SAS® Event Stream Processing 5.2 on Linux: Deployment Guide

SAS® Documentation
November 2, 2018

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS® Event Stream Processing 5.2 on Linux: Deployment Guide*. Cary, NC: SAS Institute Inc.

SAS® Event Stream Processing 5.2 on Linux: Deployment Guide

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

November 2018

5.2-P1:dplyesp0phy0lax

Contents

1

Introduction

Steps for a Successful Deployment

Before You Begin

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

  SAS Viya Deployment Guides

  If you accessed this guide directly from the Software Order Email, you are viewing the latest guide. If you are viewing a saved copy of the PDF version of this guide, the content might be outdated.
- To use this guide successfully, you should have a working knowledge of Ansible and the Linux operating system.
- In this guide, a single-machine deployment, such as a test environment, is referred to as a “rapid deployment.” A deployment that is performed across one or more machines and includes enhanced security is referred to as a “production deployment.”
- If your order included SAS Event Stream Manager, it must be installed on a separate machine from SAS Event Stream Processing.
- SAS Event Stream Processing 5.2 is compatible with both SAS 9.4 and SAS Viya. When installed along with the Cloud Analytic Services (CAS) components, SAS Event Stream Processing can provide data for analytic processing in SAS Viya. It uses the same deployment tools and processes as SAS Viya. However, SAS Event Stream Processing can also be installed as a stand-alone product without additional SAS Viya components.

Step 1 — Prepare for the Deployment

1 Perform one of the following tasks:
   - To upgrade or update an existing deployment, go directly to “Managing Your Software” on page 39.
   - To deploy a new instance of the software, continue with the following the steps.

2 Go to “System Requirements” on page 3 to learn about requirements for hardware, software, security, and clients, and more.

3 Go to “Pre-installation Tasks” on page 9 to prepare your environment before you deploy the software.

4 To perform a production deployment, perform the following tasks:

   a Go to “Preparing for a Production Deployment” on page 57 to perform the recommended configuration steps.

   b Go to “Optional Configuration for Enhanced Security” on page 60 to perform the recommended steps to secure your environment.

Step 2 — Perform the Deployment

1 Perform one of the following tasks:

   - To perform a rapid deployment, go to “Installing a Rapid Deployment” on page 21.
   - To perform a production deployment, go to “Optional Steps to Install on Multiple Machines” on page 62.

2 Go to “Post-installation Tasks” on page 27 to perform post-installation configuration.

Step 3 — Validate and Complete the Deployment

1 Go to “Validating the Deployment” on page 35 to verify that the servers were deployed correctly and to locate the log files.

2 Go to “Completing the Deployment” on page 51 for best practices after deployment, including where to find additional documentation.

Contact SAS Technical Support

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

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

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

2

System Requirements

Hardware Requirements

The topics in this section provide information about hardware requirements for a SAS Event Stream Processing deployment. If your order included SAS Event Stream Manager, a few additional requirements also apply to your system. This information is also included in this section.

General Hardware Considerations

SAS Event Stream Processing has a flexible architecture and a base set of features that have no dependencies on SAS Foundation or on SAS Viya. The SAS Event Stream Processing software is licensed per event, so you can install the software on multiple machines without violating the license agreement.

To use SAS Foundation in SAS Event Stream Processing deployments, as when, for example, you want to run SAS in a procedural window, SAS Event Stream Processing must be installed on the same machine as SAS Foundation. Depending on your version of SAS, a SAS/ACCESS engine might also be required. The following hardware requirements do not attempt to account for all usage scenarios.

Hardware Requirements for SAS Event Stream Processing

SAS Event Stream Processing can be installed as a stand-alone product. It can also coexist with SAS 9.4.

A single computer for the SAS Event Stream Processing components (ESP server, the web application server, and SAS Event Stream Processing Studio) is the minimum requirement. SAS Event Stream Processing can be deployed on a redundant computer for failover, or it can be distributed across multiple machines. On-premises deployments as well as cloud deployments are supported. You can also deploy the software on the compute layer of a Hadoop cluster, or even at the edge (on a gateway node) of a Hadoop cluster.

The following table describes a standard set of specifications for a computer where SAS Event Stream Processing is deployed:

Item	Recommended Level
CPU	4 cores (x86 architecture) Intel Xeon chip set with a minimum speed of 2.6 GHz
Memory	8 - 16 GB of RAM Memory clock speed of 1600 MHz
Disk Space and Speed	10 GB 10,000 RPM

An additional computer can be used as a thin client from which end users can access the user interface for SAS Event Stream Processing Studio. This machine requires minimal processing power and storage space and can run on Windows or UNIX.

Hardware Requirements for SAS Event Stream Manager

SAS recommends that you install SAS Event Stream Manager on a separate machine from SAS Event Stream Processing. The SAS Event Stream Manager Agent component is installed along with SAS Event Stream Processing. This component is recommended to enable all SAS Event Stream Manager functionality.

The following table describes a standard set of specifications for a machine where SAS Event Stream Manager is deployed:

Item	Recommended Level
CPU	2 cores (x86 architecture) Intel Xeon chip set with a minimum speed of 2.6 GHz
Memory	16 GB of RAM Memory clock speed of 1600 MHz
Disk Space and Speed	10 GB 10,000 RPM

Each machine that is used to access the user interface must have a minimum screen resolution setting of 1280 x 1024.

GPU Requirements

SAS Event Stream Processing supports an optional GPU environment for high-powered analytics calculations such as scoring with analytic store (ASTORE) files.

Here are the requirements for SAS Event Stream Processing in this environment:

- [ ] GPU with NVIDIA Pascal or Volta architecture
- [ ] 10 GB or more of disk space

You must perform several post-deployment steps to enable GPU functionality. For more information, see "(Optional) Enable GPU Functionality" on page 30.

Operating System Requirements

Supported Operating Systems

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

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

Requirements for All Linux Platforms

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

Libraries and Packages

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

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

- acl-2.2 or later

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

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

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

  glibc-2.22 and later (on SUSE Linux)

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

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

- libXp

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

- libXmu

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

Verifying systemd

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

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

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

```
yum update systemd
```

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

```
zypper update systemd
```

Additional Requirements for the ESP Server

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

```
-D_REENTRANT
```

```
-D_THREAD_SAFE
```

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

The SAS Event Stream Processing 5.**x** libraries for x86_64 chipsets 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.

The SAS Event Stream Processing 5.x libraries for 64-bit ARM chipsets have been built using gcc-6.2.0 on CentOS Linux 7.2.1603, using libc-2.17.so, libstdc++.so.6.0.22, and libgcc_s-6.2.0.so.1.

SAS Support for Alternative Operating Systems

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

Server Software Requirements

Java

A Java Runtime Environment (JRE) must be installed on every machine in your deployment. The playbook checks for a pre-installed version of Java that meets or exceeds the requirements. If one is found, it is used. Otherwise, the playbook attempts to install a recent version of OpenJDK and to set the path in a system

configuration file. You can also specify the path to an existing JRE in the vars.yml file before you run your playbook.

Java 1.8 is required for both SAS Event Stream Processing and SAS Event Stream Manager.

Apache httpd

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

SAS recommends that you install Apache httpd and configure the Apache HTTP Server to use certificates that comply with the security policies at your enterprise before you start the deployment process. The playbook will automatically configure the certificates to secure the server. For more information, see “Specify the Path to Certificates” on page 61.

The Apache HTTP Server must be dedicated to a single SAS Viya deployment.

Security Requirements

LDAP Requirements

Read access to your LDAP provider is required for SAS Event Stream Manager and SAS Event Stream Processing Studio.

SAS Viya requires a userDN and password in order to bind to the LDAP server. Anonymous binding is supported for clients that are authenticating to the LDAP server.

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

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

To configure LDAP to enable access to SAS Event Stream Manager and SAS Event Stream Processing Studio, follow the steps in “Configure LDAP Settings” on page 23 before you run the playbook.

User Accounts

The user account that you are using for the deployment must have super user (sudo) access. To verify that the user ID is included in the sudoers file, run the following command:

```
sudo -v
```

To verify your sudoers privileges, run the following command:

```
sudo -l
```

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

During the software deployment, one required user account (sas) and one group (also named sas) are created for you unless they already exist. Because the sas account is required for the SAS Event Stream Processing Studio component to run during normal product operation, you must not delete it or change its name. It does not run as root. If you must log on to this account, use sudo to access it.

