirror Repository

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

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

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

- downloads SAS software RPM package files from entitled yum repositories that are hosted by SAS on the Internet
 - places the downloaded files in a temporary location on the connected mirror repository (`/tmp/mirror/location` by default)
 - creates a file named `repo_override.txt` in `/opt/sas/repomirror` on the mirror host
- Note:** You can also create `repo_override.txt` on the Ansible controller by setting the `create_repo_deployment_file_on_controller` value in the `repo_vars.yml` file to `yes`.
- copies the files from the temporary location on the connected mirror repository to an Apache httpd accessible location on the unconnected mirror repository (`/var/www/html/pulp/repos` by default)
 - (optional) installs and starts Apache httpd software on the unconnected mirror repository

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

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

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

Confirm HTTP Connectivity to the Mirror Repository

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

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

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

Deploy the SAS Viya Software to the Deployment Targets

Before deploying your SAS Viya software, you must complete the steps described in [“Edit the Inventory File” on page 32](#) and [“Modify the vars.yml File” on page 37](#). After those sections are completed, perform the following steps:

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

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

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

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

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

- 3 Depending on the machines in your mirror topology, you may need to copy the `repo_override.txt` file to your Ansible controller machine.
- 4 On the Ansible controller machine, from the `sas_viya_playbook` directory, run the following command:

```
ansible-playbook site.yml -e "@full-path-to-repo-override-file"
```

Note: For more information about the `repo_override.txt` file, see [“Create the Mirror Repository” on page 115](#).

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

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

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

Uninstalling SAS Viya from Mirrored Repositories

Uninstall the Repositories

- 1 On the connected mirror repository machine, remove the SAS packages with the following command:

```
yum remove "sas-*)"
```

- 2 On the connected mirror repository machine, remove all the files in the location described in the `repo_vars.yml` file as the `mirror_loc` value with the following command.

```
rm -rf directory-location
```

For example, if you used the default value for `mirror_loc` in the `repo_vars.yml` file, the command would be the following:

```
rm -rf /opt/sas/repomirror
```

- 3 On the unconnected mirror repository machine, remove all the files in the location described in the `repo_vars.yml` file as the `httpd_doc_root` value with the following command.


```
rm -rf directory-location
```

For example, if you used the default value for `httpd_doc_root` in the `repo_vars.yml` file, the command would be the following:

```
rm -rf /var/www/html/pulp
```

- 4 (Optional) Remove the `httpd` service from the unconnected repository mirror machine.

Uninstall from the Deployment Target

Uninstalling SAS Viya software from the machines where the software is deployed is the same process that would you use if you did not have mirror repositories. For the process to remove the software, see [“Uninstalling SAS Viya” on page 81..](#)

Appendix 3

Troubleshooting

Troubleshooting SAS Viya	119
SAS Viya Services Do Not Start	119
Error: Nothing to do	119
ERROR: Procedure PCA not found ERROR: Procedure KCLUS not found	120
TimeoutError(error_message)TimeoutError	120
From Any Browser: Your Connection Is Not Private	121
From Google Chrome: Your connection is not private	121
ERROR: Unable to read a key	121
After Upgrade, One or More of the RabbitMQ Nodes Fails to Start Successfully	122
CAS Startup failure post playbook run after changing the casenv_user in vars.yml	123
INTERNAL_SERVER_ERROR Internal Server Error An error occurred. Please contact your system administrator Explanation: An invalid property value for internal.hostnames may cause this error.	123
Project creation failed with: creatingProviderError An unhandled provider creation error was detected. Setting project to failed creation state	124
Some Services Might Not Be Deregistered from Consul	124

Troubleshooting SAS Viya

SAS Viya Services Do Not Start

Explanation

If Consul is deployed, one cause might be that certain SAS Configuration Server (Consul) files are corrupted.

Resolution

- 1 Stop all services.

Note: For information about the order in which to start and stop the services, see [Order for Stopping and Starting Servers and Services](#).

- 2 Delete the `/opt/sas/viya/config/data/consul/checks/` directory
- 3 Restart all services.

Error: Nothing to do

Error

After removing the software and attempting to re-install the software:

Error: Nothing to do

Explanation

The directories that contain the software were deleted. However, the yum remove command was never run. In `/var/log/yum.log`, the last entry for the rpm message is `Installed`.

