Contents

Chapter 1 / Introduction ................................................................. 1
   Steps for a Successful Deployment ............................................. 1
   Contact SAS Technical Support .................................................. 3

Chapter 2 / System Requirements .................................................. 5
   Hardware Requirements .......................................................... 5
   Operating System Requirements ............................................... 9
   Server Software Requirements ............................................... 11
   Security Requirements .......................................................... 12
   Client Requirements ............................................................. 13
   Deployment Tools ................................................................... 14

Chapter 3 / Pre-installation Tasks ................................................... 15
   Prepare for a Rapid Deployment ................................................. 15
   Installing from a Mirror Repository ........................................... 16
   Enable Required Ports ............................................................. 19
   Set Environment Variables ....................................................... 21
   Configure SELinux .................................................................... 22
   Perform Linux Tuning ............................................................... 23
   Install Ansible ......................................................................... 26
   Create a Playbook ..................................................................... 28

Chapter 4 / Installing a Rapid Deployment ........................................ 33
   Edit the Playbook ....................................................................... 33
   Install the Software ................................................................... 35
   Complete the Deployment .......................................................... 37

Chapter 5 / Post-installation Tasks .................................................... 39
   Complete SAS Event Stream Processing Setup .............................. 39
   Complete SAS Event Stream Manager Setup ................................. 46
   (Optional) Enable GPU Functionality ....................................... 50
   (Optional) Set Up SAS Event Stream Processing Streamviewer as a
      Stand-Alone Client Application ............................................. 50
   (Optional) Set Up SAS Event Stream Processing Studio as a
      Stand-Alone Client Application ............................................. 53
   Directory Structure and Permissions ......................................... 55

Chapter 6 / Deploying with Containers .............................................. 57
   Installation Overview ............................................................... 57
   Prepare the Kubernetes Environment ......................................... 57
   Obtain the Required Files .......................................................... 58
   Copy the Certificates ................................................................ 58
   Create and Populate a Mirror Registry ...................................... 59
   Deploy the Software to the Cluster ............................................. 61

Chapter 7 / Validating the Deployment ............................................... 63
   Verify the RPM Packages ........................................................ 63
   Verify SAS Event Stream Manager Status .................................. 65
   Access Log Files ....................................................................... 65
Verify SAS Message Broker ........................................... 66
Verify SAS Infrastructure Data Server ......................... 66

Chapter 8 / Managing Your Software ................................. 69
  Overview ................................................................ 69
  Upgrading Models and Data .................................. 70
  Upgrading Your Software ..................................... 71
  Exporting SAS Event Stream Processing Streamviewer Data .................................................. 77
  Importing SAS Event Stream Processing Streamviewer Data .................................................. 78
  Updating Your Software ....................................... 79

Chapter 9 / Completing the Deployment ......................... 87
  Save Snapshot Directory Content ............................ 87
  Review Code Examples for SAS Event Stream Processing .................................................. 87
  Review Product Documentation .............................. 88

Chapter 10 / Uninstalling SAS Event Stream Processing .... 89
  What deploy-cleanup Does ................................... 89
  Use deploy-cleanup .............................................. 89
  Uninstall a Mirror Repository ................................. 91

Appendix 1 / Installing a Production Deployment ............... 93
  Pre-installation Steps for a Production Deployment .................................................. 94
  Configure a Proxy Server ....................................... 94
  Enable the Yum Cache ......................................... 95
  Perform Additional Linux Tuning for a Production Deployment ..................................... 95
  Optional Configuration for Enhanced Security .................................................. 98
  Install a Multi-Machine or a Production Deployment .................................................. 100
  (Optional) Configure Network Settings ................................. 100
  Edit the Inventory File for a Multi-Machine Deployment .................................................. 104
  Modify the vars.yml File ....................................... 107
  Specify the Path to Certificates ................................ 111
  Configure LDAP Settings ...................................... 112
  Install the Software .............................................. 112
  Complete the Deployment ..................................... 114
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.

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.”

This guide also contains instructions for installing and deploying SAS Event Stream Manager, which is automatically included with orders of SAS Event Stream Processing.

You have the option to use Docker to deploy SAS Event Stream Processing to a Kubernetes cluster. Those instructions are provided in Chapter 6, “Deploying with Containers,” on page 57.

To use this guide successfully, you should have a working knowledge of Ansible and the Linux operating system. You should be an experienced Docker and Kubernetes user if you want to use this guide to deploy SAS Event Stream Processing to a Kubernetes cluster.
SAS Event Stream Processing 6.2 is compatible with both SAS 9.4 and SAS Viya. When installed 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 or products. This guide provides the instructions for a stand-alone installation.

Step 1 — Prepare for the Deployment

1. Perform one of the following tasks:
   - To upgrade or update an existing deployment, go directly to Chapter 8, “Managing Your Software,” on page 69.
     
     Note: If the existing deployment includes SAS Event Stream Manager, it will be upgraded automatically. Otherwise, the SAS Event Stream Manager software will be installed automatically as an add-on package when the deployment is upgraded.

   - To deploy a new instance of the software, continue following these steps.

2. Go to Chapter 2, “System Requirements,” on page 5 to learn about requirements for hardware, software, security, and more.

3. Go to Chapter 3, “Pre-installation Tasks,” on page 15 to prepare your environment before you deploy the software.

4. To perform a production deployment, go to “Pre-installation Steps for a Production Deployment” on page 94 and “Optional Configuration for Enhanced Security” on page 98 to complete additional tasks (tune your operating system and secure your environment).

Step 2 — Perform the Deployment

1. Perform one of the following tasks:
   - To perform a rapid deployment, go to Chapter 4, “Installing a Rapid Deployment,” on page 33.
   
   - To perform a production deployment, go to Appendix 1, “Installing a Production Deployment,” on page 93.

2. Go to Chapter 5, “Post-installation Tasks,” on page 39 to perform post-installation configuration.
Step 3 — Validate and Complete the Deployment

1. Go to Chapter 7, “Validating the Deployment,” on page 63 to verify that the servers were deployed correctly and to locate the log files.

2. Go to Chapter 9, “Completing the Deployment,” on page 87 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.
Chapter 1 / Introduction
Hardware Requirements

The topics in this section provide information about hardware requirements for a SAS Event Stream Processing deployment.

General Hardware Considerations

SAS Event Stream Processing has a flexible architecture and can be installed on one or multiple machines. The software is licensed per event, so you can install the software on multiple machines without violating the license agreement.

The SAS Configuration Server component binds to a single private IP address per machine. If you plan to install on a host that has multiple network interface cards
(NICs), verify whether multiple NICs have been assigned IP addresses, including private IP addresses. SAS recommends that you use a network.conf file to specify the network address to be used for the deployment. Otherwise, the deployment uses default values. For more information, see “(Optional) Configure Network Settings” on page 100.

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. The default deployment installs SAS Event Stream Processing with client dependencies on some SAS Viya services, such as SAS Logon Manager.

SAS Event Stream Processing consists of an XML server (the ESP server) and three clients: SAS Event Stream Manager, SAS Event Stream Processing Studio, and SAS Event Stream Processing Streamviewer. A single machine for all components is the minimum requirement. However, SAS recommends that you install the clients on a separate computer from the ESP server.

On-premises deployments as well as cloud deployments are supported. With SAS Event Stream Processing 6.2 and later, you can deploy the software in a Kubernetes cluster using Docker. For those requirements, see “Requirements to Deploy with Containers” on page 8.

The following table describes a standard set of specifications for a machine where all SAS Event Stream Processing components are deployed (a single-machine deployment):

<table>
<thead>
<tr>
<th>Item</th>
<th>Recommended Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>8 cores (x86 architecture)</td>
</tr>
<tr>
<td></td>
<td>Intel Xeon chip set with a minimum speed of 2.6 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>16 GB of RAM</td>
</tr>
<tr>
<td></td>
<td>Memory clock speed of 1600 MHz</td>
</tr>
<tr>
<td>Disk Space and Speed</td>
<td>10 GB</td>
</tr>
<tr>
<td></td>
<td>10,000 RPM</td>
</tr>
</tbody>
</table>

The following table describes a standard set of specifications for a machine where only the ESP server is deployed. The ESP server is deployed on the machine target for the [espServer] host group:
Table 2.2 Minimum Hardware Requirements for the ESP Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Recommended Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>4 cores (x86 architecture)</td>
</tr>
<tr>
<td></td>
<td>Intel Xeon chip set with a minimum speed of 2.6 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>16 GB of RAM</td>
</tr>
<tr>
<td></td>
<td>Memory clock speed of 1600 MHz</td>
</tr>
<tr>
<td>Disk Space and Speed</td>
<td>10 GB</td>
</tr>
<tr>
<td></td>
<td>10,000 RPM</td>
</tr>
</tbody>
</table>

The following table describes a standard set of specifications for a machine where only the client applications are deployed. The clients are deployed on the machine targets for the [viprESM], [espStudio], and [espStreamviewer] host groups:

Table 2.3 Minimum Hardware Requirements for the Client Applications

<table>
<thead>
<tr>
<th>Item</th>
<th>Recommended Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>4 cores (x86 architecture)</td>
</tr>
<tr>
<td></td>
<td>Intel Xeon chip set with a minimum speed of 2.6 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>16 GB of RAM</td>
</tr>
<tr>
<td></td>
<td>Memory clock speed of 1600 MHz</td>
</tr>
<tr>
<td>Disk Space and Speed</td>
<td>10 GB</td>
</tr>
<tr>
<td></td>
<td>10,000 RPM</td>
</tr>
</tbody>
</table>

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

You can deploy the software on a redundant computer for failover. You can also deploy it on the compute layer of a Hadoop cluster, or at the edge of a Hadoop cluster on a gateway node.