The following table describes the predefined sas user account:

Account Name and Group	Parameters	Purpose
sas; member of sas group	Non-login service account without user restrictions. No password; can add password after installation if desired. Password does not expire. Default user name is required until the installation is complete. Any post-installation changes to this account do not prevent future software updates.	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 required components to run, including the web application server for SAS Event Stream Processing Studio.

Sudoers privileges are not required after the installation to run SAS Event Stream Processing. The installation directory path enables write access per user group, and it is owned by the sas user. To grant permission to edit the configuration files, the administrator must add any user requiring write access to these files to the sas group.

Client Requirements

Web Browsers

SAS Event Stream Processing Studio and Streamviewer include some advanced user interface features, which require a newer web browser. For information about supported browsers, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/34/support-for-web-browsers.html.

Screen Resolution

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

Deployment Tools

Ansible Controller Requirements

A typical Ansible deployment consists of at least one control machine (the Ansible controller) and multiple Ansible managed nodes (the machines where SAS software is installed). In a single-machine deployment, Ansible and all SAS software are installed on the Ansible controller. For more information, see “Install Ansible” on page 15.

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

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

3

Pre-installation Tasks

Prepare for a Rapid Deployment

The tasks that are described in this chapter are required for all deployments of SAS Event Stream Processing. If you are installing in a production environment or on multiple machines, follow the additional steps in “Installing a Production Deployment” on page 57.

Installing from a Mirror Repository

A mirror repository is required for all deployments on SUSE Linux. For Red Hat Enterprise Linux, a mirror repository is optional and should be used only if your machine target does not have access to the internet, or if you must always deploy the same version of software (such as for regulatory reasons).

Create a Mirror Repository

Standard Mirror Repository Creation

SAS Mirror Manager is a command-line utility for synchronizing a collection of SAS software repositories. Its primary use is to create and manage mirror repositories for software deployment.

Consider the requirements for your mirror repository:

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

To create a mirror repository with SAS Mirror Manager:

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

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

   By default, the repositories are placed in the `sas_repos` directory in the installation user's home directory. If you want to place them in another location, use the `--path` option followed by the full directory location of the mirror destination. This guide will refer to that location as **sas_repos**. However, if you choose to use a different location, replace instances of **sas_repos** in this guide with the actual location that you select. See the next sections for information about options.

   ```
   mirrormgr mirror --deployment-data path-to-deployment-zip-file-from-SOE
   --path location-of-mirror-repository --latest
   ```

   The **sas_repos** directories are broken down as follows:

   - The entitlements.json is a list of the repositories to which you are entitled.
   - The location_group_declarations.json file and the sasmd directory contain data that is used by the SAS Orchestration CLI to create the order-specific tools for your deployment.
   - Any remaining directories are the software repositories, organized by native deployment tools:
     - repos contains yum files for Linux.
     - win contains MSI files for Windows.
     - deb contains APT files for Debian.
     - bosh contains BOSH releases for BOSH.
5. (Optional) After the initial download is complete, move the file structure to a web server or shared NFS mount. The destination machine does not have to be connected to the internet.

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

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

Optionally, if you are using Red Hat Satellite, you can work with your system administrator to move the files to your Red Hat Satellite Server.

Mirror Manager Options

Specify a Distribution-Specific Subset of Files

To retrieve only the files for the Linux distribution that you are using:

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

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

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

Specify a Log Location

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

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

Enable Required Ports

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

Process	Required Port	Requires Allowed Inbound Traffic From	Notes
httpd	80, 443	anywhere (SAS Viya servers, workstation)	
SAS Event Stream Manager agent	2552	ESP servers only	Required only if your order included SAS Event Stream Manager.
SAS Infrastructure Data Server	5430–5439	SAS Viya Servers only	For a single server deployment with no failover, ports 5430-5432 must be opened in order to use the PostgreSQL tools, pgAdmin and pgpoolAdmin. Additional standby nodes each get the next available port number sequentially up to 5439.

Process	Required Port	Requires Allowed Inbound Traffic From	Notes
SAS Job Execution Launcher context	18501–18600	SAS Viya Servers only	Use a range of ports. The compute server gets the port range from the launcher during startup and attempts to use an open port in the range.
default SAS Messaging Broker AMQP client access port	5672	SAS Viya Servers only	
Vault	8200	SAS Viya Servers only	
SAS Configuration Server	8300, 8301, 8302, 8500, 8501	SAS Viya Servers only	SAS uses HashiCorp Consul as its configuration server. Ports 8301 and 8302 must be open to both UDP and TCP traffic.
default SAS Messaging Broker management web console port	15672	SAS Viya Servers only	

Any ports that will be used for ESP servers must be open to HTTP traffic. For more information, see Using the ESP Server.

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

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

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

The results contain two numbers:

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

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

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

Here is an example of the results:

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

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

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

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

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

Here is an example:

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

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

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

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

Configure SELinux

If you have enabled Security-Enhanced Linux (SELinux) in your environment, you must enable permissive mode on all of the target machines in your deployment. You can run the following command to check whether SELinux is enabled on an individual system:

```
sudo sestatus
```

For all Linux distributions, if a mode that is not permissive is returned, run the following commands:

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

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

Perform Linux Tuning

This section describes the minimal tuning that should be performed before you deploy your software in a test environment or on a single machine. A production deployment, or a deployment with additional machines, requires the additional steps that you can find in “Perform Additional Linux Tuning for a Production Deployment” on page 58.

Set the ulimit Values

Overview

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

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

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

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

```
*     -     nofile     value
```

The asterisk (*) indicates all user accounts.

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

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

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

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

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

Set the Maximum Number of Open File Descriptors and Stack Size

For each machine in your deployment:

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

2 Set the limit for open file descriptors as follows:

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

```
sas     -     nofile     150000
```

- For all other machines in the deployment, set the limit for the sas account, to at least 48000.

```
*     -     nofile     48000
```

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

```
*     -     nofile     150000
```

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

```
sas     -     stack     10240
```

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

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

Set the Maximum Number of Processes Available

For each machine in your deployment:

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

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

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

```
sas     -     nproc     100000
```

- For all other machines in the deployment, set the sas account to at least 65536.

```
*     -     nproc     65536
```

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

```
*     -     nproc     100000
```

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

Set the Semaphore Values

The following settings are required for the PostgreSQL database:

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

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

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

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

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

```
sudo sysctl -p
```

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

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

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

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

These commands allow the user to run 50000 tasks concurrently.

Install Ansible

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

Standard Ansible Installation

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

Streamlined Ansible Installation for Red Hat Enterprise Linux and Equivalent Distributions

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

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

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

```
## find out which release (6 or 7)
if   grep -q -i "release 6" /etc/redhat-release ; then
  majversion=6
elif grep -q -i "release 7" /etc/redhat-release ; then
  majversion=7
else
  echo "Apparently, running neither release 6.x nor 7.x "
fi
## Attach EPEL
```

```
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-$majversion.noarch.rpm
# Display the available repositories
sudo yum repolist
```

2 To Install Python PIP and related packages:

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

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

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

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

For Python 2.6 (and later within 2.6.x)::

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

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

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

5 To install a specific version of Ansible through PIP:

```
sudo pip install ansible==2.4.3
```

Streamlined Ansible Installation for SUSE Linux

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

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

1 To install Python's setup tools:

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

2 To Install Python PIP:

```
sudo easy_install pip
```

3 To install a specific version of Ansible through PIP:

```
sudo pip install ansible==2.4.3
```

Test Your Ansible Installation

1 To test the Ansible version:

```
ansible --version
```

Here is an example of successful output:

```
ansible 2.4.3.0
  config file =
  configured module search path = Default w/o overrides
  python version = 2.7.5 (default, May  3 2017, 07:55:04) [GCC 4.8.5 20150623 (Red Hat 4.8.5-14)]
```

2 To perform a basic ping test:

```
ansible localhost -m ping
```

Here is an example of successful output:

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

Create a Playbook

If you are installing on SUSE Linux, be sure to complete the steps in “Create a Mirror Repository” on page 10 before you create a playbook.

The SAS Orchestration Command Line Interface (CLI) uses the order information that was included in your Software Order Email (SOE) to create a playbook for deploying your SAS Viya software. Before you use the SAS Orchestration CLI, ensure that the SAS_Viya_deployment_data.zip file attachment from your SOE is copied to a directory on a machine that runs the Linux, Macintosh, or Windows operating system.

