



SAS[®] Viya[®] 3.3 for Linux: Deployment Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS® Viya® 3.3 for Linux: Deployment Guide*. Cary, NC: SAS Institute Inc.

SAS® Viya® 3.3 for Linux: Deployment Guide

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

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

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

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

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

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

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

September 2018

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

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

3.3-P1:dplyml0phy0lax

Contents

Chapter 1 / Introduction	1
About This Guide	1
How Deployment Works	3
Deployment Examples and Guidance	5
Contact SAS Technical Support	15
Chapter 2 / System Requirements	17
Hardware Requirements	18
Operating System Requirements	23
Server Software Requirements	25
Data Source and Storage Requirements	26
User and Group Requirements	31
Security Requirements	36
Client Requirements	38
Deployment Tools	39
Chapter 3 / Pre-installation Tasks	41
Make Sure That You Have the Required Files	41
Confirm the Identities of the Hosts	42
Enable Required Ports	42
Firewall Considerations	44
Configure SELinux	46
Configure a Proxy Server	46
Enable the Yum Cache	47
Enable a Shared File System	47
Install Ansible	48
(Optional) Enable Key-Based SSH Authentication	49
Set Environment Variables for SAS Event Stream Processing	50
Perform Linux Tuning	50
Chapter 4 / Installation	55
Overview	56
Modify the Initial Deployment	56
Use a Mirror Repository	56
Edit the Inventory File	56
Modify the vars.yml File	62
Configure LDAP Settings for SAS Event Stream Manager	82
SAS Viya and Multi-tenancy	83
Deploy the Software	85
Install with SAS 9.4 Software	86
Deployment Logs	87
Chapter 5 / Post-Installation Tasks	89
Configure Security	89
Configure Machine and Application Settings	94
Configure Data Access	102
Configure Data Quality	113
Chapter 6 / Validating the Deployment	115
Perform Installation Qualification on RPM Packages	115

Access CAS Server Monitor	117
Access SAS Environment Manager	118
Verify SAS Message Broker	118
Verify SAS Infrastructure Data Server	119
Verify SAS Event Stream Manager Status	119
Overview of Data Access Verification	120
Verify SAS/ACCESS Interface to Amazon Redshift	120
Verify SAS/ACCESS Interface to DB2	121
Verify SAS/ACCESS Interface to Greenplum	121
Verify SAS/ACCESS Interface to HAWQ	122
Verify SAS/ACCESS Interface to Impala	123
Verify SAS/ACCESS Interface to Microsoft SQL Server	124
Verify SAS/ACCESS Interface to MySQL	125
Verify SAS/ACCESS Interface to Netezza	125
Verify SAS/ACCESS Interface to ODBC	126
Verify SAS/ACCESS Interface to Oracle	127
Verify SAS/ACCESS Interface to PostgreSQL	127
Verify SAS/ACCESS Interface to SAP R/3	128
Chapter 7 / Completing the Deployment	131
Save Snapshot Directory Content	131
Share Important Deployment Information with the Administrators	131
Next Steps for SAS Event Stream Processing Users	132
Refer to Additional Documentation	132
Chapter 8 / Managing Your Software	133
Overview	133
Apply the CVE-2017-7547 Security Patch	134
Updating Your SAS Viya Software	134
Adding SAS Viya Software	140
Upgrading Your SAS Viya Software	143
Generate a New Ansible Playbook	151
Chapter 9 / Uninstalling SAS Viya	155
Overview	155
What deploy-cleanup Does	155
Create a Backup for SAS Event Stream Processing	156
Uninstall Command	156
Uninstall SAS Embedded Process	157
Uninstall SASHDAT Plug-ins	157
Appendix 1 / Creating High Availability PostgreSQL Clusters	159
Overview	159
HA PostgreSQL Topologies	160
Set Up a Horizontal Cluster	161
Set Up a Vertical Cluster	163
Set Up a Hybrid Cluster	164
Set Up Multiple Clusters	165
Deployment Logs	168
Verify the Deployment	169
Appendix 2 / Deploying with Yum	171
Overview	172
Run the Deployment Script	172
Deploy httpd and MOD_SSL	173
Set Up the CAS Administrator	173

Set Up the CAS Controller to Run as a Service	174
Start the Services	174
Configure SAS/ACCESS Interface to Amazon Redshift	174
Configure SAS/ACCESS Interface to DB2	175
Configure SAS/ACCESS Interface to Greenplum	176
Configure SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop	176
Configure SAS/ACCESS Interface to HAWQ	177
Configure SAS/ACCESS Interface to Impala	178
Configure SAS/ACCESS Interface to Microsoft SQL	179
Configure SAS/ACCESS Interface to MySQL	179
Configure SAS/ACCESS Interface to Netezza	180
Configure SAS/ACCESS Interface to ODBC	180
Configure SAS/ACCESS Interface to Oracle	181
Configure SAS/ACCESS Interface to PostgreSQL	182
Configure SAS/ACCESS Interface to SAP HANA	183
Configure SAS/ACCESS Interface to SAP R/3	183
Configure SAS/ACCESS Interface to Teradata	184
Configure Settings for SAS Event Stream Processing for CAS	185
Install Sample SAS Data Sets	185
Log On to SAS Studio	185
View Deployment Logs	185
Validate the Installation	186
Next Steps	186
Uninstall SAS Viya with Yum	186
Appendix 3 / Creating and Using Mirror Repositories	189
Overview	189
Requirements	190
Use Ansible to Create a Mirror Repository	191
Use Yum to Create Mirror Repositories	194
Creating a Local Copy of Documentation	201
Uninstalling SAS Viya from Mirrored Repositories	202
Appendix 4 / Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop	203
Supported Hadoop Distributions	203
Deployment Tasks for Hive Access	203
Pre-deployment Hadoop Tasks for Hive Access	204
Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop	205
Appendix 5 / Hadoop Deployment: Configuring SAS In-Database Technologies	211
Hadoop Prerequisites	212
Overview of the In-Database Deployment Package for Hadoop	212
SAS Embedded Process for SAS 9.4: Uninstall and Deploy for SAS Viya	212
Deploy the SAS Embedded Process	213
(Optional) Deploy TLS Certificates	218
SASEP-ADMIN.SH Script	219
Verify SAS Data Connect Accelerator for Hadoop	223
Additional Configuration for HCatalog File Formats	224
Add the YARN Application CLASSPATH for MapR	226
Performance Tuning for the SAS Embedded Process	226
Add the SAS Embedded Process to Nodes after the Initial Deployment	229
Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya	229

Appendix 6 / Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS	231
About CAS SASHDAT Access to HDFS	231
Supported Hadoop Distributions	232
Overview of Deployment Tasks for HDFS for Existing Hadoop Clusters	232
Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters	232
Review the Passwordless Secure Shell Requirements	233
Kerberos Requirements	233
Deploying SAS Plug-ins for Hadoop	236
Uninstalling SAS Plug-ins for Hadoop	242
sashdat-install.sh Reference	245
Appendix 7 / Teradata Deployment: Configuring SAS In-Database Technologies	249
Prerequisites	249
Overview of the In-Database Deployment Package for Teradata	250
Connections from SAS 9.4 Clients	250
Teradata Installation and Configuration	250
Installing the SAS In-Database Deployment Package for Teradata	251
(Optional) Deploy TLS Certificates	253
Configuring SAS Data Quality Accelerator for Teradata	254
Appendix 8 / Troubleshooting	261
Troubleshooting SAS Viya	261

Introduction

About This Guide	1
Get the Latest Guide	1
Tips for Getting Started	1
About Deploying SAS Event Stream Processing Products Only	2
SAS Products and Supporting Components	2
Audience	3
How Deployment Works	3
Using Ansible to Deploy SAS Viya	3
Using Yum to Deploy SAS Viya	4
How a SAS Viya Deployment Differs from a SAS 9 Deployment	4
Deployment Examples and Guidance	5
About the Deployment Examples	5
Single Machine Deployment	6
Full Deployment with a Separate, Single-Machine CAS Server	6
Full Deployment with a Distributed CAS Server	7
Cluster for High Availability PostgreSQL	8
Multi-tenant Environment	9
Hadoop Integration: Access Data in Hive	10
Hadoop Integration: CAS SASHDAT Access to HDFS	12
Teradata Integration	15
Contact SAS Technical Support	15

About This Guide

Get the Latest Guide

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

[SAS Viya Deployment Guides](#)

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

Tips for Getting Started

This guide provides concepts, steps, and reference information for deploying and updating SAS Viya software. Here are a few tips to help you get started.

- Are you new to SAS Viya? Go to [“How Deployment Works” on page 3](#) for information about using Ansible, which is the preferred tool for deploying SAS Viya for Linux.
- Do you plan to integrate SAS Viya with Hadoop or Teradata? Do you want to enable SAS Viya to support a multi-tenancy environment? Go to [“Deployment Examples and Guidance” on page 5](#) and review the example topologies and tips for navigating the documentation. Also, this section provides guidance for deploying SAS Cloud Analytic Services (CAS) to one machine or across multiple machines.
- Do you want to update, add software to, or upgrade an existing SAS Viya deployment? Go to [“Managing Your Software” on page 133](#) for more information.

Note: Before you begin the deployment process, see [“System Requirements” on page 17](#) to understand hardware and software requirements, and for guidance about improving performance within environments. Also, review [“Pre-installation Tasks” on page 41](#) to make sure that your environment is ready.

About Deploying SAS Event Stream Processing Products Only

Throughout this document, exceptions to the deployment instructions are indicated when your order contains SAS Event Stream Processing products only. In this document, this group of products is referred to as the “SAS Event Stream Processing product family.” It contains the following:

- SAS Event Stream Processing
- SAS Event Stream Manager
- SAS Event Stream Processing Analytics

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 Cloud Analytic Services (CAS)	SAS/ACCESS Interface to Amazon Redshift (on SAS Viya)
SAS Data Preparation	SAS/ACCESS Interface to DB2 (on SAS Viya)
SAS Data Quality	SAS/ACCESS Interface to Greenplum (on SAS Viya)
SAS Decision Manager	SAS/ACCESS Interface to HAWQ (on SAS Viya)
SAS Econometrics	SAS/ACCESS Interface to Hadoop (on SAS Viya)
SAS Event Stream Processing	SAS/ACCESS Interface to Impala (on SAS Viya)
SAS Event Stream Processing Analytics	SAS/ACCESS Interface to Microsoft SQL Server (on SAS Viya)
SAS Event Stream Processing for CAS	SAS/ACCESS Interface to MySQL (on SAS Viya)
SAS Event Stream Manager	SAS/ACCESS Interface to Netezza (on SAS Viya)
SAS Model Manager	SAS/ACCESS Interface to ODBC (on SAS Viya)
SAS Optimization	SAS/ACCESS Interface to Oracle (on SAS Viya)
SAS Visual Analytics	SAS/ACCESS Interface to PostgreSQL (on SAS Viya)
SAS Visual Data Mining and Machine Learning	SAS/ACCESS Interface to SAP HANA (on SAS Viya)
SAS Visual Forecasting	SAS/ACCESS Interface to Teradata (on SAS Viya)
SAS Visual Statistics	SAS In-Database Technologies for Hadoop (on SAS Viya)
SAS Visual Text Analytics	SAS In-Database Technologies for Teradata (on SAS Viya)

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

Audience

This guide is written for administrators who install and configure software for your company or organization.

- To perform the steps in this guide, you should have a working knowledge of Ansible, which is the preferred tool for deploying and updating SAS Viya. Also, you should have a working knowledge of the Linux operating system and basic commands.
- To configure SAS Viya to analyze data that is stored in a Hadoop cluster or in Teradata, you must work with the Hadoop administrator or the Teradata system administrator, respectively.

How Deployment Works

Using Ansible to Deploy SAS Viya

The Basics

You can use Ansible to deploy SAS Viya to one or multiple machines. Using Ansible gives you the most control over your deployment.

- Ansible is configuration management software that provides a straightforward approach to deploying SAS Viya. To deploy using Ansible, you customize files for your environment, and then you run a command to deploy software according to the values in those files. The set of files, known collectively as “the playbook,” provides the instructions about what software is deployed on which machines. In this guide, “run the playbook” means to deploy or update SAS Viya software.
- Before you can run the playbook, you must create one that is customized for your order. To do that, you use the SAS Orchestration CLI. The Software Order Email (SOE) that SAS sends to your business or organization contains a link to instructions on how to use the SAS Orchestration CLI.
- Each time you run the playbook, Ansible automates a series of yum commands that securely access the latest SAS Viya software to which you are entitled. The software is downloaded from repositories that are maintained by SAS or from a local mirror repository that you create and maintain at your own site.
Note: Yum is a software-package manager for Linux operating systems. SAS Viya is packaged in the RPM Package Manager (RPM) format, which simplifies installation and upgrade tasks.
- To use Ansible, you must install it first. In this guide, the machine on which you install Ansible is called the “Ansible controller.” The Ansible controller must have SSH access to the machines on which you plan to deploy SAS Viya.

Files Used for Deployment

The following files are used to deploy SAS Viya using Ansible. 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	The inventory file that is used to deploy SAS Viya. You edit the inventory.ini file to map machines (or hosts) to the SAS Viya 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. For example, you edit the vars.yml file to configure a data connector, to specify the installation type, to manage passwordless SSH settings, and so on.
sitedefault.yml	The sitedefault.yml file contains variables for more advanced implementations, such as setting up a High Availability (HA) PostgreSQL cluster and enabling SAS Viya to run in a multi-tenancy mode.

Note: SAS provides sample files (templates) in your playbook, which you can re-purpose for your deployment. After you create your playbook, look for these files in the `sas_viya_playbook/sample-inventories` subdirectory.

Installation Types

By default, a deployment using Ansible includes the installation of the full suite of products and user interfaces that you ordered. In the SAS documentation, this type of deployment is referred to as a “full deployment.”

Although SAS recommends a full deployment, you can set the installation type in the vars.yml file to “programming-only.” A programming-only deployment limits support to data scientists and programmers who use SAS Studio, or direct programming interfaces such as Python or REST APIs. Understand that this type of deployment does not include SAS Home, SAS Environment Manager, and the complete suite of services that are included with a full deployment. Also, a programming-only deployment does not support multi-tenancy. Therefore, make sure that you are providing your users with the features that they require.

Note: A programming-only deployment does not support SAS Event Stream Manager.

Using Yum to Deploy SAS Viya

Instead of using Ansible, you can use a script that includes Yum commands, and additional manual steps to deploy the programming-only interface on a single machine.

- Like the programming-only option when using Ansible, deploying with Yum does not include SAS Home, SAS Environment Manager, and the complete suite of services.
- You cannot deploy SAS Event Stream Manager by this process.
- To learn more, see [“Deploying with Yum” on page 171](#).

How a SAS Viya Deployment Differs from a SAS 9 Deployment

Besides the use of Ansible, the SAS Viya deployment differs from a SAS 9 deployment in the following ways:

- The SAS Deployment Wizard and the SAS Deployment Manager that support SAS 9.4 are not used to install and configure SAS Viya.
- Because the RPM-based deployment model works with repositories that are native to your operating system, a SAS Software Depot is not required for your SAS Viya software.

- No SAS deployment tools must be installed on target machines. All deployment actions can be remotely executed from the Ansible controller.

Deployment Examples and Guidance

About the Deployment Examples

The deployment examples provide high-level guidance for deploying SAS Viya software. Refer to the following examples for topologies that best fit your environment and users, and for tips on using the documentation.

When reviewing the examples, consider the following:

- Ansible is used to deploy SAS Viya. A benefit of using Ansible is that you can assign host groups to different machines, which can allow for dedicated servers and clustering.
- The inventory.ini file is shown in the examples. To illustrate collections of the host groups that are provided in the inventory.ini file, the following categories of software are depicted.
 - “Service layer” represents the numerous features for the visual interfaces, advanced analytics, text analytics, data management and mining, and more. Also, the service layer includes the stateful services for SAS Viya, which includes the following:
 - SAS Infrastructure Data Server, which is used to store user content. SAS Infrastructure Data Server is based on PostgreSQL.
 - SAS Message Broker, which is based on Pivotal’s messaging broker, RabbitMQ.
 - SAS Configuration Server, which uses HashiCorp’s Consul as a service configuration registry that serves as a central repository for configuration data, service discovery, and health status.
 - SAS Secret Manager, which is based on HashiCorp Vault. SAS Secret Manager uses Vault to store and generate secrets such as Transport Layer Security (TLS) certificates.
 - An Apache HTTP Server, which serves static HTML content and to proxy client connections.
 - “Programming run time” represents SAS Foundation, SAS Studio, SAS Workspace Server, SAS/CONNECT Server and any SAS/ACCESS engines.
 - “CAS server” provides the run-time environment where data management and analytics take place. The CAS server can be deployed in a distributed fashion across multiple machines, or on a single machine.
 - Distributing the CAS server across multiple machines allows for massively parallel processing (MPP) by users. An advantage to MPP is that, whenever possible, data is loaded into memory in parallel, which can result in faster load times. Also, the distributed CAS server can be configured for fault tolerance, so if a CAS worker fails, then another CAS worker can use a redundant copy of data to complete data analysis.
 - Deploying the CAS server on a single machine allows for symmetric multi-processing (SMP) by users. A single-machine CAS server performs serial loads of data into memory from a supported data source. The in-memory analytic features of a distributed server are available to the single-machine CAS server.
 - Data connectors, which vary according to the order, must be deployed to one or more machines on which CAS is running. For scenarios in which CAS is deployed to multiple machines, data connectors are deployed to the CAS controller and to each CAS worker.
- Before you deploy, refer to the system requirements to understand hardware and software requirements and recommendations.

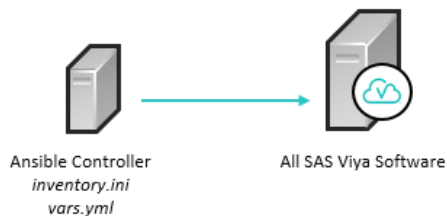
Single Machine Deployment

You can deploy the software, including the CAS server, to a single machine. This type of deployment can be useful for demonstration purposes or for deploying software for a specific group of users. For a single-machine deployment, the CAS server is deployed to support analytics and data-management processing in symmetric multiprocessing (SMP) mode.

- Using Ansible is the recommended approach. If you use Ansible, you can deploy the software to the same machine where Ansible is running, or you can deploy to a different, target machine.
- If you use yum, the programming-only environment is deployed, and SAS Home, SAS Environment Manager, and other services are not deployed.

The following example shows using Ansible to deploy SAS Viya, including the CAS server, on one machine.

Figure 1.1 SAS Viya on One Machine Using Ansible



Note: Because the CAS server is deployed to one machine, the secondary CAS controller and CAS workers are not deployed.

Full Deployment with a Separate, Single-Machine CAS Server

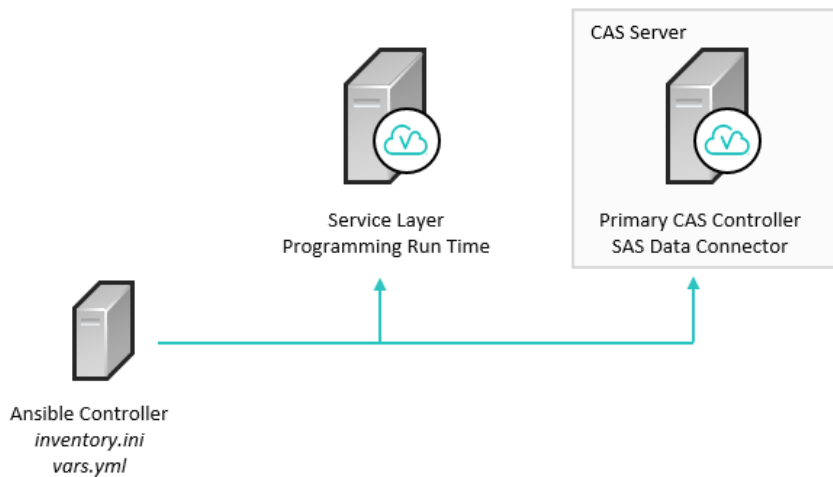
In this section, Ansible is used to perform a full deployment. Also, a single-machine CAS server is deployed to support analytics and data-management processing in symmetric multiprocessing (SMP) mode.

When reviewing the examples, consider the following information:

- A full deployment includes the service layer and the programming run time. To accomplish a full deployment, the `a11` setting is used as the installation type in the `vars.yml` file.
- To deploy the CAS server, the primary CAS controller is deployed to its own machine, and data connectors are shown as configured for use. A best practice is to configure the data connectors for use in the `vars.yml` file before running the playbook. Because a single-machine CAS server is deployed, the secondary CAS controller and the CAS workers are not deployed.

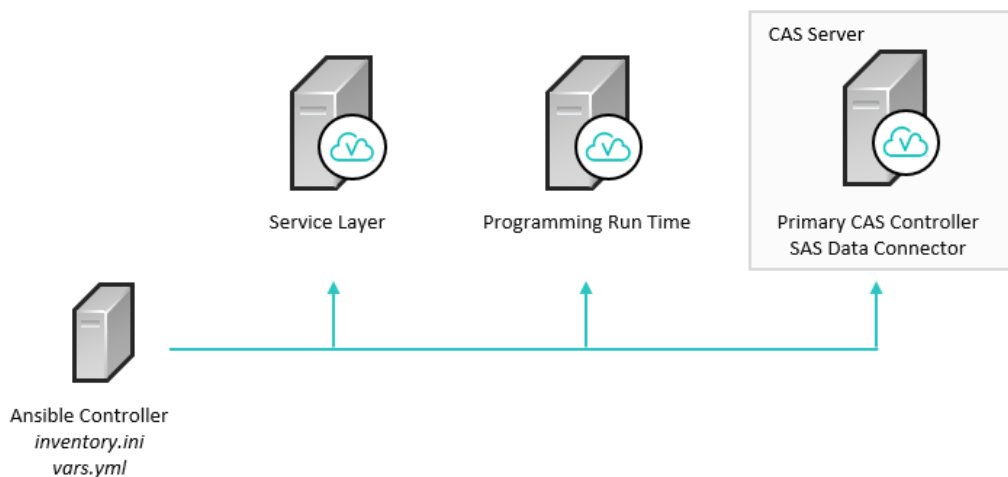
The following example shows a full deployment with single-machine CAS server and a separate machine for the service layer and the programming run time.

Figure 1.2 SAS Viya on Two Machines



The following example shows separate machines for the service layer and the programming run time.

Figure 1.3 SAS Viya on Three Machines



Full Deployment with a Distributed CAS Server

In this section, Ansible is used to deploy the service layer and the programming run time across two machines, and the CAS server is distributed across multiple machines. An advantage to deploying a distributed CAS server is that optimal processing can be achieved through massively parallel processing (MPP) for multiple users.

Consider the following information:

- A full deployment includes the service layer and the programming run time. To accomplish a full deployment, the `a11` setting is used as the installation type in the `vars.yml` file.
- To deploy the CAS server, different host groups that are related to CAS are assigned to machines in the `inventory.ini` file before you run the playbook. The primary CAS controller, the secondary CAS controller, and each CAS worker are all deployed to different machines. Data connectors are shown as configured for use

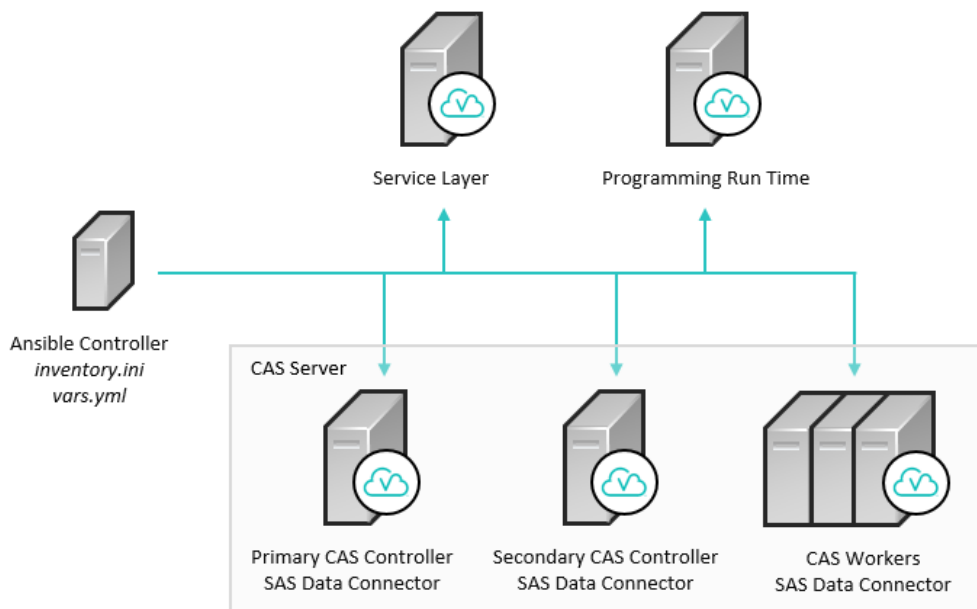
on each machine where a CAS controller or a worker is deployed. A best practice is to configure the data connector for use in the vars.yml file before you run the playbook.

Note: The secondary CAS controller, which acts as a backup controller, cannot be deployed to a machine with a CAS worker.

- When CAS is deployed across multiple machines, each machine requires passwordless SSH to communicate with the other machines. Passwordless SSH is set up by the playbook by default. You can edit the passwordless SSH settings in the vars.yml file.

The following seven-machine deployment is the primary example shown in “Multiple Machine Deployment” on page 57.

Figure 1.4 Full Deployment with a Distributed CAS Server



TIP A shared file system is recommended. The shared file system should reside on a machine other than the primary CAS controller or the secondary CAS controller. However, both controllers would use the shared file system to store data and configuration information. If the CAS primary controller fails, the secondary CAS controller could then assume the controller role and use the same data. Information about enabling a shared file system is provided later in this guide.

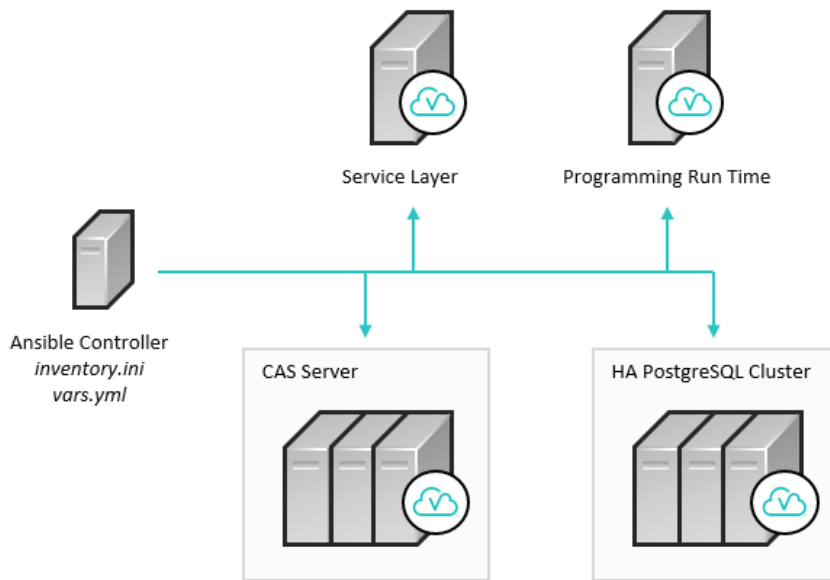
Cluster for High Availability PostgreSQL

SAS Viya uses High Availability (HA) PostgreSQL as the SAS Infrastructure Data Server. SAS Infrastructure Data Server stores user content such as reports, custom groups, comments, authorization rules, selected source definitions, attachments, audit records, and user preferences.

By default, Ansible deploys HA 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 HA PostgreSQL cluster to achieve higher performance and to support redundancy. For more information, see “Creating High Availability PostgreSQL Clusters” on page 159.

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

Figure 1.5 High Availability PostgreSQL Horizontal Cluster



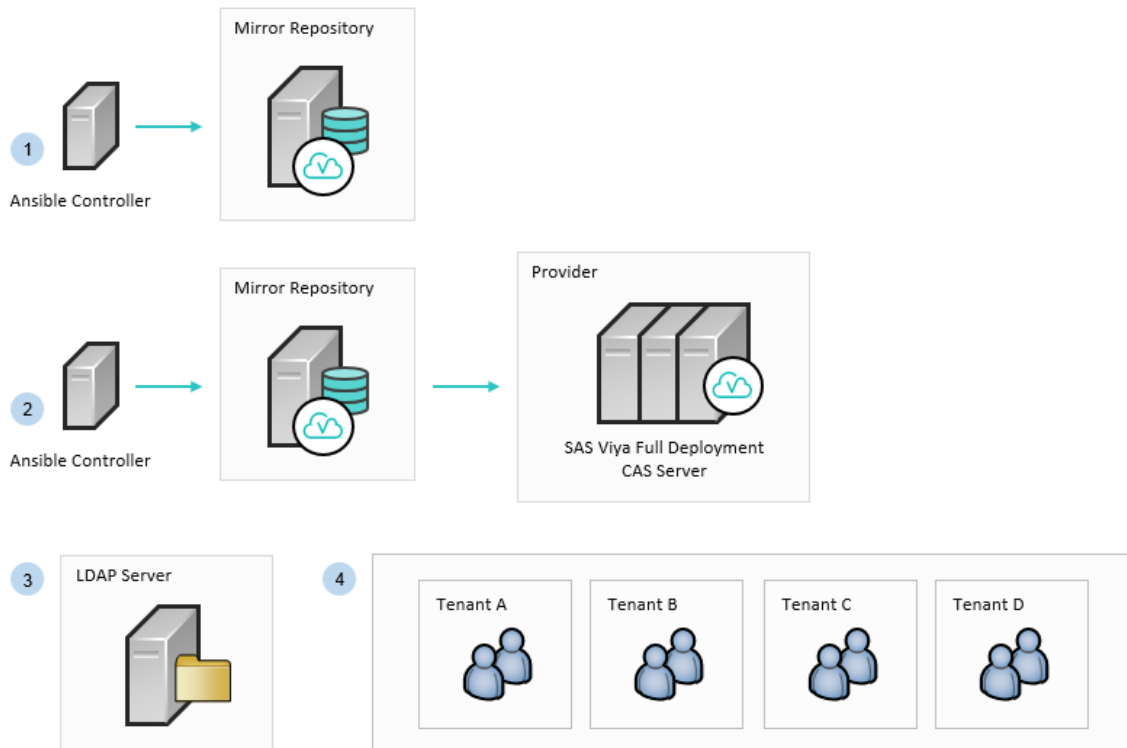
Multi-tenant Environment

SAS Viya supports a multi-tenant environment where a single instance of SAS Viya can serve multiple tenants. A single instance of SAS Viya is deployed to one or more machines. If you opt for multi-tenancy, you must enable it during the initial deployment. If you enable multi-tenancy, an initial tenant for the provider (provider-tenant) is created during deployment. After deployment, you can onboard additional tenants.

CAUTION! You cannot retrofit multi-tenancy in an environment where multi-tenancy was not enabled in the initial, provider deployment.

To enable a multi-tenant environment and then to onboard the tenants, you will consult two documents from SAS: this deployment guide, and *SAS Viya Administration*, which includes instructions for configuring required LDAP settings, and tasks for onboarding and managing tenants. Refer to the following table for information to understand the general tasks for deploying SAS Viya in a multi-tenancy environment.

Figure 1.6 Guidance Deploying and Configuring SAS Viya and Multi-tenancy



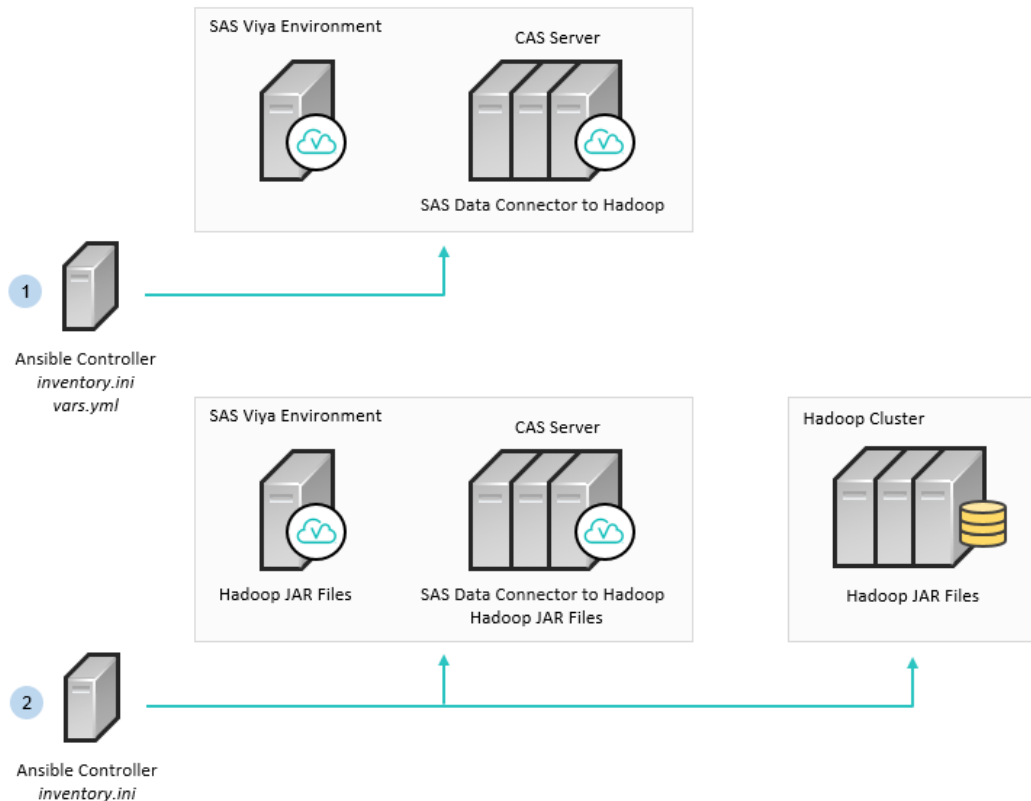
- 1 Ansible is used to set up a mirror repository. A mirror repository is recommended to ensure that the same software versions are used across tenants. For more information, see [“Creating and Using Mirror Repositories”](#) on page 189.
- 2 Ansible is used to deploy the provider. Before run the playbook, you must enable multi-tenancy, which includes editing the vars.yml and sitedefault.yml files. For more information, see [“SAS Viya and Multi-tenancy”](#) on page 83.
- 3 A single LDAP server is configured. Multiple LDAP servers are not supported. Information about the required LDAP settings is provided in [Multi-tenancy: Initial Tasks](#) in *SAS Viya Administration*.
Note: You can configure the LDAP server for multi-tenancy before or after you deploy SAS Viya. The LDAP server must be configured for multi-tenancy before you can onboard tenants.
- 4 Tenants are onboarded. For more information, see [Multi-tenancy: Initial Tasks](#) in *SAS Viya Administration*.

Hadoop Integration: Access Data in Hive

The following examples provide guidance for deploying SAS Viya to support accessing data in Hive.

In the first example, SAS Data Connector to Hadoop is configured to allow the serial loading of data from Hive tables into a distributed CAS server.

Figure 1.7 Access Data in Hive: Serial Processing



1 Ansible is used to deploy the SAS Viya environment, including a distributed CAS server. Before you begin the deployment process, an important task is to configure SAS Data Connector to Hadoop in the vars.yml file.

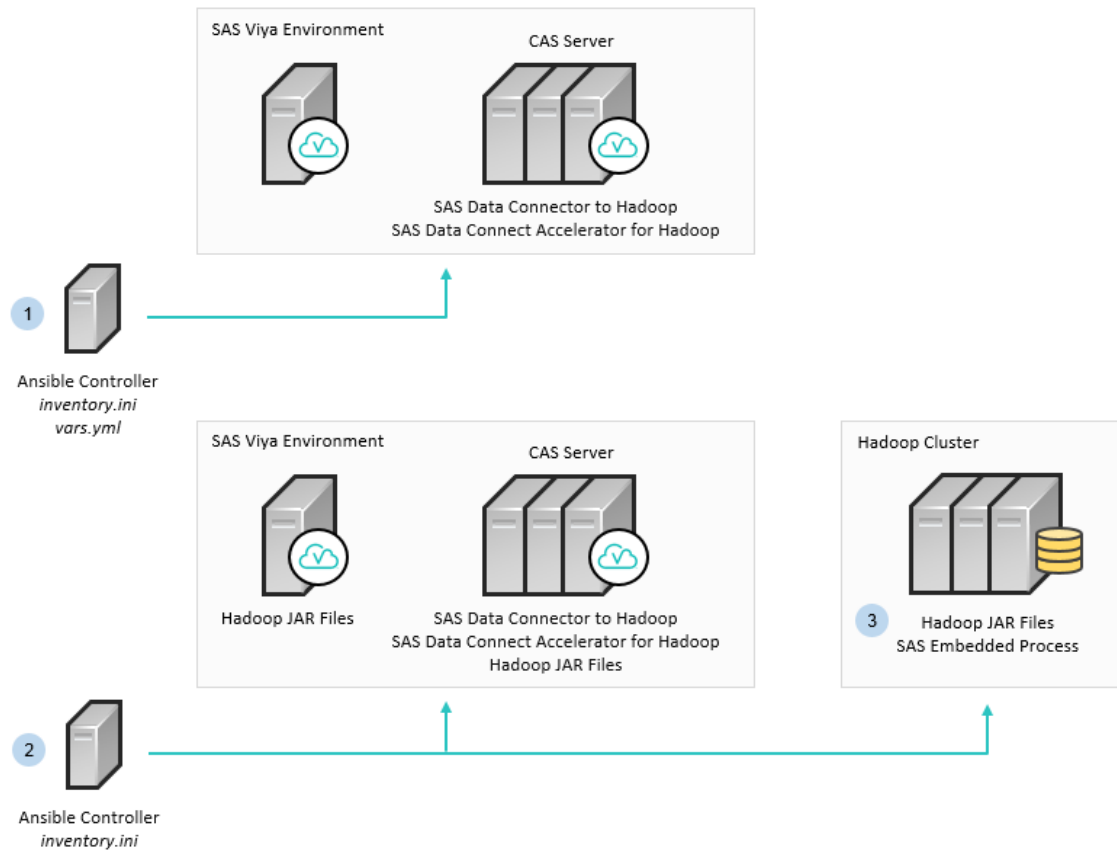
2 Ansible is run again to deploy the Hadoop JAR files to the Hadoop cluster, and then to the CAS controller and SAS programming nodes.

Note: In this example, Ansible is shown to deploy Hadoop JAR files to the required nodes. As an alternative, SAS provides the `hadoop_extract` script, which you can run manually to collect and deploy the Hadoop JAR files.

For more information about deploying the Hadoop JAR files, see [“Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop”](#) on page 203.

In the next example, SAS Data Connect Accelerator for Hadoop is configured to allow the parallel loading of Hive tables into a distributed CAS server. Also, the SAS Embedded Process, which is deployed to the Hadoop cluster, is used to provide high-speed parallel data transfer between Hive and the distributed CAS server. Processing occurs at the CAS server.

Figure 1.8 Access Data in Hive: Parallel Processing



- 1 Ansible is used to deploy the SAS Viya environment, which includes a distributed CAS server. Before you begin the deployment process, an important task is to configure SAS Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop in the `vars.yml` file.
- 2 Ansible is run again to collect the Hadoop JAR files from the Hadoop cluster, and then to deploy the files to the CAS controller and the SAS programming nodes.
Note: In this example, Ansible is shown to deploy Hadoop JAR files to the required nodes. As an alternative, SAS provides the `hadoop_extract` script, which you can run manually to collect and deploy the Hadoop JAR files.
 For more information about deploying the Hadoop JAR files, see [“Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop”](#) on page 203.
- 3 The SAS Embedded Process is deployed to all nodes in the Hadoop cluster. For more information, see [“Hadoop Deployment: Configuring SAS In-Database Technologies”](#) on page 211.

Hadoop Integration: CAS SASHDAT Access to HDFS

In this section, SAS Plug-ins for Hadoop are deployed to the Hadoop cluster. SAS Plug-ins for Hadoop enable CAS to write SASHDAT file blocks evenly across the HDFS file system. This even distribution provides a balanced workload across the machines in the cluster and enables SAS Viya analytic processes to read SASHDAT tables very quickly.

To support CAS SASHDAT access to HDFS, two deployment examples are described: co-locating the CAS server with Hadoop, and configuring access to a remote Hadoop cluster.

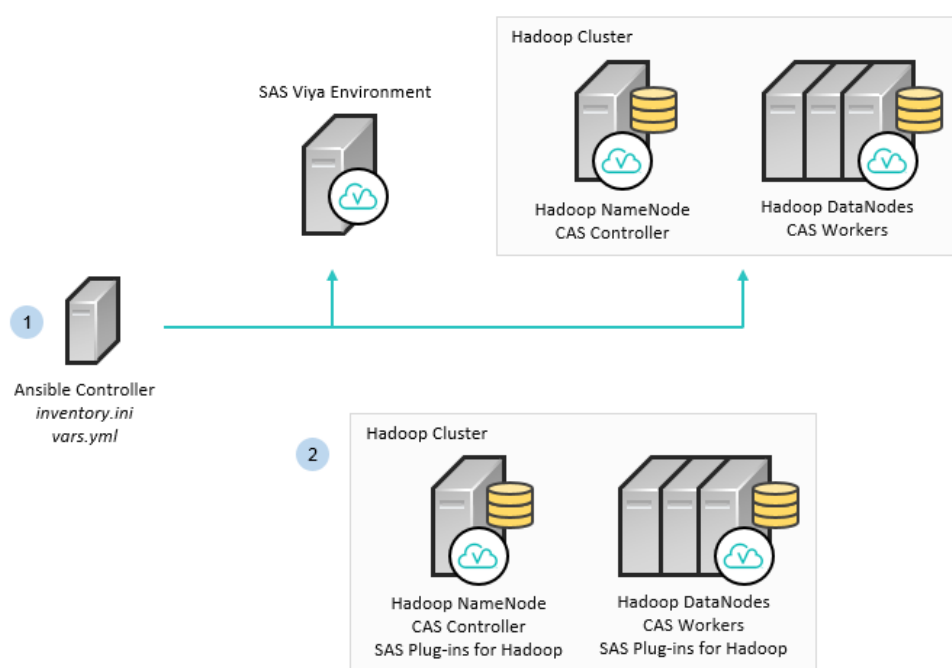
CAS Server Co-located with Hadoop

In this first example, the CAS server is deployed across Hadoop nodes. Specifically, the CAS controller is deployed to the NameNode, and the CAS workers are deployed to all DataNodes or to a subset of the DataNodes. Also, SAS Plug-ins for Hadoop is deployed to the Hadoop NameNode and all the DataNodes.

An advantage of this deployment is that SASHDAT on HDFS can serve as a local cache for CAS tables, which facilitates fast saves and loads of CAS in-memory tables. If the resource demands for your Hadoop cluster leave sufficient capacity for SAS software, consider deploying the CAS server on Hadoop nodes.

Here is an overview of the deployment steps.

Figure 1.9 CAS Deployed on All Hadoop Nodes



- 1 Ansible is used to deploy the CAS controller to the NameNode, and to deploy CAS workers to the DataNodes. Also, the SAS Viya applications are deployed. Before you begin the deployment process, an important task is to set the colocation environment variable and the Hadoop environment variable in the `vars.yml` file.

- 2 The Hadoop cluster is configured, and SAS Plug-ins for Hadoop is configured on the Hadoop NameNode and all the DataNodes. For more information, see [“Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS” on page 231](#).

Note: SAS Plug-ins for Hadoop is provided as a parcel or a stack, which you can activate with Cloudera Manager or Ambari, respectively. As an alternative, SAS provides a script that you can run to install SAS Plug-ins for Hadoop.

Consider the following information when deciding where to deploy the CAS controllers and the CAS workers:

- If your CAS license permits fewer than the total number of CPU cores in your Hadoop environment, you can deploy the CAS workers to all DataNodes and use a subset of the CPU cores.

- If the CAS workers are deployed to all DataNodes, passwordless SSH is not required for loading SASHDAT tables.
- If the CAS workers are deployed to a subset of the DataNodes, consider the following information:
 - Passwordless SSH is required in order to load SASHDAT tables.
 - When you save a SASHDAT table, data is still written locally. Therefore, the `env.CAS_ENABLE_REMOTE_SAVE` environment variable does not have to be defined. Also, data is written in parallel only to those DataNodes on which a CAS worker is deployed.
 - SAS Plug-ins for Hadoop is configured on the Hadoop NameNode and all the DataNodes.
- If you move the CAS controller to a DataNode, SAS recommends that you define the `HADOOP_NAMENODE` environment variable.

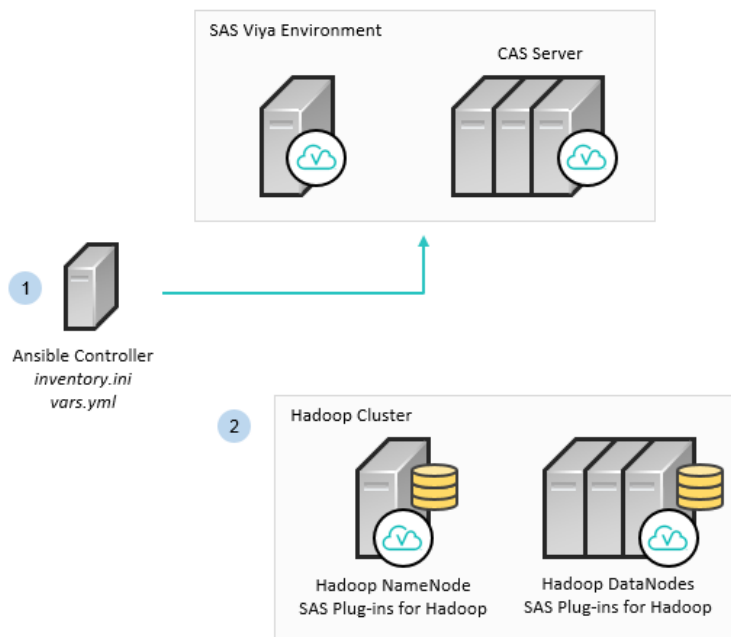
For information about CAS environment variables, see [CAS Environment Variables](#) in *SAS Viya Administration*.

Remote Access to HDFS

In this example, a distributed CAS server is deployed to machines that are not part of the Hadoop cluster. SAS Plug-ins for Hadoop is deployed to the Hadoop cluster, which enables a remote, parallel connection between the HDFS nodes and the CAS server.

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

Figure 1.10 Remote Access to HDFS



1 Ansible is used to deploy the SAS Viya environment, which includes a distributed CAS server. Before you begin the deployment process, an important task is to set the Hadoop environment variables in the vars.yml file.

2 The Hadoop cluster is configured, and the SAS Plug-ins for Hadoop are configured on each Hadoop node. For more information, see [“Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS”](#) on page 231.

Note: SAS Plug-ins for Hadoop is provided as a parcel or a stack, which you can activate with Cloudera Manager or Ambari, respectively. As an alternative, SAS provides a script that you can run to install SAS Plug-ins for Hadoop.

