SAS® Viya™ 3.1: Deployment Guide

Introduction

About This Guide

Use this guide to deploy SAS Viya in your environment.

- Make sure that you have the latest version of this deployment guide. The contents of this document are subject to continual updates.
- To use this guide successfully, you should have a working knowledge about the Linux operating system and basic commands.
- Unless specifically stated, the information in this guide pertains to the software that you ordered. You are notified if offering-specific content is required.

What's New in SAS Deployment

SAS Repositories

To ensure that you deploy the latest software, SAS provides SAS Viya in repository packages that are maintained by SAS. Specifically, the software is packaged in the RPM Package Manager (RPM) format, which simplifies installation, uninstallation, and upgrade tasks. Each time you deploy or update your software, you automatically receive the latest RPM packages that are available. If you receive a new playbook with updates, for the steps to update your software, see SAS Viya 3.1 Administration: Software Updates at http://support.sas.com/documentation/onlinedoc/viya/index.html.

Note: The RPM-based deployment model works with repositories that are native to your operating system. As a result, a SAS Software Depot is no longer required in your environment.

Industry Standard Tools

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

- SAS Viya deployment takes advantage of yum, a software package manager for Linux operating systems. Yum commands are used for secure access to RPM packages and for deploying and updating software in your environment.
- Ansible is the preferred tool for deploying SAS Viya. Ansible provides ease and flexibility for deploying to multiple machines. SAS provides an Ansible playbook that is based on your software order, and that can be customized for your environment. When you run the playbook, Ansible automates a series of yum commands that deploy the software.

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

One Deployment Guide

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

What Gets Deployed

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

- SAS Visual Data Mining and Machine Learning
- SAS Cloud Analytic Services (CAS), which is the analytics and license server for SAS Viya. CAS Server Monitor is the web application that provides the graphical user interface to SAS Cloud Analytic Services.
- Data connectors, which enable you to configure connections to data sources such as existing SAS data, Oracle databases, and Hive data in Hadoop.
- SAS Studio, a web application that provides the graphical user interface for users to submit actions and code. It can also be used to perform some post-deployment administrative tasks.
- SAS Event Stream Processing for SAS Viya (optional), which requires a separate license.

Also, during deployment, the following two accounts and one group are created, if they do not already exist. Both accounts are required.

- **`sas`** is a user account under which processes run.
- **`cas`** is a service account under which processes run for the CAS server.
- Both accounts are created in a group named **`sas`**.

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

Deployment Scenarios

Advice about the Scenarios

- In most scenarios, it is assumed that Ansible is used to deploy software. Ansible is shown as installed on a separate machine, called the Ansible controller. Instructions for installing Ansible are provided later in this guide.
- Deploying the CAS server to a dedicated machine, or in a distributed method across multiple machines, might improve analytics-processing performance for users.
- When you deploy the CAS server, a role is assigned to each machine: CAS controller or CAS worker. If you deploy the CAS server to a single machine, the controller role is assigned. For a distributed CAS server, both roles are assigned.
- To specify the target machines that are shown in the multi-machine deployments, you edit the **`hosts`** file that is associated with the playbook.
- Data connectors must be deployed to one or more machines on which CAS is running. For scenarios in which CAS is deployed to multiple machines, data connectors are deployed to the CAS controller and to each CAS worker.

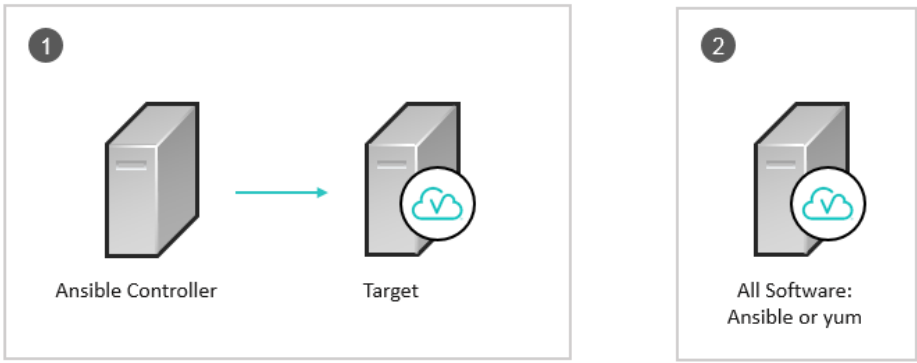
  Note: Data connectors vary according to the order.
- If you purchased SAS Event Stream Processing for SAS Viya (optional), the playbook automatically installs it on every CAS controller and worker machine. You must have an existing deployment of SAS Event Stream Processing 3.2 or a later version in your environment to provide data to the CAS actions that support streaming data. This independent installation of SAS Event Stream Processing must be running on a separate machine on which no CAS components are installed. The independent installation also enables SAS Event Stream Processing Studio.
- For deployments that use Hadoop, additional configuration is required to enable access to data in Hive or SASHDAT on Hadoop Distributed File System (HDFS). Additional configuration occurs after you deploy SAS software and the CAS controller and workers using Ansible.

Scenario 1: Single Machine

In this scenario, you can use Ansible or yum to deploy all SAS software to a single machine.

The following figure shows two options for deployment:

Two Examples of a Single-Machine Deployment



1	The Ansible machine, which is called the Ansible controller, deploys all SAS software to a different machine, the target node.
2	All SAS software is deployed to a single machine by using Ansible or yum. If Ansible is used, the target node is the same machine as the one where Ansible is installed and running.

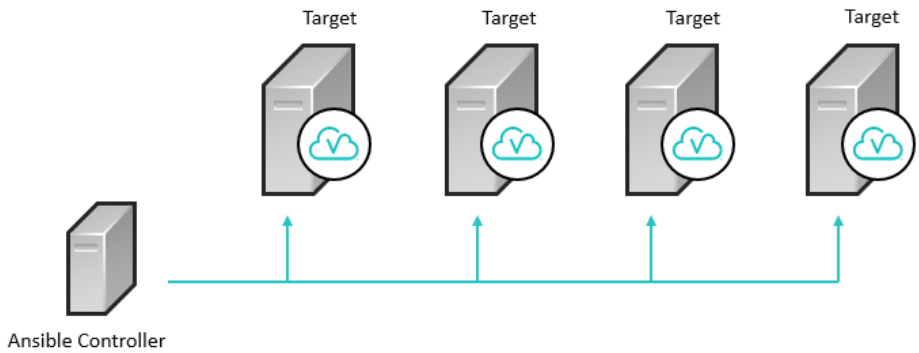
Scenario 2: Single Machine, Repeated

This scenario is useful for deploying SAS software in the following environments:

- development, testing, staging, and production environments
- the same deployment for different groups of users

The following figure shows an Ansible controller that is used to deploy the same SAS software to multiple target nodes:

Single Machine, Repeated

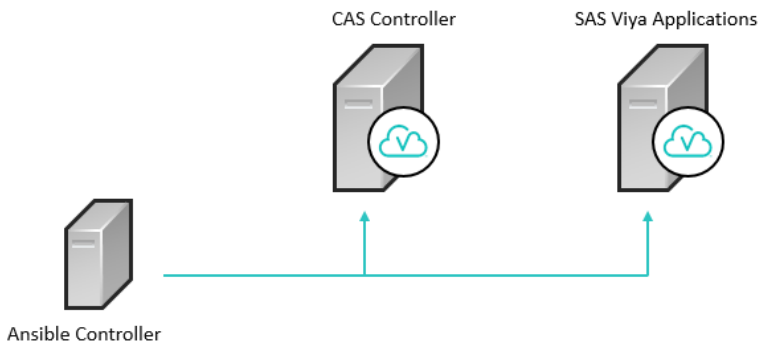


Scenario 3: Dedicated Machine for the CAS Server

The CAS server is the analytics server that the SAS procedures use for analytics processing. In this scenario, the CAS controller is deployed to a target node that is separate from other SAS software. During deployment, the CAS controller role is assigned to the target node.

> **TIP** Deploying the CAS server to a dedicated machine might improve analytics-processing performance over a deployment in which all SAS software, including the CAS server, is installed on the same machine.

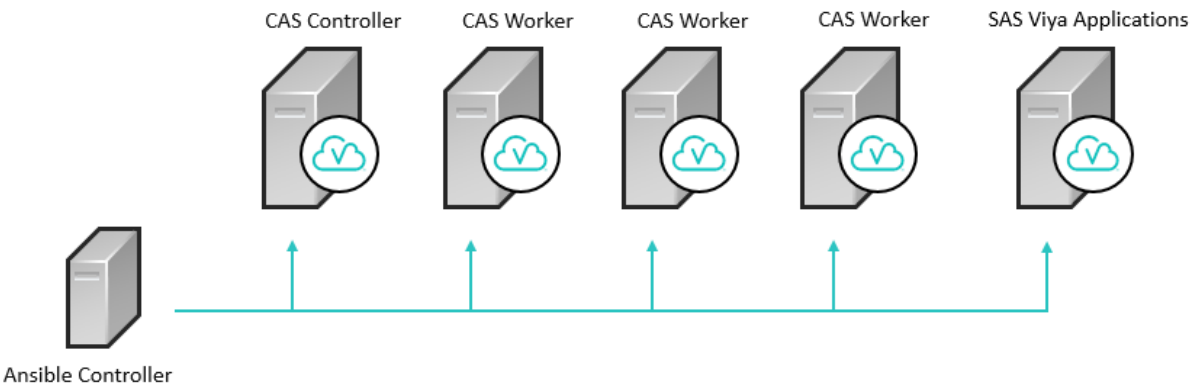
Dedicated Machine for the CAS Server



Scenario 4: Distributed CAS Server

In this scenario, CAS is deployed across two or more nodes in a clustered environment. An advantage of this scenario is that optimal processing can be achieved through massively parallel processing (MPP) for multiple users. During deployment, the CAS controller and CAS worker roles are assigned to the nodes.