Resolution

Clean up the yum repository by running the following command.

```
yum remove packagename
```

You can then re-install the software.

ERROR: Procedure PCA not found ERROR: Procedure KCLUS not found

Explanation

The installation was attempted on a system that was not completely cleaned up from a previous installation.

Resolution

Uninstall SAS/CONNECT by running the following command:

```
yum groups mark remove "SAS/CONNECT"
```

Re-install SAS/CONNECT by running the following command:

```
sudo yum groupinstall "SAS/CONNECT"
```

TimeoutError(error_message)TimeoutError

Error

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
```

Explanation

The system failed to gather mount information.

Resolution

Perform one of the following actions:

- Set `/etc/mtab` as a link to `/proc/mounts` by running the following command:

```
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.

From Any Browser: Your Connection Is Not Private

Explanation

The default self-signed certificates are not in the operating system truststore by default. The Apache Web Server is configured to use a certificate that is signed by this Certificate Authority (CA). When you open any SAS URL and navigate to the web server from a machine that does not have this CA in the truststore, you will receive the message `Your connection is not private`. The message does not indicate that there is any problem with the SAS deployment.

Resolution

SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see the Security section of the System Requirements chapter.

From Google Chrome: Your connection is not private

Issue

When attempting to access SAS Viya software from Google Chrome, the following message is displayed:

`Your connection is not private.`

Explanation

If you have previously accessed a website using https, when you access the website again, Google Chrome automatically redirects to https.

Resolution

To reset Google Chrome so that it does not redirect to https:

- 1 In the Chrome address bar, enter this command:

```
chrome://machine-name/#hsts
```

- 2 Under **Query domain**, in the **Domain** box, enter the name of the machine that was used in the URL that you were attempting to access.
- 3 Click **Query** to determine whether the machine is known to the browser.
- 4 If the machine is known to the browser, under **Delete domain**, enter that machine name in the **Domain** box. Click **Delete**.

The corrected URL should now work with the HTTP protocol.

ERROR: Unable to read a key

Issue

When running the deployment, the following message is displayed:

```
fatal: [deployTarget2]: FAILED! =>{"changed": false, "failed": true, "msg":
"Get http://localhost:8500/v1/kv/config/application/rabbitmq/username: dial tcp [::1]:8500:
getsockopt: connection refused\n\
ERROR: Unable to read a key\nGet http://localhost:8500/v1/kv/config/application/rabbitmq/password:
dial tcp [::1]:8500: getsockopt:connection refused\n\
ERROR: Unable to read a key\n"}
```

Explanation

Consul requires each machine to have a single, private IP address. It does not bind to a public IP address by default. A machine target that is specified in your inventory file has one of the following conditions:

- multiple network adapters that have been assigned private IP addresses.

- no private IP address.

Resolution

To confirm the cause of the failure, check the Consul logs for an entry that resembles the following:

```
Starting Consul agent...==> Error starting agent: Failed to get advertise address:
Multiple private IPs found. Please configure one.
```

The resolution is to configure an adapter for the Consul bind parameter in `/etc/sysconfig/sas/sas-viya-consul-default`

Note: This file was installed by the Ansible playbook. This problem can be avoided by specifying the consul bind adapter in the inventory file during deployment.

Locate the following section of the file:

```
# Consul option: -bind
# Specify the desired name of a network interface or IPv4 address.
export CONSUL_BIND_EXTERNAL=adapter-name
```

For *adapter-name*, supply the name of the adapter that Consul should use to locate the machine.

After Upgrade, One or More of the RabbitMQ Nodes Fails to Start Successfully

Error

Three log file will contain the following message:

```
=ERROR REPORT==== 16-Nov-2017::16:50:21 ===
Cluster upgrade needed but other disc nodes shut down after this one.
Please first start the last disc node to shut down.
```

Note: if several disc nodes were shut down simultaneously they may all show this message. In which case, remove the lock file on one of them and start that node. The lock file on this node is:

```
/opt/sas/viya/config/var/lib/rabbitmq-server/mnesia/rabbit@abc.unx.abc.com/nodes_running_at_shutdown
```

Explanation

The RabbitMQ cluster was not stopped or started in the correct order.