Download the SAS Orchestration CLI

The SAS Orchestration CLI can be run on Linux or Windows and it requires the Java Runtime Environment 1.8.*x*. It also requires access to the internet.

1. The SOE indicated that you should save the SAS_Viya_deployment_data.zip file attachment. If you have not already done so, save that file now.
2. Go to SAS Viya Install Center, and download the SAS Orchestration CLI for the operating system where you stored the ZIP file.

   The SOE recommended that you save the ZIP file to a machine that runs Linux, which is where you will install the SAS software that you purchased. However, you can also store it on a machine that runs Macintosh or Windows. If you use Internet Explorer to download the Linux or Macintosh version, save the file as a .tgz file instead of a .gz file.

   Note: This step requires internet connectivity.
3. Uncompress the .tgz file (Linux and Macintosh) or .zip file (Windows) in the same location where you downloaded it. The result is a file named sas-orchestration on Linux or Macintosh or a file named sas-orchestration.exe on Windows.

Create a Playbook with the SAS Orchestration CLI

Basic Command

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

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

Linux or Macintosh

```
./sas-orchestration build --input location-of-ZIP-file-including-file-name --platform
deployment-platform-tag
```

Windows

```
.\sas-orchestration.exe build --input location-of-ZIP-file-including-file-name --platform
deployment-platform-tag
```

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

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

Options

Use a Proxy Server

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

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

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

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

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

```
./sas-orchestration --java-option "-Dhttps.proxyHost=my.proxy.com --java-option "-Dhttps.proxyPort=1111"
build --input /tmp/SAS_Viya_deployment_data.zip
```

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

Use a Mirror Repository

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

```
./sas-orchestration build --input /sas/install/SAS_Viya_deployment_data.zip --platform redhat
--repository-warehouse "URL-to-mirror-repository-content"
```

Note: The repository warehouse URL must be available to all hosts that will participate in the deployment because the hosts are going to use that address to retrieve packages from the repositories. For example, if the repository warehouse is file-based, that location should be shared across hosts and should be shared at the same path on each of those hosts.

For more information about SAS Mirror Manager, see “Create a Mirror Repository” on page 10.

Help with the Options

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

Linux or Macintosh

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

Windows

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

Store the Playbook

1. If necessary, move the SAS_Viya_playbook.tgz file to a directory on your Ansible controller that can be read by other users. The recommended location is **/sas/install**.
2. In the same directory where you have saved the playbook, uncompress it.

```
tar xf SAS_Viya_playbook.tgz
```

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

Important: For a production deployment, be sure to follow the additional steps in "Preparing for a Production Deployment" on page 57.

4

Installing a Rapid Deployment

Overview

This chapter describes a rapid deployment of your SAS Event Stream Processing software. Follow the steps in this section to install the software on a single machine or in a test environment. If your order included SAS Event Stream Manager, install it on a separate machine from SAS Event Stream Processing. You will need to follow the steps in “Installing a Production Deployment” on page 57.

If you have not already done so, be sure to perform the steps in “Prepare for a Rapid Deployment” on page 9 before you begin the installation process.

Edit the Playbook

Modify the Initial Deployment

This chapter describes the initial deployment of your SAS Viya software only. For information about modifying an existing deployment with updated software or adding new software to an existing deployment, see “Managing Your Software” on page 39.

Edit the Inventory File

Overview

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

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

If you intend to install SAS Event Stream Processing on multiple machines, see "Edit the Inventory File for a Multi-Machine Deployment" on page 62 for information about editing the inventory file.

Each inventory file consists of two parts:

deployment target definition
: A specification of each machine on which SAS Event Stream Processing software will be deployed.

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

  Here is an example of a host group assignment list:

```
# The consul host group contains the Consul server.
[consul]
deployTarget
deployTarget2
```

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

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

Single-Machine Deployment

This section is applicable only if you are performing a single-machine deployment. If you are performing a multi-machine deployment, skip this section and go to "Edit the Inventory File for a Multi-Machine Deployment" on page 62.

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

```
cp samples/inventory_local.ini inventory.ini
```

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

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

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

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

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

For a local machine:

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

For a remote machine:

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

4 Save and close the inventory.ini file.

Configure LDAP Settings

The sitedefault.yml file (in the **/roles/consul/files** directory in the playbook) is used to configure authentication for SAS Event Stream Manager and SAS Event Stream Processing Studio. After you set a value with sitedefault.yml, you cannot re-run sitedefault.yml to change that value. You can re-run sitedefault.yml only to set properties that have not been set.

For more information about using the sitedefault.yml file, see Configuration Properties: Concepts in *SAS Viya Administration*.

Take these steps to enable the playbook to configure the LDAP server to enable authentication with SAS Logon Manager:

1 If you have not already copied and renamed the sitedefault.yml file, locate the sitedefault_sample.yml file on the Ansible controller machine. If you used the recommended location for uncompressing your playbook, the file is located at **/sas/install/sas_viya_playbook/roles/consul/files/sitedefault_sample.yml**. Make a copy of sitedefault_sample.yml in the same folder, and name the copy sitedefault.yml.

2 Use your preferred text editor to open sitedefault.yml.

3 Add values that are valid for your site, and save the file.

When you run your Ansible playbook using the site.yml option, the updated sitedefault.yml file is used automatically.

Install the Software

Deploy the Software

Assessment Test

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

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

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

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

Table 4.1 *Command Options Based on Password Requirements*

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

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

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

Note: For information about supported Ansible versions, see "Ansible Controller Requirements" on page 8.

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

Deployment Command

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

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

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

Add an option based on the password requirements for the user ID that performs the command, using Table 4.1 on page 24. To specify if you want to perform only an installation or configuration, see "Options" on page 24.

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

Options

To install, but not configure the software, use the same command that is described in "Deployment Command" on page 24, but replace `site.yml` with `install-only.yml`. Here is an example:

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

To configure software that has been installed only, use the full command that is described in "Deployment Command" on page 24.

Run from a Directory Other Than the Default

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

Successful Playbook Execution

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

```
PLAY RECAP *********************************************************************
```

```
deployTarget               : ok=81   changed=65   unreachable=0    failed=0
```

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

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

Retry a Failed Deployment

If your deployment fails, and you are able to respond to the error message and can recover from the error, you must restart the deployment using the appropriate deployment commands described in “Assessment Test” on page 23 and any appropriate options.

Failures can occur if there are port conflicts.

Deployment Logs

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

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

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

5

Post-installation Tasks

Complete SAS Event Stream Processing Setup

Take a few steps to complete the deployment. You must start the Metering Server, set some environment variables, and start the ESP server. You also have the option to generate and import certificates to support encryption for the ESP server.

Enable Metering for ESP Servers

The deployment process applies the product license on each machine where you have deployed SAS Event Stream Processing. However, additional steps are required in order to enable the license. You must set up and run at least one metering server to track the number of incoming events and to maintain event counts.

The metering server aggregates counts that are based on the license, the source window, and the hour of day. It stores aggregated results so that a client can query and track the total volume of messages that are processed. Enabling the metering server ensures that your ESP server is in compliance with the terms of its license. Event metering is not required on development servers because they do not contribute to the event volume that is assigned to a license.

For more information about enabling metering, see Using the Metering Server in the SAS Event Stream Processing user documentation.

Set Environment Variables for SAS Event Stream Processing

You must set some environment variables before you start SAS Event Stream Processing. For a shell that will only invoke SAS Event Stream Processing, run the following commands:

```
export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/5.2
```

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

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

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

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

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

Start SAS Event Stream Processing Studio

Additional steps are required to use SAS Event Stream Processing Studio, which provides a user interface for creating models. It is not automatically started during the installation.

1 SAS Event Stream Processing Studio requires Java 1.8. If Java 1.8 is not the default version of Java on your system, update the following script to set the SAS_JAVA_HOME environment variable:

```
/opt/sas/viya/config/etc/sysconfig/sas-javaesntl/sas-java
```

Here is an example:

```
SAS_JAVA_HOME=/usr/java/jdk1.8.0_101/jre
```

Or supply the location of the JDK, if applicable. For example:

```
SAS_JAVA_HOME=/usr/java/jdk1.8.0_101
```

Note: Do not include the **`/bin/java`** portion of the path for the definition of SAS_JAVA_HOME.