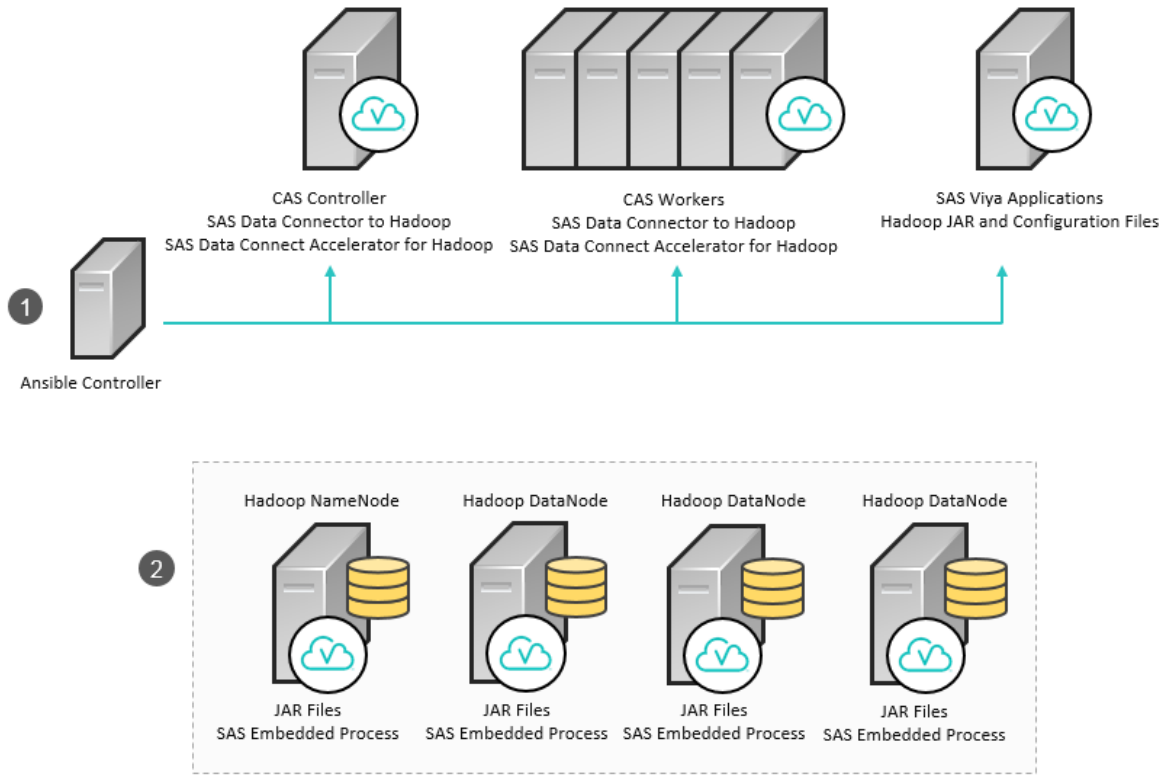
Distributed CAS Server



Hadoop Scenario 1: Access to Data in Hive

The following scenario provides guidance for deploying SAS software to support access to data in Hive:

Distributed CAS Server with Access to Data in Hive



Overview of Deployment Steps

1	Ansible is used to deploy SAS software: SAS Viya applications on one machine, and the CAS controller and worker nodes on the same machines as the data connectors. SAS Data Connector to Hadoop enables serial processing. SAS Data Connect Accelerator for Hadoop enables parallel processing between the CAS server and Hadoop. For details about deploying with Ansible, see Deploy with the Ansible Playbook on page 24.
2	Hadoop JAR files are installed. To enable parallel processing, the SAS Embedded Process is deployed to the Hadoop nodes. The installation of the JAR files and the deployment of the SAS Embedded Process are not performed using Ansible. For more information about installing and configuring this software, see Appendix C: Hadoop Deployment: Configuring the SAS Data Connector to Hadoop and Optionally, the SAS Data Connect Accelerator for Hadoop on page 59.

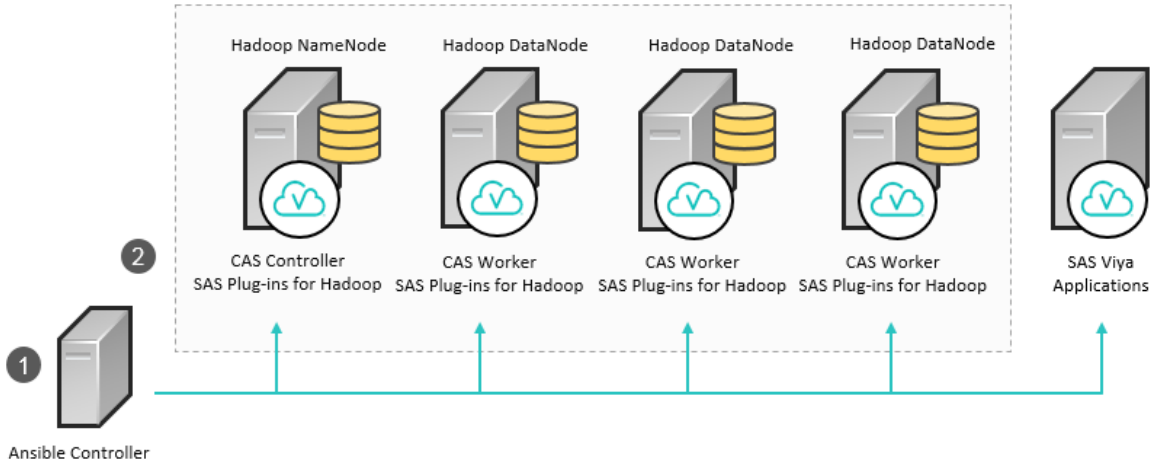
Note: This deployment model provides parallel access to data in Hive through the use of the SAS Embedded Process and SAS Data Connect Accelerator for Hadoop. However, this type of deployment does not provide the ability to save data from CAS back to a Hive table. To save data from CAS, path-based caslibs must be used. A caslib is an in-memory space that is used to hold tables, to access control lists, and to provide data source information. All data that is available to CAS through caslibs and all operations in CAS that use data are performed with a caslib in place.

Hadoop Scenario 2: Access to Data in SASHDAT on HDFS

The following scenario provides guidance for deploying the CAS server to support access to data in SASHDAT on HDFS. In this scenario, the CAS server is deployed to nodes of a Hadoop cluster. This scenario is also

referred to as a co-located deployment. This scenario optimizes CAS processing, which is close to the persistent HDFS data store, and promotes the use of SASHDAT tables.

Distributed CAS Server Co-Located on a Hadoop Cluster



Overview of Deployment Steps

1. Ansible is used to deploy SAS software: CAS controller and workers on the Hadoop nodes, and SAS Viya applications on a separate machine. For details about deploying with Ansible, see Deploy with the Ansible Playbook on page 24.

   Consider the following information before deciding how to deploy the CAS controller and workers within your Hadoop environment:

   - The CAS controller must be deployed to the NameNode or to one of the DataNodes.
   - The CAS workers must be deployed to all DataNodes or to a subset of the DataNodes.
     - To support CAS worker failover, you must deploy CAS workers to all DataNodes. No additional data connector setup is required. If your CAS license permits fewer than the complete number of CPU cores in your Hadoop environment, you can deploy CAS workers to all DataNodes and use a subset of the CPU cores.
     - If the CAS workers are deployed to a subset of the DataNodes, data is written in parallel only to those DataNodes on which a CAS worker is deployed.

2. Access to SASDAT on HDFS is configured. For more information, see Appendix D: Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS on page 80

Contact SAS Technical Support

If you need assistance with deploying your software, it is important that only SAS support personnel call the Technical Support Division.

- For US and Canadian customers, support is provided from the corporate headquarters in Cary, North Carolina. You can call (919) 677-8008, Monday through Friday.
- Customers outside of the US can obtain local-language technical support through the local office in their countries. Customers in these locations should contact their local office for specific support hours. See support.sas.com/techsup/contact/index.html for contact information for local offices.

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 discussion forums, Technical Support contact options, and other support materials that might answer your questions.

System Requirements

Hardware Requirements for SAS Visual Data Mining and Machine Learning

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

SAS strongly recommends consulting with a SAS Sizing Expert to obtain an official hardware recommendation that is based on your estimated SAS workload and number of users. The sizing information provided here is not intended as a substitution for expert advice. To request sizing expertise, send an email to contactcenter@sas.com.

SAS Visual Data Mining and Machine Learning components can be installed on a single machine or on multiple machines. For more information, see Deployment Scenarios on page 5. The installation files are automatically downloaded to the `/var/cache/yum` directory. This directory therefore requires sufficient available disk space to accommodate the installation packages. Verify that at least 10 GB of disk space are available for SAS Viya installation.

Additional space for logs is also required in `/opt/sas/viya`. The amount that is required depends on the logging level that you have set. However, the minimum amount of disk space required for the installation and for logging is 40 GB. If you do not have enough available disk space for logs, you should create a symbolic link to another partition that contains at least 40 GB of unused space. We recommend using monitoring tools to ensure that none of the locations used by the deployment fills up without warning.

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

Requirements for Single-Machine Deployment

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

Note: These values apply to the programming interface only. A future release will provide an option to enable an additional visual interface. If you intend to enable the visual interface in the future, additional resources will be required. Request sizing expertise from SAS if you anticipate reusing your target machines for this purpose when the visual interface is available.

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

An additional machine can be used as a “thin client” from which end users can access the product user interface. This machine requires minimal processing power and storage space and can run on Windows or UNIX. For more information about client requirements, see Browser Requirements on page 18.

Operating System Requirements

Supported Operating Systems

The following operating systems are supported:

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

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

Linux Prerequisites

SAS Viya deployment requires the operating system to be registered with the Red Hat Network. Registration enables you to receive periodic software updates. For a SAS software deployment, registration also enables yum to download software from SAS repositories. Verify that the machine where you perform the deployment (typically, the Ansible controller) is registered and that your subscription has been activated. To check whether the system is registered, run the following command:

```
subscription-manager version
```

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

```
subscription-manager list --installed
```

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

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

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

- libXp
- libXmu
- glibc 2.12
- the numactl package
- the X11/Xmotif (GUI) packages
- libpng12
- xterm

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

Additional Linux Requirements for SAS Event Stream Processing for SAS Viya

The SAS Event Stream Processing Engine libraries were built using gcc-4.4.7-16 and the Boost library 1.58. The Boost library 1.58 is automatically installed along with SAS Event Stream Processing. The libraries are compiled using the following compiler options:

```
-D_REENTRANT
-D_THREAD_SAFE
```

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

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

SAS Support for Alternative Operating Systems

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

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

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

Server Software Requirements

Java Requirements

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

- Oracle JRE SE version 1.8.0_92 or a later release
- OpenJDK version 1.8.x

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

The deployment playbook checks for a preinstalled version of Java that meets or exceeds the requirements. If it is found, it is used. Otherwise, the playbook attempts to install a recent version of OpenJDK and to set the path in a system configuration file. You can also specify the path to an existing JRE in your vars.yml file before you run your playbook. For more information, see Specify JRE (Optional) on page 24.

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

Data Source and Storage Requirements

Overview of Data Warehouse and Storage Requirements

If you purchased SAS Event Stream Processing for SAS Viya, a separately licensed instance of SAS Event Stream Processing 3.2 or a later version is a required data source. If you do not already have an installation of SAS Event Stream Processing, purchase the product separately and install it on a separate machine in order to

obtain a valid license. The separate license also enables SAS Event Stream Processing Studio, which is the product user interface.

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 data connectors and a data connect accelerator on your CAS controller and all CAS worker machines.

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 to be used 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 data connector or data connect accelerator for additional system requirements that apply to the CAS controller and CAS worker machines.

Data Encoding Requirement

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

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

Supported Data Sources

SAS Viya supports the following data sources:

Note: Each data source also requires a SAS data connector and possibly a SAS data connect accelerator. In some cases, a data connector might have individual system requirements.

