SAS® Data Agent 2.3 for Linux: Deployment Guide

SAS® Documentation
August 9, 2019

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS® Data Agent 2.3 for Linux: Deployment Guide*. Cary, NC: SAS Institute Inc.

SAS® Data Agent 2.3 for Linux: Deployment Guide

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

August 2019

2.3-P1:dplydagent0phy0lax

Contents

Chapter 1 / Introduction **1**
Steps for a Successful Deployment 1
How Deployment Works 2
Deployment Examples and Guidance 3
SAS Products and Supporting Components 5
Contact SAS Technical Support 6

Chapter 2 / System Requirements **7**
Hardware Requirements 8
Operating System Requirements 10
Server Software Requirements 12
Data Source and Storage Requirements 13
Security Requirements 16
User and Group Requirements 17
Deployment Tools 20

Chapter 3 / Pre-installation Tasks **21**
Create a Mirror Repository 21
Create a Playbook 24
Enable Required Ports 26
Configure SELinux 26
Configure a Proxy Server 26
Enable the Yum Cache 27
Perform Linux Tuning 27
Confirm the Identities of the Hosts 30
(Optional) Enable Key-Based SSH Authentication 30
Install Ansible 31

Chapter 4 / Installation **33**
Edit the Inventory File 33
Modify the vars.yml File 36
Deploy the Software 40
Deployment Logs 41

Chapter 5 / Post-installation **43**
SAS Data Preparation Deployment Tasks: Configure Communication 43
SAS Data Agent Deployment Tasks: Configure Communication 47
SAS Data Agent Deployment Tasks: Configure Data Access 52
Configure a Proxy Server to Communicate with the SAS Data Preparation Deployment 58

Chapter 6 / Validating the Deployment **61**
Validate Round-Trip Communication between SAS Data Preparation and SAS Data Agent 61
Perform Installation Qualification on RPM Packages 62
Verify PostgreSQL 64
Verify the SAS/ACCESS Interface to Your Databases 64

Chapter 7 / Completing the Deployment **65**
Save Snapshot Directory Content 65
Share Important Deployment Information with the Administrators 65

Refer to Additional Documentation . . . 65

Chapter 8 / Managing Your Software . . . **67**
Overview . . . 67
Updating Your SAS Viya Software . . . 68
Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4 . . . 75

Chapter 9 / Uninstalling SAS Viya . . . **79**
What deploy-cleanup Does . . . 79
Use deploy-cleanup . . . 79
Uninstall a Mirror Repository . . . 81

Appendix 1 / Creating High Availability PostgreSQL Clusters . . . **83**
Overview . . . 83
HA PostgreSQL Topologies . . . 83
Set Up a Horizontal Cluster . . . 85
Set Up a Vertical Cluster . . . 87
Set Up a Hybrid Cluster . . . 88
Set Up Multiple Clusters . . . 89
Deployment Logs . . . 92
Verify the Deployment . . . 93

Appendix 2 / Troubleshooting . . . **95**
SAS Viya Services Do Not Start . . . 95
Nothing to Do Dialog . . . 96
PCA and KCLUS Procedures Were Not Found . . . 96
Timeout Dialog . . . 96
From Any Browser: Connection Is Not Private . . . 97
From Google Chrome: Connection Is Not Private . . . 97
Unable to Read a Key . . . 98
Zypper Run Command Failed: Library Is Locked . . . 98
Zypper Run Command Failed: Shared Vault Value Has Not Been Set . . . 99
Connection Reset by Peer Network Problem . . . 99
Internet Connectivity Problems . . . 100
Invalid Host Name in the sitedefault.yml File . . . 101
SAS Data Agent Log Contains Multiple Errors . . . 102
SAS Data Agent Log: File Was Not Found . . . 103
Problems When Using the SAS Data Agent CLI . . . 103
SSL Connection Problems When the Data Agent Tenant Initialization Script is Run . . . 103

1

Introduction

Steps for a Successful Deployment . . . 1
Before You Begin . . . 1
Step 1 — Prepare for the Deployment . . . 2
Step 2 — Perform the Deployment . . . 2
Step 3 — Validate and Complete the Deployment . . . 2
How Deployment Works . . . 2
The Basics . . . 2
Files Used for Deployment . . . 3
Deployment Examples and Guidance . . . 3
Registered to SAS Data Preparation . . . 3
Registered to SAS Data Preparation in a Multi-tenant Environment . . . 4
Cluster for High Availability PostgreSQL . . . 5
SAS Products and Supporting Components . . . 5
Contact SAS Technical Support . . . 6

Steps for a Successful Deployment

Before You Begin

- Because the contents of this guide are subject to continual updates, make sure that you have the latest guide. You can always access the latest release of this guide from the following site:

  SAS Viya Deployment Guides

  If you accessed this guide directly from the Software Order Email (SOE), you are viewing the latest guide. If you are viewing a saved copy of the PDF version of this guide, the content might be outdated.
- To use this guide successfully, you should have a working knowledge of Ansible and the Linux operating system. For more information, see "How Deployment Works" on page 2.
- SAS Data Agent enables data to move from an on-premises environment to SAS Data Preparation that is running on a private cloud or a public cloud. To understand the overall deployment process and multiple deployments of SAS Data Agent, see "Deployment Examples and Guidance" on page 3.
- To configure SAS Data Agent, SAS Data Preparation must also be deployed and operational. For more information about deploying SAS Data Preparation, see SAS Viya for Linux Deployment Guide.
- SAS Data Agent and SAS Data Preparation can be ordered together. If you received an SOE that includes SAS Data Preparation, then SAS Data Agent is also included. Although SAS Data Agent is included in the order, it is not included in the list of software in the SOE.

Step 1 — Prepare for the Deployment

1 Perform one of the following tasks:
   - To update or add software to an existing deployment, go directly to “Managing Your Software” on page 67.
   - To deploy a new SAS Data Agent server, continue with the following the steps.
2 Go to “System Requirements” on page 7 to learn about requirements for hardware, software, data sources, and more.
3 Go to “Pre-installation Tasks” on page 21 to prepare your environment before you deploy the software.

Step 2 — Perform the Deployment

1 Go to “Installation” on page 33 to deploy the software. The steps for running the playbook are included in this section.
2 Go to “Post-installation” on page 43 to register the SAS Data Agent server with SAS Data Preparation and to configure data access.

Step 3 — Validate and Complete the Deployment

1 Go to “Validating the Deployment” on page 61 to verify that the servers were deployed correctly and that SAS can access your data.
2 Go to “Completing the Deployment” on page 65 to learn about post-deployment best practices, including initial administration tasks.

How Deployment Works

The Basics

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

- Each time you run the playbook, Ansible automates a series of commands that securely access the latest software to which you are entitled.
- To use Ansible, you must install it first. In this guide, the machine on which you install Ansible is called the "Ansible controller." The Ansible controller must have SSH access to the machine on which you plan to deploy SAS Data Agent.
- During the post-installation process, SAS Data Agent is registered to SAS Data Preparation. SAS Data Preparation must be deployed and operational before you can register SAS Data Agent to SAS Data Preparation. Because SAS Data Preparation and SAS Data Agent are deployed to different machines, you must perform tasks on both machines to configure communications.

  For an overview of registering SAS Data Agent to SAS Data Preparation, see "Deployment Examples and Guidance" on page 3.

Files Used for Deployment

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

File	Purpose
inventory.ini	You edit the inventory.ini file to map machines (or hosts) to the software components, which are represented as host groups within the inventory.ini file.
vars.yml	The vars.yml file includes the variables that enable you to customize your deployment.
sitedefault.yml (optional)	Typically, the sitedefault.xml is not used for the initial deployment. The sitedefault.yml file contains variables for more advanced implementations, such as setting up a high availability PostgreSQL cluster.

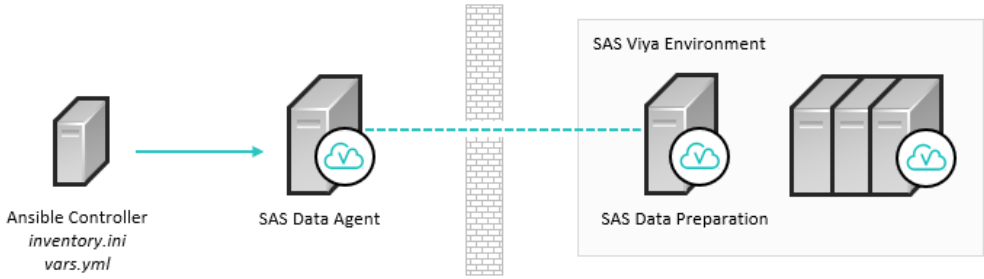
Deployment Examples and Guidance

Registered to SAS Data Preparation

SAS Data Agent enables data to move from an on-premises environment to SAS Data Preparation on SAS Viya. SAS Data Agent is deployed when data must pass through a firewall to access SAS Data Preparation. SAS Data Agent is optimized to connect to SAS Data Preparation that is running on a private cloud or a public cloud. To complete the configuration, tasks are required for both the SAS Data Agent machine and the SAS Data Preparation machine.

The following example shows a single SAS Data Agent server that is deployed using Ansible and registered to communicate with SAS Data Preparation.

Figure 1.1 *SAS Data Agent Deployed and Registered with SAS Data Preparation*



Note: You can register multiple SAS Data Agent servers with SAS Data Preparation. Each SAS Data Agent server can be registered with only one instance of SAS Data Preparation. In this guide, a single instance of SAS Data Agent is referred to as a "single-tenant SAS Data Preparation environment."

Registered to SAS Data Preparation in a Multi-tenant Environment

If SAS Data Preparation is deployed within a SAS Viya multi-tenant environment, you can register one or more SAS Data Agent servers to communicate with each tenant. Each SAS Data Agent server can be registered to only one tenant. In other words, you do not have to register a SAS Data Agent server to every tenant.

The following example provides an overview of deploying and registering multiple SAS Data Agent servers to a SAS Viya multi-tenant environment that includes SAS Data Preparation.

Figure 1.2 *SAS Data Agent Servers Registered to Different Tenants*

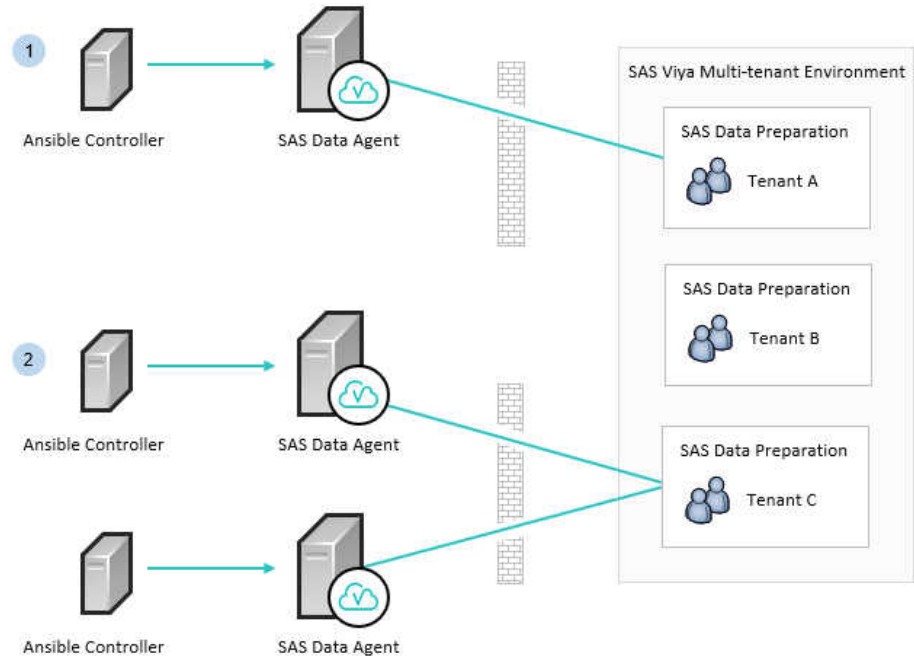


Table 1.1 *SAS Data Agent Servers Registered to Different Tenants*

1	SAS Data Agent is deployed and then registered to tenant A, which includes SAS Data Preparation.
2	Two SAS Data Agent servers are deployed and then registered to tenant C, which includes SAS Data Preparation. Notice that a SAS Data Agent server is not registered to tenant B.

Note: The steps to deploy a SAS Data Agent server and to register it to SAS Data Preparation are provided in this guide. To achieve a SAS Viya multi-tenant environment that includes SAS Data Preparation, see the following documentation:

- To deploy a SAS Viya multi-tenant environment, see SAS Viya for Linux: Deployment Guide.
- To onboard a tenant, see SAS Viya Administration: Multi-tenancy.

Cluster for High Availability PostgreSQL

SAS Data Agent uses a PostgreSQL database to store user content and preferences. By default, Ansible deploys PostgreSQL as a single node on a single machine. The standard deployment consists of one PGPool and one PostgreSQL data node. However, you can deploy a High Availability (HA) PostgreSQL cluster to achieve higher performance and to support redundancy. For more information, see "Creating High Availability PostgreSQL Clusters" on page 83.

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

Figure 1.3 *High Availability PostgreSQL Horizontal Cluster*

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:

SAS Data Agent 2.3	SAS/ACCESS Interface to ODBC (on SAS Viya)
SAS/ACCESS Interface to Amazon Redshift (on SAS Viya)	SAS/ACCESS Interface to Oracle (on SAS Viya)
SAS/ACCESS Interface to DB2 (on SAS Viya)	SAS/ACCESS Interface to PostgreSQL (on SAS Viya)
SAS/ACCESS Interface to Hadoop (on SAS Viya)	SAS/ACCESS Interface to SAP HANA (on SAS Viya)
SAS/ACCESS Interface to Microsoft SQL Server (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.

Contact SAS Technical Support

Technical support is available to all customers who license SAS software. However, you are encouraged to engage your designated on-site SAS support personnel as your first support contact. If your on-site SAS support personnel cannot resolve your issue, have them contact SAS Technical Support to report your problem.

Before you contact SAS Technical Support, explore the SAS Support website at support.sas.com/techsup/. This site offers access to the SAS Knowledge Base, as well as SAS communities, Technical Support contact options, and other support materials that might answer your questions.