2 Verify that you have set the required environment variables. For more information, see “Set Environment Variables for SAS Event Stream Processing” on page 27.

3 SAS Event Stream Processing Studio should be running when the playbook completes. Check the status of the espvm process. Run the following command on Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-espvm-default status
```

Run the following command on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl status sas-viya-espvm-default
```

4 If the espvm service is reported to be down, run the following command on Red Hat Enterprise Linux 6.*x* to start it:

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

Run the following command on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl start sas-viya-espvm-default
```

5 After you have started the service, you can access SAS Event Stream Processing Studio using a web browser that is running on Windows or Linux. Open SAS Event Stream Processing Studio from a URL with the following format:

http://*esp-studio-hostname*:*port*/SASEventStreamProcessingStudio

Note: For *esp-studio-hostname* and *port*, specify values that are appropriate for your deployment. The default port is 80.

6 Before you can open or create a model in SAS Event Stream Processing Studio, you must start the ESP server. Change directories to the following location:

```
cd /opt/sas/viya/home/SASEventStreamProcessingEngine/5.2/bin
```

7 Run the following command:

```
dfesp_xml_server -pubsub n -http port  &
```

The `-pubsub` argument specifies a port for publish and subscribe actions. Replace *n* with the appropriate port number.

The `-http` argument specifies the port for the HTTP REST API. The value of *port* cannot exceed 65535.

The ampersand (&) enables additional commands to be entered in the same window that started the server.

Note: If you have a project that is predefined, use the `-model` *url* argument and supply the URL to the XML model. Specify the full path (**`file://path`**).

For more information about the ESP server, see SAS Event Stream Processing: Using the ESP Server.

8 (Optional) To check the status of SAS Event Stream Processing Studio, run the following command on Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-espvm-default status
```

Run the following command on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl status sas-viya-espvm-default
```

Encryption and Authentication Options

SAS Event Stream Processing provides optional encryption and authentication features. You can enable encryption on TCP/IP connections within an event stream processing engine. You can also configure ESP servers to require client authentication for SAS TCP/IP clients.

To enable encryption, the OpenSSL libraries must be installed on all computer systems that run the ESP server and clients. Version 1.0.2 or later of the Transport Layer Security (TLS) Protocol is required in order to take advantage of ECDH support for encryption ciphers used in encrypted connections.

Authentication and encryption apply to the following ESP server APIs:

- The ESP Server (XML Server) HTTPS API
  - Connections that are created by a client to communicate with an ESP server
  - Connections that are created by a file and socket connector or adapter that acts as a socket client or server
  - Connections that are created by the Streamviewer component (streamviewer.html) to communicate with the ESP server using the HTTPS protocol
- C, Java, or Python Publish/Subscribe API
  - Connections that are created by a client that uses the C, Java, or Python Publish/Subscribe API to communicate with an ESP server
  - Connections that are created by an adapter to communicate with an ESP server

Configuration of these security options has been greatly simplified in SAS Event Stream Processing 5.2. For more information about enabling security for an ESP server or for Streamviewer, see SAS Event Stream Processing: Security.

(Optional) Enable GPU Functionality

The SAS GPU Reservation service aids SAS processes in resource sharing and utilization of the GPUs that are available on a system. It is required on every machine where you want to take advantage of additional GPU functionality. A check is performed to detect supported GPUs in the environment. If a GPU is detected, the service is started automatically on that machine.

You can add a GPU to your deployment at a later time. It is not necessary to perform the SAS software deployment again in order to add GPU functionality. However, the initial software deployment must have been performed on the machine. Otherwise, some requirements would not be met.

To enable GPU functionality:

1. Using a user account that has sudoers privileges, log on to the machine where the GPU has been installed.
2. Verify that the SAS GPU Reservation service (sasgpud) has been installed in **`/etc/init.d`** by a previous deployment.
3. Launch the setup script to enable and start the GPU Reservation service:

   ```
   sudo /opt/sas/viya/home/bin/sasgpud_setup
   ```

   The script checks the system for supported devices, drivers, and libraries. If the system passes the check, the script starts the GPU Reservation service. If any requirements have not been met, you see an error message, and the service is not started.

Run this script whenever a GPU device is added or removed from the system.

Complete SAS Event Stream Manager Setup

If your order included SAS Event Stream Manager, take a few steps after the installation has completed to prepare the environment. Otherwise, you can skip this section.

Configure and Restart SAS Event Stream Manager Agent

SAS Event Stream Manager Agent is a small executable program that is installed along with SAS Event Stream Manager.

Agents relay operational metrics from ESP servers to SAS Event Stream Manager, and they perform actions on the ESP servers in response to commands that they receive from SAS Event Stream Manager.

Note: SAS Event Stream Manager Agent does not operate in an environment in which SSL has been enabled for connections to the ESP server. If encryption is being applied to these connections from other SAS Event Stream Processing components, you can add an SSL connection to the ESP server from SAS Event Stream Manager. You can then use SAS Event Stream Manager without the agent. For more information about adding an SSL connection to an ESP server, see “(Optional) Enable Encryption for SAS Event Stream Manager” on page 32.

To modify agent parameters:

1. Stop the agent. Run the following command on Red Hat Enterprise Linux 6.*x*:

   ```
   sudo service sas-viya-esmagent-default stop
   ```

   Run the following command on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl stop sas-viya-esmagent-default
```

Note: If encryption is enabled for connections to the ESP server, do not restart the agent. You can still use SAS Event Stream Manager without the agent. For more information, see SAS Event Stream Manager: Connecting Directly to an ESP Server.

2 Edit the start-up script to set the correct values for some environment variables. Use your preferred text editor to open the following file for editing: **`/opt/sas/viya/home/bin/sas-esmagent`**.

3 Locate the following environment variables within the start-up script. Set their values to environment-specific values, as specified in the following table:

Table 5.1 *SAS Even Stream Manager Agent Variables*

Variable	Environment-Specific Value
ESM_DISCOVERY_HOST	Host name of the Apache HTTP Server, the machine that you assigned to the [httpproxy] host group in the inventory file.
ESM_DISCOVERY_PORT	The port where SAS Event Stream Manager is listening for communications from the agent. This should correspond to the port that is open on the Apache HTTP Server. The default is Port 80.
ESM_AGENT_HOSTNAME	The host name of the machine where you have installed SAS Event Stream Manager Agent and ESP server. (These components must be installed on the same machine.)
ESM_PORT	The port where the agent listens. The default setting is Port 2552.
ESM_FRIENDLY_NAME	The name of the agent that appears in the user interface of SAS Event Stream Manager. The default setting is “ESM Agent.”

For more information about ESP server parameters, see SAS Event Stream Processing: Using the ESP Server.

4 Save your changes to the start-up script.

5 Start the agent. Run the following command on Red Hat Enterprise Linux 6.*x*:

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

Run the following command on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl start sas-viya-esmagent-default
```

The following message indicates success: `sas-viya-esmagent-default is running`.

Log On to SAS Event Stream Manager

SAS Event Stream Manager uses SAS Logon Manager for logon functionality. SAS Logon Manager uses LDAP for user authentication. A few steps are required to configure an LDAP server during the installation. For more information, see “Configure LDAP Settings” on page 23.

1 Open the following URL:

`http://`*`host:port`*`/SASEventStreamManager`

The host is the system on which SAS Event Stream Manager is installed. The port is the port number used by the system that hosts SAS Event Stream Manager. The default port is 80.

The Sign In to SAS window is displayed.

2 Enter your user ID and password, and click **Sign In**.

If you are a member of the SASAdministrators group, the Assumable Groups window is displayed. Group membership is not required.

Successful logon to the SAS Event Stream Manager user interface indicates that the software has been installed correctly. To validate that services have been installed and started successfully, see "Verify SAS Event Stream Manager Status" on page 37.

(Optional) Enable Encryption for SAS Event Stream Manager

Enabling the Transport Layer Security (TLS) protocol for connections between SAS Event Stream Manager and the ESP server is optional. The enablement of TLS involves these processes:

- TLS on the ESP server is enabled.
- The certificate authority (CA) file is obtained and imported to the web browser and to the Java keystore.

By default, the CA file is named `ca.pem`.

For complete details about configuring these security options, see SAS Event Stream Processing: Security.

To enable TLS:

1 Obtain the CA file for the system where SAS Event Stream Manager is installed and for the clients that access the user interface.

2 On the machines where users access SAS Event Stream Manager, import the client certificate to the certificates store of your preferred web browser.

3 On the machine where SAS Event Stream Manager is running, import the client certificate to the Java keystore:

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