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

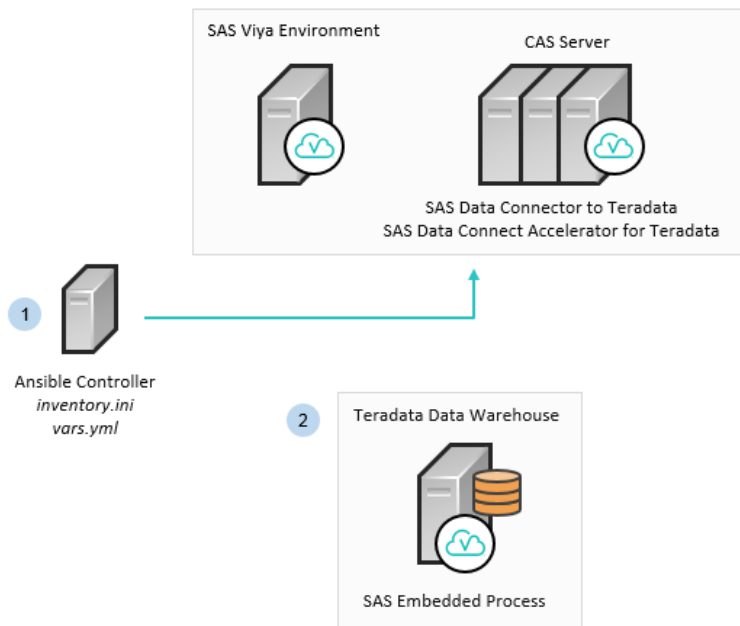
Teradata Integration

SAS provides two data connectors, which can enable access to data in Teradata.

- SAS Data Connector to Teradata enables you to load data serially from the Teradata data warehouse to the CAS server.
- SAS Data Connect Accelerator for Teradata enables you to load data in parallel using the SAS Embedded Process. SAS Embedded Process, which resides on the data appliance, is used to provide high-speed parallel data transfer between the Teradata data warehouse and the CAS server. The data is processed at the CAS server.

The following example provides guidance for deploying SAS Viya to support accessing data in Teradata.

Figure 1.11 Access to Data in Teradata: Serial and Parallel Processing



- 1 Ansible is used to deploy the SAS Viya environment, which includes a distributed CAS server. Before you begin the deployment process, you must configure SAS Data Connector to Teradata and SAS Data Connect Accelerator for Teradata in the `vars.yml` file.
- 2 The SAS Embedded Process is deployed on the Teradata appliance. For more information, see [“Teradata Deployment: Configuring SAS In-Database Technologies”](#) on page 249.

Contact SAS Technical Support

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

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

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

System Requirements

Hardware Requirements	18
Host Requirements	18
General Hardware Considerations	18
File System and Storage Requirements	21
(Optional) High-Availability Requirement	22
Operating System Requirements	23
Supported Operating Systems	23
Linux Requirements	23
Additional Linux Requirements for SAS Event Stream Processing	24
SAS Support for Alternative Operating Systems	24
Server Software Requirements	25
Java	25
Apache httpd	25
Data Source and Storage Requirements	26
Overview of Data Source Requirements	26
Supported Data Sources	26
Hadoop Requirements	27
Requirements to Transfer Data from SAS 9.4	28
Requirements for SAS/ACCESS Interface to Amazon Redshift	28
Requirements for SAS/ACCESS Interface to DB2	28
Requirements for SAS/ACCESS Interface to Greenplum	28
Requirements for SAS/ACCESS Interface to Hadoop	28
Requirements for SAS In-Database Technologies for Hadoop	28
Requirements for SAS/ACCESS Interface to HAWQ	29
Requirements for SAS/ACCESS Interface to Impala	29
Requirements for SAS/ACCESS Interface to Microsoft SQL Server	29
Requirements for SAS/ACCESS Interface to MySQL	29
Requirements for SAS/ACCESS Interface to Netezza	29
Requirements for SAS/ACCESS Interface to ODBC	29
Requirements for SAS/ACCESS Interface to Oracle	29
Requirements for SAS/ACCESS Interface to PC Files	30
Requirements for SAS/ACCESS Interface to PostgreSQL	30
Requirements for SAS/ACCESS Interface to SAP HANA	30
Requirements for SAS/ACCESS Interface to R/3	30
Requirements for SAS/ACCESS Interface to Teradata	30
Requirements for SAS In-Database Technologies for Teradata	30
User and Group Requirements	31
Set Up the User Account that Deploys the Software	31
Set Up the cas Account	31
Set Up Additional User Accounts	32
User Accounts (Reference)	34

Security Requirements	36
LDAP Requirements	36
Transport Layer Security	36
Security for a Programming-only Deployment	37
Transport Layer Security Requirements for the SAS Embedded Process	37
Requirements to Support Multi-tenancy	38
Client Requirements	38
Web Browsers for SAS Viya User Interfaces	38
Mobile Platform Support	38
Database Drivers	38
Screen Resolution	39
Deployment Tools	39
Ansible Controller Requirements	39

Hardware Requirements

Host Requirements

Each target machine in your SAS Viya 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 [“Single Machine Deployment” on page 57](#).

- 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 SAS Viya deployment. Therefore, SAS recommends that before you start the installation, you work with your network administrator to ensure that IP addresses and host names are static.

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

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

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

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

General Hardware Considerations

A full deployment is the recommended configuration for most customer requirements. For most environments, SAS recommends a full deployment of SAS Viya on multiple machines for improved performance.

SAS strongly recommends consulting with a sizing expert to obtain an official hardware recommendation that is based on your deployment type, 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 popular, 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.

The CAS Server and the Programming Runtime must not exceed your licensed core count, but the microservices and web applications are not similarly restricted. Adding RAM to the CAS Server and Programming Runtime machines should improve performance. Adding both CPU cores and RAM to the machines that host the microservices can also improve performance.

- Test machines were equipped with RAM that had a minimum memory clock speed of 1600 MHz.

Architectural Considerations

SAS Viya is built for a scalable, flexible architecture. If the amount of data that is processed is relatively small, a programming-only deployment of a few SAS Viya products will perform well on a Linux VM running on a standard 4-core computer. A full deployment that includes the visual applications requires more resources. However, in both situations, SAS Viya can scale across many nodes to meet the requirements of a particular enterprise. Therefore, to properly size your license and to select the appropriate hardware must be based on an understanding of the planned usage of the software in your environment.

The SAS Viya architecture consists of three categories of components that you should consider as you plan your deployment. These components can all be installed on the same host, or they can be distributed over multiple hosts:

- CAS Server

The CAS Server is required for all deployments, regardless of type (full or programming-only). It is licensed by CPU core, with a minimum license size of 4 cores.

The amount of RAM that is required for the CAS Server is determined by the amount of data that is processed, and by the level of user activity in the environment. However, out of the box, the amount of RAM that is required to start the CAS Server is less than 1 GB.

- Programming Run time

The Programming Run time consists of multiple components that are required for all deployments, regardless of type (full or programming-only). It includes the SAS compute server, SAS Foundation, SAS Studio, SAS Workspace Server, SAS/CONNECT Server, and any SAS/ACCESS engines that you have licensed.

The number of CPU cores that are required for the Programming Run time also depends on the license that you purchased. However, the minimum requirement is 2 cores. SAS recommends that you allocate at least 4 cores for optimal performance.

The minimum required amount of RAM for the Programming Run time is 4 GB. SAS recommends that you allocate at least 16 GB of RAM, or 4 GB for each CPU core.

- Service Layer

This category consists of components that are required for a full deployment, as well as services that support specific SAS products. The components of the Services Layer are not usage restricted. They include the Core Services host group and all the other services that support SAS Viya analytics processing.

The host groups that compose the Service Layer can be deployed on multiple hosts, and with as many CPU cores as are needed for optimal performance and availability.

Hardware Requirements by Product

The following table lists products that can be separately licensed and indicates the RAM and number of CPU cores to support individual components when they are installed on a single machine. The final row indicates the requirements when all products are installed on the same machine. These out-of-the-box requirements can be increased for larger deployments.

The table represents what is required to start all system services and to operate against a small amount of data. These guidelines do not attempt to account for all ordering scenarios, but instead are intended to illustrate typical software orders.

Products	RAM (GB)	CPU Cores
SAS Visual Analytics	48	8
SAS Visual Analytics and SAS Visual Statistics	48	8
SAS Visual Analytics, SAS Visual Statistics, and SAS Visual Data Mining and Machine Learning	64	12
SAS Visual Analytics and SAS Visual Forecasting	64	12
SAS Visual Analytics and SAS Visual Text Analytics	72	12
SAS Visual Analytics and SAS Data Preparation	56	8
SAS Visual Analytics and SAS Data Quality	56	8
SAS Visual Analytics and SAS Decision Manager	56	12
SAS Visual Analytics and SAS Model Manager	56	8
All products listed above	96	16

SAS Viya installs executables and creates configuration directories in `/opt/sas/`. The minimum available disk space that is required to install and start a full deployment of SAS Viya is less than 50 GB. However, logs and operational data will quickly 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 21](#).

These out-of-the-box requirements will accommodate the CAS Server (to the extent that the resources comply with your product licensing terms) as well as the Programming Run-time for validation purposes or light usage.

Be aware that the start-up times for the various services in the environment and the level of performance that they deliver will improve as CPU cores are added.

Hardware Requirements for SAS Deep Learning

SAS Deep Learning is included with SAS Visual Data Mining and Machine Learning. Use the requirements for SAS Visual Data Mining and Machine Learning to prepare your target machines. The two products are automatically installed together.

To enable SAS Deep Learning, the following additional requirements apply:

- A powerful graphical processing unit (GPU). SAS recommends, and has tested on, the NVIDIA Tesla P100 and K40.
- The NVIDIA drivers for Tesla. You can download the current drivers from <http://www.nvidia.com/download/driverResults.aspx/118959/en-us>.

You must also install kernel headers and development packages that correspond to the running version of the operating system kernel. For more information, see the [NVIDIA CUDA Installation Guide for Linux](#).

- The NVIDIA CUDA Toolkit for GPU. SAS recommends version 7.5 or later. You can obtain the toolkit at <http://developer.nvidia.com/cuda-zone>.

Make sure that you have met all of these requirements before you start the SAS Viya deployment process.

File System and Storage Requirements

Disk Space and Storage Requirements

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

SAS Viya 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 SAS Viya volume or of moving the volume to another location at a later time.

CAS servers automatically cache blocks of data on disk when they are working with tables whose size exceeds the available memory. The CAS controller also uses disk space for the CAS cache directory. You can configure the location of this cache in the playbook. For more information, see [Set Up the CAS Cache Directory](#).

If your order included SAS Model Studio, consider allocating additional storage space in `/opt/sas/viya/config/data/cas/default/projects`. SAS Model Studio copies the data source when a project is created. Therefore, the amount of space that is required depends on the number of saved projects and on the size of the data source. Each project that end users create will require space in this directory structure until the project is deleted. SAS recommends allocating a minimum of three times the size of the data source. For optimal performance, create this directory structure on a high-performance storage appliance, such as a SAN or another multi-device appliance.

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 SAS Viya log directory (`/var/log`) to a directory that has additional free space:

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

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

The 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.

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

Requirements for Caslib Data Access

A caslib is an in-memory space that enables the CAS server to hold tables, access control lists, and data source information. All data is available to CAS through caslibs, and all operations in CAS that use data are performed with a caslib in place. Among other functions, caslibs provide access to data from the data source and access to in-memory tables that are copied from the data source. Here are the requirements to enable users to share data that is contained in caslibs and to support CAS failover:

- In some situations, caslibs require additional persistent storage.
- To support an environment where a secondary controller is configured for failover, a shared file system is required.

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

Multiple predefined system caslibs and the Public caslib have a default location for persistent storage: `/opt/sas/viya/config/data/cas/instance-name/name-of-Public-caslib`. You can specify the instance name when you edit the playbook. If you anticipate that many users will use browsers to access the user interfaces and to import data from files, additional space for this file system will be required. SAS recommends monitoring disk usage at `/opt/sas/viya/config/data/cas`.

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

You can deploy a primary CAS controller and a secondary CAS controller to support failover. Up to two controllers are supported. For any environment with multiple controllers, a common, shared file system must be available to both controllers. If you are deploying for multi-tenancy, plan to use a shared file system for all the path-based caslibs in each tenant because each tenant will have its own CAS controller. Mount the file system at `/opt/sas/tenant/config/cas/data`. For *tenant*, substitute *viya* for a single tenant, or substitute the name of one of your tenants.

Note: If you mount the shared file system before running the playbook, be aware that the installation runs in this path as root. The root user therefore requires permissions to modify the file system across the mount. After the installation has completed, root permissions are not required. To avoid this requirement, you can set up this file system post-deployment. For more information, see [Set Up a Shared File System for CAS Controllers \(Post-Deployment\)](#) in *SAS Viya 3.3 Administration: SAS Cloud Analytic Services*.

Some caslibs use a path to a directory as the data source. Therefore, in an environment with a secondary controller, the directory must be a network location that is available at the same path on both controllers. A file system that is mounted at this location accommodates all caslibs that are created by SAS. Similarly, enable a shared file system if you are deploying a secondary controller, and you are deploying the CAS server on a massively parallel processing (MPP) system. For more information, see ["Enable a Shared File System" on page 47](#).

(Optional) High-Availability Requirement

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

SAS Studio users can set preferences and store projects, which means that all instances of SAS Studio must be able to access the same saved configuration data. You must fulfill one of the following requirements to support SAS Studio in a high-availability configuration:

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

This option requires additional setup to instruct SAS Studio to use the shared drive. For more information, see [SAS Studio: Configuration Properties](#).

Operating System Requirements

Supported Operating Systems

For the full list of supported operating systems, see

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

Linux Requirements

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

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

```
subscription-manager version
```

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

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

A list of active subscriptions is returned.

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

```
uln_register
```

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

If you have enabled Security-Enhanced Linux (SELinux) in your environment, you must enable permissive mode on all of the target machines in your deployment. For more information, see [“Configure SELinux” on page 46](#).

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

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

The following libraries are required:

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

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

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

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

```
$ yum update systemd
```

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

Be sure to follow the steps that are described in “[Perform Linux Tuning](#)” on [page 50](#) before starting the deployment process.

Additional Linux Requirements for SAS Event Stream Processing

This information is relevant for users of SAS Event Stream Processing. The ESP server libraries were built using gcc-4.4.7-16 and the Boost library 1.58. The Boost library 1.58 is automatically installed with SAS Event Stream Processing. The libraries were compiled using the following compiler options:

```
-D_REENTRANT
```

```
-D_THREAD_SAFE
```

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

The SAS Event Stream Processing 5.x libraries have been built using gcc-4.4.7-16 on Red Hat Enterprise Linux Server 6.7 using libc-2.12.so, libstdc++.so.6.0.13, and libgcc_s-4.4.7-20120601.so.1.

SAS Support for Alternative Operating Systems

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

Server Software Requirements

Java

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

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

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

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

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

```
java -version
```

Apache httpd

The deployment process 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 [“Transport Layer Security” on page 36](#).

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 58](#). 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

Overview of Data Source Requirements

You can install software to enable data retrieval from a Hadoop data store and from various data storage appliances. Depending on one or more of your data sources, you might also install one or more SAS/ACCESS products and a SAS In-Database Technologies product on your CAS controller and CAS workers.

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

- The database client for your associated database software. You might need to install the database client on the CAS controller.
- Drivers or other requirements for the SAS data connector that is included with SAS/ACCESS for use with your data source. The appropriate SAS data connector is installed by the SAS deployment onto the CAS controller and all CAS worker machines. You must install any drivers or other required software on the CAS controller.

Refer to the section that corresponds to your SAS/ACCESS product or SAS In-Database Technologies product for additional system requirements that apply to the CAS controller and CAS workers.

Supported Data Sources

SAS Viya supports the following external data sources, which require a SAS/ACCESS product. Some of these data sources offer an optional SAS In-Database technology bundle to support parallel execution. In some cases, these products might have individual requirements:

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

Support for the following external data sources is limited to integration with SAS 9.4. These sources also require a SAS/ACCESS product but do not offer a data connector to CAS:

- Apache HAWQ
- Greenplum
- IBM Netezza
- MySQL
- SAP R/3

SAS Viya also supports data sources that use a SAS data connector that is included with CAS and is not separately licensed or configured. Support for the following data sources is automatically included:

- SASHDAT on HDFS
- LASR Analytic Server (SAS 9.4)
- SAS Scalable Performance Data Engine (SPDE)
- SAS data sets

SAS Viya also supports CSV files, which do not require a SAS data connector or SAS/ACCESS product and can be accessed directly.

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

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.

Note: If you plan to deploy multiple tenants, be aware that only a single SAS Infrastructure Data Server is supported for all tenants. However, an individual tenant can deploy and use a different external data source. The corresponding SAS/ACCESS product would be required.

Hadoop Requirements

Supported Distributions and Connection Requirements

SAS Viya supports multiple third-party distributions of Hadoop.

Note: If you upgrade your Hadoop version and have already deployed SAS Viya with SASHDAT, then you must perform steps to redeploy SAS Viya with Hadoop. For more information, see [SAS Note 60118](#).

For the full list of supported Hadoop distributions, see: <https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-databases.html>.

You can connect to data as follows:

- For SASHDAT on HDFS, CAS components are typically installed on all SAS servers in your deployment and on every machine in your Hadoop cluster. No additional SAS data connector setup is required.

Loading a table from HDFS requires Read/Write access to the `/tmp` directory. Permission should be granted to the user ID of the session process, which is either the cas user ID or the host account of an end user who is starting a session.

- For Hive, SAS/ACCESS Interface to Hadoop and possibly SAS In-Database Technologies for Hadoop are required. These products have individual system requirements, which are documented below.

Note: Apache Hadoop 0.23, 2.4.0, and 2.7.1 and later versions are supported only as a Hadoop cluster that is co-located with CAS for access to SASHDAT on HDFS.

SAS Support for Alternative Releases of Hadoop Distributions

SAS identifies the specific set of Hadoop distributions that are supported with each SAS product release. The SAS policy that applies to alternative releases or distributions of Hadoop is documented at the following website: <http://support.sas.com/resources/thirdpartysupport/v94/hadoop/alternative-hadoop-distributions.html>. The same policy that applies to SAS 9.4 also applies to SAS Viya.

Requirements to Transfer Data from SAS 9.4

For SAS 9.4 deployments that are earlier than SAS 9.4 TS1M5 (SAS 9.4M5), SAS/CONNECT is required in the environment in order to transfer data from other SAS deployments and operating systems to SAS Viya.

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

By contrast, SAS 9.4M5 is integrated with SAS Viya directly. As a result, SAS/CONNECT is no longer required in order to transfer data from SAS 9.4M5. All SAS programming clients in a 9.4M5 environment can call procedures that are enabled in SAS Viya and submit DATA step code, operating directly on CAS data sources. Examples of SAS programming clients are SAS Studio, SAS Enterprise Guide, SAS Data Integration Studio, and SAS Data Management Studio.

SAS/CONNECT is still supported, but if you are running SAS 9.4M5, it is no longer required in order to transfer data into SAS Viya.

Requirements for SAS/ACCESS Interface to Amazon Redshift

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

For information about supported Amazon Redshift versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to DB2

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

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

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

Requirements for SAS/ACCESS Interface to Greenplum

SAS/ACCESS Interface to Greenplum (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported Greenplum versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to Hadoop

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

For information about supported Hadoop versions and additional requirements, see:

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

Requirements for SAS In-Database Technologies for Hadoop

SAS In-Database Technologies for Hadoop (on SAS Viya) includes SAS Data Connect Accelerator for Hadoop.

For information about supported Hadoop versions and additional requirements, see:

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

Requirements for SAS/ACCESS Interface to HAWQ

SAS/ACCESS Interface to HAWQ (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported Apache HAWQ versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to Impala

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

For information about supported Impala versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to Microsoft SQL Server

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

For information about Microsoft SQL Server support, see:

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

Requirements for SAS/ACCESS Interface to MySQL

SAS/ACCESS Interface to MySQL (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported MySQL versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to Netezza

SAS/ACCESS Interface to Netezza requires SAS Foundation and SAS Viya.

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

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

Requirements for SAS/ACCESS Interface to ODBC

SAS/ACCESS Interface to ODBC (on SAS Viya) 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.

For information about ODBC support, see:

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

Requirements for SAS/ACCESS Interface to Oracle

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

You must install the Oracle client on the CAS controller.

For information about supported Oracle versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to PC Files

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

SAS/ACCESS Interface to PC Files enables access to the following file formats:

- .jmp
- .spss
- .stata
- .xlsx or .xls

No additional software is required.

Requirements for SAS/ACCESS Interface to PostgreSQL

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

For information about supported PostgreSQL versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to SAP HANA

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

For information about supported SAP HANA versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to R/3

SAS/ACCESS Interface to R/3 (on SAS Viya) requires SAS Foundation and SAS Viya.

For information about supported SAP R/3 versions and requirements, see:

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

Requirements for SAS/ACCESS Interface to Teradata

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

For information about supported Teradata versions and requirements, see:

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

Requirements for SAS In-Database Technologies for Teradata

SAS In-Database Technologies for Teradata (on SAS Viya) includes SAS Data Connect Accelerator for Teradata and SAS Data Quality Accelerator for Teradata.

SAS In-Database Technologies for Teradata requires SAS/ACCESS Interface to Teradata. The SAS Embedded Process for Teradata is included with SAS In-Database Technologies for Teradata.

For information about supported Teradata versions and additional requirements, see:

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

User and Group Requirements

Set Up the User Account that Deploys the Software

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

- Super user (sudo) or root access.

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

```
sudo -v
```

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

```
sudo -l
```

- Appropriate permissions to create subdirectories in the directory path where you have saved the playbook. The recommended path is `/sas/install/sas_viya_playbook`. For more information, see [“Make Sure That You Have the Required Files” on page 41](#).
- A home directory.
- A subdirectory for the installation user’s SSH keys: `~/ .ssh`.

Set Up the cas Account

If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

A user account and a group are required for the deployment. As part of the pre-deployment validation, the playbook checks for a user account named `cas` and its membership in a group named `sas`.

This user account is critical to the deployment. As a member of the Super User role in the visual administration interface (SAS Environment Manager), the `cas` user account always has unrestricted access to CAS. It functions as a back-end service account.

- If you plan to use the customized script to perform a yum deployment, the required user account and the group are created automatically.
- If the policies in your environment do not allow for the creation of a `cas` user and a `sas` group, identify alternative but equivalent values for `casenv_user` and `casenv_group` in the `vars.yml` file before you run the playbook. Make sure that the alternative group is the primary group of the alternative `casenv_user`. For more information, see [“Set Up the CAS Admin User” on page 67](#).

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

- 1 Create the group named `sas`, or its equivalent, if one does not exist.
- 2 Create the user account. The recommended user name is `cas`. Assign the user account to the `sas` group or an equivalent group.
- 3 Make sure that the `cas` account has a login shell (`/bin/bash` is recommended).
- 4 Verify that the user account exists on each host where a CAS component is running. Also, verify that the account has a consistent UID and GID on all machines in your deployment.

Note: Use the **usermod** command to align the UIDs of any mismatched user accounts. For any groups with mismatched GIDs, use the **groupmod** command.

An SSH public key is required in the `$HOME/.ssh` directory of the `cas` user. The playbook can perform this step automatically, or you can configure your own passwordless SSH for the `cas` user before the deployment. For more information, see [“Set Up Passwordless SSH for CAS” on page 65](#).

When the deployment completes, the `cas` user (or the equivalent user that you configured in `vars.yml`) might not have logon access to SAS Studio or to CAS Server Monitor. Use an LDAP account for this purpose. For more information, see [“Set Up Administrative Users” on page 93](#)

Set Up Additional User Accounts

If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#). For information about user account requirements for these products, see [“User Accounts \(Reference\)” on page 34](#).

A few other user accounts are required in order to configure and run the software after the deployment process has completed. The information in the following sections provides an overview of the requirements that apply to user accounts for a programming-only deployment and a full deployment. In a full deployment, prepare accounts for both programmers and non-programmers in order to access all user interfaces.

Set Up Accounts for Users of the Visual Interfaces

Note: This section applies only to a full deployment.

The following requirements apply to the user accounts that can access SAS Viya visual interfaces, including SAS Visual Analytics:

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

Set Up Accounts for Programmers

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

In a programming-only deployment, users will not log on to SAS Viya visual interfaces. Instead, they will log on to SAS Studio. Account requirements for programmers resemble the requirements for the `cas` account. However, the following factors apply to these users:

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

Set Up Accounts for Compute Server Users

The SAS compute server and its compute service enable end users to submit SAS programs and jobs for processing. Several SAS Viya products use the compute service to process programs and statements that were written in the SAS language. For more information, see [SAS Compute Server and Service](#) in *SAS Viya Administration: Programming Environment Servers*.

Compute servers are launched in the operating system under the user ID of the requesting user. Therefore, your authentication scheme must be configured to enable the operating system on each of the machines where the compute server is running to recognize the users' LDAP identities. For example, you can use PAM or SSSD to enable this integration.

The following products use compute server functionality: SAS Visual Data Mining and Machine Learning, SAS Model Manager, SAS Visual Forecasting, and SAS Decision Manager.

Users of these products require the following resources:

- LDAP accounts for the visual interfaces
- recognition of their LDAP accounts by the operating system where the compute server is installed
- a home directory that can be accessed by the compute server each time that a process starts

Set Up Accounts for Multi-tenant Deployments

SAS Viya supports a multi-tenant configuration. To enable multi-tenancy, you must select a multi-tenant deployment in the `sitedefault.yml` file before you run the playbook.

All tenants share a single LDAP server. Within that server, each tenant must have a unique organizational unit (OU) defined, with a separate OU for the provider. After the playbook has been run, the system includes a single tenant that can be accessed only by the users in the provider OU. The creation of additional tenants takes place post-deployment.

Before you modify your playbook to prepare for the installation, perform initial configuration of your LDAP server for multi-tenancy:

- 1 Create the provider OU. Here is an example:

```
dc=example,dc=com
  ou=provider
  ou=groups
  ou=users
```

The following detailed example uses LDIF syntax:

```
dn: cn=sas,ou=groups,ou=provider,dc=sas,dc=com
distinguishedName: cn=sas,ou=groups,ou=provider,dc=sas,dc=com
displayName: Tenant-admin-group-for-provider
gidNumber: value
objectClass: groupOfUniqueNames
objectClass: extensibleObject
uniqueMember: uid=sas,ou=people,ou=provider,dc=sas,dc=com
cn: sas
```

Note: The provider DN must be supplied as `provider`.

- 2 Verify that the `sas` group within the provider OU has a consistent GID on all machines in the deployment.

You were instructed to create the `sas` group in “[Set Up the cas Account](#)” on page 31. Multi-tenant deployments apply extra security in the `/opt/sas/viya/config` directory. To ensure that users within the provider tenant have access to critical services (including SAS Studio, the SAS compute server, and others),

all provider users must be members of the sas group. If they are not members of the SAS group, the additional security will prevent them from using those servers and services.

3 Configure a cas user within the provider OU.

The following detailed example uses LDIF syntax:

```
dn: uid=cas,ou=people,ou=provider,dc=sas,dc=com
uid: cas
cn: cas
sn: Admin
loginShell: /bin/bash
distinguishedName: uid=cas,ou=people,ou=provider,dc=sas,dc=com
displayName: CAS-Administrator-and-process-owner
userPassword: password
objectClass: inetOrgPerson
objectClass: extensibleObject
uidNumber: value
gidNumber: value
homeDirectory: /home/cas
```

You were instructed to create the cas user (or its equivalent) in [“Set Up the cas Account” on page 31](#). If you wait until after the deployment process has completed to add a cas user to the provider OU, be aware that the UID and GID must match on all machines in the deployment. In addition to this LDAP user, an identical account must exist on each host.

4 (Optional) If your provider or tenants will have secondary CAS controllers to enable failover, set up a shared file system. For more information, see [“Enable a Shared File System” on page 47](#).

The deployment process automatically creates an internal user account for an administrator within the provider tenant. You can set up separate groups for administrative users and for non-administrative users within each tenant OU, and you can add tenant users to one of these groups. The tenant creation process provides these groups with access to critical files and other resources that are otherwise restricted. The users that are defined within tenant OUs should not be added to the sas group. Tenant users must instead be members of their tenant’s user group. For more information about tenant onboarding, see [Multi-tenancy: Initial Tasks](#) in *SAS Viya Administration*.

User Accounts (Reference)

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

The table identifies and describes SAS Viya user accounts. Because these accounts are required for the installation and for running services during the product’s normal operation, do not delete them or change their names. These user accounts do not require root or sudo privileges.

Account Name and Group	Description	Purpose
sas; member of sas group	<p>A non-logon service account without user restrictions.</p> <p>No password. You can add a password after installation, if necessary, but make sure that it does not expire.</p> <p>The default user name is required.</p> <p>A login shell is required.</p> <p>The sas group is an administration group, not a general user group.</p>	<p>Required for the installation, and created automatically.</p> <p>The installation process sets user and group ownership permissions on all of the installation files. This user must exist to enable ownership.</p> <p>After the installation has completed, this user account enables required components to run.</p> <p>The sas group is intended to allow access to administrative features, such as logs and backup. It is the group owner of many files on disk. Restrict membership in this group to administrators.</p>
cas; member of sas group	<p>The process owner of CAS processes. No default password is assigned, but a password is required if you plan to use this account as the CAS administrator. If you are using both operating-system and LDAP accounts, which are required for a full deployment, verify that this user has a single set of credentials that are valid for all applicable authentication providers. In addition, verify that the sas group is this user's primary group.</p> <p>The cas user must be able to connect from the CAS controller to each CAS worker without providing a password. If the CAS server is running in an analytic cluster environment (with multiple CAS workers), passwordless SSH can be configured by the Ansible playbook. For more information, see “Accept the Passwordless SSH Default” on page 65.</p> <p>A login shell is required.</p> <p>The “cas” user name is recommended. This user name enables the deployment to assign SSH keys. To assign a different user name, modify the casenv_user parameter in the vars.yml file.</p>	<p>Required for managing and enabling CAS. If you are using Ansible to deploy SAS Viya, create this user account and add it to the sas group before you start the deployment. (If you are using the customized script that is described in Appendix B: Deploying with Yum, the deployment creates the user account for you.)</p> <p>This user corresponds to the CAS (Superuser) role in the CAS administration interface, CAS Server Monitor.</p>
sasboot	<p>Created during the deployment, with an expired password.</p> <p>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 option in case your identity provider becomes unavailable. For more information, see <i>SAS Viya Administration: Identity Management</i>.</p>	<p>Administrator account that is used for preliminary logon to the visual administration interface.</p> <p>This account is not recognized by SAS Studio, the programming interface.</p>
user account for SAS Studio access	<p>A user account that is defined on the operating system of the machine where SAS Studio will be installed.</p>	<p>User account that is used for preliminary logon to the programming administration interface.</p>

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

After the installation has completed, the sas user account enables required components to run, including the web application server for SAS Event Stream Processing Studio. Sudoers privileges are not required after the installation to run SAS Event Stream Processing. The installation directory path enables write access per user group, and it is owned by the sas user. To grant permission to edit the configuration files, the administrator must add any user requiring write access to these files to the sas group. SAS Event Stream Manager users are authenticated in LDAP when they log on.

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

- apache
- postgres

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

Security Requirements

LDAP Requirements

LDAP is required for SAS Viya visual interfaces, including SAS Event Stream Manager. It is not required in a programming-only deployment.

To support the visual interfaces, SAS Viya must have Read access to your LDAP provider. SAS Viya requires a userDN and password in order to bind to the LDAP server. Anonymous binding is supported for clients that are authenticating to the LDAP server.

If the mail attribute is specified for LDAP accounts, it must have a non-null value that is unique for each user.

LDAP is also required to support multi-tenancy. All tenants must be contained within the same LDAP server. Multiple LDAP servers are not supported. Within that server, each tenant must have a unique organizational unit (OU) defined, with a separate OU for the provider, named `provider`. SAS recommends setting up the provider space before you start the deployment process. For more information, see [“Set Up Accounts for Multi-tenant Deployments” on page 33](#).

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

For most user interfaces, configuring SAS Viya to access your LDAP provider is a post-deployment task. For more information, see [“Configure Your Environment with SAS Environment Manager” on page 91](#). SAS Event Stream Manager is an exception. To configure LDAP to enable access to SAS Event Stream Manager, follow the steps in [“Configure LDAP Settings for SAS Event Stream Manager” on page 82](#) before you run the playbook.

Transport Layer Security

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

SAS Viya uses an Apache HTTP server 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 SAS Viya. The default security is the only security that is available in a programming-only deployment.

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 64](#).
- 4 Perform a full deployment rather than a programming-only deployment.

The playbook can then enhance the security of your SAS Viya 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 SAS Viya user interfaces from a web browser.
- Add custom certificates to the self-signed certificates that a full deployment provides on all machines.
- Upgrade the security protocol and ciphers that are enabled by default using the `sas-ssl.conf` file.

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

Security for a Programming-only Deployment

SAS recommends the installation of a full deployment, which includes the product visual interfaces and microservices. If you instead select programming-only as the deployment type, the playbook cannot provide the same level of security by default.

In a programming-only deployment, TLS is not enabled by default. In addition, SAS Secret Manager, which stores and distributes certificates, is not installed. As a result, network connections from the Apache HTTP Server to backend services are not encrypted.

Transport Layer Security Requirements for the SAS Embedded Process

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

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

Requirements to Support Multi-tenancy

Multi-tenant deployments use Access Control Lists (ACLs) to protect data and configuration of tenants and to restrict access to the provider OU. The file system that contains the `/opt/sas/viya/config` directories for both tenants and the provider must be mounted with support for ACLs.

You must also set up DNS records for tenant-specific subdomains. As the users within each tenant access SAS Viya components, the host name that they use to access the SAS deployment identifies their tenant membership.

Each tenant is reachable by a URL that is derived from the provider's URL. Here is the format for a typical tenant URL

tenant-ID.provider-URL

Here is an example of a provider URL:

`sasviya.mycompany.com`

Here is an example of a tenant URL:

`mytenant.sasviya.mycompany.com`

You must verify that the DNS server for your enterprise is configured to route to these address spaces. You can create a wildcard subdomain entry as a time-saving step.

Client Requirements

Web Browsers for SAS Viya User Interfaces

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

Some SAS Viya user interfaces include some advanced features that require recent versions of popular web browsers. For information about supported web browsers and the corresponding platforms to access SAS user interfaces, see

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

Mobile Platform Support

Support for mobile devices is not yet available for all SAS Viya user interfaces. For information about mobile device support, see

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

Database Drivers

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

Screen Resolution

The minimum screen resolution for each client machine that will access the SAS Viya user interfaces is 1280 x 1024.

Deployment Tools

Ansible Controller Requirements

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

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

A typical Ansible deployment consists of at least one control machine (the Ansible controller) and multiple Ansible managed nodes (the machines where SAS software is installed). In a single-machine deployment, Ansible and all SAS software are installed on the Ansible controller. For more information, see “[Install Ansible](#)” on page 48.

In a distributed deployment, the managed nodes use a secure shell (SSH) framework for connections to the Ansible controller. Verify network connectivity between the controller and the managed nodes. Connectivity is also required between all machines in the deployment and from the controller to the SAS yum repositories. For more information, see “[Firewall Considerations](#)” on page 44.

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

Pre-installation Tasks

<i>Make Sure That You Have the Required Files</i>	41
<i>Confirm the Identities of the Hosts</i>	42
<i>Enable Required Ports</i>	42
<i>Firewall Considerations</i>	44
<i>Configure SELinux</i>	46
<i>Configure a Proxy Server</i>	46
Overview	46
Using curl	46
Using yum	46
<i>Enable the Yum Cache</i>	47
<i>Enable a Shared File System</i>	47
<i>Install Ansible</i>	48
Standard Ansible Installation	48
Streamlined Ansible Installation	48
Test Your Ansible Installation	49
<i>(Optional) Enable Key-Based SSH Authentication</i>	49
<i>Set Environment Variables for SAS Event Stream Processing</i>	50
<i>Perform Linux Tuning</i>	50
Set the MaxStartups Value	50
Set the ulimit Values	51
Set the Semaphore Values	52
Change the Default Time-outs	52

Make Sure That You Have the Required Files

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

```
tar xf SAS_Viya_playbook.tgz
```

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.

Enable Required Ports

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

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

Process	Required Port	Requires Allowed Inbound Traffic From	Notes
SAS Studio	7080 (if you are performing a full deployment, the deployment will use ephemeral ports, so no port needs to be reserved)	SAS Viya Servers only	Not required for SAS Visual Investigator.
Vault	8200	SAS Viya Servers only	
SAS Configuration Server	8300, 8301, 8302, 8400, 8500, 8501, 8600	SAS Viya Servers only	SAS uses HashiCorp Consul as its configuration server. Ports 8301, 8302, and 8600 are open for both UDP and TCP traffic.
Object Spawner	8591		
CAS Server Monitor	8777	anywhere (SAS Viya servers)	Used by clients to make REST HTTP calls to CAS, as with the Python REST interface.
default SAS Messaging Broker management web console port	15672	SAS Viya Servers only	
SAS/CONNECT Spawner management	17541	anywhere (SAS Viya servers, SAS 9.X servers, workstation)	
SAS/CONNECT Spawner	17551	anywhere (SAS Viya servers, SAS 9.X servers, workstation)	
SAS Model Manager launcher context	18201–18250	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
SAS Job Execution launcher context	18501–18600	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
SAS Visual Forecasting launcher context	18601–19000	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
SAS Cloud Analytic Services Server	19990-19999	SAS Viya Servers only	

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

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

The Linux operating system defines a specific series of network service ports as an ephemeral port range. These ports are designed for use as short-lived IP communications and are allocated automatically from within this range. If a required port is within the range of the ephemeral ports for a host, another application can attempt to claim it and cause services to fail to start. Therefore, you must exclude the required ports in the table from the ports that can be allocated from within the ephemeral port range.

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

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

The results contain two numbers:

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

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

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

Here is an example of the results:

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

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

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

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

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

Here is an example:

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

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

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

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

Firewall Considerations

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

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

- <https://linux.oracle.com/>

This website provides instructions for registering with the Oracle ULN.

- 2 Ensure that the firewall allows access to the SAS repositories by running 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 `firewallproxy` whitelist and try the command again. Repeat this step until you successfully transfer the `bigfile.bin` file.

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

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

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

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

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

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

- 4 To stop `iptables`, run the following commands.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

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

To stop `firewalld`, run the following commands.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

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

Configure 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
```

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@proxy-server-FQDN:8080/
export http_proxy=http://user-name:password@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, hostnames, 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 hostnames, 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=proxy-server-FQDN:8080/
proxy_username=user-name
proxy_password=password
```

Enable the Yum Cache

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

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

```
keepcache = 1
```

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

Enable a Shared File System

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

Note: If you prefer to set up the shared file system after the deployment is complete, you can skip the steps in this section and see [Set Up a Shared File System for CAS Controllers \(Post-Deployment\)](#).

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

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

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

The shared file system is mounted for the CAS server and for the tenant in a single-tenant deployment or for the provider in a multi-tenant deployment. For a multi-tenant deployment, you must repeat similar steps for each tenant that will use a secondary CAS controller. For more information about adding tenants, see [Multi-tenancy: Initial Tasks](#).

Install Ansible

Ansible is third-party software that provides automation and flexibility for deploying software to multiple machines. If you decide to use Ansible to deploy your software, you must install a supported version of Ansible.

Standard Ansible Installation

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

Streamlined Ansible Installation

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

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

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

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

- 2 To Install Python PIP and related packages:

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

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

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

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

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

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

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

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

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

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

Test Your Ansible Installation

- 1 To test the Ansible version:

```
ansible --version
```

Here is an example of successful output:

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

- 2 To perform a basic ping test:

```
ansible localhost -m ping
```

Here is an example of successful output:

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

(Optional) Enable Key-Based SSH Authentication

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

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

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

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

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

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

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

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

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

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

Set Environment Variables for SAS Event Stream Processing

If your order included SAS Event Stream processing, you must set some environment variables to enable it to run. For a shell that will only invoke SAS Event Stream Processing, run the following commands:

```
export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0
export LD_LIBRARY_PATH=$DFESP_HOME/lib:/opt/sas/viya/home/SASFoundation/sasexe
export PATH=$PATH:$DFESP_HOME/bin
```

If you need to maintain your LD_LIBRARY_PATH setting for another SAS product, change the second command that is listed above to the following:

```
export LD_LIBRARY_PATH=$DFESP_HOME/lib:/opt/sas/viya/home/SASFoundation/sasexe:$LD_LIBRARY_PATH
```

(Optional) To enable SSL on connections between SAS Event Stream Processing Studio and ESP servers, configure the following environment variable:

```
DFESP_SSLPATH=path-to-OpenSSL-shared-object
```

This setting assumes that you have installed the OpenSSL libraries on all computer systems that run the client and server. When you install SAS Event Stream Processing Encryption and Authentication Overlay, you install OpenSSL. The DFESP_SSLPATH environment variable should be set to the path that contains libssl.so and libcrypto.so. By default, when the Encryption and Authentication Overlay package is installed, the path is as follows:

```
/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/lib
```

or \$DFESP_HOME/lib.

SAS Event Stream Processing includes the internal component SAS Micro Analytic Service. To use the Anaconda Python support in SAS Micro Analytic Service, you need to set additional variables for your version of Python. For instructions, see *SAS Micro Analytic Service: Programming and Administration Guide*, which is available on the [SAS Event Stream Processing product page](#).

Depending on the shell environment that you use, you can also add these export commands to your .bashrc file or .profile file to update the settings automatically. Another option is to create a configuration shell script and copy it to your /etc/profile.d directory.

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 MaxStartups Value

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

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


```
MaxStartups 100
```

If your value for MaxStartups is in the format of three numbers separated by colons, ensure the first number is 100.

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

Set the ulimit Values

Overview

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

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

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

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

```
* - nofile value
```

The asterisk (*) indicates all user accounts.

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

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

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

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

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

Set the Maximum Number of Open File Descriptors 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
```

- 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 `*-nproc.conf` file. For Red Hat Enterprise Linux 6.7 or an equivalent distribution, the file location is `/etc/security/limits.d/90-nproc.conf`. For Red Hat Enterprise Linux 7.1 or an equivalent distribution, the file is `/etc/security/limits.d/20-nproc.conf`

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

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

```
sas - nproc 100000
```

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

```
* - nproc 65536
```

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

```
* - nproc 100000
```

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

Set the Semaphore Values

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 or later or equivalent distributions. If you are using a Linux distribution earlier than Red Hat Enterprise Linux 7.1, you should skip this section.

To change the default time-out values:

- 1 Open the `/etc/systemd/system.conf` file.
- 2 Find the two variables that control time-outs: `DefaultTimeoutStartSec` and `DefaultTimeoutStopSec`.

- 3 If the lines that contain these variables are not already uncommented, uncomment each line by removing the number sign (#).
- 4 Assign both the `DefaultTimeoutStartSec` and `DefaultTimeoutStopSec` variables a value of `1800s`.

```
DefaultTimeoutStartSec=1800s
```

```
DefaultTimeoutStopSec=1800s
```

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

Installation

Overview	56
Modify the Initial Deployment	56
Use a Mirror Repository	56
Edit the Inventory File	56
Overview	56
Single Machine Deployment	57
Multiple Machine Deployment	57
Modify the vars.yml File	62
Set the Deployment Label	62
Set the Pre-deployment Validation Parameters	62
Specify the Installation Type	63
Specify Security Settings	64
Specify the Path to Certificates	64
Define Multiple Invocations	65
(Optional) Specify JRE	65
Set Up Passwordless SSH for CAS	65
Define the CAS User Group	67
Set Up the CAS Admin User	67
Change the CAS Instance Name	68
Add Data Source Information	68
Set Up the CAS Cache Directory	78
Set the CAS Virtual Port	79
Set Up HDFS and Co-location	80
Create the sasv9_deployment.cfg File	81
Configure LDAP Settings for SAS Event Stream Manager	82
SAS Viya and Multi-tenancy	83
SAS Event Stream Processing and Multi-tenancy	83
Overview of Multi-tenancy	83
Considerations for Multi-tenancy and LDAP	83
Enable Multi-tenancy	83
Deploy the Software	85
Assessment Test	85
Command Line	85
Commands	85
Options	85
Run from a Directory Other than the Default	86
Successful Playbook Execution	86
Retry a Failed Deployment	86
Install with SAS 9.4 Software	86

Overview

This chapter describes the first installation of your SAS Viya software with the playbook created by the SAS Orchestration CLI. For information about installing your software on a single machine with the yum utility, see [“Deploying with Yum” on page 171](#).

Modify the Initial Deployment

This chapter describes the initial deployment of your SAS Viya software only. For information about modifying an existing deployment with updated software or adding new software to an existing deployment, [“Managing Your Software” on page 133](#).

Use a Mirror Repository

By default, SAS downloads and installs the latest software available from the software repositories. If your deployment does not have access to the internet or if you must always deploy the same version of software (such as for regulatory reasons), you should create and use a mirror repository for deployment. For details about creating and using mirror repositories, see [“Creating and Using Mirror Repositories” on page 189](#).

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`.

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

Each inventory file consists of two parts:

deployment target definition

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

host group assignment list

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

Here is an example of a host group assignment list:

```
# The CommandLine host group contains command line interfaces for remote interaction with services.
[CommandLine]
deployTarget
deployTarget2
```

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

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

Single Machine Deployment

This section is applicable only if you are performing a single-machine deployment. If you are performing a multi-machine deployment, skip this section and go to [“Specify the Machines in the Deployment” on page 57](#).

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

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

Note: Using an inventory file in any location other than the root directory can seriously affect the deployment of your software. If you do not want to copy a sample file into the root directory, ensure that the inventory file that you do use is in the root directory.

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

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

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

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

For a local machine:

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

For a remote machine:

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

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

Multiple Machine Deployment

Specify the Machines in the Deployment

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:

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 Viya software will be deployed on the machine that is running Ansible:

```
deployTarget ansible_connection=local
```

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

```
sas-service ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ .ssh/id_rsa
sas-programming ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ .ssh/id_rsa
cas-controller-1 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ .ssh/id_rsa
cas-backup-1 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ .ssh/id_rsa
cas-worker-1 ansible_host=host5.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ .ssh/id_rsa
cas-worker-2 ansible_host=host6.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ .ssh/id_rsa
cas-worker-3 ansible_host=host7.example.com ansible_user=user1 ansible_ssh_private_key_file=~/ .ssh/id_rsa
```

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

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

Assign the Target Machines to Host Groups

The second section in the inventory file is used to assign deployment targets to each host group. Under each group, assign machines to the group by using the appropriate alias.

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

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

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

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

```
[AdminServices]
sas-service

[AdvancedAnalytics]
sas-service

[CASServices]
sas-service

[CommandLine]
sas-service

[ComputeServer]
sas-programming

[ComputeServices]
sas-service

[CoreServices]
sas-service

[DataMining]
sas-service

[DataServices]
sas-service

[HomeServices]
sas-service

[ModelServices]
sas-service

[Operations]
sas-programming

[ReportServices]
sas-service

[ReportViewerServices]
sas-service

[ScoringServices]
sas-service

[ThemeServices]
sas-service

[configuratn]
sas-service

[consul]
```

```
sas-service

[httpproxy]
sas-service

[pgpoolc]
sas-service

[programming]
sas-programming

[rabbitmq]
sas-service

[sas-casserver-primary]
cas-controller-1

[sas-casserver-secondary]
cas-backup-1

[sas-casserver-worker]
cas-worker-1
cas-worker-2
cas-worker-3

[sasdatasvrc]
sas-service

[sas-all:children]
AdminServices
AdvancedAnalytics
CASServices
CommandLine
ComputeServer
ComputeServices
CoreServices
DataMining
DataServices
HomeServices
ModelServices
Operations
ReportServices
ReportViewerServices
ScoringServices
ThemeServices
configuratn
consul
httpproxy
pgpoolc
programming
rabbitmq
sas-casserver-primary
sas-casserver-secondary
sas-casserver-worker
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.
- If you are using HDFS co-located with CAS, then [sas-casserver-primary] and [sas-casserver-worker] should be assigned to machines in the Hadoop cluster.
- The first host in the [sas-casserver-primary] list is used by the tenant in a single-tenant deployment or by the provider in a multi-tenant deployment. Only one configuration of CAS (including one primary controller and one secondary controller) per tenant is supported. Therefore, if you change the first host in the list, you are changing the primary CAS controller for a single-tenant deployment or, for multi-tenant deployments, you are changing the primary CAS controller for the provider. Any additional hosts in the [sas-casserver-primary] list are used in a multi-tenant environment. The configuration for those additional hosts (primary controller, secondary controller, or worker) are determined by the *tenant-vars.yml* file.