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

2

System Requirements

Hardware Requirements **8**
- Host Requirements 8
- General Hardware Considerations 8
- File System and Storage Requirements 9

Operating System Requirements **10**
- Supported Operating Systems 10
- Linux Requirements 10
- Additional Requirements for Red Hat Enterprise Linux and Oracle Linux 11
- Additional Requirements for SUSE Linux 12
- SAS Support for Alternative Operating Systems 12

Server Software Requirements **12**
- Java 12
- Apache httpd 13

Data Source and Storage Requirements **13**
- Supported Data Sources 13
- Requirements for SAS/ACCESS Interface to Amazon Redshift 14
- Requirements for SAS/ACCESS Interface to DB2 14
- Requirements for SAS/ACCESS Interface to Hadoop 14
- Requirements for SAS/ACCESS Interface to Microsoft SQL Server 14
- Requirements for SAS/ACCESS Interface to ODBC 15
- Requirements for SAS/ACCESS Interface to Oracle 15
- Requirements for SAS/ACCESS Interface to PostgreSQL 15
- Requirements for SAS/ACCESS Interface to SAP HANA 16
- Requirements for SAS/ACCESS Interface to Teradata 16

Security Requirements **16**
- Transport Layer Security 16

User and Group Requirements **17**
- Overview: User Accounts 17
- Set Up the User Account that Deploys the Software 18
- Set Up User Accounts for SAS Data Agent Users 18
- User Accounts (Reference) 18
- Services that Require Root Privileges 19

Deployment Tools **20**
- Ansible Controller Requirements 20

Hardware Requirements

Host Requirements

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

- A static IP address

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

  Important: On the Linux machine that will host the CAS server, make sure that the host name in `/etc/hosts` is specified in all lowercase letters. If you change the host name to comply with this requirement, verify that the image is stable and that all other services are working correctly before you start the deployment process.

- A host name that can be resolved to an IP address

  Both the fully qualified domain name (FQDN) and IP address of each machine in the deployment should be added to their `/etc/hosts` files to enable reverse DNS resolution.

- An FQDN that is 64 characters or fewer in length

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

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

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

General Hardware Considerations

SAS strongly recommends consulting with a sizing expert to obtain an official hardware recommendation that is based on your deployment topology, the estimated SAS workload, and the number of users. To request sizing expertise, contact your SAS account representative. If you need assistance in determining your SAS account representative, send an email to contactcenter@sas.com.

CPU and RAM Recommendations

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

Consider the following as you prepare for the deployment process:

- The hardware guidelines in this guide reflect baseline standards. For a production environment, CPU, RAM, and disk resources should be increased after the expected amount of data to be processed and number of concurrent users are taken into consideration.
- Overall system performance will improve with the addition of both RAM and CPU cores.
- Test machines were equipped with RAM that had a minimum memory clock speed of 1600 MHz.

Baseline Hardware Requirements

The following table indicates the minimum RAM and number of CPU cores that are required to support SAS Data Agent on a single machine.

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

Additional RAM should be added based on the expected amount of data that will be processed. More resources are required for multiple-user, production-scale deployments that use large data sets.

Table 2.1 *Minimum Hardware Requirements for SAS Data Agent*

Product	RAM (GB)	CPU Cores
SAS Data Agent	8	4

The playbook installs executables and creates configuration directories in **/opt/sas/**. The minimum available disk space that is required to install and start SAS Viya and SAS Data Agent is less than 8 GB. However, logs and operational data can grow to exceed that amount. Therefore, the actual space that is required will depend on the amount of data and the level of activity in your specific deployment. For more information, see “File System and Storage Requirements” on page 9.

File System and Storage Requirements

Disk Space Considerations

Verify that at least 48 GB of disk space are available for your SAS Data Agent installation. The installation files are automatically downloaded to the **/var/cache/yum** directory.

The software is installed in the **/opt** directory on each target machine. In many cases, this directory is in a file system with 50 GB or fewer of disk space. To increase available disk space for the installation, SAS recommends that you mount additional volumes at **/opt/sas** instead of to a subdirectory of **/opt/sas**. Mounting a volume in the installation directories increases the difficulty of uninstalling the software or of moving the volume to another location at a later time.

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.

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

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

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

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

Operating System Requirements

Supported Operating Systems

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

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

Linux Requirements

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

Libraries and Packages

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

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

- acl-2.2 or later

  The acl package is installed with Red Hat Enterprise Linux by default. For SUSE Linux, it is available in the base repositories.

- curl-7.19.7-53 or later

  On Red Hat Enterprise Linux 6.7 and later within 6.**x**, apply the RHSA-2017:0847-01 security update for curl to ensure that you have a supported version of the utility.

  On Oracle Linux 6.7 and later within 6.**x**, apply the ELSA-2017-0847 security update.

  Red Hat Enterprise Linux 7.**x**, Oracle Linux 7.**x**, and SUSE Linux 12.2 have a supported version of curl by default.

- glibc-2.12-1.166.el6 and later (on Red Hat Enterprise Linux 6.*x* or the equivalent). Refer to RHBA-2015:1465 on the Red Hat Customer Portal to obtain the latest updated package list.

  glibc-2.17-107.el7 and later (on Red Hat Enterprise Linux 7.*x* or the equivalent). Refer to RHSA-2016:2573 on the Red Hat Customer Portal to obtain the latest updated package list.

  glibc-2.22 and later (on SUSE Linux)

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

  libpng12 (on Red Hat Enterprise Linux 7.*x*, Oracle Linux 7.*x*, or SUSE Linux)

- libXp

  Note: For SUSE Linux, the package is named libXpm4.

- libXmu

- net-tools

- the numactl package

- systemd version 219-30 or later
- the X11/Xmotif (GUI) packages
- xterm

Verifying systemd

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

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

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

```
yum update systemd
```

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

```
zypper update systemd
```

Disabling the requiretty Setting

On some versions of Red Hat Enterprise Linux, a default setting causes errors with selected SAS Data Agent start-up scripts. Scripts that execute as root or with sudoers permissions cannot run if the default `requiretty` setting is enabled.

To avoid these errors, take one of the following steps:

- Disable requiretty entirely by removing or commenting out the following line in the **`/etc/sudoers`** file:

  ```
  Defaults   requiretty
  ```

- Disable requiretty for the root user. Add the following line to the **`/etc/sudoers`** file:

  ```
  Defaults:root    !requiretty
  ```

Additional Requirements for Red Hat Enterprise Linux and Oracle Linux

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

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

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

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

```
subscription-manager version
```

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

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

A list of active subscriptions is returned.

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

```
uln_register
```

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

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.

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

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

Additional Requirements for SUSE Linux

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

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

```
sudo zypper in which
```

SAS Support for Alternative Operating Systems

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

Server Software Requirements

Java

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

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

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

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

```
java -version
```

Apache httpd

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

SAS recommends that you install Apache httpd and configure the Apache HTTP Server to use certificates that comply with the security policies at your enterprise before you start the deployment process. The playbook will automatically configure the certificates to secure the server. For more information, see "Enhance Default Security Settings" on page 17.

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 must be dedicated to a single SAS Viya deployment.

Data Source and Storage Requirements

Supported Data Sources

SAS Data Agent supports the following external data sources, which require a SAS/ACCESS product. In some cases, these products might have individual requirements:

- Amazon Redshift
- Apache Hive
- IBM DB2
- Microsoft SQL Server
- Data sources that are accessible with an ODBC driver
- Oracle
- PostgreSQL
- SAP HANA
- Teradata

SAS Data Agent also supports CSV files, which do not require a SAS/ACCESS product and can be accessed directly.

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

Requirements for SAS/ACCESS Interface to Amazon Redshift

SAS/ACCESS Interface to Amazon Redshift (on SAS Viya) includes SAS Data Connector to Amazon Redshift. It also includes a required ODBC driver.

The required client software is installed automatically. In order to reference a Data Source Name (DSN) in your connections, some post-installation configuration is required. For more information, see “Configure SAS/ACCESS to Amazon Redshift” on page 52.

Requirements for SAS/ACCESS Interface to DB2

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

IBM DB2 Connect™ must also be licensed if you plan to connect to IBM DB2 databases that are running on AS/400, VSE, VM, MVS, and z/OS systems. The following DBMS products are supported:

- IBM DB2 version 10.5 or later
- Client utilities for IBM DB2 version 10.5 or later

SAS recommends installing the latest FixPack on the client and server.

Some post-installation configuration is required in order to register the SAS Data Agent Server. For more information, see “Configure SAS/ACCESS Interface to DB2” on page 53.

Requirements for SAS/ACCESS Interface to Hadoop

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

SAS Data Agent supports the following Hadoop third-party distributions:

- Cloudera CDH 5.5 and later releases
- Hortonworks HDP 2.4 and later releases
- MapR 5.2 and later releases
- Amazon Elastic MapReduce

Post-installation configuration is required. For more information, see “Configure SAS/ACCESS Interface to Hadoop” on page 53.

Requirements for SAS/ACCESS Interface to Microsoft SQL Server

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

SAS/ACCESS Interface to Microsoft SQL Server supports the following Microsoft products:

- Microsoft Azure SQL Database
- Microsoft SQL Server 2012 or later

SAS/ACCESS to Microsoft SQL Server also supports the following cloud variants of Microsoft SQL Server:

- Amazon RDS Microsoft SQL Server (Microsoft SQL Server 2012 or later)
- Microsoft Azure SQL Database

The client software is installed automatically along with SAS/ACCESS Interface to Microsoft SQL Server.

If your deployment requires encryption, be aware that the SSL library is not included with SAS/ACCESS Interface to Microsoft SQL Server. You can determine whether the SSL library is installed on your machine by running the following command:

```
locate libssl.so | xargs ls -al
```

If required, install the OpenSSL library from https://www.openssl.org.

Post-installation configuration might be required. For more information, see “Configure SAS/ACCESS Interface to Microsoft SQL Server” on page 54.

Requirements for SAS/ACCESS Interface to ODBC

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

Before you can use SAS Data Agent with ODBC, an ODBC driver is required for the data source from which you want to access data. ODBC drivers are often available from DBMS vendors and other third-party ODBC driver developers. Your ODBC driver must comply with the ODBC 3.5 (or later) specification.

Note: The ODBC driver that you select might require additional DBMS software in order to enable network access.

Some post-installation configuration is required. For more information, see “Configure SAS/ACCESS Interface to ODBC” on page 55.

Requirements for SAS/ACCESS Interface to Oracle

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

SAS Data Agent requires the following Oracle components:

- Oracle Database 11gR2 or Oracle Server version 12c
- Oracle Client 11gR2 or Oracle Client version 12c (64-bit libraries)

SAS/ACCESS Interface to Oracle supports the following cloud variants of Oracle:

- Amazon RDS Oracle (11gR2 or 12c)
- Oracle Cloud Platform (11gR2 or 12c)

Post-installation configuration is required. For more information, see “Configure SAS/ACCESS Interface to Oracle” on page 56.

Requirements for SAS/ACCESS Interface to PostgreSQL

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

SAS Data Agent supports PostgreSQL Database version 9.4.4 or a later version.

It also supports the following cloud variants of PostgreSQL:

- Amazon Aurora (PostgreSQL engine version 9.6 or later)
- Amazon RDS PostgreSQL (Engine version 9.6 or later)

SAS Data Agent requires a driver manager and an ODBC driver for PostgreSQL. SAS provides both of these ODBC client components and installs them automatically. In order to reference a Data Source Name (DSN) in your connections, post-installation configuration is required. For more information, see “Configure SAS/ACCESS Interface to PostgreSQL” on page 56.

Requirements for SAS/ACCESS Interface to SAP HANA

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

SAS/ACCESS Interface to SAP HANA requires the ODBC driver (64-bit) for SAP HANA from SAP. This driver is part of the SAP HANA Client. The following SAP products are also required:

- SAP HANA SPS 11 Server or later
- SAP HANA ODBC Client for SPS 11 or later

Some post-installation configuration is required. For more information, see “Configure SAS/ACCESS Interface to SAP HANA” on page 57.

Requirements for SAS/ACCESS Interface to Teradata

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

SAS Data Agent supports the following products:

- Teradata Database 15.10 or later
- Teradata CLIv2 client libraries, TTU 15.10 or later for Linux (64-bit libraries)

SAS/ACCESS Interface to Teradata supports Teradata Database 15.10 or later on the following cloud platforms:

- Teradata IntelliCloud
- Amazon Web Services
- Microsoft Azure
- VMware

Post-installation configuration is required. For more information, see “Configure SAS/ACCESS Interface to Teradata” on page 58.

Security Requirements

Transport Layer Security

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

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

How Default Security Is Applied

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

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

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

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

Enhance Default Security Settings

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

1. Install the Apache httpd module and the Apache mod_ssl module on all the web servers in your environment.
2. Add certificates that conform to the policies at your enterprise.
3. Specify the location of the intermediate certificates and the root CA when you edit the playbook. For more information, see “Specify the Path to Certificates” on page 38.

The playbook can then enhance the security of your software deployment automatically. It detects the CA chain that is configured for mod_ssl and incorporates it into the truststores for all other machines in your deployment. On machines that are targets for Consul deployment, the playbook performs additional security configuration.

(Optional) You can also perform these actions after the playbook has been run:

- Block external connections to port 80.
- Use HTTPS for access to the user interfaces from a web browser.
- Add custom certificates to the self-signed certificates that the playbook provides on all machines.
- Upgrade the security protocol and ciphers that are enabled by default using the sas-ssl.conf file.
- Prevent administrators from altering the default permissions on subdirectories of `opt/sas/viya`. Use your preferred network monitoring or security tool to monitor permissions on subdirectories of `opt/sas/viya` after the deployment has completed.

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

User and Group Requirements

Overview: User Accounts

In addition to an installation account with sudoers privileges, SAS Data Agent requires one service account. This service account owns critical files and is used to run various processes.

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

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

Set Up the User Account that Deploys the Software

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

- Super user (sudo) or root access.

  To verify that your user ID is included in the sudoers file, run the following command:

  ```
  sudo -v
  ```

  As an alternative, to verify your sudoers privileges, run this command:

  ```
  sudo -l
  ```

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

  This user account must be able to access the root and sas accounts as “sudo”.

- Appropriate permissions to create subdirectories in the directory where you saved the playbook. The recommended path is **`/sas/install/sas_viya_playbook`**. For more information, see “Store the Playbook” on page 25.

- A home directory.

Set Up User Accounts for SAS Data Agent Users

Additional user accounts are required in order to configure and run the software after the deployment process has completed.

The following requirements apply to the user accounts that can access SAS Data Agent:

- Each user must be able to authenticate to the LDAP provider.
- To administer SAS Data Agent, users must be included in the Data Agent Administrators group.

  Adding users to SAS Data Agent groups is a post-deployment task.

- To access SAS Data Agent features, these users must be included in the Data Agent Power Users group.

User Accounts (Reference)

This section provides reference information about user accounts that are required for SAS Data Agent. The table identifies and describes these accounts. Because these accounts are required for the installation and for running services during the product’s normal operation, do not delete them or change their names. These user accounts do not require root or sudo privileges.