GPU Requirements

SAS Event Stream Processing supports an optional graphics processing unit (GPU) environment for high-powered analytics calculations, such as scoring with analytic store (ASTORE) files. A GPU enhances the deep learning functionality in SAS Event Stream Processing streaming analytics.

Here are the requirements for GPU support in most SAS Event Stream Processing environments:

- GPU with NVIDIA Maxwell, Pascal, or Volta architecture
You must perform some post-deployment steps to enable GPU functionality. For more information, see "(Optional) Enable GPU Functionality" on page 50.

Requirements to Deploy with Containers

You can use a Docker image that is available in the SAS software repository to create containers that run SAS Event Stream Processing in a Kubernetes cluster. Both Docker Community Edition (CE) and Docker Enterprise Edition (EE) are supported. The deployment process has been tested with Docker 17.05.0-ce or later.

To check the Docker version on your machine:

docker -v

Kubernetes 1.14 or later is required. Reserve at least 10 GB of free disk space on at least one of the Kubernetes worker nodes. You will run commands to create the images on these machines.

To determine your version of Kubernetes:

kubectl version

The machine where Docker is installed must meet the following minimum requirements:

<table>
<thead>
<tr>
<th>Item</th>
<th>Recommended Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Dual or quad core x86_64 compatible processor</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB of available RAM</td>
</tr>
<tr>
<td>Disk Space</td>
<td>750 MB or more of free space for the installation (depending on the optional components that were selected in the software order)</td>
</tr>
</tbody>
</table>

Note: In this release, the Docker image does not include GPU support.
Operating System Requirements

Supported Operating Systems

For a list of supported operating systems, see https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html.

Note: SAS Event Stream Processing can also be installed on Microsoft Windows, but a separate package, based on your software order, is required.

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.

- curl–7.19.7-53 or later (Red Hat Enterprise Linux 6.x)
  On Red Hat Enterprise Linux 6.x, apply the RHSA-2017:0847-01 security update for curl to ensure that you have a supported version of the utility.
  On Oracle Linux 6.7 and later within 6.x, apply the ELSA-2017-0847 security update.

- curl–7.29.0-25 or later (Red Hat Enterprise Linux 7.x)
  Red Hat Enterprise Linux 7.x, Oracle Linux 7.x, and SUSE Linux 12.2 have a supported version of curl by default.

- glibc-2.12-1.166.el6 and later (on Red Hat Enterprise Linux 6.x or the equivalent). Refer to RHBA-2015:1465 on the Red Hat Customer Portal to obtain the latest updated package list.
glibc-2.17-107.el7 and later (on Red Hat Enterprise Linux 7.x or the equivalent). Refer to RHSA-2016:2573 on the Red Hat Customer Portal to obtain the latest updated package list.

glibc-2.22 and later (on SUSE Linux)

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

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

libXp

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

libXmu

net-tools

nss 3.36.0-7 or later

the numactl package

systemd version 219-30 or later

the X11 and Motif (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 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 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.

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 the client applications: SAS Event Stream Manager, SAS Event Stream Processing Studio, and SAS Event Stream Processing Streamviewer.

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 the client applications, follow the steps in “Configure LDAP Settings” on page 35 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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-login service account without user restrictions.</td>
<td>Required for the installation.</td>
</tr>
<tr>
<td>No password; can add password after installation if desired.</td>
<td>The installation process sets user and group ownership permissions on all of the installation files. This user must exist to enable ownership.</td>
</tr>
<tr>
<td>Password does not expire.</td>
<td>After the installation has completed, this user account enables required components to run, including the web application server for SAS Event Stream Processing Studio.</td>
</tr>
<tr>
<td>Default user name is required until the installation is complete.</td>
<td></td>
</tr>
<tr>
<td>Any post-installation changes to this account do not prevent future software updates.</td>
<td></td>
</tr>
</tbody>
</table>

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


#### 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 26.

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/35/support-for-operating-systems.html#ansible.

Ansible is not used for a containerized deployment to a Kubernetes cluster. For the requirements to deploy with Docker to a Kubernetes cluster, see “Requirements to Deploy with Containers” on page 8.
# Pre-installation Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare for a Rapid Deployment</td>
<td>15</td>
</tr>
<tr>
<td>Obtain the Required Files</td>
<td>16</td>
</tr>
<tr>
<td>Installing from a Mirror Repository</td>
<td>16</td>
</tr>
<tr>
<td>Create a Mirror Repository</td>
<td>17</td>
</tr>
<tr>
<td>Enable Required Ports</td>
<td>19</td>
</tr>
<tr>
<td>Set Environment Variables</td>
<td>21</td>
</tr>
<tr>
<td>Configure SELinux</td>
<td>22</td>
</tr>
<tr>
<td>Options for Deploying SAS Viya with SELinux</td>
<td>22</td>
</tr>
<tr>
<td>Disable SELinux and Deploy SAS Viya</td>
<td>22</td>
</tr>
<tr>
<td>Configure the Environment to Accommodate SELinux</td>
<td>22</td>
</tr>
<tr>
<td>Disable SELinux, Deploy SAS Viya, then Re-enable SELinux</td>
<td>23</td>
</tr>
<tr>
<td>Perform Linux Tuning</td>
<td>23</td>
</tr>
<tr>
<td>Set the ulimit Values</td>
<td>24</td>
</tr>
<tr>
<td>(SUSE Linux Only) Change the Maximum Number of Operating System Tasks</td>
<td>26</td>
</tr>
<tr>
<td>Install Ansible</td>
<td>26</td>
</tr>
<tr>
<td>Standard Ansible Installation</td>
<td>26</td>
</tr>
<tr>
<td>Streamlined Ansible Installation for Red Hat Enterprise Linux and</td>
<td>26</td>
</tr>
<tr>
<td>Equivalent Distributions</td>
<td>26</td>
</tr>
<tr>
<td>Streamlined Ansible Installation for SUSE Linux</td>
<td>27</td>
</tr>
<tr>
<td>Test Your Ansible Installation</td>
<td>28</td>
</tr>
<tr>
<td>Create a Playbook</td>
<td>28</td>
</tr>
<tr>
<td>Download the SAS Orchestration CLI</td>
<td>29</td>
</tr>
<tr>
<td>Create a Playbook with the SAS Orchestration CLI</td>
<td>29</td>
</tr>
<tr>
<td>Store the Playbook</td>
<td>31</td>
</tr>
</tbody>
</table>

## Prepare for a Rapid Deployment

Most of 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 Appendix 1, “Installing a Production Deployment,” on page 93. A mirror repository is required only for SUSE Linux.

If you are using Docker and Kubernetes to deploy SAS Event Stream Processing to a Kubernetes cluster, the pre-installation steps are different. Skip to Chapter 6, “Deploying with Containers,” on page 57 for the instructions.
Obtain the Required Files

When you order SAS software, SAS sends a Software Order Email (SOE) to your business or organization that includes information about the software order. Follow the steps in this section to save the attached ZIP file and download the additional file that is required.

1. Save the SAS_Viya_deployment_data.zip file that was attached to your SOE to a directory on the machine where you intend to deploy your software.
   This file contains entitlement certificates that will enable you to download the SAS software.

2. As directed by your SOE, download the SAS Orchestration CLI for the operating system where you stored the ZIP file. It is downloaded as a TGZ file (if your machine is running Linux) or ZIP file (Macintosh or Windows).
   The SOE recommended that you save the Orchestration file to a machine that runs Linux, which is where you will install your software. But you could also store it on a machine that runs Macintosh or Windows. Save the TGZ file in the same directory where you saved the SAS_Viya_deployment_data.zip file.

   Note: If you used Internet Explorer to download the Linux or Macintosh version of the SAS Orchestration CLI, change the file extension from .gz to .tgz.

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.