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

The [sas-casserver-secondary] host group is used only by the tenant in a single-tenant deployment or by the provider in a multi-tenant deployment. Secondary controllers for additional tenants are determined by the *tenant-vars.yml* file. To support failover for predefined libraries, a shared file system must be available for the primary and secondary controllers. For more information about the shared file system, see “[Enable a Shared File System](#)” on page 47.

The [sas-casserver-worker] host group is used only by the tenant in a single-tenant deployment or by the provider in a multi-tenant deployment. Workers for additional tenants are determined by the *tenant-vars.yml* file.

Only one configuration of CAS per tenant is supported. Multiple instances of CAS per tenant are not currently supported.

- [ComputeServer] supports more than one host during initial deployment for both single-tenant and multi-tenant deployments. If multiple hosts are configured, home directories must be located on shared storage devices as configured by the customer. Examples of shared storage are a shared directory, CAS, or other accessible location. Failover is not supported. In the event of a failure, a session will be established on a different host, and the user must log on to re-establish state. In a multi-tenant environment, hosts are shared across all tenants. Adding additional hosts to this host group after the initial deployment is not currently supported.
- [programming] has the same conditions as [ComputeServer].
- [CommandLine] host group should include every host for which an administrator runs either the command-line interface (CLI) or the operations infrastructure. If you are using the https scheme (the default) to access SAS Viya Web applications and plan to onboard tenants, the first host in the [CommandLine] host group must be a host that is present in the [programming], [ComputeServer], or [sas-casserver-*] host group to successfully onboard multiple tenants.

In order for the operations infrastructure to manage audit records:

- Include the machine in the [cas-server-primary] host group for the provider or single tenant.
- Include the machine in the [cas-server-secondary] host group.

Note: For details about auditing, refer to [SAS Viya 3.3 Administration / Auditing](#) .

- If you purchased the optional SAS Event Stream Processing for CAS component, it is automatically installed on all machines where CAS components are installed. However, SAS Event Stream Processing, SAS Event Stream Processing Studio, Streamviewer, and SAS Event Stream Manager must be placed in the corresponding host groups.
- If the machines that you specify for [pgpoolc] or [sasdatasvc] do not have an alias of deployTarget in the deployment target reference, you must open the *sas_viya_playbook/vars.yml* file and replace the

instance of `deployTarget` under `INVOCATION VARIABLES` with the alias that you used in the deployment target reference:

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

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

Note: By default, your deployment includes a single-machine, single-node instance of HA PostgreSQL, used as the SAS Infrastructure Data Server. To deploy HA PostgreSQL with multiple nodes, see [“Creating High Availability PostgreSQL Clusters” on page 159](#).

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. Note that all entries in the `vars.yml` file are case-sensitive.

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"
```

Set the Pre-deployment Validation Parameters

The setting of the `VERIFY_DEPLOYMENT` variable determines the extent of the pre-deployment validation that the playbook performs. If the variable is set to `true` (the default), all of the following actions take place. If the variable is set to `false`, only the Ansible version check is performed. Use the following command to run the validation check without running the entire playbook: `ansible-playbook system-assessment.yml`.

Check the Ansible Version

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

Note: For information about supported Ansible versions, see [“Ansible Controller Requirements” on page 39](#).

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 62](#).

- 2 Verify that a CAS primary controller host is defined.

Note: For information about assigning software to machines, see [“Specify the Machines in the Deployment” on page 57](#).

- 3 Verify that each machine's fully qualified domain name contains less than or equal to 64 characters.
- 4 Verify that each machine in the inventory file can successfully connect to every other machine in the inventory file.

Note: For more information about modifying the inventory file, see [“Specify the Machines in the Deployment” on page 57](#).

- 5 Verify that each machine's fully qualified domain name resolves to the same address for every other machine.
- 6 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 46](#).

- 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 57](#).

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

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

Note: For more information about setting ulimits, see [“Set the ulimit Values” on page 51](#).

Specify the Installation Type

Use the `sas_install_type` variable to specify which interface should be installed.

```
sas_install_type: installation-type
```

Valid values follow:

- **programming** installs only the programming interface, including CAS, SAS Foundation, and SAS Studio
- **a11** installs all the software. **a11** is the default.

If your software order contains only products in the SAS Event Stream Processing family, you can use only **a11**. If your order contains products in the SAS Event Stream Processing family as well as other products, and you choose **programming**, the SAS Event Stream Processing products will not be deployed. For a description of the products in that group, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

Note: When you start the installation, using the `-e "sas_install_type="` command-line option overrides the value that is set in the `vars.yml` file. For more information about the command-line options, see “Options” on page 85.

Specify Security Settings

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

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

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

Specify the Path to Certificates

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

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

To set `HTTPD_CERT_PATH`:

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

Here is an example of the modified variable:

```
HTTPD_CERT_PATH:
```

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

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

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

- 3 Save and close the vars.yml file.

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 159](#).

(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.e16_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/support-for-jre.html>.

Set Up Passwordless SSH for CAS

Manage Passwordless SSH

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

If SAS Cloud Analytic Server (CAS) is deployed on multiple machines, each machine requires passwordless SSH in order to communicate with the others. Passwordless SSH is set up by the playbook by default.

You have three choices for managing passwordless SSH:

- Allow SAS to create a default passwordless SSH with a single user. See [“Accept the Passwordless SSH Default” on page 65](#) for more information about the default process.
- Use your own passwordless SSH. See [“Use Your Own Passwordless SSH” on page 66](#).
- Use the deployment process to create a customized passwordless SSH. Customization can include users other than the default. See [“Create Customized Passwordless SSH” on page 66](#).

Accept the Passwordless SSH Default

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

```

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

```

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

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

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

Use Your Own Passwordless SSH

If you choose to use your own passwordless SSH, you must set the `cas` user to be a user that you have already configured for passwordless SSH. For details, see [“Set Up the CAS Admin User” on page 67](#).

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

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

```

setup_sas_users: false

```

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

Create Customized Passwordless SSH

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

Note: If you add an existing user to the `vars.yml` file, but with an attribute that is different from an attribute that was set elsewhere, the user attribute in the `vars.yml` file takes precedence.

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

Note: Comments have been removed from the following example.

```

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

```

- 2 Edit the fields as follows:

- a Ensure that the `setup_sas_users` variable is set to `true`.
- b Create a list of user accounts and attributes under `sas_users`.

Here are the attributes:

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

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

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

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

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

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

Define the CAS User Group

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

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

- 1 Open the `vars.yml` file.
- 2 In the `casenv_group` field, insert the user group name.
- 3 Save and close the `vars.yml` file.

Set Up the CAS Admin User

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the

SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

If you are performing a full deployment, you should designate an LDAP user to be the CAS admin user. For programming-only deployments, the CAS admin user can be a host account.

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

```
casenv_admin_user: valid-user
```

- 4 Save and close the vars.yml file.

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

Note: This user must have a single set of credentials that are valid for all applicable authentication providers. In a full deployment, dual authentication occurs for logon to CAS Server Monitor and access to CAS from SAS Studio.

Change the CAS Instance Name

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

Changing this variable is not supported in SAS Viya 3.3.

Add Data Source Information

Overview of the Data Sources

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

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

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

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

If you copy and paste from this guide, preserve indents as shown in each example. An indent is equivalent to three spaces.

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

SAS/ACCESS Interface to Amazon Redshift

For the following steps, depending on how you have configured the Amazon Redshift ODBC driver delivered with SAS/ACCESS Interface to Amazon Redshift, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
  1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
  2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
  3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment, remove the number sign (#).
- 4 Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#CAS_SETTINGS:
  1: ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
  2: ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
  3: ODBCHOME=/opt/sas/viya/home/lib64/accessclients
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

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

SAS/ACCESS Interface to DB2

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

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

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment, remove the number sign (#).
- 4 Under CAS_SETTINGS, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

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

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

SAS/ACCESS Interface to Greenplum

For the following steps, depending on how you have configured the Greenplum ODBC driver delivered with SAS/ACCESS Interface to Greenplum, you might need to specify the `odbc.ini` file, the `odbcinst.ini` file, or both files. The following examples include both files.

Follow these steps to edit the `vars.yml` file:

- 1 Open the `vars.yml` file.
- 2 Under `FOUNDATION_CONFIGURATION`, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
  1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
  2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
  3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

For bulk loading, add the following lines.

```
  5: GPHOME_LOADERS=Greenplum-bulk-loader-install-location
  6: GPLOAD_HOME=Greenplum-install-location
  7: GPLOAD_PORT=Greenplum-bulk-load-port
```

- 3 Uncomment the `FOUNDATION_CONFIGURATION` line. To uncomment the line, remove the number sign (`#`).
- 4 Save and close the `vars.yml` file.

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

For the following steps, if you installed your own version of Java, insert its location in the `JAVA_HOME` field. If you are using the JRE that is installed with your SAS software, its default location is `/usr/lib/jvm/jre-1.8.0`. The default should be used unless you edit the playbook to specify a different location for the installation of the JRE.

Follow these steps to edit the `vars.yml` file:

- 1 Open the `vars.yml` file.
- 2 Under `FOUNDATION_CONFIGURATION`, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
  1: JAVA_HOME=location-of-your-Java-8-JRE
  2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

If you are using MapR, add the following line.

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

- 3 Uncomment the `FOUNDATION_CONFIGURATION` line. To uncomment the line, remove the number sign (`#`).
- 4 Under `CAS_SETTINGS`, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#CAS_SETTINGS:
  1: JAVA_HOME=location-of-your-Java-8-JRE
  2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/lib/amd64/server
```

If you are using MapR, add the following line.

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

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

SAS/ACCESS Interface to HAWQ

For the following steps, depending on how you have configured the Greenplum ODBC driver delivered with SAS/ACCESS Interface to HAWQ, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
 1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
 2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
 3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
 4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 4 Save and close the vars.yml file.

SAS/ACCESS Interface to Impala

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

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

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

```
#FOUNDATION_CONFIGURATION:
 1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
 2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
 3: CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
 4: EASYSOFT_UNICODE=YES
 5: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
/opt/cloudera/impalaodbc/lib/64
```

Note: The EASYSOFT_UNICODE variable should be added only to set the encoding for the SAS client to UTF-8.

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

- Under `CAS_SETTINGS`, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

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

```
#CAS_SETTINGS:
  1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
  2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
  3: CLUSTERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
/opt/cloudera/impalaodbc/lib/64
```

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

SAS/ACCESS Interface to Microsoft SQL Server

For the following steps, depending on how you have configured the Microsoft SQL Server ODBC driver delivered with SAS/ACCESS Interface to Microsoft SQL Server, you might need to specify the `odbc.ini` file, the `odbcinst.ini` file, or both files. The following examples include both files.

Follow these steps to edit the `vars.yml` file:

- Open the `vars.yml` file.
- Under `FOUNDATION_CONFIGURATION`, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
  1: ODBCINI=/opt/sas/spre/home/lib64/accessclients/odbc.ini
  2: ODBCINST=/opt/sas/spre/home/lib64/accessclients/odbcinst.ini
  3: ODBCHOME=/opt/sas/spre/home/lib64/accessclients
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

- Uncomment the `FOUNDATION_CONFIGURATION` line. To uncomment the line, remove the number sign (#).
- Under `CAS_SETTINGS`, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#CAS_SETTINGS:
  1: ODBCINI=/opt/sas/viya/home/lib64/accessclients/odbc.ini
  2: ODBCINST=/opt/sas/viya/home/lib64/accessclients/odbcinst.ini
  3: ODBCHOME=/opt/sas/viya/home/lib64/accessclients
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

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

SAS/ACCESS Interface to MySQL

Follow these steps to edit the `vars.yml` file:

- Open the `vars.yml` file.
- Under `FOUNDATION_CONFIGURATION`, add the following line as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
  1: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-MySQL-client
```

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 4 Save and close the vars.yml file.

SAS/ACCESS Interface to Netezza

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

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
  1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
  2: ODBCINST=location-of-your-odbc.ini-file-including-file-name
  3: NZ_ODBC_INI_PATH=path-to-the-Netezza-configuration-files
  4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Netezza-client
```

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 4 Save and close the vars.yml file.

SAS/ACCESS Interface to ODBC

Follow these steps to edit the vars.yml file:

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

For DataDirect:

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

For iODBC:

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

For unixODBC:

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

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

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

For DataDirect:

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

For iODBC:

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

For unixODBC:

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

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

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

SAS/ACCESS Interface to Oracle

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

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

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

```
#CAS_SETTINGS:
 1: ORACLE_HOME=Oracle-home-directory
```



```
2: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

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

SAS/ACCESS Interface to PostgreSQL

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

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

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

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

```
#CAS_SETTINGS:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: PGCLIENTENCODING=UTF-8
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-PostgreSQL-client
```

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

SAS/ACCESS Interface to SAP HANA

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

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client
```

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

```
#CAS_SETTINGS:
1: ODBCINI=location-of-your-odbc.ini-file-including-file-name
```

```
2: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-your-SAP-HANA-client
```

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

SAS/ACCESS Interface to SAP R/3

Follow these steps to edit the vars.yml file:

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes:

```
#FOUNDATION_CONFIGURATION:
1: RFC_INI=path-to-the-R/3-ini-file
2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-R/3-client
```

- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 4 Save and close the vars.yml file.

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

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

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

- 3 Open the vars.yml file.
- 4 Under FOUNDATION_CONFIGURATION, add the following lines as shown, including the indentions, spaces, and numerical prefixes.

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

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

Here is an example of the TTU Default LD_LIBRARY_PATH:

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

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

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

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

Here is an example of the TTU Default LD_LIBRARY_PATH:

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

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

Specify Multiple Data Connectors

Note: When adding multiple SAS/ACCESS products, make sure that the lines that you add are in the same block and are numbered consecutively from first to last. Even though the lines for the SAS/ACCESS products can be mixed in the block, ensure that the lines for each product remain in the order that was provided in the preceding sections.

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

Here is an example of a block for both the DataDirect version of SAS/ACCESS Interface to ODBC and for SAS/ACCESS Interface to Oracle.

- 1 Open the vars.yml file.
- 2 Under FOUNDATION_CONFIGURATION, add the appropriate lines as shown, including the indentions, spaces, and numerical prefixes:


```
#FOUNDATION_CONFIGURATION:
  1: ODBCHOME=ODBC-home-directory
  2: ODBCINI=location-of-your-odbc.ini-file-including-file-name
  3: ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
  4: ORACLE_HOME=Oracle-home-directory
  5: TWO_TASK=ORACLE_SID
  6: ORAENV_ASK=NO
  7: SASORA=V9
  8: PATH=$PATH:$ORACLE_HOME:bin
  9: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
 10: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```
- 3 Uncomment the FOUNDATION_CONFIGURATION line. To uncomment the line, remove the number sign (#).
- 4 Under CAS_SETTINGS, add the appropriate lines as shown, including the indentions, spaces, and numerical prefixes:

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

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

Set Up the CAS Cache Directory

Change the CAS Cache Directory

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

SAS Cloud Analytics Services (CAS) is the in-memory analytic server for SAS Viya. As a memory efficiency, CAS organizes in-memory data in blocks and memory maps the blocks. The blocks are stored as temporary files in directories on the host. The controller also uses the cache directory temporarily to store uploaded files.

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

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

SAS supports only local file systems for the CAS cache such as EXT4 and XFS. It does not support network file systems such as GPFS.

To change the CAS cache:

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

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

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

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

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

The directory structure must be identical on each controller and worker, but the controller host does not require the same volume of space as each worker. To conserve disk space on the controller, the directories that are specified in `CAS_DISK_CACHE` can occupy as little as one partition or one logical volume. Likewise, the directories can be specified as file system links to a single directory. In all cases, make sure that the directories do not use the same disk or the same volume as the root file system.

Tune the CAS Cache Directory

Here are some tuning tips for the CAS cache directory:

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

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

Table 4.1 Sample Configurations

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

Set the CAS Virtual Port

To ensure that the CAS Server Monitor is accessible:

- 1 Open the vars.yml file.

- 2 In the CAS_CONFIGURATION block, add the following line for CAS_VIRTUAL_PORT after the env line.

```
CAS_CONFIGURATION:
  env:
    CAS_VIRTUAL_PORT: 443
```

- 3 Save and close the vars.yml file.

Set Up HDFS and Co-location

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

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

```
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
  cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    #colocation: 'none'
```

Note: For descriptions of HDFS and co-location, see [“CAS Server Co-located with Hadoop” on page 13](#).

If you include a machine in the host group [sas-casserver-worker] in the inventory file, the playbook assumes that you are performing a massively parallel processing (MPP) deployment. This means that your CAS deployment includes a controller and at least one worker. When the playbook runs, it removes the number sign (#) from the colocation variable and adds a mode variable that is set to `mpp`. You must continue to edit the CAS_CONFIGURATION section as follows:

- 1 Open the vars.yml file.
- 2 If you are deploying some or all of your CAS machines on the same machines where HDFS is running, revise the variables' values as follows:

```
CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    HADOOP_NAMENODE: namenode-host-name
    HADOOP_HOME: location-of-your-Hadoop-home-directory-on-the-HDFS-server
  cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    colocation: 'hdfs'
    mode: 'mpp'
```

Note: HADOOP_NAMENODE can be up to two host names, the primary and standby namenodes, separated by a colon. For example:

```
HADOOP_NAMENODE=namenode1:namenode2
```

Note: If you intend to use remote HDFS, ensure that the path used for HADOOP_HOME includes `/lib/hadoop`. For example: `/opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib/hadoop`.

- 3 If you are deploying CAS on machines completely separate from the HDFS machines, revise the variables' values as follows:

```

CAS_CONFIGURATION:
  env:
    #CAS_DISK_CACHE: /tmp
    HADOOP_NAMENODE: namenode-host-name
    HADOOP_HOME: location-of-your-Hadoop-home-directory-on-the-HDFS-server
    CAS_ENABLE_REMOTE_SAVE: 1
    CAS_REMOTE_HADOOP_PATH: 'SASHDAT-executables-directory-on-the-HDFS-server'
  cfg:
    #gcport: 0
    #httpport: 8777
    #port: 5570
    colocation: 'hdfs'
    mode: 'mpp'

```

By default, the deployment will search HADOOP_HOME and `/opt/sas/HDATHome/bin` for the HDAT plug-ins. You should supply a value for CAS_REMOTE_HADOOP_PATH only if you are using a location for the HDAT plug-ins other than HADOOP_HOME or `/opt/sas/HDATHome/bin`.

Note: HADOOP_NAMENODE can be up to two host names, the primary and standby namenodes, separated by a colon. For example:

```
HADOOP_NAMENODE=namenode1:namenode2
```

Note: If you intend to use remote HDFS, ensure that the path used for HADOOP_HOME includes `/lib/hadoop`. For example: `/opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib/hadoop`.

4 Save and close the vars.yml file.

Note: For more information about CAS environment variables, see [SAS Viya Administration: SAS Cloud Analytic Services](#).

Create the sasv9_deployment.cfg File

Overview

SAS Viya uses the `sasv9_deployment.cfg` file to set system options. To create that file at deployment, add values to the `vars.yml` file before running the playbook.

sasv9_deployment.cfg

1 Open the `vars.yml` file if it is not already open.

2 Go to the `SASV9_CONFIGURATION` block of variables.

```

# Creates a workspaceserver sasv9_deployment.cfg file
#SASV9_CONFIGURATION:
#1: '/* Comment about OPTION */'
#2: 'OPTION value'

```

3 Uncomment the second and fourth lines by removing the number sign (#) from the beginning of the line. Also, to add comments for the new system options, uncomment the third line

```

# Creates a workspaceserver sasv9_deployment.cfg file
SASV9_CONFIGURATION:
1: '/* Comment about OPTION */'
2: 'OPTION value'

```

4 Add the system options and comments, as appropriate, and ensure that the line numbers are incremented by one for each line that you add to the `SASV9_CONFIGURATION` block. Here is an example:

```
# Creates a workspace server sasv9_deployment.cfg file
SASV9_CONFIGURATION:
  1: '/* aligning output with page boundary */'
  2: 'ALIGNSASIOFILES'
  3: 'BYLINE'
  4: '/* setting the paper size */'
  5: 'PAPERSIZE=LETTER'
```

Note: For the list of available system options, see [SAS 9.4 and SAS Viya 3.3 Programming Documentation / System Options](#).

Use the indentation that is already in the vars.yml file. Ensure that the value in each line is enclosed in single quotation marks. Comments must include the comments set of characters, /* and */, within the quotation marks.

- 5 When you have entered all the options that you want to use, save and close the vars.yml file.

Modify the sasv9_deployment.cfg File after Deployment

After your software has been deployed, you might find it necessary to change the configuration settings that were created by the steps in this section. You can modify the sasv9_deployment.cfg file directly, but you will lose any customizations if the playbook is run again.

To prevent the loss of your customizations, SAS recommends that you create a separate file (referred to as a “usermods” file) to be consumed by subsequent deployments. For the changes to the sasv9_deployment.cfg file, create a usermods file named sasv9_usermods.cfg by copying the sasenv_deployment file and renaming it. Make your changes to the sasv9_usermods.cfg file. It will override the sasv9_deployment.cfg file in subsequent deployments.

Configure LDAP Settings for SAS Event Stream Manager

If your order contains only SAS Event Stream Processing and SAS Event Stream Manager, take some steps to configure SAS Logon Manager before starting the deployment. The playbook can configure the LDAP server to enable SAS Logon Manager if you supply the required parameters.

- 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 Use your preferred text editor to open sitedefault.yml.
- 3 Add values that are valid for your site, and save the file.

When you run your Ansible playbook using the site.yml option, as instructed in “[Deploy the Software](#)” on page 85, the updated sitedefault.yml file is used.

SAS Viya and Multi-tenancy

SAS Event Stream Processing and Multi-tenancy

If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. If you later want to add on multi-tenant SAS Viya software, you cannot and must deploy that software on separate machines. If you deploy SAS Viya with multi-tenancy now and later want to add products from the SAS Event Stream Processing family, you cannot and must deploy them on separate machines.

Note: For a description of the products in the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

Overview of Multi-tenancy

The term *multi-tenant* identifies software that allows several groups of users to interact with a single instance of the software. Each group is called a *tenant*. This section describes how to ensure that your SAS Viya software deployment supports multi-tenancy.

If you anticipate having multiple tenants, even if you have only one tenant initially, you must enable multi-tenancy. Changing from a single-tenant deployment to a multi-tenant deployment requires that you re-deploy your software and make the modifications described in this section. If you do not want your SAS Viya deployment to support multi-tenancy, skip this section.

Considerations for Multi-tenancy and LDAP

SAS recommends that you enter the LDAP information about your multi-tenant deployment with SAS Environment Manager after installing your software (see [“Configure the Connection to Your Identity Provider” on page 91](#)). If you use SAS Environment Manager, you must remove the `sas.identities.providers.ldap.connection` block from your `sitedefault.yml` file. You can remove the block while performing the steps in the next section. See step 4.

Instead of using SAS Environment Manager, administrators with experience deploying SAS Viya might prefer to complete the `sas.identities.providers.ldap.connection` block of the `sitedefault.yml` file. SAS recommends this approach only to automate a deployment. For more information about the variables in the `sas.identities.providers.ldap.connection` block, see [“sas.identities.providers.ldap.connection”](#) in *SAS Viya Administration: Configuration Properties*.

Enable Multi-tenancy

To enable a multi-tenant deployment:

- 1 To prevent tenants from being able to read SAS data sets that they do not own, add a value to the `vars.yml` file for multi-tenancy:
 - a Open `vars.yml`.
 - b In the `STUDIO_CONFIGURATION` block, under the `init` section, add the following variable and value.

```
STUDIO_CONFIGURATION:
  init:
    #sasstudio.appserver.port_comment: '# Port that Studio is listening on'
    #sasstudio.appserver.port: 7080
    #sasstudio.appserver.https.port: 7443
```

```
#webdms.workspaceServer.hostName: localhost
#webdms.workspaceServer.port: 8591
webdms.showSystemRoot: false
```

c Save and close the vars.yml file.

- 2 If you have not already done so, make a copy of sitedefault_sample.yml and name the copied file sitedefault.yml. The file sitedefault_sample.yml is in the directory where you uncompressed your playbook. The recommended location is `/sas/install/sas_viya_playbook/roles/consul/files/`.
- 3 Edit sitedefault.yml.

Note: It is critical that you maintain the indentation in sitedefault.yml. Each level in the file is indented four spaces. The examples in this step include the indentation.

- a At the beginning of the application block and at the same level as `sas.identities.providers.ldap.connection`, add the following line:

```
config:
  application:
    sas.multi.tenancy.enabled: true
```

- b On the next two lines, list the internal host names that are used to access the provider zone or that are used in a subdomain to access other zones.

Here is an example:

```
config:
  application:
    sas.multi.tenancy.enabled: true
  zones:
    internal.hostnames: comma-separated-list-of-internal-host-names
```

The comma-separated list of internal host names should specify any hosts that will be used to access the provider and any domain from which tenant subdomains will be potentially built. A host name from this list with a prepended tenant name will be used as the URL to reach each tenant after each one has been onboarded.

- 4 (Optional) If you decide to use SAS Environment Manager to set up your LDAP identities, locate the `sas.identities.providers.ldap.connection` block, and delete it. Otherwise, skip this step. For more information, see [“Considerations for Multi-tenancy and LDAP” on page 83](#).
- 5 Add wildcard versions of the host names to your DNS using the method that your administrator recommends. For example, if you added `hostname1.company.com` and `hostname2.company.com` to the `zones.internal.hostnames` variable in your `sitedefault.yml` file, you would add `*.hostname1.company.com` and `*.hostname2.company.com` to your DNS that point to the host name or IP address of the HTTP server of your SAS installation. The DNS record type is irrelevant. Therefore, you can use A record, AAA record, or CNAME record as long as all tenant-specific subdomains will resolve to your SAS installation.

After you run the playbook and perform post-installation tasks, your deployment will have a single tenant, which is referred to as the *provider*. To administer multi-tenancy, including adding more tenants, see [Multi-tenancy: Initial Tasks](#).

Deploy the Software

Assessment Test

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

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

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

Command Line

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

```
command [ option ]
```

The command that you select is determined by your deployment and password requirements. See [“Commands” on page 85](#).

You can select an option to specify the interface to the software to be installed in your environment. You can also specify the level of installation or configuration to perform. See [“Options” on page 85](#).

Commands

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

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

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

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

Options

The following options can be specified in the command line:

-e "sas_install_type=programming"

deploys only the programming interface, including CAS, SAS Foundation, and SAS Studio.

--tags install

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

--tags config

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

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

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

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

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

Note: Using the `-e "sas_install_type=` commands overwrites the values that is set in the `vars.yml` file. For more information about setting the installation type, see [“Specify the Installation Type” on page 63](#).

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 SAS Viya configuration options. Refer to the Ansible documentation to find the appropriate `ansible.cfg` file and add those options.

Successful Playbook Execution

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

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

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

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 [“Commands” on page 85](#) and any appropriate options.

Failures may occur if there are port conflicts. See [“Install with SAS 9.4 Software” on page 86](#) for a potential source of port conflicts.

Install with SAS 9.4 Software

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

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

- For SAS/CONNECT, the `sasenv_connect_port` variable
- For SAS Studio, the `sasstudio.appserver.port` in the `STUDIO_CONFIGURATION` block
- For the object spawner, the `sasPort` in the `SPAWNER_CONFIGURATION` block

Note: If you change the port value for the object spawner, you must also change the value of `webdms.workspaceServer.port` in the `STUDIO_CONFIGURATION` block to match the port number that you specified in the `SPAWNER_CONFIGURATION` block.

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

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

To change the value:

- 1 Remove the number sign from the beginning of the variable for the port number that you want to change.
- 2 Change the port value to the one that you want to use.
- 3 Save and close the `vars.yml` file.

Here is the earlier example revised in this way:

```
SPAWNER_CONFIGURATION:  
  sasPort: 8592
```

Deployment Logs

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

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

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


Post-Installation Tasks

Configure Security	89
Set the Password for the CAS Administrator or Another Administrative Account	89
Change the Administrative User Password for SAS Message Broker	90
Configure Your Environment with SAS Environment Manager	91
Configure Machine and Application Settings	94
Configure a Symbolic Link to a Storage Platform	94
Verify That Licenses Are Applied	95
Complete SAS Event Stream Processing Setup	95
Complete SAS Event Stream Manager Setup	98
Configure High Availability in SAS Studio	100
Configure the Default Backup Schedule	102
Configure Data Access	102
Configure SAS/ACCESS Interface to Amazon Redshift	102
Configure SAS/ACCESS Interface to DB2	103
Configure SAS/ACCESS Interface to Greenplum	104
Configure SAS/ACCESS Interface to HAWQ	104
Configure SAS Data Connector to Hadoop	105
Configure SAS/ACCESS Interface to Impala	106
Configure SAS/ACCESS Interface to Microsoft SQL Server	107
Configure SAS/ACCESS Interface to MySQL	108
Configure SAS/ACCESS Interface to Netezza	108
Configure SAS/ACCESS Interface to ODBC	108
Configure SAS/ACCESS Interface to Oracle	110
Configure SAS Data Connector to PostgreSQL	110
Configure SAS/ACCESS Interface to SAP HANA	111
Configure SAS/ACCESS Interface to SAP R/3	112
Configure SAS/ACCESS Interface to Teradata	112
Configure Database Connectivity for SAS Event Stream Processing	113
Configure Data Quality	113
Configure the Quality Knowledge Base	113

Configure Security

Set the Password for the CAS Administrator or Another Administrative Account

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the

SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

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

```
sudo passwd cas
```

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

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

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

Change the Administrative User Password for SAS Message Broker

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

Note: The tasks in this section are applicable if you deployed all of your software. If you deployed the programming interface only, skip this section.

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

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

- 3 Change to this directory:

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

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

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

-t account-type

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

-u user-ID

identifies the client user ID for SAS Message Broker.

--promptpw

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

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

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

- 5 Restart all SAS Viya services. Restarting the SAS Viya services activates the changes to the credentials for SAS Message Broker. For more information, refer to [SAS® Viya™ 3.3 Administration Guide: General Servers and Services](#).

Configure Your Environment with SAS Environment Manager

If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

The tasks in this section are applicable if you deployed all of your software. If you deployed the programming interface only, skip this section.

Sign In as the sasboot User

Your SAS environment is deployed with an initial administrator account that is named `sasboot`. The password for this account has expired by default, so you must reset the password before you can sign in. Follow these steps:

- 1 Locate the most recent log for the SAS Logon service in `/var/log/sas/viya/saslogon/default`.

Note: SAS Logon is installed on one or more machines to which you are assigned in the CoreServices host group in the inventory file. For information about the inventory file, see [“Edit the Inventory File” on page 56](#).

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

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

A message similar to the following is displayed

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

- 3 Sign in from a URL with this format:

```
https://reverse-proxy-server/SASLogon/reset_password?code=xxxxxxx
```

Note: Use the host name from the machine that you assigned to the [httpproxy] host group in the inventory file. For information about the inventory file, see [“Edit the Inventory File” on page 56](#).

Make a note of this SAS Environment Manager URL to share with any other users of your SAS Viya software, as described in [“Share Important Deployment Information with the Administrators” on page 131](#).

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

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

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

Then go to the log and obtain the new URL. The URL expires 24 hours after the SAS Logon service restarts. For security purposes, the URL that is specified in a browser or in a text editor also expires, even if the password is not reset.




After you reset the password, SAS Environment Manager automatically opens in your browser.

- 5 Click the **Yes** button for all of the assumable groups so that you have the permissions to perform subsequent tasks.

Configure the Connection to Your Identity Provider

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

Note: Only LDAP-based identity providers are supported. These instructions assume that you have basic familiarity with LDAP administration. For details about properties, “[sas.identities.providers.ldap](#)” in *SAS Viya Administration: Configuration Properties*.

- 1 Select the  from the side menu .
- 2 On the Environment page, select **Basic Services** from the list, and then select the **Identities service** from the list of services.
- 3 In the **sas.identities.providers.ldap.user** section, click . In the New Configuration window, follow these steps:


- a Specify a value for the following required field: **baseDN**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

Note: When using the LDAP protocol, passwords are transmitted over the network in plaintext. To secure the deployment, SAS recommends that you configure LDAPS. For details, refer to [Configure SAS Viya to Connect to LDAPS Provider](#) in *SAS® Viya 3.3 Administration: Data in Motion*.

For each property that represents a user-level field in SAS, specify a corresponding property in the LDAP provider software.

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

- b Click **Save**.


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

- a Specify a value for the following required field: **baseDN**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

For each property that represents a group-level field in SAS, specify a corresponding property in the LDAP provider software.

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

- b Click **Save**.

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

- a Specify values for the following required fields: **host**, **password**, **port**, **url**, and **userDN**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

- b Click **Save**.







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

On the Users page, select **Users** from the list in the toolbar. Your users should appear after a few minutes. It is not necessary to restart any servers or services. Then select **Groups** from the list to display your groups.

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

Configure the Connection to the Mail Service


After installing a new SAS Viya deployment, you must configure the connection to your mail service. Complete these steps while you are signed in as the sasboot user.

- 1 Select the  from the side menu .
- 2 On the Environment page, select **Basic Services** from the list, and then select **Mail service** from the list of services.
- 3 In the **sas.mail** section, click . In the Edit Configuration window, follow these steps:
 - a Specify a value for the following required fields: **host** and **port**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.
 - b Click **Save**.
- 4 (Optional) To enable the health check for the mail service, perform the following steps.
 - a Select  from the side menu .
 - b On the Environment page, select **Basic Services** from the list, and then select **Mail service** from the list of services.
 - c In the **management.health.mail** section, click .
 - d Turn the **enabled** toggle to **on**.
 - e Click **Save**.

When this toggle is set and after the mail service is restarted, you can view the status of the mail service from the dashboard page of SAS Environment Manager. If the mail host is not configured or is configured incorrectly, or if it cannot connect to the SMTP mail server, the mail service will indicate that it is in a failed state.

Set Up Administrative Users

While you are signed on to SAS Environment Manager as the sasboot user, set up at least one SAS Administrator user, as follows:

- 1 On the Users page in SAS Environment Manager, select **Custom Groups** from the list in the toolbar.
- 2 In the left pane, click **SAS Administrators**.
- 3 In the **Members** section of the right pane, click , and add one or more members to the group (including your own account, if applicable).
- 4 Sign out from SAS Environment Manager so that you are no longer signed in as the sasboot user.
- 5 If you added your own account to the SAS Administrators group, you can sign on again to SAS Environment Manager using that account.

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

```
https://reverse-proxy-server/SASEnvironmentManager
```

TIP Since SAS Administrators is an assumable group, the following prompt is displayed: Do you want to opt in to all of your assumable groups?. Select **Yes** if you want the extra permissions that are associated with the SAS Administrators group. The selection remains in effect until you sign out.

Sign In Using LDAP Credentials





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

`https://reverse-proxy-server/SASEnvironmentManager`

Sign in as one of the SAS Administrators that you set up in [“Set Up Administrative Users” on page 93](#).

Disable the Password Reset Feature and Reset the sasboot Password

When you are finished setting up LDAP and the initial administrative users, you should reset the password for the sasboot user. For additional security, you can then disable the password reset feature. This prevents password reset links from being written to the log each time the SASLogon service is restarted.

- 1 Sign in to SAS Environment Manager as an administrative user and select  from the side menu .
- 2 On the Environment page, select **Definitions** from the drop-down list.
- 3 In the left pane, select **sas.logon.initial**. Then select  at the top of the right pane. If a definition already exists, you can select  to edit the existing definition.
- 4 In the New sas.logon.initial Configuration window or the Edit sas.logon.initial Configuration window, set **reset.enabled** to **off**.
- 5 Click **Save**.
- 6 Restart the SASLogon service. For more information, see [General Servers and Services: Operate](#) in *SAS Viya Administration: General Servers and Services*.

Note: After you disable this feature, you can still change the sasboot password if the existing password is known. Enter the URL for SAS Viya with the path `/SASLogon/change_password`. If you are already signed in as another user, first sign out and then sign back in as sasboot using the current password. You can then complete the steps to change the password.

Configure SAS Viya to Connect to LDAPS Provider

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

Configure Machine and Application Settings

Configure a Symbolic Link to a Storage Platform

Note: This section applies only to the following products:

- SAS Visual Forecasting
- SAS Visual Data Mining and Machine Learning in SAS Model Studio

■ SAS Visual Text Analytics

To ensure proper performance of your solutions, create a symbolic link for the `/opt/sas/viya/config/data/cas` directory to a high-performance storage platform. Examples of high-performance storage platforms include SAN, NVMe, and multiple drive disk arrays.

Verify That Licenses Are Applied

Note: Licenses for SAS Event Stream Processing components are applied automatically by the deployment process.

During installation, a license is applied to both the CAS in-memory compute engine and the SAS Foundation compute engine. To ensure proper operation of the engines, you should verify that the licenses were applied properly.

For details, see [Licensing: How To](#) in *SAS® Viya 3.3 Administration: Licensing*.

If the licenses were not applied, use the instructions to apply the licenses.

Complete SAS Event Stream Processing Setup

If your order included SAS Event Stream Processing, take a few steps to complete the deployment. You must start the Metering Server and also start the ESP server. You also have the option to generate and import certificates to support encryption for the ESP server. Otherwise, you can skip this section.

Enable Metering for ESP Servers

If your order included SAS Event Stream Processing, you must take additional steps to enable the product license. The playbook applies the product license on each machine where you have deployed SAS Event Stream Processing. However, you must set up and run at least one metering server to track the number of incoming events and to maintain event counts on your ESP servers.

The metering server aggregates counts that are based on the license, the source window, and the hour of day. It stores aggregated results so that a client can query and track the total volume of messages that are processed. Enabling the metering server ensures that your ESP server is in compliance with the terms of its license. Event metering is not required on development servers because they do not contribute to the event volume that is assigned to a license.

For more information about enabling metering, see [Using the Metering Server](#) in the SAS Event Stream Processing user documentation.

Log On to SAS Event Stream Processing Studio

SAS Event Stream Processing Studio can run as an independent application, or it can be integrated into SAS Viya authentication and used with SAS Logon Manager. The settings that you selected for SAS Event Stream Processing when the playbook was run determine the authentication method to use for logon.

Note: If you edit the inventory file correctly, SAS Event Stream Processing Studio can be installed on a separate machine while still using integrated authentication.

- 1 SAS Event Stream Processing Studio requires Java 1.8. If Java 1.8 is not the default version of Java on your system, update the following script to set the `SAS_JAVA_HOME` environment variable:

```
/opt/sas/viya/config/etc/sysconfig/sas-javaesnt1/sas-java
```

Here is an example of how to set the variable:

```
SAS_JAVA_HOME=/usr/java/jdk1.8.0_101/jre
```

Or supply the location of the JDK, if applicable. For example:

```
SAS_JAVA_HOME=/usr/java/jdk1.8.0_101
```

Note: Do not include the `/bin/java` portion of the path for the definition of `SAS_JAVA_HOME`.

- 2 (Optional) If you plan to import models from SAS Model Manager, increase the default memory allocation for the Java Virtual Machine (JVM). Configuring the JVM is a post-deployment task. For more information, see [Configuration Properties: Java Virtual Machine \(JVM\)](#) in the SAS Viya Administration documentation.
- 3 Verify that you have set the required environment variables. For more information, see [“Set Environment Variables for SAS Event Stream Processing” on page 50](#).

- 4 SAS Event Stream Processing Studio should be running when the playbook completes. Check the status of the `esvm` process. Run the following command on Red Hat Enterprise Linux 6.x:

```
sudo service sas-viya-esvm-default status
```

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl status sas-viya-esvm-default
```

- 5 If the `esvm` service is reported to be down, run the following command on Red Hat Enterprise Linux 6.x to start it:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl start sas-viya-esvm-default
```

- 6 Access SAS Event Stream Processing Studio using a web browser that is running on Windows or Linux. The URL format depends on your deployment topology:

```
scheme://reverse-proxy-server/SASEventStreamProcessingStudio
```

```
scheme://esp-studio-hostname:port/SASEventStreamProcessingStudio
```

In a programming-only deployment, the scheme is `http`. In a full deployment, the scheme is `https`.

For *reverse-proxy-server*, specify the hostname of the machine that you assigned to the `[httpproxy]` host group in the inventory file. Use this format when SAS Event Stream Processing Studio uses the same authentication method and proxy server as other SAS Viya products.

For *esp-studio-hostname* and *port*, specify values that are appropriate for your deployment. Use this format when SAS Event Stream Processing Studio is an independent installation that does not use the same authentication method and proxy server as other SAS Viya products.

The default port is 8080. For information about changing the default port, see [“\(Optional\) Change the SAS Event Stream Processing Studio Port” on page 97](#).

- 7 Before you can open or create a model in SAS Event Stream Processing Studio, you must start the ESP server. Change directories to the following location:

```
cd /opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/bin
```

- 8 Run the following command:

```
dfesp_xml_server -pubsub n -http port &
```

The `-pubsub` argument specifies a port for publish and subscribe actions. Replace *n* with the appropriate port number.

The `-http` argument Specifies the port for the HTTP REST API. The value of *port* cannot exceed 65535.

The ampersand (&) enables additional commands to be entered in the same window that started the server.

Note: If you have a project that is predefined, use the `-model url` argument and supply the URL to the XML model. Specify the full path (`file://path`).

For more information about the ESP server, see [SAS Event Stream Processing: Using the ESP Server](#).

(Optional) Change the SAS Event Stream Processing Studio Port

If you have installed SAS Event Stream Processing Studio on a separate machine, and if you have not configured it to use integrated SAS Viya authentication, you can change its port settings. The default port, 8080, is appropriate for most environments. But if you configured the playbook to co-locate SAS Event Stream Processing Studio (the [espStudio] host group) and SAS Configuration Server (the [consul] host group), you cannot change the default port.

- 1 Use your preferred text editor to open and edit the following file:

```
sudo vi /opt/sas/viya/home/bin/sas-espvm
```

- 2 Locate the following line in the file:

```
export java_option_server_port="-Dserver.port=8080"
```

- 3 Change the default port, 8080, to the appropriate port.
- 4 Save and close the file.
- 5 Restart the espvm service by running the following commands on Red Hat Enterprise Linux 6.x:

```
sudo service sas-viya-espvm-default stop
sudo service sas-viya-espvm-default start
```

Run the following commands on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-espvm-default
sudo systemctl start sas-viya-espvm-default
```

(Optional) Enable Encryption and Authentication for SAS Event Stream Processing

SAS Event Stream Processing provides optional encryption and authentication features. The required OpenSSL encryption libraries are installed automatically when you install SAS Event Stream Processing. You can then enable encryption with OpenSSL on TCP/IP connections within an event stream processing engine. You can also configure ESP servers to require client authentication for SAS TCP/IP clients. Authentication and encryption apply to the following ESP server APIs:

- The ESP Server (XML Server) HTTPS API
 - Connections that are created by the XML Client (dfesp_xml_client) to communicate with an ESP server using the HTTPS protocol
 - Connections that are created by the Streamviewer component (streamviewer.html) to communicate with the ESP server using the HTTPS protocol
- C or Java Publish/Subscribe API
 - Connections that are created by a client that uses the C or Java Publish/Subscribe API to communicate with an ESP server
 - Connections that are created by an adapter to communicate with an ESP server

If you enable authentication for an ESP server, you must then provide authentication tokens or credentials in Streamviewer. You can copy and paste the token directly into an appropriate dialog box in Streamviewer. Alternatively, you can specify a URL that supplies the token. Authentication tokens and credentials are cached for the duration of a Streamviewer session.

For more information about enabling security for an ESP server or for Streamviewer, see [SAS Event Stream Processing: Security](#).

(Optional) Enable Encryption for SAS Event Stream Processing Studio

Secure Sockets Layer (SSL) encryption can be applied to the connections that are made between SAS Event Stream Processing Studio and SAS ESP servers. To enable SSL for SAS Event Stream Processing Studio and the clients that access it, you must generate a pair of certificates, copy them to the required locations, and add the client certificate to your browser and to the Java keystore.

- 1 Verify that the OpenSSL libraries exist on all machines where SAS Event Stream Processing components or clients will run.

Locate the `libcrypto.so` and `libssl.so` files. They are installed by default in `/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0/ssl/lib`. Obtain them from OpenSSL if required.

- 2 Verify that the `DFESP_SSLPATH` environment variable specifies the pathname for the OpenSSL shared libraries.
- 3 Obtain SSL certificates for the machine where SAS Event Stream Processing Studio is installed and for the clients that will access the user interface. Use OpenSSL or your preferred method to generate site-signed or third-party-signed certificates.
- 4 On the machines from which end users will access SAS Event Stream Processing Studio, import the client certificate to the certificates store of your preferred web browser.
- 5 On the machine where SAS Event Stream Processing Studio is running, import the client certificate to the Java keystore by running the following command:

```
$JAVA_HOME/jre/bin/keytool -importcert -keystore keystore-location -file path-to-file
-storepass password -noprompt -alias alias
```

Here is an example:

```
$JAVA_HOME/jre/bin/keytool -importcert -keystore $JAVA_HOME/jre/lib/security/cacerts
-file $DFESP_HOME/etc/ca.pem -storepass P4ssw0rd -noprompt -alias myalias
```

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

- 6 Restart the SAS Event Stream Processing Studio service. Run the following command, as appropriate:

For Red Hat Enterprise Linux 6.7:

```
sudo service sas-viya-espvm-default stop
sudo service sas-viya-espvm-default start
```

For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl stop sas-viya-espvm-default
sudo systemctl start sas-viya-espvm-default
```

Complete SAS Event Stream Manager Setup

If your order included SAS Event Stream Manager, take a few steps after the installation has completed to prepare the environment. Otherwise, you can skip this section.

Configure and Restart the SAS Event Stream Manager Agent

The SAS Event Stream Manager Agent is a small executable program that is installed along with SAS Event Stream Processing. If your order did not include SAS Event Stream Manager, you can skip this section.

Agents relay operational metrics from ESP servers to SAS Event Stream Manager, and they perform actions on the ESP servers in response to commands that they receive from SAS Event Stream Manager.

Take the following steps to modify agent parameters:

- 1 Stop the agent. Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-esmagent-default
```

- 2 Edit the start-up script to set the correct values for some environment variables. Use your preferred text editor to open the following file for editing: `/opt/sas/viya/home/bin/sas-esmagent`.
- 3 Locate the following environment variables within the start-up script. Set their values to environment-specific values, as specified in the following table:

Variable	Environment-Specific Value
ESM_DISCOVERY_HOST	Host name of the Apache HTTP Server, the machine that you assigned to the [httpproxy] host group in the inventory file.
ESM_DISCOVERY_PORT	The port where SAS Event Stream Manager is listening for communications from the agent. This should correspond to the port that is open on the Apache HTTP Server. The default is Port 80.
ESM_AGENT_HOSTNAME	The host name of the machine where you have installed SAS Event Stream Manager Agent and ESP server. (These components must be installed on the same machine.)
ESM_PORT	The port where the agent listens. The default setting is Port 2552.
ESM_FRIENDLY_NAME	The name of the agent that appears in the user interface of SAS Event Stream Manager. The default setting is "ESM Agent."

For more information about ESP server parameters, see [SAS Event Stream Processing: Using the ESP Server](#).