- Apache Hive
- Impala
- Data sources accessible with an ODBC driver
- Oracle
- PC files, which support the following file extensions:
  - `.jmp`
  - `.spss`
  - `.stata`
  - `.xlsx` or `.xls`
- PostgreSQL
- Teradata

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

SAS Viya also supports the following data sources, which use SAS data connectors that are automatically included with CAS and are not separately licensed or configured.

- SASHDAT on HDFS

- LASR Analytic Server (SAS 9.4)
- SAS data sets

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

Hadoop Requirements

Supported Releases of Hadoop Distributions

SAS Viya supports the following Hadoop third-party distributions:

- Cloudera CDH 5.7 and later releases
- Hortonworks HDP 2.4 and later releases
- MapR 5.1 and later releases

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.
- For Hive, SAS Data Connector to Hadoop and possibly SAS Data Connect Accelerator for Hadoop are required. The SAS Data Connector do 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 on support.sas.com. The same policy that applies to SAS 9.4 also applies to SAS Viya.

Requirements to Import Data from SAS 9.4

SAS/CONNECT is required in the SAS Visual Data Mining and Machine Learning environment to move data from other SAS deployments and operating systems into SAS Viya. SAS/CONNECT can convert data from a non-UTF-8 encoded SAS session to the UTF-8 format that SAS Viya requires.

SAS/CONNECT is not included with a standard SAS Visual Data Mining and Machine Learning order. You must order it separately. If you order SAS/CONNECT, the required commands to install it are automatically included in your playbook.

Requirements for SAS Data Connector to Hadoop

Note: SAS Data Connector to Hadoop is included in the SAS/ACCESS Interface to Hadoop (on SAS Viya).

SAS Data Connector to Hadoop requires a supported Hadoop distribution. For more information, see Supported Releases of Hadoop Distributions on page 15.

The SAS Data Connector to Hadoop also requires the following:

- Hive

  In addition, Hive (specifically, the hadoop_extract.sh script) requires the following software:

  - Oracle JRE version 1.8 or a later version.
  - Python, strace, and wget (which are included in your version of Linux).
- MapReduce
- YARN

- HCatalog for processing non-delimited Hive file types.

Requirements for SAS Data Connect Accelerator for Hadoop

Note: SAS Data Connect Accelerator for Hadoop is included in the SAS In-Database Technologies for Hadoop (on SAS Viya).

SAS Data Connect Accelerator for Hadoop requires a supported Hadoop distribution. For more information, see Supported Releases of Hadoop Distributions on page 15.

The SAS Data Connect Accelerator for Hadoop also requires the following:

- Hive
- MapReduce
- YARN
- HCatalog for processing non-delimited Hive file types.
- SAS Embedded Process for Hadoop deployed. For details, see Deploy the SAS Embedded Process for Hadoop for Parallel Processing on page 64.

  Note: In order to load data in parallel with the SAS Embedded Process, 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.

Requirements for SAS Data Connector to Impala

Note: SAS Data Connector to Impala is included in the SAS/ACCESS Interface to Impala (on SAS Viya).

SAS Data Connector to Impala requires Impala Server version 2.5 or a later version. It also requires the ODBC Driver for Impala, version 2.5.33 or a later version.

In addition, the ODBC Driver for Impala requires a compatible ODBC Driver Manager, such as the unixODBC Driver Manager.

Requirements for SAS Data Connector to ODBC

Note: SAS Data Connector to ODBC is included in the SAS/ACCESS Interface to ODBC (on SAS Viya).

SAS Data Connector to ODBC enables access to multiple data source types by means of a generic ODBC driver.

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

You must install the ODBC driver on the CAS controller.

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

Requirements for SAS Data Connector to Oracle

Note: SAS Data Connector to Oracle is included in the SAS/ACCESS Interface to Oracle (on SAS Viya).

SAS Data Connector to Oracle requires the Oracle client release 12c (64-bit libraries) or later releases.

You must install the Oracle client on the CAS controller.

Requirements for SAS Data Connector to PostgreSQL

Note: SAS Data Connector to PostgreSQL is included in the SAS/ACCESS Interface to PostgreSQL (on SAS Viya).

SAS Data Connector to PostgreSQL can connect to a PostgreSQL Database version 9.4.4 or a later version.

SAS Data Connector to PostgreSQL requires a driver manager and an ODBC driver for PostgreSQL. SAS provides both of these ODBC client components.

Requirements for SAS Data Connector to Teradata

Note: SAS Data Connector to Teradata is included in the SAS/ACCESS Interface to Teradata (on SAS Viya).

The following database management system (DBMS) products are supported:

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

Requirements for SAS Data Connect Accelerator for Teradata

Note: SAS Data Connect Accelerator for Teradata is included in the SAS In-Database Technologies for Teradata (on SAS Viya).

The following DBMS products are supported:

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

The SAS Data Connect Accelerator for Teradata also requires SAS Embedded Process to Teradata.

Note: In order to load data in parallel with SAS Embedded Process, 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.

Security Requirements

User Account Requirements

Verify that the following requirements have been met before you start the deployment:

- Administrator privileges have been granted on the Linux machine where you are launching the SAS software deployment.
- The user account that you are using for the deployment must have super user (sudo) access. Verify that the user ID is included in the sudoers file. Run the following command:

  ```
  sudo -v
  ```

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

  ```
  sudo -l
  ```

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

During deployment, two required accounts (one service account and one user account) and one group are created for you unless they already exist. Because these accounts are required for installation and for running services during the product's normal operation, do not delete them or change their names. These accounts do not run as root. If you must log on to one of these accounts, use sudo to access them.

The following table identifies and describes the predefined accounts:

Account Name and Group	Parameters	Purpose
sas; member of sas group	UID: 1001 GID: 1001 Non-login service account without user restrictions. No password. You can add a password after installation, if necessary. The password does not expire. The default user name is required.	Required for the installation The installation process sets user and group ownership permissions on all of the installation files. This user must exist to enable ownership. After the installation has completed, this user account enables the required components to run, including the web application server for SAS Studio.
cas; member of sas group	UID: 1002 GID: 1001 Typical user account that is subject to user restrictions. No default password is assigned. If the CAS server is running in an analytic cluster environment (with multiple CAS workers), and if you use an Ansible playbook for the deployment, passwordless SSH is configured by default.	Required for managing Cloud Analytic Services (CAS). Use this user account to log on to CAS Server Monitor.

All of the users who will launch CAS sessions must have a consistent UID and GID on all machines in your deployment. Similarly, during the deployment, the cas user that is described above also requires a consistent UID and GID. Consistent user and group IDs are typically shared on all machines automatically by the deployment process. However, if you are deploying with some custom user accounts, you might have to use the UNIX `usermod` command to modify the UIDs of any mismatched user accounts to make them consistent. For groups with mismatched GIDs, use the `groupmod` command.

Authentication Requirements

SAS Visual Data Mining and Machine Learning supports pluggable authentication modules (PAM) for authentication.

Default PAM configuration files (**`/etc/pam.d/`*`service`***) are installed for the CAS server and SAS Studio. For *service*, substitute *cas* for the CAS server or *sasauth* for SAS Studio. For information about updating these files for your environment, see *SAS Viya Administration: Authentication*.

Transport Layer Security (TLS) Software Requirements for SAS Embedded Process

If you are using the SAS Embedded Process, to secure the data transfer between your cluster and CAS, the following software is required on each node in the cluster:

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

Browser Requirements

Web Browsers for SAS Studio and CAS Server Monitor

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

- Apple Safari 6.0 (on Apple OS X) and later versions
- Google Chrome 27 and later versions
- Microsoft Internet Explorer 9, 10, or 11

  Note: Microsoft Edge is not supported.
- Mozilla Firefox 21 and later versions

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

Ansible Controller Requirements

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

A typical Ansible deployment consists of at least one control machine, the Ansible controller, and Ansible managed nodes, which are the machines where SAS software is installed. In a single-machine deployment that uses Ansible rather than yum, Ansible and all SAS software are installed on the Ansible controller.

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

In a distributed deployment, the managed nodes use a secure shell (SSH) framework for connections to the Ansible controller. Verify network connectivity between the controller machine and the managed nodes. Connectivity is also required between all machines in the deployment and from the controller to the SAS yum repositories. For more information, see Firewall Considerations on page 21.

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. For more information, see Linux Prerequisites on page 12.

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

- paramiko
- PyYAML
- jinja2

For more information about Ansible installation, see Install Ansible on page 21.

Pre-installation Tasks

Make Sure That You Have the Required Files

When you order SAS software, SAS sends a Software Order Email (SOE) to your business or organization that includes information about the software order. The SOE directs you to save its attached .tgz file and the license file to a directory on your Ansible controller. If you have not already done so, you must save those files before performing any of the steps in this section.

In the same directory where you have saved the .tgz file, uncompress it. A sas_viya_playbook subdirectory is added, containing the following files:

- a second copy of the license file
- the entitlement_certificate.pem and SAS_CA_Certificate.pem files
- the customized_deployment_script.sh file
- the files that make up the SAS Viya playbook, referred to in the rest of this guide as “the playbook”

Configure SELinux

On every target machine, for all Linux distributions, run the following commands:

```
sudo setenforce 0
sudo sed -i.bak -e 's/SELINUX=enforcing/SELINUX=permissive/g' /etc/selinux/config
```

Enable Required Ports

The following ports are used by SAS Viya and should be available before you begin to deploy your software. If any firewalls are configured on the operating system or at the network level, the same ports should be opened there also.

Process	Required Port
CAS Server Starting Port	5570
CAS Communicator Port	5580
Object Spawner	8591
CAS Server Monitor	8777
SAS/CONNECT Spawner	17551
SAS Studio	38080

Firewall Considerations

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. The IP addresses for content delivery services vary by region. For details about the list of IP addresses, see Public CIDR Lists for Red Hat.
2. Ensure that the firewall allows access to the following yum repositories that are hosted by SAS so that content can be delivered for deployment:
   - https://ses.sas.download/
   - https://bwp1.ses.sas.download/
   - https://bwp2.ses.sas.download/
   - https://sesbw.sas.download
3. If you are using Red Hat Enterprise Linux 6.7 or CentOS 6.7, run the following commands:

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

   If you are using any other version of Linux, including other versions of Red Hat Enterprise Linux or CentOS, run the following commands:

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

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

Configure the Use of a Proxy Server

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

Enable the Yum Cache

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

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

```
keepcache = 1
```

Install Ansible

Ansible is third-party software that provides automation and flexibility for deploying software to multiple machines. (For more information about Ansible's role in the deployment and for a comparison with yum, see Industry Standard Tools on page 4.) If you decide to use Ansible to deploy your software, use the information in this section to install and configure Ansible.