Resolution

- 1 Stop and restart the RabbitMQ nodes.
 - a Manually stop all nodes in the reverse order in which they were started during the upgrade.
 - b Manually restart the nodes in the order that is listed in the inventory file's [rabbitmq] target group. You should wait for each rabbitMQ node to start completely before advancing to the next node in the list. If all nodes start successfully, skip to step 4.
- 2 For each rabbitMQ node that failed to start, connect to that target host and remove the lock file. For example::

```
a myhost$ ssh targethost.targetdomain.com
targethost$ sudo rm
/opt/sas/viya/config/var/lib/rabbitmq-server/mnesia/rabbit@<targethost.targetdomain.c
om>/nodes_running_at_shutdown
targethost$ exit
myhost$
```

- b** Manually restart the nodes in the order that is listed in the inventory file's [rabbitmq] target group. You should wait for each rabbitMQ node to start completely before advancing to the next node in the list. If all nodes start successfully, skip to step 4.
- 3** On each failed target host, remove the rabbitMQ Mnesia database as follows:
 - a** On each failed target machine, remove the following directory and all contents:


```
targethost$ sudo rm -R /opt/sas/viya/config/var/lib/rabbitmq-server/mnesia
```
 - b** On each failed target machine, remove the internal rabbit cluster indicator file for SAS Viya:


```
targethost$ sudo rm
/opt/sas/viya/config/var/lib/rabbitmq-server/sasrabbitmq/sas.cluster.configured
```
 - c** If required, reset the rabbitMQ cluster password. As part of the original deployment, you were instructed to change the default RabbitMQ client password. When you remove the Mnesia database, the password is reset to the system default. To change the password for the cluster, on one of the RabbitMQ target machines, run the following command.


```
targethost$ sudo /opt/sas/viya/home/bin
```
- 4** Rerun the playbook to continue the upgrade process after all rabbitMQ service failures have been resolved.

CAS Startup failure post playbook run after changing the casenv_user in vars.yml

Explanation

The administrator has changed the casenv_user, which causes the CAS controller start-up to fail.

Resolution

- 1** Edit the `/opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig_tenant_default` file, where tenant is either "viya" or the tenant name.
- 2** Change the line with `restrictServerLaunch=old user` to `restrictServerLaunch=new user`.
- 3** Rerun the playbook.

INTERNAL_SERVER_ERROR Internal Server Error An error occurred. Please contact your system administrator Explanation: An invalid property value for internal.hostnames may cause this error.

Explanation

You might have an error in `sitedefault.yml` such as an incorrect value for `internal.hostnames`. However, you cannot correct the error and rerun the playbook. The `sitedefault.yml` file is used to set site-based values for properties during an initial deployment. On a subsequent run of the deployment playbook, properties that were previously set are not modified. The `sitedefault.yml` preserves any customer-based modifications to these values. If you rerun the playbook, only `sitedefault.yml` properties that have no value in the environment are applied.

Resolution

SAS Environment Manager is the preferred tool to modify the site-based property values. During deployment, you can also use the `sas-bootstrap-config` command with the `--force` option before you rerun the playbook. To modify the values, the `--force` option is required. Here is an example of how to modify the internal host name:

```
cd /opt/sas/viya/home/bin/
cd /opt/sas/viya/home/bin/
./sas-bootstrap-config --consul
https://localhost:8501 --token-file
../../config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.to
ken kv write config/application/zones/internal.hostnames
correct-value-for-hostname
```

Project creation failed with: creatingProviderError An unhandled provider creation error was detected. Setting project to failed creation state

Explanation

This error is not an issue for the deployment. The configuration bootstrap process will retry until it is successful.

Resolution

Restart the data mining service.

Some Services Might Not Be Deregistered from Consul

Explanation

In SAS Viya 3.2, the following services were supported:

- recipeExecutionProvider
- SASVisualDataBuilder
- data-preparation-plans

In SAS Viya 3.3, these services have been removed or substituted with other microservices. In an upgrade scenario, it is possible that they might not be fully deregistered from Consul. The Consul log will repeatedly record messages about these failing services.

Resolution

To prevent the recording of these errors in the log file, you can manually deregister the services from Consul.