Default Account Name and Group	Description	Purpose
sas; member of sas group	A service account without user restrictions. A login shell is required. No password. You can add a password after installation, if necessary, but make sure that it does not expire. The default user name is required. The sas group is an administration group, not a general user group.	Required for the installation, and created automatically. The installation process sets user and group ownership permissions on all the installation files. This user must exist to enable ownership. After the installation has completed, this user account enables required components to run. The sas group is intended to allow access to administrative features, such as logs and backup. It is the group owner of many files on disk. Restrict membership in this group to administrators.
sasboot	Created during the deployment, with an expired password. After the deployment has completed, use this account to log on to the SAS visual interface in order to configure the connection to your identity provider and to set up user accounts. The sasboot account is typically not used after that. However, it provides an indirect login option in case your identity provider becomes unavailable. The sasboot user is internal only to SAS. It does not exist on a host or in LDAP. For more information, see *SAS Viya Administration: Identity Management*.	Administrator account that is used for preliminary logon to the administration interface.

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

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

- apache
- postgres

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

Services that Require Root Privileges

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

- identsvcs and launchsvcs—Authorize and perform the launching of the CAS server. These services must run as root because, on Linux, the root identity is required in order to start a running process under a different identity. The launchsvcs process creates a CAS session under the identity of the user who submitted the request. The identsvcs process authenticates users when they attempt to connect to a CAS server with a username and password using PAM.
- consul-template—Supports SAS Configuration Server, which is based on HashiCorp Consul. SAS Configuration Server is a registry that contains service configuration data and status information. The consul-

template process extracts configuration change data from the server and updates the appropriate service configuration file.

- vault—Supports SAS Secrets Manager, which is based on HashiCorp Vault. It stores and generates secrets such as certificates.
- RabbitMQ—Supports SAS Message Broker, which is a message service that is based on Pivotal RabbitMQ. SAS Message Broker manages and routes messages among SAS Viya components.
- Apache httpd—Supports the Apache HTTP Server, which provides security and load balancing for multiple SAS Viya components.

Deployment Tools

Ansible Controller Requirements

A typical Ansible deployment consists of at least one control machine (the Ansible controller) and multiple Ansible managed nodes (the machines where SAS software is installed). In a single-machine deployment, Ansible and all SAS software are installed on the Ansible controller. For more information, see “Install Ansible” on page 31.

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

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

3

Pre-installation Tasks

Create a Mirror Repository **21**
SAS Mirror Manager and the Mirror Repository 22
Using SAS Mirror Manager with a Proxy Server 23
Specify a Log Location 23

Create a Playbook **24**
Download the SAS Orchestration CLI 24
Create a Playbook with the SAS Orchestration CLI 24
Store the Playbook 25

Enable Required Ports **26**
Ports to Be Made Available 26

Configure SELinux **26**

Configure a Proxy Server **26**
Overview 26
Using curl 27
Using yum 27

Enable the Yum Cache **27**

Perform Linux Tuning **27**
Set the ulimit Values 28
Set the Semaphore Values 29
Change the Default Time-outs 29
(SUSE Linux Only) Change the Maximum Number of Operating System Tasks 30

Confirm the Identities of the Hosts **30**

(Optional) Enable Key-Based SSH Authentication **30**

Install Ansible **31**
Standard Ansible Installation 31
Streamlined Ansible Installation for Red Hat Enterprise Linux and Equivalent Distributions 31
Streamlined Ansible Installation for SUSE Linux 32
Test Your Ansible Installation 32

Create a Mirror Repository

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

SAS Mirror Manager and the Mirror Repository

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

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

To create a mirror repository with SAS Mirror Manager:

1. The Software Order Email (SOE) indicated that you should save the SAS_Viya_deployment_data.zip file attachment. If you have not already saved the file, save it now.
2. Download SAS Mirror Manager from the SAS Mirror Manager download site to the machine where you want to create your mirror repository. If you use Internet Explorer to download the Linux or Macintosh version, save the file as a .tgz file instead of a .gz file.

   Note: This step requires internet connectivity. If you receive warnings or errors regarding connectivity, see "Internet Connectivity Problems" on page 100.
3. Uncompress the downloaded file.
4. Run the following command:

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

   ./mirrormgr mirror --deployment-data *path-to-deployment-zip-file-from-SOE* --platform *Linux-distribution-value* --latest

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

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

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

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

   The **sas_repos** directories are explained as follows:

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

- deb contains APT files for Debian.

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

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

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

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

Using SAS Mirror Manager with a Proxy Server

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

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

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

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

Example 1: An HTTPS site.

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

Example 2: HTTPS with the certificate location.

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

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

Example 3: An HTTP site.

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

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

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

Specify a Log Location

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

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

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

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

Create a Playbook

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

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

Download the SAS Orchestration CLI

1. The SOE indicated that you should save the SAS_Viya_deployment_data.zip file attachment. If you have not already done so, save that file now.
2. Go to the SAS Orchestration CLI download site and download the SAS Orchestration CLI for the operating system where you stored the ZIP file. The SOE recommended that you save the ZIP file to a machine that runs Linux, which is where you will install your SAS Viya software. However, you can also store it on a machine that runs Macintosh or Windows. If you use Internet Explorer to download the Linux or Macintosh version, save the file as a .tgz file instead of a .gz file.

   Note: This step requires internet connectivity. If you receive warnings or errors regarding connectivity, see "Internet Connectivity Problems" on page 100.
3. Uncompress the .tgz file (Linux and Macintosh) or .zip file (Windows) in the same location where you downloaded it. The result is a file named sas-orchestration on Linux or Macintosh or a file named sas-orchestration.exe on Windows.

Create a Playbook with the SAS Orchestration CLI

Basic Command

To create a playbook, use the command that is appropriate for the operating system where the SAS Orchestration CLI is located.

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

Linux or Macintosh

```
./sas-orchestration build --input location-of-ZIP-file-including-file-name --platform
deployment-platform-tag --deployment-type data-agent-on-premises
```

Windows

```
.\sas-orchestration.exe build --input location-of-ZIP-file-including-file-name --platform
deployment-platform-tag --deployment-type data-agent-on-premises
```

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

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

Options

Use a Proxy Server

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

```
--java-option "-Dhttps.proxyHost=proxy-server-IP-address-or-host-name"
```

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

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

If you use both options, they should not be combined into a single option. The following is an example of using both options on a Linux machine:

```
./sas-orchestration --java-option "-Dhttps.proxyHost=my.proxy.com" --java-option "-Dhttps.proxyPort=1111"
build --input /tmp/SAS_Viya_deployment_data.zip --deployment-type data-agent-on-premises
```

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

Use a Mirror Repository

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

```
./sas-orchestration build --input /sas/install/SAS_Viya_deployment_data.zip --platform redhat
--repository-warehouse "URL-to-mirror-repository-content" --deployment-type data-agent-on-premises
```

Here is an example:

```
.\sas-orchestration build --input c:\sas\install\SAS_Viya_deployment_data.zip
--repository-warehouse "file:///sas_repos" --deployment-type data-agent-on-premises
```

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

For more information about SAS Mirror Manager, see “Create a Mirror Repository” on page 21.

Help with the Options

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

Linux or Macintosh

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

Windows

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

Store the Playbook

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

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

```
tar xf SAS_Viya_playbook.tgz
```

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

Enable Required Ports

The following ports are used by SAS Viya and should be available before you begin to deploy your software. The same ports should also be available for any firewalls that are configured on the operating system or the network.

Table 3.1 *Ports to Be Made Available*

Process	Required Port	Requires Allowed Inbound Traffic From
Apache HTTP Server	443 (external)	SAS Data Agent
PostgreSQL	5431	SAS Data Agent
FSNet port	25141	SAS Data Agent
SAS Data Agent port	26301	SAS Data Agent

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
```

If you get a message that the command is not enabled, you do not have SELinux, so no action is required.

Configure a Proxy Server

Overview

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

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

Using curl

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

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

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

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

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

Using yum

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

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

Enable the Yum Cache

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

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

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

```
keepcache = 1
```

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

Perform Linux Tuning

This section describes tuning that should be performed on your Linux machines before you deploy your software. For information about tuning that can be performed after you deploy your software, see Tuning the Linux Operating System.

Set the ulimit Values

Overview

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

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

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

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

```
*     -     nofile     value
```

The asterisk (*) indicates all user accounts.

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

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

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

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

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

Set the Maximum Number of Open File Descriptors and Stack Size

For each machine in your deployment:

1. Open the **/etc/security/limits.conf** file.
2. Set the limit for open file descriptors as follows:
   - If PostgreSQL will be deployed on the machine, set the limit (using the nofile item) to 150000 for the sas user.

     ```
     sas     -     nofile     150000
     ```

   - If you are deploying SAS Visual Investigator or SAS Intelligence and Investigation Management and the machine is running Elasticsearch, set the limit to at least 65536 for the sas user.

     ```
     sas     -     nofile     65536
     ```

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

     ```
     *     -     nofile     48000
     ```

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

   ```
   *     -     nofile     150000
   ```

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

   ```
   sas     -     stack     10240
   ```

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

4 Save and close the **/etc/security/limits.conf** file.

Set the Maximum Number of Processes Available

For each machine in your deployment:

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

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

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

  ```
  sas     -     nproc     100000
  ```

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

  ```
  *     -     nproc     65536
  ```

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

```
*     -     nproc     100000
```

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

Set the Semaphore Values

For each machine on which PostgreSQL will be deployed.

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

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

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

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

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

```
sudo sysctl -p
```

Change the Default Time-outs

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

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

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

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

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

These commands allow the user to run 50000 tasks concurrently.

Confirm the Identities of the Hosts

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

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

(Optional) Enable Key-Based SSH Authentication

Note: 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.

Install Ansible

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

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.

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

Streamlined Ansible Installation for Red Hat Enterprise Linux and Equivalent Distributions

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

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

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

```
## 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 Because EPEL will no longer be required, you can remove it with the following command:

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

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

Specific versions of Python modules are required. Here are some examples:

```
sudo pip install --upgrade pip==9.0.3
sudo pip install pycparser==2.14
sudo pip install idna==2.7
```

For Python 2.7 (and later within 2.7.x):

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

5 To install a specific version of Ansible through PIP:

```
sudo pip install ansible==2.7.12
```

Streamlined Ansible Installation for SUSE Linux

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

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

1 To install Python's setup tools:

```
sudo zypper install python-setuptools
```

2 To Install Python PIP:

```
sudo easy_install pip
```

3 To install a specific version of Ansible through PIP:

```
sudo pip install ansible==2.7.12
```

Test Your Ansible Installation

1 To test the Ansible version:

```
ansible --version
```

Here is an example of successful output:

```
ansible 2.7.12
  config file =
  configured module search path = Default w/o overrides
  python version = 2.7.15 (default, May  14 2018, 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"
}
```

4

Installation

Edit the Inventory File . . . 33
Overview . . . 33
Multiple inventory.ini Files . . . 34
Update the File . . . 34
Modify the vars.yml File . . . 36
Multiple vars.yml Files . . . 36
Set the Deployment Label . . . 36
Set the Pre-deployment Validation Parameters . . . 37
Specify Security Settings . . . 38
Specify the Path to Certificates . . . 38
Change the Repository Warehouse . . . 39
Define Multiple Invocations . . . 39
(Optional) Specify JRE . . . 39
Deploy the Software . . . 40
Assessment Test . . . 40
Deployment Command . . . 40
Install Only . . . 41
Run from a Directory Other Than the Default . . . 41
Successful Playbook Execution . . . 41
Retry a Failed Deployment . . . 41
Deployment Logs . . . 41

Edit the Inventory File

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`**.

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.

Multiple inventory.ini Files

If you deploy SAS Data Agent more than once, you must create and maintain a separate inventory.ini file for each deployment you perform. To do so, copy the inventory.ini file from the uncompressed playbook and paste it in the same location with a name that is different but meaningful for the specific deployment of SAS Data Agent.

Update the File

Specify the Machines in the Deployment

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

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

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

Table 4.1 *Descriptions of Components of the Deployment Target Reference*

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

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

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

```
deployTarget ansible_connection=local
```

The following example specified the deployment target when SAS Data Agent will be deployed on a machine that is not running Ansible:

```
deployTarget ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
```

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 is an example that uses the parameter:

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

To specify a machine address that is used by other machines in the deployment, add the `internal_deployment_ipv4_override` parameter to the deployment target reference. Using such a parameter ensures that intra-deployment connections to that machine route through the preferred address. Here is an example:

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

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

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

Assign the Target Machines to Host Groups

The second section in the inventory file is used to assign deployment targets to each host group. Under each group, assign machines to the group by using the appropriate alias. In most cases the only change that should be made is to change the machine name to match the one you used in the deployment target reference.

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

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

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

```
[CommandLine]
deployTarget

[DataAgent]
deployTarget

[consul]
deployTarget

[httpproxy]
deployTarget

[pgpoolc]
deployTarget

[sasdatasvrc]
deployTarget
```

```
[sas_all:children]
CommandLine
DataAgent
consul
httpproxy
pgpoolc
sasdatasvrc
```

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.
- Ensure that any machine you place in the [DataAgent] host group is also listed in the [consul] host group.
- 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. To deploy HA PostgreSQL with multiple nodes, see "Creating High Availability PostgreSQL Clusters" on page 83.

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. It is found in trhe top level of the uncompressed playbook (if you have used the defaults, the location is **`/sas/install/sas_viya_playbook/vars.yml`**). Note that all entries in the vars.yml file are case-sensitive.

Note: There are more variables in the vars.yml file than are described in this section. If a variable in the vars,yml file is not described in this section, you should make no changes for it.

Multiple vars.yml Files

If you deploy SAS Data Agent more than once and require different values for any of the variables in the vars.yml file for the separate deployments, you must create and maintain a separate vars.yml file for each variation you will need. To do so, copy the vars.yml file from the uncompressed playbook and paste it in the same location with a name that is different but meaningful for the specific deployment of SAS Data Agent.

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: "va-04april2017"
```

Note: Do not change the value of DEPLOYMENT_LABEL after the software has been deployed, including when performing an update.

Set the Pre-deployment Validation Parameters

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

Check the Ansible Version

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

Note: For information about supported Ansible versions, see "Ansible Controller Requirements" on page 20.

Verify Machine Properties

The playbook checks each machine in the deployment to ensure that the necessary conditions for deployment are met. If any of the following conditions is not met, a warning is given and the playbook stops 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 36.

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

3. 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 34.

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

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

Create and Verify sas User and sas Group

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

Verify System Requirements

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

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

   Note: For more information about setting the SELinux mode, see "Configure SELinux" on page 26.

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

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

   Note: For more information about assigning packages to machines, see "Specify the Machines in the Deployment" on page 34.

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

   Note: For more information about setting ulimits, see "Set the ulimit Values" on page 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 **true**. Only the HTTPS port will be available after the software is deployed.

Specify the Path to Certificates

Note: By default, when SAS Viya is deployed, it 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'
  ```

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

3 Save and close the vars.yml file.

Change the Repository Warehouse

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

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

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

Define Multiple Invocations

The `INVOCATION_VARIABLES` block is used to set the parameters of a High Availability (HA) PostgreSQL cluster of more than one machine. For details, see "Creating High Availability PostgreSQL Clusters" on page 83.