Installation Steps

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

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

- If you are installing on a machine with Red Hat Enterprise Linux, ensure that you are registered to the Red Hat Network as described at Linux Prerequisites on page 12.
- If you are using the equivalent of Red Hat Enterprise Linux 6.7, use the following command:

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

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

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

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

```
sudo yum repolist all
```

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

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

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

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

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

Test Your Ansible Installation

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

```
ansible localhost -m ping
```

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

Perform Linux Tuning

Set the MaxStartups Variable

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

Set the ulimits

The Linux operating system provides controls that enable you to limit the maximum number of open file descriptors and the maximum number of processes that a user ID can use. Perform the following steps to ensure that the limits are high enough for each machine in your deployment to function correctly.

Perform the following steps as the root user ID. For distributed CAS server installations, you can edit the files on one machine and copy the files to the other machines.

1 To set the maximum number of open file descriptors for each machine in your deployment, edit **/etc/security/limits.conf** file on each machine, adding the following line or verifying that it already exists:

```
*     -     nofile     20480
```

2 For each machine in your deployment, edit the appropriate **`*-nproc.conf`** file and change the value for nproc from the default value of 1024 to 65536.

Note: In the filename *-nproc.conf, * is a wildcard that refers to a unique prefix to the nproc.conf filename that varies according to the version of Red Hat Enterprise Linux that is used.

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

After you edit and save the file, the changes appear as follows:

```
# Default limit for number of user's processes to prevent
# accidental fork bombs.
# See rhbz #432903 for reasoning.

*          soft    nproc     65536
root       soft    nproc     unlimited
```

Installing SAS Viya

Deploy with the Ansible Playbook

Specify JRE (Optional)

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

1. With a text editor, open the **`sas_viya_playbook/vars.yml`** file.
2. Set the value of sas_install_java to false. For example:

   ```
   sas_install_java: false
   ```

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

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

4. Save and close the vars.yml file.

For a list of supported versions of Java, see Java Requirements on page 13.

Set the CAS Cache Directory

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

The cache directory can be changed to one that has more space. If you decide to change the cache, be sure to select a directory that accounts for tables that are loaded from other data sources and tables created as outputs from CAS actions. The size required differs for each user, but can run from gigabytes to terabytes. You can also provide a list of directories to be used as cache. If you use a list, each time the server needs to use disk, it uses the next path in the list. This strategy is used to distribute the load across disk volumes.

To change the CAS cache:

1. With a text editor, open the **`sas_viya_playbook/vars.yml`** file.
2. In the CAS_CONFIGURATION section, remove the number sign (#) that precedes the CAS_DISK_CACHE variable.
3. Remove the **`/tmp`** value from the variable and replace it with the directory that you want to use as the CAS cache. If you want to use more than one directory, list them all with colons separating the directories. For example:

   ```
   CAS_CONFIGURATION:
      env:
        CAS_DISK_CACHE: /var/tmp:/var/tmp2:/var/tmp3
   ```

   The best practice is to create directories dedicated to caching that are owned by the ID that executes the CAS server (cas by default). Each directory should be set up identically on each CAS node. All CAS processes must have Read, Write, and Execute permissions for these directories. Therefore, permissions

must be granted to the server's ID and the ID of any CAS user that connects through programming interfaces like SAS and Python.

4 Save and close the vars.yml file.

Add Data Source Information

Overview of the Data Sources

If your order includes one or more data connectors, you must edit the vars.yml file to include information that is needed to install and configure the specific data connector. If you intend to use HDFS, you must also edit the vars.yml file.

The vars.yml file contains an example of a typical CAS_SETTINGS block that is commented out. The following sections contain examples of CAS_SETTINGS blocks that are appropriate for the specific connector. To customize the file, either uncomment the lines and edit the existing CAS_SETTINGS block or create a new CAS_SETTINGS block using the example's format.

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

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

Connect to HDFS

If you require CAS SASHDAT access to HDFS, follow these steps to edit the vars.yml file.

1 Open the vars.yml file.

2 Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.

3 Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

```
1: HADOOP_HOME=location-of-your-Hadoop-home-directory
2: HADOOP_NAMENODE=primary-namenode-host-name
```

4 Save and close the vars.yml file.

SAS Data Connector to Hadoop and SAS Data Connect Accelerator for Hadoop

Follow these steps to edit the vars.yml file.

1 Open the vars.yml file.

2 Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.

3 Under CAS_SETTINGS:, add the following lines, including the spaces and numerical prefixes:

```
1: JAVA_HOME=location-of-your-Java-8-JRE
2: 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 playbook to specify a different location for the installation of the JRE.

4 Save and close the vars.yml file.

SAS Data Connector to Impala

Follow these steps to edit the vars.yml file.

1 Open the vars.yml file.

2 Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.

3 Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

```
1: CLOUDERAIMPALAINI=location-of-your-cloudera.impalaodbc.ini-file
2: SIMBAINI=location-of-your-cloudera.impalaodbc.ini-file
3: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-Impala-ODBC-driver:
/opt/cloudera/impalaodbc/lib/64
```

4 Save and close the vars.yml file.

SAS Data Connector to ODBC

Follow these steps to edit the vars.yml file.

1 Open the vars.yml file.

2 Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.

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

For DataDirect:

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

For iODBC:

```
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=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

For unixODBC:

```
1: ODBCSYSINI=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 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 vars.yml file.

SAS Data Connector to Oracle

Follow these steps to edit the vars.yml file.

1 Open the vars.yml file.

2 Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.

3 Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

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

4 Save and close the vars.yml file.

SAS Data Connector to PostgreSQL

Follow these steps to edit the vars.yml file.

1. Open the vars.yml file.
2. Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.
3. Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

```
1: ODBCINI=location-of-your-odbc.ini-file
2: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-PostgreSQL-ODBC-driver:
/opt/sas/viya/home/lib64/lib
```

4. Save and close the vars.yml file.

SAS Data Connector to Teradata

Follow these steps to edit the vars.yml file.

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. Open the vars.yml file.
4. Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.
5. Under CAS_SETTINGS, add the following lines, including the spaces and numerical prefixes:

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

   An example of the TTU Default LD_LIBRARY_PATH is

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

6. Save and close the vars.yml file.

Specifying Multiple Data Connectors

Note: When adding multiple data connectors, 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 data connectors can be mixed in the block, ensure that the lines for each data connector remain in the order provided in the preceding sections.

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

1. Open the vars.yml file.
2. Remove the hashtag (#) from the beginning of the CAS_SETTINGS line.
3. Under CAS_SETTINGS, add the appropriate lines. Here is an example of a block for both the DataDirect version of SAS Data Connector to ODBC and for SAS Data Connector to Oracle:

```
1: ODBCHOME=/dbi/odbc/dd7.1.4
2: ODBCINI=/r/ge.unx.sas.com/vol/vol310/u31/fedadmin/ODBC/odbc_714_MASTER.ini
```

```
3: ORACLE_HOME=/dbi/oracle/12c
4: LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ODBCHOME/lib
5: LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Note: In the fourth line, each variable is separated with a colon.

4 Save and close the vars.yml file.

Set Up the cas User

If you decide not to use the cas user to be created for you by default, you must edit the playbook so that the correct user is used in the deployment.

If you decide to create your own cas user, understand that the user must be able to connect from the CAS controller machine to each CAS machine without providing a password. In addition, the **id_rsa** key for the user must be stored in the **$HOME/.ssh** directory.

To use your own cas user, follow these steps:

1 Open the vars.yml file.

2 In the casenv_user field, insert the user name.

3 In the casenv_group field, insert the user group.

4 Save and close the vars.yml file.

Set Up the CAS Admin User

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

1 Open the vars.yml file.

2 Remove the comment symbol (#) from the #casenv_admin_user field.

3 In that same field, insert the name of a user that exists and that can log on:

```
casenv_admin_user: valid-user
```

4 Save and close the vars.yml file.

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

Set Up Passwordless SSH for CAS

How Passwordless SSH Is Set Up

If CAS is deployed on multiple machines, each machine requires passwordless SSH in order to communicate with the others. Passwordless SSH is set up by the Ansible playbook by default.

If no changes are made, the deployment process occurs as follows:

1 SSH keys are set up for the cas user account that is created during the RPM installation.

2 A set of keys is created for each 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 on the CAS cluster.

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

Based on the action that you want to take, use the following list to find the steps that you should perform:

- To set up your own passwordless SSH, see Use an Existing Passwordless SSH on page 29.
- To allow SAS to create the passwordless SSH with a single user (cas), no user action is required.
- To add users to the passwordless SSH that SAS creates, see Edit the SSH Properties on page 29.

Use an Existing 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 User on page 28.

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.

Edit the SSH Properties

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

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

Note: Comments have been removed from the following example.

```
tmp_ssh_dir: /tmp/sas_deploy/ssh
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 Set the tmp_ssh_dir variable to a directory that can be used as the temporary location where keys are created for each user that is defined in this code block.

b Ensure that the setup_sas_users variable is set to true.

c 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**.

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

e 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.

Edit the Inventory File

Ansible uses an inventory file to define the machines to be included in a deployment and the software to be installed on them. For multi-machine deployments, the **sas_viya_playbook/hosts** is used as the inventory file. The **sas_viya_playbook/host_local** file is used for a single-machine deployment.

Note about Sharing the hosts File and the host_local File

The hosts and host_local files are generated for a specific software order. Do not copy these files from one playbook and attempt to use them in another playbook.

Edit the host_local File

If you are performing a multi-machine deployment, you should skip this section and go to Define the Machines in the Deployment on page 30.

The first line of the host_local file is a deployment target reference, defining the machine on which the SAS Viya software is being deployed. If you use Ansible locally (on the same machine where you are deploying SAS Viya software), you should use the host_local file without modification. If you use Ansible remotely, you should modify the first line in the host_local file to include the location of the machine where SAS Viya is being deployed using the following format:

```
deployTarget ansible_ssh_host=host1.example.com
```

Save the host_local file. Go to Deploy the Software on page 33 for the commands to deploy your SAS Viya software.

Define the Machines in the Deployment

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

```
deployTarget ansible_ssh_host=<machine address> ansible_ssh_user=<userid> ansible_ssh_private_key_file=
<keyfile>
```

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

Component of the Deployment Target Declaration	Description
deployTarget	the alias that is used by Ansible to refer to the physical machine definition. Choose a meaningful alias such as ansible-controller.
ansible_ssh_host	the IP address of the remote machine.
ansible_ssh_user	the user ID that is used by Ansible to connect to each of the remote machines and to run the deployment.
ansible_ssh_private_key_file	the private key file that corresponds to the public key that was previously installed on each of the remote machines. This file typically resides in your **~/.ssh** directory.

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

```
deployTarget ansible_connection=local
```

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

```
main ansible_ssh_host=host1.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
controller ansible_ssh_host=host2.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
worker-1 ansible_ssh_host=host3.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
worker-2 ansible_ssh_host=host4.example.com ansible_ssh_user=user1 ansible_ssh_private_key_file=~/.ssh/id_rsa
```

Assign the Target Machines to Ansible Groups

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

```
[sas-viya]
main

[configuratn]

[consul]

[httpproxy]

[pgpoolc]

[platform-apps]

[platform-services]

[rabbitmq]

[sasdatasvrc]

[sas-casserver-primary]
controller

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

```
worker-2

[va-apps]

[sas-all:children]
sas-viya
configuratn
consul
httpproxy
pgpoolc
platform-apps
platform-services
rabbitmq
sasdatasvrc
sas-casserver-primary
sas-casserver-worker
va-apps
```

Notice that only three groups have any targeted machines assigned to them: sas-viya, sas-casserver-primary, and sas-casserver-worker. For this release of SAS Viya software, those three groups are the only ones that have software associated with them. The definition of the groups in terms of what software they deploy is described in the following table:

Group Name	Description
sas-viya	SAS Visual Data Mining and Machine Learning and associated products
sas-casserver-primary	the CAS controller (and SAS Event Stream Processing for SAS Viya, if it is in your order)
sas-casserver-worker	the CAS worker (and SAS Event Stream Processing for SAS Viya, if it is in your order)

The remaining groups in the list should have no listed targets, as shown in the preceding example.