4. In the directory where you saved SAS_Viya_deployment_data.zip, uncompress it.
   Several subdirectories are created: /ca-certificates, /entitlement-certificates, and a /licenses directory.

Orders for SAS Event Stream Processing include software packages for all the supported Linux platforms.

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.

SAS Mirror Manager downloads the software that you ordered and creates a mirror repository. It can create the mirror repository in a specified location, such as a shared NFS mount point or a web server that serves the files with HTTP. The default location for the download is the `sas_repos` directory that is created in the installation user’s home directory. Make sure that the default location for the download and the destination for the mirror repository have adequate space.

This guide refers to the default location as `sas_repos`. If you want to specify the mirror destination, use the `--path` option, followed by the full directory path. In addition, replace instances of `sas_repos` that are used in this guide with the actual location that you select.

The `sas_repos` directories and files are explained as follows:

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

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 Microsoft Internet Explorer or Microsoft Edge 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. (Optional) Add the location of SAS Mirror Manager to your PATH environment variable.

Note: This step is not required. However, the example SAS Mirror Manager commands in this section assume that you have added the recommended location to your PATH.
Run the following basic command to create the mirror repository in the default location:

```
export PATH=/opt/sas/viya/home/bin:$PATH
```

**Note:** All the software to which your order entitles you is downloaded if you use the basic command in the previous step. To download software for selected target platforms, skip to the next step.

```
mirrormgr mirror --deployment-data path-to-SAS_Viya_deployment_data.zip
```

By default, the repositories are placed in the `sas_repos` directory in the installation user’s home directory. Use the `--path` option, followed by the full directory location of the mirror destination, to change this location.

6. (Optional) Run the following command to see a list of the platforms that you can select for the download operation:

```
mirrormgr list remote platforms --deployment-data path-to-SAS_Viya_deployment_data.zip
```

7. (Optional) Use the `--platform` option and one of the values that were returned by the `list remote platforms` command to download software only for a selected target platform:

```
mirrormgr mirror --deployment-data path-to-SAS_Viya_deployment_data.zip --path location-of-mirror-repository --platform platform --latest
```

8. (Optional) Use the `--latest` option to exclude any obsolete packages from the mirror repository that is being created.

If you use this option, be sure to use it with any subsequent `mirrormgr` commands. For example, to compare the contents of your mirror with the contents of SAS repositories, use the `diff` command with `--latest` if you used this option with the `mirror` command.

9. (Optional) After the initial download is complete, move the file structure to a web server or shared NFS mount point. Internet connectivity is not required for the destination machine.

Depending on your platform, you can use tools like `rsync` and `scp` to move the files. Here is a typical command for `rsync`:

```
rsync -av --progress sas_repos target_machine:/var/www/html/pulp/
```
Specify a Log Location

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

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

Enable Required Ports

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

Table 3.1  Ports to Be Made Available

<table>
<thead>
<tr>
<th>Process</th>
<th>Required Port</th>
<th>Requires Allowed Inbound Traffic From</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>httpd</td>
<td>80, 443</td>
<td>anywhere (SAS Viya servers, workstation)</td>
<td></td>
</tr>
<tr>
<td>SAS Infrastructure Data Server</td>
<td>5430–5439</td>
<td>SAS Viya Servers only</td>
<td>For a single server deployment with no failover, ports 5430-5432 must be opened in order to use the optional third-party PostgreSQL tools, pgAdmin and pgpoolAdmin. Additional standby nodes each get the next available port number sequentially up to 5439.</td>
</tr>
<tr>
<td>default SAS Messaging Broker AMQP client access port</td>
<td>5672</td>
<td>SAS Viya Servers only</td>
<td></td>
</tr>
<tr>
<td>Vault</td>
<td>8200</td>
<td>SAS Viya Servers only</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Required Port</td>
<td>Requires Allowed Inbound Traffic From</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------------</td>
<td>---------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAS Configuration Server</td>
<td>8300, 8301, 8302, 8500, 8501</td>
<td>SAS Viya Servers only</td>
<td>SAS uses HashiCorp Consul as its configuration server. Ports 8301 and 8302 must be open to both UDP and TCP traffic.</td>
</tr>
<tr>
<td>default SAS Messaging Broker management</td>
<td>15672</td>
<td>SAS Viya Servers only</td>
<td></td>
</tr>
<tr>
<td>web console port</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Any ports that will be used for ESP servers must be open to HTTP traffic. For more information, see Setting Up and 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-limitinclusive-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

Set Environment Variables

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

export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/6.2
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

(Optional) Enable data sharing so that you can embed charts and import data from this instance of SAS Event Stream Processing Streamviewer. Use the value “true” to enable data sharing and to disable CSRF protection.

export SV_ENABLE_DATA_SHARING=true

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.
Configure SELinux

If Security-Enhanced Linux (SELinux) is enabled in your environment, it must be disabled or accommodated before you can use Ansible to deploy SAS Viya.

Options for Deploying SAS Viya with SELinux

If SELinux is required in your environment, it is supported on Red Hat Enterprise Linux if you perform some additional tasks. On SUSE Linux, SELinux is not supported.

To determine the present status of SELinux in your environment:

```
sudo sestatus -v
```

- If you see a message that the command is not enabled, SELinux is not active in your environment. You can use Ansible to deploy SAS Viya.
- If a mode that is not **permissive** is returned, SELinux is enabled. In order to deploy SAS Viya, you must select one of the following options:
  - Disable SELinux and deploy SAS Viya.
  - Configure the environment to accommodate SELinux.
  - Disable SELinux, deploy SAS Viya, and then re-enable SELinux after the deployment has completed.

Disable SELinux and Deploy SAS Viya

If SELinux is active in your environment, one option is to configure **permissive** mode for SELinux on all the target machines in your deployment.

**Note:** Permissive mode effectively disables SELinux.

To change the mode value to **permissive** on all target machines in your deployment:

```
sudo setenforce 0
```

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

Configure the Environment to Accommodate SELinux

To deploy SAS Viya with SELinux enabled on all target machine in your deployment, perform all the following tasks:
1 Disable the pre-installation check that determines whether SELinux is active. For more information, see “Verify System Requirements” on page 108.

2 Configure SELinux to enable the Apache HTTP Server. By default, SELinux does not allow the Apache httpd component to access the network.

   Run the following command on any machines that are deployTargets for the [htpproxy] host group in the inventory.ini file:
   
   ```bash
   sudo setsebool -P httpd_can_network_connect 1
   ```

3 Make sure that the SELinux Policy deny_unknown status is set to allowed. Perform the following steps:

   a Run the following command to determine the current SELinux settings:
   
   ```bash
   sudo sestatus -v
   ```

   b Check the value of Policy deny_unknown status in the output. If the value is not allowed, you must change the policy setting.

   c As root, edit the `/etc/selinux/semanage.conf` file.

   d Add the following line:
   
   ```bash
   handle-unknown=allow
   ```

   e As root, run the following command to rebuild and reload the policy:
   
   ```bash
   semodule -B
   ```

---

**Disable SELinux, Deploy SAS Viya, then Re-enable SELinux**

A final option is to disable SELinux during the deployment and then re-enable it as soon as the deployment has completed. If you select this option, perform tasks 2 and 3, as described in “Configure the Environment to Accommodate SELinux”.

---

**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 95.
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
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.

```
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. 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](http://docs.ansible.com/ansible/latest/intro_installation.html). You should always follow the Ansible documentation and choose the installation method that works best for your IT environment.

Not all versions of Ansible that are available for installation are supported by SAS Viya. For a list of supported Ansible versions, see: [https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html#ansible](https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html#ansible). On that same page, SAS provides a list of supported versions of Python. Python support is determined by the release of Ansible that you install.

Streamlined Ansible Installation for Red Hat Enterprise Linux and Equivalent Distributions

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

1 Run the following commands to attach the EPEL repository to your server:
   ```bash
   ## 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
   # Display the available repositories
   sudo yum repolist
   ```

2 To Install Python PIP and related packages:
   ```bash
   sudo yum install -y python python-setuptools python-devel openssl-devel
   sudo yum install -y python-pip gcc wget automake libffi-devel python-six
   ```

3 Because EPEL will no longer be required, you can remove it with the following command:
   ```bash
   sudo yum remove -y epel-release
   ```

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

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

   For Python 2.6 (and later within 2.x):
   ```bash
   sudo pip install --upgrade pip==9.0.3
   sudo pip install pip==9.0.3
   sudo pip install pycparser==2.14
   sudo pip install idna==2.7
   ```

   For Python 2.7 (and later within 2.7.x):
   ```bash
   sudo pip install pip==19.3.1
   sudo pip install setuptools==42.0.2
   ```

   For Python 3.5 (and later within 3.x):
   ```bash
   sudo pip install --upgrade pip setuptools
   ```

5 To install a specific version of Ansible through PIP:
   ```bash
   sudo pip install ansible==2.7.2
   ```

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:
   ```sh
sudo zypper install python-setuptools
   ```

2. To Install Python PIP:
   ```sh
   sudo easy_install pip
   ```

3. To install a specific version of Ansible through PIP:
   ```sh
   sudo pip install ansible==2.7.2
   ```

Test Your Ansible Installation

1. To test the Ansible version:
   ```sh
   ansible --version
   ```
   Here is an example of successful output:
   ```text
   ansible 2.7.2
   config file =
   configured module search path = Default w/o overrides
   python version = 2.7.15 (default, May 14 2019, 07:55:04)
   [GCC 4.8.5 20150623 (Red Hat 4.8.5-14)]
   ```

2. To perform a basic ping test:
   ```sh
   ansible localhost -m ping
   ```
   Here is an example of successful output:
   ```yaml
   [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 17 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 Orchestration CLI download site](#) and download the SAS Orchestration CLI for the operating system where you stored the ZIP file.