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

Deploy the Software

Assessment Test

Before you deploy the software, SAS recommends that you run the following command to assess the readiness of your system for deployment. Before running the command, ensure that you are at the top level of the playbook in the **sas_viya_playbook** directory.

Note: The command should be run as a root or sudoer user.

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

Add an option based on the password requirements for the user ID that performs the command:

Table 4.2 *Command Options Based on Password Requirements*

Password Requirements	Option
Does not require passwords	use the command as written
Requires a sudo password only	**--ask-become-pass**
Requires an SSH password only	**--ask-pass**
Requires both a sudo and an SSH password	**--ask-pass --ask-become-pass**

If you receive an unexpected error, run the following command to ensure that you are using a supported version of Ansible.

```
ansible-playbook --version
```

Note: For information about supported Ansible versions, see "Ansible Controller Requirements" on page 20.

If you are using a supported version of Ansible and still receive errors from the system assessment, fix those errors before you run the deployment command.

Deployment Command

Ensure that you are at the top level of the playbook in the **sas_viya_playbook** directory. Here is the basic syntax for the command to run the playbook and deploy the software:

Note: The command should be run as a root or sudoer user.

```
ansible-playbook site.yml [ option ]
```

If you are deploying SAS Data Agent with a vars.yml or inventory.ini file that does not have the default name, such as if you have created separate files for multiple deployments, the command looks like this:

```
ansible-playbook -i inventory-file-name site.yml -e "@vars-file-name" [ option ]
```

Add an option based on the password requirements for the user ID that performs the command, using Table 4.2 on page 40. To specify if you want to perform only an installation or configuration, see "Install Only" on page 41.

In addition, SAS recommends adding a -vvv option to enable verbose logging. This option will assist SAS Technical Support in diagnosing any issues you might need to contact them about.

Install Only

To install, but not configure the software, use the same command that is described in “Deployment Command” on page 40, but replace `site.yml` with `install-only.yml`. Here is an example:

```
ansible-playbook install-only.yml --ask-pass --ask-become-pass -vvv
```

To configure software that has been installed only, use the full command that is described in “Deployment Command” on page 40.

Run from a Directory Other Than the Default

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

Successful Playbook Execution

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

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

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

If the deployment is successful, the software is deployed to the **`/opt/sas`** directory.

Retry a Failed Deployment

If your deployment fails, and you are able to respond to the error message and can recover from the error, you must restart the deployment using the appropriate deployment commands described in “Assessment Test” on page 40 and any appropriate options.

Failures can occur if there are port conflicts.

Deployment Logs

Logs for Ansible deployments are stored in **`sas_viya_playbook/deployment.log`**. If you used the recommended location for uncompressing your playbook, the file is located at **`/sas/install/sas_viya_playbook/deployment.log`**.

To view the logs from the yum installation commands that are used in your deployment, run the following commands:

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

5

Post-installation

SAS Data Preparation Deployment Tasks: Configure Communication **43**
Synchronize Certificates between the Cloud and On-Premises Deployments: SAS Data Preparation Deployment 43
Restart the Data Agent Services 44
Register Remote SAS Data Agent Servers and OAuth Client 45

SAS Data Agent Deployment Tasks: Configure Communication **47**
Synchronize Certificates between the Cloud and On-Premises Deployments: SAS Data Agent Machine 47
Add Groups and Add Users to Groups 48
Configure On-Premises Connection to the SAS Viya Cloud Deployment 50

SAS Data Agent Deployment Tasks: Configure Data Access **52**
Configure SAS/ACCESS to Amazon Redshift 52
Configure SAS/ACCESS Interface to DB2 53
Configure SAS/ACCESS Interface to Hadoop 53
Configure SAS/ACCESS Interface to Microsoft SQL Server 54
Configure SAS/ACCESS Interface to ODBC 55
Configure SAS/ACCESS Interface to Oracle 56
Configure SAS/ACCESS Interface to PostgreSQL 56
Configure SAS/ACCESS Interface to SAP HANA 57
Configure SAS/ACCESS Interface to Teradata 58

Configure a Proxy Server to Communicate with the SAS Data Preparation Deployment **58**

SAS Data Preparation Deployment Tasks: Configure Communication

Synchronize Certificates between the Cloud and On-Premises Deployments: SAS Data Preparation Deployment

Note: If you need a reminder of the topology of your deployment, including the relationship between SAS Data Preparation and SAS Data Agent, see "Deployment Examples and Guidance" on page 3.

1 Get the certificate in either of the following ways:

- On the machine that is assigned to the [httpproxy] host group in the inventory.ini file for the SAS Data Agent deployment, locate the HTTP certificate file `/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/httpproxy-deployTarget-ca.crt`. For more

information about assigning machines to host groups, see "Assign the Target Machines to Host Groups" on page 35.

Note: If you set up a reverse proxy as a best practice, then you should get the file from the endpoint. The value for the certificate name, *deployTarget*, might be different in your deployment, depending on the host alias that is assigned to the [httpproxy] host group in the inventory file.

Copy the certificate to the machine that is assigned to the [CommandLine] host group in the SAS Data Preparation deployment where the sas user ID has Read and Write access. For example, copy the file to the **/tmp** directory. The location of the certificate will be used in step 2 as the value for *path-to-cert*.

- On a machine that is assigned to the [CommandLine] host group in the SAS Data Preparation deployment, run the following command:

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

  ```
  openssl s_client -connect root-endpoint-of-sas-data-agent:443 | sed -ne
  '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/data_agent.crt
  ```

  This command captures the certificate from the SAS Data Preparation deployment and places it in a file named **/tmp/data_agent.crt**. The file name can be customized as needed in the command. The file name will also be used in step 2 as the value for *path-to-cert*.

2 As the sas user, add the certificate to Consul on the machine that is assigned to the [CoreServices] host group in the SAS Data Preparation deployment:

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

  ```
  source /opt/sas/viya/config/consul.conf;

  export CONSUL_HTTP_SSL=true;

  /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
  /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
  kv write --key "cacerts/op1" --value "$(cat path-to-cert)"
  ```

  Note: The *op1* value is a unique name that you assign to the http certificate. The *path-to-cert* value is the full path and file name of the certificate from the SAS Data Agent deployment.

  Here is an example:

  ```
  source /opt/sas/viya/config/consul.conf;
  export CONSUL_HTTP_SSL=true;
  /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
  /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
  kv write --key "cacerts/data_agent.crt" --value "$(cat /tmp/data_agent.crt)"
  ```

3 From the Ansible controller machine where you ran the SAS Data Preparation deployment, and while using the installation account, change directories to the location where the playbook is located:

  ```
  cd sas-viya-playbook-directory
  ```

4 On the Ansible controller in the SAS Data Preparation deployment, rebuild the truststore:

  ```
  ansible-playbook utility/rebuild-trust-stores.yml
  ```

Restart the Data Agent Services

On each machine that is assigned to the [DataServices] host group in the SAS Data Preparation deployment in your cloud environment, restart the SAS Data Agent REST microservices by running the following commands:

- For Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-dagentcont-default restart
sudo service sas-viya-dagentmgmt-default restart
```

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentcont-default
sudo systemctl restart sas-viya-dagentmgmt-default
```

Register Remote SAS Data Agent Servers and OAuth Client

Change the Directory for the SAS User

On the CAS controller machine, which is assigned to the [sas_casserver_primary] host group in the SAS Data Preparation deployment, change to a directory where the sas user has Write permission. Run the following command:

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

Register the SAS Data Agent Server to a Single Tenant SAS Data Preparation Environment

Before you register the SAS Data Agent server, you must have already installed the single-tenant SAS Data Preparation environment.

On the CAS controller machine in the SAS Data Preparation deployment, as the sas user, run the da_reg_server.sh script to register the Data Agent OAuth client and to apply rules and register the remote Data Agent server.

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

```
/opt/sas/viya/home/bin/da_reg_server.sh --customerid shared
--remotehost on-premises-Data-Agent-or-http-proxy-server
--remoteport on-premises-Data-Agent-or-http-proxy-port
--sasadministratoruser sasboot-or-a-user-in-the-SAS-Administrator-group
--sasadministratorpassword password --secret secret-Vault-string
--sas-endpoint https://SAS-Data-Preparation-cloud-hostname:443
```

Note: You can ignore the following message:

```
/opt/sas/viya/home/bin/da_reg_server.sh: line 124: 0: Permission denied
```

Register the SAS Data Agent Server to a Multi-Tenant SAS Data Preparation Environment

Before you register the SAS Data Agent server to a tenant, you must have already onboarded the tenant in the SAS Data Preparation environment.

On the CAS controller machine in the SAS Data Preparation deployment, as the sas user, run the da_reg_server.sh script to register each Data Agent OAuth client and to apply rules and register the remote SAS Data Agent server. There are two methods for registering to a tenant:

- To register to the provider tenant, use --tenantid provider, and use --provider-endpoint with --tenant-endpoint:

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

```
/opt/sas/viya/home/bin/da_reg_server.sh --tenantid provider
--remotehost on-premises-Data-Agent-or-http-proxy-server
--remoteport on-premises-Data-Agent-or-http-proxy-port
--sasadministratoruser sasboot-or-a-user-in-the-SAS-Administrator-group
```

```
--sasadministratorpassword password --secret secret-Vault-string
--provider-endpoint https://SAS-Data-Preparation-cloud-hostname:443
--tenant-endpoint https://SAS-Data-Preparation-cloud-hostname:443
```

- To register to a tenant, use --tenantID *tenantID*, and use --provider-endpoint with --tenant-endpoint:

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

```
/opt/sas/viya/home/bin/da_reg_server.sh --tenantid tenantID
--remotehost on-premises-Data-Agent-or-http-proxy-server
--remoteport on-premises-Data-Agent-or-http-proxy-port
--sasadministratoruser sasprovider-or-a-user-in-the-tenant-SAS-Administrator-group
--sasadministratorpassword password --secret secret-Vault-string
--provider-endpoint https://SAS-Data-Preparation-cloud-hostname:443
--tenant-endpoint https://SAS-Data-Preparation-cloud-hostname:443
```

Note: You can ignore the following message:

```
/opt/sas/viya/home/bin/da_reg_server.sh: line 124: 0: Permission denied
```

Argument Details

Note: The script detects any required options that are missing and displays all required options along with their values. A blank value indicates that a value needs to be specified.

Note: If you access help using the `da_reg_server.sh --help` command, be aware that the `--adminuser` argument is no longer valid and is not required.

Required arguments:

--customerid
The customer ID of an on-premises SAS Data Agent deployment. This variable is used only in single-tenant deployments. The default value is shared and must be unique for each data agent that is registered.

--provider-endpoint
The full host name and the port of SAS Data Preparation in your cloud environment. This variable is used only in multi-tenant deployments.

--remotehost
The fully qualified host name or IP of an on-premises SAS Data Agent server.

--remoteport
The port of a SAS Data Agent server. There is no default value.

--sasadministratoruser
The user who is your SAS Administrator in your cloud environment.

--sasadministratorpassword
The password for the user who is your SAS Administrator in your cloud environment.

--sas-endpoint
The full host name and the port of SAS Data Preparation in your cloud environment. This variable is used only in single-tenant deployments. There is no default value.

--secret
Any string that follows standard password guidelines. The string is used by OAuth client registration to authenticate users from the on-premises SAS Data Agent.

--tenantid
The tenant ID of the on-premises SAS Data Agent deployment. This variable is used only for multi-tenant deployments. The default value is shared, and it must be a unique value for each tenant.

--tenant-endpoint
The full host name and the port of the SAS Data Agent, including the tenant subdomain. This variable is used only for multi-tenant deployments.

Optional arguments:

--conopts
A string that can be added to or that can overwrite the default conopts string that is stored in Consul in the key value (KV) store. **application/data-agent/dagentsrv-*customer-instance*/conopts**

--proxydomain
The outbound proxy domain.

--proxyhost
The outbound proxy host name.

--proxyport
The outbound proxy port.

--registeredservername
Any valid string that can be used to register a second SAS Data Agent service. Use this option to register a second SAS Data Agent server or to register a SAS Data Agent server without using the default naming convention.

--regoverwrite
A value that is used to overwrite an existing SAS Data Agent service registration of the same name. A value of Y overwrites the registration, and a value of N does not overwrite the registration.

--sitename
The name that is used to describe an on-premises Data Agent server. The default name is private.

SAS Data Agent Deployment Tasks: Configure Communication

Synchronize Certificates between the Cloud and On-Premises Deployments: SAS Data Agent Machine

1 Get the certificate in either of the following ways:

- On the machine that is assigned to the [CommandLine] host group in the SAS Data Preparation deployment, locate the HTTP certificate file **/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/httpproxy-*deployTarget*-ca.crt**.

  Note: The value for the certificate name, *deployTarget*, might be different in your environment, depending on the host alias that is assigned to the [httpproxy] host group in the inventory file.

  Copy the certificate to a machine assigned to the [httpproxy] host group in the SAS Data Agent deployment where the sas user ID has Read and Write access. For example, copy the file to the **/tmp** directory.

- On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, run the following command:

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

```
openssl s_client -connect ${root-endpoint-of-sas-viya-with-data-prep}:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/dataprep.crt
```

2 As the sas user, add the certificate to Consul on the machine that is assigned to the [CommandLine] host group in the SAS Data Agent deployment:

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

```
source /opt/sas/viya/config/consul.conf;

export CONSUL_HTTP_SSL=true;

/opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --key "cacerts/cloud1" --value "$(cat path-to-cert)"
```

Note: The *cloud1* value is a unique name that you assign to the http certificate. The *path-to-cert* value is the full path and file name of the certificate that you copied from your SAS Data Preparation machine.

Here is an example:

```
source /opt/sas/viya/config/consul.conf;
export CONSUL_HTTP_SSL=true;
/opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv write --key "cacerts/httpproxy-deployTarget-ca.crt" --value
"$(cat /tmp/httpproxy-deployTarget-ca.crt)"
```

3 From the Ansible controller machine where you ran the SAS Data Agent deployment, and while using the installation account, change directories to the location where the playbook is located:

```
cd sas_viya_playbook-directory
```

4 On the Ansible controller in the SAS Data Agent deployment, rebuild the truststore:

```
ansible-playbook utility/rebuild-trust-stores.yml
```

Add Groups and Add Users to Groups

Strategies for Adding Groups and Users

SAS Data Agent requires two sets of user capabilities that are defined by two custom groups: Data Agent Administrators and Data Agent Power Users.