Consider the following issues when editing the inventory file:

- It is strongly recommended that you do not remove any host groups from the list or any entries from the [sas-all:children] list unless you are an experienced Ansible user. A host group can have no entries under it, but the host group should not be removed even if it is empty. Removing a host group that contains targeted machines from the [sas-all:children] list can result in critical tasks not being executed on those targeted machines.
- If you are using HDFS, [sas-casserver-primary] and [sas-casserver-worker] should be assigned to machines in the Hadoop cluster.
- If Ansible is installed on the same machine that is assigned to [sas-casserver-primary] or [sas-casserver-worker], you must edit the vars.yml file. Remove the number sign (#) from the sasenv_cas_host variable and set the variable to the machine that is assigned to [sas-casserver-primary].
- If you purchased the optional SAS Event Stream Processing for SAS Viya component, it is automatically installed on all machines on which CAS components are installed. Installation of SAS Event Stream Processing for SAS Viya on the same machine with the CAS server enables two additional CAS action sets. An installation of SAS Event Stream Processing 3.2 or later as a separate component with a freestanding license is required as a data source.

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

Deploy the Software

Command Line

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

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

For a multi-machine deployment:

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

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

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

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

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

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.

Log On to SAS Studio

Perform the following steps to log on:

1 Open SAS Studio from a URL with this format:

```
http://host-name:38080
```

Use the host name from the machine that you assigned to the sas-viya Ansible group in the inventory file. For more information assigning machines, see Assign the Target Machines to Ansible Groups on page 31.

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.

Modify an Existing Deployment

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

Install with SAS 9.4 Software

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

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

- CAS_CONFIGURATION
- STUDIO_CONFIGURATION
- SPAWNER_CONFIGURATION

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

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

To change the value:

1 Remove the number sign from the beginning of the variable for the port number that you want to change.

2 Change the port value to the one that you want to use. Here is the earlier example revised in this way:

```
SPAWNER_CONFIGURATION:
  sasPort: 8592
```

3 Save and close the vars.yml file.

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

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

Deployment Logs

Logs for Ansible deployments are stored in **sas_viya_playbook/deployment.log**.

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

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

Manual Configuration Tasks

Ensure That the Same JRE Is Used across the Deployment

To ensure that the same JRE is used by both SAS Foundation and SAS Studio, perform the following steps:

1 Go to the machine that you have previously assigned to the sas-viya Ansible group in the inventory file. In the following example, main is the machine assigned to the sas-viya group in the inventory file:

```
[sas-viya]
main
```

For more information assigning machines, see Assign the Target Machines to Ansible Groups on page 31.

2 Using a text editor, open the **/opt/sas/viya/home/SASFoundation/bin/sasenv_local** file:

```
sudo vi /opt/sas/viya/home/SASFoundation/bin/sasenv_local
```

3 In the file, locate the block of text, `End of sasenv_local`, and insert the following code before the text block:

```
export SAS_USER_JAVA_HOME=path-to-the-JRE
export LD_LIBRARY_PATH="$SAS_USER_JAVA_HOME"/lib/amd64/server:"$SAS_USER_JAVA_HOME"/lib/amd64:
$LD_LIBRARY_PATH
```

Note: In the previous code, the length of the second export command requires two lines. However, if you copy and paste the command to your sasenv_local file, make sure that the command occupies only a single line.

Here is an example:

```
export SAS_USER_JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.101-3.b13.el6_8.x86_64/jre
export LD_LIBRARY_PATH="$SAS_USER_JAVA_HOME"/lib/amd64/server:"$SAS_USER_JAVA_HOME"/lib/amd64:
$LD_LIBRARY_PATH
################################################################################
# End of sasenv_local
################################################################################
```

4 Save and close **/opt/sas/viya/home/SASFoundation/bin/sasenv_local**.

Set the Password for the CAS Administrator or Another Administrative Account

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

```
sudo passwd cas
```

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

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

Configure the cas.colocation Variable for Multiple Machine or Co-located Deployments

1 On the CAS controller, navigate to the **/opt/sas/viya/config/etc/cas/default/** directory.

2 Open and edit the casconfig.lua file.

3 Add the value of the cas.colocation variable, as appropriate:

- If CAS is not co-located on the Hadoop cluster, add the cas.colocation variable and set the value to **none**, as follows:

  ```
  cas.colocation="none"
  ```

- If CAS is co-located on the Hadoop cluster, add the cas.colocation variable and set the value to **hdfs**, as follows:

  ```
  cas.colocation="hdfs"
  ```

4 Save and close the casconfig.lua file.

5 Copy the updated casconfig.lua file to all CAS worker nodes.

6 Restart the CAS controller:

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

Configure the SAS Data Connector to Impala

Note: This information is applicable only if you ordered SAS Data Connector 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 SAS Technical Support website support.sas.com.

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. Here are examples:

- Bourne Shell:

  ```
  $ LD_LIBRARY_PATH=$ODBCHOME/lib64:Impala-ODBC-driver-install-directory
  /lib/64:$LD_LIBRARY_PATH
  $ export LD_LIBRARY_PATH
  ```

  The variable *Impala-ODBC-driver-install-directory* represents the installation directory for the Impala ODBC driver.

- C Shell

  ```
  setenv LD_LIBRARY_PATH
  $ODBCHOME/lib64:Impala-ODBC-driver-install-directory/lib/64:${LD_LI
  BRARY_PATH}
  ```

  The variable $ODBCHOME represents the installation directory for the ODBC Driver Manager.

3 Set the environment variable for the driver. SAS Data Connector to Impala supports multiple ODBC driver for use with Impala. The default driver that is used by SAS Data Connector to Impala is the Cloudera Impala ODBC driver. The Cloudera driver requires that the CLOUDERAIMPALAINI environment variable be set to the path and filename of the .cloudera.impalaodbc.ini configuration file. For more information, see *Cloudera ODBC Driver for Impala Installation Guide*.

4 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. For more information about selecting a different driver vendor, see the SAS/ACCESS Interface to Impala on SAS Viya section of *SAS/ACCESS 9.4 for Relational Databases Reference*.

Configure the SAS Data Connector to ODBC

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

1 Edit the odbc.ini file in your ODBC 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.

2 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.

Configure the SAS Data Connector to Oracle

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

Ensure that the variable for the shared library path points to the location of the Oracle shared libraries. The name of this variable is operating system-dependent. This variable setting is required because the SAS Data Connector to Oracle executable must know the location of the Oracle shared libraries in order to use them.

Configure the SAS Data Connector to PostgreSQL

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

Provide connection specifics in one of the following ways:

- reference a Data Source Name (DSN)

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

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

- in your code

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

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

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

Validating the Deployment

Perform Installation Qualification on RPM Packages

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

Here is the basic command to verify RPM packages:

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

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

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

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

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

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

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

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

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

The error indicators are shown in the following format:

```
SM5DLUGT c
```

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

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

The meaning of each error indicator is described as follows:

- S - file size

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

- M - file mode

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

- 5 - MD5 checksum

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

- D - major and minor numbers

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

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

- L - symbolic link

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

- U - file owner

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

- G - file group

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

- M - modification time

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

- c - configuration file

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

Access CAS Server Monitor

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

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

Here is an example:

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

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

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

Note: To access CAS Server Monitor, the password must be set for the cas user ID or other administrative account. To set the password, see Set the Password for the CAS Administrator or Another Administrative Account on page 36.

Verify SAS Data Connector to ODBC

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

To verify that SAS Data Connector to ODBC was successfully deployed, run the following SAS code:

```
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 you do not see table names that you recognize, you should perform the configuration steps again.

Verify SAS Data Connector to Oracle

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

To verify that SAS Data Connector to Oracle was successfully deployed, run the following SAS code:

```
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 you do not see table names that you recognize, you should perform the configuration steps again.

Uninstalling SAS Viya

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 51.

Uninstall from a Single Machine

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

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

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

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

When the appropriate command is executed, Ansible performs a group uninstallation, which removes your SAS Viya software, including both certificates. It also renames the `/opt/sas/viya` directory to `/opt/sas/viya_<epoch>`, where `<epoch>` specifies the UNIX epoch (the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970). The uninstallation does not remove the customized script that you received with your SOE, and it does not remove any users that have been set up.

Uninstall from Multiple Machines

To uninstall your SAS Viya software from a deployment with more than one machine, run the following command:

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

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

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

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

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

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

Repeat this step to uninstall each CAS worker.

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

Next Steps

After you validate the deployment, you can do the following:

- Perform initial administrative tasks and start using the software. For more information, refer to the guides on the SAS Viya Documentation page. Also, help is available from the SAS Viya product and administrative interfaces.
- Refer to the appendices in this guide for additional tasks that you might perform, based on your environment.
- Start using SAS Event Stream Processing for SAS Viya (optional installation). A freestanding installation of SAS Event Stream Processing is required to provide data for the supported CAS actions. A separate *Deployment Guide* is provided. The product user documentation is included in SAS Help Center. A link to all SAS Event Stream Processing documentation 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.

Appendix A: Deploying with Yum

Use this appendix for instructions to deploy only the elements of the programming interface of your SAS Viya software on a single machine.