   The SOE recommended that you save the ZIP file to a machine that runs Linux, which is where you install the SAS software that you purchased. However, you can also store it on a machine that runs Macintosh or Windows. If you use Microsoft Internet Explorer or Microsoft Edge 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 runs, rather than by the operating system where your SAS Viya software is deployed. After you create the playbook, you can move it to the machine where you deploy your software. Enter each command on a single line. Multiple lines are used here to improve readability.

**Linux or Macintosh**

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

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

For `deployment-platform-tag`, specify the target operating system, one of the following:

- `redhat` for Red Hat Enterprise Linux or an equivalent distribution, such as Oracle Linux
- `suse` for SUSE Linux Enterprise Server

For `deployment-architecture-tag`, specify the target chip for the deployment: `x64` for 64-bit chips.

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:

```bash
--java-option "-Dhttps.proxyHost=proxy-server-IP-address-or-host-name"
--java-option "-Dhttps.proxyUser=user-name"
--java-option "-Dhttps.proxyPassword=user-password"
```

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

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

For normal usage, when communicating with SAS, the options must start with `Dhttp`.

The Java options should not be combined into a single option. Here is an example of using the options on a Linux machine:

```bash
./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.

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

Here is an example for Windows:

```bash
./sas-orchestration build --input c:\sas\install\SAS_Viya_deployment_data.zip --repository-warehouse file://sas_repos
```
Note: The repository warehouse URL must be available to all hosts in the deployment to retrieve packages from the repositories. For example, if the repository warehouse is file-based, then that location should be shared across hosts and should be shared at the same path on each of those hosts. For more information about URLs, consult with your system administrator.

For more information about SAS Mirror Manager, see “Create a Mirror Repository” on page 17.

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 “Pre-installation Steps for a Production Deployment” on page 94.
Installing a Rapid Deployment

Edit the Playbook

The procedures that are described in this section are required to install SAS Event Stream Processing on a single computer, with basic security settings. The rapid installation procedures for a single-machine deployment or for a simple test environment do not include some required steps for a multi-machine or full-featured deployment. To install the software in a production environment, skip to Appendix 1, “Installing a Production Deployment,” on page 93.

Edit the Inventory File

Ansible uses an inventory file to identify 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 functions 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.

You can manually edit the default inventory.ini file, or you can copy and adapt a template from the sas_viya_playbook/samples subdirectory instead. This directory contains templates for different types of deployments, including a single-machine deployment. Copy the template that you want to use, rename it inventory.ini, and place it in the sas_viya_playbook directory, replacing the existing inventory.ini file.

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.
If you intend to install SAS Event Stream Processing on multiple machines, see “Edit the Inventory File for a Multi-Machine Deployment” on page 104 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. You can specify the machines where a host group is 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

To prepare an inventory file for a single-machine deployment:

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 software is being deployed and the user account that controls Ansible. Make sure that the user account has root or sudo permissions. With Ansible installed on the same machine where you are deploying SAS Event Stream Processing, do 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. Save and close the inventory.ini file.
Configure LDAP Settings

The sitedefault.yml file, which is in the /roles/consul/files directory in the playbook, is used to configure authentication for the SAS Event Stream Processing client applications. After the initial deployment, you cannot simply modify sitedefault.yml to change an existing value and deploy the software again. You can modify sitedefault.yml only to set property values that have not already 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. Locate the sitedefault_sample.yml file, located on the same machine as the playbook. 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 modify sitedefault.yml.

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

   Check your work carefully.

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

Install 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:

ansible-playbook system-assessment.yml

Before running the command, ensure that you are at the top level of the playbook in the sas_viya_playbook directory.

Add an option based on the password requirements for the user ID that performs the command:
Table 4.1

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td>use the command as written</td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td>--ask-become-pass</td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td>--ask-pass</td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td>--ask-pass --ask-become-pass</td>
</tr>
</tbody>
</table>

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

ansible-playbook --version


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:

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

Add an option based on the password requirements for the user ID that performs the command, using the table in "Assessment Test" on page 35.

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 basic command, 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 basic command:

```
ansible-playbook site.yml [ option ]
```
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, restart the deployment using the appropriate commands that are described in “Assessment Test” on page 35 and any appropriate options.

Be aware that 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
```

Complete the Deployment

When you have installed the SAS software and resolved any issues, you are ready to perform some final configuration tasks.

Proceed to Chapter 5, “Post-installation Tasks,” on page 39 for instructions on completing your SAS Event Stream Processing deployment.
Post-installation Tasks

Complete SAS Event Stream Processing Setup .......................... 39
  Enable Metering for ESP Servers .................................. 40
  Start the ESP Server .................................................. 41
  Log on to SAS Event Stream Processing Studio .................. 41
  Log on to SAS Event Stream Processing Streamviewer ......... 42
  Encryption and Authentication Options ......................... 43
  (Optional) Enable Kerberos Connections from SAS Event Stream Processing Studio .......... 44
  (Optional) Enable Kerberos Connections from SAS Event Stream Processing Streamviewer . 45

Complete SAS Event Stream Manager Setup .......................... 46
  Log on to SAS Event Stream Manager .............................. 46
  Configure the ESP Server for SAS Event Stream Manager ...... 46
  (Optional) Configure Kerberos Connections for SAS Event Stream Manager .................. 49

(Optional) Enable GPU Functionality ................................. 50

(Optional) Set Up SAS Event Stream Processing Streamviewer as a Stand-Alone Client Application .......... 50
  SAS Event Stream Processing Streamviewer Database Options ........ 51
  Start the Stand-Alone Application ................................ 51
  Connecting to Secure ESP Servers from SAS Event Stream Processing Streamviewer .......... 53
  Stop SAS Event Stream Processing Streamviewer ................ 53

(Optional) Set Up SAS Event Stream Processing Studio as a Stand-Alone Client Application .............. 53
  Modify the Start-Up Script ........................................... 54
  Connecting to Secure ESP Servers from SAS Event Stream Processing Studio ................. 55

Directory Structure and Permissions ................................. 55

Complete SAS Event Stream Processing Setup

A few steps are required to complete the deployment. This section also describes configuration steps to support a Kerberos environment. Kerberos is optional.
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.

The method to use to start the metering server depends on whether the metered billing server has been installed on the same machine as the ESP server. The playbook installs the metered billing server on the machine target for the [Operations] host group. The ESP server is installed on the machine target for the [espServer] host group.

When the [Operations] and [espServer] host group targets are the same machine, run the following basic command to start the metering server:

dfesp_metering -d

The server is started using its default port (31001).

The -d argument creates a log file and a database in the configuration directory (/opt/sas/viya/config).

When the [Operations] and [espServer] host group targets are on different machines, perform these steps:

1. Log on to the machine that hosts the [Operations] host group.
2. Create a directory where the metered billing server can write a database and log files.
3. Set the DFESP_METER_DATA environment variable. The value must be the full path to the writable directory that you created. Here is an example:

   ```bash
   export DFESP_METER_DATA=/opt/sas/viya/config/etc/ESPMeteredBilling/default
   ```

4. Run the basic command:

   ```bash
   dfesp_metering -d
   ```

   The server is started using its default port.

   The -d argument creates a log file in the configuration directory that you specified as the value of the environment variable.

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

When the playbook has completed, the SAS Event Stream Processing Studio, SAS Event Stream Processing Streamviewer, and SAS Event Stream Manager processes are already running. Before you can open or create a model in SAS Event Stream Processing Studio, you must start the ESP server.

If you want to use SAS Event Stream Manager to manage the SAS Event Stream Processing environment, you can take some additional steps to set up a connection between the ESP server and SAS Event Stream Manager. For more information, see “Configure the ESP Server for SAS Event Stream Manager” on page 46.