Here are three strategies for creating the Data Agent Administrators and Data Agent Power Users groups:

- Create and use the default Data Agent Administrators group and the Data Agent Power Users group.

  For details about the Data Agent Administrators and Data Agent Power Users groups, refer to Add Users to Groups in the *Cloud Data Exchange for SAS Viya: Administrator's Guide*.

- Use existing custom groups or use existing LDAP groups.

  For both of these cases, edit the **/etc/sysconfig/sas/sas-viya-dagentsrv-default** file and specify the non-default names.

- Create new groups that are not the default group names.

  You must edit the **/etc/sysconfig/sas/sas-viya-dagentsrv-default** file and specify the non-default names.

To set up groups and users:

1 Decide which type of groups to use.

2 For non-default group names, edit the sysconfig file as explained in "Configure Non-Default Names for Groups" on page 49.

3 To create new groups, verify that the groups do not already exist, and then create the groups as explained in "Add Default Groups or New Groups" on page 49.

4 To add users to groups, see "Add Users to the New Group or Existing Groups" on page 50.

How to Create Groups and Users

You can use the command line or SAS Environment Manager to set up groups and users.

To use SAS Environment Manager, follow these steps but use the user interface information in Identity Management How To: SAS Environment Manager in *SAS Viya Administration Guide: Identity Management*.

To use the command line:

On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, as the sas user, perform these steps:

1 To set up the certification file:

```
export SSL_CERT_FILE=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem
```

2 To set the endpoint:

```
/opt/sas/viya/home/bin/sas-admin profile set-endpoint https://cloud-machine:443
```

3 To get the authentication token:

```
/opt/sas/viya/home/bin/sas-admin auth login --user sas-admin-or-tenant-admin
--password sas-admin-or-tenant-admin-password
```

Configure Non-Default Names for Groups

If you are using non-default names for groups, edit the **/etc/sysconfig/sas/sas-viya-dagentsrv-default** file and add the following lines:

```
DA_ADMIN_GRP="your-Data-Agent-Administrators-group"
DA_POWER_GRP="your-Data-Agent-Power-Users-group"
```

Add Default Groups or New Groups

You can add users to groups using the command line or SAS Environment Manager. Verify whether the groups already exist. If the groups do not exist, create the groups.

To create groups, on the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, as the sas user, perform the following steps:

Note: Names must be the exact group names (not IDs).

1 To add the default Data Agent Administrators group or a new group, run the following command:

```
/opt/sas/viya/home/bin/sas-admin identities create-group --id "your-Data-Agent-Administrators-group-ID"
--name "your-Data-Agent-Administrators-group-name"
--description "your-Data-Agent-Administrators-group-description"
```

Here is an example of how to add the default Data Agent Administrators group:

```
/opt/sas/viya/home/bin/sas-admin identities create-group --id "DataAgentAdministrators"
--name "Data Agent Administrators"
--description "Data Agent Administrators group"
```

2 To add the default Data Agent Power Users group or a new group, run the following command:

```
/opt/sas/viya/home/bin/sas-admin identities create-group --id "your-Data-Agent-Power-Users-group-ID"
--name "your-Data-Agent-Power-Users-group-name"
--description "your-Data-Agent-Power-Users-group-description"
```

Here is an example of how to add the default Data Agent Power Users group:

```
/opt/sas/viya/home/bin/sas-admin identities create-group --id "DataAgentPowerUsers"
--name "Data Agent Power Users" --description "Data Agent Power Users group"
```

Add Users to the New Group or Existing Groups

You can add users to groups using the command line or SAS Environment Manager.

On the machine that is assigned to the [CommandLine] host group in the SAS Data Agent deployment, run the following commands to add users to groups:

Note: Ensure that the user IDs are the same as the user name, but without spaces.

1 As the sas user, add an LDAP user that has access to SAS Data Preparation in the cloud deployment. Add the LDAP user to the Data Agent Administrators group, to your new group, or to an existing group by running the following command:

```
/opt/sas/viya/home/bin/sas-admin identities add-member --group-id "your-Data-Agent-Administrators-group-ID"
--user-member-id "data-agent-user-ID"
```

Here is an example:

```
/opt/sas/viya/home/bin/sas-admin identities add-member --group-id "DataAgentAdministrators"
--user-member-id "dataagentuserID"
```

2 To add a user to the Data Agent Power Users group, to your new group, or to an existing group, run the following command:

```
/opt/sas/viya/home/bin/sas-admin identities add-member --group-id "your-Data-Agent-Power-Users-group-ID"
--user-member-id "power-user-ID"
```

Here is an example:

```
/opt/sas/viya/home/bin/sas-admin identities add-member --group-id "DataAgentPowerUsers"
--user-member-id "DA-power-user"
```

Configure On-Premises Connection to the SAS Viya Cloud Deployment

Configure the SAS Data Agent Machine to Access the SAS Data Preparation Machine

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, as the sas user, edit the following file:

```
/etc/sysconfig/sas/sas-viya-dagentsrv-default
```

2 Uncomment the option for your DA_SERVICES_HOST and set it to the machine name that is assigned to the [CoreServices] host group in the SAS Data Preparation deployment:

```
DA_SERVICES_HOST=SAS-Data-Preparation-host-name
```

Note: You were given a URL for access to your cloud resources. Here is an example:

```
https://root-endpoint-of-SAS-viya-with-data-prep:443
```

You would then specify `mycompany.cloudprovider.providercompany.com` for the DA_SERVICES_HOST value.

3 If your system uses a forward proxy on the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, specify the proxy information. This machine is used to communicate with the SAS Data Preparation deployment.

```
DA_SERVICES_PROXY_SERVER=proxy-server-FQDN
DA_SERVICES_PROXY_PORT=proxy-host-port-number
```

If your proxy requires credentials:

```
DA_SERVICES_PROXY_USER=proxy-username
DA_SERVICES_PROXY_PASS=proxy-paassword
```

4 Save the file.

Add OAuth Secret to Vault

On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, as the sas user, add the OAuth client secret to Vault by running the following command:

- For single-tenant deployments:

```
/opt/sas/viya/home/bin/da_init_tenant.sh --customerid shared --secret secret
```

- For multi-tenant deployments:

```
/opt/sas/viya/home/bin/da_init_tenant.sh --tenantid tenantID --secret secret
```

Note: The secret phrase can be customized but must match the string that you configured in "Register Remote SAS Data Agent Servers and OAuth Client" on page 45. The value for `--customerid` must be the same as the string that you configured in "Register Remote SAS Data Agent Servers and OAuth Client" on page 45.

Configure the DA_TENANT_ID Variable

When you ran da_reg_server.sh and da_init_tenant.sh, if you specified a value other than shared for either --customerid or --tenantid :

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, edit the **`/etc/sysconfig/sas/sas-viya-dagentsrv-default`** file.

2 Add the following line:

```
export DA_TENANT_ID=customerid-or-tenantid-value
```

Note: When setting an environment variable for the customerid, use the DA_TENANT_ID variable.

3 Save the sas-viya-dagentsrv-default file.

Restart the SAS Data Agent Server

On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul and ensures that the Apache HTTP server is aware of the SAS Data Agent server.

Check the Log File

1 As the sas user on the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, go to the **/opt/sas/viya/config/var/log/dagentsrv/default** directory.

2 Locate and open the da_*short-host-name*_**year_month_day_PID**.log file. An example file name is da_ddtjkss03_2018-07-26_24096.log.

3 At the end of the log file, look for the following messages that indicate success:

```
2018-05-22T10:53:17,386 INFO  [00000012] App :sas - Reserved IPv6 port 26301 for server listen
(connection 1).
2018-05-22T10:53:17,386 INFO  [00000012] App :sas - Activated listen on IPv6 port 26301
(connection 1).
2018-05-22T10:53:17,386 INFO  [00000004] App.Program :sas - SAS Data Agent is running under
the user identity sas.
2018-05-22T10:53:17,386 INFO  [00000004] App.Program :sas - SAS Data Agent has completed
initialization.
2018-05-22T10:53:17,386 INFO  [00000004] Admin.Operations :sas - SAH061999I Server SAS Data
Agent, State, running
```

In addition, ensure that there are no ERROR messages in the log. If you have errors, see "Troubleshooting" on page 95.

SAS Data Agent Deployment Tasks: Configure Data Access

Configure SAS/ACCESS to Amazon Redshift

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

1 To reference a Data Source Name (DSN) in your connection, add the DSN to the odbc.ini file.

   a On the SAS client node, edit the **/opt/sas/spre/home/lib64/accessclients/odbc.ini** file and add your DSN definition

   b On the CAS node, edit the **/opt/sas/viya/home/lib64/accessclients/odbc.ini** and add your DSN definition.

   For an example DSN definition, see the [Amazon RedShift Wire Protocol] template in the odbc.ini file.

2 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

3 Add the following lines:

```
export ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
export ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
export ODBCHOME=/opt/sas/viya/home/lib64/accessclients
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

4 Save and close the sas-viya-dagentsrv-default file.

5 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to DB2

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

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

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 sas-viya-dagentsrv-default file.

4 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to Hadoop

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

To manually configure the variables:

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

2 Add the following lines:

```
export JAVA_HOME=location-of-your-Java-8-JRE
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

If you installed your own version of Java, insert its location in the JAVA_HOME field. If you are using the JRE that is installed with your SAS software, its default location is /usr/lib/jvm/jre-1.8.0. The default should be

used unless you edit the vars.yml file in the playbook to specify a different location for the installation of the JRE.

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

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

4 Save and close the sas-viya-dagentsrv-default file.

5 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to Microsoft SQL Server

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

1 To reference a Data Source Name (DSN) in your connection, add the DSN to the odbc.ini file.

a On the SAS client node, edit the **/opt/sas/spre/home/lib64/accessclients/odbc.ini** file and add your DSN definition

b On the CAS node, edit the **/opt/sas/viya/home/lib64/accessclients/odbc.ini** and add your DSN definition.

For an example DSN definition, see the [SQL Server Wire Protocol] template in the odbc.ini file.

Note: If your deployment will have encryption enabled, perform the following steps on the client machine and the CAS nodes:

- Add the SSLLibName connection option to the DSN and set it to the absolute path for the OpenSSL SSL library file.
- Add the CryptoLibName connection option to the DSN and set it to the absolute path for the OpenSSL cryptographic library file.

2 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

3 Add the following lines:

```
export ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
export ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
export ODBCHOME=/opt/sas/viya/home/lib64/accessclients
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

4 Save and close the sas-viya-dagentsrv-default f file.

5 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent environment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to ODBC

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

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

2 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 ODBCINI=location-of-your-odbcinst.ini-file-including-file-name
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

For iODBC:

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

For unixODBC:

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

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

3 Save and close the sas-viya-dagentsrv-default file.

4 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to Oracle

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

To manually configure the variables:

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

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 sas-viya-dagentsrv-default file.

4 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

  ```
  sudo systemctl restart sas-viya-dagentsrv-default
  ```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to PostgreSQL

Note: This information is applicable only if you ordered SAS /ACCESS to PostsgreSQL (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.

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

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=UTF-8
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/sas/spre/home/lib64
```

3 Save and close the sas-viya-dagentsrv-default file.

4 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure SAS/ACCESS Interface to SAP HANA

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

1 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployemnt, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

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 LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client
```

3 Save and close the sas-viya-dagentsrv-default file.

4 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

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).

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 machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, use a text editor to open the sas-viya-dagentsrv-default file.

```
sudo vi /etc/sysconfig/sas/sas-viya-dagentsrv-default
```

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-directory/msg/%N:$NLSPATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:Teradata-TTU-installation-path-including-lib64-directory:
$LD_LIBRARY_PATH
export THREADONOFF=1
```

An example of the TTU Default LD_LIBRARY_PATH is

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

5 Save and close the sas-viya-dagentsrv-default f file.

6 On the machine that is assigned to the [DataAgent] host group in the SAS Data Agent deployment, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul. The script also verifies that the Apache HTTP server is aware of the SAS Data Agent server that was just registered.

Configure a Proxy Server to Communicate with the SAS Data Preparation Deployment

Configuration steps are required if the Consul UI for the SAS Data Preparation deployment communicates with SAS Data Agent through a proxy server. You can configure the proxy server to enable communication using either of two methods. Choose the method that is appropriate for your security policies and performance requirements.

- The proxy server can be configured to operate in pass-through mode. This method provides the best performance and does not require additional changes for security certificates. Refer to the documentation for your proxy server for information about how to configure pass-through mode.
- The truststores for each segment of the connection between the Consul UI and SAS Data Agent should be updated with information about the segments with which it communicates. The SAS Data Agent truststore should be updated with certificate information about the proxy server. The truststore that is used by Consul UI should be updated with certificate information about the proxy server. The proxy server truststore should be updated with certificate information about both the Consul UI and SAS Data Agent.

6

Validating the Deployment

Validate Round-Trip Communication between SAS Data Preparation and SAS Data Agent 61
Perform Installation Qualification on RPM Packages 62
Verify PostgreSQL 64
Verify the SAS/ACCESS Interface to Your Databases 64

Validate Round-Trip Communication between SAS Data Preparation and SAS Data Agent

When you run `sas-admin dagentsrv services list` from the SAS Data Agent, the following validations occur:

- SAS Data Preparation can handle authentication and communication.
- SAS Data Agent microservices are running.
- SAS Data Agent is running and SAS Data Preparation can communicate to it.
- At a minimum, if the listing shows —_SERVER_ and BASE, the SAS Data Agent was correctly initialized and is awaiting further instructions.
- The Data Agent itself is running, and SAS Data Preparation can communicate to it.

These validations ensure that a complete round-trip is successful.

On the SAS Data Agent machine, as the sas user, perform these tasks:

1 Run the following commands and enter the specified responses:

```
export
SSL_CERT_FILE=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem

/opt/sas/viya/home/bin/sas-admin profile init
Enter configuration options:

Service Endpoint>
https://data-preparation-in-the-cloud.sas.com:443

Output type (text|json|fulljson)> text

Enable ANSI colored output (y/n)?> y
Saved 'Default' profile to /home/sas/.sas/config.json.
```

If you receive the following error message, ensure that you exported the SSL_CERT_FILE correctly.

```
level=debug msg="An error occurred when making request." err="Get
```

```
https://dataagent-test-tue-cloud.dmmdev.sashq-d.openstack.sas.com/dataAgentConte
nt/servers: x509: certificate signed by unknown authority"
Error reading list of available servers.
```

2 Specify the user ID that was added as a member of the Data Agent Administrator's group in "Add Groups and Add Users to Groups" on page 48.

```
/opt/sas/viya/home/bin/sas-admin auth login --user data-agent-admin-ID
--password password
```

If you receive an error message like the following message, ensure that the *data-agent-admin-ID* that you specify for the `--user` option is a member of either the Data Agent Administrators group or the Data Agent Power users group.

```
Error reading list of services. API[DriverConnect]
TKTSRETURN[TKTS_ERROR(-2130708478)] SQLSTATE:[42000]
MESSAGE:[DSN "ADMIN" not found or user not authorized.,
VENDOR_CODE:[-2130708184].
Connection does not exist
```

3 To validate communication, run the following command:

Note: If your deployment has multiple SAS Data Agent servers, specify which Data Agent server to use to validate communication. To obtain the list of servers, specify `server list`.