- 4 Save your changes to the start-up script.
- 5 Start the agent. Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl start sas-viya-esmagent-default
```

The following message indicates success: `sas-viya-esmagent-default is running`.

(Optional) Modify JVM Settings for SAS Event Stream Manager

If you plan to import models from SAS Model Manager, you should increase the default memory allocation for the Java Virtual Machine (JVM).

Configuring the JVM is a post-deployment task. You can use SAS Environment Manager to change the default memory settings. SAS recommends that you initially change the setting to 512 MB. For more information, see [Configuration Properties: Java Virtual Machine \(JVM\)](#) in the SAS Viya Administration documentation.

Log On to SAS Event Stream Manager

If your order included SAS Event Stream Manager, it is installed in your environment by the playbook. SAS Event Stream Manager uses SAS Logon Manager for logon functionality. SAS Logon Manager requires LDAP for user authentication.

- 1 Open SAS Event Stream Manager from a URL with the following format:

```
https://reverse-proxy-server/SASEventStreamManager
```

For *reverse-proxy-server*, use the host name from the machine that you assigned to the [httpproxy] host group in the inventory file.

The Sign In to SAS window is displayed.

- 2 Enter your user ID and password, and click **Sign In**.

Successful logon to the SAS Event Stream Manager user interface indicates that the software has been installed correctly. To validate that services have been installed and started successfully, see [“Verify SAS Event Stream Manager Status” on page 119](#).

Configure High Availability in SAS Studio

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

Note: The tasks in this section are applicable if you only deployed the programming-only interface .

Note: To use high availability in SAS Studio:

- a shared file system is required. For details, see [“\(Optional\) High-Availability Requirement” on page 22](#)
 - When you ran the playbook, you specified multiple hosts for [programming]. For details, see [“Assign the Target Machines to Host Groups” on page 58](#).
- 1 Identify your programming hosts. The programming proxy hosts are the hosts that have been listed in the [programming] host group in the inventory file.
 - 2 For each programming host, do the following:

- a Stop SAS Studio.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.x:

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

- b Determine the unique IP address of each SAS Studio instance.

```
hostname -i
```

- c Locate and edit the SAS Studio configuration file `SASCONFIG/etc/sasstudio/default/init_usermods.properties`.

- d Change the following line and add the unique IP address for that SAS Studio instance.

```
sasstudio.appserver.instanceid=sasstudio-<IP-Address>
```

Note: When you enter the IP address, replace the periods (xxx.xxx.xxx.xxx) with hyphens (xxx-xxx-xxx-xxx). An example is 123-123-123-123.

- e Save and close the SAS Studio configuration file.

After you have completed the preceding steps on all programming hosts, you have a list of IP addresses for each programming host.

```
<IP address of first SAS Studio host>
<IP address of second SAS Studio host>
<IP address of third SAS Studio host>
```

3 Start SAS Studio.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.x or later:

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

4 Identify your http proxy hosts. The http proxy hosts are the hosts that have been listed in the [httpproxy] host group in the inventory file.

5 On each http proxy host, do the following:

- a Locate and edit the `/etc/httpd/conf.d/proxy.conf` file.

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

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

- c Add the following lines to map balancers. Substitute the appropriate hosts, ports, and IP addresses.

Note: When you enter the IP address, replace the periods (xxx.xxx.xxx.xxx) with hyphens (xxx-xxx-xxx-xxx). An example is 123-123-123-123.

```
<Proxy balancer://SASStudio-cluster>
  BalancerMember http://SAS-Studio-host-1:SAS-Studio-port/ route=sasstudio-SAS-Studio-host-1-IP
  BalancerMember http://SAS-Studio-host-2:SAS-Studio-port/ route=sasstudio-SAS-Studio-host-2-IP
  ProxySet scolonpathdelim=on stickysession=JSESSIONID
</Proxy>
ProxyPass /SASStudio balancer://SASStudio-cluster/SASStudio
ProxyPassReverse /SASStudio balancer://SASStudio-cluster/SASStudio
```

Here is an example:

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

- d Save and close the proxy.conf file.

Ensure that you modified the proxy.conf file for each http proxy host.

6 On each http proxy host, start httpd:

- For Red Hat Enterprise Linux 6.7:

```
sudo service httpd restart
```

- For Red Hat Enterprise Linux 7.x or later:

```
sudo systemctl restart httpd
```

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

```
scheme://reverse-proxy-server/SASStudio
```

In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.

Configure the Default Backup Schedule

- 1 In SAS Environment Manager, confirm that the DEFAULT_BACKUP_SCHEDULE has been created. For details, see [Default Scheduling in Backup](#) in the *SAS Viya 3.3 Administration / Backup and Restore* .
- 2 Check the logs at `/opt/sas/viya/config/var/log/deploymentBackup/default` and `/opt/sas/viya/config/var/log/backup-agent/default`. If the following message is in the deploymentBackup log, restart the deploymentBackup service.

```
ServiceSchedule] c.sas.backup.util.BackupScheduleManager :
service [BACKUP_SCHEDULE_ERROR] Cannot schedule backup since maximum retry attempt
is reached and one of the dependent services is still not running
```

Restart the deploymentBackup service. Confirm that the following message is now in the log:

```
ServiceSchedule] c.sas.backup.util.BackupScheduleManager :
service Default schedule created for BackupService to run backup job every Sunday 1AM
```

- 3 Set the sharedVault location and ensure that the permissions on the designated location are set. For details, see [Backup and Restore: Service Configuration](#). in the *SAS Viya 3.3 Administration / Backup and Restore* .
- 4 In SAS Environment Manager, select the DEFAULT_BACKUP_SCHEDULE and then select **Scheduling**. To force an immediate backup, select **Run**.
- 5 To confirm that the backup ran successfully, in the **Manage Backup and Recovery** window, verify that the DEFAULT_BACKUP_SCHEDULE is now listed.

Configure Data Access

Configure SAS/ACCESS Interface to Amazon Redshift

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Amazon Redshift” on page 69](#).

- 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** For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 3** Add the following lines:

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

- 4** Save and close the `workspaceserver_usermods.sh` file.

- 5** Using a text editor, open the `cas_usermods.settings` file.

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

- 6** 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
```

- 7** Save and close the `cas_usermods.settings` file.

Configure SAS/ACCESS Interface to DB2

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

During installation, you should have configured the location of the shared libraries in the `vars.yml` file. If you did not set up the location of the shared libraries in the `vars.yml` file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the `vars.yml` file. For information, see [“SAS/ACCESS Interface to DB2” on page 69](#).

- 1** For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 2** Add the following lines:

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

- 3** Save and close the `workspaceserver_usermods.sh` file.

- 4** Using a text editor, open the `cas_usermods.settings` file.

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

- 5** Add the following lines:

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

- 6** Save and close the `cas_usermods.settings` file.

Configure SAS/ACCESS Interface to Greenplum

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Greenplum” on page 70](#).

- 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 [Greenplum Wire Protocol] template in the odbc.ini file.

- 2 For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 3 Add the following lines:

Note: Depending on how you have configured your ODBC driver, you might need to specify the odbc.ini file, the odbcinst.ini file, or both files. The following examples include both files.

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

For bulk loading, add the following lines.

```
export GPHOME_LOADERS=Greenplum-bulk-loader-installation-location
export Gpload_HOME=Greenplum-installation-location
export Gpload_PORT=Greenplum-bulk-load-port
```

- 4 Save and close the `workspaceserver_usermods.sh` file.

Configure SAS/ACCESS Interface to HAWQ

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to HAWQ” on page 71](#).

- 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 [Greenplum Wire Protocol] template in the odbc.ini file.

- For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- Add the following lines:

Note: Depending on how you have configured your ODBC driver, you might need to specify the `odbc.ini` file, the `odbcinst.ini` file, or both files. The following examples include both files.

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

- Save and close the `workspaceserver_usermods.sh` file.

Configure SAS Data Connector to Hadoop

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

During installation, you should have configured the location of the shared libraries and the library path in the `vars.yml` file. To ensure that any redeployment contains these configuration settings, you must also make these changes in the `vars.yml` file. For information, see [“SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop” on page 70](#).

To manually configure the variables:

- For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 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.

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

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

- Save and close the `workspaceserver_usermods.sh` file.
- On the CAS node(s), use a text editor to edit the `cas_usermods.settings` file:

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

- 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.

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

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

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

Configure SAS/ACCESS Interface to Impala

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

During installation, you should have configured the location of the shared libraries in the `vars.yml` file. If you did not set up the location of the shared libraries in the `vars.yml` file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the `vars.yml` file. For information, see “SAS/ACCESS Interface to Impala” on page 71.

- Install a third-party ODBC Driver Manager. The Impala ODBC driver is an ODBC API-compliant shared library. In addition, the Impala ODBC driver requires that you also install a third-party ODBC Driver Manager. A version of the unixODBC Driver Manager is available for download from the unixODBC website <http://www.unixodbc.org/>.
- To enable the Impala driver to be loaded dynamically at run time, include the full pathname of the shared library in the shared library path.
- For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 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 ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
export CLUDERAIMPALAODBC=location-of-your-cludera.impalaodbc.ini-file
export EASYSOFT_UNICODE=YES
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:/opt/cludera/impalaodbc/lib/64
```

Note: The `EASYSOFT_UNICODE` variable should only be added if you want to set the encoding for the SAS client to UTF-8.

- Save and close the `workspaceserver_usermods.sh` file.
- Using a text editor, open the `cas_usermods.settings` file.

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

- 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 ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
export CLUDERAIMPALAODBC=location-of-your-cludera.impalaodbc.ini-file
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
/opt/cludera/impalaodbc/lib/64
```

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

- 9 To use an Impala ODBC driver from a different vendor than SAS/ACCESS Interface to Impala on SAS Viya, set either the SAS_IMPALA_DRIVER_VENDOR environment variable or the DRIVER_VENDOR connection option. Here are some examples:

- Set the environment variable to use the MapR Impala ODBC driver:

```
SAS_IMPALA_DRIVER_VENDOR=MAPR
export SAS_IMPALA_DRIVER_VENDOR
```

- When defining the caslib, set the DRIVER_VENDOR variable to use the Progress DataDirect Impala ODBC driver:

```
action addCaslib lib="datalib" datasource={srctype="impala", server="impserver", schema="default",
DRIVER_VENDOR="DATADIRECT"} ; run
```

Currently, the only valid values for the driver vendor are DATADIRECT and MAPR.

Configure SAS/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).

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Microsoft SQL Server” on page 72](#).

- 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.

- 2 For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 3 Add the following lines:

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

- 4 Save and close the `workspaceserver_usermods.sh` file.
- 5 On the CAS node(s), use a text editor to edit the `cas_usermods.settings` file:

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

- 6 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
```

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

Configure SAS/ACCESS Interface to MySQL

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to MySQL” on page 72](#).

- 1 On the host(s) in the [programming] host group, use a text editor to edit the workspaceserver_usermods.sh file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 2 Add the following line:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-MySQL-client-library
```

- 3 Save and close the workspaceserver_usermods.sh file.

Configure SAS/ACCESS Interface to Netezza

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Netezza” on page 73](#).

- 1 For each host that is specified in the [programming] host group, use a text editor to edit the workspaceserver_usermods.sh file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 2 Add the following lines:

```
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbc.ini-file-including-file-name
export NZ_ODBC_INI_PATH=path-to-the-Netezza-configuration-files
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Netezza-client
```

- 3 Save and close the workspaceserver_usermods.sh file.

Configure SAS/ACCESS Interface to ODBC

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to ODBC” on page 73](#).

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

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

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

- 2 For each host that is specified in the [programming] host group, use a text editor to edit the `workspaceserver_usermods.sh` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

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

For DataDirect:

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

For iODBC:

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

For unixODBC:

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

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

- 4 Save and close the `workspaceserver_usermods.sh` file.
- 5 On the CAS node(s), use a text editor to edit the `cas_usermods.settings` file:

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

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

For DataDirect:

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

For iODBC:

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

For unixODBC:

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

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

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

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).

During installation, you should have configured the location of the shared libraries and the library path in the vars.yml file. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Oracle” on page 74](#).

To manually configure the variables:

- 1 For each host that is specified in the [programming] host group, use a text editor to edit the workspaceserver_usermods.sh file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 2 Add the following lines:

```
export ORACLE_HOME=Oracle-home-directory
export TWO_TASK=ORACLE_SID
export ORAENV_ASK=NO
export SASORA=V9
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

- 3 Save and close the workspaceserver_usermods.sh file.
- 4 On the CAS node(s), use a text editor to edit the cas_usermods.settings file:

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

- 5 Add the following lines:

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

- 6 Save and close the cas_usermods.settings file.

Configure SAS Data Connector to PostgreSQL

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

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

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

- Reference a Data Source Name (DSN).

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

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

- Specify connection information in your code.

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

```
[ODBC Drivers]
```

```

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

```

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

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to PostgreSQL” on page 75](#).

- 1 On the SAS client node, use a text editor to edit the sasenv_deployment file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 2 Add the following lines:

```

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

```

For bulk loading, add the following line:

```
export PATH=$PATH:path-to-PostgreSQL-bulk-loading
```

- 3 Save and close the sasenv_deployment file.
- 4 On the CAS node(s), use a text editor to edit the cas_usermods.settings file:

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

- 5 Add the following lines:

```

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

```

- 6 Save and close the cas_usermods.settings file.

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).

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to SAP HANA” on page 75](#).

- 1 For each host that is specified in the [programming] host group, use a text editor to edit the workspaceserver_usermods.sh file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

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

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

- 5 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
```

- 6 Save and close the cas_usermods.settings file.

Configure SAS/ACCESS Interface to SAP R/3

Note: This information is applicable only if you ordered SAS/ACCESS Interface to SAP R/3 on SAS Viya).

During installation, you should have configured the location of the shared libraries in the vars.yml file. If you did not set up the location of the shared libraries in the vars.yml file, you must configure the variable manually. To ensure that any redeployment has the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to SAP R/3” on page 76](#).

- 1 For each host that is specified in the [programming] host group, use a text editor to edit the workspaceserver_usermods.sh file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 2 Add the following lines:

```
export RFC_INI=path-to-the-SAP-R/3-ini-file
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-SAP-R/3-client
```

- 3 Save and close the workspaceserver_usermods.sh file.

Configure SAS/ACCESS Interface to to Teradata

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

During installation, you should have configured the location of the shared libraries and the library path in the vars.yml file. To ensure that any redeployment contains the configuration settings, you must also make these changes in the vars.yml file. For information, see [“SAS/ACCESS Interface to Teradata and SAS In-Database Technologies for Teradata” on page 76](#).

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 For each host that is specified in the [programming] host group, use a text editor to edit the workspaceserver_usermods.sh file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/workspaceserver_usermods.sh
```

- 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 COPER=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=$LD_LIBRARY_PATH:Teradata-TTU-installation-path-including-lib64-directory:
Teradata-TTU-installation-path-including-lib-directory:$LD_LIBRARY_PATH
```

An example of the TTU Default LD_LIBRARY_PATH is

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

5 Save and close the `workspaceserver_usermods.sh` file.

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

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

7 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
```

An example of the TTU Default LD_LIBRARY_PATH is

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

8 Save and close the `cas_usermods.settings` file.

Configure Database Connectivity for SAS Event Stream Processing

Database connections are optional for SAS Event Stream Processing. If your order included SAS Event Stream Processing, you can enable database connectivity by performing some configuration tasks after the installation has completed. For a full discussion of database connections for SAS Event Stream Processing, see [Using the Database Connector and Adapter](#).

Configure Data Quality

Configure the Quality Knowledge Base

Note: This information is applicable only if your order contains SAS Data Quality.

For an overview of SAS Data Quality, see [SAS Data Quality 3.3 / Getting Started](#). SAS Data Quality relies on a collection of rules and reference data called a SAS Quality Knowledge Base (QKB). To use the SAS Data Quality DATA step functions in CAS or to invoke Data Quality transformations in a job in the SAS Data Preparation application suite, you must configure SAS Data Quality to use a QKB.

To configure SAS Data Quality to use the QKB that is included in your software deployment:

- 1 Import the QKB to CAS.
 - For full deployments, use SAS Environment Manager to import the QKB to CAS. For instructions, see [Import a QKB](#) in *SAS® Viya 3.3 Administration: QKB Management*.
 - For programming-only deployments, use a CAS action to import the QKB to CAS. For instructions, see [SAS Data Quality 3.3 / Data Quality Action Programming Guide / importQKBFromCaslib Action](#).
- 2 Update the CAS configuration on the CAS controller machine by editing the `casconfig_usermods.lua` file:

- In Ansible-based deployments, the `casconfig_usermods.lua` file is located in the `/opt/sas/viya/config/etc/cas/default` directory on the CAS controller machine.
- For single-machine, programming-only environments that have been deployed using yum, the `casconfig_usermods.lua` file may not exist and will need to be created in the `/opt/sas/viya/config/etc/cas/default` directory.

Edit the `casconfig_usermods.lua` file by using the following command:

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

- 3 Add or locate the `cas.dqSetupLoc` and `cas.dqLocale` variables in the `casconfig_usermods.lua` file. Specify the QKB that you want to make the default QKB, and the ISO code name for the QKB locale that you want to make the default QKB locale. For information about QKB ISO code names, see [QKB Locale ISO Codes](#) on the SAS Support website.
 - For the value of `dqSetupLoc`, be sure to specify the name of the QKB exactly as you specified it when you imported your QKB to CAS.
 - For a list of the locales that are supported by your QKB, browse the contents of your QKB in SAS Environment Manager or refer to the documentation for your QKB.

```
cas.dqSetupLoc="default-QKB"  
cas.dqLocale="ISO-code-name-for-default-QKB-locale"
```

Here is an example:

```
cas.dqSetupLoc="QKBCI28"  
cas.dqLocale="ENUSA"
```

- 4 Save and close the `casconfig_usermods.lua` file.
- 5 Restart the CAS controller.

Validating the Deployment

<i>Perform Installation Qualification on RPM Packages</i>	115
<i>Access CAS Server Monitor</i>	117
<i>Access SAS Environment Manager</i>	118
<i>Verify SAS Message Broker</i>	118
<i>Verify SAS Infrastructure Data Server</i>	119
<i>Verify SAS Event Stream Manager Status</i>	119
<i>Overview of Data Access Verification</i>	120
<i>Verify SAS/ACCESS Interface to Amazon Redshift</i>	120
<i>Verify SAS/ACCESS Interface to DB2</i>	121
<i>Verify SAS/ACCESS Interface to Greenplum</i>	121
<i>Verify SAS/ACCESS Interface to HAWQ</i>	122
<i>Verify SAS/ACCESS Interface to Impala</i>	123
<i>Verify SAS/ACCESS Interface to Microsoft SQL Server</i>	124
<i>Verify SAS/ACCESS Interface to MySQL</i>	125
<i>Verify SAS/ACCESS Interface to Netezza</i>	125
<i>Verify SAS/ACCESS Interface to ODBC</i>	126
<i>Verify SAS/ACCESS Interface to Oracle</i>	127
<i>Verify SAS/ACCESS Interface to PostgreSQL</i>	127
<i>Verify SAS/ACCESS Interface to SAP R/3</i>	128

Perform Installation Qualification on RPM Packages

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

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

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

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

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

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

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

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

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

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

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

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

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

The error indicators are shown in the following format:

```
SM5DLUGT c
```

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

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

The meaning of each error indicator is described as follows:

- S File size. RPM keeps track of file sizes. A difference of even one byte triggers a verification error.
- M File mode. The permissions mode is a set of bits that specifies access for the file's owner, group members, and others. Even more important are two additional bits that determine whether a user's group or user ID should be changed if they execute the program that is contained in the file. Since these bits permit any user to become root for the duration of the program, you must be cautious with a file's permissions.
- 5 MD5 checksum. The MD5 checksum of a file is a 128-bit number that is mathematically derived from the contents of the file. The MD5 checksum conveys no information about the contents of the original file, but, any change to the file results in a change to the MD5 checksum. RPM creates MD5 checksums for all files that it manipulates, and stores the checksums in its database. If one of these files is changed, the MD5 checksum changes and the change is detected by RPM.
- D Major and minor numbers. Device character and block files contain a major number. The major number is used to communicate information to the device driver that is associated with the special file. For example, under Linux, the special files for SCSI disk drives should have a major number of 8, and the major number for an IDE disk drive's special file should be 3. Any change to a file's major number could produce disastrous effects. RPM tracks such changes. A file's minor number is similar to the major number, but conveys different information to the device driver. For disk drives, this information can consist of a unit identifier.
- L Symbolic link. If a file is a symbolic link, RPM checks the text string that contains the name of the symbolically linked file.
- U File owner. Most operating systems keep track of each file's creator, primarily for resource accounting. Linux and UNIX also use file ownership to help determine access rights to the file. In addition, some files, when executed by a user, can temporarily change the user's ID, normally to a more privileged ID. Therefore, any change of file ownership might have significant effects on data security and system availability.

- G File group. Similar to file ownership, a group specification is attached to each file. Primarily used for determining access rights, a file's group specification can also become a user's group ID if that user executes the file's contents. Therefore, any changes in a file's group specification are important and should be monitored.
- T Modification time. Most operating systems keep track of the date and time that a file was last modified. RPM keeps modification times in its database.
- c Configuration file. This is useful for quickly identifying configuration files, since they are likely to change and therefore are unlikely to verify successfully. You could also get a d in this slot, indicating that the file is for documentation, which is also likely to change often.

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

Note: In SAS Viya 3.3, the following files are modified during the deployment process. If you perform a verification and receive error indications for the following files, they can be safely ignored. The following are the default pathnames.

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

Access CAS Server Monitor

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

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

Note: In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.

- For single tenant:

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

Here is an example:

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

- For multi-tenant:

`scheme://tenant.reverse-proxy-server/cas-tenant-instance-http`

Here is an example:

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

Log on using one of the SAS Administrator users that you established in [“Set Up Administrative Users” on page 93](#).

Note: To access CAS Server Monitor, the password must be set for the cas user ID or other administrative account. To set the password, see [“Set the Password for the CAS Administrator or Another Administrative Account” on page 89](#).

If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the `Your connection is not private` message. SAS recommends replacing the certificates before giving end-users access to SAS Viya. For information, see [“Transport Layer Security” on page 36](#).

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

Access SAS Environment Manager

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

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

- 1 Go to the machine you assigned to the [AdminServices] host group.
- 2 Open SAS Environment Manager from a URL with the following format:

```
https://reverse-proxy-server/SASEnvironmentManager
```

Note: If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the `Your connection is not private` message. SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see [“Transport Layer Security” on page 36](#)

- 3 Sign on as one of the SAS Administrators that you set up in [“Set Up Administrative Users” on page 93](#).

Verify SAS Message Broker

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

- 1 To verify that SAS Message Broker has been deployed correctly, go to the machine that you assigned to the [rabbitmq] host group.
- 2 Open a browser and go to the following address:

- If HTTPS is enabled:

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

Note: If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the `Your connection is not private` message. SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see [HTTPS Access to SAS Message Broker](#).

- If HTTP is enabled:

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

Note: To determine whether HTTPS or HTTP is enabled, see “Specify Security Settings” on page 64.

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

Verify SAS Infrastructure Data Server

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

Use these steps to verify that SAS Infrastructure Data Server has been deployed correctly.

- 1 On the machine that you assigned to the [pgpoolc] host group, run the following command:

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

- 2 If SAS Infrastructure Data Server 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 the node is running.

Verify SAS Event Stream Manager Status

To verify that a deployment of SAS Event Stream Manager has completed successfully, check that the required SAS services are available. You can check the status of all the SAS Event Stream Manager services by running the following the following commands on Red Hat Enterprise Linux 6.x:

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

```
sudo service sas-viya-esm-webui-default status
```

Run the following commands on Red Hat Enterprise Linux 7.x:

```
sudo systemctl status sas-viya-esm-service-default
```

```
sudo systemctl status sas-viya-esm-webui-default
```

Here is typical command output from Red Hat Enterprise Linux 6.7 to indicate that the software is running normally:

```
sas-viya-esm-service-default is running
sas-viya-esm-webui-default is running
```

The output is different on Linux 7.x, but it reports that each service is running.


Overview of Data Access Verification

If any of the verification steps for data access return an error, perform the appropriate configuration steps again. For details, see [“Configure Data Access” on page 102](#).

Verify SAS/ACCESS Interface to Amazon Redshift

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

To verify that SAS Data Connector to Amazon Redshift was successfully deployed:

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Amazon Redshift LIBNAME statement:

```
libname arslib redshift server=Redshift-host-name database=Redshift-database-name user="user-ID"
password=user-password;
```

If SAS/ACCESS to Amazon Redshift was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Amazon Redshift:

```
caslib rslib datasource=(srctype="redshift", username="user-ID", password="password",
server="Redshift-host-name", database="Redshift-database-name");
```

```
proc casutil;
list files incaslib="rslib";
run;
```


If the data connector was successfully deployed, the results are the names of the tables in Amazon Redshift.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to DB2

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

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to DB2 LIBNAME statement:

```
libname db2lib db2 database="DB2-database-name" user="user-ID"
password="user-password";
```

If SAS/ACCESS to DB2 was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to DB2:

```
caslib db2clib datasource=(srctype="db2", username="user-ID",
password="password", database="DB2-database-name");

proc casutil;
list files incaslib="db2clib";
run;
```

If the data connector was successfully deployed, the results are the names of the tables in DB2.


If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to Greenplum

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

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

- 1 Sign on to SAS Studio:

- a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*.
In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
 - 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Greenplum LIBNAME statement:

```
libname glib greenplum server="greenplum-host-name" database="greenplum-database-name" user="user-ID"
password="user-password";
```


If SAS/ACCESS to Greenplum was successfully deployed, the execution of the LIBNAME statement will return results without error.

If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to HAWQ

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

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*.
In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to HAWQ LIBNAME statement:


```
libname hawqlib hawq server="hawq-host-name" database="hawq-database-name" user="user-ID"
password="user-password";
```


If SAS/ACCESS to HAWQ was successfully deployed, the execution of the LIBNAME statement will return results without error.

If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to Impala

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services**.
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.

- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Impala LIBNAME statement:

```
libname imp impala server="Impala-host-name" database="Impala-database-or-schema-name" user="user-ID"
password="user-password";
```

If SAS/ACCESS to Impala was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Impala::

```
caslib implib datasource=(srctype="impala", username="user-ID",
password="user-password", server="Impala-host-name", database="Impala-database-or-schema-name");

proc casutil;
  list files incaslib="implib";
run;
```


If the data connector was successfully deployed, the results are the names of the tables in Impala.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to Microsoft SQL Server

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

To verify that SAS Data Connector to Microsoft SQL Server was successfully deployed:

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Microsoft SQL Server LIBNAME statement:

```
libname mslib sqlsvr dsn="DSN-from-odbc.ini" user="user-ID" password="user-password";
```

If SAS/ACCESS to Microsoft SQL Server was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Microsoft SQL Server:

```
caslib msclib datasource=(srctype="sqlserver", username="user-ID",  
password="password", odbc_dsn="DSN-from-odbc.ini");  
  
proc casutil;  
list files incaslib="msclib";  
run;
```


If the data connector was successfully deployed, the results are the names of the tables in Microsoft SQL Server.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to MySQL

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

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.

- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to MySQL LIBNAME statement:

```
libname mylib mysql server="mysql-host-name" database="mysql-database-name" user="user-ID"
password="user-password";
```

If SAS/ACCESS to MySQL was successfully deployed, the execution of the LIBNAME statement will return results without error.


If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to Neteeza

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

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.

- b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Netezza LIBNAME statement:

```
libname nlib netezza server="netezza-host-name" database="netezza-database-name" user="user-ID"
password="user-password";
```


If SAS/ACCESS to Netezza was successfully deployed, the execution of the LIBNAME statement will return results without error.

If an error was returned on the execution of the LIBNAME statement, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to ODBC

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

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to ODBC LIBNAME statement:

```
libname olib odbc dsn="DSN-from-odbc.ini" user="user-ID" password="password";
```

If SAS/ACCESS to ODBC was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to ODBC:

```
caslib odbclib datasource=(srctype="odbc", username="user-ID",
password="password", odbc_dsn="DSN-from-odbc.ini");
```


```
proc casutil;
list files incaslib="odbclib";
run;
```

If the data connector was successfully deployed, the results are the names of the tables in ODBC.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to Oracle

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*. In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Oracle LIBNAME statement:

```
libname olib oracle path="path-to-database" user="user-ID" password="user-password";
```

If SAS/ACCESS to Oracle was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Oracle:

```
caslib oralib datasource=(srctype="oracle" username="user-ID" password="password"
path="path-to-database" schema="schema-ID");

proc casutil;
list files incaslib="oralib";
run;
```


If the data connector was successfully deployed, the results are the names of the tables in Oracle.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to PostgreSQL

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

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

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*.
In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services**.
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to PostgreSQL LIBNAME statement:


```
libname plib postgres server="PostgreSQL-host-name" database="PostgreSQL-database-name" user="user-ID"
password="password";
```

If SAS/ACCESS to PostgreSQL was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to PostgreSQL:

```
caslib pglib datasource=(srctype="postgres", username="user-ID", password="password",
server="PostgreSQL-host-name", database="PostgreSQL-database-name");

proc casutil;
list files incaslib="pglib";
run;
```

If the data connector was successfully deployed, the results are the names of the tables in PostgreSQL.


If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Verify SAS/ACCESS Interface to SAP R/3

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to R/3 (on SAS Viya).

To verify that SAS Data Connector to SAP R/3 was successfully deployed:

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: *scheme://reverse-proxy-server/SASStudio*.
In a programming-only deployment, the scheme is http. In a full deployment, the scheme is https.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.

- b** Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
 - c** Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d** In the toolbar, click  to run the new CAS session code.
- 3** From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to SAP R/3 LIBNAME statement:

```
libname bwides r3 user="user-ID" password="user-password" client=800 ashost="sap-host-name" sysnr=06;
```

If SAS/ACCESS to SAP R/3 was successfully deployed, the execution of the LIBNAME statement will return results without error.

If an error was returned on the execution of the LIBNAME statement or no table information returned for the data connector, you should perform the configuration steps again.

Completing the Deployment

<i>Save Snapshot Directory Content</i>	131
<i>Share Important Deployment Information with the Administrators</i>	131
<i>Next Steps for SAS Event Stream Processing Users</i>	132
<i>Refer to Additional Documentation</i>	132

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 deployment type: Did you deploy the programming interface only or did you perform a full deployment?
 - If you used the customized script from your playbook (described in [“Deploying with Yum” on page 171](#)), you deployed the programming interface only.
 - If you used Ansible, the type was determined by either the `sas_install_type` variable in the `vars.yml` file (see [“Specify the Installation Type” on page 63](#)) or the command line that you used when deploying your software (see [“Options” on page 85](#)).
- The URL to access the software: What products did you deploy?
 - If you deployed only products from the SAS Event Stream Processing Family, there is no URL to share.
 - If you deployed the programming interface only, your administrators should use SAS Studio. The URL is `http://reverse-proxy-server/SASStudio`.

- If you performed a full deployment, your administrators should use SAS Environment Manager. You used SAS Environment Manager to configure your environment for a full deployment as described in [“Configure Your Environment with SAS Environment Manager” on page 91](#). Use the same URL that you used in that section.

Next Steps for SAS Event Stream Processing Users

If your order included SAS Event Stream Processing, you might need to complete the following steps:

- The Streamviewer component was also installed along with SAS Event Stream Processing. Streamviewer is a graphical user interface that enables you to visualize events as they stream through event stream processing models. Its use is optional.
A few configuration steps are required before you can start using Streamviewer. For more information, see [Setting Up and Running Streamviewer](#).
- If your order included SAS Event Stream Processing for CAS, you now have the option to use an additional CAS action set, espCluster. A second SAS Event Stream Processing action set, loadStreams, is included with all SAS Viya orders. For more information, see [Using SAS Event Stream Processing with SAS Cloud Analytic Services Actions](#).
- Read additional documentation about SAS Event Stream Processing. Links to all SAS Event Stream Processing documentation are available on the [SAS Event Stream Processing product page](#). All product user documentation is also available via single sign-on from the SAS Event Stream Processing user interfaces.

Refer to Additional Documentation

After you validate the deployment, unless you deployed products from the SAS Event Stream Processing family only, you can perform initial administrative tasks. For more information, refer to [SAS Viya 3.3 Administration: Initial Tasks](#).

The documentation for SAS Event Stream Processing is available on the [SAS Event Stream Processing product page](#). All product user documentation is also available via single sign-on from the SAS Event Stream Processing user interfaces (SAS Event Stream Processing Studio and Streamviewer).

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

You can also refer to the appendices in this guide for additional tasks that you might perform based on your environment. For example, the appendixes include information to help you configure your Hadoop infrastructure.

Managing Your Software

Overview	133
What Is an Update?	133
What Is an Upgrade?	134
What Is an Add-On Product?	134
What Is a New Ansible Playbook?	134
Apply the CVE-2017-7547 Security Patch	134
Updating Your SAS Viya Software	134
Overview	134
List the Packages That Are Available for Update	135
Update Your SAS Viya Software	135
Adding SAS Viya Software	140
Overview	140
How to Add SAS Viya Software	140
Upgrading Your SAS Viya Software	143
Overview	143
Update SAS Visual Text Analytics with Yum	143
Prepare to Upgrade SAS Viya Software	144
Prepare to Upgrade SAS Event Stream Processing Software	145
Change the dbmsowner User Password	146
Stop a Clustered RabbitMQ Configuration	147
Upgrade SAS Viya Software	148
Preserve Access Controls for Database Caslibs	150
Generate a New Ansible Playbook	151

Overview

What Is an Update?

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

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

What Is an Upgrade?

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

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

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

What Is an Add-On Product?

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

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

What Is a New Ansible Playbook?

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

Apply the CVE-2017-7547 Security Patch

Important: The patch is applied only at sites that run versions of PostgreSQL that are earlier than version 9.4.13.

A new security patch, [CVE-2017-7547](#), fixes a password security issue in PostgreSQL databases.

Sites that run SAS Viya 3.3 have a newer version of PostgreSQL (version 9.4.13) that contains the fix for this security issue.

However, the following sites must manually apply patch SAS Infrastructure Data Server with patch CVE-2017-7547.

- sites running SAS Viya 3.2 (and earlier) that upgrade to SAS Viya 3.3 or SAS Viya 3.4
- sites running SAS Viya 3.2 (and earlier) that are not running PostgreSQL (version 9.4.13) or later

For explicit steps to apply the patch, see [Apply the CVE-2017-7547 Security Patch](#) in *SAS Viya Administration*.

Updating Your SAS Viya Software

Overview

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

- If you used an Ansible playbook for your initial installation, you should update with Ansible.
- If you used yum for your initial installation, you should update with yum.
- If you mirrored your software, you need to update the mirror.
- Using Ansible, you can modify your deployment from programming-only to full.

Updating SAS Viya software requires an outage period because some SAS Viya services are stopped and restarted automatically during the update process. The update process is the same regardless of whether the deployment is single-tenant or multi-tenant.

Note: 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.

List the Packages That Are Available for Update

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

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

Note: If you are working with a mirrored repository, you must synchronize the connected mirror repository machine with SAS before checking for updates. To synchronize, run the following command on the Ansible controller machine:

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

After the synchronization, list the packages on the machines where the software has been deployed.

Update Your SAS Viya Software

Overview

The update process brings your deployed software up-to-date with the latest compatible software. You perform the update with the same command that was used to install SAS Viya, and you use the same software order and the same playbook. Running the playbook updates all software to the latest version.

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

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

After the mirror repository has been synched, you can then update your deployment using the mirror repository. If you have not deployed SAS software before, then you can proceed with your initial deployment using the steps in the installation chapter beginning with [“Edit the Inventory File” on page 56](#).

There are two tools that can be used to install SAS Viya: Ansible or yum. For multi-machine deployments, Ansible is the preferred tool for installing and updating software.

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 command: `sudo -v` or `sudo -l`.

Update with Ansible

To update a SAS Viya deployment using Ansible:

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

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

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

On each machine in your deployment, create a file that lists the SAS 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/viya_yumgroups.txt
```

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

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

- 2 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.

By default, the update process backs up the following files:

For CAS:

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

For SAS Object Spawner:

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

For SAS/CONNECT:

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

For programming-only deployments:

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

- 3 On the pgpool server machine, stop the SAS Infrastructure Data Server cluster. If your environment has more than one cluster configured, ensure all are shutdown.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-sasdatasvrc-postgres
```

- 4 If you have deployed SAS Event Stream Processing, perform the following steps:

- a Stop the SAS Event Stream Processing Studio (espm) service.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-espm-default
```

- b** If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where the Streamviewer files are installed and running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

- c** Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

- 5** To initiate the update, run the same command and options that you ran when you performed the initial deployment.

For example, if you used the command and options `-e "sas_install_type=programming"` for your initial deployment, you would run the same command and options for an update. For more information, see [“Deploy the Software” on page 85](#).

- 6** (Optional) To modify your deployment from programming-only to full, see [“Modify the Deployment Type with Ansible” on page 137](#).

- 7** Wait for all services to start.

- 8** (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 136](#)) and after the update.

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

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

Modify the Deployment Type with Ansible

Overview

The deployment type can be modified by adding additional options during the update process. To modify the deployment, run the current playbook using the existing `inventory.ini` file and the `vars.yml` file. Running the playbook updates all software to the latest version.

If you change from a programming deployment to a full deployment, you must complete the configuration for the visual attributes of the deployment. For more information see [“Post-Installation Tasks” on page 89](#).

When the programming interface is deployed, SAS Studio is accessible on port 7080. However, when updating from a programming-only deployment to a full deployment, SAS Studio uses a dynamically assigned port.

Note: If you make changes to the `proxy.conf` file and then rerun the playbook, those changes are overwritten. A copy of the `proxy.conf` should be created in the `/etc/httpd/conf.d` directory when you rerun the playbook. Use this copy, along with the instructions in “[Post-Installation Tasks](#)” on page 89 to make changes to the updated `proxy.conf` file.

Deployment Modification Commands

The following options can be specified in the Ansible command line:

- To deploy only the programming interface, including CAS, SAS Foundation, and SAS Studio:

```
-e "sas_install_type=programming"
```

When you deploy the programming interface, SAS Studio uses port 7080.

- To deploy the full set of interfaces:

```
-e "sas_install_type=all"
```

The default option is `all` unless another option is specified.

Update with Yum

If you used `yum` to install a single-machine programming interface, you should use `yum` to update the SAS Viya software on that machine.

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

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

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

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

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

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

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

- 2 Stop the PostgreSQL service.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-sasdatasvrc-postgres
```

- 3 If you have deployed SAS Event Stream Processing, perform the following steps:

- a Stop the SAS Event Stream Processing Studio (espm) service.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-espm-default
```

- b (Optional) If you installed Streamviewer, stop the Streamviewer process:


```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where Streamviewer is installed and running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

c Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

4 To update all SAS Viya software on the machine:

```
sudo yum update "@SAS*" "@CAS*" "sas-*
```

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

5 At the prompt *Is this ok*, review the available updates and then enter *y*.

6 Start the PostgreSQL service.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl start sas-viya-sasdatasvrc-postgres
```

7 If the CAS controller fails to start because of a permission denied error, navigate to `/opt/sas/viya/config/data/cas/default/referenceData`, delete the files, and start the CAS controller again.

8 (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 138](#)) and after the update.

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

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

Repeat these steps for each machine in the deployment.

Adding SAS Viya Software

Overview

Here are some of the most common scenarios for adding SAS Viya software to your existing deployment:

- Adding new software from your initial SAS Viya order.
You ordered software and did not install all of it. Or you installed the software, but chose not to configure it.
- Deploying additional software from a new SAS Viya order.
The additional software is not a part of your original SAS Viya order. You might have made another order and now have to download and deploy the new order.
- Re-installing and reconfiguring SAS software.
You want to move SAS software to a new machine.
- Applying updates (maintenance) to SAS software that requires also updating its configuration.
You were unable to finish applying the maintenance updates and you need to rerun the playbook in order to complete the updates to your configuration.

Adding SAS Viya software to an existing deployment requires an outage period because some SAS Viya services are stopped and restarted automatically during the update process. The update process is the same regardless of whether the deployment is single-tenant or multi-tenant.

Before you begin, you should consider reviewing the [“Introduction” on page 1](#), [“System Requirements” on page 17](#), and [“Pre-installation Tasks” on page 41](#) chapters of this guide.

How to Add SAS Viya Software

To add SAS software and update a SAS Viya deployment:

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

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

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

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

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

- 2 If you are adding software to an order that was created with a mirrored repository, rename `/opt/sas/repomirror/repo_override.txt` to preserve it after the new `repo_override.txt` file is created.

```
mv /opt/sas/repomirror/repo_override.txt /opt/sas/repomirror/repo_override_old.txt
```

- 3 When you purchase additional products, you receive a new Software Order Email (SOE) from SAS. Use your SOE to download the SAS Orchestration CLI.

- 4 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site.
- 5 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/` 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/addon/
```

- 6 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.

```
diff /sas/install/sas_viya_playbook/vars.yml /sas/addon/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory.ini /sas/addon/sas_viya_playbook/inventory.ini
```

- b If the new files contain new content, then merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.
 - c If the original vars.yml file contains a value for the casenv_tenant variable, it must be removed. Run the following commands to remove the registered CAS service.

```
cd /opt/sas/viya/home/bin
./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token \
agent service deregister \
"cas-{casenv_tenant}-default-http"

./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token \
agent service deregister \
"cas-{casenv_tenant}-default"
```

- d If you have questions about whether to add a key/value pair from an older file to the new file, contact SAS Technical Support.

- 7 If you have deployed SAS Event Stream Processing, perform the following steps:

- a Stop the SAS Event Stream Processing Studio (esvvm) service.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-esvvm-default
```

- b** (Optional) If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where Streamviewer is running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

- c** Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

- 8** If you are adding software to an order that was created with a mirrored repository, on the Ansible controller machine, from the `sas_viya_playbook` directory, run the following command:

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

- 9** If you are adding software to an order that was created with a mirrored repository, install your SAS software using the steps in [“Deploy the SAS Viya Software to the Deployment Targets” on page 194](#). Otherwise, install your SAS Viya software using the steps in the installation chapter beginning with [“Modify the vars.yml File” on page 62](#).

- 10** If you removed the CAS service that is associated with a `casenv_tenant` value described in Step 5, ensure that any bookmarked URLs are updated to remove that value and use `cas-shared-default-http` instead. For example, if your original deployment contained a `casenv_tenant` value of `viya32`, then

```
http://host.company.com/cas-viya32-default-http
```

should be changed to the following URL:

```
http://host.company.com/cas-shared-default-http
```

Note: Do not include `casenv_tenant` in your new `vars.yml`. This property has been deprecated.

- 11** After the software has been installed, complete the following tasks, as appropriate:

- a** [“Configure the Connection to the Mail Service” on page 93](#).
- b** [“Configure SAS Viya to Connect to LDAPS Provider ” on page 94](#).
- c** [“Configure a Symbolic Link to a Storage Platform” on page 94](#).
- d** [“Verify That Licenses Are Applied” on page 95](#).
- e** If SAS Event Stream Processing is added, [“Complete SAS Event Stream Processing Setup” on page 95](#).
- f** If SAS Event Stream Manager is added, [“Complete SAS Event Stream Manager Setup” on page 98](#).
- g** [“Configure High Availability in SAS Studio” on page 100](#).
- h** If you have added any SAS/ACCESS software to your deployment as part of upgrading to SAS Viya 3.3, see the appropriate topics in [“Configure Data Access” on page 102](#) for instructions to configure the new software.
- i** [“Configure Data Quality” on page 113](#).
- j** [Validate the Deployment on page 115](#).
- k** [Complete the Deployment on page 131](#).
- l** Any appendixes containing information relevant to your deployment.

12 (Optional) After the process has completed, record the new list of installed software.

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

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

On each machine in your deployment, create a file that lists the SAS 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
```

Note: If messages appear that are similar to the following example

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

These messages can be safely ignored.

You can see the differences between the previous and current deployments by comparing the lists of installed software before ([Step 1 on page 140](#)) and after performing this task.

Upgrading Your SAS Viya Software

Overview

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

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

Upgrading SAS Viya software requires an outage period because some SAS Viya services are stopped and restarted automatically during the update process. The update process is the same regardless of whether the deployment is single-tenant or multi-tenant.

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

Update SAS Visual Text Analytics with Yum

If upgrading a SAS Visual Text Analytics 8.2 system to a SAS Visual Text Analytics 8.3 system, or transferring a project from a SAS Visual Text Analytics 8.2 environment to a SAS Visual Text Analytics 8.3 environment, and your deployment is on Red Hat Enterprise Linux or an equivalent distribution, perform the steps in this section.

Note: Do not use Ansible to update SAS Visual Text Analytics before upgrading SAS Viya.

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`.

See [“Updating Your SAS Viya Software” on page 134](#) for more information.

On each machine defined in the [VisualTextAnalytics] host group in the inventory file, perform the following steps.

- 1 Run the following command.

```
yum info sas-text-gateway
```

- 2 Find `sas-text-gateway` in the response to the command. If `2017` is not present in the `Release` field for `sas-text-gateway`, then skip the rest of these steps for the current machine.

- 3 Run the following commands.

```
sudo service sas-viya-text-gateway-default stop
sudo yum update sas-text-gateway
sudo service sas-viya-text-gateway-default start
```

- 4 As a member of the SAS Administrators group, log on to SAS Model Studio.

```
http://hostname/ModelStudio
```

Use the host name of the machine that you assigned to the [httpproxy] host group in the inventory file. For more information about assigning machines, see [“Assign the Target Machines to Host Groups” on page 58](#).

The project listing will be displayed which completes the Visual Text Analytics update.

- 5 Log off from SAS Model Studio.

Prepare to Upgrade SAS Viya Software

To prepare to upgrade a SAS Viya deployment:

Note: Be sure to follow the steps that are described in [“Perform Linux Tuning” on page 50](#) on the target machine before starting the upgrade process. Also be aware that system requirements for RAM, CPU, and disk space are likely to change with each SAS Viya release. Verify that your environment meets the requirements that are listed in [“System Requirements” on page 17](#).

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

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

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

On each machine in your deployment, create a file that lists the SAS 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/viya_yumgroups.txt
```

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

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

- 2 When performing an upgrade, you receive a new Software Order Email (SOE) from SAS. Use your SOE to download the SAS Orchestration CLI.
- 3 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site.
- 4 You must extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to `/sas/install/`, you might extract the new playbook to `/sas/upgrade/` instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:
 - To preserve the original `vars.yml` file and the inventory file.

- To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following:

```
tar xf SAS_Viya_playbook.tgz -C /sas/upgrade/
```

5 Merge the vars.yml file and the inventory file from the previous deployment into the new playbook.

- a Compare the two vars.yml files, and compare the two inventory files to check for additions or changes in the newer set of files.

```
diff /sas/install/sas_viya_playbook/vars.yml /sas/upgrade/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory-file /sas/upgrade/sas_viya_playbook/inventory.ini
```

- b If the new files contain new content, merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.
- c If the original vars.yml file from the deployment that is being upgraded contains a value for the casenv_tenant variable, it must be removed. Run the following commands to remove the registered CAS service.

```
cd /opt/sas/viya/home/bin
./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/c
lient.token \
agent service deregister \
"cas-{casenv_tenant}-default-http"

./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/c
lient.token \
agent service deregister \
"cas-{casenv_tenant}-default"
```

- d If you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.

Prepare to Upgrade SAS Event Stream Processing Software

If you purchased and installed SAS Event Stream Processing or SAS Event Stream Manager, additional steps are required in the upgrade process.

- 1 Create a backup copy of the SAS Event Stream Processing Studio database in order to preserve project files. Follow these steps:
 - a Stop the SAS Event Stream Processing Studio (esvvm) service by running the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-esvvm-default
```

- b Create a backup copy of the database, which is a single binary file (studio.mv.db). You can copy it to any directory location outside the SAS Event Stream Processing installation directory structure.

The location and filename of the database are determined by the environment variable `ESP_STUDIO_DB`. By default, it is stored in `/opt/sas/viya/config/data/espvm/`.

To create the backup, run the following command:

```
cp studio.mv.db directory-name
```

- 2 If you are upgrading from Streamviewer 4.3, find its process ID so that you can kill the Streamviewer service:

```
ps -ef
```

Kill the Streamviewer process, substituting the process ID that was returned in the previous step:

```
kill -9 process-ID
```

- 3 If you are upgrading from SAS Event Stream Manager 4.3, run the following command to delete the schema:

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv delete 'config/esm-service/spring/datasource/schema'
```

Change the dbmsowner User Password

Although SAS Viya 3.2 permitted the use of nonalphanumeric characters in passwords, SAS Viya 3.3 does not support nonalphanumeric characters in passwords. To remove any nonalphanumeric characters that might exist in the dbmsowner account password, you must change the dbmsowner account password before you upgrade your deployment from SAS Viya 3.2 to SAS Viya 3.3.