To start an ESP server:

1. Change directories:
   
   ```bash
   cd /opt/sas/viya/home/SASEventStreamProcessingEngine/version/bin
   ```
   
   For `version`, specify the version of the SAS Event Stream Processing software.

2. Start the ESP server. Here is an example of the command:
   
   ```bash
   ./dfesp_xml_server -pubsub n -http port &
   ```
   
   The ampersand (`&`) enables additional commands to be entered in the same window that started the server. Other values that provide server start-up instructions are defined in the esp-properties.yml configuration file. For more information, see Server Configuration Properties.

You can also configure the ESP server to start and run as a system daemon. You can then use init.d scripts to start and stop it. For more information, see Setting Up the ESP Server as a Daemon.

Log on to SAS Event Stream Processing Studio

Additional steps might be required to use SAS Event Stream Processing Studio, which provides a user interface for creating models. It is automatically started during the installation. However, you can start it manually if you find that the service is not running.

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:

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

   Here is an example:

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

   As an alternative, supply the location of the JDK, if applicable. Here is an example:

   ```bash
   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” on page 21.

3 SAS Event Stream Processing Studio should be running when the playbook
completes.

   To check the status of the espvm process:
   
   For Red Hat Enterprise Linux 6.x:
   ```
   sudo service sas-viya-espvm-default status
   ```
   
   For 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, to start it:

   For Red Hat Enterprise Linux 6.x:
   ```
   sudo service sas-viya-espvm-default start
   ```
   
   For Red Hat Enterprise Linux 7.x or SUSE Linux:
   ```
   sudo systemctl start sas-viya-espvm-default
   ```

5 When the service is running, 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 (Optional) To check the status of SAS Event Stream Processing Studio:

   For Red Hat Enterprise Linux 6.x:
   ```
   sudo service sas-viya-espvm-default status
   ```
   
   For Red Hat Enterprise Linux 7.x or SUSE Linux:
   ```
   sudo systemctl status sas-viya-espvm-default
   ```

---

**Log on to SAS Event Stream Processing Streamviewer**

When the deployment process has completed, take the following steps to access
SAS Event Stream Processing Streamviewer:

1 The Streamviewer process is started automatically by the playbook. To check the
status of the process:

   For Red Hat Enterprise Linux 6.x:
   ```
   sudo service sas-viya-espstrmvwr-default status
   ```
   
   For Red Hat Enterprise Linux 7.x or SUSE Linux:
   ```
   sudo systemctl status sas-viya-espstrmvwr-default
   ```
sudo systemctl status sas-viya-espstrmvwr-default

2 If it is not running, start the process:
For Red Hat Enterprise Linux 6.x:
sudo service sas-viya-espstrmvwr-default start
For Red Hat Enterprise Linux 7.x or SUSE Linux:
sudo systemctl start sas-viya-espstrmvwr-default

3 When the process is running, open the following URL:
http://ESP-server-host-name/SASEventStreamProcessingStreamviewer
For ESP-server-host-name, substitute the host name of the machine where SAS Event Stream Processing Streamviewer is installed.

4 Enter your user ID and password and click Sign in.
When you successfully log on to SAS Event Stream Processing Streamviewer, the home page appears.

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 SAS Event Stream Processing Streamviewer component 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

For more information about enabling security for an ESP server or for SAS Event Stream Processing Streamviewer, see SAS Event Stream Processing: Security.
(Optional) Enable Kerberos Connections from SAS Event Stream Processing Studio

Note: If Kerberos is not used for authentication in your environment, skip this section.

When Kerberos is configured for the machine where the ESP server is running, additional setup is required to enable connections from SAS Event Stream Processing Studio.

For SAS Event Stream Manager, the required steps to enable Kerberos connections are similar. For more information, see “(Optional) Configure Kerberos Connections for SAS Event Stream Manager” on page 49.

1. Contact a system administrator who maintains Kerberos configuration at your organization. Request the location of the keytab file and the user principal name for the machine where the ESP server is running.

2. Using a user account with sudoers privileges, log on to the machine where you have installed SAS Event Stream Processing Studio.

3. Use your preferred text editor to modify the following file, or create it if it does not already exist:

   /opt/sas/viya/config/etc/sysconfig/espvm.conf

4. Add the following lines to set two environment variables, substituting the values that you obtained from your system administrator. Here is an example:

   export ESM_KEYTAB_LOCATION=/etc/keytab-file-name
   export ESM_USER_PRINCIPAL=user-name/fully-qualified-host-name@KERBEROS-REALM

   For keytab-file-name, substitute the name of the keytab file such as krb5.keytab.

   For user-name, substitute the primary portion of the user principal name, which is typically a user name.

   For fully-qualified-host-name, substitute the fully qualified host name of the machine where the ESP server is running. An example is myhost.machine.domain.com.

   For KERBEROS-REALM, substitute the name of the Kerberos realm of which the user is a member, such as MYREALM.COM.

5. Save your changes to the file.

6. Restart the SAS Event Stream Processing Studio service. Run the appropriate command:

   For Red Hat Enterprise Linux 6.7:
   
   ```bash
   sudo service sas-viya-espvm-default stop
   sudo service sas-viya-espvm-default start
   ```

   For Red Hat Enterprise Linux 7.x or SUSE Linux:
   
   ```bash
   sudo systemctl stop sas-viya-espvm-default
   ```
sudo systemctl start sas-viya-espvm-default

(Optional) Enable Kerberos Connections from SAS Event Stream Processing Streamviewer

Note: If Kerberos is not used for authentication in your environment, skip this section.

When Kerberos is configured for the machine where the ESP server is running, additional setup is required to enable connections from SAS Event Stream Processing Streamviewer.

1 Contact a system administrator who maintains Kerberos configuration at your organization. Request the location of the keytab file and the user principal name for the machine where the ESP server is running.

2 Using a user account with sudoers privileges, log on to the machine where you have installed SAS Event Stream Processing Streamviewer.

3 Use your preferred text editor to modify the following file, or create it if it does not already exist:

   /opt/sas/viya/config/etc/sysconfig/espstrmvwr.conf

4 Add the following lines to set two environment variables, substituting the values that you obtained from your system administrator. Here is an example:

   export ESM_KEYTAB_LOCATION=/etc/keytab-file-name
   export ESM_USER_PRINCIPAL=user-name/fully-qualified-host-name@KERBEROS-REALM

   For keytab-file-name, substitute the name of the keytab file such as krb5.keytab.

   For user-name, substitute the primary portion of the user principal name, which is typically a user name.

   For fully-qualified-host-name, substitute the fully qualified host name of the machine where the ESP server is running. An example is myhost.machine.domain.com.

   For KERBEROS-REALM, substitute the name of the Kerberos realm of which the user is a member, such as MYREALM.COM.

5 Save your changes to the file.

6 Restart the Streamviewer process:

   For Red Hat Enterprise Linux 6.7:

   sudo service sas-viya-espstrmvwr-default stop
   sudo service sas-viya-espstrmvwr-default start

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

   sudo systemctl stop sas-viya-espstrmvwr-default
   sudo systemctl start sas-viya-espstrmvwr-default
Complete SAS Event Stream Manager Setup

If you plan to use SAS Event Stream Manager to automate the deployment of SAS Event Stream Processing projects and monitor their health, take a few steps after the installation has completed to prepare the environment. Otherwise, you can skip this section.

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 35.

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 65.

Configure the ESP Server for SAS Event Stream Manager

Note: If you do not plan to use SAS Event Stream Manager, skip this section.

To manage SAS Event Stream Processing instances with SAS Event Stream Manager, you must set up connections to ESP servers. You can use the SAS Event Stream Manager user interface to manually locate and define ESP servers that are running in your environment. However, you can also start your ESP servers with some additional instructions that enable secure, persistent sockets between SAS Event Stream Manager and ESP servers. SAS Event Stream Manager can then locate and manage ESP servers automatically.

To start an ESP server:
1. Provide SAS Logon Manager with a client ID and client secret for SAS Event Stream Manager. First, obtain the value of the SAS Configuration Server (Consul) token for your environment:

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

```
sudo cat /opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token
```

2. Run a curl command to request a registration token for a new client. In this example, the client is named app:

```
curl -X POST "http://localhost/SASLogon/oauth/clients/consul?callback=false&serviceId=app" -H "X-Consul-Token: X-Consul-Token-value"
```

For `X-Consul-Token-value`, substitute the value for the Consul token, which you obtained from the previous step.

Note: Specify the command on a single line. This request must pass a `callback=false` query string parameter and authenticate directly by passing a Consul token. If the Consul token that you specified in the command is valid, SAS Logon Manager returns the OAuth access token for registration in the response.

3. Use the registration token to register the client ID. This step establishes the ESP server as a new client of SAS Logon Manager. Run the following curl command:

```
  "client_id": "client-id",
  "client_secret": "client-secret",
  "scope": ["openid", "*"]
  "resource_ids": "none",
  "authorities": ["uaa.none"],
  "authorized_grant_types": ["password"]
}'
```

Note: You can find more information about the required steps to configure a new client for SAS Logon Manager in Obtain an Access Token Using Password Credentials in SAS Viya Administration: Authentication.

4. Create an XML file with filename esm.xml. Make sure that it uses the required syntax.

Here is an example:

```
<esm>
    <server name="SAS-Event-Stream-Manager-host">
        <url>http://reverse-proxy-server</url>
        <port>port-number</port>
        <context-path>context-path-to-SAS-Event-Stream-Manager</context-path>
    <auth>
```
For SAS-Event-Stream-Manager-host, substitute the host name of the machine where SAS Event Stream Manager is running.

For reverse-proxy-server, substitute the fully-qualified host name of the machine where the SAS Viya HTTP proxy server is running.

(Optional) For port-number, substitute the port where SAS Event Stream Manager is listening. This parameter is only needed if your instance is running without the SAS Viya HTTP proxy service.

(Optional) For context-path-to-SAS-Event-Stream-Manager, substitute the context path to your instance of SAS Event Stream Manager that is deployed without SAS Viya services. If nothing is specified, the default context path (/SASEventStreamManager) is used.

For client-ID, substitute the client ID that you provided to SAS Logon Manager for the SAS Event Stream Manager instance.

For client-secret, substitute the client secret that you provided to SAS Logon Manager for SAS Event Stream Manager.

For user-name, substitute a user name for an LDAP user account that is valid for use with SAS Logon Manager.

For password, substitute the password that corresponds to the user account that you specified.

Repeat the <server></server> section of the file as many times as required to accommodate all SAS Event Stream Manager servers.

5 Save the file in a network-accessible directory.

6 Change directories:
   cd /opt/sas/viya/home/SASEventStreamProcessingEngine/version/bin
   For version, specify the version of the SAS Event Stream Processing software.

7 Start the ESP server. Here is an example of the command:
   ./dfesp_xml_server -esm file://full-path-to-file/esm.xml
   The -esm file://esm.xml argument instructs the ESP server to use the values in the file that you created previously. Other values that provide server start-up instructions are defined in the esp-properties.yml configuration file. For more information, see Server Configuration Properties.

When it is started with the optional -esm file://esm.xml argument, the ESP server automatically registers with SAS Event Stream Manager, which can then manage it. The esm.xml file instructs the ESP server where to locate SAS Event Stream Manager. The ESP server registers itself with SAS Logon Manager as a new client with a new secret. SAS Logon Manager can then provide a token that enables the ESP server to set up a persistent web socket for secure communications with SAS Event Stream Manager.
(Optional) Configure Kerberos Connections for SAS Event Stream Manager

Note: If you do not plan to use SAS Event Stream Manager or if Kerberos is not enabled in your environment, skip this section.

When Kerberos is configured for the machine where the ESP server is running, additional setup is required. When Kerberos is used for authentication, you must edit a configuration file to enable SAS Event Stream Manager to connect to the ESP server.

1. Contact a system administrator who maintains Kerberos configuration at your organization. Request the location of the keytab file and the user principal name for the machine where the ESP server is running.

2. Using a user account with sudoers privileges, log on to the machine where you have installed SAS Event Stream Manager.

3. Use your preferred text editor to modify the following file, or create it if it does not already exist:

   `/opt/sas/viya/config/etc/sysconfig/esm-service.conf`

4. Add the following lines to set two environment variables, substituting the values that you obtained from your system administrator. Here is an example:

   ```
   export ESM_KEYTAB_LOCATION=/etc/keytab-file-name
   export ESM_USER_PRINCIPAL=user-name/fully-qualified-host-name@KERBEROS-REALM
   ```

   For `keytab-file-name`, substitute the name of the keytab file such as `krb5.keytab`.

   For `user-name`, substitute the primary portion of the user principal name, which is typically a user name.

   For `fully-qualified-host-name`, substitute the fully qualified host name of the machine where the ESP server is running. An example is `myhost.machine.domain.com`.

   For `KERBEROS-REALM`, substitute the name of the Kerberos realm of which the user is a member, such as `MYREALM.COM`.

5. Save your changes to the file.

6. 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
   ```
(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.

(Optional) Set Up SAS Event Stream Processing Streamviewer as a Stand-Alone Client Application

SAS Event Stream Processing Streamviewer is a web-based client that is used to visualize events that stream through event stream processing models. By default, SAS Event Stream Processing Streamviewer is installed as an integrated component of SAS Event Stream Processing. You can start using SAS Event Stream Processing Streamviewer as soon as the deployment process has completed.

However, you can also install SAS Event Stream Processing Streamviewer as a stand-alone client application. If you prefer to run it as an integrated component, skip this section.
SAS Event Stream Processing Streamviewer Database Options

SAS Event Stream Processing Streamviewer database support has changed in version 6.1 and later. By default, SAS Event Stream Processing Streamviewer is configured to use the same PostgreSQL database that SAS Event Stream Processing Studio uses. The PostgreSQL database is compatible with other SAS Viya products.

When you instead deploy SAS Event Stream Processing Streamviewer in stand-alone mode, it uses a different database. The SAS Event Stream Processing Streamviewer JAR file includes H2, a file-based database engine. When it runs as a stand-alone application, only the H2 database is supported. The configuration is stored in a file that is created when you invoke the application from the command line. For more information about H2, see http://www.h2database.com/.

Note: Some previous versions of SAS Event Stream Processing Streamviewer supported additional database management systems. If you have used one of those databases with a previous version of SAS Event Stream Processing Streamviewer, you can export your SAS Event Stream Processing Streamviewer data from that database and import it into version 6.2. For more information, see “Exporting SAS Event Stream Processing Streamviewer Data” on page 77.

Start the Stand-Alone Application

SAS Event Stream Processing Streamviewer is packaged in the streamviewer-6.2.jar file, which is located in /opt/sas/viya/home/libexec by default. All supported JDBC drivers are included in this JAR file.

You will use a shell script to start SAS Event Stream Processing Streamviewer as a stand-alone application. The script is saved in $DFESP_HOME/bin/streamviewer.sh by default.

1 Switch to the sas user account in order to have appropriate permissions:
   sudo -u sas -i

2 (Optional) If SAS Event Stream Processing Streamviewer will be running in a Kerberos environment, modify the start-up script to enable Kerberos connections to the ESP server.
   a Contact a system administrator who maintains Kerberos configuration at your organization. Request the location of the keytab file and the user principal name for the machine where the ESP server is running.
   b Use your preferred text editor to modify the script file:
      sudo vi /opt/sas/viya/home/bin/streamviewer.sh
   c At the beginning of the file, add the following two lines, which set the required Kerberos environment variables:
export ESM_KEYTAB_LOCATION=/etc/keytab-file-name
export ESM_USER_PRINCIPAL=user-name/fully-qualified-host-name@KERBEROS-REALM

For keytab-file-name, substitute the name of the keytab file such as krb5.keytab.
For user-name, substitute the primary portion of the user principal name, which is typically a user name.
For fully-qualified-host-name, substitute the fully qualified host name of the machine where the ESP server is running. An example is myhost.machine.domain.com.

d  Save and close the file.

3 Run a command that resembles the following:

streamviewer.sh -http http_port additional arguments

Here is an example:

streamviewer.sh -http 5990

For http_port, substitute an available port number. Do not specify the http or pubsub port that you specified when starting the ESP server.

For additional arguments, you can substitute one of the arguments that are described in the following table:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h2file</td>
<td>Specifies the name of a file where you want H2 to store the configuration. The file is created if it does not already exist. Specify the database file directory with a full path or point to the appropriate relative directory.</td>
</tr>
<tr>
<td>database_file</td>
<td>The H2 database file has the name <code>config</code>. If the database file <code>config</code> does not exist, the script creates the file in the current directory.</td>
</tr>
</tbody>
</table>

To log on to the SAS Event Stream Processing Streamviewer user interface, open the following URL:

http://Streamviewer-host-name/SASEventStreamProcessingStreamviewer

For Streamviewer-host-name, substitute the host name of the machine where SAS Event Stream Processing Streamviewer is installed.
Connecting to Secure ESP Servers from SAS Event Stream Processing Streamviewer

If you have set up authentication for an ESP server, you must provide authentication tokens or security credentials to the stand-alone application in order to connect to the ESP server. When connecting to a secure server, check the Is Secure box of the Edit ESP Server window. SAS Event Stream Processing Streamviewer prompts you for authentication information for the type of authentication that you have set up for the ESP server. For example, if you are using OAuth for authentication, an OAuth token is requested. If you are using SAS Logon Manager, a user name and password are requested. If you are using access control, make sure that the permissions.yml file is configured correctly.