`--data-agent` *SAS-Data-Agent-server-name*

```
/opt/sas/viya/home/bin/sas-admin dagentsrv services list
```

Here is typical output:

```
Name            Type     Domain   Version   Options
__SERVER__      server   ---      2.3       PURGE_CACHE=30;  CACHE=(NAME=AS;TIMEOUT=300)
BASE            base     ---      2.3       ---
```

Perform Installation Qualification on RPM Packages

Some of your SAS software is collected in RPM (Red Hat Package Manager) packages.

1 To qualify the installation of your RPM packages, run the basic RPM commands:

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

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

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

2 To verify SAS Data Agent deployment, to obtain a list of the relevant RPM packages that are deployed on your system:

```
sudo rpm -Vv sas-dagentsrv
```

Here is the output:

```
.........   /opt/sas/viya/home/SASFoundation
.........   /opt/sas/viya/home/SASFoundation/sasexe
.........   /opt/sas/viya/home/SASFoundation/sasexe/fsnetiom.so
.........   /opt/sas/viya/home/SASFoundation/utilities
.........   /opt/sas/viya/home/SASFoundation/utilities/bin
.........   /opt/sas/viya/home/SASFoundation/utilities/bin/tktsql
```

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

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

Here is a successful verification that shows the list of files that make up the RPM but with no error indicators:

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

Here is an unsuccessful verification that provides error indicators next to the file name:

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

The error indicators are shown in the following format:

```
SM5DLUGT c
```

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

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

The meaning of each error indicator is described as follows:

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

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

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. Under Linux, the special files for SCSI disk drives should have a major number of 8, and the major number for an IDE disk drive's special file should be 3. Any change to a file's major number could produce disastrous effects. RPM tracks such changes. A file's minor number is similar to the major number, but conveys different information to the device driver. For disk drives, this information can consist of a unit identifier.

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

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

G File group. Similar to file ownership, a group specification is attached to each file. Primarily used for determining access rights, a file's group specification can also become a user's group ID if that user

executes the file's contents. Therefore, any changes in a file's group specification are important and should be monitored.

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

c Configuration file. This is useful for quickly identifying configuration files, since they are likely to change and therefore are unlikely to verify successfully. You could also get a d in this slot, indicating that the file is for documentation, which is also likely to change often.

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

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

- **`/opt/sas/viya/config/etc/evmcltsvcs/sas-ops-agent-update.sh`**
- **`/opt/sas/viya/config/etc/evmsvrops/sas-ops-agentsrv-update.sh`**

Verify PostgreSQL

1 On the machine that you assigned to the [pgpoolc] host group, to check status:

- On Red Hat Enterprise Linux 6.*x* and Linux 7.*x*:

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

- For SUSE Linux:

```
sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status
```

2 If PostgreSQL is running appropriately, you should receive a response like this:

```
PGPool is running with PID=11445
 Checking Postgresql nodes status...
 node_id | hostname | port | status | lb_weight |  role   | select_cnt | load_balance_node | replication_delay
 ---------+----------+------+--------+-----------+---------+------------+-------------------+-------------------
  0       | machine1 | 5452 | up     | 0.250000  | primary | 1          | true              | 0
  1       | machine2 | 5452 | up     | 0.250000  | standby | 0          | false             | 0
  2       | machine3 | 5452 | up     | 0.250000  | standby | 0          | false             | 0
  3       | machine4 | 5452 | up     | 0.250000  | standby | 0          | false             | 0
 (4 rows)
```

A status of `up` for a node indicates that the node is running.

Verify the SAS/ACCESS Interface to Your Databases

To validate the SAS/ACCESS interface to your databases, refer to the Getting Started in the *Cloud Data Exchange for SAS Viya : Administrator's Guide*.

Note: On the **Data Sources tab**, be sure to select a **Source type** of Cloud Data Exchange (CDE).

7

Completing the Deployment

Save Snapshot Directory Content 65
Share Important Deployment Information with the Administrators 65
Refer to Additional Documentation 65

Save Snapshot Directory Content

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

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

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

Share Important Deployment Information with the Administrators

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

- The location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 25.
- The URL to access the software.

Refer to Additional Documentation

After you validate the deployment, you can perform initial administrative tasks. For more information, refer to Cloud Data Exchange for SAS Viya: Administrator’s Guide.

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

Also, the appendix in this guide provides information to help you set up High Availability PostgreSQL.

8

Managing Your Software

Overview 67
What Is an Update? 67
What Is an Add-On Product? 67
What Is an Upgrade? 68
Updating Your SAS Viya Software 68
Overview 68
User Requirements for Performing the Update 69
Synchronize the Mirror Repository 69
(Optional) List the Packages That Are Available for Update 70
Update with Yum 70
Update with Zypper 71
Update with Ansible 72
Generate a New Ansible Playbook 73
Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4 75
Overview 75
Add SAS Viya Software 75
User-Related Tasks 78

Overview

What Is an Update?

An update provides modifications for features that are not working as intended or adds minor software enhancements and compatibility. Software updates are released to address security issues when they occur, to address minor bugs discovered in the software, and to improve the operation of hardware or peripherals. These incremental updates improve the operation of your software and are small enough that they do not require a new order. Updated software is intended to be compatible with existing configuration, content, and data. To perform an update, you will run the same tools that were run during the initial deployment. You might determine that your software needs updating or you might be notified by SAS that updates are available.

To update your SAS Viya software, see “Updating Your SAS Viya Software” on page 68.

What Is an Add-On Product?

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

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

For information about adding on a product, see “Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4” on page 75.

What Is an Upgrade?

There are two types of upgrade associated with SAS Viya 3.4 software. You can upgrade products in SAS Viya 3.4, such as SAS Econometrics 8.3 to SAS Econometrics 8.4. You can also upgrade to SAS Viya 3.4 from earlier versions of SAS Viya, such as SAS Viya 3.2 or SAS Viya 3.3.

Upgrading Products in SAS Viya 3.4

An upgrade to SAS Viya products adds significant feature changes or improvements to those products. The changes are important enough to require a new order. To upgrade products, you will perform the same steps that are used for adding on products. You will need a new order, and you must get an updated version of the Orchestration CLI to create a new playbook. Upgrading products will also update your currently deployed software.

If you are not sure if the product that you are upgrading from and the product that you are upgrading to are both in SAS Viya 3.4, go to the Product Compatibility Matrix to determine the SAS Viya version that your products are running on. If the product that you upgrading from is from a version of SAS Viya earlier than 3.4, you will actually be upgrading SAS Viya and should use that process.

For information about upgrading a product in SAS Viya 3.4, see “Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4” on page 75.

Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya

An upgrade for SAS Viya adds significant feature changes or improvements to SAS Viya. To perform an upgrade, you will run the same tools that were run during the initial deployment. You will need a new order to upgrade your deployed software, and you must get an updated version of the Orchestration CLI to create a new playbook. Add-on products present in the order will be installed as part of the upgrade process. An upgrade might require changes to the deployed software’s configuration.

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

For information about upgrading to SAS Viya 3.4 from other versions of SAS Viya, see Upgrading to SAS Viya 3.4 from Earlier Versions of SAS Viya in *SAS Viya for Linux: Deployment Guide*.

Note: If you have upgraded SAS Data Preparation in SAS Viya 3.4, then you should upgrade SAS Data Agent in SAS Viya 3.4.

Updating Your SAS Viya Software

Overview

An update replaces some or all of your deployed software with the latest versions of that software. You perform the update with the same command that was used to install SAS Viya, and use the same software order and the same playbook. If you have deployed SAS Data Agent multiple times, you should perform these steps for each deployment.

- To see what updates are available for your deployed software, go to the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

- Use the same tool (Ansible, yum, or Zypper) to update that you used to install. For example, if you used an Ansible playbook for your initial installation, update with Ansible.
- SAS might update components of the Ansible playbook that is used to deploy your SAS Viya software. You will need to download the current version of the SAS Orchestration CLI to generate a new Ansible playbook for your deployment, and then run the new Ansible playbook.

Here are other considerations when preparing for an update:

- The update process preserves any user-modified configuration values in the vars.yml file, but changes made to other files in the deployment might be lost. Therefore, SAS recommends that you make changes to vars.yml when possible in order to avoid any loss of customizations that you made to other files.
- You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.
- If you are using a PDF version of this guide, go to the Deployment Guides web page at https://support.sas.com/en/documentation/install-center/sas-viya/deployment-guides.html and verify that you have the latest version of the deployment documentation before you start the update process. The release date of each document is located in the bottom right corner of the front page.
- Updating SAS Viya software requires an outage period because some SAS Viya services are stopped and restarted automatically during the update process.

User Requirements for Performing the Update

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

Synchronize the Mirror Repository

If you are using a mirror repository:

1. (Optional) To list the packages that are available for the update process, run the following command on the machine where the mirror repository is located:

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

   ```
   mirrormgr mirror diff --deployment-data path-to-deployment-zip-file-from-SOE
   --path path-to-mirror-destination --latest
   ```

   Note: The `mirrormgr mirror diff` command returns a list of the available files for all supported platforms of the products in the deployment. To filter out unwanted content from the output of the command, use a pipe operator and the grep command. For example, add the following at the end of the preceding command to filter out all file names containing the text `suse`:

   ```
   | grep -v "suse"
   ```

2. Synchronize the deployment's mirror repository with the SAS mirror repository. Use the same options to update the mirror repository that you used to create the mirror repository. For more information, see "SAS Mirror Manager and the Mirror Repository" on page 22.

   To synchronize, run the following command on the machine where the connected mirror repository is located:

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

   ```
   mirrormgr mirror --deployment-data path-to-deployment-zip-file-from-SOE
   --path path-to-mirror-destination --latest
   ```

3. If you are in a deployment without internet access, move the files from the machine where the connected mirror repository is located to the machine where the unconnected mirror repository is located.

(Optional) List the Packages That Are Available for Update

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

on Red Hat Enterprise Linux:

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

on SUSE Linux:

```
sudo zypper list-updates | grep "sas-"
```

Update with Yum

Important: You can use yum to update your software only if your deployment is on Red Hat Enterprise Linux or an equivalent distribution.

To update a SAS Viya deployment using yum, repeat these steps for each machine in the deployment:

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

   a On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

      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
      ```

   b On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

      For example, you can run the following command to create a text file that lists all the SAS host groups:

      For Red Hat Enterprise Linux:

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

      Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

2 Stop all the SAS services on the machine:

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

3 On each machine in your deployment, to show the available updates for all SAS Viya software on the machine:

   ```
   sas_groups=$(LANG=en_US sudo yum grouplist -v "sas-*" | awk -F'[()]' \
   '/^Installed Groups:/{found=1;next}/^[[:alnum:]]/{found=0}found \
   {print"@"$(NF-1)}')

   sudo yum update $sas_groups $(rpm -qg "SAS")
   ```

   You must run these commands to update any external software applications on which the SAS yum groups depend.

4 At the prompt `Is this ok`, review the available updates and then enter `y`.

5 Restart the services that are installed on the machine, including the CAS controller, a SAS object spawner, and SAS Studio. You must also start SAS/CONNECT if it is included in your deployment.

To restart all the SAS services on the machine:

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

6 (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 use the following command to create a text file that lists all the SAS RPM packages:

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

On each machine in your deployment, create a file that lists the SAS yum 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 yum groups:

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

You can see the differences between the previous and current deployments by comparing the lists of installed software before the update (Step 1 on page 70) and after the update.

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

Update with Zypper

Important: You can use zypper to update your software only if your deployment is on SUSE Linux or an equivalent distribution.

To update a SAS Viya deployment using zypper, repeat these steps for each machine in the deployment:

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

a On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 25.

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
```

b On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see “Store the Playbook” on page 25.

For example, you can run the following command to create a text file that lists all the SAS host groups:

For SUSE Linux, to list the SAS packages:

```
sudo rpm -qa | grep "sas-"
```

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

2 Stop all the SAS services on the machine:

```
sudo /etc/init.d/sas-viya-all-services stop
```

3 To update all SAS Viya software on the machine:

```
sudo zypper update "sas-*"
```

4 At the prompt `Continue? [y/n]`, review the available updates and then enter `y`.

5 Restart the services that are installed on the machine, including the CAS controller, a SAS object spawner, and SAS Studio. You must also start SAS/CONNECT if it is included in your deployment.

To restart all the SAS services on the machine:

```
sudo /etc/init.d/sas-viya-all-services start
```

6 (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 use the following command to create a text file that lists all the SAS RPM packages:

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

You can see the differences between the previous and current deployments by comparing the lists of installed software before the update (Step 1 on page 71) and after the update.

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

Update with Ansible

To update a SAS Viya deployment using Ansible:

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

a On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

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
```

b On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

For example, you can run the following command to create a text file that lists all the SAS host groups:

For Red Hat Enterprise Linux:

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

For SUSE Linux, to list the SAS packages:

```
sudo rpm -qa | grep "sas-"
```

Note: If you receive a message such as the following, it can be ignored:

```
Repository repository-name is listed more than once in the configuration
```

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

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

3 When you start the update, use the same command and options that you used when you performed the initial deployment. For more information, see "Deploy the Software" on page 40.

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. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25. For example, you can use the following command to create a text file that lists all the SAS RPM packages:

```
sudo rpm -qg 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. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25. For example, you can run the following command to create a text file that lists all the SAS host groups:

on Red Hat Enterprise Linux:

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

To list the SAS packages on SUSE Linux, run the following command:

```
sudo rpm -qa | grep "sas-"
```

Note: If you receive a message such as the following, it can be ignored:

```
Repository repository-name is listed more than once in the configuration
```

You can see the differences between the previous and current deployments by comparing the lists of installed software before the update (Step 1 on page 72) and after the update.

Generate a New Ansible Playbook

You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

If updates are needed in the Ansible playbook, to generate and apply a new Ansible playbook for your deployment:

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

a On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

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
```

b On each machine in your deployment, create a file that lists the SAS host groups that are installed on a machine. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

For example, you can run the following command to create a text file that lists all the SAS host groups:

For Red Hat Enterprise Linux:

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

For SUSE Linux, to list the SAS packages:

```
sudo rpm -qa | grep "sas-" > > /sas/install/new_viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

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 "Create a Playbook" on page 24.

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/update/** 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/update/
```

5 Merge the vars.yml file and the inventory file from the previous deployment into the new playbook. If the previous inventory file contains any spaces that are used to indent machine names, do not include the extra spaces.

a Compare the two 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/update/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory-file /sas/update/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 you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.