CAUTION! Be sure that you deregister only the following services: recipeExecutionProvider SASVisualDataBuilder data-preparation-plans Removing other services will cause other failures.

To deregister the services, follow these steps:

- 1 Change to the executable directory:

```
cd /opt/sas/viya/home/bin/
```

- 2 To list the services in Consul that are required to be deregistered:

```
./sas-bootstrap-config agent check list | grep -i recipeExecutionProvider
```

The following information is returned:

```
"checkID": "service:recipeexecutionprovider-10-123-4-56",
"name": "Service 'recipeExecutionProvider' check",
"output": "Get http://machine.name.com:43345/recipeExecutionProvider/commons/health:
dial tcp 10.120.4.61:43345: getsockopt: connection refused",
"serviceID": "recipeexecutionprovider-10-123-4-56",
"serviceName": "recipeExecutionProvider",
```


Note: If you have multiple services running on multiple machines, more than one entry will be returned from the preceding command. Each checkID will correspond to the IP address of the machine where the service is running. Each checkID value should be deregistered.

- 3 To remove each of the checkID IP instances that are shown by the agent check list command, use the information from the checkID value in the preceding command output to deregister the health check:

```
./sas-bootstrap-config agent check deregister --id service:recipeexecutionprovider-10-123-4-56
```

- 4 To deregister the service, find out the ID for the service by running the following command:

```
./sas-bootstrap-config agent service list | grep -i recipeExecutionProvider
"recipeexecutionprovider-10-123-4-56": {
  "ID": "recipeexecutionprovider-10-123-4-56",
  "Service": "recipeExecutionProvider",
```

Note: If you have multiple services running on multiple machines, more than one entry will be returned from the preceding command. Each ID will correspond to the IP address of the machine where the service is running. Each ID value should be deregistered.

- 5 To remove each of the IP instances that are shown, use the output from the ID in the preceding command to deregister the service:

```
./sas-bootstrap-config agent service deregister recipeexecutionprovider-10-123-4-56
```

To remove the remaining services, repeat the preceding steps.

To remove SASVisualDataBuilder:

- 1 To list the SASVisualDataBuilder service in Consul that is required to be deregistered:

```
./sas-bootstrap-config agent check list | grep -i SASVisualDataBuilder
"checkID": "service:sasvisualdatabuilder-10-123-4-56",
  "name": "Service 'SASVisualDataBuilder' check",
  "output": "Get http://machine.name.com:46529/SASVisualDataBuilder/commons/health:
dial tcp 10.120.4.61:46529: getsockopt: connection refused",
  "serviceID": "sasvisualdatabuilder-10-123-4-56",
  "serviceName": "SASVisualDataBuilder",
```

- 2 To deregister the health check, use the output from the checkID value in the preceding command:

```
./sas-bootstrap-config agent check deregister --id service:sasvisualdatabuilder-10-123-4-56
```

- 3 To list the VisualDataBuilder service in Consul that is required to be deregistered:

```
./sas-bootstrap-config agent service list | grep -i VisualDataBuilder
"sasvisualdatabuilder-10-123-4-56": {
  "ID": "sasvisualdatabuilder-10-123-4-56",
  "Service": "SASVisualDataBuilder",
```

- 4 To deregister the service, use the output from the ID in the preceding command:

```
./sas-bootstrap-config agent service deregister sasvisualdatabuilder-10-123-4-56
```

To remove data-preparation-plans:

- 1 To list the data-preparation-plans services in Consul that are required to be deregistered:

```
./sas-bootstrap-config agent check list | grep -i data-preparation-plans
"checkID": "service:data-preparation-plans-10-123-4-56",
  "serviceID": "data-preparation-plans-10-123-4-56",
```

- 2 To deregister the health check, use the output from the checkID value in the preceding command:

```
./sas-bootstrap-config agent check deregister --id service:data-preparation-plans-10-123-4-56
```

- 3 To list the data-preparation-plans services in Consul that are required to be deregistered:

```
./sas-bootstrap-config agent service list | grep -i data-preparation-plans
```

```
"data-preparation-plans-10-123-4-56": {  
  "ID": "data-preparation-plans-10-123-4-56",
```

- 4 To deregister the service, use the output from the ID in the preceding command:

```
./sas-bootstrap-config agent service deregister data-preparation-plans-10-123-4-56
```