Note: If you are using a permissions file for access control, every user-object combination must have its Read and Write access explicitly defined as true or false for all engines, projects, continuous queries, and windows in a model. You cannot view model objects for which you do not have explicit Read access.

Stop SAS Event Stream Processing Streamviewer

You can terminate the stand-alone application from the UNIX command line with the Ctrl + C or kill command.

(Optional) Set Up SAS Event Stream Processing Studio as a Stand-Alone Client Application

SAS Event Stream Processing Studio is a web-based client that enables you to create, edit, upload, publish, and test event stream processing models. By default, SAS Event Stream Processing Studio is installed as an integrated component of SAS Event Stream Processing. However, you can install it as a stand-alone application. If you do not want to run it as a separate application, you can start using SAS Event Stream Processing Studio after you complete the steps in "Log on to SAS Event Stream Processing Studio" on page 41.

Note: When SAS Event Stream Processing Studio is running in stand-alone mode, it cannot integrate with SAS Model Manager.
Modify the Start-Up Script

To enable SAS Event Stream Processing Studio to run in stand-alone mode, modify the start-up script:

**Note:** When run as a stand-alone application, SAS Event Stream Processing Studio does not use the same proxy server, configuration database, or logon service as SAS Event Stream Processing.

1. Stop the SAS Event Stream Processing Studio service:
   ```
sudo service sas-viya-espvm-default stop
   ```
2. Use your preferred text editor to open the start-up script for modification:
   ```
sudo vi /opt/sas/viya/home/bin/sas-espvm
   ```
3. Search for ESP_STUDIO_DB in the start-up script. Uncomment the line that contains ESP_STUDIO_DB. To uncomment, remove the number sign (#):
   ```
export ESP_STUDIO_DB= /opt/sas/viya/config/data/espvm/studio
   ```
   The line of code that is now enabled sets the ESP_STUDIO_DB environment variable, which defines the location of the database. SAS Event Stream Processing Studio will now use the H2 database that is installed with the software in order to store project data.
4. (Optional) If SAS Event Stream Processing Studio will be running in a Kerberos environment, two additional environment variables are required in order to enable Kerberos connections to the ESP server.
   a. Contact a system administrator who maintains Kerberos configuration at your organization. Request the location of the keytab file and the user principal name for the machine where the ESP server is running.
   b. At the beginning of the sas-espvm start-up script, add the following lines, substituting the values that you obtained from your system administrator. Here is an example:
      ```
export ESM_KEYTAB_LOCATION=/etc/keytab-file-name
export ESM_USER_PRINCIPAL=user-name/fully-qualified-host-name@KERBEROS-REALM
      ```
      For `keytab-file-name`, substitute the name of the keytab file such as krb5.keytab.
      For `user-name`, substitute the primary portion of the user principal name, which is typically a user name.
      For `fully-qualified-host-name`, substitute the fully qualified host name of the machine where the ESP server is running. An example is myhost.machine.domain.com.
      For `KERBEROS-REALM`, substitute the name of the Kerberos realm of which the user is a member such as MYREALM.COM.
5. Save and close the start-up script.
Start the service:

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

Connecting to Secure ESP Servers from SAS Event Stream Processing Studio

If you have set up authentication for an ESP server, you must provide authentication tokens or security credentials to the stand-alone application in order to connect to the ESP server. When connecting to a secure server, check the **Is Secure** box in the Edit ESP Server window. SAS Event Stream Processing Studio prompts you for authentication information for the authentication that you have set up for the ESP server. For example, if you are using OAuth for authentication, an OAuth token is requested. If you are using access control, make sure that the permissions.yml file is configured correctly.

**Note:** If you are using a permissions file for access control, every user-object combination must have its Read and Write access explicitly defined as true or false for all engines, projects, continuous queries, and windows in a model. You cannot view model objects for which you do not have explicit Read access.

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. Membership in this group also enables a user to create files in this directory that are required to use SAS Event Stream Processing Studio. For more information, see “User Accounts” on page 12.

Files for SAS Event Stream Manager are located in `./opt/sas/viya/home/SASEventStreamManager/`. 
Deploying with Containers

Installation Overview

This section describes steps to deploy a containerized version of SAS Event Stream Processing. Using Kubernetes to manage the SAS Event Stream Processing containers makes it easy to deploy and run the ESP server and SAS Event Stream Processing clients.

Before you begin, be sure to verify that your target machine meets the requirements in “Requirements to Deploy with Containers” on page 8.

Prepare the Kubernetes Environment

A Kubernetes cluster is required in order to deploy SAS Event Stream Processing in containers. Clusters that are managed by various third-party providers, such as Amazon EKS or Microsoft AKS, are supported. Share the following requirements with the system administrator who administers Kubernetes in your environment:

- A dedicated namespace for the SAS Event Stream Processing cluster. SAS recommends creating a namespace because the default Kubernetes namespaces are less secure.

- A persistent storage volume. It must be ready to bind to the PersistentVolumeClaim that the SAS Event Stream Processing metering server and the ESP server pods make in Kubernetes.
Obtain the Required Files

When you order SAS software, SAS sends a Software Order Email (SOE) to your business or organization that includes information about the software order. Follow the steps in this section to save the attached ZIP file and download the additional file that is required.

1. Save the SAS_Viya_deployment_data.zip file that was attached to your SOE to a directory on the machine where you intend to deploy your software.

   This file contains entitlement certificates that will enable you to download the SAS software.

2. As directed by your SOE, download the SAS Orchestration CLI for the operating system where you stored the ZIP file. It is downloaded as a TGZ file (if your machine is running Linux) or ZIP file (Macintosh or Windows).

   The SOE recommended that you save the Orchestration file to a machine that runs Linux, which is where you will install your software. But you could also store it on a machine that runs Macintosh or Windows. Save the TGZ file in the same directory where you saved the SAS_Viya_deployment_data.zip file.

   Note: If you used Internet Explorer to download the Linux or Macintosh version of the SAS Orchestration CLI, change the file extension from .gz to .tgz.

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.

4. In the directory where you saved SAS_Viya_deployment_data.zip, uncompress it.

   The following subdirectories are created: /ca-certificates and /entitlement-certificates.

Your entitlements enable you to install SAS Event Stream Processing on any of the supported platforms. Continue to follow the steps in this section to use the SAS Docker image to deploy SAS Event Stream Processing to a Kubernetes cluster.

Copy the Certificates

The SOE includes a ZIP file attachment, which contains entitlement certificates and a license file. These steps assume that you have already uncompressed the ZIP file and saved your SOE to a known location. To prepare for a deployment of SAS Event Stream Processing with containers, perform the following steps on the machine where you have installed the Docker engine:
Create a directory:

```bash
sudo mkdir -p /etc/docker/certs.d/
```

Create a directory under `/etc/docker/certs.d/` with the same name as the host name of the machine that contains the SAS secure software repository:

```bash
sudo mkdir -p /etc/docker/certs.d/ses.sas.download/
```

Copy the certificate files that were included in the ZIP file to the entitlement directory:

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

```bash
sudo cp ./ca-certificates/SAS_CA_Certificate.pem /etc/docker/certs.d/ses.sas.download/ca.crt
sudo cp ./entitlement-certificates/entitlement_certificate.pem /etc/docker/certs.d/ses.sas.download/client.key
sudo cp ./entitlement-certificates/entitlement_certificate.pem /etc/docker/certs.d/ses.sas.download/client.cert
```

These certificates will ensure that communications between the Docker registry server and the SAS secure software repositories are authenticated and encrypted.

Start or restart Docker:

```bash
sudo systemctl stop docker.service
sudo systemctl start docker.service
```

---

Create and Populate a Mirror Registry

A mirror registry is required in order to enable the local Docker engine to authenticate to the SAS secure repository.

Note: Sudoers privileges are required to perform some of these steps. As an alternative, membership in the docker group grants privileges that are equivalent to those of the root user.

1. To provide the required level of permissions to run Docker commands, create a docker group and add your user account to it:

```bash
sudo groupadd docker
```

```bash
sudo usermod -aG docker user-name
```

For `user-name`, substitute your user name, or the user name of the account that will perform the deployment.

2. Log off from the machine. Then log back on to the machine.
3 To restart Docker, run:

```sh
sudo systemctl start docker.service
sudo systemctl enable docker.service
```

4 Open the SOE that you received from SAS. The message text includes the names of SAS secure repositories from which your order entitles you to download software.

5 On the machine where you installed Docker, start a registry container. Here is an example:

```sh
docker run -d -p 5000:5000 --restart=always --name registry registry-name:2
```

For `registry-name`, substitute the name of your private registry.