```
$JAVA_HOME/jre/bin/keytool -importcert -keystore keystore-location -file path-to-file -storepass password -noprompt -alias alias
```

Here is an example that assumes that you use `ca.pem` as the CA file:

```
$JAVA_HOME/jre/bin/keytool -importcert -keystore /opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.jks -file ca.pem -storepass changeit -noprompt -alias myalias
```

4 Restart the SAS Event Stream Manager service. Run the appropriate command:

For Red Hat Enterprise Linux 6.7:

```
sudo service sas-viya-esm-service-default stop
sudo service sas-viya-esm-service-default start
```

For Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl stop sas-viya-esm-service-default
sudo systemctl start sas-viya-esm-service-default
```

Directory Structure and Permissions

After you install SAS Event Stream Processing, the files for the engine, the user interface components, and the authentication package are located in the following directory:

/opt/sas/viya/home/SASEventStreamProcessingEngine/

Configuration files for adapters and logs are located in the following directory:

/opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/

The basic directory path enables write access per user group, and it is owned by the sas user. To grant permission to users to edit the configuration files, the administrator must add them to the sas group.

6

Validating the Deployment

Verify the RPM Packages

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

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

Then you can run this basic command to verify an individual RPM package from the list that is returned:

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

The full name of each RPM is not required. For example, to verify the contents of the sas-espbase-5.2.20180109.06.x86_64 package, run the following command:

```
rpm -Vv sas-espbase
```

Note: Run the preceding commands for each host on which you have deployed SAS Event Stream Processing and its optional web application components.

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

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

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

```
rpm -Vv sas-espexam
......... /opt/sas/viya/home/lib/esp/sas-init-functions
```

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

```
rpm -Vv sas-espexam
package sas-espexam is not installed
```

The error indicators are shown in the following format:

```
SM5DLUGT c
```

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

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

The meaning of each error indicator is described as follows:

- S

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

- M

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

- 5

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

- D

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

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

- L

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

- U

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

- G

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

- T

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

- c

  Configuration file. This is useful for quickly identifying configuration files because they are likely to change and therefore are unlikely to verify successfully.

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

Verify SAS Event Stream Manager Status

To verify that a deployment of SAS Event Stream Manager has completed successfully, check that the required SAS services are available. You can check the status of all the SAS Event Stream Manager services by running the following the following commands on Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-esm-service-default status

sudo service sas-viya-esm-webui-default status
```

Run the following commands on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl status sas-viya-esm-service-default

sudo systemctl status sas-viya-esm-webui-default
```

Here is typical command output from Red Hat Enterprise Linux 6.7 to indicate that the software is running normally:

```
sas-viya-esm-service-default is running
sas-viya-esm-webui-default is running
```

The output is different on Linux 7.*x* or SUSE Linux, but it reports that each service is running.

Access Log Files

If you encounter difficulties during the deployment, log files that include information about installation and service status are written to the following directory:

`/opt/sas/viya/config/var/log/`

If the deployment fails, check the logs in this location first.

Verify SAS Message Broker

1. To verify that SAS Message Broker has been deployed correctly, go to the machine that you assigned to the [rabbitmq] host group.
2. Open a browser and go to the following address:
   - If HTTPS is enabled:

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

     Note: If you did not add compliant certificates and instead kept the default security settings and certificates, you will see the message `Your connection is not private`. SAS recommends that you replace the certificates before you give end users access to SAS Viya. For details, see HTTPS Access to SAS Message Broker.
   - If HTTP is enabled:

     `http://`*`RabbitMQ-IP-address`*`:15672/#/`

   If the RabbitMQ logon window appears, then SAS Message Broker is functioning as expected.

Verify SAS Infrastructure Data Server

Use these steps to verify that SAS Infrastructure Data Server has been deployed correctly.

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

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

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

- For SUSE Linux:

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

2 If SAS Infrastructure Data Server is running appropriately, you should receive a response like this:

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

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

7

Managing Your Software

Overview

SAS Event Stream Processing supports both upgrades and updates. The two procedures are distinct and separate.

What Is an Upgrade?

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

The installation process for previous releases of SAS Event Stream Processing has been changed significantly in SAS Event Stream Processing 5.2. To upgrade a previous version of the software to version 5.2, start by consulting the "System Requirements" on page 3 section of this guide. Verify system requirements, perform pre-installation tasks, and install the software.

SAS recommends that you create a backup of the deployed software environment before you perform an upgrade.

What Is an Update?

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

You might determine that your software requires an update, or you might be notified by SAS that updates are available.

Upgrading Models and Data

Upgrading SAS Event Stream Processing from version 4.3 or 5.1 to version 5.2 is supported. For earlier versions, uninstalling the older version of the software is required.

Support for SAS Micro Analytic Service (MAS) modules and stores has moved from the Procedural window to the Calculate window in SAS Event Stream Processing 5.2. Migrating from a Procedural window to a Calculate window requires minimal changes to your XML code. For more information, see Migrating from a Procedural Window to a Calculate Window.

Migrating models and data that you generated from a previous release of SAS Event Stream Processing is supported on a limited basis. You can migrate your XML code from SAS Event Stream Processing 3.2 or later to the current release by running the dfesp_xml_migrate script. For more information, see Migrating XML Code across Product Releases. You can also import files from SAS Event Stream Processing 3.2, 4.x, or 5.1. However, if you plan to import files that you created with SAS Event Stream Processing 3.2, be aware of the following issues:

- Multiple XML elements in SAS Event Stream Processing 5.*x* have changed since 3.2. You must replace the elements that differ. Opening a legacy project in SAS Event Stream Processing Studio does not automatically upgrade your XML code to a valid format.
- Review your C++ code that was used with SAS Event Stream Processing 3.2. You must replace the registerMethod_ds2 function with the registerMethod_DS2TS function.
- The default date format of %Y-%m-%d %H:%M:%S for CSV timestamp and datetime fields is no longer valid. The new ESP_DATETIME fields contain a 64-bit integer that represents seconds since UNIX epoch. The new ESP_TIMESTAMP fields contain a 64-bit integer that represents microseconds since UNIX epoch.
- In addition, you can no longer specify an alternative date format when initializing a SAS Event Stream Processing engine. To pass CSV events using an alternative date format, that format must now be specified on the connector or adapter that is the source or sink of CSV data. All connectors and adapters that support CSV include an optional DateFormat parameter for this purpose.

To upgrade models that you created in SAS Event Stream Processing 4.*x* to the current version, take the following steps:

1. In SAS Event Stream Processing Studio 4.*x*, export the 4.*x* models that you want to use in the newer version of SAS Event Stream Processing.
2. Install SAS Event Stream Processing.
3. Use SAS Event Stream Processing Studio to import the 4.*x* models that you previously exported. For more information, see *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*.

As noted previously, you can import models that you created in SAS Event Stream Processing Studio 3.2 by running the dfesp_xml_migrate script to migrate your XML code to the 5.*x* XML schema.

Upgrading Your Software

Overview

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

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

Upgrading SAS software requires an outage period because some services are stopped and restarted automatically during the upgrade process.

Note: Upgrading a deployment on SUSE Linux is not supported in this release because support for SUSE Linux is new in SAS Event Stream Processing 5.2.

Prepare to Upgrade

To prepare to upgrade your deployment:

Note: System requirements for RAM, CPU, and disk space are likely to change with each release. Verify that your environment meets the requirements that are listed in "System Requirements" on page 3.

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

   On each machine in your deployment, create a file that lists the names and versions of all the RPM packages that are installed. Create this file in the directory on each machine where you stored deployment and maintenance files. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

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

   On each machine in your deployment, create a file that lists the SAS yum groups that are installed on a machine. Create this file in the directory on each machine where you stored deployment and maintenance files. For example, you can run the following command to create a text file that lists all the SAS yum groups:

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

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

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

2 When performing an upgrade, you receive a new Software Order Email (SOE) from SAS. Use your SOE to download the SAS Orchestration CLI.

3 Using the SAS Orchestration CLI that you downloaded, create a playbook using the instructions on the SAS Orchestration Command Line Interface (CLI) download site. For more information, see "Create a Playbook" on page 17.