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 saved from the Software Order Email (SOE). 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
```

- SAS Studio:

```
netstat -an |grep 38080
```

If the output from the command is empty, the port is available and no changes are required. If you receive any output, the port is not available. 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 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.

- SAS Studio:

  Open the **/opt/sas/viya/config/etc/sasstudio/default/init_usermods.properties** file. Add the following property:

  ```
  sasstudio.appserver.port=available-port-number
  ```

Apply the Licenses for SAS and CAS Software

1 Locate the license file that you previously saved.

2 Run the command to apply the license to your SAS software:

```
sudo su -s "/bin/sh" -c "/opt/sas/viya/home/SASFoundation/utilities/bin/apply_license /opt/sas
/installfiles/license-file-name" sas
```

You receive a message that your license has been applied.

3 To apply the license to CAS, copy your license file into the CAS default configuration directory. Here is an example:

```
sudo cp /opt/sas/installfiles/license-file-name /opt/sas/viya/config/etc/cas/default/
```

4 Open the config.lua file:

```
sudo vi /opt/sas/viya/config/etc/cas/default/casconfig.lua
```

5 Locate the env.CAS_LICENSE variable in the casconfig.lua file, and specify the directory path that contains the license. Here is an example:

```
env.CAS_LICENSE = config_loc .. '/SASViyaV0300_09J9P5_Linux_x86-64.txt'
```

6 Save and close the casconfig.lua file.

Register Your SAS Software

Perform the following command to register your SAS Viya software:

```
sudo su -s "/bin/sh" -c "/opt/sas/viya/home/SASFoundation/utilities/bin/post_install build_registry" sas
```

You receive a message that the build registry tasks have completed.

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. As an alternative, 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 FIRSTADMINUSER variable with the name of a user that exists and that can log on. Here is an example of the two instances:

```
<Administrator name="FIRSTADMINUSER-User-SuperUser" user="FIRSTADMINUSER" 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 runs 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.

```
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
```

Configure the SAS Data Connector to Hadoop and the SAS Data Connect Accelerator for Hadoop

The information in this section is applicable only if you ordered SAS Data Connector to Hadoop or SAS Data Connect Accelerator for Hadoop.

Follow these steps to configure CAS access to the data source:

1 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

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`** line. If you are using the JRE that is installed with your SAS software, its default location is **`/usr/lib/jvm/jre-1.8.0`**.

3 Save and close the cas.settings file.

Configure the SAS Data Connector to Impala

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

Follow these steps to configure SAS Data Connector to Impala:

1 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

2 Add the following lines:

```
export CLOUDERAIMPALAINI=location-of-your-cloudera.impalaodbc.ini-file
export SIMBAINI=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
```

3 Save and close the cas.settings file.

Configure the SAS Data Connector to ODBC

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

Follow these steps to configure SAS Data Connector 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 Using a text editor, open the cas.settings file.

```
sudo vi /opt/sas/viya/home/SASFoundation/cas.settings
```

3 Add the following lines that are appropriate for the version of ODBC that you are using.

- DataDirect:

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

- 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=$LD_LIBRARY_PATH:$ODBCHOME/lib
```

- 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 you do not have ODBCSYSINI set in your environment, then ODBCINI and ODBCINSTINI should be full paths to the respective files, including the filenames.

4 Save and close the cas.settings file.

Configure the SAS Data Connector to Oracle

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

Follow these steps to configure SAS Data Connector to Oracle:

1 Ensure that the variable for the shared library path points to the location of the Oracle shared libraries. The name of this variable is operating system-dependent. This variable setting is required because the SAS Data Connector to Oracle executable must know the location of the Oracle shared libraries in order to use them.

2 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller. Add the following lines:

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

3 Save and close the cas.settings file.

Configure the SAS Data Connector to PostgreSQL

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

Follow these steps to configure SAS Data Connector to PostgreSQL:

1 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

2 Add the following lines:

```
export ODBCINI="location-of-your-odbc.ini-file"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:location-of-ODBC-driver-manager-used-with-PostgreSQL-ODBC-driver:
/opt/sas/viya/home/lib64/lib
```

3 Save and close the cas.settings file.

Configure the SAS Data Connector to Teradata

The information in this section is applicable only if you ordered SAS Data Connector toTeradata.

Follow these steps to configure SAS Data Connector toTeradata:

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 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

4 Add the following lines:

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

Here is an example of the TTU default LD_LIBRARY_PATH:

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

5 Save and close the cas.settings file.

Configure Settings for SAS Event Stream Processing for SAS Viya

The information in this section is applicable only if you ordered SAS Event Stream Processing for SAS Viya .

Follow these steps to configure CAS settings for SAS Event Stream Processing:

1 Locate the cas.settings file in the `/opt/sas/viya/home/SASFoundation` directory on the CAS controller.

2 Use a text editor to modify the following file: **`/opt/sas/viya/home/SASFoundation/cas.settings`**

3 Add the following lines to the cas.settings file:

```
export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/4.2.0
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DFESP_HOME/lib
export TKPATH=$TKPATH:$DFESP_HOME/lib/tk
```

4 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:

```
http://<hostname>:38080
```

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 40.

Uninstall SAS Viya with Yum

Perform the following steps to uninstall your SAS Viya software with yum:

1 Stop all the services with the following command:

```
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 -qg SAS)
```

7 Remove the entitlement certificate with the following command:

```
sudo rm /etc/pki/sas/private/entitlement_certificate.pem
```

8 Rename the **viya** directory with the following command:

```
sudo mv /opt/sas/viya/ /opt/sas/viya_$(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 B: Creating and Using Mirror Repositories

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

General Requirements

The environment must meet the following requirements.

The Ansible controller must have the following:

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

The mirror repository host must have the following:

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

Methods to Create a Mirror Repository

You can create a mirror repository using either of two methods: Ansible to orchestrate the required steps or yum and shell scripting. Ansible should be used in most scenarios. For more information about installing Ansible, see Install Ansible on page 21.

Use Ansible

These instructions assume that Ansible has been installed on a separate machine from the machine on which the software will be deployed. Perform the following steps to create and use a mirror repository using Ansible:

1. Go to the directory that was created when you uncompressed your playbook from the Software Order Email (SOE). For more information, see Make Sure That You Have the Required Files on page 20.
2. Using a text editor, create a file named repohost.ini that contains the following content:

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

   [repohost]
   machine-name
   ```

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

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

   The results of running this command follow:

- The host *machine-name* begins running as an Apache httpd server from which the deployment process obtains files.
- The file **/var/www/html/pulp/repos/repo.override.txt** is created.
- A new customized_deployment_script.sh file is created, and is located in**/tmp/mirror/location/**. It will be modified to use the mirror repository.
- The Ansible controller downloads all the RPM files.
- The RPMs are copied to *machine-name*.

If httpd is not installed, run the following command:

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

The results of running this command follow:

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

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

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

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

7 Perform the steps that are described in Deploy with the Ansible Playbook on page 24 up to Deploy the Software on page 33. Do not run the command in that step. Instead, run the following command:

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

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

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

8 After the deployment has run, continue with the deployment instructions at Manual Configuration Tasks on page 36.

Use Yum

Perform the following steps to create and use a mirror repository using yum.

1 On the machine where you intend to host the mirror repository, run the following command:

```
sudo yum install yum-utils createrepo httpd
```

2 Copy the SAS_Viya_playbook.tgz file from your Software Order Email (SOE) to the repository mirror host.

3 Extract the files from SAS_Viya_playbook.tgz:

```
tar xf SAS_Viya_playbook.tgz
```

4 Make a copy of the customized_deployment_script.sh file and change its name to setup_repos.sh:

```
cp customized_deployment_script.sh setup_repos.sh
```

5 Use a text editor to open setup_repos.sh.

6 Comment out the last statement in setup_repos.sh by adding a number sign (#) to the beginning of the line. 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"
```

7 Save and close setup_repos.sh.

8 As root, run setup_repos.sh to configure the required certificates and the SAS hosted repositories. Running the script enables the repository mirror host to mirror the content.

9 Using a text editor, create a new file named createrepos.sh that contains the following content:

```
#!/bin/bash
rpm --import /etc/pki/sas/rpm-gpg/RPM-GPG-KEY-sas
mkdir -p /var/www/html/pulp/repos
for f in $(ls /etc/yum.repos.d/sas-*.repo | cut -f4 -d/ | sed s/.repo//g | grep -v sas-meta)
do
  reposync --gpgcheck -l -n --repoid=${f} \
           --download_path=/var/www/html/pulp/repos --downloadcomps --download-metadata
  cd /var/www/html/pulp/repos/${f}
  createrepo -v /var/www/html/pulp/repos/${f}/ -g comps.xml
done
```

10 Set the execute bit for createrepos.sh:

```
sudo chmod +x createrepos.sh
```

11 Run createrepos.sh:

```
sudo createrepos.sh
```

12 Using a text editor, 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.

13 Restart or reload the httpd service as needed.

a Check the status of the httpd service:

```
sudo service httpd status
```

b If httpd is running, reload it:

```
sudo service httpd reload
```

c If httpd is not running, start it:

```
sudo service httpd start
```

Note: If your repository mirror host cannot access the Internet, the steps to this point can be performed on a server that can contact the Internet. The content of **/var/www/html/pulp/repos** should then be packed into a TAR file and moved to the designated repository mirror host and unpacked in **/var/www/html/pulp**. On that new host, ensure that httpd is installed. Create the repo.conf file, which is described in step 12. Reload the httpd configuration as described in step 13.

14 Using a text editor, create **/etc/yum.repos.d/sas.repo** as follows.

a Open setup_repos.sh.

b Locate the line that begins with `yum install`. Here is an example:

```
yum install "sas-connect-v.03.01.0-rpm" "sas-va-8.1.0-rpm" "sas-mchnlrng-8.1.1-rpm"
"sas-statviya-8.1.0-rpm"
```

c Create a new file named **/etc/yum.repos.d/sas.repo** .

d Open the file that you just created:

```
sudo vi /etc/yum.repos.d/sas.repo
```

e For each repository that is listed in the yum install command of setup_repos.sh, add the corresponding block from the following list to sas.repo. Be sure to include the repository name in brackets at the beginning of each block.

For example, if the yum install command in setup_repos.sh includes `sas-pcfile-v.03.01.0-rpm`, add the block that begins `[sas-pcfile-v.03.01.0-rpm]` to sas.repo.

In each block, replace *mirror-repository-host-address* with the address of the mirror repository host.

Here is an example of the full list of repositories:

```
[sas-pcfile-v.03.01.0-rpm]
name = pcfile-v.03.01.0-rpm
baseurl=mirror-repository-host-address/pulp/repos/sas-pcfile-v.03.01.0-rpm/
enabled=1
sslverify=0
sslcacert=
sslclientcert=
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x

[sas-connect-v.03.01.0-rpm]
name = connect-v.03.01.0-rpm
baseurl=mirror-repository-host-address/pulp/repos/sas-connect-v.03.01.0-rpm/
enabled=1
sslverify=0
sslcacert=
sslclientcert=
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x

[sas-va-8.1.0-rpm]
name = va-8.1.0-rpm
baseurl=mirror-repository-host-address/pulp/repos/sas-va-8.1.0-rpm/
enabled=1
sslverify=0
sslcacert=
sslclientcert=
gpgcheck=0
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x

[sas-statviya-8.1.0-rpm]
name = statviya-8.1.0-rpm
baseurl=mirror-repository-host-address/pulp/repos/sas-statviya-8.1.0-rpm/
enabled=1
sslverify=0
sslcacert=
sslclientcert=
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x

[sas-mchnlrng-8.1.1-rpm]
name = mchnlrng-8.1.1-rpm
baseurl=mirror-repository-host-address/pulp/repos/sas-mchnlrng-8.1.1-rpm/
enabled=1
sslverify=0
sslcacert=
sslclientcert=
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-SAS-9.x
```

f Save and close the setup_repos.sh and sas.repo files.