The script, `sds_change_user_pw.sh`, changes either the dbmsowner account password or the sas account password for SAS Infrastructure Data Server. It also synchronizes the new password with configuration files and SAS Configuration Server, which is based on Consul.

Note: To change the password, you must know the current password. For more information, see [Get Current Passwords](#) in SAS Viya 3.3 Administration: Infrastructure Servers.

Note: Changing the dbmsowner user password is not required when upgrading programming-only environments.

- 1 Log on to the machine as the SAS install user (`sas`).
- 2 Locate the data server environment variables file, `sds_env_var.sh`, and record its location.
By default, `sds_env_var.sh` resides in `/opt/sas/viya/config/etc/sasdatasvrc/postgres/pgpool0`.
- 3 Collect the following information, which you will specify when prompted by the `sds_env_var.sh` script. You will run the script in a later step.
 - database user name
 - current database password
 - new database password

Note: Your password must conform to the password policy. For more information, see [Password Policy](#) in SAS Viya Administration: Infrastructure Servers.

- 4 Using the location of `sds_env_var.sh` that was noted in step 2, run the script:

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


TIP If you run the script from the directory where it resides, you might see several `cannot open [No such file or directory]` messages. This is a known issue, and you can safely ignore these messages.

- 5 Enter the information that you collected in step 3 in response to the script's prompts.

After you provide values for the prompts, the script connects to SAS Configuration Server and sets all instances of the database user password that it finds. Changes made in the configuration server are synchronized with the proper SAS Infrastructure Data Server configuration files. Finally, the script runs the necessary SQL commands in the data server to set the permissions for the database user.

- 6 To verify that your password was successfully changed, connect to the data server's first database, **postgres**, using the PostgreSQL interactive terminal, `psql`:

```
/opt/sas/viya/home/bin/psql -h data-server-machine-name -U dbmsowner postgres
```

This command connects you to PostgreSQL as the dbmsowner.

- 7 When prompted, enter the new password for dbmsowner.

- 8 Type `\q` to exit the `psql` interface.

- 9 Stop all SAS Viya services.

To stop all the SAS services on the machine, run the following command:

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

- 10 Start all SAS Viya services.

To start all the SAS services on the machine, run the following command:

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

Stop a Clustered RabbitMQ Configuration

If RabbitMQ is deployed in a clustered configuration, additional steps are required in the upgrade process.

- 1 On your Ansible controller host, locate the `[rabbitmq]` host group in your playbook inventory file.
- 2 If only one RabbitMQ target is defined, skip the rest of this section and upgrade your SAS Viya deployment.
- 3 If more than one RabbitMQ target is defined, log on to the last rabbitMQ target.
- 4 Stop the RabbitMQ server using the appropriate command:

- If the RabbitMQ target is a 6.x Linux system, run the following command:

```
sudo service sas-viya-rabbitmq-server-default stop
```

- If the RabbitMQ target is a 7.x Linux system, run the following command:

```
sudo systemctl stop sas-viya-rabbitmq-server-default
```

- 5 For the other RabbitMQ targets, log on to each RabbitMQ target and run a command to stop RabbitMQ, in the reverse order in which they are listed in the inventory.ini file.

- 6 Edit the file `/sas/upgrade/sas_viya_playbook/internal/config-start.yml` to set the line `include: rabbitmq.yml` immediately after the line `include: pgpoolc.yml`. Here is an example:

```
include: pgpoolc.yml
```

```
include: rabbitmq.yml
```

- 7 Save your changes to `/sas/upgrade/sas_viya_playbook/internal/config-start.yml` and close the text editor.

Upgrade SAS Viya Software

To upgrade a SAS Viya deployment:

- 1 Log on to the primary PostgreSQL machine in your deployment.

- 2 Run the following command:

```
sudo cat -n /opt/sas/viya/config/etc/sasdatasvrc/postgres/pgpool0/pool.cdf
```

All entries in the command's output should display `healthy`.

- 3 Run the following command:

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

- 4 Open `vars.yml` and locate the `INVOCATION_VARIABLES` section.

- 5 Compare the `NODE_TYPE:` of each node in the Postgres cluster to the output of the `sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status` command.

- P - Primary
- S - Secondary

If `NODE_TYPE:` for each node in `vars.yml` does not match the output of the `sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status` command, you must edit `vars.yml`.

- 6 Compare the hostnames in the output of the `sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status` command with the hostname assignments in `inventory.ini`. If the hostnames do not match, you must edit `inventory.ini`.
- 7 Compare the deploy target assignments for each node in `inventory.ini` to the deploy target assignments for each node in the `INVOCATION_VARIABLES` section of `vars.yml`. If the deploy target assignments do not match, edit `vars.yml` to match `inventory.ini`.
- 8 If you are upgrading software from an order that was created with a mirrored repository, on the Ansible controller machine, from the `sas_viya_playbook` directory, run the following command:

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

- 9 If you are upgrading software from an order that was created with a mirrored repository, install your SAS software using the steps in [“Deploy the SAS Viya Software to the Deployment Targets” on page 194](#). Otherwise, install your SAS Viya software using the steps in the installation chapter beginning with [“Modify the vars.yml File” on page 62](#).
- 10 If you removed the CAS service that is associated with the `casenv_tenant` variable described in Step 5 of [“Prepare to Upgrade SAS Viya Software” on page 144](#), ensure that any bookmarked URLs are updated to remove that value and use `cas-shared-default-http` instead.

For example, if your original deployment contained a `casenv_tenant` value of `viya32`, you should change it from `http://host.company.com/cas-viya32-default-http` to `http://host.company.com/cas-shared-default-http`.

Note: Do not include `casenv_tenant` in your new `vars.yml`. This variable is no longer used.

- 11 If the deployment uses a custom theme, perform the following steps:

- a As a member of either the Application Administrators group or the SAS Administrators group, log on to SAS Theme Designer.

- b Select the custom theme.
 - c Click **Unpublish**.
 - d Click Publish.
 - e When the theme's status changes to Published, log off of SAS Theme Designer.
- 12** After the software has been installed, complete the following tasks, as appropriate:
- a [“Configure the Connection to the Mail Service” on page 93.](#)
 - b [“Configure SAS Viya to Connect to LDAPS Provider ” on page 94.](#)
 - c [“Configure a Symbolic Link to a Storage Platform” on page 94.](#)
 - d [“Verify That Licenses Are Applied” on page 95.](#)
 - e If your upgrade includes SAS Event Stream Processing, [“Complete SAS Event Stream Processing Setup” on page 95.](#)
 - f If your upgrade includes SAS Event Stream Manager, [“Complete SAS Event Stream Manager Setup” on page 98.](#)
 - g [“Configure High Availability in SAS Studio” on page 100.](#)
 - h If you have added any SAS/ACCESS software to your deployment as part of upgrading to SAS Viya 3.3, see the appropriate topics in [“Configure Data Access” on page 102](#) for instructions to configure the new software.
 - i [“Configure Data Quality” on page 113.](#)
 - j [Validate the Deployment on page 115.](#)
 - k [Complete the Deployment on page 131.](#)
 - l Any appendixes containing information relevant to your deployment.

- 13** To add SAS caslib ACLs, as the sas user, run the following command. Here is an example:

Note: Be sure to provide a URL that is appropriate for the system that you are performing the deployment on.

```
/opt/sas/viya/home/share/deployment/add_new_caslib_controls.sh
--sas-endpoint "http://my.sas.services.com:80"
```

- 14** The script will interactively prompt you to log on. You must log on using a profile that is a member of the SASAdministrators group. After you log on, the script will run against all CAS servers in the environment and set the new ACLs, and then exit.

- 15** (Optional) After the upgrade process has completed, record the new list of installed software.

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

```
sudo rpm -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 upgrade ([Step 1 on page 144](#)) and after the upgrade.

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

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

Preserve Access Controls for Database Caslibs

SAS Viya 3.3 includes changes to how access controls for database caslibs are stored. Path-based caslibs that use a directory as a data source are unaffected.

TIP If you have database caslibs in a SAS Viya 3.2 deployment, but you do not have access controls or you do not need to preserve the access controls, you can drop the caslib and add an identical caslib with the same name. Using the same name enables SAS Visual Analytics reports to remain valid. Perform the drop and add after the upgrade to SAS Viya 3.3.

Immediately after the upgrade to SAS Viya 3.3, the change to the access controls prevents data access with caslibs that use databases. Specifically, the server does not perform the loadTable, save, columnInfo, and fileInfo actions. Here is the error message that is displayed:

```
Caslib caslib-name is from an old release and cannot be used. Create a new caslib and copy
the access controls to it.
```

To preserve existing access controls, perform the following steps for each database caslib after SAS Viya 3.3 has been deployed and SAS Cloud Analytic Services (CAS) has been started with the new release.

- 1 Temporarily add a caslib with the same data source as the original caslib. Use a temporary name such as OdbclibNew. Use the same database server, port, schema, and so on, as applicable for the data source.
- 2 From a SAS session that is running SAS Viya 3.3 or SAS 9.4M5, run the copyObjects action. In this example, odbclib is the existing caslib and temp-odbclib is the new temporary caslib.

CAUTION! The copyObjects action is a restricted action that is designed for this specific purpose. Do not attempt to use it for any other purpose elsewhere.

```
cas;
proc cas;
  accessControl.assumeRole / adminRole="data";
run;

  accessControl.copyObjects /
    fromObjectSelector={caslib="odbclib" objType="caslib"}
    toObjectSelector={caslib="temp-odbclib" objType="caslib"};
run;
```

- 3 Drop the original caslib. In the example, odbclib is the caslib.
- 4 Add a caslib that is identical to the original caslib. Use the same name as the original caslib and the same data source information.
- 5 Restore the access controls from the temporary caslib to the newly created caslib:

```
cas;
proc cas;
  accessControl.copyObjects /
    fromObjectSelector={caslib="temp-odbclib" objType="caslib"}
    toObjectSelector={caslib="odbclib" objType="caslib"};
run;
```

```
    accessControl.dropRole / adminRole="data";
run;
```

6 Drop the temporary caslib.

Generate a New Ansible Playbook

If updates are needed in the Ansible playbook, to generate and apply a new Ansible playbook for your deployment:

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

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

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

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

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

2 Use the Software Order Email (SOE) for your original deployment to download the current version of the SAS Orchestration CLI.

3 Using the SAS Orchestration CLI that you downloaded, create a new playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site.

4 You must extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to `/sas/install/`, you might extract the new playbook to `/sas/upgrade/` instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:

- To preserve the original vars.yml file and the inventory file.
- To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following:

```
tar xf SAS_Viya_playbook.tgz -C /sas/upgrade/
```

5 Merge the vars.yml file and the inventory file from the previous deployment into the new playbook.

a Compare the two vars.yml files, and compare the two inventory files since there could be additions or changes in the newer set of files.

```
diff /sas/install/sas_viya_playbook/vars.yml /sas/upgrade/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory-file /sas/upgrade/sas_viya_playbook/inventory.ini
```

b If the new files contain new content, then merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.

- c If the original vars.yml file from the deployment that is being upgraded contains a value for the casenv_tenant variable, it must be removed. Run the following commands to remove the registered CAS service.

```
cd /opt/sas/viya/home/bin
./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/c
lient.token \
agent service deregister \
"cas-{casenv_tenant}-default-http"

./sas-bootstrap-config \
--token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/c
lient.token \
agent service deregister \
"cas-{casenv_tenant}-default"
```

- d If you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.
- 6 If you have deployed SAS Event Stream Processing or SAS Event Stream Manager, perform the following steps:

- a Stop the SAS Event Stream Processing Studio (esvvm) service.

Run the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-espvm-default
```

- b If you have installed Streamviewer, find its process ID so that you can kill the Streamviewer service:

```
ps -ef
```

Kill the Streamviewer process, substituting the process ID that was returned in the previous step:

```
kill -9 process-ID
```

- c Stop the Metering Server (SAS Event Stream Processing 5.1 and later only):

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

- 7 To apply the new Ansible playbook, change to the directory where the new playbook is located:

```
cd /sas/upgrade/
```

Run the following command:

```
ansible-playbook site.yml
```

- 8 If you removed the CAS service that is associated with a casenv_tenant variable (described in Step 3), ensure that any bookmarked URLs are updated to remove that value and use **cas-shared-default-http** instead. For example, if your original deployment contained a casenv_tenant value of viya32, you should change it from `http://host.company.com/cas-viya32-default-http` to `http://host.company.com/cas-shared-default-http`.

Note: Do not include casenv_tenant in your new vars.yml. This variable is no longer used.

- 9 (Optional) After the process has completed, record the new list of installed software.

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

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

On each machine in your deployment, create a file that lists the SAS 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 performing this task ([Step 1 on page 151](#)) and after.

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

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


Uninstalling SAS Viya

<i>Overview</i>	155
<i>What deploy-cleanup Does</i>	155
<i>Create a Backup for SAS Event Stream Processing</i>	156
<i>Uninstall Command</i>	156
<i>Uninstall SAS Embedded Process</i>	157
<i>Uninstall SASHDAT Plug-ins</i>	157

Overview

This section describes how to uninstall SAS Viya software if it was deployed using Ansible. For information about uninstalling yum deployments, see [“Uninstall SAS Viya with Yum” on page 186](#).

What deploy-cleanup Does

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

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

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

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

The `deploy-cleanup` command renames the `/opt/sas/viya` directory to `/opt/sas/viya_epoch`. Also, the `/opt/sas/spre` directory is renamed as `/opt/sas/spre_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.

Create a Backup for SAS Event Stream Processing

Note: If your software order does not include SAS Event Stream Processing, you should skip this section.

Before you run `deploy-cleanup.yml` to uninstall SAS Viya, create a backup copy of the SAS Event Stream Processing Studio database in order to preserve project files. Follow these steps:

- 1 Stop the SAS Event Stream Processing Studio (esvvm) service by running the following command on Red Hat Enterprise Linux 6.x:

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

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-esvvm-default
```

- 2 Create a backup copy of the database, which is a single binary file (`studio.mv.db`). You can copy it to any directory location outside the SAS Event Stream Processing installation directory structure.

The location and filename of the database are determined by the environment variable `ESP_STUDIO_DB`. By default, it is stored in `/opt/sas/viya/config/data/esvvm/`.

To create the backup, run the following command:

```
cp studio.mv.db directory-name
```

- 3 (Optional) If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://hostname:http_port/exit"
```

Replace *hostname* with the host name of the server where the Streamviewer files are installed and running.

Replace *http_port* with the port number that you provided when you started Streamviewer with the startup script.

For more information, see [Starting Streamviewer](#).

Uninstall Command

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

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

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

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

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

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

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

```
yum remove httpd
```

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

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

```
ps -A
```

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

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

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

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

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

```
ps -A
```

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

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

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

Uninstall SAS Embedded Process

If your software deployment includes SAS Embedded Process, uninstall it using the instructions at [“Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya” on page 229](#).

Uninstall SASHDAT Plug-ins

If your software deployment includes CAS SASHDAT Access to HDFS, uninstall it using the instructions at [“Uninstalling SAS Plug-ins for Hadoop” on page 242](#).

Appendix 1

Creating High Availability PostgreSQL Clusters

<i>Overview</i>	159
<i>HA PostgreSQL Topologies</i>	160
<i>Set Up a Horizontal Cluster</i>	161
Edit the inventory.ini File	161
Edit the vars.yml File	162
<i>Set Up a Vertical Cluster</i>	163
Edit the inventory.ini File	163
Edit the vars.yml File	163
<i>Set Up a Hybrid Cluster</i>	164
Edit the inventory.ini File	164
Edit the vars.yml File	164
<i>Set Up Multiple Clusters</i>	165
Modify inventory.ini and vars.yml Files	165
Configure Services to the Clusters	168
<i>Deployment Logs</i>	168
<i>Verify the Deployment</i>	169

Overview

Note: If your software order contains products from the SAS Event Stream Processing product family only, you can skip this section. You should still perform the tasks in this section for products that are not contained in the SAS Event Stream Processing product family. For a description of the SAS Event Stream Processing product family, see [“About Deploying SAS Event Stream Processing Products Only” on page 2](#).

SAS Viya uses High Availability (HA) PostgreSQL as the SAS Infrastructure Data Server. By default, when you use the instructions in [“Installation” on page 55](#), Ansible deploys HA PostgreSQL as a single node on a single machine. However, HA PostgreSQL supports other topologies. This appendix describes those topologies and explains how to use Ansible to deploy them.

HA PostgreSQL Topologies

The standard PostgreSQL deployment with SAS Viya consists of one PGPool and one PostgreSQL data node. All data connection and database requests are routed through PGPool. You connect to PGPool just as you would connect to PostgreSQL, using standard database connectors. With SAS Viya we also have the ability to deploy High Availability PostgreSQL, a clustered database containing one PGPool and one or more data nodes. One data node is designated as a primary and all others are standby nodes. Replication happens in real time to keep the data nodes in sync. All write requests are routed to the primary data node by PGPool; read requests can be distributed across all data nodes, allowing for higher performance. In the event that the primary data node is lost, PGPool will automatically promote a standby node to primary and reestablish replication from the new primary to the remaining standby data nodes.

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

A cluster can be deployed in four possible configurations:

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

For multi-node deployments, PGPool node can be colocated with data nodes or deployed on its own machine. Note that colocating nodes on a machine provides increased read throughput but also increases the risk of node loss should that machine become unavailable.

The following table demonstrates how nodes can be distributed in the multi-node topologies.

Cluster Configuration	Server	Port	Role
Horizontal	Server 1	5432	Primary
	Server 2	5432	Standby
	Server 3	5432	Standby
	Server 4	5432	Standby
Vertical	Server 1	5532	Primary
	Server 1	5533	Standby
	Server 1	5534	Standby
	Server 1	5535	Standby
Hybrid	Server 1	5632	Primary

Cluster Configuration	Server	Port	Role
	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 `sasdatasvc` 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

sasdatasvc

- `NODE_NUMBER`: the sequential node identifier starting at 0
- `NODE_TYPE`: P for primary or S for standby. There can be only one primary per cluster.>
- `PG_PORT`: The PostgreSQL database port. PGPool talks to the database on this port. Clients use the `PGPOOL_PORT`.
- `SANMOUNT`: the location where the data files will be placed
- `SERVICE_NAME`: the unique name that you assign to your cluster

Set Up a Horizontal Cluster

Edit the `inventory.ini` File

Modify the `inventory.ini` file 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 57](#). Then assign the machines to the host groups as described at [“Assign the Target Machines to Host Groups” on page 58](#). 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 `[sasdatasvc]`.

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 85](#) 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 57](#). Then assign the machines to the host groups as described at [“Assign the Target Machines to Host Groups” on page 58](#). 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]
deloydTarget1
'''
[sasdatasvrc]
deloydTarget1
...

```

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 85](#) 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 57](#). Then assign the machines to the host groups as described at [“Assign the Target Machines to Host Groups” on page 58](#). 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 sasdatasvc nodes. This is what that section would look like when filled out for our example:

```
# Multiple invocation definitions
INVOCATION_VARIABLES:
  deployTarget1:
    pgpoolc:
      - PCP_PORT: '5631'
        PGPOOL_PORT: '5630'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
        SERVICE_NAME: postgres
    sasdatasvc:
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5632'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
        SERVICE_NAME: postgres
      - NODE_NUMBER: '1'
        NODE_TYPE: S
        PG_PORT: '5633'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
        SERVICE_NAME: postgres
  deployTarget2:
    sasdatasvc:
      - NODE_NUMBER: '2'
        NODE_TYPE: S
        PG_PORT: '5632'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
        SERVICE_NAME: postgres
      - NODE_NUMBER: '3'
        NODE_TYPE: S
        PG_PORT: '5633'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
        SERVICE_NAME: postgres
```

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

After you save the vars.yml file and you complete the other deployment steps, use the commands described at [“Deploy the Software” on page 85](#) to deploy your SAS Viya software, including HA PostgreSQL.

Set Up Multiple Clusters

Modify inventory.ini and vars.yml Files

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

- a single-node cluster with pgpoolc and sasdatasvc on a machine named deployTarget1
- a horizontal cluster with pgpoolc on deployTarget1 and a sasdatasvc node on each machine
- a vertical cluster with pgpoolc on deployTarget3 and all the sasdatasvc nodes on deployTarget4
- a hybrid cluster with pgpoolc on deployTarget1, two sasdatasvc nodes on deployTarget2, and two more sasdatasvc nodes on deploytarget3

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

```

deployTarget1 ansible_host=host.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ssh/id_rsa
deployTarget2 ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ssh/id_rsa
deploytarget3 ansible_host=host3.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ssh/id_rsa
deploytarget4 ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/ssh/id_rsa
...
[pgpoolc]
deployTarget1
deployTarget3
deployTarget4
...
[sasdatasvrc]
deployTarget1
deployTarget2
deployTarget3
deployTarget4
...

```

This is how the INVOCATION_VARIABLES section of the vars.yml file would be filled out:

```

# Multiple invocation definitions
INVOCATION_VARIABLES:
  deployTarget1:
    pgpoolc:
      - PCP_PORT: '5431'
        PGPOOL_PORT: '5430'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_hybrid
      - PCP_PORT: '5461'
        PGPOOL_PORT: '5460'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
    sasdatasvrc:
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5452'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_horizontal
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5462'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres
  deployTarget2:
    sasdatasvrc:
      - NODE_NUMBER: '0'
        NODE_TYPE: P
        PG_PORT: '5432'
        SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
        SERVICE_NAME: postgres_hybrid
      - NODE_NUMBER: '2'
        NODE_TYPE: S

```

```

    PG_PORT: '5433'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_hybrid
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5452'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_horizontal
deployTarget3:
  pgpoolc:
  - PCP_PORT: '5441'
    PGPOOL_PORT: '5440'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_vertical
  sasdatasvc:
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5432'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_hybrid
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5433'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_hybrid
  - NODE_NUMBER: '2'
    NODE_TYPE: S
    PG_PORT: '5452'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_horizontal
deployTarget4:
  pgpoolc:
  - PCP_PORT: '5451'
    PGPOOL_PORT: '5450'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_horizontal
  sasdatasvc:
  - NODE_NUMBER: '0'
    NODE_TYPE: P
    PG_PORT: '5442'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '1'
    NODE_TYPE: S
    PG_PORT: '5443'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '2'
    NODE_TYPE: S
    PG_PORT: '5444'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'
    SERVICE_NAME: postgres_vertical
  - NODE_NUMBER: '3'
    NODE_TYPE: S
    PG_PORT: '5445'
    SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvc'

```

```

SERVICE_NAME: postgres_vertical
- NODE_NUMBER: '3'
NODE_TYPE: S
PG_PORT: '5452'
SANMOUNT: '{{ SAS_CONFIG_ROOT }}/data/sasdatasvrc'
SERVICE_NAME: postgres_horizontal

```

Note: If you are deploying multiple clusters, one of the PG Pools must be named `postgres`, and each PG Pool name must be unique across clusters. In addition, each cluster must contain one `sasdatasvrc` node with a `NODE_TYPE` of `P`.

Configure Services to the Clusters

By default, all microservices connect to the HA Postgres cluster that is named `postgres`. You can configure individual services to use additional HA Postgres clusters (if they exist) by adding service-specific sections to the `sitedefault.yml` file.

- 1 If you have not already copied and renamed the `sitedefault.yml` file, locate the `sitedefault_sample.yml` file on the Ansible controller machine. If you used the recommended location for uncompressing your playbook, the file is located at `/sas/install/sas_viya_playbook/roles/consul/files/sitedefault_sample.yml`. Make a copy of `sitedefault_sample.yml` and name the copy `sitedefault.yml`.
- 2 Open the `sitedefault.yml` file.
- 3 At the end of the existing file and at the same indentation level as `application`, add the following content:

```

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

```

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

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

```

config:
  application:
  ...
  authorization:
    sas:
      database:
        databaseServerName: postgres-horizontal
        spring.datasource.password: ${sas.database.postgres-horizontal.password}

```

- 4 Save and close the `sitedefault.yml` file.

Deployment Logs

Each PG Pool node and HA PostgreSQL data node has its own set of directories for logging. The logs for PG Pool are located at

```
/opt/sas/viya/config/var/log/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 multi-node deployment.

The verification log is called `sds_status_check_date-timestamp.log`. It can be found in the `pgpool` log folder of each cluster. The fastest way to determine whether your HA PostgreSQL deployment was successful is to read the verification log.

Appendix 2

Deploying with Yum

<i>Overview</i>	172
<i>Run the Deployment Script</i>	172
<i>Deploy httpd and MOD_SSL</i>	173
<i>Set Up the CAS Administrator</i>	173
<i>Set Up the CAS Controller to Run as a Service</i>	174
<i>Start the Services</i>	174
<i>Configure SAS/ACCESS Interface to Amazon Redshift</i>	174
<i>Configure SAS/ACCESS Interface to DB2</i>	175
<i>Configure SAS/ACCESS Interface to Greenplum</i>	176
<i>Configure SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop</i>	176
<i>Configure SAS/ACCESS Interface to HAWQ</i>	177
<i>Configure SAS/ACCESS Interface to Impala</i>	178
<i>Configure SAS/ACCESS Interface to Microsoft SQL</i>	179
<i>Configure SAS/ACCESS Interface to MySQL</i>	179
<i>Configure SAS/ACCESS Interface to Netezza</i>	180
<i>Configure SAS/ACCESS Interface to ODBC</i>	180
<i>Configure SAS/ACCESS Interface to Oracle</i>	181
<i>Configure SAS/ACCESS Interface to PostgreSQL</i>	182
<i>Configure SAS/ACCESS Interface to SAP HANA</i>	183
<i>Configure SAS/ACCESS Interface to SAP R/3</i>	183
<i>Configure SAS/ACCESS Interface to Teradata</i>	184
<i>Configure Settings for SAS Event Stream Processing for CAS</i>	185
<i>Install Sample SAS Data Sets</i>	185
<i>Log On to SAS Studio</i>	185
<i>View Deployment Logs</i>	185
<i>Validate the Installation</i>	186
<i>Next Steps</i>	186
<i>Uninstall SAS Viya with Yum</i>	186

Overview

Use this appendix for instructions to deploy only the programming interface of your SAS Viya software on a single machine.

Note: SAS Event Stream Processing and SAS Event Stream Manager cannot be deployed by the process described in this appendix.

Run the Deployment Script

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

If you moved the certificates, open the `customized_deployment_script.sh` file that was included in the playbook that you generated. Use a text editor to specify the directory path that contains the certificates. Here is an example:

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

- 2 Save and close the `customized_deployment_script.sh` file.
- 3 If you are installing SAS Viya on a machine that is already running SAS 9.4 software, determine whether required ports are available by running the following commands:

- SAS Object Spawner:

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

- SAS/CONNECT:

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

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

- 4 Run the script:

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

- 5 Run the following command:

Note: For improved readability, the command occupies several lines. It should be run as a single line.

```
export SASPREHOME=/opt/sas/spre/home ; for file in $(ls -1 ${SASPREHOME}/SASFoundation
/install/install.d/); do su - -s /bin/bash -c "${SASPREHOME}/SASFoundation/utilities/bin/post_install
$(basename $file | cut -d "." -f1)" sas ; done
```

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

- SAS Object Spawner:

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

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

- SAS/CONNECT:

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

Deploy httpd and MOD_SSL

- 1 Run the command to deploy httpd and MOD_SSL:

```
sudo yum install httpd mod_ssl
```

- 2 Create the proxy.conf file:

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

- 3 Copy and paste the following content into the proxy.conf file:

Note: Substitute the appropriate hosts.

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

- 4 Save and close the proxy.conf file.

- 5 Start the httpd service.

- For Red Hat Enterprise Linux 6.7:

```
sudo service httpd start
```

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl restart httpd.service
```

Set Up the CAS Administrator

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

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

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

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

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

Here is an example of the replaced values:

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

- 3 Save and close the perms.xml file

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

```
sudo passwd cas
```

Set Up the CAS Controller to Run as a Service

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

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

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

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

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

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

- 3 Add the new service with the following command:

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

Start the Services

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

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

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.0 and later:

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

Configure SAS/ACCESS Interface to Amazon Redshift

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Amazon Redshift.

To configure SAS/ACCESS Interface to Amazon Redshift:

- 1 To reference a Data Source Name (DSN) in your connection, add the DSN to the `odbc.ini` file.
 - a Edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition
 - b 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 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 3 Add the following lines:

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

- 4 Save and close the `sasenv_deployment` file.

- 5 Open the `cas_usermods.settings` file.

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

- 6 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
```

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

Configure SAS/ACCESS Interface to DB2

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to DB2.

To configure SAS/ACCESS Interface to DB2:

- 1 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 2 Add the following lines:

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

- 3 Save and close the `sasenv_deployment` file.

- 4 Using a text editor, open the `cas_usermods.settings` file.

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

- 5 Add the following lines:

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

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

Configure SAS/ACCESS Interface to Greenplum

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Greenplum.

To configure SAS/ACCESS Interface to Greenplum:

- 1 To reference a Data Source Name (DSN) in your connection, add the DSN to the `odbc.ini` file.
 - a Edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition
 - b Edit the `/opt/sas/viya/home/lib64/accessclients/odbc.ini` and add your DSN definition.

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

- 2 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 3 Add the following lines:

Note: Depending on how you have configured your ODBC driver, you might need to specify the `odbc.ini` file, the `odbcinst.ini` file, or both files. The following examples include both files.

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

For bulk loading, add the following lines.

```
export GPHOME_LOADERS=Greenplum-bulk-loader-installation-location
export GPLOAD_HOME=Greenplum-installation-location
export GPLOAD_PORT=Greenplum-bulk-load-port
```

- 4 Save and close the `sasenv_deployment` file.

Configure SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Hadoop or SAS In-Database Technologies for Hadoop.

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

- 1 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 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 `sasenv_deployment` file.

- 5 Use a text editor to edit the `cas_usermods.settings` file:

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

- 6 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.

- 7 If you are using MapR, add the following line:

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

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

Configure SAS/ACCESS Interface to HAWQ

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to HAWQ.

To configure SAS/ACCESS Interface to HAWQ:

- 1 To reference a Data Source Name (DSN) in your connection, add the DSN to the `odbc.ini` file.
 - a Edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition
 - b Edit the `/opt/sas/viya/home/lib64/accessclients/odbc.ini` and add your DSN definition.

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

- 2 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 3 Add the following lines:

Note: Depending on how you have configured your ODBC driver, you might need to specify the `odbc.ini` file, the `odbcinst.ini` file, or both files. The following examples include both files.

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

- 4 Save and close the `sasenv_deployment` file.

Configure SAS/ACCESS Interface to Impala

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Impala.

To configure SAS/ACCESS Interface to Impala:

- 1 Install a third-party ODBC Driver Manager. The Impala ODBC driver is an ODBC API-compliant shared library. In addition, the Impala ODBC driver requires that you also install a third-party ODBC Driver Manager. A version of the unixODBC Driver Manager is available for download from the unixODBC website <http://www.unixodbc.org/download.html>.

- 2 To enable the Impala driver to be loaded dynamically at run time, include the full pathname of the shared library in the shared library path.

- 3 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 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 ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
export CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
export EASYSOFT_UNICODE=YES
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
/opt/cloudera/impalaodbc/lib/64
```

Note: The `EASYSOFT_UNICODE` variable should only be added if you want to set the encoding for the SAS client to UTF-8.

- 5 Save and close the `sasenv_deployment` file.

- 6 Open the `cas_usermods.settings` file.

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

- 7 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 ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbcinst.ini-file-including-file-name
export CLOUDERAIMPALAODBC=location-of-your-cloudera.impalaodbc.ini-file
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
/opt/cloudera/impalaodbc/lib/64
```

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

- 9 To use an Impala ODBC driver from a different vendor than SAS/ACCESS Interface to Impala on SAS Viya, set either the `SAS_IMPALA_DRIVER_VENDOR` environment variable or the `DRIVER_VENDOR` connection option. Here are some examples:

- Set the environment variable to use the MapR Impala ODBC driver:

```
SAS_IMPALA_DRIVER_VENDOR=MAPR
export SAS_IMPALA_DRIVER_VENDOR
```


- When defining the caslib, set the DRIVER_VENDOR variable to use the Progress DataDirect Impala ODBC driver:

```
action addCaslib lib="datalib" datasource={srctype="impala", server="impserver", schema="default",
DRIVER_VENDOR="DATADIRECT"} ; run
```

Currently, the only valid values for the driver vendor are DATADIRECT and MAPR.

Configure SAS/ACCESS Interface to Microsoft SQL

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Microsoft SQL.

To configure SAS/ACCESS Interface to Microsoft SQL:

- 1 To reference a Data Source Name (DSN) in your connection, add the DSN to the odbc.ini file.
 - a Edit the `/opt/sas/spre/home/lib64/accessclients/odbc.ini` file and add your DSN definition
 - b 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.

- 2 On the SAS client node, use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 3 Add the following lines:

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

- 4 Save and close the `cas_usermods.settings` file.
- 5 On the CAS node(s), use a text editor to edit the `cas_usermods.settings` file:

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

- 6 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
```

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

Configure SAS/ACCESS Interface to MySQL

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to MySQL.

To configure SAS/ACCESS Interface to MySQL:

- 1 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 2 Add the following line:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-MySQL-client-library
```

- 3 Save and close the `sasenv_deployment` file.

Configure SAS/ACCESS Interface to Netezza

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to Netezza.

To configure SAS/ACCESS Interface to Netezza:

- 1 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 2 Add the following lines:

```
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
export ODBCINST=location-of-your-odbc.ini-file-including-file-name
export NZ_ODBC_INI_PATH=path-to-the-Netezza-configuration-files
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-Netezza-client
```

- 3 Save and close the `sasenv_deployment` file.

Configure SAS/ACCESS Interface to ODBC

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to ODBC.

To configure SAS/ACCESS Interface to ODBC:

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

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

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

- 2 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

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

For DataDirect:

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

For iODBC:

```
export ODBCINI=location-of-your-odbc.ini-file-including-file-name
```

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

For unixODBC:

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

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

- 4 Save and close the sasenv_deployment file.
- 5 Use a text editor to edit the cas_usermods.settings file:

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

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

For DataDirect:

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

For iODBC:

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

For unixODBC:

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

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

- 7 Save and close the cas_usermods.settings file.

Configure SAS/ACCESS Interface to Oracle

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

To configure SAS/ACCESS Interface to Oracle:

- 1 Use a text editor to edit the sasenv_deployment file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 2 Add the following lines:

```
export ORACLE_HOME=Oracle-home-directory
export TWO_TASK=ORACLE_SID
export ORAENV_ASK=NO
export SASORA=V9
```

```
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

- 3 Save and close the `sasenv_deployment` file.
- 4 Use a text editor to edit the `cas_usermods.settings` file:

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

- 5 Add the following lines:

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

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

Configure SAS/ACCESS Interface to PostgreSQL

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to PostgreSQL. Use the following steps to configure SAS/ACCESS Interface to PostgreSQL.

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
```

- 1 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
```

- 2 Add the following lines:

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

For bulk loading, add the following line:

```
export PATH=$PATH:path-to-PostgreSQL-bulk-loading
```

3 Save and close the `sasenv_deployment` file.

4 Use a text editor to edit the `cas_usermods.settings` file:

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

5 Add the following lines:

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

6 Save and close the `cas_usermods.settings` file.

Configure SAS/ACCESS Interface to SAP HANA

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to SAP HANA.

To configure SAS/ACCESS Interface to SAP HANA:

1 Use a text editor to edit the `sasenv_deployment` file: .

```
sudo vi /opt/sas/<viya/config/etc/workspaceserver/default/sasenv_deployment
```

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 `sasenv_deployment` file.

4 Use a text editor to edit the `cas_usermods.settings` file:

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

5 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
```

6 Save and close the `cas_usermods.settings` file.

Configure SAS/ACCESS Interface to SAP R/3

Note: The information in this section is applicable only if you ordered SAS/ACCESS Interface to SAP R/3.

To configure SAS/ACCESS Interface to SAP R/3:

1 Use a text editor to edit the `sasenv_deployment` file: .

```
sudo vi /opt/sas/viya/config/etc/workspaceserver/default/sasenv_deployment
:
```

2 Add the following lines:

```
export RFC_INI=path-to-the-SAP-R/3-ini-file
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path-to-the-SAP-R/3-client
```

- 3 Save and close the `sasenv_deployment` file.

Configure SAS/ACCESS Interface to Teradata

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

To configure SAS/ACCESS Interface to Teradata:

- 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 Use a text editor to edit the `sasenv_deployment` file:

```
sudo vi /opt/sas/<tenant>/config/etc/workspaceserver/default/sasenv_deployment
```

- 4 Add the following lines:

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

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

An example of the TTU Default `LD_LIBRARY_PATH` is

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

- 5 Save and close the `sasenv_deployment` file.
- 6 Use a text editor to edit the `cas_usermods.settings` file:

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

- 7 Add the following lines:

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

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

An example of the TTU Default `LD_LIBRARY_PATH` is

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

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

Configure Settings for SAS Event Stream Processing for CAS

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

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

- 1 On the CAS controller, navigate to the `/opt/sas/viya/config/etc/cas/default` directory on the CAS controller.
- 2 Use your preferred text editor to modify the `cas.settings` file. Add the following lines:

```
export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/5.1.0
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DFESP_HOME/lib
export TKPATH=$TKPATH:$DFESP_HOME/lib/tk
```
- 3 Save and close the `cas.settings` file.

Install Sample SAS Data Sets

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

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

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

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

Log On to SAS Studio

Perform the following steps to log on:

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

```
https://webserver-host-name/SASStudio
```
- 2 Log on using the credentials for your operating system account.

Note: To log off from SAS Studio, click **Sign Out** on the toolbar. Do not use the **Back** button on your web browser.

View Deployment Logs

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

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

Validate the Installation

After you complete the procedures in this appendix, you should validate the installation. For details, see [“Validating the Deployment” on page 115](#).

Next Steps

After you validate your software, see [“Completing the Deployment” on page 131](#) for information about using your software.

Uninstall SAS Viya with Yum

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

- 1 Stop all the services.

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

- 2 Remove the cascontroller service with the following commands:

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

- 3 Remove the products by following these steps:

- a Open the `customized_deployment_script.sh` file that was included in the playbook, which you saved from the Software Order Email (SOE).
- b To obtain the list of products to remove, locate the yum groupinstall command in the shell script file. Here is an example:

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

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

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

- 4 Remove the repositories by following these steps:

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

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

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

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



```
"sas-statviya-8.1.0-rpm-latest"
```

- 5** Remove the main repository definition with the following command:

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

- 6** Remove any remaining components with the following command:

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

- 7** Remove the entitlement certificate with the following command:

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

- 8** Rename the **viya** and **spre** directories with the following commands:

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

```
sudo mv /opt/sas/spre/ /opt/sas/spre_$(date +%s)
```

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

- 9** Close the `customized_deployment_script.sh` file.

Appendix 3

Creating and Using Mirror Repositories

Overview	189
Requirements	190
Ansible Controller	190
Connected Repository Mirror	190
Unconnected Repository Mirror	190
Deployment Targets	191
Machine Combinations	191
Use Ansible to Create a Mirror Repository	191
Confirm That Ansible Is Installed on the Ansible Controller	191
Confirm the Identities of the Hosts	192
Prepare the rephosts Inventory File	192
Confirm Network Connectivity and Ansible Accessibility	192
(Optional) Install and Enable Apache httpd	193
Create the Mirror Repository	193
Confirm HTTP Connectivity to the Mirror Repository	194
Deploy the SAS Viya Software to the Deployment Targets	194
Use Yum to Create Mirror Repositories	194
Prepare the Connected Mirror Repository	194
Create and Use the createrepos.sh File	195
Prepare the Unconnected Mirror Repository	195
Create the repo.conf File	196
Restart the httpd Service	196
Confirm HTTP Connectivity to the Mirror Repository	197
Install from the Repository Using Ansible	197
Install from the Repository Using Yum	199
Creating a Local Copy of Documentation	201
Uninstalling SAS Viya from Mirrored Repositories	202
Uninstall the Repositories	202
Uninstall from the Deployment Target	202

Overview

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

Requirements

The instructions in this appendix assume a topology that consists of one or more machines that perform these roles: an Ansible controller, a mirror repository host connected to the internet, a mirror repository host that is not connected to the internet, and deployment targets. All machines described in this chapter must meet the operating system requirements described in “[Operating System Requirements](#)” on page 23. The following topics describe each type of machine and additional requirements.

Ansible Controller

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

- does not require internet access.
- requires network connectivity and Ansible accessibility to itself, as well as to the connected repository mirror, the unconnected repository mirror, and the deployment target machines.
- must have Ansible installed. For information about supported Ansible versions, see <https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-operating-systems.html>.
- must be capable of controlling itself through Ansible.

Connected Repository Mirror

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

- must have internet access.
- must be capable of control by the Ansible controller.
- has 100 GB of free disk space in `/opt/sas/repomirror` to hold a temporary archive of the repository mirror files.

Unconnected Repository Mirror

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

- does not require internet access.
- is reachable from your deployment target machine or machines by HTTP.
- can be controlled by your Ansible controller machine.
- has 100 GB of free disk space in `/var/www/html/pulp` to hold the mirror repository files.

Deployment Targets

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

Machine Combinations

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

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

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

Use Ansible to Create a Mirror Repository

Confirm That Ansible Is Installed on the Ansible Controller

- 1 Run the following command on the Ansible controller:

```
ansible --version
```

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

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

- 3 If your results are different, Ansible has not been installed on the machine. To install Ansible on the machine, see [“Install Ansible” on page 48](#).

Confirm the Identities of the Hosts

Ensure that the output of the `hostname`, `hostname -f`, and the `hostname -s` command prints good and expected output.

Prepare the repohosts Inventory File

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

2 Ensure that the `repohosts` file is writable.

```
chmod +w repohosts
```

3 Open the `repohosts` file.

4 The beginning of the file contains the following lines:

```
lighthost ansible_host=<machine_address>
darkhost ansible_host=<machine_address>
```

In the first line, replace `<machine_address>` with any resolvable address, such as the IP address or the fully qualified domain name, for the machine that is the connected mirror repository. In the second line, replace `<machine_address>` with any resolvable address for the machine that is the unconnected mirror repository. If either mirror repository will be running Ansible, replace the target declaration for the appropriate machine with `ansible_connection=local`. Here is an example:

```
lighthost ansible_connection=local
```

Note: Do not use `127.0.0.1` as an IP address for any machines in the file `repohosts`.

If you add `ansible_user` information, ensure that the same user is added to both lines.

5 Save and close the `repohosts` file.

Confirm Network Connectivity and Ansible Accessibility

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

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

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

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

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

(Optional) Install and Enable Apache httpd

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

Note: The following process installs and starts httpd, but it does not change the system settings so that it will run by default. If you reboot the unconnected repository machine, you will have to restart httpd as you would any other service.

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

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

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

3 Open `repo_vars.yml`.

4 Locate the following line:

```
# setup_httpd_server: no
```

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

```
setup_httpd_server: yes
```

6 Save and close the `repo_vars.yml` file.

7 On the unconnected mirror repository machine, ensure that firewall software is not running. Use the commands in steps 3 and 4 of “[Firewall Considerations](#)” on page 44.

Create the Mirror Repository

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

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

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

- downloads SAS software RPM package files from entitled yum repositories that are hosted by SAS on the internet

- creates a file named `repo_override.txt` in `/opt/sas/repomirror` on the mirror host

Note: You can also create `repo_override.txt` on the Ansible controller by setting the `create_repo_deployment_file_on_controller` value in the `repo_vars.yml` file to `yes`.

- copies the files from the temporary location on the connected mirror repository to an Apache httpd accessible location on the unconnected mirror repository (`/var/www/html/pulp/repos` by default)

- (Option) installs and starts Apache httpd software on the unconnected mirror repository

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

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

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

Confirm HTTP Connectivity to the Mirror Repository

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

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

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

Deploy the SAS Viya Software to the Deployment Targets

Before deploying your SAS Viya software, you must complete the steps described in [“Edit the Inventory File” on page 56](#) and [“Modify the vars.yml File” on page 62](#). After those sections are completed, perform the following steps:

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

```
ansible -i inventory.ini -m ping all
```

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

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

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

- 3 Depending on the machines in your mirror topology, you may need to copy the `repo_override.txt` file to your Ansible controller machine.
- 4 On the Ansible controller machine, from the `sas_viya_playbook` directory, run the following command:

```
ansible-playbook site.yml -e "@full-path-to-repo-override-file"
```

Note: For more information about the `repo_override.txt` file, see [“Create the Mirror Repository” on page 193](#).

- 5 When the playbook has finished running, the command results should be similar to the following:

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

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

Use Yum to Create Mirror Repositories

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

Prepare the Connected Mirror Repository

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

- 2 Extract the files from `SAS_Viya_playbook.tgz`:

```
tar xf SAS_Viya_playbook.tgz
```

Create and Use the `createrepos.sh` File

- 1 Go to the `sas_viya_playbook` directory on the connected mirror repository.
- 2 Using a text editor, create a new file named `createrepos.sh` that contains the following content:

```
#!/bin/bash

sudo yum install yum-utils

cp customized_deployment_script.sh setup_repos.sh

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

./setup_repos.sh

MIRRORLOC=/opt/sas/repomirror
if [ ! -d ${MIRRORLOC} ]; then
    mkdir -p ${MIRRORLOC}
fi

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

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

- 3 Save and close `createrepos.sh`.
- 4 Set the Execute bit for `createrepos.sh`:
- 5 Run `createrepos.sh` to extract the contents of the SAS repositories and to create a tar ball:

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

Prepare the Unconnected Mirror Repository

- 1 Move the `soe_defaults.yml` file from `sas_viya_playbook/internal` to the `sas_viya_playbook` directory.

```
cp sas_viya_playbook/internal/soe_defaults.yml sas_viya_playbook/soe_defaults.yml
```

- 2 Copy the tar file that was created from the repository synchronization and the `soe_defaults.yml` from the connected mirror repository to the unconnected mirror repository. Here is an example:

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

```
scp /opt/sas/repomirror/repomirror.tar.gz soe_defaults.yml user@darkhost:/tmp
```

- 3 On the unconnected mirror repository, go to the tmp directory.
- 4 Use a text editor to create a new file named yumrepocreation.sh that contains the following content:

```
#!/bin/bash

sudo yum install yum-utils createrepo httpd

REPOLOC=/var/www/html/pulp/repos
ORDERABLE=$(grep METAREPO_SOE_ORDERABLE soe_defaults.yml | awk -F"'" '{ print $2 }')
# Make the directory that will house the yum repository
if [ ! -d ${REPOLOC} ]; then
    mkdir -p ${REPOLOC}
fi

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

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

- 5 Save and close the yumrepocreation.sh file.
- 6 Set the Execute bit for the yumrepocreation.sh file:

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

- 7 Run the yumrepocreation.sh file.

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

Create the repo.conf File

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

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

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

Restart the httpd Service

Restart or reload the httpd service as needed.