4 If you have a playbook from a previous installation of SAS Event Stream Manager, extract the new playbook to a location that is different from that of your original playbook. For example, if you extracted your original playbook to **/sas/install/**, you might extract the new playbook to **/sas/upgrade/** instead. You must extract the new playbook to a location that is different from the one that you used for your deployment for these reasons:

- To preserve the original vars.yml file and the inventory file.
- To ensure that the playbook directory correctly reflects what is delivered. If a new playbook is mistakenly extracted over an existing playbook, files that were removed in the newer playbook would still be available and could negatively affect the process for researching and resolving deployment issues.

To extract the new playbook, use a command that is similar to the following:

```
tar xf SAS_Viya_playbook.tgz -C /sas/upgrade/
```

5 If applicable, merge the vars.yml file and the inventory file from the previous deployment into the new playbook. If the previous inventory file contains any spaces that are used to indent machine names, do not include the extra spaces.

Note: For upgrades of SAS Event Stream Processing that did not include SAS Event Stream Manager, you will not have these files from a previous deployment. You will need to edit these files to specify the installation targets and other deployment parameters. For more information about these files, see "Installing a Rapid Deployment" on page 21.

a Compare the two vars.yml files, and compare the two inventory files to check for additions or changes in the newer set of files.

```
diff /sas/install/sas_viya_playbook/vars.yml /sas/upgrade/sas_viya_playbook/vars.yml
diff /sas/install/sas_viya_playbook/inventory-file /sas/upgrade/sas_viya_playbook/inventory.ini
```

b If the new files contain new content, merge your customized edits from the two original files into the two new files. If a key/value pair in the original file is not included in the new file, you do not need to add the key/value pair to the new file. If you have any questions, contact SAS Technical Support.

c If you have questions about whether to add a key/value pair from an original file to the new file, contact SAS Technical Support.

Back up Your Software and Stop Services

A few additional steps are required in the upgrade process.

1 Create a backup copy of the SAS Event Stream Processing Studio database in order to preserve project files. Follow these steps:

a Stop the SAS Event Stream Processing Studio (espvm) service by running the following command on Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-espvm-default stop
```

Run the following command on Red Hat Enterprise Linux 7.*x*:

```
sudo systemctl stop sas-viya-espvm-default
```

b Create a backup copy of the database, which is a single binary file (studio.mv.db). You can copy it to any directory location outside the SAS Event Stream Processing installation directory structure.

The location and filename of the database are determined by the environment variable ESP_STUDIO_DB. By default, it is stored in **/opt/sas/viya/config/data/espvm/**.

To create the backup, run the following command:

```
cp studio.mv.db directory-name
```

2 Stop the ESP server. Change directories to the following location:

```
cd /opt/sas/viya/home/SASEventStreamProcessingEngine/5.2/bin
```

3 Run the following command:

```
dfesp_xml_client -url "http://host:port/SASESP/server/state?value=stopped" -put
```

Replace *host-name* with the host name of the machine where the ESP server is running.

Replace *http-port* with the port number that you provided when you started the ESP server.

4 (Optional) If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where Streamviewer is running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

For more information, see Starting Streamviewer.

5 Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

6 If you installed SAS Event Stream Manager, run the following command to delete the schema:

```
sudo /opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
kv delete 'config/esm-service/spring/datasource/schema'
```

Upgrade SAS Software

To upgrade your deployment:

1 Log on to the PostgreSQL machine in your deployment.

2 Run the following command:

```
sudo cat -n /opt/sas/viya/config/etc/sasdatasvrc/postgres/pgpool0/pool.cdf
```

All entries in the command's output should display `healthy`.

3 Run the following command:

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

4 Open vars.yml and locate the `INVOCATION_VARIABLES` section.

5 Compare the `NODE_TYPE:` of each node in the Postgres cluster to the output of the `sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status` command.

- `P` - Primary
- `S` - Secondary

If `NODE_TYPE:` for each node in vars.yml does not match the output of the `sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status` command, you must edit vars.yml.

6 Compare the hostnames in the output of the `sudo /etc/init.d/sas-viya-sasdatasvrc-postgres status` command with the hostname assignments in inventory.ini. If the hostnames do not match, you must edit inventory.ini.

7 Compare the deploy target assignments for each node in inventory.ini to the deploy target assignments for each node in the `INVOCATION_VARIABLES` section of vars.yml. If the deploy target assignments do not match, edit vars.yml to match inventory.ini.

8 Install your SAS software using the steps in the installation chapter.

9 After the software has been installed, complete the following tasks, as appropriate:

   a "Complete SAS Event Stream Processing Setup" on page 27.

   b If your upgrade includes SAS Event Stream Manager, "Complete SAS Event Stream Manager Setup" on page 30.

   c Validate the Deployment on page 35.

   d Steps that are contained in the appendix, if they are relevant to your deployment.

10 (Optional) After the upgrade process has completed, record the new list of installed software.

   On each machine in your deployment, create a file that lists the names and versions of all the RPM packages of the SAS software that are installed. Create this file in the directory on each computer where you stored deployment and maintenance files. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

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

   On each machine in your deployment, create a file that lists the SAS yum groups that are installed on a machine. Create this file in the directory on each machine where you stored deployment and maintenance files. For example, you can run the following command to create a text file that lists all the SAS yum groups:

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

   You can see the differences between the previous and current deployments by comparing the lists of installed software before the upgrade and after the upgrade.

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

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

Updating Your Software

Overview

An update replaces some or all of your deployed software with the latest versions of that software. You perform the update with the same command that was used to install SAS Event Stream Processing, and use the same software order and the same playbook.

- If you used an Ansible playbook for your initial installation, you should update with Ansible.
- If you mirrored your software, update the mirror. Run the following command:

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

Updating the software requires an outage period because some services are stopped and restarted automatically during the update process.

Note: The update process preserves any user-modified configuration values in the vars.yml file, but changes made to other files in the deployment might be lost.

You will need the location of the directory on each machine where you stored deployment and maintenance files.

If you are using a PDF version of this guide, go to the Deployment Guides web page at https://support.sas.com/en/documentation/install-center/viya/deployment-guides.html and verify that you have

the latest version of the deployment documentation before you start the update process. The release date of each document is located in the bottom right corner of the front page.

User Requirements for Performing the Update

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

List the Packages That Are Available for Update

Deployments without a Mirror Repository

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

on Red Hat Enterprise Linux:

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

on SUSE Linux:

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

Deployments with a Mirror Repository

Important: How you list packages for deployments with a mirror repository depends on whether you have internet access.

With Internet Access

To list packages in a mirror repository in a deployment with internet access:

1. List the packages that are available for the update process by running the following command on the machine where the mirror repository is located:

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

2. Before performing the update, you must synchronize the mirror repository with SAS. To synchronize, run the following command on the machine where the mirror repository is located:

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

Without Internet Access

To list packages in a mirror repository in a deployment without internet access:

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

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

2. Before performing an update, you must synchronize the mirror repository with SAS. To synchronize, run the following command on the machine where the connected mirror repository is located:

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

3 Move the files from the machine where the connected mirror repository is located to the machine where the unconnected mirror repository is located.

Update with Yum

You can only use yum to update your software if your deployment is on Red Hat Enterprise Linux or an equivalent distribution. To update your deployment using yum, repeat these steps for each machine in the deployment:

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

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

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

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

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

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

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

2 Stop all the SAS services on the machine:

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

3 Stop the SAS Event Stream Processing Studio (espvm) service.

Run the following command on Red Hat Enterprise Linux 6.x:

```
sudo service sas-viya-espvm-default stop
```

Run the following command on Red Hat Enterprise Linux 7.x:

```
sudo systemctl stop sas-viya-espvm-default
```

4 (Optional) If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where Streamviewer is installed and running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

5 Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

6 To update all SAS software on the machine:

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

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

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

8 Restart the services that are installed on the machine. To restart all the SAS services on the machine:

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

9 (Optional) After the update process has completed, record the new list of installed software. For example, you can use the following command to create a text file that lists all the SAS RPM packages:

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

You can see the differences between the previous and current deployments by comparing the lists of installed software before the update and after the update.

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

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

10 Configuration files for adapters and logs are located in the following directory: **/opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/**

The update process creates backup copies of your configuration files, with different filenames. For example, .sav is appended to the file extension. To preserve configuration changes, rename the configuration files when the update process has completed.

Update with Zypper

You can only use zypper to update your software if your deployment is on SUSE Linux or an equivalent distribution. To update your deployment using zypper, repeat these steps for each machine in the deployment:

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

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

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

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

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

2 Stop the SAS Event Stream Processing Studio (espvm) service.