15 Copy the sas.repo file to **/etc/yum.repos.d/sas.repo** on each machine on which SAS Viya will be installed.

16 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 (if SAS/CONNECT is not included in your software order, you can step this command):

  ```
  netstat -an | grep 17551
  ```

- SAS Studio:

  ```
  netstat -an | grep 38080
  ```

If a command does not produce any output, then the port is available and no changes are required. If you receive any output, then the port is not available. Make a note of any blocked product for additional steps to be taken after the deployment has been performed.

17 Copy the final yum groupinstall command from customized_deployment_script.sh. 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"
```

18 Run the copied yum groupinstall command on each machine on which SAS Viya will be installed.

19 If you have any blocked products from step 16, 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 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.

- SAS Studio:

  Open the **/opt/sas/viya/config/etc/sasstudio/default/init_usermods.properties** file. Add the following property:

  ```
  sasstudio.appserver.port=available-port-number
  ```

20 Finish the deployment by following the steps described in Appendix A: Deploying with Yum on page 46, starting with Apply the Licenses for SAS and CAS Software on page 47.

Appendix C: Hadoop Deployment: Configuring the SAS Data Connector to Hadoop and Optionally, the SAS Data Connect Accelerator for Hadoop

Supported Hadoop Distributions

Before you set up Hadoop, you must make sure that your Hadoop distribution is supported by SAS Viya. For details, see Supported Releases of Hadoop Distributions on page 15.

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 59.
2. Deploy the Hadoop JAR files. For more information, see Configure SAS Data Connector to Hadoop for Serial Processing on page 61.
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 for Hadoop for Parallel Processing on page 64.

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.
- 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. This user account is used to manually deploy the SAS Embedded Process in the Deploy Manually on page 65 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 information about CAS settings for YARN, refer to *SAS Viya Administration: SAS Cloud Analytic Services*.
- 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 Data Connector to Hadoop for Serial Processing

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

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 (which are included in your version of Linux) have been installed.
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.
      - Permission to write to the directories that are specified in the hadoop.client.jar.filepath property and the hadoop.client.config.filepath property.

- 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 from which the script collects JAR files and configuration files from the Hadoop cluster:

```
hadoop.client.jar.filepath=directory-path
hadoop.client.config.filepath=directory-path
```

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
```

g Set the directory in which the script creates a log filename. The script creates the file sashadoopconfig-*time-stamp*.log. An example of a filename is sashadoopconfig-2016-03-29-12.54.39.logsashadoop-timestamp.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 consoleAappender 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
```

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.

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 your run the hadoop_extract.sh script.

Verify SAS Data Connector to Hadoop

To verify that the software has been successfully deployed, run the following SAS code:

```
cas mysession;

caslib hivelib datasource=(srctype="hadoop" server="server-name"
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;
list files incaslib="hivelib";
run;
```

If the data connector is successfully deployed, the results are a list of the names of tables in the Hive data source.

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.

Deploy the SAS Embedded Process for Hadoop for Parallel Processing

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.

Uninstall the SAS Embedded Process for 9.4

Options for Uninstallation

Before you install the SAS Embedded Process for Viya, you must uninstall the SAS Embedded Process for 9.4 version on your Hadoop cluster.

CAUTION! You must uninstall the SAS 9.4 Embedded Process using the same method that you used to install the SAS 9.4 Embedded Process.

- To manually uninstall, run the following command:

  ```
  sasep-admin-sh -remove
  ```
- To uninstall with Cloudera Manager, see Uninstall with Cloudera Manager on page 64.
- To uninstall with Ambari, see Uninstall with Ambari on page 65.

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 Run the following command to delete the stack:

```
./delete_stack.sh Ambari-Admin-User-Name
```

2 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.

3 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.

4 Enter **y** to restart the Ambari server. The SASEP service is no longer listed on Ambari dashboard user interface.

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 65.

  TIP Many options are available for installing the SAS Embedded Process.For more information, see SASEP-ADMIN.SH Script on page 70.

- To deploy with your appropriate cluster manager:
  - To deploy with Cloudera Manager, see Deploy the SAS Embedded Process with Cloudera Manager on page 68.
  - To deploy with Hortonworks Ambari, see Deploy the SAS Embedded Process with Ambari on page 69.

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-11.00000-1.sh.file.

4 Copy the sepcorehadp-11.00000-1.sh. file from the client to the *EPInstallDir* on the Hadoop cluster. Here is an example that uses secure copy.

```
scp sepcorehadp-11.00000-1.sh username@hdpclus1:/EPInstallDir
```

Note: The location to which you transfer the sepcorehadp-11.00000-1.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. For more information, see Hadoop Prerequisites on page 64.

1 Navigate to the location on your Hadoop master node to which you copied the sepcorehadp-11.00000-1.sh file.

```
cd /EPInstallDir
```

2 Use the following command to unpack the sepcorehadp-11.00000-1.sh file.

```
./sepcorehadp-11.00000-1.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-11.00000-1.sh
```

Note: During the installation process, the sepcorehadp-11.00000-1.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/utilities
```

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
06/20/15 09:51:26 06/27/15 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
     renew until 06/22/15 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 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 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-11.00000-1.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 SAS Embedded Process:

1 On the CAS controller machine, navigate to the **/opt/sas/viya/home/share/ep** directory.

2 Copy the parcel directory to the tmp directory of the file system of the host on which Cloudera Manager is installed.

3 From the **tmp** directory, run the following command:

Note: The user account that you use to run the script must have super user (sudo) or root access.

```
./install_parcel.sh  -v distro
```

The tmp directory is the location to which you copied the parcel directory from the Viya installation. The variable *distro* represents one of the following Linux distributions: redhat5, redhat6, suse11x, ubuntu10, ubuntu12, ubuntu14, debian6, or debian7.Select the appropriate value.

Here is an example:

```
./install_parcel.sh  -v redhat6
```

4 When prompted to restart Cloudera Manager, select **Y**.

5 Log on to Cloudera Manager.

6 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 SASEP 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 When prompted, click **Close**.

7 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. Choose any node that is part of your cluster. Click **OK**, and then click **Continue**.

A file is added to HDFS for each of the services as follows:

SASEP: **`/sas/ep/config/ep-config.xml`**

f Enter your hdfs user name. The default user name is hdfs. If your cluster is Kerberos enabled, a valid Kerberos ticket for your hdfs user name must be available on the node that was selected for the SAS Embedded Process service. The configuration file /sas/ep/config/ep-config.xml is now added to the hdfs file system.

g Click **Continue**, and then click **Finish**.

Note: If the services that you have just deployed are 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/`** directory. Copy the entire share directory to a temporary directory on your Hadoop cluster machine.

b Navigate to the **/opt/sas/viya/home/share/ep/stack** directory and run the following command:

```
./install_sasepstack.sh ambariAdminUsernam
```

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.

The **Customize Services** panel appears.

The SASEP service stacks are 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 appears. Enter your Kerberos credentials in the **admin_principal** and **admin_password** text boxes.

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.

e 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.

f Click **Complete**. The stacks are now installed on all nodes of the cluster.

The SASEP service is displayed on the Ambari dashboard.

g 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**

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 the 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 74 and -host on page 74

-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.

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. Deploy Manually on page 65

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 74 and -host on page 74

- 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 74 and -host on page 74

-x on page 74

“Encrypt Data Transfer when Using the SAS Data Connect Accelerator” in *Encryption in SAS Viya 3.1*

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 74 and -host on page 74

-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 74 and -host on page 74

-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 -hdfs on page 74

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 -hdfs on page 74

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, 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 61.

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/2.3.0.0-2557/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.

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 78.

- the number of SAS Embedded Process MapReduce tasks per node

  For more information, see Specify the Number of MapReduce Tasks on page 78.

- 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 78.

- the buffers for input data

  For more information, see Specify the Number of Input Buffers and an Optimal Buffer Size on page 78.

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 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 is allowed to use.

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 SASEP-ADMIN.SH Syntax on page 71.

Appendix D: Hadoop Deployment: Configuring CAS SASHDAT Access to HDFS

Supported Hadoop Distributions

Before you set up Hadoop, ensure that your Hadoop distribution is supported by SAS Viya. For more information, see Supported Releases of Hadoop Distributions on page 15.

Overview of Deployment Tasks for HDFS for Existing Hadoop Clusters

During installation, your CAS software was deployed to the nodes on your Hadoop cluster. For an overview of this deployment scenario, see Hadoop Scenario 2: Access to Data in SASHDAT on HDFS on page 8.

To configure your existing Hadoop cluster:

1. Perform the Hadoop pre-deployment tasks.Pre-deployment Checklist for HDFS and the Existing Hadoop Clusters on page 80.
2. Configure your implementation of Hadoop:
   - For Apache, see Configure the Existing Apache Hadoop Cluster to Interoperate with the CAS Server on page 82.
   - For Cloudera, see Configure the Existing Cloudera Hadoop Cluster to Interoperate with the CAS Server on page 83.
   - For Hortonworks, see Configure the Existing Hortonworks Data Platform Hadoop Cluster to Interoperate with the CAS Server on page 87.
3. Verify CAS SASHDAT Access to HDFS. For details, see Verify CAS SASHDAT Access to HDFS on page 90.

Pre-deployment Tasks for HDFS

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.

- Each machine in the cluster must be able to resolve the host name of all the other machines in the cluster.
- The NameNode and the secondary NameNode cannot be defined as the same host.
- 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
  ```

- 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 Kerberos and Secure Shell (SSH), review the requirements and perform the appropriate tasks in Kerberos Requirements on page 81 and Review the Passwordless Secure Shell Requirements on page 81.

Kerberos Requirements

SAS Viya operates with Kerberos as follows:

- 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.
- 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 Java is set up appropriately.

- 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.
- If you are using the Oracle JRE or the IBM JRE, use the two JAR files in place of the existing local_policy.jar file and the US_export_policy.jar file. These files are located in your JRE location. This location is typically the **`JAVA_HOME/jre/lib/security/`** directory. These files can be obtained from the IBM or Oracle website.
- 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, the files do not need to be replaced.

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
```

Review the Passwordless Secure Shell Requirements

Here are the passswordless 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 on page 82.
- Passwordless SSH is required on all machines in the Hadoop cluster for the user accounts that run a CAS session. By default, the user account is the cas user, and 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.

Configure Passwordless SSH to Use Kerberos

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
```

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.

Configure the Existing Apache Hadoop Cluster to Interoperate with the CAS Server

1 Locate your SAS Viya installation directory on the CAS controller and then locate the following files:

```
/opt/sas/viya/home/SASFoundation/hdatplugins/SAS_VERSION
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.cas.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.grid.provider.yarn.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.lasr.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sascasfd
/opt/sas/viya/home/SASFoundation/hdatplugins/sashdfsfd
/opt/sas/viya/home/SASFoundation/hdatplugins/start-namenode-cas-hadoop.sh
```

2 If you cannot locate the **`/opt/sas/viya/home/SASFoundation/hdatplugins`** directory, then install the RPM package sas-hdatplugins-timestamp.x86_64.rpm.