- 1 Check the status of the httpd service.
 - For Red Hat Enterprise Linux 6.7:

```
sudo service httpd status
```

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl status httpd.service
```

2 If httpd is already running, reload it.

- For Red Hat Enterprise Linux 6.7:

```
sudo service httpd reload
```

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl reload httpd.service
```

3 If httpd is not running, start it.

- For Red Hat Enterprise Linux 6.7:

```
sudo service httpd start
```

- For Red Hat Enterprise Linux 7.0 and later:

```
sudo systemctl start httpd.service
```

Confirm HTTP Connectivity to the Mirror Repository

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

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

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

Note: You might need to change your firewall software configuration on the unconnected mirror repository machine in order for the curl command to succeed. Another option is to temporarily stop the firewall software on the unconnected mirror repository machine using the commands in steps 3 and 4 of [“Firewall Considerations” on page 44](#).

Install from the Repository Using Ansible

Confirm that Ansible Is Installed on the Ansible Controller

- 1 Run the following command on the Ansible controller:

```
ansible --version
```

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

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

- 3 If your results are different, Ansible has not been installed on the machine. To install Ansible on the machine, see [“Install Ansible” on page 48](#).

Confirm the Identities of the Hosts

Ensure that the output of the hostname, hostname -f, and the hostname -s command prints good and expected output.

Prepare the rephosts Inventory File

- 1 On the Ansible controller machine, locate the rephosts file in the directory where you uncompressed the SAS_Viya_playbook.tgz file. If you followed the suggestions in this guide, that file is located at `/sas/install/sas_viya_playbook/utility/rephosts`.
- 2 Open the rephosts file.
- 3 The beginning of the file contains the following lines:

```
lighthost ansible_host=<machine_address>
darkhost ansible_host=<machine_address>
```

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

Note: Do not use 127.0.0.1 as an IP address for any machines in the file rephosts.

- 4 Save and close the rephosts file.

Confirm Network Connectivity and Ansible Accessibility

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

```
ansible -i utility/rephosts -m ping all
```

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

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

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

Install and Enable Apache httpd (Optional)

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

Note: The following process installs and starts httpd, but it does not change the system settings so that it will run by default. If you reboot the unconnected repository machine, you will have to restart httpd as you would any other service.

- 1 On the Ansible controller machine, locate the `repo_vars.yml` file in the `/sas_viya_playbook/utility` directory.
- 2 Run the following command to ensure that the file is writeable:

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

3 Open `repo_vars.yml`.

4 Locate the following line:

```
# setup_httpd_server: no
```

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

```
setup_httpd_server: yes
```

6 Save and close the `repo_vars.yml` file.

7 On the unconnected mirror repository machine, ensure that firewall software is not running. Use the commands in steps 3 and 4 of [“Firewall Considerations” on page 44](#).

Deploy the SAS Viya Software

Before deploying your SAS Viya software, you must complete the steps described in [“Edit the Inventory File” on page 56](#) and [“Modify the vars.yml File” on page 62](#). After those sections are completed, perform the following steps:

1 Run the main deployment pass with the `site.yml` playbook.

```
ansible-playbook site.yml -e "@/opt/sas/repomirror/repo_override.txt"
```

By default, the `repo_override.txt` file is placed in `/opt/sas/repomirror`. To change the default location, modify the value for the `mirror_loc` variable in the `repo_vars.yml` file.

2 When the playbook finishes, the output should look similar to this:

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

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

Install from the Repository Using Yum

Create the `sas-manual.repo` File

1 In the `sas_viya_playbook` directory on the connected mirror repository, create a new file named `createrepodefn.sh` that contains the following content:

```
#!/bin/bash

REPOURI="http://xxx.xxx.xxx.xxx"
ORDERABLE=$(grep METAREPO_SOE_ORDERABLE soe_defaults.yml | awk -F"|" '{ print $2 }')

for repo in ${ORDERABLE}; do
    NAME=$(sed -e 's/^ //' -e 's/"$//' <<<"$repo")
    cat << EOL >> sas-manual.repo
[${NAME}]
name=${NAME}
baseurl=${REPOURI}/pulp/repos/${NAME}/
enabled=1
sslverify=0
sslcert=
sslclientcert=
```

```
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x
```

```
EOL
done
```

- 2 Change the value of REPOURI to the location of the unconnected mirror repository.

```
REPOURI="http://example.company.com"
```

- 3 Save and close the createrepodefn.sh script.
- 4 Set the Execute bit for the createrepodefn.sh script.

```
chmod +x createrepodefn.sh
```

- 5 Run the createrepodefn.sh script.

```
./createrepodefn.sh
```

- 6 The script creates a file named sas-manual.repo. Copy the sas-manual.repo file to each deployment target.

```
scp sas-manual.repo user@deploytarget:/tmp
```

- 7 On each deployment target, as a user with root privileges, copy sas-manual.repo file to `/etc/yum.repos.d/`.

Install on the Target Machines

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

- SAS Object Spawner:

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

- SAS/CONNECT:

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

Note: If SAS/CONNECT is not included in your software order, skip this command.

If a command does not produce any results, then the port is available for use and no changes are required. If the command produces results, then the port is already being used by a product and is blocked for other products to use. Note any blocked products in preparation for additional steps to be performed after the installation has completed.

- 2 Copy the final yum groupinstall command from the `customized_deployment_script.sh` file on the connected mirror repository. Here is an example:

```
# Install the software
yum groupinstall "SAS/CONNECT" "SAS CAS for Event Stream Processing" "SAS Machine Learning"
"SAS CAS for Machine Learning" "SAS Statistics" "SAS CAS for Statistics" "SAS Foundation"
"SAS CAS for Visual Analytics"
```

- 3 Run the copied yum groupinstall command on each machine on which SAS Viya will be installed.
- 4 If you have any blocked products from step 1, modify the required file as follows:
 - SAS Object Spawner:

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

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

- SAS/CONNECT:

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

- 5 Ensure that the license file from the `SAS_Viya_playbook` directory on the connected mirror repository is available to all machines in the deployment. Copy the license file to the unconnected mirror repository as needed.
- 6 Complete the deployment by following the steps described in “[Deploying with Yum](#)” on page 171, starting with “[Deploy httpd and MOD_SSL](#)” on page 173.

Creating a Local Copy of Documentation

You can configure your software to give your users access to local documentation. Here are two instances where access to local documentation would be useful:

- You have customized your documentation.
- Your SAS system is highly secure, and it does not have access to the internet. Because the SAS documentation is cloud-hosted, it would not be accessible through the internet.

Note: The cloud-hosted SAS documentation is frequently updated. The SAS administrator should refresh the local copy on a regular basis to ensure that your users have up-to-date information.

You can download PDF versions of the documentation, or you can create customized versions of the documentation. Create an HTML page with links to all documents that make up your local documentation collection, and create a link to this page.

To configure local documentation, follow these steps:

- 1 Access SAS Environment Manager.
- 2 Select **Configuration** from the left navigation bar.
- 3 Under the **View** menu, select **Definitions**.
- 4 Select the `sas.htmlcommons` definition.
- 5 Click **New Configuration**.
- 6 On the New htmlcommons Configuration pane, click **Add Property** to add the following two properties:
 - **additionalHelpMenuUrl** — Specify the path to the HTML page that contains links to your local documentation.
 - **additionalHelpMenuLabel** — Provide a meaningful label for the link for your users to access. If you do not provide this parameter, a default label of **Additional Help** is used.
- 7 Click **Save** on the New htmlcommons Configuration pane.

Users see a new item in the **Help Menu** list, between the **Help Center** and **About** entries. Using this link opens the specified HTML page.

Uninstalling SAS Viya from Mirrored Repositories

Uninstall the Repositories

- 1 On the connected mirror repository machine, remove the SAS packages with the following command:

```
yum remove "sas-*
```

- 2 On the connected mirror repository machine, remove all the files in the location described in the `repo_vars.yml` file as the `mirror_loc` value with the following command.

```
rm -rf directory-location
```

For example, if you used the default value for `mirror_loc` in the `repo_vars.yml` file, the command would be the following:

```
rm -rf /opt/sas/repomirror
```

- 3 On the unconnected mirror repository machine, remove all the files in the location described in the `repo_vars.yml` file as the `httpd_doc_root` value with the following command.

```
rm -rf directory-location
```

For example, if you used the default value for `httpd_doc_root` in the `repo_vars.yml` file, the command would be the following:

```
rm -rf /var/www/html/pulp
```

- 4 (Optional) Remove the `httpd` service from the unconnected repository mirror machine.

Uninstall from the Deployment Target

Uninstalling SAS Viya software from the machines where the software is deployed is the same process that would you use if you did not have mirror repositories. For the process to remove the software, see [“Uninstalling SAS Viya” on page 155](#).

Appendix 4

Hadoop Deployment: Configuring SAS Access to Hadoop and SAS Data Connector to Hadoop

<i>Supported Hadoop Distributions</i>	203
<i>Deployment Tasks for Hive Access</i>	203
<i>Pre-deployment Hadoop Tasks for Hive Access</i>	204
Pre-deployment Checklist for Hive Access	204
Security	205
<i>Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop</i>	205
Requirements to Deploy JAR Files on the CAS Controller	205
Install the Hadoop JAR Files on the CAS Controller	206
Verify SAS Data Connector to Hadoop	209
Set Up Multiple Hadoop Versions for Multiple Hadoop Servers	210

Supported Hadoop Distributions

Before you set up Hadoop, you must make sure that your Hadoop distribution is supported by SAS Viya. For details, see [“Hadoop Requirements”](#) on page 27.

Deployment Tasks for Hive Access

For Hive access, perform the following tasks:

- 1 Perform the pre-deployment tasks. For more information, see [“Pre-deployment Hadoop Tasks for Hive Access”](#) on page 204.
- 2 Deploy the Hadoop JAR files. For more information, see [“Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop”](#) on page 205.
- 3 If you are using the SAS Data Connect Accelerator for Hadoop, deploy the SAS Embedded Process. For more information, see [“Deploy the SAS Embedded Process”](#) on page 213.

Pre-deployment Hadoop Tasks for Hive Access

Pre-deployment Checklist for Hive Access

Before you install SAS Viya software that interacts with Hadoop and Hive, it is recommended that you verify your Hadoop environment. Use the following checklist:

- Ensure that you have configured SAS Data Connector to Hadoop and, if required, SAS Data Connect Accelerator for Hadoop. For details, see [“SAS/ACCESS Interface to Hadoop and SAS In-Database Technologies for Hadoop” on page 70](#).
- Understand and verify your Hadoop user authentication.
- Have sudo access on the NameNode.
- Enable the HDFS user with Write permission to the root of HDFS.

The HDFS user home directory, `/user/user-account`, must exist and must have `drwxrwxrwx` permissions for the HDFS user directory. If you deploy the SAS Embedded Process, this user account is used to manually deploy in the [“Deploy the SAS Embedded Process” on page 213](#) section.

- Verify that the Hadoop master node can connect to the Hadoop slave nodes using passwordless SSH. For more information, see the Linux manual pages about **ssh-keygen** and **ssh-copy-id**.
- Understand and verify your security setup.
 - Verify that you can use your defined security protocol to connect from your client machine, which is outside of the SAS Viya environment) to your Hadoop cluster.
 - It is highly recommended that you enable Kerberos or another security protocol for data security. If your cluster is secured with Kerberos, you must obtain a Kerberos ticket. You also must have knowledge of any additional security policies.
 - For clusters that have Kerberos security enabled, verify that you have a valid ticket on the node on which the Hive2 service is running.
- Gain working knowledge about the Hadoop distribution that you are using (for example, Cloudera or Hortonworks).

You also need working knowledge about the HDFS, MapReduce 2, YARN, and Hive services. For more information, see the Apache website or the vendor’s website.

For MapR, you must install the MapR client. The installed MapR client version must match the version of the MapR cluster that SAS Viya connects to. For more information, see the MapR documentation.

- Verify that the HCatalog, HDFS or Hive, MapReduce, and YARN services are running on the Hadoop cluster. SAS Viya software uses these various services, which ensure that the appropriate JAR files are located during the configuration.
- For the Hive server:
 - Identify the machine on which the Hive server is running. If the Hive server is not running on the same machine as the NameNode, note the server and port number of the Hive server for future configuration.
 - Know the host name of the Hive server and the host name of the NameNode.
- For MapReduce:
 - Know the location of the MapReduce home directory.
 - Request permission to restart the MapReduce service.
 - Verify that you can run a MapReduce job successfully.

Security

Kerberos Security

SAS Data Connector to Hadoop can be configured for a Kerberos ticket cache-based logon authentication by using MIT Kerberos 5 Version 1.9.

Note: SAS Viya must be configured for pluggable authentication module (PAM) support.

If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you must manually add the Java Cryptography Extension `local_policy.jar` file to each instance of `JAVA_HOME` on the Hadoop cluster. If you are located outside the United States, you must also manually add the `US_export_policy.jar` file. The addition of these files is governed by the United States import control restrictions.

If you are using the Oracle JRE or the IBM JRE, the appropriate JAR file must also replace the existing `local_policy.jar` file and the `US_export_policy.jar` file in your JRE location. This location is typically the `JAVA_HOME/jre/lib/security/` directory. You can obtain the appropriate file from the Oracle website or the IBM website.

It is recommended that you back up the existing `local_policy.jar` file and the `US_export_policy.jar` file first in case they need to be restored.

If you are using the OpenJDK, the files do not need to be replaced.

JDBC Read Security for Hive

SAS Data Connector to Hadoop can access Hadoop data through a JDBC connection to Hive. Depending on your release of Hive, Hive might not implement Read security. A successful connection from SAS Viya can allow Read access to all data that is accessible to the identity that is used to access the Hive server. Hive can be secured with Kerberos. SAS Data Connector to Hadoop supports Kerberos 5 Version 1.9 or a later release.

Configure SAS/ACCESS to Hadoop and SAS Data Connector to Hadoop

Requirements to Deploy JAR Files on the CAS Controller

- Hadoop cluster manager:
 - host name and port number
 - credentials (account name and password)
- Hive service host name
- SSH credentials of the Linux account that has access to the machine on which the Hive service has been installed and is running.
- If your deployment includes MapReduce users from Windows clients, after you run the `hadoop_extract.sh` script, you must follow the instruction to edit the `mapred-site.xml` file and set the `mapreduce.app-submission.cross-platform` property to `true`.

Install the Hadoop JAR Files on the CAS Controller

Overview of Installing the Hadoop JAR Files

You can install the Hadoop JAR files by using either of the following methods:

- Ansible
- Manual steps

Install the Hadoop JAR Files with Ansible

- 1 Ensure that Python, strace, and wget have been installed on the Hadoop cluster from the package repositories for your Linux distribution.
- 2 On the Ansible controller, run the following command in order to enable passwordless SSH:

Note: If the SAS install user is different from the user that is set up on the Hadoop cluster, you might want to specify the `ssh-copy-id` specifically for that user for the Hadoop cluster.

```
ssh-copy-id Hive-server-machine
```

- 3 Edit the `inventory.ini` file to add the Hadoop cluster machine to the list of target references at the beginning of the file. For more information, see [“Specify the Machines in the Deployment” on page 57](#).
- 4 In the `inventory.ini` file, add a machine target for the Hadoop Hive node. Also, beneath the list of target machines, add the `[hadooptracr1]` group. Add the new Hadoop machine target to the new group.

```
hadoop-cluster ansible_host=ansible-host ansible_ssh_user=user
[hadooptracr1]
hadoop-cluster
```

For more information see [“Assign the Target Machines to Host Groups” on page 58](#).

- 5 Open the `all` file that is located in the directory where you unpacked the Ansible playbook:

```
sudo vi /sas/install/group_vars/all
```

- 6 Modify the following variables using the descriptions in the comments in the `all` file:

Note: If the directory does not exist, it is automatically created when you run the Ansible playbook.

- `hadoop_conf_home`: `/opt/sas/viya/config/data/hadoop`
- `lib_folder_name`: `lib`
- `conf_folder_name`: `conf`

Note: These directories correspond to a JAR file path of `/opt/sas/viya/config/data/hadoop/lib` and to a configuration file path of `/opt/sas/viya/config/data/hadoop/conf`.

- 7 Save and close the `all` file.

- 8 Run the playbook:

```
ansible-playbook utility/hadooptracer-launch.yml
```

Ansible will copy files to the Hadoop cluster node and then to the CAS controller and SAS programming nodes.

- 9 (Cloudera and Hortonworks distributions only) Verify that the required Hadoop JAR files are successfully collected:

```
ansible-playbook utility/hadooptracer-validation.yml
```

Install the Hadoop JAR Files Manually

The `hadoop_extract` script and the `sas_hadoop_config.properties` file are located on the CAS controller machine. The `hadoop_extract` script collects the Hadoop library JAR files and its configuration files from the Hadoop cluster. It also makes the files available to the SAS Viya products that require access to the Hadoop cluster. The `hadoop_extract` script uses information from the `sas_hadoop_config.properties` file.

Note: The `hadoop_extract` script was formerly known as the Hadoop tracer script.

Note: As an alternative, you can use the `-p` option to specify an alternative properties file.

- 1 Ensure that Python, `strace`, and `wget` have been installed on the Hadoop cluster from the package repositories for your Linux distribution.
- 2 Ensure that the user who runs the script has a home directory in HDFS that has Read and Write access. For example, the user `jsmith` who is running the script owns the `/user/jsmith` home directory.
- 3 Locate the `sas_hadoop_config.properties` file or an alternative properties file on the CAS controller machine in the `/opt/sas/viya/home/SASFoundation/etc` directory. Edit the appropriate `.properties` file and make the following changes:

- a Set the distribution name:

```
hadoop.cluster.distribution.name=distribution
```

```
distribution = cloudera | hortonworks | mapr
```

- b Set the qualified host name of the node on which the Hadoop Hive service is deployed.

```
hadoop.cluster.hivenode.hostname=hostname
```

- c Ensure that the following requirements have been met on the machine on which the Hive2 services is running.

- A valid SSH account.
- A home directory for the `hadooptracer.log` file. The `hadooptracer.log` file is written to the home directory of the `hadoop.cluster.hivenode.ssh.account` user.
- If your Hadoop cluster includes Kerberos that has been enabled, the user account should also include a configured Kerberos principal. A valid Kerberos ticket for the same user account must be available on the node on which the Hive2 service is running.

- d Set the user name and password for SSH authentication to the machine on which the Hive2 service is running. Instead of entering an SSH password, the password property can be left blank in order to be prompted for the password.

```
hadoop.cluster.hivenode.ssh.account=user-account-name
```

Note: The user account is not required to also be an administrative account. The user account must be a Hadoop user account.

- e Set the directories to which the script will store XML and JAR files:

```
hadoop.client.jar.filepath=directory-path
```

```
hadoop.client.config.filepath=directory-path
```

Note: Each of the paths for `hadoop.client.config.filepath`, `hadoop.client.jar.filepath`, and `hadoop.client.configfile.repository` must be unique and must be accessible from all CAS machines, such as a shared file system location.

On the CAS controller, ensure that Write permission has been granted to the directories that are specified in the `hadoop.client.jar.filepath` property and the `hadoop.client.config.filepath` property.

- f** Set the location to which the JAR files and configuration files are backed up. The script creates a new directory `hive/hivenode-name/time-stamp` under the specified repository.

```
hadoop.client.configfile.repository=directory-path
```

Note: The paths for `hadoop.client.config.filepath`, `hadoop.client.jar.filepath`, and `hadoop.client.configfile.repository` must be different.

- g** Set the directory part and the first part of the log filename. The script creates a log file and names it using the first part of the log filename with a timestamp. The script creates the file `sashadoopconfig_time-stamp.log`. An example of a filename is `sashadoopconfig_2017-04-14-10.16.33.log`.

```
hadoop.client.sasconfig.logfile.name=/directory-path/sashadoopconfig
```

- h** To increase the amount of information that is logged, change the default value of the following properties from 0 to 3:

```
hadoop.client.config.log.level=3
```

Here are the supported values:

1 (default)

adds INFO messages.

2

adds DEBUG messages.

3

adds consoleAppender to the log plus level 1 (HadoopTracer.py output).

- i** Select the option that specifies how the script should filter the JAR files. Using this option, the script detects any duplicate JAR files (files with the same name) and replaces them with files that are based on the selected option.

Here are the supported values:

latest (default)

Duplicate JAR files are replaced by the latest version.

none

JAR files are extracted without filtering.

When you run the `hadoop_extract.sh` script, by default, any duplicate names of JAR files that are extracted from the cluster are removed. The latest version of the JAR file with the duplicate name is copied to the specified location. To keep multiple versions of the JAR files, set the `hadoop.tracer.filter` in the `sas_hadoop_config.properties` file to `none`. The default is `latest`.

```
hadoop.tracer.filter=latest
```

- 4** For MapR, add the JAR filename `hadoop-0.20.2-dev-core.jar` to the current exclusion list as follows:

```
hadoop.jar.exclusion.list=derby,spark-examples,hadoop-0.20.2-dev-core.jar
```

- 5** Locate the installation directory on the CAS controller machine, and navigate to the `/opt/sas/viya/home/SASFoundation/utilities/bin` directory, which contains the script.

Note: The user who runs the script must have a PATH that includes the required Java version (1.8 or later release).

Note: You can specify a different properties file by specifying the `-p` option. Here is an example:

```
./hadoop_extract.sh -p alternative-properties-file
```

Run the script:

```
./hadoop_extract.sh
```

You are prompted for the Hive password, which is the password for the SSH user account. The SSH user account connects to the Hadoop cluster that corresponds to the `hadoop.cluster.hivenode.ssh.account` name. The account name is specified in the `sas_hadoop_config.properties` file.

Note: Some error messages in the console output for `hadooptracer.py` are normal and do not necessarily indicate a problem with the JAR and configuration file collection process. However, if the files are not collected as expected or if you experience problems connecting to Hadoop with the collected files, contact SAS Technical Support and include the `hadooptracer.log` file.


- 6 If your deployment includes MapReduce users from Windows clients, locate the `mapred-site.xml` file in the `hadoop-client.config.filepath` directory. Edit the `mapred-site.xml` file and set the property `mapreduce.app-submission.cross-platform` equal to `true`. Here is an example:

```
<property>
  <name>mapreduce.app-submission.cross-platform</name>
  <value>true</value>
</property>
```

Note: Be sure to make this modification after you run the `hadoop_extract.sh` script.

Verify SAS Data Connector to Hadoop

To verify that the software has been successfully deployed:

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: `https://http-proxy-host-name/SASStudio`.
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services**.
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.
- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Hadoop LIBNAME statement:

```
option set=SAS_HADOOP_CONFIG_PATH="path-to-config-files";
option set=SAS_HADOOP_JAR_PATH="path-to-jar-files"; libname hdplib hadoop server="hive-hadoop-host-name" user=
password=user-password;
```

Note: Do not use the `USER=` argument if your Hadoop cluster is secured by Kerberos.

For more information, see [LIBNAME Statement Specifics for Hadoop](#).

If SAS/ACCESS to Hadoop was successfully deployed, the execution of the LIBNAME statement will return results without error.

- 4 From SAS Studio, edit and run the following SAS code to verify SAS Data Connector to Hadoop:

```
caslib hdlib datasource=(srctype="hadoop", dataTransferMode="serial", username="user-ID",
server="hadoop-host-name",
```

```
hadoopjarpath="path-to-jar-files",  
hadoopconfigdir="path-to-config-files",  
schema="Hadoop-schema-name");  
  
proc casutil;  
list files incaslib="hdlib";  
run;
```

If the data connector was successfully deployed, the results are the names of the tables in Hive.

If an error was returned on the execution of the LIBNAME statement or no table information was returned for the data connector, you should perform the configuration steps again.

Set Up Multiple Hadoop Versions for Multiple Hadoop Servers

If you have multiple Hadoop servers that run different Hadoop versions:

- The version of the JAR files in the `hadoopJarPath` directory on the CAS server must match the version of the JAR files on the Hadoop server to which CAS connects.
- Each CAS session can connect only to Hadoop clusters of one configured `hadoopJarPath` version.
- Separate concurrent CAS sessions can independently connect to different versions of Hadoop clusters.

To support multiple Hadoop versions:

- 1 Create and populate separate directories with version-specific Hadoop JAR files for each Hadoop version.
- 2 Start separate CAS sessions, and point each separate CAS session to one of the `hadoopJarPath` versions.

Upgrading your Hadoop server version might involve multiple active Hadoop versions. The same multi-version instructions apply.

Appendix 5

Hadoop Deployment: Configuring SAS In-Database Technologies

Hadoop Prerequisites	212
Overview of the In-Database Deployment Package for Hadoop	212
SAS Embedded Process for SAS 9.4: Uninstall and Deploy for SAS Viya	212
Deploy the SAS Embedded Process	213
Methods to Deploy the SAS Embedded Process	213
Deploy Manually	213
Deploy the SAS Embedded Process with Cloudera Manager	216
Deploy the SAS Embedded Process with Ambari	217
(Optional) Deploy TLS Certificates	218
SASEP-ADMIN.SH Script	219
Overview of the SASEP-ADMIN.SH Script	219
SASEP-ADMIN.SH Syntax	219
Verify SAS Data Connect Accelerator for Hadoop	223
Additional Configuration for HCatalog File Formats	224
Overview of HCatalog File Types	224
Prerequisites for HCatalog Support	224
SAS Client Configuration	225
SAS Server-Side Configuration	225
Add the YARN Application CLASSPATH for MapR	226
Performance Tuning for the SAS Embedded Process	226
Overview of Performance Tuning Properties	226
Change the Trace Level	227
Specify the Number of MapReduce Tasks	228
Specify the Amount of Memory That the SAS Embedded Process Uses	228
Specify the Number of Input Buffers and an Optimal Buffer Size	228
Add the SAS Embedded Process to Nodes after the Initial Deployment	229
Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya	229
Options for Uninstallation	229
Uninstall Manually	229
Uninstall with Cloudera Manager	229
Uninstall with Ambari	230

Hadoop Prerequisites

The SAS In-Database Technologies for Hadoop on SAS Viya includes SAS Data Connect Accelerator for Hadoop and the SAS Embedded Process for Hadoop. The installation of the in-database deployment package for Hadoop involves writing a configuration file to HDFS and deploying files on all the data nodes. The following tasks can occur automatically, depending on your Hadoop and HDFS permissions.

- The CAS controller and each CAS worker node must have an IP address that can be routed to externally from the SAS Embedded Process nodes.
- Deploying files across all nodes requires passwordless SSH.

Note: If you run the SAS Embedded Process installation script (`sasep-admin.sh`) with `sudo` access, the script detects the Hadoop cluster topology and automatically deploys the files across all data nodes. Otherwise, you must specify the hosts on which the SAS Embedded Process for Hadoop is installed when you run the install script.

Note: The passwordless SSH user must also have Read, Write, and Execute permissions on the installation directory. The directory structure of the nodes in must match that of the installation directory.

- Writing the configuration file requires Write permission to HDFS.

Note: The SAS Embedded Process installation script creates the configuration file on the local file system in the `EPInstallDir/conf` folder. If you run the SAS Embedded Process installation script with `sudo` access, the script automatically creates and writes the configuration file to HDFS during the initial deployment. If you do not run the script with `sudo` access, you must manually copy the configuration file to HDFS.

- The parallel loading of data using SAS Data Connect Accelerator for Hadoop requires a massively parallel processing (MPP) system.

Overview of the In-Database Deployment Package for Hadoop

SAS In-Database Technologies Hadoop for SAS Viya includes SAS Data Connect Accelerator and the SAS Embedded Process for Hadoop. This section describes how to install and configure the in-database deployment package for Hadoop.

SAS Embedded Process for SAS 9.4: Uninstall and Deploy for SAS Viya

You should always install the latest release of the SAS Embedded Process.

- 1 Uninstall the existing SAS Embedded Process for SAS 9.4 before installing the SAS Embedded Process for SAS Viya. For details, see [“Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya” on page 229](#).
- 2 Install the SAS Embedded Process for SAS Viya.

Deploy the SAS Embedded Process

Methods to Deploy the SAS Embedded Process

You can either deploy manually or deploy automatically by using the cluster manager for your Hadoop distribution:

- To deploy manually, see “Deploy Manually” on page 213.

TIP Many options are available for installing the SAS Embedded Process. For more information, see “SASEP-ADMIN.SH Script” on page 219 .

- To deploy with your appropriate cluster manager:
 - To deploy with Cloudera Manager, see “Deploy the SAS Embedded Process with Cloudera Manager” on page 216.
 - To deploy with Hortonworks Ambari, see “Deploy the SAS Embedded Process with Ambari” on page 217.

CAUTION! You must uninstall the SAS 9.4 Embedded Process or SAS Viya Embedded Process using the same method that you used to install the SAS 9.4 Embedded Process or SAS Viya Embedded Process. For details about how to uninstall, see “Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya” on page 229.

Deploy Manually

- 1 On the Hadoop master node, create a new directory that is not part of an existing directory structure such as `/opt/sasep`.

This path is created on each node in the Hadoop cluster during installation of the SAS Embedded Process. It is recommended that you do not use existing system directories such as `/usr`. This new directory is referred to as *EPInstallDir* throughout this section.

- 2 On the CAS controller node, navigate to the `/opt/sas/viya/home/share/ep` directory.
- 3 Locate the `sepcorehadp-14.00000-n.sh` file, where *n* is a number that indicates the latest version of the file.
- 4 Copy the `sepcorehadp-14.00000-n.sh` file from the client to *EPInstallDir* on the Hadoop cluster. Here is an example that uses the secure copy command:

```
scp sepcorehadp-14.00000-n.sh username@hdpclus1:/EPInstallDir
```

Note: The location to which you transfer the `sepcorehadp-14.00000-n.sh` file becomes the SAS Embedded Process home and is referred to as *EPInstallDir* throughout this section.

To install the SAS Embedded Process for Hadoop, follow these steps:

Note: Passwordless SSH is required in order to install the SAS Embedded Process for Hadoop. Also, Write permission to HDFS might be required.

- 1 Navigate to the location on your Hadoop master node to which you copied the `sepcorehadp-14.00000-n.sh` file.

```
cd /EPInstallDir
```

- 2 Use the following command to unpack the `sepcorehadp-14.00000-n.sh` file.

```
./sepcorehadp-14.00000-n.sh [--verbose]
```

Note: The `--quiet` option is enabled by default. Only error messages are displayed. The `--verbose` option causes all messages to be displayed that are generated during the installation process. Using verbose messaging can increase the time that is required to perform the installation.

After this script has completed its execution and the files are unpacked, the following directory structure is created:

```
EPInstallDir/SASEPHome
EPInstallDir/sepcorehadp-14.00000-n.sh
```

Note: During the installation process, the `sepcorehadp-14.00000-n.sh` is copied to all data nodes. Do not remove or move this file from the `EPInstallDir/SASEPHome` directory.

The `SASEPHome` directory should have the following structure:

```
EPInstallDir/SASEPHome/bin
EPInstallDir/SASEPHome/jars
EPInstallDir/SASEPHome/misc
EPInstallDir/SASEPHome/sasexe
EPInstallDir/SASEPHome/utilitiesEPInstallDir/SASEPHome/security
```

The `EPInstallDir/SASEPHome/jars` directory contains the SAS Embedded Process JAR files:

```
EPInstallDir/SASEPHome/jars/sasephdp0-*.jar
EPInstallDir/SASEPHome/jars/sasephdp1-*.jar
EPInstallDir/SASEPHome/jars/sasephdp2-*.jar
```

The `EPInstallDir/SASEPHome/bin` directory should contain the following script:

```
EPInstallDir/SASEPHome/bin/sasep-admin.sh
```

- 3 If your Hadoop cluster is secured with Kerberos and you have sudo access, the HDFS user must have a valid Kerberos ticket in order to access HDFS. You can obtain a valid Kerberos ticket with the `kinit` command.

```
sudo su - root
su - hdfs | hdfs-userid
kinit -kt location-of-keytab-file-user-for-which-you-are-requesting-a-ticket principal-name
exit
```

Note: The default HDFS user is `hdfs`. You can specify a different user ID with the `-hdfsuser` argument when you run the `sasep-admin.sh -add` script. If you use a different hdfs superuser, ensure that the user has a home directory in HDFS before you run the `sasep-admin.sh -add` command. For example, if the hdfs superuser is `prodhdfs`, ensure that the `/user/prodhdfs` directory exists in HDFS.

To check the status of your Kerberos ticket on the server, as the `hdfs` user, run the `klist` command. Here is an example of the command and its output:

```
klist
Ticket cache: FILE/tmp/krb5cc_493
Default principal: hdfs@HOST.COMPANY.COM

Valid starting    Expires          Service principal
06/20/17 09:51:26 06/27/17 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
    renew until 06/22/17 09:51:26
```

- 4 Run the `sasep-admin.sh` script depending on whether you have sudo access.

If you have sudo access, complete the following steps to deploy the SAS Embedded Process on all nodes. Review all of the information in this step and the script syntax before you run the script.

- a Run the `sasep-admin.sh` script as follows.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -add
```

- b The `sepcorehadp-11.00000-n.sh` file is copied to all data nodes.

Note: If you have `sudo` access, the SAS Embedded Process installation script (`sasep-admin.sh`) detects the Hadoop cluster topology and installs the SAS Embedded Process on all DataNode nodes. The install script also installs the SAS Embedded Process on the host node from which you run the script (the Hadoop master NameNode). The SAS Embedded Process is installed even if a DataNode is not present. To add the SAS Embedded Process to new nodes at a later time, you should run the `sasep-admin.sh` script with the `-host <hosts>` option. In addition, a configuration file, `ep-config.xml`, is automatically created and written to the `EPInstallDir/SASEPHome/conf` directory and to the HDFS file system in the `/sas/ep/config` directory.

If you do not have `sudo` access, complete the following steps to deploy the SAS Embedded Process installation across all nodes. Review all of the information in this step and the script syntax before you run the script.

Note: If you do not have `sudo` access, the passwordless SSH user must have Read, Write, and Execute permissions on the `EPInstallDir` directory.

- a Run the `sasep-admin.sh` script as follows:

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -x -add -hostfile host-list-filename | -host <">host-list<">
```

Note: If you do not have `sudo` access, you must use the `-x` option and specify the hosts on which the SAS Embedded Process is deployed with either the `-hostfile` or `-host` option. Automatic detection of the Hadoop cluster topology is not available when you run the installation script with the `-x` option.

CAUTION! The SAS Embedded Process must be installed on all nodes that are capable of running a MapReduce job. The SAS Embedded Process must also be installed on the host node from which you run the script (the Hadoop master NameNode). Otherwise, the SAS Embedded Process does not function properly.

The `sepcorehadp-14.00000-n.sh` file is copied to all nodes that you specify. The configuration file, `ep-config.xml`, is created and written to the `EPInstallDir/SASEPHome/conf` directory.

- b Manually copy the `ep-config.xml` configuration file to HDFS.

Note: This step must be performed by a user that has Write permission to the HDFS root folder `/`. If your Hadoop cluster is secured with Kerberos, the user who copies the configuration file to HDFS must have a valid Kerberos ticket.

- i Log on as your HDFS user or as the user that you use to access HDFS.

- ii Create the `/sas/ep/config` directory for the configuration file.

```
hadoop fs -mkdir -p /sas/ep/config
```

- iii Navigate to the `EPInstallDir/SASEPHome/conf` directory.

- iv Use the Hadoop `copyFromLocal` command to copy the `ep-config.xml` file to HDFS.

```
hadoop fs -copyFromLocal ep-config.xml /sas/ep/config/ep-config.xml
```

- 5 Verify that the SAS Embedded Process was successfully installed by running the `sasep-admin.sh` script with the `-check` option.

If you ran the `sasep-admin.sh` script with `sudo` access, run the following command. By default, this command verifies that the SAS Embedded Process was installed on all nodes.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -check
```

If you ran the `sasep-admin.sh` script with the `-x` argument, run the following command. This command verifies that the SAS Embedded Process was installed on the hosts that you specified.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -x -check -hostfile host-list-filename | -host <">host-list<">
```

- 6 Verify that the configuration file, `ep-config.xml`, was written to the HDFS file system.

```
hadoop fs -ls /sas/ep/config/ep-config.xml
hadoop fs -cat /sas/ep/config/ep-config.xml
```

Note: If your Hadoop cluster is secured with Kerberos, you must have a valid Kerberos ticket in order to access HDFS. Otherwise, you can use the WebHDFS browser.

Note: The `/sas/ep/config` directory is created automatically when you run the installation script with `sudo` access. If you used the `-genconfig` option to specify a non-default location, use that location to locate the `ep-config.xml` file.

Deploy the SAS Embedded Process with Cloudera Manager

The following deployment steps assume either of these scenarios: the SASEP rpm package has been installed directly on the Cloudera Manager server, or the SASEP rpm package has been installed on a network location that is accessible to the Cloudera Manager server.

To deploy the SAS Embedded Process:

- 1 On the CAS controller machine, create a temporary directory on the file system of the host on which Cloudera Manager is installed.

```
mkdir -p /tmp/sasep
```

- 2 Navigate to the `/opt/sas/viya/home/share/ep/parcel` directory.

```
cd /opt/sas/viya/home/share/ep/parcel
```

- 3 Copy the parcel directory to the new directory under `tmp`:

```
cp -r * /tmp/sasep
```

- 4 Grant permission to the user account that you use to run the `install_parcel.sh` script. The user account must have super user (`sudo`) access or root access and must have Execute permission on the script. Here is an example:

- 5 From the `tmp` directory, run the following command:

```
./install_parcel.sh -v distro
```

The `tmp` directory is the location to which you copied the parcel directory from the SAS Viya installation. The variable `distro` represents one of the following Linux distributions: `redhat5`, `redhat6`, `redhat7`, `suse11x`, `ubuntu10`, `ubuntu12`, `ubuntu14`, `debian6`, or `debian7`. Select the appropriate value.

Here is an example:

```
./install_parcel.sh -v redhat6
```

- 6 When prompted to restart Cloudera Manager, select `y`.
- 7 Log on to Cloudera Manager.
- 8 Activate the SASEP parcel:

- a From the Menu bar, select **Hosts** ⇨ **Parcels**.

Note: If the SASEP parcel is missing, run **Check for new parcel**.

- b On the row for the SAS EP parcel, click **Distribute** to copy the parcel to all nodes.

- c Click **Activate**. Answer OK to the Activation prompt. You might be prompted to either restart the cluster or to close the window.

CAUTION! Do not restart the cluster.

- d If prompted, click **Close**.

9 Add the SASEP service to create the SASEP configuration file in HDFS.

- a Navigate to Cloudera Manager Home.
- b In Cloudera Manager, select the ▼ next to the name of the cluster, and then select **Add a Service**. The Add Service Wizard appears.
- c Select the SASEP service and click **Continue**.
- d On the **Add Service Wizard** ⇒ **Select the set of dependencies for your new service** page, select the dependencies for the service. Click **Continue**.

Note: The dependencies are automatically selected for this service.
- e On the **Add Service Wizard** ⇒ **Customize Role Assignments** page, select a node for the service. In the next step you must have a valid Kerberos keytab, so be sure to select a node that has a Kerberos keytab for the hdfs user on that node. Choose any single node. Click **OK**, and then click **Continue**.
- f Enter your hdfs user name. The default user name is hdfs. If your cluster is Kerberos enabled, a valid Kerberos keytab for your hdfs user name must be available on the node that was selected for the SAS Embedded Process service.
- g Click **Continue**, and then click **Finish**.

A file is added to HDFS for each of the services as follows:

SASEP: `/sas/ep/config/ep-config.xml`

Note: If the service that you have just deployed is started, navigate to Cloudera Manager Home and stop the services.

Deploy the SAS Embedded Process with Ambari

1 To launch the script:

- a On the CAS controller machine, navigate to the `/opt/sas/viya/home/share/ep/stack` directory. Copy the entire stack directory to a temporary directory (`/tmp`) on your Hadoop Ambari Cluster Manager machine.
- b Navigate to the `/tmp/stack` directory and run the following command with `sudo` or as root:

```
./install_sasepstack.sh ambariAdminUsernam
```

Note: To complete the installation process, the Ambari server must be restarted.

- To restart the Ambari server with the script, enter `y`.
- To manually restart the script at a later time, press `n`.

After the script finishes, the following message is displayed:

```
You can install the SASEP stack now from Ambari Cluster Manager.
```

2 On the Ambari server, log on to Ambari and deploy the services:

- a Click **Actions** and select **+ Add Service**.

The **Add Service Wizard** page and the **Choose Services** panel appear.

- b** In the **Choose Services** panel, select the **SASEP** service. Click **Next**.

The **Assign Slaves and Clients** panel appears.

- c** In the **Assign Slaves and Clients** panel, ensure that the NameNode, HDFS_CLIENT, and HCAT_CLIENT are selected where you want the stack to be deployed. By default, the three clients are selected. SAS recommends that you select all clients.

Note: On the **Assign Slaves and Clients** panel, place your pointer over the host name to view the details for NameNode, HDFS_CLIENT, and HCAT_CLIENT.

- d** Click **Next**. The **Customize Services** panel appears.

The SASEP service stacks are listed.

- e** Do not change any settings on the **Customize Services** panel. Click **Next**.

Note: By default, Ambari will not retain the credentials that you provide unless you have configured encrypted passwords for storage in Ambari. If you have not configured Ambari for password encryption, you will be prompted to provide credentials whenever cluster changes are made.

If your cluster is secured with Kerberos, the **Configure Identities** panel appears. Enter your Kerberos credentials in the **admin_principal** and **admin_password** text boxes. Click **Next**.

The **Review** panel appears.

- f** Review the information about the panel. If the information is correct, click **Deploy**.

The **Install, Start, and Test** panel appears. After the stack is installed on all nodes, click **Next**.

The **Summary** panel appears.

- g** Click **Complete**. The stacks are now installed on all nodes of the cluster.

The SASEP service is displayed on the Ambari dashboard.

- h** After you deploy all of the services, verify that the following file exists in the Hadoop file system:

SASEP: `/sas/ep/config/ep-config.xml`

(Optional) Deploy TLS Certificates

If you are using a SAS Data Connect Accelerator, the data that is transferred between the data provider and the CAS server is not encrypted by default. However, SAS Viya supports TLS encryption between the data provider and the CAS server. When Viya 3.3 is deployed, TLS is enabled and configured on the CAS server (server side). The deployment process provides a default level of encryption for data in motion. Options are set in the vars.yml file and are defined in the casconfig_deployment.lua file. These settings enable data connector encryption and specify the location of the TLS private key and the password.

However, you must take additional steps to enable encryption on the data provider. The prerequisites and the process for enabling TLS encryption on the data provider are different for each data provider. The first step is to deploy the TLS certificates across all nodes in the cluster.

- 1** On the CAS controller machine, locate the TLS certificates in the trustedcerts.pem file in the `/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/` directory.
- 2** Copy the trustedcerts.pem file to the `security/certs` directory on the Hadoop master node.

- 3 To complete the deployment of TLS encryption, you also must update a `dcsecurity.properties` file. Copy both the `.pem` file and the `dcsecurity.properties` file to all nodes on the CAS server. For more information about how to complete the deployment, see [Encrypt Data Transfer When Using the SAS Data Connect Accelerator in Encryption](#) in *SAS Viya: Data in Motion*.

SASEP-ADMIN.SH Script

Overview of the SASEP-ADMIN.SH Script

The `sasep-admin.sh` script enables you to perform the following actions:

- Install or uninstall the SAS Embedded Process for Hadoop on a single node or a group of nodes.
- Generate a SAS Embedded Process configuration file and write the file to an HDFS location.
- Install a hot fix to the SAS Embedded Process.
- Check whether the SAS Embedded Process is installed correctly.
- Display all live data nodes on the Hadoop cluster.
- Display the Hadoop configuration environment.
- Display the Hadoop version information for the Hadoop cluster.
- Display the version of the SAS Embedded Process that is installed.
- Deploy the security settings for SAS Data Connect Accelerator for Hadoop across all nodes in the cluster.

Note: The installation of the SAS Embedded Process for Hadoop involves writing a configuration file to HDFS and deploying files on all data nodes. These two tasks can occur automatically, depending on your Hadoop and HDFS permissions.

If you run the SAS Embedded Process install script (`sasep-admin.sh`) with `sudo` access, the script detects the Hadoop cluster topology and installs the SAS Embedded Process on all `DataNode` nodes. The install script also installs the SAS Embedded Process on the host node on which you run the script (the Hadoop master `NameNode`). In addition, a configuration file, `ep-config.xml`, is created and written to the HDFS file system.

If you do not have `sudo` access, you must specify the hosts on which the SAS Embedded Process is installed. In addition, you must manually copy the `ep-config.xml` configuration file to the HDFS file system.

SASEP-ADMIN.SH Syntax

Action options syntax:

`sasep-admin.sh`

```
<-x> -add < -hostfile host-list-filename | -host <">host-list<"> >
      <-maxscp number-of-copies > <-hdfsuser user-ID >
```

`sasep-admin.sh`

```
<-x> -genconfig < HDFS-filename > <-force>
```

`sasep-admin.sh`

```
<-x > -hotfix hotfix-filename < -hostfile host-list-filename | -host <">host-list<"> >
      <-maxscp number-of-copies > <-hdfsuser user-ID >
```

`sasep-admin.sh`

```
<-x > -remove < -hostfile host-list-filename | -host <">host-list<"> >
```

<-hdfsuser *user-ID* >

sasep-admin.sh

<-x > -security deploy | reset < -hostfile *host-list-filename* | -host <">*host-list*<"> >
<-force>

Informational options syntax:

sasep-admin.sh <-x > <-check < -hostfile *host-list-filename* | -host <">*host-list*<"> > <-hdfsuser *user-ID* > >

sasep-admin.sh <-env>

sasep-admin.sh <-hadoopversion >

sasep-admin.sh <-nodelist>

sasep-admin.sh <-version >

Action Arguments

-add

installs the SAS Embedded Process.

Requirement If you have sudo access, the script automatically retrieves the list of data nodes from the Hadoop configuration. If you do not have sudo access, you must use the -x argument and either the -hostfile or -host argument.

Tip If you add nodes to the Hadoop cluster, you can specify the hosts on which to install the SAS Embedded Process by using the -hostfile or -host option. The -hostfile option and the -host option are mutually exclusive.

See [-hostfile on page 222](#) and [-host on page 219](#)

-genconfig <*HDFS-filename*> <-force>

generates the SAS Embedded Process configuration file in the *EPInstallDir/SASEPHome/conf* directory of the local file system.

Requirement If you do not have sudo access, you must use the -x argument.

Interactions When used without the -x argument, the script creates the ep-config.xml configuration file and writes the file to both the *EPInstallDir/SASEPHome/conf* directory on the local file system and the */sas/ep/config/* directory on HDFS. You can change the filename and the HDFS location by using the *HDFS-filename* argument. *HDFS-filename* must be the fully qualified HDFS pathname that points to the location of the configuration file.

When used with the -x argument, the script does not write the configuration file to the HDFS. You must manually copy the file to the HDFS. For information, see ["Deploy Manually" on page 213](#).

Note The -genconfig argument creates two identical configuration files under *EPInstallDir/SASEPHome/conf/* on the local file system: ep-config.xml and sasep-site.xml. The sasep-site.xml file might be copied to the client side under a folder that is in the classpath. When the sasep-site.xml file is loaded from the classpath, the configuration file on the HDFS location is not used. However, if sasep-site.xml is not found in the classpath, a configuration file must exist on the HDFS. The configuration file must exist either on the default HDFS location */sas/ep/config/ep-config.xml* or in the location that is set in the *sas.ep.config.file* property.

Tips Use the -genconfig argument to generate a new SAS Embedded Process configuration file when you upgrade to a new version of your Hadoop distribution.

Use the *HDFS-filename* argument to specify another location and configuration filename. If you decide to generate the configuration file in a non-default HDFS location, you must set the

sas.ep.config.file property in the mapred-site.xml to the value that you specify in the -genconfig option.

This argument generates an updated ep-config.xml file. Use the -force argument to overwrite the existing configuration file.