```
sudo /etc/init.d/sas-viya-espvm-default stop
```

3 (Optional) If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where Streamviewer is installed and running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

4 Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

5 To update all SAS software on the machine:

```
sudo zypper update "sas-*"
```

6 At the prompt `Continue? [y/n]`, review the available updates and then enter `y`.

7 To restart all the SAS services on the machine:

```
sudo /etc/init.d/sas-viya-all-services start
```

8 (Optional) After the update process has completed, record the new list of installed software.

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

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

You can see the differences between the previous and current deployments by comparing the lists of installed software before the update and after the update.

Update with Ansible

To update your deployment using Ansible:

1 (Optional) Record the existing list of installed software before you begin. For example, you can run the following command to create a text file that lists all the SAS RPM packages:

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

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

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

2 Review the *_deployment.* files in the existing deployment for any user-modified changes. If there are any user-modified changes to the *_deployment.* files, back up the file and update the vars.yml file with the changes before you perform the update. If you have questions, contact SAS Technical Support.

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

3 Stop the SAS Event Stream Processing Studio (espvm) service by running the following command

on Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-espvm-default stop
```

on Red Hat Enterprise Linux 7.*x* or SUSE Linux:

```
sudo systemctl stop sas-viya-espvm-default
```

4 (Optional) If you installed Streamviewer, stop the Streamviewer process:

```
$DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
```

Replace *host-name* with the host name of the machine where Streamviewer is running.

Replace *http-port* with the port number that you provided when you started Streamviewer with the start-up script.

For more information, see Starting Streamviewer.

5 Stop the Metering Server:

```
dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
```

Replace *host-name* with the host name of the machine where the Metering Server is running.

Replace *http-port* with the port number for the Metering Server. By default, it uses port 31001.

6 To initiate the update, run the same command and options that you ran when you performed the initial deployment. For more information, see “Deploy the Software” on page 23.

7 (Optional) After the update process has completed, record the new list of installed software. For example, you can use the following command to create a text file that lists all the SAS RPM packages:

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

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

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

You can see the differences between the previous and current deployments by comparing the lists of installed software before the update and after the update.

8 Configuration files for adapters and logs are located in the following directory: **/opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/**

The update process creates backup copies of your configuration files, with different filenames. For example, .sav is appended to the file extension. To preserve configuration changes, rename the configuration files when the update process has completed.

8

Completing the Deployment

Save Snapshot Directory Content

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

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

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

View Code Examples for SAS Event Stream Processing

Code examples to help you write programs are installed along with the software. You can find the examples in the following directory after the deployment has completed:

`/opt/sas/viya/home/SASEventStreamProcessingEngine/5.2/examples/`

The examples directory includes files for C++, XML, Python, and Java. It also includes a readme_examples.txt file, which briefly describes each example and its usage.

SAS recommends that you copy the examples that you require to a writable directory on the local computer so that you can run them.

Two documents are helpful in understanding the examples. You can find links on the SAS Event Stream Processing product page to the following user guides:

- *DataFlux Expression Language Reference Guide*
- *SAS Micro Analytic Service Programming and Administration Guide*

Review Example Templates for SAS Event Stream Manager

Example files are provided to help you learn to use SAS Event Stream Manager. You can find the example job templates in the SAS Event Stream Manager examples package, which you can download from the SAS Support Knowledge Base.

The package includes the resources that are required to create a deployment and deploy a job. A full set of instructions for using example job templates is included in the *SAS Event Stream Manager: User's Guide*, which is available on the SAS Event Stream Manager product page.

Review Product Documentation

After you install, configure, and verify the deployment, you are ready to begin writing applications that capture and analyze streaming event data in real time.

The next step is to consult the product documentation. The product 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 (SAS Event Stream Processing Studio and Streamviewer).

SAS recommends starting with *SAS Event Stream Processing: Overview*, which provides an introduction to product features and explains how to proceed with creating event stream processing models and incorporating them into applications.

If you have set up the optional Streamviewer component, you can find more information about it in a separate guide. For a full set of instructions about using Streamviewer, see Visualizing Event Streams with Streamviewer.

If you also purchased SAS Event Stream Manager, you are ready to begin using it to manage SAS Event Stream Processing applications. The *SAS Event Stream Manager User's Guide* explains how to work with the projects that you create in SAS Event Stream Processing Studio and how to manage SAS Event Stream Processing deployments. You can find this guide on the SAS Event Stream Manager product page.

9

Uninstalling SAS Event Stream Processing

What deploy-cleanup Does

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

1 Stop all SAS services.

2 Remove all SAS RPMs.

3 Delete any remaining SAS .pid files.

4 Delete the entitlement_certificate.pem and SAS_CA_Certificate.pem files.

The deploy-cleanup command renames the **/opt/sas/viya** directory to **/opt/sas/viya_*epoch***. Also, the **/opt/sas/spre** directory is renamed as **/opt/sas/spre_*epoch***.

The uninstallation does not remove the customized script that you received with your SOE, and it does not remove any users that have been set up.

Create a Backup for SAS Event Stream Processing

Before you run deploy-cleanup.yml to uninstall SAS Event Stream Processing, create a backup copy of the SAS Event Stream Processing Studio database in order to preserve project files. Follow these steps:

1 Stop the SAS Event Stream Processing Studio (espvm) service by running the following command on Red Hat Enterprise Linux 6.*x*:

```
sudo service sas-viya-espvm-default stop
```

Run the following command on Red Hat Enterprise Linux 7.*x* and SUSE Linux:

```
sudo systemctl stop sas-viya-espvm-default
```

2 Create a backup copy of the database, which is a single binary file (studio.mv.db). You can copy it to any directory location outside the SAS Event Stream Processing installation directory structure.

   The location and filename of the database are determined by the environment variable ESP_STUDIO_DB. By default, it is stored in **/opt/sas/viya/config/data/espvm/**.

   To create the backup, run the following command:

   ```
   cp studio.mv.db directory-name
   ```

3 If you installed Streamviewer, stop the Streamviewer process:

   ```
   $DFESP_HOME/bin/dfesp_xml_client -url "http://hostname:http_port/exit"
   ```

   Replace *hostname* with the host name of the server where the Streamviewer files are installed and running.

   Replace *http_port* with the port number that you provided when you started Streamviewer with the startup script.

   For more information, see Starting Streamviewer.

Uninstall from a Single Machine

To uninstall your 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 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).

Use deploy-cleanup

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

2 Here is the basic syntax for the command to run the playbook and deploy the software:

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

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

Add an option based on the password requirements for the user ID that performs the command, using Table 9.1 on page 55.

Table 9.1 *Command Options Based on Password Requirements*

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

Here is an example of the deploy command that requires both sudo and SSH passwords:

```
ansible-playbook deploy-cleanup.yml --ask-pass --ask-become-pass
```

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

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

   ```
   yum remove httpd
   ```

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

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

      ```
      ps -A
      ```

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

   c. Remove the epmd process by running the following command:

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

3. The sas-configuration-cli process could remain running on your system. To stop the process, perform the following steps on every machine in your deployment:

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

      ```
      ps -A
      ```

   b. In the results, find “sas-configuration-cli” in the far right column, and then locate its process ID (PID) in the far left column. If “sas-configuration-cli” is not listed, then you can move on to the next machine.

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

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

Uninstall a Mirror Repository

If your deployment includes a mirror repository and you want to remove it as well, you can run a basic Linux command to do so. Because all the files of the mirror repository are contained in a single directory, use the following command to remove the mirror repository:

```
sudo rm -rf path-to-mirror-repository
```

If you did not change the default location of the SAS Mirror Manager log when you deployed your software, you should also remove the log from **/.local/share/mirrormgr** in the home directory of the install user.

Appendix 1

Installing a Production Deployment

Preparing for a Production Deployment

This guide has provided you with the steps that are required to install SAS Event Stream Processing on a single computer. The rapid installation procedures for a single-machine deployment or for a simple test environment did not include some required steps for a multi-machine or full-featured deployment. This section of the document provides pre-installation steps for a production environment.