6 To apply the new Ansible playbook, change to the directory where the new playbook is located:

```
cd /sas/update/
```

Run the following command:

```
ansible-playbook site.yml
```

7 (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. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25. 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/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. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25. For example, you can run the following command to create a text file that lists all the SAS host groups:

on Red Hat Enterprise Linux:

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

To list the SAS packages on SUSE Linux, run the following command:

```
sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt
```

You can see the differences between the previous and current deployments by comparing the lists of installed software before performing this task (Step 1 on page 73) and after.

Note: If you receive a message such as the following, it can be ignored:

```
Repository repository-name is listed more than once in the configuration
```

Adding SAS Viya Software to a Deployment and Upgrading Products in SAS Viya 3.4

Overview

The procedure to add SAS Viya software to an existing deployment and to upgrade a product in SAS Viya 3.4 is the same. Adding new software to your deployment or upgrading a product will also update your currently deployed software.

Adding SAS Viya software to an existing deployment or upgrading SAS Viya products requires an outage period. During the process, all SAS Viya services must be stopped and then restarted.

This chapter includes all the steps that are required for the adding SAS Viya software regardless of the version of the source environment and the software installed.

You will need the location of the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

Note: If you have upgraded SAS Data Preparation in SAS Viya 3.4, then you should upgrade SAS Data Agent in SAS Viya 3.4.

Add SAS Viya Software

To add SAS Viya software and update a SAS Viya deployment:

1. Before you begin, you should review the "Introduction" on page 1, "System Requirements" on page 7, and "Pre-installation Tasks" on page 21 chapters of this guide.
2. (Optional) Record the existing list of installed software before you begin.
   a. On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

      Run the following command to create a text file that lists all the RPM packages:

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

b On each machine in your deployment, create a file that lists the yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

- Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:

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

- Run the following command to create a text file that lists the RPM packages on SUSE Linux:

```
sudo rpm -qa | grep "sas-" > /sas/install/viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

3 If your deployment used a mirror repository, you must download the current version of SAS Mirror Manager. For more information, see "SAS Mirror Manager and the Mirror Repository" on page 22.

4 If you are adding software and your software is mirrored, you must update the mirror using the ZIP file that is attached to your new Software Order Email (SOE) before you perform these steps. For more information, see "Synchronize the Mirror Repository" on page 69. If you are upgrading existing software to a new version, you might want to create a new mirror so that you can delete the old files after the upgrade. For more information, see "SAS Mirror Manager and the Mirror Repository" on page 22.

5 Download the latest SAS Orchestration CLI. For more information, see "Create a Playbook" on page 24.

6 Using the SAS Orchestration CLI that you downloaded, create a new playbook. For more information, see "Create a Playbook" on page 24.

You must extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to **/sas/install/**, you might extract the new playbook to **/sas/addon/** or **/sas/upgrade/** instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:

- To preserve the original vars.yml file and the inventory file.
- To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following

```
tar xf SAS_Viya_playbook.tgz -C /sas/new-playbook-directory-name/
```

7 Merge the vars.yml file and the inventory file from the previous deployment into the new playbook. If the previous inventory file contains any spaces that are used to indent machine names, do not include the extra spaces.

a Compare the two vars.yml files, and compare the two inventory files. Check for additions or changes in the newer set of files. Be sure to evaluate the comments to determine whether the requirements for host groups changed between releases of the software.

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

```
diff /sas/install/sas_viya_playbook/vars.yml
/sas/new-playbook-directory-name/sas_viya_playbook/vars.yml
```

```
diff /sas/install/sas_viya_playbook/inventory.ini
/sas/new-playbook-directory-name/sas_viya_playbook/inventory.ini
```

Note: The [consul], [httpproxy], and [operations] host groups must be present in ***/sas/new-playbook-directory-name/sas_viya_playbook/inventory.ini*** and must contain the host entries from the original deployment.

Note: The [consul] and [httpproxy] host groups must be present in each ***/sas/new-playbook-directory-name/sas_viya_playbook/CAS-server-inventory-file-name*** and must contain the host entries from the original deployment.

b If the new files contain new content, then merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.

Note: All host groups that are present in the inventory file from the previous deployment must remain on the same machines in the inventory file for the new deployment. New host groups that were not in the previous deployment should be assigned to machines in the current deployment. Review the comments that precede each host group before assigning host groups to machines. For more information, see "Assign the Target Machines to Host Groups" on page 35.

c If you have questions about whether to add a key/value pair from an older file to the new file, contact SAS Technical Support.

8 Complete the tasks in "File System and Storage Requirements" on page 9, as appropriate.

9 To verify the health of the SAS Infrastructure Data Server before running the playbook, perform the task in Verify the Health of the SAS Infrastructure Data Server in *SAS Viya for Linux: Deployment Guide*.

10 It is recommended to add a 404 redirect in your deployment's web server to prevent users from accessing the deployment before the process is completed. See your web server documentation.

Note: As administrators, use a secondary URL to complete the deployment steps.

11 Install your SAS Viya software beginning with "Modify the vars.yml File" on page 36.

12 After you install the software, you must complete the post-installation tasks that are appropriate for your deployment.

a If you added any SAS/ACCESS software to your deployment, configure data access on page 52 as appropriate.

b For any added products that require configuration, see "Post-installation" on page 43.

c To stop and then start `sas-viya-dagentsrv`, perform one of the following actions, as appropriate:

- On Red Hat Enterprise Linux 6.*x* or an equivalent distribution:

```
sudo service sas-viya-dagentsrv-default stop
sudo service sas-viya-dagentsrv-default start
```

- On Red Hat Enterprise Linux 7.*x* or an equivalent distribution:

```
sudo systemctl stop sas-viya-dagentsrv-default
sudo systemctl start sas-viya-dagentsrv-default
```

For more information, see Start and Stop Servers and Services in *General Servers and Services: SAS Viya Administration*.

d Validate the deployment on page 61.

e Complete the deployment on page 65.

13 (Optional) Record the new list of installed software.

a On each machine in your deployment, create a file that lists the names and versions of the RPM packages of the SAS Viya software that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

Run the following command to create a text file that lists the RPM packages:

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

b On each machine in your deployment, create a file that lists the SAS yum groups or packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For more information about this directory, see "Store the Playbook" on page 25.

- Run the following command to create a text file that lists the yum groups on Red Hat Enterprise Linux:

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

- Run the following command to create a text file that lists the RPM packages on SUSE Linux:

```
sudo rpm -qa | grep "sas-" > /sas/install/new_viya_packages.txt
```

Note: If you receive a message such as the following, it can be ignored.

```
Repository repository-name is listed more than once in the configuration
```

You can see the differences between the previous and current deployments by comparing the lists of installed software that precedes the update (Step 2 on page 75) and that follows the update.

c To verify that a specific update was applied, compare the contents of the text file created in Step 13a on page 77 to the packages that are listed for the specific update. The package list for a specific update is available in the Manifest View for the update on the SAS Viya Hot Fix Availability web page at http://ftp.sas.com/techsup/download/hotfix/HF2/Viya_home.html.

User-Related Tasks

Perform the following steps:

1 If you created a 404 redirect in your deployment's web server in Step 10 on page 77, remove the 404 redirect. See your web server documentation.

2 Inform users that they must perform the following actions:

- Clear web browser caches before using the upgraded deployment.
- Change any entries in web browser bookmarks from `SASHome` to `SASDrive`.

9

Uninstalling SAS Viya

What deploy-cleanup Does 79
Use deploy-cleanup 79
Uninstall a Mirror Repository 81

What deploy-cleanup Does

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

1 Stop all SAS services.

2 Remove all SAS RPMs.

3 Delete any remaining SAS .pid files.

4 Delete the entitlement_certificate.pem and SAS_CA_Certificate.pem files.

The deploy-cleanup command renames the **/opt/sas/viya** directory to **/opt/sas/viya_*epoch*.** .

The uninstallation does not remove the customized script that you received with your SOE, and it does not remove any users that have been set up.

Use deploy-cleanup

Using deploy-cleanup.yml removes the directory structure created by the deployment. Before removing the structure, you may want to first remove the SAS Data Agent secret stored in SAS Secrets Manager. To do so, follow the instructions in Remove Data Agent Secret in *Cloud Data Exchange for SAS Viya: Administrator's Guide*.

Use the following steps to run deploy-cleanup.yml and remove any processes that might remain.

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

2 Here is the basic syntax for the command to run the playbook and uninstall the software:

Note: The command should be run as a root or sudoer user.

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

Add an option based on the password requirements for the user ID that performs the command, using Table 9.1 on page 80.

Table 9.1 *Command Options Based on Password Requirements*

Password Requirements	Option
Does not require passwords	use the command as written
Requires a sudo password only	`--ask-become-pass`
Requires an SSH password only	`--ask-pass`
Requires both a sudo and an SSH password	`--ask-pass --ask-become-pass`

Here is an example of the deploy command that requires both sudo and SSH passwords:

```
ansible-playbook deploy-cleanup.yml --ask-pass --ask-become-pass
```

3 If you have multiple deployments of SAS Data Agent, you will need to run a similar command for each:

```
ansible-playbook -i additional-inventory-file-name deploy-cleanup.yml -e "@additional-vars-file-name"
```

Add an option based on the password requirements for the user ID that performs the command, using Table 9.1 on page 80.

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 perform the following steps on every machine in your deployment:

   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. If “sas-configuration-cli” is not listed, then you can move on to the next machine.

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

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

Uninstall a Mirror Repository

If your deployment includes a mirror repository and you want to remove it as well, you can run a basic Linux command to do so. Because all the files of the mirror repository are contained in a single directory, use the following command to remove the mirror repository:

```
sudo rm -rf path-to-mirror-repository
```

If you did not change the default location of the SAS Mirror Manager log when you deployed your software, you should also remove the log from **`/.local/share/mirrormgr`** in the home directory of the install user.

Appendix 1

Creating High Availability PostgreSQL Clusters

Overview . . . 83
HA PostgreSQL Topologies . . . 83
Set Up a Horizontal Cluster . . . 85
Edit the inventory.ini File . . . 85
Edit the vars.yml File . . . 86
Set Up a Vertical Cluster . . . 87
Edit the inventory.ini File . . . 87
Edit the vars.yml File . . . 87
Set Up a Hybrid Cluster . . . 88
Edit the inventory.ini File . . . 88
Edit the vars.yml File . . . 88
Set Up Multiple Clusters . . . 89
Modify inventory.ini and vars.yml Files . . . 89
Configure Services to the Clusters . . . 92
Deployment Logs . . . 92
Verify the Deployment . . . 93

Overview

By default, when you use the instructions in “Installation” on page 33, 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 multinode deployments, PGPool node can be co-located with data nodes or deployed on its own machine. Note that co-locating nodes on a machine provides increased read throughput but also increases the risk of node loss should that machine become unavailable.

The following table demonstrates how nodes can be distributed in the multinode topologies.

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.

When you revise the vars.yml file for your cluster, the following variables under INVOCATION_VARIABLES should be modified:

pgpoolc

- PCP_PORT: the PCP port for the PGPool instance
- PGPOOL_PORT: the PGPool port. This is the primary port that all database connections will go to.
- SANMOUNT: the location where the data files will be placed
- SERVICE_NAME: the unique name that you assign to your cluster. This value must match the SERVICE_NAME of all sasdatasvrc nodes that will attach to this pgpoolc in the cluster.

sasdatasvrc

- NODE_NUMBER: the sequential node identifier starting at 0
- NODE_TYPE: P for primary or S for standby. There can be only one primary per cluster.>
- PG_PORT: The PostgreSQL database port. PGPool talks to the database on this port. Clients use the PGPOOL_PORT.
- SANMOUNT: the location where the data files will be placed
- SERVICE_NAME: the unique name that you assign to your cluster. This value must match the SERVICE_NAME of the pgpoolc node that this data node will attach to in the cluster.

Set Up a Horizontal Cluster

Edit the inventory.ini File

Modify the inventory.ini file as described in order to describe the topology that you are using. First, define all the machines in your deployment as described at “Specify the Machines in the Deployment” on page 34. 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 horizontal cluster described in the table above, with PGPool being on the same machine as the first HA PostgreSQL node. (The example shows only the entries related to HA PostgreSQL):

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget4 ansible_host=host4.example.comx ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
'''
[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
```

```
...
```

Edit the vars.yml File

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

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

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

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 40 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 34. 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]
deloyTarget1
'''
[sasdatasvrc]
deployTarget1
...
```

Edit the vars.yml File

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

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

```
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
     SERVICE_NAME: postgres
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5535'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres
```

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

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 40 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 34. 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 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 hybrid cluster example from the table above, this section would describe a two machines, with a subsection for pgpoolc on the same machine as two of the sasdatasvrc nodes. This is what that section would look like when filled out for our example:

```
# Multiple invocation definitions
INVOCATION_VARIABLES:
  deployTarget1:
   pgpoolc:
   - PCP_PORT: '5631'
```

```
      PGPOOL_PORT: '5630'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5632'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5633'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
  deployTarget2:
    sasdatasvrc:
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5632'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    - NODE_NUMBER: '3'
      NODE_TYPE: S
      PG_PORT: '5633'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
```

Note that the first machine has a single pgpoolc entry and two sasdatasvrc entries. The nodes are numbered from 0 to 3, and node 0 is the primary node. The PORT entries for either machine show a different port in order to avoid any conflict. The entry for SANMOUNT: will read the deployment and use the location of the SAS _CONFIG_ROOT directory and append the directory name.

After you save the vars.yml file and you complete the other deployment steps, use the commands described at “Deploy the Software” on page 40 to deploy your SAS Viya software, including HA PostgreSQL.

Set Up Multiple Clusters

Modify inventory.ini and vars.yml Files

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

- a single-node cluster with pgpoolc and sasdataservc on a machine named deployTarget1
- a horizontal cluster with pgpoolc on deployTarget1 and a sasdatasrvc node on each machine
- a vertical cluster with pgpoolc on deployTarget3 and all the sasdatasrvc nodes on deployTarget4
- a hybrid cluster with pgpoolc on deployTarget1, two sasdatasrvc nodes on deployTarget2, and two more sasdatasrvc nodes on deploytarget3

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

```
deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
```

```
deploytarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
deploytarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
...
[pgpoolc]
deployTarget1
deployTarget3
deployTarget4
...
[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
...
```

This is how the INVOCATION_VARIABLES section of the vars.yml file would be filled out:

```
# Multiple invocation definitions
 INVOCATION_VARIABLES:
  deployTarget1:
    pgpoolc:
    - PCP_PORT: '5431'
      PGPOOL_PORT: '5430'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_hybrid
    - PCP_PORT: '5461'
      PGPOOL_PORT: '5460'
      SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
    sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5452'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_horizontal
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5462'
      SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres
  deployTarget2:
    sasdatasvrc:
    - NODE_NUMBER: '0'
      NODE_TYPE: P
      PG_PORT: '5432'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_hybrid
    - NODE_NUMBER: '2'
      NODE_TYPE: S
      PG_PORT: '5433'
      SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
      SERVICE_NAME: postgres_hybrid
    - NODE_NUMBER: '1'
      NODE_TYPE: S
      PG_PORT: '5452'
```

```
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal
deployTarget3:
  pgpoolc:
  - PCP_PORT: '5441'
    PGPOOL_PORT: '5440'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  sasdatasvrc:
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5432'
    SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_hybrid
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5433'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_hybrid
  - NODE_NUMBER: '2'
    NODE_TYPE: S
    PG_PORT: '5452'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal
deployTarget4:
  pgpoolc:
  - PCP_PORT: '5451'
    PGPOOL_PORT: '5450'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal
  sasdatasvrc:
  - NODE_NUMBER: '0'
    NODE_TYPE: P
    PG_PORT: '5442'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5443'
    SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '2'
    NODE_TYPE: S
    PG_PORT: '5444'
    SANMOUNT:'{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5445'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5452'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
    SERVICE_NAME: postgres_horizontal
```

Note: If you are deploying multiple clusters, one of the PGPools must be named postgres, and each PGPool name must be unique across clusters. In addition, each cluster must contain one sasdatasvrc node with a NODE_TYPE of P.

Configure Services to the Clusters

By default, all microservices connect to the HA Postgres cluster that is named postgres. You can configure individual services to use additional HA Postgres clusters (if they exist) by adding service-specific sections to the sitedefault.yml file.

1. 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 indention level as `application`, add the following content:

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

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

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

```
config:
    application:
...
    authorization:
        sas:
            database:
                databaseServerName: postgres-horizontal
          spring.datasource.password: ${sas.database.postgres-horizontal.password}
```

4. Save and close the sitedefault.yml file.

Deployment Logs

Each PGPool node and HA PostgreSQL data node has its own set of directories for logging. The logs for PGPool are located at

```
/opt/sas/viya/config/var/log/sasdatasvrc/postgres/pgpool0/
```

The log for the HA PostgreSQL nodes is located at

```
/opt/sas/viya/config/var/log/sasdatasvrc/postgres/node0/
```

Verify the Deployment

The deployment performs a verification of the HA PostgreSQL cluster before it completes. This verification first confirms that connections can be made to PGPool and to all data nodes, and then runs queries on all of the nodes. The verification also performs write and delete operations to ensure that values that are written to or removed from the primary data node are replicated to all of the standby nodes in a multinode deployment.

The verification log is called sds_status_check_*date-timestamp*.log. It can be found in the pgpool log folder of each cluster. The fastest way to determine whether your HA PostgreSQL deployment was successful is to read the verification log.

Appendix 2

Troubleshooting

SAS Viya Services Do Not Start . . . 95
Nothing to Do Dialog . . . 96
PCA and KCLUS Procedures Were Not Found . . . 96
Timeout Dialog . . . 96
From Any Browser: Connection Is Not Private . . . 97
From Google Chrome: Connection Is Not Private . . . 97
Unable to Read a Key . . . 98
Zypper Run Command Failed: Library Is Locked . . . 98
Zypper Run Command Failed: Shared Vault Value Has Not Been Set . . . 99
Connection Reset by Peer Network Problem . . . 99
Internet Connectivity Problems . . . 100
Invalid Host Name in the sitedefault.yml File . . . 101
SAS Data Agent Log Contains Multiple Errors . . . 102
SAS Data Agent Log: File Was Not Found . . . 103
Problems When Using the SAS Data Agent CLI . . . 103
SSL Connection Problems When the Data Agent Tenant Initialization Script is Run . . . 103

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 in *SAS Viya Administration: General Servers and Services* .

2 Delete the `/opt/sas/viya/config/data/consul/checks/` directory.

3 Restart all services.

Nothing to Do Dialog

Error

After removing the software and attempting to re-install the software, this message is displayed:

```
Error: Nothing to do
```

Explanation

The directories that contain the software were deleted. However, the yum remove command was never run. In **`/var/log/yum.log`**, the last entry for the rpm message is `Installed`.

Resolution

Clean up the yum repository by running the following command.

```
yum remove packagename
```

You can then re-install the software.

PCA and KCLUS Procedures Were Not Found

Error

```
ERROR: Procedure PCA not found
```

or

```
ERROR: Procedure KCLUS not found
```

Explanation

The installation was attempted on a system that was not completely cleaned up from a previous installation.

Resolution

Uninstall SAS/CONNECT by running the following command:

```
yum groups mark remove "SAS/CONNECT"
```

Re-install SAS/CONNECT by running the following command:

```
sudo yum groupinstall "SAS/CONNECT"
```

Timeout Dialog

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 timeout value for Ansible as follows:

```
timeout=number-of-seconds
```

Deploy your software by running the Ansible playbook again.

From Any Browser: Connection Is Not Private

Explanation

The default self-signed certificates are not in the operating system truststore by default. The Apache Web Server is configured to use a certificate that is signed by this Certificate Authority (CA). When you open any SAS URL and navigate to the web server from a machine that does not have this CA in the truststore, you will receive the message `Your connection is not private`. The message does not indicate that there is any problem with the SAS deployment.

Resolution

SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see the Security section of the System Requirements chapter.

From Google Chrome: Connection Is Not Private

Error

When attempting to access SAS Viya software from Google Chrome, the following message is displayed:

```
Your connection is not private.
```

Explanation

If you have previously accessed a website using https, when you access the website again, Google Chrome automatically redirects to https.

Resolution

To reset Google Chrome so that it does not redirect to https:

1. In the Chrome address bar, enter this command:

```
chrome://machine-name//#hsts
```

2. Under **Query domain**, in the **Domain** box, enter the name of the machine that was used in the URL that you were attempting to access.
3. Click **Query** to determine whether the machine is known to the browser.
4. If the machine is known to the browser, under **Delete domain**, enter that machine name in the **Domain** box. Click **Delete**.

The corrected URL should now work with the HTTP protocol.

Unable to Read a Key

Error

When running the deployment, the following message is displayed:

```
fatal: [deployTarget2]: FAILED! =>{"changed": false, "failed": true, "msg":
"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.

Zypper Run Command Failed: Library Is Locked

Error

```
Zypper run command failed with return code 7
```

Explanation

The Zypp system management library is locked because PackageKit is running.

Resolution

Ensure that the PID in the error message has released the lock, and then run the playbook again.

Zypper Run Command Failed: Shared Vault Value Has Not Been Set

Error

The deployment receives error messages such as `ERROR 11789` and `Zypper run command failed with return code 7.`

Explanation

The shared vault was not set before the deployment process was completed. The deployment did not fail and should succeed if there are no other errors.

Resolution

You can set the shared vault value after the deployment is complete. For details, see Backup and Restore: Initial Tasks in *SAS Viya 3.4 Administration: Backup and Restore*.

Connection Reset by Peer Network Problem

Error

Deployments on Red Hat Enterprise Linux might receive a "Connection reset by peer" message when installing or applying updates for SAS Viya. This is usually indicative of networking issues.

Resolution

Deployments on Red Hat Enterprise Linux might receive a "Connection reset by peer" message during deployment or when applying updates for SAS Viya. This is usually indicative of networking issues.

To change the retries and timeout values for yum:

1. Open the **`/etc/yum.conf`** file as root or with sudo on the affected machine. Here is an example of a typical **`/etc/yum.conf`** file:

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800
```

2. If the `retries` and `timeout` variables are present, ensure that they are set to 20 and 120, respectively. If those variables are not present in the file, add them.

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
```

```
metadata_expire=1800
retries=20
timeout=120
...
```

3 Save and close the **/etc/yum.conf** file.

4 Repeat these steps for every affected machine.

If you continue to get the "Connection reset by peer" message, reopen the **/etc/yum.conf** file and revise these values upward.

Internet Connectivity Problems

Performing the pre-installation and installation tasks requires connection to the internet and, specifically, SAS repositories online. If you receive errors about connecting to the internet, you should work with your system administrator to correct them. The following steps are provided as guidelines for general areas of connectivity.

1 If you are deploying on Red Hat or an equivalent distribution, ensure that your firewall is open in order to allow access to the IP address of the content delivery servers that provide updates from Red Hat or Oracle. The IP addresses for content delivery services vary by region. For more information about the list of IP addresses, see one of the following websites:

- Public CIDR Lists for Red Hat
- https://linux.oracle.com/

  This website provides instructions for registering with the Oracle ULN.

2 Ensure that the firewall allows access to the SAS repositories.

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

```
tar xf SAS_Viya_playbook.tgz
```

Be sure to leave a compressed copy of the .tgz file in the same location.

b Run the following command from the playbook subdirectory (**/sas/install/sas_viya_playbook** if you used the recommended location for uncompressing your playbook).

```
curl -OLv --cert ./entitlement_certificate.pem --cacert ./SAS_CA_Certificate.pem
https://ses.sas.download/ses/repos/meta-repo/bigfile.bin
```

If the firewall is set up correctly, the command successfully transfers the bigfile.bin file. If a connection fails, add any failing server to your firewall proxy whitelist and try the command again. Repeat this step until you successfully transfer the bigfile.bin file.

3 The firewall service should not be running while you deploy your software.

a Create a list of the services that are running by performing the appropriate command from the list below.

- For Red Hat Enterprise Linux 6.*x*:

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

- For Red Hat Enterprise Linux 7.*x* and SUSE Linux:

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

b Use the following table to identify the name of the service that you should look for in the output from the command.

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

Table A2.1 *Firewall Services by Linux Distribution and Version*

Linux Version	Service
Red Hat Enterprise Linux earlier than 7.1	iptables
Red Hat Enterprise Linux 7.1 or later	firewalld
Oracle Linux earlier than 7.1	iptables
Oracle Linux 7.1 or later	firewalld
CentOS Linux earlier than 7.1	iptables
CentOS Linux 7.1 or later	firewalld
SUSE Linux	SuSEfirewall2.service

If the firewall service from the table is listed in the output of the command, then the firewall is running and you should continue to the next step. Otherwise, you do not need to take any further actions.

c To stop iptables, run the following commands:

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

To stop firewalld, run the following commands:

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

To stop SuSEfirewall2.service, run the following commands:

```
sudo systemctl stop SuSEfirewall2.service
sudo systemctl disable SuSEfirewall2.service
sudo systemctl stop SuSEfirewall2_init.service
sudo systemctl disable SuSEfirewall2_init.service
```

Invalid Host Name in the sitedefault.yml File

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
```

SAS Data Agent Log Contains Multiple Errors

Issue

The SAS Data Agent log file contains errors like these:

```
2018-08-09T11:35:58,811 ERROR [00000009] App.DFASCL.Provider.MID :sas - Unable
to load extension: (tkmttok)
2018-08-09T11:35:58,815 ERROR [00000009] App.SQLServices.SQLProcessor :sas -
SQLProcessor.ProcExecute/ExecSQL function failed (rc=-2130708478).
2018-08-09T11:35:58,815 ERROR [00000009] App.SQLServices.SQLProcessor :sas - SQL
node "System Configuration" execution failed (phase=Start.Epilog).
2018-08-09T11:35:58,831 ERROR [00000009] App.Server :sas - 0; ; ; : Unexpected
failure encountered: Function failed.
2018-08-09T11:35:58,831 ERROR [00000009] App.Program :sas - 0; ; ; V_CTOR,
CREATE SERVICE CONTEXT: Unexpected failure encountered: Function failed.
2018-08-09T11:35:58,831 ERROR [00000009] App.Server :sas - 0; ; ; V_CTOR:
Start-up error - Creation of server instance failed
```

Resolution

1. Evaluate the **`/etc/sysconfig/sas/sas-viya-dagentsrv-default`** file. Ensure that the DA_SERVICES_HOST variable has the correct value. See “Configure the SAS Data Agent Machine to Access the SAS Data Preparation Machine” on page 50 for more information.

2. Re-run the on-premises da_init_tenant.sh script, ensuring that the value for the tenantid is correct. See “Add OAuth Secret to Vault” on page 51 and “Configure the DA_TENANT_ID Variable” on page 51.

3. Review the command that is used for da_reg_server.sh to verify that the SAS Data Preparation cloud provider registration was correct. See “Register Remote SAS Data Agent Servers and OAuth Client” on page 45.

4. Review the command that is used to add groups using the sas-admin-identities CLI. If the command did not contain all the parameters, use the following command to delete the group:

   ```
   /opt/sas/viya/home/bin/sas-admin identities delete-group --id "group-ID"
   ```

   Then add the group again, using the steps in “Add Default Groups or New Groups” on page 49.

SAS Data Agent Log: File Was Not Found

Issue

The SAS Data Agent log file contains errors like this:

```
2018-08-16T13:22:25,520 ERROR [00000009] App.Program :sas - The file was not
found:
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tls/certs/dagentsrv/def
ault/dagentsrv_encrypted_default.c
```

Resolution

Ensure that the requiretty setting has been disabled in the sudoers file. See “Disabling the requiretty Setting” on page 11.

Problems When Using the SAS Data Agent CLI

Issue

When you use the SAS Data Agent CLI, the following error is displayed:

```
Error reading information for service BASE.
IOException[javax.net.ssl.SSLHandshakeException:
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target]
Host[ddtmzsuse03.DDT.sashq-r.openstack.sas.com] Port[443].
HTTPS with javax.net.ssl.trustStore =
[/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.j
ks].
```

Resolution

Ensure that the certificates are properly synchronized. See “Synchronize Certificates between the Cloud and On-Premises Deployments: SAS Data Preparation Deployment” on page 43 and “Synchronize Certificates between the Cloud and On-Premises Deployments: SAS Data Agent Machine” on page 47.

SSL Connection Problems When the Data Agent Tenant Initialization Script is Run

Issue

When you run the da_init_tenant.sh script, an error similar to the following is displayed:

```
Mon Aug 13 14:20:33 EDT 2018 - register OAuth client secret for "dagentsrv-bar"
curl: (35) SSL connect error
Mon Aug 13 14:20:33 EDT 2018 - Failed to obtain register OAuth client secret for
dagentsrv-bar
```

Resolution

Ensure that you are using the correct version of curl. See “ Linux Requirements” on page 10.