```
sudo rpm -i /sas-hdatplugins-03.00.00-20160315.083831547133.x86_64.rpm
```

3 Change directories to the location of the full path that corresponds to **`$HADOOP_PREFIX`**.

Note: `$HADOOP_HOME` has been deprecated.

4 Create a new subdirectory **`sas`** under the **`$HADOOP_PREFIX /share/hadoop/`** directory.

5 Locate and propagate to the following three JAR files to the **`$HADOOP_PREFIX/share/hadoop/sas`** directory on each machine in the Apache Hadoop cluster:

```
sas.cas.hadoop.jar
sas.lasr.hadoop.jar
sas.grid.provider.yarn.jar
```

6 Locate the sashdfsfd file, the sascasfd file, and the start-namenode-cas-hadoop.sh file, and propagate them to the **`$HADOOP_PREFIX/bin`** directory on each machine in the Apache Hadoop cluster.

7 Locate the SAS_VERSION file and propagate it to the **$HADOOP_PREFIX** directory on each machine in the Apache Hadoop cluster.

8 On the machine to which you initially installed Apache Hadoop, add the following SAS properties for the HDFS configuration to the **$HADOOP_PREFIX/etc/hadoop/hdfs-site.xml** file:

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>
```

Configure the Existing Cloudera Hadoop Cluster to Interoperate with the CAS Server

Overview of Deployment Methods

You can either deploy manually or deploy automatically by using the cluster manager for your Hadoop distribution:

CAUTION! If you are using Red Hat Enterprise Linux versions 7.1 and later within 7.x, then you must deploy manually.

- To deploy manually, see Deploy Manually on page 83.
- To deploy with Cloudera Manager, see Deploy with Cloudera Manager on page 85.

Deploy Manually

Use Cloudera Manager to configure your existing Cloudera Hadoop (CDH 5) deployment to interoperate with the CAS server.

1 Locate your SAS Viya installation directory on the CAS controller and then locate the following files:

```
/opt/sas/viya/home/SASFoundation/hdatplugins/SAS_VERSION
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.cas.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.grid.provider.yarn.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.lasr.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sascasfd
```

```
/opt/sas/viya/home/SASFoundation/hdatplugins/sashdfsfd
/opt/sas/viya/home/SASFoundation/hdatplugins/start-namenode-cas-hadoop.sh
```

2 If you cannot locate the **/opt/sas/viya/home/SASFoundation/hdatplugins** directory, then you must install the RPM package sas-hdatplugins-timestamp.x86_64.rpm.

```
sudo rpm -i /sas-hdatplugins-03.00.00-20160315.083831547133.x86_64.rpm
```

3 Locate the full path of the cluster. The default HADOOP_PREFIX location is the **/opt/cloudera/parcels/CDH-*version*** directory.

4 Locate and propagate the following three JAR files to the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop/lib** directory on each machine in the cluster.

```
sas.cas.hadoop.jar
sas.lasr.hadoop.jar
sas.grid.provider.yarn.jar
```

5 Locate the sashdfsfd file, the sascasfd file, and the start-namenode-cas-hadoop.sh file, and propagate them to the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop/bin** directory of each machine in the CDH cluster. Here is an example:

```
/opt/cloudera/parcels/CDH-version/lib/hadoop/bin
```

6 Locate the SAS_VERSION file and propagate it to the **/opt/cloudera/parcels/CDH-*version*/lib/hadoop/** directory on each machine in the CDH cluster.

7 Log on to Cloudera Manager as an administrator.

8 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
```

9 Add the following lines to the advanced configuration for service-wide configuration. 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>
```

10 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>
```

Note: When Cloudera Manager prioritizes the HDFS client configuration, the client safety valve is used. When Cloudera Manager prioritizes anything else (such as YARN), the service safety valve is used. A best practice is to update the values of the safety valves. For more information, see the Cloudera documentation.

11 If you are not using the version of Java that is supplied with Cloudera, add the location of the JAVA_HOME variable. Navigate to the **Gateway Default Group**. Under **Advanced**, add the location of the JAVA_HOME variable to the **HDFS Client Environment Advanced Configuration Snippet for hadoop-env.sh (Safety Valve)** property. Here is an example:

```
JAVA_HOME=/usr/lib/java/jdk1.7.0_07
```

12 Save your changes and deploy the client configuration to each machine in the Hadoop cluster.

13 In Cloudera Manager, restart the HDFS service and any dependencies.

14 To test the Hadoop cluster with SAS test jobs, create the **`/test`** directory in the HDFS. You might need to first set the HADOOP_HOME variable. To set the HADOOP_HOME variable, run the following commands as the user that is running HDFS, which is typically hdfs.

```
$HADOOP_HOME/bin/hadoop fs -mkdir /test
$HADOOP_HOME/bin/hadoop fs -chmod 777 /test
```

Deploy with Cloudera Manager

CAUTION! If you are using Red Hat Enterprise Linux versions 7.1 and later within 7.x, then you must deploy manually.

The following deployment steps assume that the hdatplugins rpm package has been installed directly on one of the following machines:

- the Cloudera Manager server
- on a machine in the network that is accessible to the Cloudera Manager server

CAUTION! When the Cloudera Hadoop parcel is upgraded, the HDATPlugins parcel must be deactivated and then reactivated.

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.

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 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.

6 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
```

7 Add the following lines to the advanced configuration for service-wide configuration. 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>
```

8 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>
```

Note: When Cloudera Manager prioritizes the HDFS client configuration, the client safety valve is used. When Cloudera Manager prioritizes anything else (such as YARN), the service safety valve is used. A best practice is to update the values of the safety valves. For more information, see the Cloudera documentation.

9 Navigate to the **HDFS Environment Client Safety Valve** and add the following property to both the **HDFS (Service-wide)** and **Gateway Default Group** fields:

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/cloudera/parcels/SASHDAT/*
```

10 If you are not using the version of Java that is supplied with Cloudera, add the location of the JAVA_HOME variable. Navigate to the **Gateway Default Group**. Under **Advanced**, add the location of the JAVA_HOME variable to the **HDFS Client Environment Advanced Configuration Snippet for hadoop-env.sh (Safety Valve)** property. Here is an example:

```
JAVA_HOME=/usr/lib/java/jdk1.7.0_07
```

11 Save your changes and deploy the client configuration to each machine in the Hadoop cluster.

12 In Cloudera Manager, restart the HDFS service and any dependencies.

13 To test the Hadoop cluster with SAS test jobs, create the **`/test`** directory in the HDFS. You might need to first set the HADOOP_HOME variable. To set the HADOOP_HOME variable, run the following commands as the user that is running HDFS, which is typically hdfs.

```
$HADOOP_HOME/bin/hadoop fs -mkdir /test
$HADOOP_HOME/bin/hadoop fs -chmod 777 /test
```

Configure the Existing Hortonworks Data Platform Hadoop Cluster to Interoperate with the CAS Server

Overview of Deployment Methods

- To deploy manually, see Deploy Manually on page 88.
- To deploy with Ambari, see Deploy with Ambari on page 89.

Deploy Manually

Use the Ambari interface to configure your existing Hortonworks Data Platform (HDP) deployment to interoperate with the CAS server.

1 Locate your SAS Viya installation directory on the CAS controller and then locate the following files:

```
/opt/sas/viya/home/SASFoundation/hdatplugins/SAS_VERSION
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.cas.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.grid.provider.yarn.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sas.lasr.hadoop.jar
/opt/sas/viya/home/SASFoundation/hdatplugins/sascasfd
/opt/sas/viya/home/SASFoundation/hdatplugins/sashdfsfd
/opt/sas/viya/home/SASFoundation/hdatplugins/start-namenode-cas-hadoop.sh
```

2 If you cannot locate the **/opt/sas/viya/home/SASFoundation/hdatplugins** directory, then you must install the RPM package sas-hdatplugins-timestamp.x86_64.rpm.

```
sudo rpm -i /sas-hdatplugins-03.00.00-20160315.083831547133.x86_64.rpm
```

3 Locate the full path of the HDP cluster. The default location is the **/usr/hdp/*version*/hadoop** directory.

4 Locate and propagate the following three JAR files to the **/usr/hdp/*version*/hadoop/lib** directory on each machine in the HDP Hadoop cluster:

```
sas.cas.hadoop.jar
sas.lasr.hadoop.jar
sas.grid.provider.yarn.jar
```

5 Locate the sashdfsfd file, the sascasfd file, and the start-namenode-cas-hadoop.sh file, and propagate them to the **bin** directory on each machine in the HDP cluster:

```
/usr/hdp/version/hadoop/bin
```

6 Locate the SAS_VERSION file and propagate it to the following directory on each machine in the HDP cluster:

```
/usr/hdp/version/hadoop/
```

7 In the Ambari interface, create a custom hdfs-site.xml file and add the following properties:

   a Click **HDFS Service**.

   b Choose **Config Section**.

   c Click **Advanced**.

   d 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

   com.sas.cas.hadoop.service.datanode.port
   15453

dfs.namenode.fs-limits.min-block-size
0

8 Save the properties and restart all HDP services and MapReduce services.

9 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
```

Deploy with Ambari

The following deployment steps assume that the hdatplugins rpm package is installed directly on one of the following machines:

CAUTION! When the Hortonworks Hadoop parcel is upgraded, the HDATPlugins parcel 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 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.

In the Ambari interface, create a custom hdfs-site.xml file and add the following properties:

a Click **HDFS Service**.

b Choose **Config Section**.

c Click **Advanced**.

d 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`

com.sas.cas.hadoop.service.datanode.port
`15453`

dfs.namenode.fs-limits.min-block-size
`0`

e Save the properties and restart all HDP services and MapReduce services.

f 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
```

Verify CAS SASHDAT Access to HDFS

Note: Before you perform the steps to verify CAS SASHDAT access to HDFS, ensure that you have installed the CAS controller and the worker nodes on the Hadoop cluster. In addition, ensure that you have completed the tasks in the section Configure the cas.colocation Variable for Multiple Machine or Co-located Deployments on page 37.

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.
```

Appendix E: SAS In-Database Deployment: Configuring SAS Viya to Access Teradata

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 for Hadoop 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, see Install the Support Functions for the SAS Embedded Process on page 93.

Connections from SAS 9.4 Clients

The following SAS 9.4 clients can connect to a Teradata server on which the SAS Viya version of the SAS Embedded Process for Teradata has been installed:

- 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 92.
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 93.
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 *SAS Viya Administration: Encryption*.

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-11.00000-*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-11.00000-*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-11.00000-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 92.
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-11.00000-*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
   ```

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 (SASTblOp). 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 theTeradata maintenance release 15.00.04.

Appendix F: Troubleshooting

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

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

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