Examples

The following example generates the configuration files under *EPInstallDir/SASEPHome/conf* on the local file system and the ep-config.xml configuration file under */sas/ep/config* on the HDFS:

```
./sasep-admin.sh -genconfig
```

The following example overwrites the configuration files under *EPInstallDir/SASEPHome/conf* on the local file system and under */sas/ep/config* on the HDFS, if it already exists:

```
./sasep-admin.sh -genconfig -force
```

The following example generates the configuration files under *EPInstallDir/SASEPHome/conf* on the local file system and under */home/hadoop/* on the HDFS:

```
./sasep-admin.sh -genconfig /home/hadoop/ep-config.xml
```

The following example generates the configuration files under *EPInstallDir/SASEPHome/conf* on the local file system only:

```
./sasep-admin.sh -x -genconfig
```

The following example overwrites the configuration files under *EPInstallDir/SASEPHome/conf* on the local file system only:

```
./sasep-admin.sh -x -genconfig -force
```

-hotfix *hotfix-filename*

distributes a hot fix package.

Requirements Hot fixes must be installed using the same user ID that performed the initial software installation.

Hot fixes should be installed following the installation instructions provided by SAS Technical Support.

-remove

removes the SAS Embedded Process.

Requirement If you do not have sudo access, you must use the -x argument and either the -hostfile or -host argument. The -hostfile option and the -host option are mutually exclusive.

Interactions When used without the -x argument and you have sudo access, the script automatically retrieves the list of data nodes from the Hadoop configuration. In addition, the script automatically removes the epconfig.xml file from the HDFS.

When used with -x argument, the SAS Embedded Process is removed from all hosts that you specify. However, the ep-config.xml file must be removed manually from the HDFS.

See [-hostfile on page 222](#) and [-host on page 219](#)

- security deploy | reset <-force>

deploys or resets security settings across all nodes in the cluster.

Requirement If you do not have sudo access, you must use the -x argument.

Note To overwrite security settings without a prompt, use the -force argument.

Tip You can specify one or more hosts for which you want to check the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` option and the `-host` option are mutually exclusive.

See [-hostfile on page 222](#) and [-host on page 219](#)

[-x on page 222](#)

“Encrypt Data Transfer when Using the SAS Data Connect Accelerator” in *Encryption in SAS Viya 3.3*

Informational Arguments

-check

checks whether the SAS Embedded Process is installed correctly on all data nodes.

Tip You can specify the hosts for which you want to check the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` option and the `-host` option are mutually exclusive.

See [-hostfile on page 222](#) and [-host on page 219](#)

-env

displays the SAS Embedded Process install script and the Hadoop configuration environment.

-hadoopversion

displays the Hadoop version information for the Hadoop cluster.

-nodelist

displays all live DataNodes on the Hadoop cluster.

Requirement `sudo` access is required.

-version

displays the version of the SAS Embedded Process that is installed.

Parameters for Action and Informational Arguments

-x

if you do not have `sudo` access, runs the script solely under the current user’s credential.

Requirements This option must be the first argument passed to the script.

A list of hosts must be provided with either the `-hostfile` or `-host` argument.

If you do not have `sudo` access, you must use the `-x` argument.

Interaction If you use the `-x` argument to install the SAS Embedded Process, that is, with the `-add` argument, you must use the `-x` argument in any other `sasep-admin.sh` script action that supports it.

See [-hostfile on page 222](#) and [-host on page 219](#)

-hostfile *host-list-filename*

specifies the full path of a file that contains the list of hosts on which the SAS Embedded Process is installed or removed.

Requirement The `-hostfile` or `-host` argument is required if you do not have `sudo` access.

Interaction Use the `-hostfile` argument in conjunction with the `-add`, `-hotfix`, or `-remove` arguments.

See [-hdfsuser on page 223](#)

Example `-hostfile /opt/sasep/ep.hosts`

-host <"> host-list <">

specifies the target host or host list on which the SAS Embedded Process is installed or removed.

Requirements If you specify more than one host, the hosts must be enclosed in double quotation marks and separated by spaces or commas.

The `-host` or `-hostfile` argument is required if you do not have `sudo` access.

Interaction Use the `-host` argument in conjunction with the `-add`, `-hotfix`, or `-remove` arguments.

See [-hdfsuser on page 223](#)

Example
`-host "server1 server2 server3"`
`-host bluesvr`
`-host "blue1, blue2, blue3"`

-maxscp number-of-copies

specifies the maximum number of parallel copies between the master and data nodes.

Default 10

Interaction Use this argument in conjunction with the `-add` or `-hotfix` argument.

-hdfsuser user-ID

specifies the user ID that has Write access to the HDFS root directory.

Note: The `hdfs` folder `/users/user-id` must exist. Otherwise, the command fails.

Default `hdfs`

Interactions This argument has no affect if you use the `-x` argument.

Use the `-hdfsuser` argument in conjunction with the `-add`, `-check`, or `-remove` argument in order to change, check, or remove the HDFS user ID.


Note The user ID is used to copy the SAS Embedded Process configuration files to the HDFS.

Verify SAS Data Connect Accelerator for Hadoop

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

To verify that the software has been successfully deployed:

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: `https://http-proxy-host-name/SASStudio`
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.

- b Select **Snippets** ⇒ **SAS Viya Cloud Analytic Services** .
- c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
- d In the toolbar, click  to run the new CAS session code.

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

```
cas mysession;

caslib hivelib datasource=(srctype="hadoop" server="server name"
dataTransferMode="parallel"
hadoopconfigdir="path-to-directory-containing-Hadoop-config-files-collected-with-hadoop_extract.sh"
hadoopjarpath="path-to-directory-containing-Hadoop-JAR-files-collected-with hadoop_extract.sh");
proc casutil;
load casdata="Hive table to load" casout="CAS table name"
incaslib="hivelib";
run;
```

The SAS code loads the table from Hive into CAS. You can check the log to verify that the load was successful. As an option, to view the data, run the following code to assign a libref to the caslib and view the table with PROC PRINT:

```
libname caslib cas caslib=hivelib;
proc print data=caslib.<CAS table name>; run;
```

If SAS Data Connect Accelerator and the SAS Embedded Process have been successfully deployed, the results show the appearance of data in the table. If you do not see the data, you should perform the configuration steps again.

Additional Configuration for HCatalog File Formats

Overview of HCatalog File Types

HCatalog is a table management layer that presents a relational view of data in the HDFS to applications within Hadoop. With HCatalog, data structures that are registered in the Hive metastore, including SAS data, can be accessed through standard MapReduce code and Apache Pig. HCatalog is included in Apache Hive.

The SAS Embedded Process for Hadoop uses HCatalog to process the following complex, non-delimited Apache file formats: Avro, ORC, Parquet, and RCFile.

Prerequisites for HCatalog Support

Here are additional prerequisites for accessing complex, non-delimited file types such as Avro or Parquet:

- Hive and HCatalog must be installed on all nodes of the Hadoop cluster.
- HCatalog support depends on the version of Hive that is running on your Hadoop distribution. See the following table for more information.

Note: For MapR distributions, Hive 0.13.0 build: 1501 or later must be installed for access to any HCatalog file type.

File Type	Required Hive Version
Avro	0.14
ORC	0.11
Parquet	0.13
RCFile	0.6

SAS Client Configuration

Note: If you used the `hadoop_extract.sh` script to install the Hadoop JAR files, the configuration tasks in this section are unnecessary. SAS client configuration was completed using the script. For more information, see [“Install the Hadoop JAR Files on the CAS Controller” on page 206](#).

The following additional configuration tasks must be performed:

- The `hive-site.xml` configuration file must be included in the `hadoopConfigDir` path.
- The following Hive or HCatalog JAR files must be included in the `hadoopJarPath` path.
 - `hive-hcatalog-core-*.jar`
 - `hive-webhcat-java-client-*.jar`
 - `jdo-api*.jar`
- If you are using MapR, the following Hive or HCatalog JAR files must be included in the `SAS_HADOOP_JAR_PATH`.
 - `hive-hcatalog-hbase-storage-handler-0.13.0-mapr-1408.jar`
 - `hive-hcatalog-server-extensions-0.13.0-mapr-1408.jar`
 - `hive-hcatalog-pig-adapter-0.13.0-mapr-1408.jar`
 - `datanucleus-api-jdo-3.2.6.jar`
 - `datanucleus-core-3.2.10.jar`
 - `datanucleus-rdbms-3.2.9.jar`

For more information about the `hadoopConfigDir` path and the `hadoopJarPath` path, see the CASLIB statement in the *SAS Viya Cloud Analytic Services: Language Reference*.

SAS Server-Side Configuration

The SAS Embedded Process deployment automatically sets the HCatalog CLASSPATH in the `ep-config.xml` file. You could also manually append the HCatalog CLASSPATH to the MapReduce configuration property `mapreduce.application.classpath` in the `mapred-site.xml` file on the client side.

Here is an example of an HCatalog CLASSPATH for a Cloudera distribution:

```
/opt/cloudera/parcels/CDH-version/bin/./lib/hive/lib/*,  
/opt/cloudera/parcels/CDH-version/lib/hive-hcatalog/libexec/./share/hcatalog/*
```

Here is an example of an HCatalog CLASSPATH for a Hortonworks distribution:

```
/usr/hdp/version/hive-hcatalog/libexec/./share/hcatalog/*,/usr/hdp/2.4.2.0-258/hive/lib/*
```

Add the YARN Application CLASSPATH for MapR

Two configuration properties specify the YARN application CLASSPATH: `yarn.application.classpath` and `MapReduce.application.classpath`. If you do not specify the YARN application CLASSPATH, MapR uses the default CLASSPATH. However, if you specify the MapReduce application CLASSPATH, the YARN application CLASSPATH is ignored. The SAS Embedded Process for Hadoop requires both the YARN application CLASSPATH and the MapReduce application CLASSPATH.

To ensure that the YARN application CLASSPATH exists, you must manually add the YARN application CLASSPATH to the `yarn-site.xml` file. Without the manual definition in the configuration file, the MapReduce application master fails to start a YARN container.

Here is the default YARN application CLASSPATH for Linux:

```
$HADOOP_CONF_DIR,  
$HADOOP_COMMON_HOME/share/hadoop/common/*,  
$HADOOP_COMMON_HOME/share/hadoop/common/lib/*,  
$HADOOP_HDFS_HOME/share/hadoop/hdfs/*,  
$HADOOP_HDFS_HOME/share/hadoop/hdfs/lib/*,  
$HADOOP_YARN_HOME/share/hadoop/yarn/*,  
$HADOOP_YARN_HOME/share/hadoop/yarn/lib/*
```

Here is the default YARN application CLASSPATH for Windows:

```
%HADOOP_CONF_DIR%,  
%HADOOP_COMMON_HOME%/share/hadoop/common/*,  
%HADOOP_COMMON_HOME%/share/hadoop/common/lib/*,  
%HADOOP_HDFS_HOME%/share/hadoop/hdfs/*,  
%HADOOP_HDFS_HOME%/share/hadoop/hdfs/lib/*,  
%HADOOP_YARN_HOME%/share/hadoop/yarn/*,  
%HADOOP_YARN_HOME%/share/hadoop/yarn/lib/*
```

Note: On MapR, the YARN application CLASSPATH does not resolve the symbols or variables that are included in pathnames such as `$HADOOP_HDFS_HOME`.

TIP To apply any change that you make to the cluster, you must restart the node managers.

Performance Tuning for the SAS Embedded Process

Overview of Performance Tuning Properties

You can tune the SAS Embedded Process by editing certain properties in the `ep-config.xml` file or the `mapred-site.xml` file, as appropriate.

The `ep-config.xml` file is created when you install the SAS Embedded Process. By default, the file is located in the `/sas/ep/config/ep-config.xml` directory.

The `mapred-site.xml` file is copied to the client machine when the `hadoop_extract.sh` script was run. By default, the file is located in the directory that you specified for the `hadoop.client.config.filepath` variable.

You can change the values of the following properties:

- trace levels

For more information, see [“Change the Trace Level” on page 227](#).

- the number of the SAS Embedded Process MapReduce tasks per node

For more information, see [“Specify the Number of MapReduce Tasks” on page 228](#).

- the maximum amount of memory in bytes that the SAS Embedded Process is allowed to use

For more information, see [“Specify the Amount of Memory That the SAS Embedded Process Uses” on page 228](#).

- the buffers for input data

For more information, see [“Specify the Number of Input Buffers and an Optimal Buffer Size” on page 228](#).

- the number of concurrent input reader threads

Each reader thread takes a file split from the input splits queue, opens the file, positions itself at the beginning of the split, and starts reading the records. Each record is stored in a native buffer that is shared with the DS2 container. When the native buffer is full, it is pushed to the DS2 container for processing. When a reader thread finishes reading a file split, it takes another file split from the input splits queue. The default number of input threads is 3.

```
<property>
  <name>sas.ep.input.threads</name>
  <value>number-of-input-threads</value>
</property>
```

- the number of output writer threads

The `sas.ep.output.threads` property is used to change the number of output writer threads. The SAS Embedded Process super writer technology improves performance by writing output data in parallel and producing multiple parts of the output file per mapper task. Each writer thread is responsible for writing one part of the output file. The default number of output threads is 2.

```
<property>
  <name>sas.ep.output.threads</name>
  <value>number-of-output-threads</value>
</property>
```

Change the Trace Level

You can modify the level of tracing by changing the value of the `sas.ep.server.trace.level` property in the `ep-config.xml` file. The default value is 4 (`TRACE_NOTE`).

```
<property>
  <name>sas.ep.server.trace.level</name>
  <value>trace-level</value>
</property>
```

The *trace-level* represents the level of trace that is produced by the SAS Embedded Process. Here are the *trace-level* values:

Note: Trace options can produce a significant volume of output. If trace options are not required for troubleshooting or monitoring, set the *trace-level* value to 0.

```
0
  TRACE_OFF
1
  TRACE_FATAL
```

2	TRACE_ERROR
3	TRACE_WARN
4	TRACE_NOTE
5	TRACE_INFO
10	TRACE_ALL

Specify the Number of MapReduce Tasks

You can specify the number of the SAS Embedded Process MapReduce Tasks per node by changing the `sas.ep.superreader.tasks.per.node` property in the `ep-config.xml` file. The default number of tasks is 6.

```
<property>
  <name>sas.ep.superreader.tasks.per.node</name>
  <value>number-of-tasks</value>
</property>
```

Specify the Amount of Memory That the SAS Embedded Process Uses

The SAS Embedded Process is managed by the Hadoop MapReduce framework. Load balancing and resource allocation are managed by YARN. Adjust the YARN container limits to change the amount of memory that the SAS Embedded Process. For information about how CAS uses memory, see [“Memory”](#) in *SAS Cloud Analytic Services: Fundamentals*.

Specify the Number of Input Buffers and an Optimal Buffer Size

You can specify the number of buffers in which to store input data and the optimal size of one input buffer. You specify this information by changing the `sas.ep.input.buffers` and `sas.ep.optimal.input.buffer.size` properties in the `mapred-site.xml` file.

The default value of the `sas.ep.input.buffer` property is 4 buffers. The default value of the `sas.ep.optimal.input.buffer.size` property is 1MB.

```
<property>
  <name>sas.ep.input.buffers</name>
  <value>number-of-buffers</value>
</property>

<property>
  <name>sas.ep.optimal.input.buffer.size.mb</name>
  <value>buffer-size-in-MB</value>
</property>
```

Add the SAS Embedded Process to Nodes after the Initial Deployment

After the initial deployment of the SAS Embedded Process, you might add more nodes to your Hadoop cluster. Also, you might replace selected nodes. In these instances, you can install the SAS Embedded Process on the new nodes.

Run the `sasep-admin.sh` script and specify the nodes on which to install the SAS Embedded Process. For more information, see the `-add` argument in “[-add](#)” on page 220.

Uninstall the SAS Embedded Process for SAS 9.4 or SAS Viya

Options for Uninstallation

- To uninstall manually, see “[Uninstall Manually](#)” on page 229.
- To uninstall with Cloudera Manager, see “[Uninstall with Cloudera Manager](#)” on page 229.
- To uninstall with Ambari, see “[Uninstall with Ambari](#)” on page 230.

Uninstall Manually

To uninstall manually, run the following commands:

```
cd EPInstallDir/SASEPHome/bin/  
./sasep-admin.sh -remove
```

Uninstall with Cloudera Manager

- 1 Log on to Cloudera Manager.
 - 2 Stop the SAS EP service if it is running.
 - 3 From the **Menu** bar, select **Hosts** ⇒ **Parcels**.
 - 4 Select the SASEP parcel.
 - 5 Deactivate the SASEP parcel.
 - 6 Remove the SASEP parcel.
 - 7 Delete the SASEP parcel.
 - 8 When prompted, click **Close**.
- CAUTION!** Do not restart the cluster.
- 9 Run the following command to remove the `/sas/ep` directory.

```
hadoop fs -rm -r -f /sas/ep
```

Uninstall with Ambari

Note: Root or passwordless sudo access is required in order to remove the stack.

- 1 On the CAS controller machine, navigate to the `/opt/sas/viya/home/share/ep/stack` directory.
- 2 Copy the entire stack directory to a temporary directory (`/tmp`) on your Hadoop Ambari Cluster Manager machine.
- 3 Navigate to the `/tmp/stack` directory and run the following command to delete the stack:

```
./delete_stack.sh Ambari-Admin-User-Name
```

- 4 Enter the Ambari administrator password at the prompt. A message appears that offers options for removal. Enter one of the options, as appropriate:
 - Enter 1 to remove the SASEP config file only.
 - Enter 2 to remove a specific version of the SASEP service.
 - Enter 3 to remove all versions of the SASEP service.
- 5 You are prompted to restart the Ambari server in order to complete the removal of the SASEP service. To remove the SAS Embedded Process, you must restart the Ambari server.
- 6 Enter **y** to restart the Ambari server. The SASEP service is no longer listed on the Ambari dashboard user interface.

Appendix 6

Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS

<i>About CAS SASHDAT Access to HDFS</i>	231
<i>Supported Hadoop Distributions</i>	232
<i>Overview of Deployment Tasks for HDFS for Existing Hadoop Clusters</i>	232
<i>Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters</i>	232
<i>Review the Passwordless Secure Shell Requirements</i>	233
<i>Kerberos Requirements</i>	233
Overview	233
Host Requirements	233
User Account Requirements for Authentication	234
Java Requirements	234
Kerberos Service and Service Principal Name Requirements	234
Kerberos Keytab Requirements	235
Configure Passwordless SSH to Use Kerberos and GSSAPI	235
<i>Deploying SAS Plug-ins for Hadoop</i>	236
Overview of Deploying SAS Plug-ins for Hadoop	236
Copying the Plug-ins Files to the Hadoop Cluster	236
Installing SAS Plug-ins for Hadoop	236
Configuring HDFS Service Properties	238
Verifying CAS SASHDAT Access to HDFS	242
<i>Uninstalling SAS Plug-ins for Hadoop</i>	242
sashdat-install.sh	242
Cloudera Manager	243
Ambari	244
<i>sashdat-install.sh Reference</i>	245
Overview and Requirements	245
Syntax	246
Options	246
Add Examples	247
Remove Examples	247

About CAS SASHDAT Access to HDFS

Some SAS Viya applications that rely on Hadoop will also require SAS Plug-ins for Hadoop. Supported Hadoop distributions that are modified with SAS Plug-ins for Hadoop enable CAS to write SASHDAT file blocks evenly

across the HDFS file system. This even distribution provides a balanced workload across the machines in the cluster and enables SAS Viya analytic processes to read SASHDAT tables very quickly.

Supported Hadoop Distributions

Before you set up Hadoop, ensure that your Hadoop distribution is supported by SAS Viya. For more information, see [SAS Viya Support for Databases](#).

Note: When an existing Hadoop cluster is shared between SAS 9.4 and SAS Viya, you must deploy the SAS Plug-ins for Hadoop that are delivered with SAS Viya. For details, refer to sections in this appendix. The SAS Viya HDAT Plug-ins are backward compatible with all SAS LASR Analytic Server versions. However, the SAS Plug-ins for Hadoop deployment from SAS LASR is not compatible with the CAS server for SAS Viya.

Overview of Deployment Tasks for HDFS for Existing Hadoop Clusters

During the SAS Viya installation, your CAS software was deployed in one of the following ways:

- the CAS controller and workers were deployed to the nodes on your Hadoop cluster. For an overview of this deployment scenario, see [“Hadoop Integration: CAS SASHDAT Access to HDFS” on page 12](#).
- the CAS controller and the CAS workers were deployed to nodes that are not part of the Hadoop cluster. For an overview of this deployment scenario, see [“Remote Access to HDFS” on page 14](#).

To configure your existing Hadoop cluster:

- 1 Perform the Hadoop pre-deployment tasks. For more information, see [“Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters” on page 232](#).
- 2 Deploy SAS Plug-ins for Hadoop. For more information, see [“Deploying SAS Plug-ins for Hadoop” on page 236](#).
- 3 Verify CAS SASHDAT Access to HDFS. For more information, see [“Verifying CAS SASHDAT Access to HDFS” on page 242](#).

Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters

Here are the requirements for existing Hadoop clusters that are configured for use with the CAS server.

- Verify that the following CAS environment variables are set correctly for your Hadoop environment: `cas.colocation`, `HADOOP_NAMENODE`, and `HADOOP_HOME`.
 - During the SAS Viya installation, values for the CAS environment variables are set in the `vars.yml` file before you run the playbook. After you run the playbook, the settings for the CAS environment variables are stored in the `/opt/sas/viya/config/etc/cas/default/cas.settings` file and the `/opt/sas/viya/config/etc/cas/default/casconfig_deployment.lua` file.
 - For more information about updating CAS environment variables, see [SAS Cloud Analytic Services: Overview](#) in the *SAS Viya 3.3 Administration / SAS Cloud Analytic Services*.

- Each machine in the cluster must be able to resolve the host name of all the other machines in the cluster.
- The time must be synchronized across all machines in the cluster.
- For Cloudera 5 only, all machines that are configured for the CAS server must be in the same role group.
- For Secure Shell (SSH), review the requirements and perform the appropriate tasks in “[Kerberos Requirements](#)” on page 233 .
- For Kerberos, review the requirements and perform the appropriate tasks in “[Kerberos Requirements](#)” on page 233 .

Review the Passwordless Secure Shell Requirements

Here are the passwordless Secure Shell (SSH) requirements:

- To support Kerberos, enable the GSSAPI authentication methods in your implementation of SSH.

Note: If you are using Kerberos, see “[Configure Passwordless SSH to Use Kerberos and GSSAPI](#)” on page 235.
- Passwordless SSH is required for connections from all CAS machines to all machines in the Hadoop cluster. Passwordless SSH is required for the user account that runs the CAS server and for the user accounts that run CAS sessions. By default, the user account that runs the CAS server and CAS sessions is the cas user. Also, passwordless SSH is set up by default.
- If you are running a co-located deployment and use a subset of the machines, passwordless SSH is required for the user account that runs the CAS session. By default, the user account is the cas user, and all CAS nodes are set up with passwordless SSH. Passwordless SSH is also required for the user account that is used to start the CAS server.
- Passwordless SSH is required when a block of data exists on a Hadoop node that exists outside of the Hadoop nodes in the CAS session.

Kerberos Requirements

Overview

SAS Viya does not directly interact with Kerberos. Instead, SAS Viya relies on the underlying operating system and the APIs to handle the requests for tickets, the management of ticket caches, and the authentication of users. For an overview of Kerberos and SAS Viya, refer to [Kerberos](#) in *SAS Viya Administration: SAS Logon Manager*.

Host Requirements

For Kerberos, on the CAS server, the `/etc/hosts` file contains the host names of the machines in the cluster. Each host name is specified in this format:

short-name fully-qualified-domain-name

Here is an example:

```
abchost abchost.abcdomain
```

User Account Requirements for Authentication

Here are the user account requirements for authentication:

- SAS Viya must be configured for pluggable authentication module (PAM) support.
- The default administrative user for the CAS server deployments is the cas local user account. It is recommended that you change this account to a network account so that the local cas user does not generate a credentials cache.

Ensure that the network user account has generated a credentials cache in the location that is defined in your krb5.conf file or in the `/tmp/` directory:

- 1 Log on to CAS Server Monitor as the user. Verify the time at which you logged on.
- 2 Verify that the file has a timestamp that is equal to the time that you logged on to CAS Server Monitor. Here is an example:

```
/tmp/krb5cc_53736
```

Java Requirements

Ensure that Java is set up appropriately.

For the Hadoop cluster:

- The SAS script that is deployed to the Hadoop nodes) requires the JAVA_HOME environment variable to be correctly set for the Hadoop cluster. The SAS script will also be called by the remote connection from the SAS Cloud Analytic Services hosts. The SAS script will run `$HADOOP_HOME/bin/hadoop`. Most Hadoop distributions will define the JAVA_HOME environment variable in `$HADOOP_HOME/libexec/hadoop-config.sh`. Therefore, it is important that this script is validated to ensure that the value of JAVA_HOME is correctly set.
- If you are using Advanced Encryption Standard (AES) encryption with Kerberos, manually add the Java Cryptography Extension `local_policy.jar` file in each place that JAVA_HOME resides in the Hadoop cluster. If you are located outside the United States, you must also manually add the `US_export_policy.jar` file. The addition of these files is governed by the United States import control restrictions.

For the SAS Cloud Analytic Services (CAS) hosts:

- If you are using the Oracle JRE or the IBM JRE, use the two JAR files rather than the existing `local_policy.jar` file and the `US_export_policy.jar` file. These files are in your JRE location, which is typically the `JAVA_HOME/jre/lib/security/` directory. These files can be obtained from the IBM website or the Oracle website.
- If you are using the Oracle JRE or the IBM JRE, it is recommended that you back up the existing `local_policy.jar` file and the `US_export_policy.jar` file in case they ever need to be restored. If you are using the OpenJDK, you do not need to back up these files because they will unlikely need to be restored.

Kerberos Service and Service Principal Name Requirements

A Kerberos principal is a service or user that is known to the Kerberos system. A Kerberos principal is required for SAS Cloud Analytic Services.

In SAS Viya 3.2, this principal must be in the format of a Service Principal Name (SPN). The SPN includes a realm name, which is the capitalization of the domain name. .

Note: In the SPN, REALM must be specified in uppercase characters.

Service Class/Fully Qualified Hostname@REALM

The default *Service Class* is “**sascas**”. A different Service Class can be used if required.

Here is an example:

```
sascas/cascontroller.mycompany.com@MYCOMPANY.COM
```

With an Active Directory Kerberos Key Distribution Center (KDC), the User Principal Name (UPN) for the user must be the same as the SPN because Active Directory allows the initialization of a Kerberos Ticket-Granting Ticket only for a UPN. This restriction does not apply to other Kerberos distributions because the other Kerberos distributions do not distinguish between a UPN and an SPN.

Kerberos Keytab Requirements

- A Kerberos Keytab is required for the principal to be used by SAS Cloud Analytic Service
 - The default Kerberos Keytab location and filename is `/etc/sascas.keytab`. The location can be changed.
 - The keytab file contains the long-term keys for the principal.
 - The keytab file must be available on the CAS Controller for the operating system account running the CAS Controller process. By default, this is the `cas` account.
- To use the GSSAPI for SSH, the end user must be able to obtain a service ticket to connect to the remote machine. The SSH client will request a service ticket for:

```
host/<Fully Qualified Hostname>
```

- The host Service Principal Name must be registered in the Key Distribution Center (KDC). The SSH Daemon will check the default keytab of the operating system for a long-term key associated with the host principal. The default keytab on Linux is `/etc/krb5.keytab`. Therefore, this Kerberos Keytab must contain the long-term keys for the Host principal.

Configure Passwordless SSH to Use Kerberos and GSSAPI

Traditionally, public key authentication in SSH is used in order to meet the passwordless access requirement. For Secure Mode Hadoop, GSSAPI with Kerberos is used as the passwordless SSH mechanism. GSSAPI with Kerberos meets the passwordless SSH requirements and also supplies Hadoop with the credentials that are required for users in order to perform operations in HDFS with SASHDAT files. Certain options must be specified in the SSH daemon configuration file and the SSH client configuration files to support a default configuration of the SSH Daemon (SSHD).

- 1 In the `sshd_config` file, specify the `GSSAPIAuthentication` option:

```
GSSAPIAuthentication yes
```

Note: By default, the SSH Daemon will validate Service Tickets where the SPN matches only the host’s current host name. In multi-homed or systems using a DNS alias, the SSH connection will fail. The SSH Daemon can be configured to validate the Service Ticket using any value within the default Kerberos Keytab. To enable the SSH Daemon to use any value in the Kerberos Keytab, the property `GSSAPIStrictAcceptorCheck` must be set to `no`.

```
GSSAPIStrictAcceptorCheck no
```

- 2 In the `ssh_config` file, specify these options:

```
Host *.domain.net
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

where *domain.net* is the domain name that is used by the machine in the Hadoop cluster.

TIP Even though you can specify `host *`, use of the wildcard is not recommended because it would allow GSSAPI Authentication with any host name.

Deploying SAS Plug-ins for Hadoop

Overview of Deploying SAS Plug-ins for Hadoop

To deploy SAS Plug-ins for Hadoop, follow these steps:

- 1 **Copy** the SAS Plug-ins for Hadoop files to the Hadoop cluster.
- 2 **Install** SAS Plug-ins for Hadoop using one of the following methods:
 - the `sashdat-install.sh` script (supplied by SAS)
Note: The `sashdat-install.sh` script installs SAS Plug-ins for Hadoop on all supported Hadoop distributions.
 - parcel with Cloudera Manager
 - stack with Ambari
- 3 **Configure** HDFS Service properties.

Copying the Plug-ins Files to the Hadoop Cluster

Follow these steps to copy SAS Plug-ins for Hadoop files to the Hadoop cluster:

- 1 On the CAS controller, change to the `/opt/sas/viya/home/SASFoundation/hdatplugins` directory. The directory contains the following files:
 - `sashdat-install.sh`
 - `sashdat-03.03.gz`
- 2 Copy the `sashdat-install.sh` and `sashdat-03.03.gz` files from the CAS controller to the `/tmp` directory of the NameNode host of your Hadoop cluster.
Note: Ensure that the file permissions are set to 0755.
- 3 Go to one of the following sections:
 - “`sashdat-install.sh`” on page 237
 - “Cloudera Manager” on page 237
 - “Ambari” on page 238

Installing SAS Plug-ins for Hadoop

Depending on your Hadoop distribution, you can install SAS Plug-ins for Hadoop by using `sashdat-install.sh`, Cloudera Manager, or Ambari.

sashdat-install.sh

You can use the `sashdat-install.sh` script supplied by SAS to install SAS Plug-ins for Hadoop on all supported Hadoop distributions.

- 1 Make sure that have reviewed “[Overview and Requirements](#)” on page 245.
- 2 Log on to the Hadoop NameNode machine (blade 0) with a UNIX account that has sudo privileges and passwordless SSH access to every machine in the Hadoop cluster.
- 3 Change to the directory that was specified in [Step 2](#), and run the `sashdat-install.sh` script using one of the following commands:

- Deploy with the ‘hdfs’ account querying the hdfs service for the list of machines:

```
sashdat-install.sh -add
```

Here is an example:

```
./sashdat-install.sh -add
```

- Deploy supplying your own list of machines:

```
sashdat-install.sh -add -hostfile host-list-filename
```

Here is an example:

```
./sashdat-install.sh -add -hostfile /tmp/my_hosts
```

- Deploy specifying a different parent installation path:

```
./sashdat-install.sh -add -hdathome /opt/my_path/
```

For more information, see “[sashdat-install.sh Reference](#)”.

Cloudera Manager

You can use Cloudera Manager with parcel to install SAS Plug-ins for Hadoop on all supported Cloudera Hadoop distributions.

- 1 On the CAS controller machine, navigate to the `/opt/sas/viya/home/SASFoundation/hdatplugins/parcel/` directory. Copy the parcel directory to the `tmp` directory of the file system of the host where Cloudera Manager is installed.

Note: Ensure the files in the parcel directory have executable permissions.

- 2 From the `tmp` directory, run the following script:

Note: The user account that you use to run the script must have super user (sudo) or root access.

```
./install_parcel.sh -v distro
```

where `tmp` directory is the file system location where you copied from the SAS Viya installation and `distro` is the following Linux distribution: redhat6.

Here is an example:

```
install_parcel.sh -v redhat6
```

- 3 Select **Y** when asked to restart the Cloudera Manager server.
- 4 Log on to Cloudera Manager as administrator.
- 5 Activate the parcel.
 - a Click **Distribute** to copy the parcel to all nodes.

- b Click **Activate**. You are prompted to restart the cluster or to close the window.
- c When prompted, click **Close**.

CAUTION! Do not restart the cluster.

Ambari

You can use Ambari with stack to install SAS Plug-ins for Hadoop on all supported Cloudera Hadoop distributions.

Note: The following deployment steps assume that the `hdatplugins rpm` package is installed directly on one of the following machines:

CAUTION! When the Hortonworks Hadoop stack is upgraded, the HDATPlugins stack must be deactivated and then reactivated. If the Hortonworks Hadoop level is upgraded in **Express** mode on Ambari, the HDATPlugins stack must be restarted. If the Hortonworks Hadoop level is upgraded in **Rolling** mode, a restart of the HDATPlugins stack is not required.

- the Ambari server
 - a machine in the network that is accessible to the Ambari server
- 1 To launch the script, on the CAS controller machine, navigate to the `/opt/sas/viya/home/SASFoundation/hdatplugins/stack/` directory. Copy the `/stack` directory to the `/tmp` directory of the host where the Ambari Server is installed, and run the following command:

```
./install_hdatplugins.sh Ambari-admin-username
```

After the script finishes running, this message is displayed: You can install the HDATPLUGINS stack now from Ambari Cluster Manager.

- 2 Log on to Ambari. On the Ambari server, deploy the services.
 - a Click **Actions** and select **+ Add Service**. The Add Service Wizard page and the Choose Services panel open.
 - b In the Choose Services panel, select **SASHDAT**. Click **Next**. The Assign Slaves and Clients panel opens.
 - c In the Assign Slaves and Clients panel under **Client**, select all data nodes and all name nodes where you want the stack to be deployed. The Customize Services panel opens. The SASHDAT stack is listed.
 - d Do not change any settings on the Customize Services panel. Click **Next**.

Note: If your cluster is secured with Kerberos, the Configure Identities panel opens. Enter your Kerberos credentials in the `admin_principal` text box and the `admin_password` text box. Click **Next**. The Review panel opens.
 - e Review the information in the panel. If the values are correct, click **Deploy**. The Install, Start, and Test panel opens. After the stack is installed on all nodes, click **Next**. The Summary panel opens.
 - f Click **Complete**. The stacks are now installed on all nodes of the cluster. SASHDAT is displayed on the Ambari dashboard.
 - g On every node, all files in the `/usr/hdp/Hadoop-version/hadoop/bin` directory must be executable with file permissions of 755.

Configuring HDFS Service Properties

Configure HDFS service properties for SAS Plug-ins for Hadoop based on your Hadoop distribution.

Cloudera Hadoop

Note: If Cloudera Manager provides a choice between classic and new layouts, use classic layout.

To use Cloudera Manager to configure HDFS service properties for SAS Plug-ins for Hadoop, follow these steps:

- 1 Log on to Cloudera Manager as an administrator.
- 2 From Cloudera Manager Home, select the HDFS service. Within the HDFS service, select **Configuration** to edit the HDFS configuration properties.

Note: In the following steps, you must edit specific HDFS configuration properties. Locate the property to edit by specifying its name in the search bar.

- a In the `dfs.namenode.plugins` property, add the following line to the plug-in configuration for the NameNode:

```
com.sas.cas.hadoop.NameNodeService
```

- b In the `dfs.datanode.plugins` property, add the following line to the plug-in configuration for the DataNode:

```
com.sas.cas.hadoop.DataNodeService
```

- 3 Navigate to the Service-Wide group. Under Advanced, add the following lines to the HDFS Service Advanced Configuration Snippet (Safety Valve) for `hdfs-site.xml` property.

```
<property><name>com.sas.cas.service.allow.put</name> <value>>true</value></property>
<property><name>com.sas.cas.hadoop.service.namenode.port</name> <value>15452</value></property>
<property><name>com.sas.cas.hadoop.service.datanode.port</name> <value>15453</value></property>
<property><name>dfs.namenode.fs-limits.min-block-size</name> <value>0</value></property>
<property><name>com.sas.cas.hadoop.short.circuit.command</name>
<value>/opt/sas/HDATHome/bin/sascasfd</value></property>
```

Note: You can change the port for the SAS name node and data node plug-ins. This example shows the default ports (15452 and 15453, respectively).

Note: The SAS Plug-ins for Hadoop installation directory, `HDATHome`, is deployed under `/opt/sas/`, by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

- 4 Navigate to the Gateway Default Group. Under Advanced, add the following lines to the HDFS Client Advanced Configuration Snippet (Safety Valve) for `hdfs-site.xml` property.

```
<property><name>com.sas.cas.service.allow.put</name> <value>>true</value></property>
<property><name>com.sas.cas.hadoop.service.namenode.port</name> <value>15452</value></property>
<property><name>com.sas.cas.hadoop.service.datanode.port</name> <value>15453</value></property>
<property><name>dfs.namenode.fs-limits.min-block-size</name> <value>0</value></property>
<property><name>com.sas.cas.hadoop.short.circuit.command</name> <value>/opt/sas/HDATHome/bin/sascasfd</value>
</property>
```

- 5 Navigate to the Service-Wide group. Under Advanced, add the following line to the HDFS Service Environment Advanced Configuration Snippet (Safety Valve) property:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

- 6 Navigate to the Gateway Default Group. Under Advanced, add the following property in HDFS Client Environment Advanced Configuration Snippet (Safety Valve) for `hadoop-env.sh`:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

- 7 Click Cloudera Manager Home, and then select the Yarn service. Within the Yarn service, navigate to the Gateway Default Group by clicking **Configuration and Gateway Default Group** ⇒ **Advanced**. Add the following property in Gateway Client Environment Advanced Configuration Snippet (Safety Valve) for `hadoop-env.sh`:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

- 8 Save changes.
- 9 From the Cloudera Manager home, select the drop-down list for your cluster and select **Deploy Client Configuration**. In the dialog box, select **Deploy Client Configuration**, and then click **Close**.
- 10 Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see [“Verifying CAS SASHDAT Access to HDFS” on page 242](#).

Hortonworks Data Platform Hadoop

To use Ambari to configure Hortonworks HDFS service properties for SAS Plug-ins for Hadoop, follow these steps:

- 1 Log on to Ambari.
- 2 Click **HDFS Service**.
- 3 Choose **Config Section**.
- 4 Click **Advanced**.
- 5 Select **Custom hdfs-site** and add the following properties:

dfs.namenode.plugins

```
com.sas.cas.hadoop.NameNodeService
```

dfs.datanode.plugins

```
com.sas.cas.hadoop.DataNodeService
```

com.sas.cas.service.allow.put

```
true
```

com.sas.cas.hadoop.service.namenode.port

```
15452
```

Note: You can change the port for the SAS name node and data node plug-ins. This example shows the default ports (15452 and 15453, respectively).

com.sas.cas.hadoop.service.datanode.port

```
15453
```

dfs.namenode.fs-limits.min-block-size

```
0
```

com.sas.cas.hadoop.short.circuit.command

```
/opt/sas/HDATHome/bin/sascasfd
```

Note: The SAS Plug-ins for Hadoop installation directory, `HDATHome`, is deployed under `/opt/sas/`, by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

- 6 Save the properties.
- 7 Add the following statement to the **hadoop-env template** of HDFS on the **Advanced hadoop-env** tab, in the section, # Set Hadoop-specific environment variables here:

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

Note: Ensure that the export command occupies a single line.

- 8 Restart all Hortonworks Data Platform (HDP) services and MapReduce services.
- 9 Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS. For details, see “Verifying CAS SASHDAT Access to HDFS” on page 242.

Apache Hadoop

To configure Apache Hadoop HDFS service properties for SAS Plug-ins for Hadoop, follow these steps:

- 1 Define the following properties in `$HADOOP_HOME/etc/hadoop/hdfs-site.xml` and propagate the changes across all nodes in your Hadoop cluster:

Note: The SAS Plug-ins for Hadoop installation directory, `HDATHome`, is deployed under `/opt/sas/` by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

Note: Adjust values appropriately for your deployment. The port numbers should be valid port numbers.

```
<property>
<name>dfs.namenode.plugins</name>
<value>com.sas.cas.hadoop.NameNodeService</value>
</property>
<property>
<name>dfs.datanode.plugins</name>
<value>com.sas.cas.hadoop.DataNodeService</value>
</property>
<property>
<name>com.sas.cas.service.allow.put</name>
<value>>true</value>
</property>
<property>
<name>com.sas.cas.hadoop.service.namenode.port</name>
<value>15452</value>
</property>
<property>
<name>com.sas.cas.hadoop.service.datanode.port</name>
<value>15453</value>
</property>
<property>
<name> dfs.namenode.fs-limits.min-block-size</name>
<value>0</value>
</property>
<property>
<name>com.sas.cas.hadoop.short.circuit.command</name>
<value>/opt/sas/HDATHome/bin/sascasfd</value>
</property>
```

- 2 On every machine in the cluster, in `/etc/hadoop/hadoop-env.sh`, in the section, # Set Hadoop-specific environment variables here, set `HADOOP_CLASSPATH` to the following value:

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

Note: Ensure that the export command occupies a single line.

- 3 Run a SAS test job to verify that data in CAS is saved in the SASHDAT format in HDFS,

Verifying CAS SASHDAT Access to HDFS

- 1 To create the `/test` directory in HDFS, run the following commands as the `hdfs` user. The `/test` directory is used for testing the Hadoop cluster with SAS test jobs.

```
hadoop fs -mkdir /test
hadoop fs -chmod 777 /test
```

- 2 To verify that the software has been successfully deployed, run the following SAS code:

```
cas mysession;
caslib testhdat datasource=(srctype="hdfs") path="/test";
proc casutil;
  load data=sashelp.zipcode;
  save casdata="zipcode" replace;
run;
```

- 3 If you have successfully saved the data in CAS to the SASHDAT format in HDFS, the following message appears in the log output:

```
NOTE: Cloud Analytic Services saved the file zipcode.sashdat to HDFS in caslib
TESTHDAT.
```

Uninstalling SAS Plug-ins for Hadoop

Depending on your Hadoop distribution, you can uninstall SAS Plug-ins for Hadoop by using `sashdat-install.sh`, Cloudera Manager, or Ambari.

`sashdat-install.sh`

You can use the `sashdat-install.sh` script that is supplied by SAS to uninstall SAS Plug-ins for Hadoop on all supported Hadoop distributions.

Note: Starting with the version of SAS Plug-ins for Hadoop (version 03.03) that shipped with SAS Viya 3.3, you can run `sashdat-install.sh -remove` to uninstall an existing deployment of SAS Plug-ins for Hadoop.

- 1 Make sure that have reviewed [“Overview and Requirements” on page 245](#).
- 2 Log on to the Hadoop NameNode machine (blade 0) with a UNIX account that has sudo privileges and passwordless SSH access to every machine in the Hadoop cluster.
- 3 Remove or disable these properties in `$HADOOP_HOME/etc/hadoop/hdfs-site.xml`:

- `<name>dfs.namenode.plugins</name>`
`<value>com.sas.cas.hadoop.NameNodeService</value>`
- `<name>dfs.datanode.plugins</name>`
`<value>com.sas.cas.hadoop.DataNodeService</value>`
- `<name>com.sas.cas.hadoop.service.namenode.port</name>`
`<value>15452</value>`
- `<name>com.sas.cas.hadoop.service.datanode.port</name>`
`<value>15453</value>`
- `<name>com.sas.cas.service.allow.put</name>`


```
<value>>true</value>
```

- `<name>dfs.namenode.fs-limits.min-block-size</name>`
`<value>0</value>`
- `<name>com.sas.cas.hadoop.short.circuit.command</name>`
`<value>/opt/sas/HDATHome/bin/sascasfd</value>`

4 Change to the `/opt/sas/viya/home/SASFoundation/hdatplugins` directory, and run the `sashdat-install.sh` script using one of the following commands:

- Uninstall with the 'hdfs' account querying the hdfs service for the list of machines:

```
sashdat-install.sh -remove
```

Here is an example:

```
./sashdat-install.sh -remove
```

- Uninstall supplying your own list of machines:

```
sashdat-install.sh -remove -hostfile host-list-filename
```

Here is an example:

```
./sashdat-install.sh -remove -hostfile /tmp/my_hosts
```

- Uninstall specifying a different parent installation path:

```
./sashdat-install.sh -remove -hdathome /opt/my_path/
```

For more information, see [“sashdat-install.sh Reference”](#).

Cloudera Manager

Note: These steps for removing SAS Plug-ins for Hadoop apply to the version of SAS Plug-ins for Hadoop (version 03.03) that shipped with SAS Viya 3.3 and later.

- 1 Log on to the Cloudera Manager as an administrator.
- 2 From Cloudera Manager Home, select the HDFS service. Within the HDFS service, select **Configuration** to remove the HDFS configuration properties.

Note: In the following steps, you must remove specific HDFS configuration properties. Locate the property to remove by specifying its name in the search bar.

- a In the `dfs.namenode.plugins` property, remove the following line from the plug-in configuration for the NameNode:

```
com.sas.cas.hadoop.NameNodeService
```

- b In the `dfs.datanode.plugins` property, remove the following line from the plug-in configuration for the DataNode:

```
com.sas.cas.hadoop.DataNodeService
```

- 3 Navigate to the Service-Wide group. Under Advanced, remove the following lines from the HDFS Service Advanced Configuration Snippet (Safety Valve) for `hdfs-site.xml` property.

Note: The SAS Plug-ins for Hadoop installation directory, `HDATHome`, is deployed under `/opt/sas/` by default. If you have chosen a different installation path, use the different path where necessary in this step and in later steps.

- 4 Navigate to the Gateway Default Group. Under Advanced, remove the following lines from the HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property.

```
<name>com.sas.cas.service.allow.put</name>
<value>true</value>
<name>com.sas.cas.hadoop.service.namenode.port</name>
<value>15452</value>
<name>com.sas.cas.hadoop.service.datanode.port</name>
<value>15453</value>
<name>dfs.namenode.fs-limits.min-block-size</name>
<value>0</value>
<name>com.sas.cas.hadoop.short.circuit.command</name>
<value>/opt/sas/HDATHome/bin/sascasfd</value>
```

- 5 Navigate to the Gateway Default Group. Under Advanced, remove the following lines from the HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property.