The steps that are described in this section should be completed in addition to the required steps that are described in “Pre-installation Tasks” on page 9.

Configure a Proxy Server

Overview

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

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

Using curl

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

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

In addition, ensure that HTTP requests between machines in the deployment are not routed through the proxy server during deployment by adding the IP addresses, host names, or domains for the SAS Viya machines to

the no_proxy variable in your profile.d script. For example, if the SAS Viya machines are using the IP addresses, 10.255.47.131 and 10.255.47.132, and the host names, machine1.example.com and machine2.example.com, you can configure no_proxy as follows:

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

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

Using yum

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

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

Enable the Yum Cache

Note: SUSE Linux does not use yum as a deployment tool. If you are using SUSE Linux, you should skip this section.

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

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

```
keepcache = 1
```

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

Perform Additional Linux Tuning for a Production Deployment

This section describes tuning that should be performed on your Linux machines before you deploy your software. These steps are not required for a rapid deployment on a single machine in a test environment. However, SAS strongly recommends that you follow these steps in your production environment, or in a deployment that includes SAS Event Stream Manager.

For a production deployment, perform these steps in addition to those that are described in “Pre-installation Tasks” on page 9.

Set the ulimit Values

Overview

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

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

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

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

```
*     -     nofile     value
```

The asterisk (*) indicates all user accounts.

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

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

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

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

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

Set the Maximum Number of Open File Descriptors and Stack Size

For each machine in your deployment:

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

     ```
     sas     -     nofile     150000
     ```

   - For all other machines in the deployment, set the limit for the sas account, to at least 48000.

     ```
     *     -     nofile     48000
     ```

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

   ```
   *     -     nofile     150000
   ```

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

   ```
   sas     -     stack     10240
   ```

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

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

Set the Maximum Number of Processes Available

For each machine in your deployment:

1. Open the appropriate file. For Red Hat Enterprise Linux 6.7 or an equivalent distribution, open **/etc/security/limits.d/90-nproc.conf**. For Red Hat Enterprise Linux 7.1 and greater or an equivalent distribution, open **/etc/security/limits.d/20-nproc.conf**. For SUSE Linux, open**/etc/security/limits.conf**.
2. Set the limit for the number of processes as follows:
   - If PostgreSQL will be deployed on the machine, set the limit (using the nproc item) to 100000 for the sas user.

     ```
     sas     -     nproc     100000
     ```

   - For all other machines in the deployment, set the sas account to at least 65536.

     ```
     *     -     nproc     65536
     ```

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

```
*     -     nproc     100000
```

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

Set the Semaphore Values

For each machine on which PostgreSQL will be deployed.

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

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

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

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

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

```
sudo sysctl -p
```

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

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

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

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

These commands allow the user to run 50000 tasks concurrently.

Optional Configuration for Enhanced Security

The steps that are required to perform a rapid deployment did not include optional configuration to secure your environment. In a production deployment, SAS recommends taking these additional steps.

Transport Layer Security

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

The deployment process provides a default level of data encryption. However, you should perform several additional actions to increase the level of security on your systems.

How Default Security Is Applied

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

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

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

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

Enhance Default Security Settings

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

1. Install the Apache httpd module and the Apache mod_ssl module on all the web servers in your environment.
2. Add certificates that conform to the policies at your enterprise.
3. Specify the location of the intermediate certificates and the root CA when you edit the playbook. For more information, see “Specify the Path to Certificates” on page 61.

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

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

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

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

Specify the Path to Certificates

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

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

To set HTTPD_CERT_PATH:

1. Open the vars.yml file. If you used the recommended location for uncompressing your playbook, the file is located at `/sas/install/sas_viya_playbook/vars.yml`.

   For more information about the options that you can set in this file, see “Modify the vars.yml File” on page 65.

2 Set the value of HTTPD_CERT_PATH based on the following conditions. Ensure that any value you use is enclosed in single quotation marks (').

- If your SSLCertificateChainFile contains the root certificate authority (CA) and all intermediate certificates, remove the existing value for HTTPD_CERT_PATH. Ensure that all browsers and clients have the root CA in their truststore.

  Here is an example of the modified variable:

  ```
  HTTPD_CERT_PATH:
  ```

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

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

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

  Note: The default value for `HTTPD_CERT_PATH` in the vars.yml file is the most likely location for the necessary file. If that file is in the default location, no changes are required.

3 Save and close the vars.yml file.

Optional Steps to Install on Multiple Machines

You can use the same Ansible playbook to install SAS Event Stream Processing on multiple machines. Use the steps in this section to set up your inventory file to install SAS Event Stream Processing components on separate machines.

SAS recommends installing SAS Event Stream Manager on a separate machine. Follow the steps in this section if you ordered SAS Event Stream Manager.

Edit the Inventory File for a Multi-Machine Deployment

Specify the Machines in the Deployment

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

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

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

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

Table A1.1 Descriptions of Components of the Deployment Target Reference

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

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

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

```
deployTarget ansible_connection=local
```

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

```
sas-esp-host ansible_host=host1.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
sas-service ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
sas-esp-studio-host ansible_host=host4.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
sas-esm-host ansible_host=host5.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa
```

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

```
sas-service ansible_host=host2.example.com ansible_user=user1 ansible_ssh_private_key_file=
~/.ssh/id_rsa consul_bind_adapter=eth0
```

Assign the Target Machines to Host Groups

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

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

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

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

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

```
[CommandLine]
sas-service-host

[CoreServices]
sas-service-host

[consul]
sas-service-host

[espServer]
sas-esp-host

[espStreamviewer]
sas-esp-studio-host

[espStudio]
sas-esp-studio-host

[httpproxy]
sas-service-host

[pgpoolc]
sas-service-host

[rabbitmq]
sas-service-host

[sasdatasvrc]
sas-service-host

[viprESM]
sas-esm-host

[sas-all:children]
CommandLine
CoreServices
consul
espServer
espStreamviewer
espStudio
httpproxy
pgpoolc
rabbitmq
sasdatasvrc
viprESM
```

Consider the following issues when editing the inventory file:

- SAS recommends that you do not remove any host groups from the list or any entries from the [sas-all:children] list unless you are an experienced Ansible user. A host group can have no entries under it, but the host group should not be removed, even if it is empty. Removing a host group that contains targeted machines from the [sas-all:children] list can result in critical tasks not being executed on those targeted machines.

- If you purchased SAS Event Stream Manager, the machine where you intend to install it must be specified in both the [viprESM] and [consul] host groups. Do not install SAS Event Stream Manager and SAS Event Stream Processing on the same machine.
- If the machines that you specify for [pgpoolc] or [sasdatasvrc] do not have an alias of deployTarget in the deployment target reference, you must open the **sas_viya_playbook/vars.yml** file and replace the instance of deployTarget under INVOCATION VARIABLES with the alias that you used in the deployment target reference:

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

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

Now take the steps that are described in "Modify the vars.yml File" on page 65 to continue your multi-machine deployment.

Modify the vars.yml File

As its name suggests, the vars.yml file contains deployment variables that enable you to customize your deployment to meet your needs. Note that all entries in the vars.yml file are case-sensitive.

Set the Deployment Label

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

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

```
DEPLOYMENT_LABEL: "va-04april2017"
```

Set the Pre-deployment Validation Parameters

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

Check the Ansible Version

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

Note: For information about supported Ansible versions, see "Ansible Controller Requirements" on page 8.

Verify Machine Properties

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

1. Verify that the DEPLOYMENT_LABEL variable has content and contains only lowercase alphabetic characters, numbers, and hyphens.
2. Verify that each computer's fully qualified domain name contains fewer than or equal to 64 characters.
3. Verify that each computer in the inventory file can successfully connect to every other machine in the inventory file.

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

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

Create and Verify sas User and sas Group

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

Specify Security Settings

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

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

By default, SECURE_CONSUL is set to **`true`** and DISABLE_CONSUL_HTTP_PORT is set to **`true`**. Only the HTTPS port will be available after the software is deployed.

Change the Repository Warehouse

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

```
REPOSITORY_WAREHOUSE: "location-of-new-mirror-repository"
```

(Optional) Specify JRE

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

1 With a text editor, open the vars.yml file.

2 Set the value of sas_install_java to **`false`**. For example:

```
sas_install_java: false
```

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

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

4 Save and close the vars.yml file.