```
<name>com.sas.cas.service.allow.put</name>
<value>true</value>
<name>com.sas.cas.hadoop.service.namenode.port</name>
<value>15452</value>
<name>com.sas.cas.hadoop.service.datanode.port</name>
<value>15453</value>
<name>dfs.namenode.fs-limits.min-block-size</name>
<value>0</value>
<name>com.sas.cas.hadoop.short.circuit.command</name>
<value>/opt/sas/HDATHome/bin/sascasfd</value>
```

- 6 Navigate to the HDFS Environment Client Safety Valve. Remove the following property from the HDFS Service Environment Advanced Configuration Snippet (Safety Valve) Server-wide:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

- 7 Navigate to the Gateway Default Group. Remove the following property from the HDFS Client Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh Gateway Default Group:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

- 8 From Cloudera Manager Home, select the YARN service. Within the YARN service, navigate to the Gateway Default Group. Remove the following property from the Client Safety Valve Gateway Client Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh Gateway Default Group:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```

- 9 From the Menu bar, select **Hosts** ⇒ **Parcels**.

10 Select the SASHDAT parcel.

11 Deactivate the SASHDAT parcel.

12 Remove the SASHDAT parcel.

13 Delete the SASHDAT parcel.

14 When prompted, click **Close**.

Ambari

Note: These steps for removing SAS Plug-ins for Hadoop apply to the version of SAS Plug-ins for Hadoop (version 03.03) that shipped with SAS Viya 3.3 and later.

Note: To remove the stack, root or passwordless sudo access is required.

- 1 Log on to Ambari as an administrator, and stop all HDP services.
- 2 Delete the custom `hdfs-site.xml` file that contains the SAS Plug-ins for Hadoop properties (such as, the `com.sas.cas.*` properties).
- 3 Remove the following statement from the **hadoop-env template** of HDFS on the **Advanced hadoop-env** tab, in the section, # Set Hadoop-specific environment variables here:


```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/sas/HDATHome/lib/*
```
- 4 On the CAS controller machine, navigate to the `/opt/sas/viya/home/SASFoundation/hdatplugins/stack/` directory and run the following command to delete the stack:


```
/delete_stack.sh Ambari-Admin-User-Name
```
- 5 At the prompt, enter the Ambari administrator password. A message appears that offers options for removal.
- 6 Enter one of the following options:
 - Enter 1 to remove only the SASHDAT service.
 - Enter 2 to remove a specific version of the SASHDAT service.
 - Enter 3 to remove all versions of the SASHDAT service.

To complete the removal of the SASHDAT service, you are prompted to restart the Ambari server.
- 7 Enter **y** to restart the Ambari server.

The SASHDAT service is no longer listed on the Ambari dashboard.

sashdat-install.sh Reference

Overview and Requirements

The `sashdat-install.sh` script enables you to deploy SAS Plug-ins for Hadoop on a [SAS Viya Support for Databases](#). The script provides an alternative to Cloudera parcels and Ambari stacks.

The UNIX account with which the script is run requires sudo privileges and passwordless SSH access to every machine in the Hadoop cluster when adding and removing SAS Plug-ins for Hadoop. No sudo access is required when you are checking whether the plug-ins are correctly installed on all data nodes.

When adding or removing SAS Plug-ins for Hadoop, the `sashdat-install.sh` script attempts to query the Hadoop configuration to automatically discover the machine name for all of the nodes in the cluster. In order to query the `hdfs` service for machine names, the script assumes that your site uses the default Hadoop user account 'hdfs.' You can provide a different Hadoop account with execution permissions for the `hdfs` command, or, provide your own list of machine names.

You must provide a list of machine names under these conditions:

- the `hdfs` service is down.
- you are adding new machines to the cluster.
- you want to use a list of machines that is different from what is in the Hadoop configuration.

Syntax

- Add SAS Plug-ins for Hadoop:

```
sashdat-install.sh -add <-hostfile host-list-filename | -host "host-list"> <-  
hdfsuser user-ID> <-hdathome parent-installation-path>
```

- Remove SAS Plug-ins for Hadoop:

```
sashdat-install.sh -remove <-hostfile host-list-filename | -host host-list> <-  
hdfsuser user-ID> <-hdathome parent-installation-path>
```

- Check whether SAS Plug-ins for Hadoop is properly installed:

```
sashdat-install.sh -x -check <-hdathome parent-installation-path>
```

- Query the version of SAS Plug-ins for Hadoop:

```
sashdat-install.sh -version <-hdathome parent-installation-path>
```

Options

-add

installs SAS Plug-ins for Hadoop on all machines in the cluster, or on a user-supplied the list of machines.

Requirements The UNIX user account with which you run the `sashdat-install.sh` script must have sudo permissions and passwordless SSH access to every machine in the Hadoop cluster.

The script assumes that your site uses the default Hadoop user account, 'hdfs,' with which the script automatically retrieves the list of data nodes from the Hadoop configuration. If your site does not use the 'hdfs' user account, then you must use the `-hdfsuser user-ID` option to provide a valid Hadoop user account with execution permissions for the `hdfs` command. Or, you can provide your own list of machines using either the `-hostfile` or `-host` option.

-hdathome *parent-installation-path*

(optional) specifies a custom parent installation path for the plug-ins instead of the default `/opt/sas` path. The subdirectory `HDATHome` will be created under the specified `-hdathome` path.

-hdfsuser *user-ID*

(optional) specifies the user ID that has execution permissions for `hdfs` to run the `hdfs dfsadmin -report` command to retrieve the machine names of the nodes in the Hadoop cluster.

The `-hdfsuser` option is not required when the default Hadoop account, 'hdfs,' is present, or when you supply your own list of machines in the Hadoop cluster using the `-hostfile` or `-host` option.

-hostfile *host-list-filename*

(optional) specifies the full path of the file that contains the list of machine names for all of the cluster nodes on which the plug-ins are installed or removed.

Requirement The host list file must contain one fully qualified machine name per line.

Example

```
machine001.example.com  
machine002.example.com  
machine003.example.com  
machine004.example.com
```

-host "*host-list*"

(optional) specifies the list of machine names for all of the cluster nodes on which the plug-ins are installed or removed

Requirement If you specify more than one machine, the names must be separated by spaces or commas. The wildcard character, the asterisk (*), is allowed.

Examples `-host server1 server2 server3`

`-host blue1,blue2,blue3`

`-host bluesvr*`

-remove

removes the plug-ins on all machines in the cluster, or on a user-supplied the list of machines.

Requirements The UNIX user account with which you run the sashdat-install.sh script must have sudo permissions and passwordless ssh access to every machine in the Hadoop cluster.

The script assumes that your site uses the default Hadoop user account, 'hdfs,' with which the script automatically retrieves the list of data nodes from the Hadoop configuration. If your site does not use the 'hdfs' user account, then you must use the `-hdfsuser user-ID` option to provide a valid Hadoop user account with execution permissions for the `hdfs` command. Or, you can provide your own list of machines using either the `-hostfile` or `-host` option.

-version <-hdathome parent-installation-path>

displays the version of the plug-ins that are installed.

Example `./sashdat-install.sh -version`

-x -check <-hdathome parent-installation-path>

checks whether the plug-ins are installed correctly on all data nodes.

Tip You can specify the hosts for which you want to check the plug-ins by using the `-hostfile` or `-host` option.

Example `./sashdat-install.sh -x -check`

Add Examples

This section demonstrates various ways to use sashdat-install.sh to add SAS Plug-ins for Hadoop to your supported Hadoop cluster:

Add using the 'hdfs' account to query Hadoop for a list of machines:

`./sashdat-install.sh -add`

Add using the 'my-hdfs' account to query Hadoop for a list of machines:

`./sashdat-install.sh -add -hdfsuser my-hdfs`

Add specifying a user-supplied list of machines:

`./sashdat-install.sh -add -hostfile /tmp/my_hosts`

Add specifying a user-supplied installation path:

`./sashdat-install.sh -add -hdathome /var/my_sasplugins/`

Remove Examples

This section demonstrates various ways to use sashdat-install.sh to remove SAS Plug-ins for Hadoop to your supported Hadoop cluster:

Remove using the 'hdfs' account to query Hadoop for a list of machines:

```
./sashdat-install.sh -add
```

Remove using the 'my-hdfs' account to query Hadoop for a list of machines:

```
./sashdat-install.sh -remove -hdfsuser my-hdfs
```

Remove specifying a user-supplied list of machines:

```
./sashdat-install.sh -remove -hostfile /tmp/my_hosts
```

Remove specifying a user-supplied installation path:

```
./sashdat-install.sh -remove -hdathome /var/my_sasplugins/
```

Appendix 7

Teradata Deployment: Configuring SAS In-Database Technologies

<i>Prerequisites</i>	249
<i>Overview of the In-Database Deployment Package for Teradata</i>	250
<i>Connections from SAS 9.4 Clients</i>	250
<i>Teradata Installation and Configuration</i>	250
<i>Installing the SAS In-Database Deployment Package for Teradata</i>	251
Copy the SAS In-Database Deployment Packages for Teradata to the Server Machine	251
Install the SAS In-Database Deployment Package with the Teradata Parallel Upgrade Tool	251
Verify the Connection to Teradata	252
Install the Support Functions for the SAS Embedded Process	253
<i>(Optional) Deploy TLS Certificates</i>	253
<i>Configuring SAS Data Quality Accelerator for Teradata</i>	254
Overview	254
SAS In-Database Deployment Package for Teradata	254
Using the dq_install.sh Script	254
Using the dq_grant.sh Script	255
Locating the QKB	255
Packaging the QKB	255
Installing the QKB Package File with the Teradata Parallel Upgrade Tool	256
Validating the Accelerator Installation	257
Troubleshooting the Accelerator Installation	258
Updating and Customizing a QKB	259
Removing SAS Data Quality Accelerator from the Teradata Database	259
Using the dq_uninstall.sh Script	260

Prerequisites

The SAS in-database deployment package requires the following:

- version 15.10 of the Teradata client and server environment.
- the CAS controller and each CAS worker node must have an IP address that can be routed to externally from the SAS Embedded Process nodes.
- approximately 200 MB of disk space in the /opt file system on each Teradata Trusted Parallel Appliance (TPA) node.

Overview of the In-Database Deployment Package for Teradata

SAS In-Database Technologies Teradata for SAS Viya includes SAS Data Connect Accelerator, SAS Data Quality Accelerator for Teradata and the SAS Embedded Process for Teradata, as well as a security configuration file. This section describes how to install and configure the in-database deployment package for Teradata.

The SAS Embedded Process is a SAS server process that runs within Teradata to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that are installed on your Teradata system.

If you are using SAS Data Connect Accelerator for Teradata and you want to secure data transfer between your Teradata cluster and CAS, use the security configuration file.

Note: If you are adding additional nodes, the version of the SAS Embedded Process must be the same for the existing and new nodes.

Note: In addition to installing the in-database deployment package for Teradata, you must also install a set of SAS Embedded Process functions in the Teradata database. The functions package for the SAS Embedded Process is downloadable from Teradata. For more information “[Install the Support Functions for the SAS Embedded Process](#)” on page 253.

Connections from SAS 9.4 Clients

The following SAS 9.4 clients can connect to a Teradata Server that has installed the SAS Viya version of SAS Embedded Process for Teradata:

- SAS Analytics Accelerator for Teradata
- SAS High-Performance Analytics
- SAS In-Database Code Accelerator for Teradata
- SAS LASR
- SAS Scoring Accelerator for Teradata

Teradata Installation and Configuration

To install and configure the SAS In-Database Technologies for Teradata:

- 1 Install the in-database deployment package. For more information, see “[Installing the SAS In-Database Deployment Package for Teradata](#)” on page 251.
- 2 Install the support functions for the SAS Embedded Process. For more information, see “[Install the Support Functions for the SAS Embedded Process](#)” on page 253.
- 3 (Optional) If you are using SAS Data Connect Accelerator, and you want to secure the data transfer between your Teradata or Hive cluster and CAS, you must enable security. For more information, see “[\(Optional\) Deploy TLS Certificates](#)” on page 253.

Installing the SAS In-Database Deployment Package for Teradata

Copy the SAS In-Database Deployment Packages for Teradata to the Server Machine

- 1 Locate the SAS in-database deployment package file, `sepcoretera-14.00000-n.x86_64.rpm`. *n* is a number that indicates the latest version of the file.
- 2 Navigate to the `/opt/sas/viya/home/share/ep` directory. This directory was created when you installed SAS Viya.
- 3 Locate the `sepcoretera-14.00000-n.x86_64.rpm` file. *n* is a number that indicates the latest version of the file.
- 4 Copy this file to a temporary directory on the Teradata machine. Make sure that you copy the file to the server machine according to the procedures that are used at your site. Here is an example of a secure copy command.

```
scp sepcoretera-14.00000-n.x86_64.rpm root@teramach1:/temporary-dir
```

This package file is readable by the Teradata Parallel Upgrade Tool.

Install the SAS In-Database Deployment Package with the Teradata Parallel Upgrade Tool

This installation should be performed by a Teradata systems administrator in collaboration with Teradata Customer Services. A Teradata Change Control is required when a package is added to the Teradata server. Teradata Customer Services has developed change control procedures for installing the SAS in-database deployment package.

The steps assume knowledge about the Teradata Parallel Upgrade Tool and your environment. For more information about using the Teradata Parallel Upgrade Tool, see the *Parallel Upgrade Tool (PUT) Reference*, which is included in the Teradata Online Publications site at <http://www.info.teradata.com/GenSrch/eOnLine-Srch.cfm>. On this page, search for “Parallel Upgrade Tool” and download the appropriate document for your system.

Follow these steps to use the Teradata Parallel Upgrade Tool to install the SAS in-database deployment package.

Note: The Teradata Parallel Upgrade Tool prompts are subject to change as Teradata enhances its software.

- 1 Locate the in-database deployment packages on your server machine. The location must be accessible from at least one of the Teradata nodes. For more information, see [“Copy the SAS In-Database Deployment Packages for Teradata to the Server Machine” on page 251](#).
- 2 Start the Teradata Parallel Upgrade Tool.
- 3 Be sure to select all Teradata TPA nodes for installation, including Hot Stand-By Nodes.
- 4 If Teradata Version Migration and Fallback (VM&F) is installed, you might be prompted about whether to use VM&F. If you are prompted, choose Non-VM&F installation.


- 5 If the installation is successful, `sepcoretera-14.00000-n.x86_64` is displayed. *n* is a number that indicates the latest version of the file.

Alternatively, you can manually verify that the installation is successful by running these commands from the shell prompt.

```
psh "rpm -q -a" | grep sepcoretera
```

Verify the Connection to Teradata

To verify that SAS Data Connector to Teradata and SAS Data Connect Accelerator for Teradata were successfully deployed:

- 1 Sign on to SAS Studio:
 - a Open SAS Studio from a URL with the following format: `https://http-proxy-host-name/SASStudio`
 - b Enter the credentials for your operating system account.
- 2 Start a CAS session:
 - a In the navigation pane, open the **Snippets** section.
 - b Select **Snippets** ⇒ **Cloud Analytic Services** .
 - c Right-click **New CAS Session** and select **Open**. The snippet opens in the code editor.
 - d In the toolbar, click  to run the new CAS session code.

- 3 From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS to Teradata LIBNAME:

```
libname tlib teradata server="teradata-host-name" database="teradata-database-name" user="user-ID"
password="user-Password";
```

If SAS/ACCESS to Teradata was successfully deployed, the execution of the libname will return without error.

- 4 From SAS Studio, edit and run the following SAS code to verify the SAS Data Connector to Teradata:

```
caslib tdlb datasource=(srctype="teradata", dataTransferMode="auto", username="user-ID",
password="user-Password",
server="teradata-host-name", database="teradata-database-namee");
```

```
proc casutil;
list files incaslib="tdlib";
run;
```

If the validation of the libname or data connector failed, error was return on the execution of the libname or no table information was returned for the data connector, you should perform the configuration steps again.

- 5 From SAS Studio, edit and run the following SAS code to verify the SAS Data Connect Accelerator for Teradata:

```
caslib teraplib datasource=(srctype="teradata", dataTransferMode="parallel" username="<user ID>",
password="<password>", server="<Teradata host name>", database="<Teradata database name>");
```

```
proc casutil;
list files incaslib="teraplib";
run;
```

If the data connector was successfully deployed, the results are the names of the tables in Teradata. If you do not see table names that you recognize, you should perform the configuration steps again.

Install the Support Functions for the SAS Embedded Process

The support function (sasepfunc) package for the SAS Embedded Process includes stored procedures that generate SQL to interact with the SAS Embedded Process. The support function package also includes functions that load the SAS program and other run-time control information into shared memory. The setup script for the support function package creates the SAS_SYSFNLIB database and the fast path functions in TD_SYSFNLIB.

The support function package is available from the Teradata Software Server. For access to the package that includes the installation instructions, contact your local Teradata account representative or the Teradata consultant that supports your SAS and Teradata integration activities.

CAUTION! If you are using Teradata 15, you must drop the SAS_SYSFNLIB.SASEP_VERSION function to disable the Teradata Table Operator (SASTbOp). Otherwise, your output can contain missing rows or incorrect results. To drop the function, enter the following command:

```
drop function SAS_SYSFNLIB.SASEP_VERSION
```

This issue is fixed in the Teradata maintenance release 15.00.04.

(Optional) Deploy TLS Certificates

If you are using a SAS Data Connect Accelerator, the data that is transferred between the data provider and the CAS server is not encrypted by default. However, SAS Viya supports TLS encryption between the data provider and the CAS server. When Viya 3.3 is deployed, TLS is enabled and configured on the CAS server (server side). The deployment process provides a default level of encryption for data in motion. Options are set in the vars.yml file and are defined in the casconfig_deployment.lua file. These settings enable data connector encryption and specify the location of the TLS private key and the password.

However, you must take additional steps to enable encryption on the data provider. The prerequisites and the process for enabling TLS encryption on the data provider are different for each data provider. The first step is to deploy the TLS certificates across all nodes in the cluster.

- 1 On the CAS controller machine, locate the TLS certificates in the trustedcerts.pem file in the `/opt/sas/viya/config/etc/SASSEcurityCertificateFramework/cacerts/` directory.
- 2 Copy the trustedcerts.pem file to the SAS Embedded Process `security/certs` directory on Teradata.
- 3 To complete the deployment of TLS encryption, you also must update a dcsecurity.properties file. Both the .pem file and the dcsecurity.properties file must then be copied to all nodes on the server. For more information on how to complete the deployment, see [Encrypt Data Transfer When Using the SAS Data Connect Accelerator in Encryption](#) in *SAS Viya: Data in Motion*.

Configuring SAS Data Quality Accelerator for Teradata

Overview

You can use SAS Data Quality technology in the Teradata database via the SAS Data Quality Accelerator for Teradata. To use SAS Data Quality Accelerator for Teradata, you must perform the following tasks after deploying the SAS In-Database Technologies for Teradata (SAS Embedded Process):

- install SAS data quality stored procedures in the Teradata database
- deploy a SAS Quality Knowledge Base (QKB) in the Teradata database

The SAS In-Database Technologies deployment provides shell scripts that enable you to install and manage the data quality stored procedures within the Teradata database. In addition, it contains a shell script that enables you to package the QKB for deployment inside the Teradata database.

The QKB is a collection of files that store data and logic that support data management operations. SAS software products reference the QKB when performing data management operations on your data.

Each Teradata node needs approximately 8 GB for the QKB.

SAS In-Database Deployment Package for Teradata

The SAS In-Database deployment package for Teradata (sepcoretera) installs three scripts in the `/opt/sas/spre/home/SASFoundation/install/pgm` directory of the Teradata database server:

- a stored procedure creation script named `dq_install.sh`
- a user authorization script named `dq_grant.sh`
- a stored procedure removal script named `dq_uninstall.sh`

Run the `dq_install.sh` script to create the data quality stored procedures in the Teradata database and the `dq_grant.sh` script to grant users permission to execute the data quality stored procedures.

The `dq_uninstall.sh` script is provided to enable you to remove the data quality stored procedures from the database. You must remove any data quality stored procedures that have already been installed from the Teradata database before upgrading or re-installing either SAS Data Quality Accelerator for Teradata or the SAS Embedded Process.

Note: All three scripts must be run as the root user.

Using the `dq_install.sh` Script

The `dq_install.sh` script is located in the `/opt/sas/spre/home/SASFoundation/install/pgm` directory of the Teradata database server.

The `dq_install.sh` script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the `DBC_PASS=`, `DBC_SRVR=`, and `DBC_USER=` variables.

Running `dq_install.sh` puts the data quality stored procedures into the `SAS_SYSFNLIB` database and enables the accelerator functionality.

Here is the syntax for executing `dq_install.sh`:

```
./dq_install.sh <-l log-path>
```

log-path

specifies an alternative name and location for the dq_install.sh log. When this parameter is omitted, the script creates a file named dq_install.log in the current directory.

The next step in the installation is to grant users permission to execute the stored procedures.

Using the dq_grant.sh Script

The dq_grant.sh shell script is provided to enable the Teradata system administrator to grant users authorization to the data quality stored procedures. The dq_grant.sh script is located in the `/opt/sas/spre/home/SASFoundation/install/pgm` directory of the Teradata database server.

The dq_grant.sh script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_SRVR=, DBC_USER=, and DBC_PASS= variables.

Here is the syntax for executing dq_grant.sh:

```
./dq_grant.sh <-l log-path> user-name
```

log-path

specifies an alternative name and location for the dq_grant.sh log. When this parameter is omitted, the script creates a file named dq_grant.log in the current directory.

user-name

is the user name to which permission is being granted. The target user account must already exist in the Teradata database.

The authorizations granted by dq_grant.sh supplement existing authorizations that the target user account already has in the Teradata database.

You can verify that authorization was granted successfully for a user by logging on to the database as the user and issuing the following command in a Basic Teradata Query (BTEQ) session:

```
call sas_sysfnlib.dq_debug();
```

The command will fail if the user does not have permission. Otherwise, it will have no effect.

The data quality stored procedures are not yet ready to use. A QKB must be installed in the Teradata database for the data quality stored procedures to be usable.

Locating the QKB

The QKB is located in the following directory:

```
UNIX: /opt/sas/spre/home/share/refdata/qkb
```

Packaging the QKB

Before a QKB can be deployed in the Teradata database, you must package it into an .rpm file. An .rpm file is a file that is suitable for installation on Linux systems that use RPM package management software. SAS Data Quality Accelerator for Teradata provides the qkb_pack script to package the QKB into an .rpm file.

qkb_pack is created in the following directory during deployment:

UNIX

```
/opt/sas/spre/home/SASFoundation/install/pgm
```

You must execute qkb_pack from the `/opt/sas/spre/home/SASFoundation/install/pgm` location.

Here is the syntax for executing qkb_pack:

Example Code A7.1 UNIX:

```
./qkb_pack.sh qkb-dir out-dir
```

qkb-dir

specify the path to the QKB. Use the name of the QKB's root directory. Typically, the root directory is found at the following directories:

UNIX: `/opt/sas/spre/home/share/refdata/qkb/product-identifier/product-version`

out-dir

specify the directory where you want the package file to be created.

Here is an example of a command that you might execute in order to package a SAS QKB for Contact Information that resides on a UNIX computer.

```
cd /opt/sas/spre/home/SASFoundation/install/pgm
./qkb_pack.cmd
/opt/sas/spre/home/share/refdata/qkb/CI/28 /tmp
```

The package file that is created in /tmp will have a name in the following form:

```
sasqkb_product-version-timestamp.noarch.rpm
```

product

is a two-character product code for the QKB, such as CI (for Contact Information) or PD (for Product Data).

version

is the version number of the QKB.

timestamp

is a UNIX datetime value that indicates when qkb_pack was invoked. A UNIX datetime value is stored as the number of seconds since January 1, 1970.

noarch

indicates that the package file is platform-independent.

Here is an example of an output filename representing the QKB for Contact Information 28:

```
sasqkb_ci-28.0-1474057340608.noarch.rpm
```

After running qkb_pack, put the sasqkb package file on your Teradata database server in a location where it is available for both reading and writing. The package file must be readable by the Teradata Parallel Upgrade Tool. You need to move this package file to the server machine in accordance with procedures used at your site.

Installing the QKB Package File with the Teradata Parallel Upgrade Tool

This installation should be performed by a Teradata systems administrator in collaboration with Teradata Customer Services. A Teradata Change Control is required when a package is added to the Teradata server. Teradata Customer Services has developed change control procedures for installing the SAS in-database deployment package.

The steps assume full knowledge of the Teradata Parallel Upgrade Tool and your environment. For more information about using the Teradata Parallel Upgrade Tool, see the Parallel Upgrade Tool (PUT) Reference, which is on the Teradata Online Publications site located at <http://www.info.teradata.com/GenSrch/eOnLine-Srch.cfm>. On this page, search for "Parallel Upgrade Tool" and download the appropriate document for your system.

The following section explains the basic steps to install the sasqkb package file using the Teradata Parallel Upgrade Tool.

Note: It is not necessary to stop and restart the Teradata database when you install a QKB. However, if the SAS Embedded Process is running, you must stop it and then re-start it after the QKB is installed. It is also necessary

to stop and restart the SAS Embedded Process for QKB updates. For information about stopping and restarting the SAS Embedded Process, see [Controlling the SAS Embedded Process](#) in *SAS 9.4 and SAS Viya 3.2 Programming Documentation / In-Database Products: User's Guide*.

- 1 Start the Teradata Parallel Upgrade Tool.
- 2 Be sure to select all Teradata TPA nodes for installation, including Hot Stand-By nodes.
- 3 If Teradata Version Migration and Fallback (VM&F) is installed, you might be prompted about whether to use VM&F. If you are prompted, choose Non-VM&F installation.

You can verify that the QKB installation was successful by running the following command from the shell prompt on one of the Teradata nodes.

```
psh "rpm -q -a" | grep sasqkb
```

If the installation was successful, the command returns the version number of the sasqkb package. Failure to return an output indicates that a library of that name could not be found.

The QKB is installed in the `/opt/qkb/default` directory of each Teradata node.

You are now ready to validate the data quality stored procedures for use.

Validating the Accelerator Installation

Here is a simple BTEQ program that can be used to verify that the SAS Data Quality Accelerator for Teradata is operational.

The code first lists the locales that are installed in the QKB. Then it creates a table named `Dqacceltest` and executes the `DQ_GENDER()` stored procedure on the table. Before running the example, substitute a real value for the `output_table_1`, `output_table_2`, and `locale` variables throughout the program. For `locale`, use one of the values returned by the `DQ_LIST_LOCALES()` stored procedure. This example assumes that the SAS Data Quality Accelerator for Teradata is using the QKB for Contact Information.

The example also sets the SAS Data Quality Accelerator `DQ_OVERWRITE_TABLE` option to create temporary output tables in the SAS Data Quality Accelerator session. If you run the example again in the same SAS Data Quality Accelerator session, the new output tables overwrite any existing output tables and the output tables are automatically discarded at the end of the session. The `DROP TABLE` statement removes table `Dqacceltest` from your database.

```
call sas_sysfnlib.dq_list_locales('mydb.output_table_1');
select * from mydb.output_table_1;

call sas_sysfnlib.dq_set_option('DQ_OVERWRITE_TABLE', '1');

create table mydb.dqacceltest (id_num integer, name varchar(64))
  unique primary index(id_num);

insert into mydb.dqacceltest (id_num, name) values (1, 'John Smith');
insert into mydb.dqacceltest (id_num, name) values (2, 'Mary Jones');

call sas_sysfnlib.dq_gender('Name', 'mydb.dqacceltest', 'name', 'id_num',
  'mydb.output_table_2', 'locale');

select gender from mydb.output_table_2;
drop table mydb.dqacceltest;
```

If the request was successful, the `SELECT` statement produces an output table that contains the following:

Gender

M

F

Troubleshooting the Accelerator Installation

Q. I ran the sample code and the output tables were not created in my user schema. What now?

A. The stored procedures can fail if one or more of the following conditions are true:

- The request specifies an output location to which the user does not have Write permission. Verify that you have access to the database that is specified in the *output_table* parameters.
- The data quality stored procedures are not installed correctly. Verify that the stored procedures are in the SAS_SYSFNLIB database by executing the following command in BTEQ:

```
select TableName from dbc.tables where databasename='SAS_SYSFNLIB'
and tablename like 'dq_%';
```

The command should return a list similar to the following:

Note: This is an incomplete list.

```
TableName
-----
dq_set_qkb
dq_match_parsed
dqi_drop_view_if_exists
dqi_get_option_default
dq_debug
dq_propercase
dqi_tbl_dbname
dqi_drop_tbl_if_exists
dq_set_option
dqt_error
dq_standardize
dq_standardize_parsed
dq_debug2
dqi_invoke_table
dq_lowercase
dq_set_locale
dq_extract
dq_uppercase
dq_list_bindings
dqi_replace_tags
dq_list_defns
dqi_call_ep
dqi_get_bool_option
dqi_gen_toktxt
dqt_codegen
dq_match
dq_parse
dqt_trace
dq_pattern
dqi_clear_tok_tbls
dqt_tokname_tmp
dq_format
```



```

dq_list_locales
dq_invoke_scalar
dq_invoke_prepared
dq_bind_token
dq_gender

```

If the procedures are absent, run the `dq_install.sh` script again, making sure that you are logged in as Teradata system administrator.

- Permission to the data quality stored procedures is not granted correctly. Verify that the target user name submitted to the `dq_grant.sh` script is a valid user account in the Teradata database. Verify that the database server and granter information in the `dq_grant.sh` shell script is correct.
- The QKB is not in the correct location. Look for subdirectories similar to the following in the `/opt/qkb/default` directory on the Teradata nodes: `chopinfo`, `grammar`, `locale`, `phonetx`, `regexlib`, `scheme`, and `vocab`.
- Your SQL request does not use the Teradata dialect. The stored procedures are invoked with the `CALL` keyword from any product that supports the Teradata SQL dialect. When you submit the data quality stored procedures in the SAS SQL procedure using explicit pass-through, the database connection is made in ANSI mode by default. You must specify the `MODE=` option to switch to Teradata mode. Refer to the SAS/ACCESS Interface to Teradata documentation for more information about the `MODE=` option. Refer to the appropriate documentation about how to set Teradata mode in other client programs.

Updating and Customizing a QKB

SAS provides regular updates to the QKB. It is recommended that you update your QKB each time a new one is released. For a listing of the latest enhancements to the QKB, see “What’s New in SAS Quality Knowledge Base.” The What’s New document is available on the [Quality Knowledge Base \(QKB\) for SAS and DataFlux Documentation](#) site on support.sas.com.

Check the What’s New document for each QKB to determine which definitions have been added, modified, or deprecated, and to learn about new locales that might be supported. Contact your SAS software representative to order updated QKBs and locales. To deploy a new QKB, follow the steps in “[Packaging the QKB](#)” on page 255 and “[Installing the QKB Package File with the Teradata Parallel Upgrade Tool](#)” on page 256. The accelerator supports one QKB in the Teradata database.

The standard definitions in the QKB are sufficient for performing most data quality operations. However, you can use the Customize feature of DataFlux Data Management Studio to modify the QKB definitions to meet specific needs.

If you want to customize your QKB, SAS recommends that you customize your QKB on a local workstation before copying it to the Teradata database for deployment. When updates to the QKB are required, merge your customizations into an updated QKB locally, and copy the updated, customized QKB to the Teradata node. This enables you to deploy a customized QKB to the Teradata database using the same steps that you would use to deploy a standard QKB. Copying your customized QKB from a local workstation into your cluster also means that you will have a backup of the QKB on your local workstation. See the online Help provided with your SAS Quality Knowledge Base for information about how to merge any customizations that you have made into an updated QKB.

Removing SAS Data Quality Accelerator from the Teradata Database

Before you can upgrade, re-install, or permanently remove SAS Data Quality Accelerator for Teradata or the SAS Embedded Process, you must remove any existing data quality stored procedures from the Teradata database. The stored procedures are removed from the Teradata database by using the `dq_uninstall.sh` script. For more information about this script, see “[Using the dq_uninstall.sh Script](#)” on page 260.

It is not necessary to remove the QKB when upgrading or re-installing software. QKB deployment steps automatically overwrite an older version of the QKB when you install a new one. For information to replace the QKB, see “[Updating and Customizing a QKB](#)” on page 259 and “[Locating the QKB](#)” on page 255.

When you are permanently removing SAS Data Quality Accelerator for Teradata from the Teradata database server, follow whatever procedure is appropriate at your site for removing the QKB. The Teradata administrator also needs to remove data quality authorizations from the Teradata database in accordance with site procedures.

Using the dq_uninstall.sh Script

Note: To stop the embedded process, see [Controlling the SAS Embedded Process](#) in *SAS 9.4 and SAS Viya 3.2 Programming Documentation / In-Database Products: User’s Guide*. Stopping the SAS Embedded Process ensures that none of the accelerator files are locked when dq_uninstall.sh attempts to remove them.

The accelerator provides the dq_uninstall.sh shell script for removing the data quality stored procedures from the Teradata database. The dq_uninstall.sh script is located in the `/opt/sas/spre/home/SASFoundation/install/pgm` directory of the Teradata database server.

The dq_uninstall.sh script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_PASS=, DBC_SRVR=, and DBC_USER= variables.

Here is the syntax for executing dq_uninstall.sh:

```
./dq_uninstall.sh <-l log-path>
```

log-path

specifies an alternative name and location for the dq_uninstall.sh log. When this parameter is omitted, the script creates a file named dq_uninstall.log in the current directory.

Running dq_uninstall.sh disables the SAS Data Quality Accelerator for Teradata functionality and removes the data quality stored procedures from the database.

Appendix 8

Troubleshooting

Troubleshooting SAS Viya	261
SAS Viya Services Do Not Start	261
Error: Nothing to do	261
ERROR: Procedure PCA not found ERROR: Procedure KCLUS not found	262
TimeoutError(error_message)TimeoutError	262
From Any Browser: Your Connection Is Not Private	263
From Google Chrome: Your connection is not private	263
ERROR: Unable to read a key	263
After Upgrade, One or More of the RabbitMQ Nodes Fails to Start Successfully	264
CAS Start-up failure post playbook run after changing the casenv_user in vars.yml	265
INTERNAL_SERVER_ERROR Internal Server Error An error occurred. Please contact your system administrator	265
Project creation failed with: creatingProviderError An unhandled provider creation error was detected. Setting project to failed creation state	266
Some Services Might Not Be Deregistered from Consul	266
Status command reports that sas-viya-esmagent-default service is "not ready"	268
"Connection reset by peer" Error Message	269

Troubleshooting SAS Viya

SAS Viya Services Do Not Start

Explanation

If Consul is deployed, one cause might be that certain SAS Configuration Server (Consul) files are corrupted.

Resolution

- 1 Stop all services.

Note: For information about the order in which to start and stop the services, see [Order for Stopping and Starting Servers and Services](#).

- 2 Delete the `/opt/sas/viya/config/data/consul/checks/` directory
- 3 Restart all services.

Error: Nothing to do

Error

After removing the software and attempting to re-install the software:

Error: Nothing to do

Explanation

The directories that contain the software were deleted. However, the yum remove command was never run. In `/var/log/yum.log`, the last entry for the rpm message is `Installed`.

Resolution

Clean up the yum repository by running the following command.

```
yum remove packagename
```

You can then re-install the software.

ERROR: Procedure PCA not found ERROR: Procedure KCLUS not found**Explanation**

The installation was attempted on a system that was not completely cleaned up from a previous installation.

Resolution

Uninstall SAS/CONNECT by running the following command:

```
yum groups mark remove "SAS/CONNECT"
```

Re-install SAS/CONNECT by running the following command:

```
sudo yum groupinstall "SAS/CONNECT"
```

TimeoutError(error_message)TimeoutError**Error**

When running the deployment:

```
TimeoutError(error_message)\nTimeoutError:
  Timer expired\n", "rc": 257} 13:15:37 |
INFO: | * 13:15:37 |
WARNING: | Execution return code '2'
is not the expected value '0' 13:15:37 |
INFO: | * 13:15:37 |
INFO: | Updating deployment times data
for step deploy_time with value 19 13:15:37 |
INFO: | * 13:15:37 |
WARNING: | Ansible execution
encountered failures
```

Explanation

The system failed to gather mount information.

Resolution

Perform one of the following actions:

- Set `/etc/mstab` as a link to `/proc/mounts` by running the following command:

```
sudo ln -s /proc/mounts /etc/mstab
```

- Edit the `ansible.cfg` file and add or change the time-out value for Ansible as follows:

```
timeout=number-of-seconds
```

Deploy your software by running the Ansible playbook again.

From Any Browser: Your Connection Is Not Private

Explanation

The default self-signed certificates are not in the operating system truststore by default. The Apache Web Server is configured to use a certificate that is signed by this Certificate Authority (CA). When you open any SAS URL and navigate to the web server from a machine that does not have this CA in the truststore, you will receive the message `Your connection is not private`. The message does not indicate that there is any problem with the SAS deployment.

Resolution

SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see the Security section of the System Requirements chapter.

From Google Chrome: Your connection is not private

Issue

When attempting to access SAS Viya software from Google Chrome, the following message is displayed:

```
Your connection is not private.
```

Explanation

If you have previously accessed a website using `https`, when you access the website again, Google Chrome automatically redirects to `https`.

Resolution

To reset Google Chrome so that it does not redirect to `https`:

- 1 In the Chrome address bar, enter this command:

```
chrome://machine-name/#hsts
```

- 2 Under **Query domain**, in the **Domain** box, enter the name of the machine that was used in the URL that you were attempting to access.
- 3 Click **Query** to determine whether the machine is known to the browser.
- 4 If the machine is known to the browser, under **Delete domain**, enter that machine name in the **Domain** box. Click **Delete**.

The corrected URL should now work with the HTTP protocol.

ERROR: Unable to read a key

Issue

When running the deployment, the following message is displayed:

```
fatal: [deployTarget2]: FAILED! =>{"changed": false, "failed": true, "msg":
"Get http://localhost:8500/v1/kv/config/application/rabbitmq/username: dial tcp [::1]:8500:
getsockopt: connection refused\n\
ERROR: Unable to read a key\nGet http://localhost:8500/v1/kv/config/application/rabbitmq/password:
dial tcp [::1]:8500: getsockopt:connection refused\n\
ERROR: Unable to read a key\n"}
```

Explanation

Consul requires each machine to have a single, private IP address. It does not bind to a public IP address by default. A machine target that is specified in your inventory file has one of the following conditions:

- multiple network adapters that have been assigned private IP addresses.

- no private IP address.

Resolution

To confirm the cause of the failure, check the Consul logs for an entry that resembles the following:

```
Starting Consul agent...=> Error starting agent: Failed to get advertise address:
Multiple private IPs found. Please configure one.
```

The resolution is to configure an adapter for the Consul bind parameter in `/etc/sysconfig/sas/sas-viya-consul-default`

Note: This file was installed by the Ansible playbook. This problem can be avoided by specifying the consul bind adapter in the inventory file during deployment.

Locate the following section of the file:

```
# Consul option: -bind
# Specify the desired name of a network interface or IPv4 address.
export CONSUL_BIND_EXTERNAL=adapter-name
```

For *adapter-name*, supply the name of the adapter that Consul should use to locate the machine.

After Upgrade, One or More of the RabbitMQ Nodes Fails to Start Successfully

Error

The log file contains the following message:

```
=ERROR REPORT==== 16-Nov-2017::16:50:21 ===
Cluster upgrade needed but other disk nodes shut down after this one.
Please first start the last disk node to shut down.
```

Note: if several disk nodes were shut down simultaneously they may all show this message. In which case, remove the lock file on one of them and start that node. The lock file on this node is:

```
/opt/sas/viya/config/var/lib/rabbitmq-server/mnesia/rabbit@abc.unx.abc.com/nodes_running_at_shutdown
```

Explanation

The RabbitMQ cluster was not stopped or started in the correct order.

Resolution

- 1 Stop and restart the RabbitMQ nodes.
 - a Manually stop all nodes in the reverse order in which they were started during the upgrade.
 - b Manually restart the nodes in the order that is listed in the inventory file's [rabbitmq] target group. You should wait for each rabbitMQ node to start completely before advancing to the next node in the list. If all nodes start successfully, skip to step 4.
- 2 For each rabbitMQ node that failed to start, connect to that target host and remove the lock file. Here is an example:
 - a


```
myhost$ ssh targethost.targetdomain.com
targethost$ sudo rm
/opt/sas/viya/config/var/lib/rabbitmq-server/mnesia/rabbit@<targethost.targetdomain.c
om>/nodes_running_at_shutdown
targethost$ exit
myhost$
```

- b Manually restart the nodes in the order that is listed in the inventory file's [rabbitmq] target group. You should wait for each rabbitMQ node to start completely before advancing to the next node in the list. If all nodes start successfully, skip to step 4.
- 3 On each failed target host, remove the rabbitMQ Mnesia database as follows:
 - a On each failed target machine, remove the following directory and all contents:


```
targethost$ sudo rm -R /opt/sas/viya/config/var/lib/rabbitmq-server/mnesia
```
 - b On each failed target machine, remove the internal rabbit cluster indicator file for SAS Viya:


```
targethost$ sudo rm
/opt/sas/viya/config/var/lib/rabbitmq-server/sasrabbitmq/sas.cluster.configured
```
 - c Reset the rabbitMQ cluster password. As part of the original deployment, you were instructed to change the default RabbitMQ client password. When you remove the Mnesia database, the password is reset to the system default. To change the password for the cluster, on one of the RabbitMQ target machines, run the following commands:


```
targethost$ cd /opt/sas/viya/home/bin
sudo ./sas-rabbitmq-acc-admin change_passwd -t client -u sasclient --promptpw
```
- 4 Rerun the playbook to continue the upgrade process after all rabbitMQ service failures have been resolved.

CAS Start-up failure post playbook run after changing the casenv_user in vars.yml

Explanation

The administrator has changed the casenv_user, which causes the CAS controller start-up to fail.

Resolution

- 1 Edit the `/opt/sas/viya/home/SASFoundation/utilities/bin/launchconfig_tenant_default` file, where tenant is either "viya" or the tenant name.
- 2 Change the line with `restrictServerLaunch=old user` to `restrictServerLaunch=new user`.
- 3 Rerun the playbook.

INTERNAL_SERVER_ERROR Internal Server Error An error occurred. Please contact your system administrator

Explanation

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/
./sas-bootstrap-config --consul --force
https://localhost:8501 --token-file
../../config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.to
ken kv write config/application/zones/internal.hostnames
correct-value-for-hostname
```

Project creation failed with: creatingProviderError An unhandled provider creation error was detected. Setting project to failed creation state

Explanation

This error is not an issue for the deployment. The configuration bootstrap process will retry until it is successful.

Resolution

Restart the data mining service.

Some Services Might Not Be Deregistered from Consul

Explanation

In SAS Viya 3.2, the following services were supported:

- recipeExecutionProvider
- SASVisualDataBuilder
- data-preparation-plans

In SAS Viya 3.3, these services have been removed or substituted with other microservices. In an upgrade scenario, it is possible that they might not be fully deregistered from Consul. The Consul log will repeatedly record messages about these failing services.

Resolution

To prevent the recording of these errors in the log file, you can manually deregister the services from Consul.

CAUTION! Be sure to deregister only the following services: recipeExecutionProvider SASVisualDataBuilder data-preparation-plans Removing other services will cause other failures.

To deregister the services, follow these steps:

- 1 Change to the executable directory:

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

- 2 To list the services in Consul that are required to be deregistered:

```
./sas-bootstrap-config agent check list | grep -i recipeExecutionProvider
```

The following information is returned:

```
"checkID": "service:recipeexecutionprovider-10-123-4-56",
"name": "Service 'recipeExecutionProvider' check",
"output": "Get http://machine.name.com:43345/recipeExecutionProvider/commons/health:
dial tcp 10.120.4.61:43345: getsockopt: connection refused",
"serviceID": "recipeexecutionprovider-10-123-4-56",
"serviceName": "recipeExecutionProvider",
```

Note: If you have multiple services running on multiple machines, more than one entry will be returned from the preceding command. Each checkID will correspond to the IP address of the machine where the service is running. Each checkID value should be deregistered.

- 3 To remove each of the checkID IP instances that are shown by the agent check list command, use the information from the checkID value in the preceding command output to deregister the health check:

```
./sas-bootstrap-config agent check deregister --id service:recipeexecutionprovider-10-123-4-56
```

- 4 To deregister the service, find out the ID for the service by running the following command:

```
./sas-bootstrap-config agent service list | grep -i recipeExecutionProvider
"recipeexecutionprovider-10-123-4-56": {
  "ID": "recipeexecutionprovider-10-123-4-56",
  "Service": "recipeExecutionProvider",
```

Note: If you have multiple services running on multiple machines, more than one entry will be returned from the preceding command. Each ID will correspond to the IP address of the machine where the service is running. Each ID value should be deregistered.

- 5 To remove each of the IP instances that are shown, use the output from the ID in the preceding command to deregister the service:

```
./sas-bootstrap-config agent service deregister recipeexecutionprovider-10-123-4-56
```

To remove the remaining services, repeat the preceding steps.

To remove SASVisualDataBuilder:

- 1 To list the SASVisualDataBuilder service in Consul that is required to be deregistered:

```
./sas-bootstrap-config agent check list | grep -i SASVisualDataBuilder
"checkID": "service:sasvisualdatabuilder-10-123-4-56",
  "name": "Service 'SASVisualDataBuilder' check",
  "output": "Get http://machine.name.com:46529/SASVisualDataBuilder/commons/health:
dial tcp 10.120.4.61:46529: getsockopt: connection refused",
  "serviceID": "sasvisualdatabuilder-10-123-4-56",
  "serviceName": "SASVisualDataBuilder",
```

- 2 To deregister the health check, use the output from the checkID value in the preceding command:

```
./sas-bootstrap-config agent check deregister --id service:sasvisualdatabuilder-10-123-4-56
```

- 3 To list the VisualDataBuilder service in Consul that is required to be deregistered:

```
./sas-bootstrap-config agent service list | grep -i VisualDataBuilder
"sasvisualdatabuilder-10-123-4-56": {
  "ID": "sasvisualdatabuilder-10-123-4-56",
  "Service": "SASVisualDataBuilder",
```

- 4 To deregister the service, use the output from the ID in the preceding command:

```
./sas-bootstrap-config agent service deregister sasvisualdatabuilder-10-123-4-56
```

To remove data-preparation-plans:

- 1 To list the data-preparation-plans services in Consul that are required to be deregistered:

```
./sas-bootstrap-config agent check list | grep -i data-preparation-plans
"checkID": "service:data-preparation-plans-10-123-4-56",
  "serviceID": "data-preparation-plans-10-123-4-56",
```

- 2 To deregister the health check, use the output from the checkID value in the preceding command:

```
./sas-bootstrap-config agent check deregister --id service:data-preparation-plans-10-123-4-56
```

- 3 To list the data-preparation-plans services in Consul that are required to be deregistered:

```
./sas-bootstrap-config agent service list | grep -i data-preparation-plans
```

```
"data-preparation-plans-10-123-4-56": {
  "ID": "data-preparation-plans-10-123-4-56",
```

- 4 To deregister the service, use the output from the ID in the preceding command:

```
./sas-bootstrap-config agent service deregister data-preparation-plans-10-123-4-56
```

Status command reports that sas-viya-esmagent-default service is “not ready”

This problem affects the SAS Event Stream Manager agent service only.

Explanation

Running the services status command, `sudo service sas-viya-all-services status`, reports that `sas-viya-esmagent-default` service is `Not ready`. However, the agent service might actually be running. The `sas-viya-all-services` script reports only on SAS Viya services that register with Consul.

Resolution

You can confirm the status of the service if you run `sudo systemctl status sas-viya-esmagent-default`. If it is running, a message states that it is `Active (running)`. If you instead run `sudo service sas-viya-all-services`, the status might report that the service is `not ready`.

You can ignore the incorrect status of `not ready` that is reported by the `all-services` script. Or you can edit the agent initialization script so that Consul registration is no longer used as a criterion to determine agent service status.

To edit the agent init script:

- 1 On the machine where you have installed the SAS Event Stream Manager agent, change directories to the following location: `/etc/init.d/`
- 2 Open the file named `sas-viya-esmagent-default` for editing.
- 3 Add the following line just below the header:

```
# sas-consul-register: False
```

Here is an example:

```
#!/bin/bash
# Copyright (c) 2017, SAS Institute Inc., Cary, NC, USA, All Rights Reserved
#sas-consul-register: False
```

- 4 Save your changes to the file.
- 5 Stop and restart the agent.

On Red Hat Enterprise Linux 6.x, run the following commands:

```
sudo service sas-viya-esmagent-default stop
sudo service sas-viya-esmagent-default start
```

On Red Hat Enterprise Linux 7.x, run the following commands:

```
sudo systemctl stop sas-viya-esmagent-default
sudo systemctl start sas-viya-esmagent-default
```

Running the following command should now report that the `sas-viya-esmagent-default` service is running:

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

"Connection reset by peer" Error Message

Explanation

Deployments on Red Hat Enterprise Linux may receive a "Connection reset by peer" message during deployment or when applying updates for SAS Viya. This is usually indicative of networking issues.

Resolution

Yum, the update and install utility that is reporting the error, can be configured to allow for more retries and a larger timeout in an attempt to work around the issue. Additionally, retrying the operation will sometimes resolve the issue.

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. This 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.