6 Download SAS Mirror Manager from the SAS Mirror Manager download site to the machine where you want to create your mirror registry.

7 Uncompress the downloaded file.

8 (Optional) Add the location of SAS Mirror Manager to your PATH environment variable.

```
export PATH=/opt/sas/viya/home/bin:$PATH
```

9 Use SAS Mirror Manager to retrieve a list of available tags:

```
mirrormgr list remote docker tags --deployment-data SAS_Viya_deployment_data.zip --latest
```

10 Download the SAS Event Stream Processing images that were included in the list that was returned:

```
docker pull --all-tags ses.sas.download/esp-100.0.0-x64_redhat_linux_7-docker-latest/
docker inspect --all-tags ses.sas.download/esp-100.0.0-x64_redhat_linux_7-docker-latest/
```

11 Tag each image:

```
docker tag sas-repository/name-of-image/registry-host-name:port/tag
```

For `sas-repository`, substitute the name of the SAS secure repository. This information is included in your SOE.
For *name-of-image*, substitute the name of each image that you retrieved in the previous step.

For *registry-host-name:port*, substitute the host name and port of the machine that hosts your private registry.

Here is an example of the full command:

```
docker tag ses.sas.download/esp-100.0.0-x64_redhat_linux_7-docker-latest/sas-esp-base:version-numerical-identifier localhost:5000/sas-esp-base
```

12 Add each image to your Docker registry:

```
docker push registry-host-name:port/tag-for-base-image
```

For *registry-host-name:port*, substitute the host name and port of the machine that hosts your private registry.

---

**Deploy the Software to the Cluster**

Make sure that a system administrator has created the Kubernetes cluster and namespace. When you have taken the steps that are described earlier in this section to copy the certificates and set up the mirror registry, you are ready to deploy SAS Event Stream Processing in containers.

SAS provides tools that enable you to deploy and test an open ESP server in a Kubernetes cluster. The available resources consist of a set of scripts and YAML template files that enable you to start SAS Event Stream Processing containers and deploy them in your cluster. Navigate to the following GitHub project area to find the scripts and templates: https://github.com/sassoftware/esp-kubernetes.

In the same GitHub project area, SAS provides sample projects (XML files) that you can run in the ESP server and validate your Kubernetes deployment.
Chapter 6 / Deploying with Containers
Validating the Deployment

Verify the RPM Packages .......................................................... 63
Verify SAS Event Stream Manager Status ................................. 65
Access Log Files ................................................................. 65
Verify SAS Message Broker ...................................................... 66
Verify SAS Infrastructure Data Server ....................................... 66

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-6.2.20180109.06.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 commands on Red Hat Enterprise Linux 6.x:

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

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

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

The output is different on Linux 7.x or SUSE Linux, but it reports that the 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, a message appears, stating that 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...

<table>
<thead>
<tr>
<th>node_id</th>
<th>hostname</th>
<th>port</th>
<th>status</th>
<th>lb_weight</th>
<th>role</th>
<th>select_cnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>machine1</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>primary</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>machine2</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>standby</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>machine3</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>standby</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>machine4</td>
<td>5452</td>
<td>up</td>
<td>0.250000</td>
<td>standby</td>
<td>0</td>
</tr>
<tr>
<td>load_balance_node</td>
<td>replication_delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(4 rows)

A status of up for a node indicates the node is running.
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.
To upgrade a previous version of the software to version 6.2, start by consulting the Chapter 2, “System Requirements,” on page 5 section of this guide. Verify system requirements, perform pre-installation tasks, and install the new 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 5.x or 6.1 to version 6.2 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. 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.x. However, if you plan to import files that you created with SAS Event Stream Processing 3.2, be aware of the following issues:

- If you are using the Rabbit MQ connector, an additional post-upgrade step might be required. In previous releases, this connector did not have a protobuf message format type. Instead, you configured the connector with the rmqtype parameter set to binary. When the protofile and protomsg parameters were also set, protobuf support was used. With SAS Event Stream Processing 6.1 and later, the rmqtype parameter must be explicitly set to protobuf as soon as the upgrade has completed.

- If you are upgrading from SAS Event Stream Processing 5.x with SSL enabled, be aware of an additional upgrade step. In previous releases, you configured settings in the security-properties.yml file to enable SSL. Specifying the location of the certificate file (which contained the encryption key) was not required because SAS Event Stream Processing looked for a server.pem file if nothing was specified. However, SAS Event Stream Processing 6.1 and later require the location of the certificate file to enable SSL. Be sure to follow the steps in "Merge..."
Configuration Changes into a New File” on page 76 in order to enable SSL for SAS Event Stream Processing 6.2.

- Multiple XML elements in SAS Event Stream Processing 6.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 6.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.

---

## Prepare to Upgrade

Perform a few steps to prepare to upgrade your deployment.

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 Chapter 2, “System Requirements,” on page 5.

---

**Note:** Do not use a network.conf file to perform an upgrade. Previous releases of SAS Event Stream Processing used a different method to identify addresses to which

---

1. If you are upgrading from a version earlier than SAS Event Stream Processing 6.1, check for configuration files that have been customized.

   With SAS Event Stream Processing 6.1 and later, a single configuration file, esp-properties.yml, is used. Some customizations that you have made to an earlier version will not be used after the upgrade process has completed unless you merge them into the new file.

   On the machine where SAS Event Stream Processing is installed, compare any modified files with the copies that were included with the older version of the software. Some configuration files are not applicable to older releases of SAS Event Stream Processing, and the command produces a “not found” error.

   Run the following commands:

   **Note:** Specify each diff command on a single line. Multiple lines are used here for improved readability.

   ```bash
   diff -u $DFESP_HOME/etc/connectors.excluded /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/connectors.excluded
   diff -u $DFESP_HOME/etc/esp-logger.xml /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/esp-logger.xml
   diff -u $DFESP_HOME/etc/metatags.conf /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/metatags.conf
   diff -u $DFESP_HOME/etc/security-properties.yml /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/security-properties.yml
   ```

   2. Save the results of the diff commands to a file. You will refer to this file when you are ready to merge the configuration changes into esp-properties.yml. For more information, see “Merge Configuration Changes into a New File” on page 76.

   3. (Optional) Record the existing list of installed software before the upgrade.

      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

---

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

**Note:** Your upgrade must use a newer version of the SAS Orchestration CLI.

5. 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 28.

6. If you have a playbook from a previous installation of SAS Event Stream Processing, 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/
```

7. 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.

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.

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

Upgrade Using a Mirror Repository

If you are upgrading a SAS Viya deployment that used a mirror repository and you want to use a mirror repository again, you must create a new mirror repository to upgrade to SAS Event Stream Processing 6.2. For more information, see “Create a Mirror Repository” on page 17.

Back up Your Software and Stop Services

If you have SAS Event Stream Processing Studio 5.x, create a backup copy of the database. Otherwise, skip this section.

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

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

   For version, substitute the version of the installed SAS Event Stream Processing software, such as 5.2.

4. 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.

5. If you installed SAS Event Stream Processing Streamviewer, stop the Streamviewer process:
Upgrade SAS Software

To upgrade your deployment:

1. Log on to the PostgreSQL machine in your deployment.

2. Run the following command:
   
   ```bash
   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:
   
   ```bash
   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 PostgreSQL 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 SAS Event Stream Processing and SAS Event Stream Manager. Follow the steps in Chapter 4, "Installing a Rapid Deployment," on page 33 to install the software.
Note: Available upgrades are downloaded and installed. If you installed it previously, SAS Event Stream Manager is upgraded automatically. Otherwise, it is installed.

9 After the software has been installed, complete the following tasks, as appropriate:
   b Validate the Deployment on page 63.

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:
   
   ```bash
   sudo rpm -qq 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:
   
   ```bash
   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

---

**Merge Configuration Changes into a New File**

In "Prepare to Upgrade" on page 72, you were instructed to compare versions of SAS Event Stream Processing configuration files and save a file reflecting the differences between them. Now merge any differences that you found into the new configuration file for SAS Event Stream Processing 6.2, named esp-properties.yml. Or, if no changes were detected, your upgrade has completed.

To merge configuration changes into a new file:

1 Open the file that contains the results of the diff commands that you saved in "Prepare to Upgrade" on page 72.

2 Use your preferred text editor to modify the following file: /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default/esp-properties.yml.
3. For the customizations that you found in `connectors.excluded`, locate the `connectors:` section of `esp-properties.yml`. For each connector that you excluded, set the connector’s value to false.

4. For the customizations that you found in `esp-logger.xml`, copy the changes from your diff file to the logging: section of `esp-properties.yml`.

5. For the customizations that you found in `metatags.conf`, copy the changes from your diff file to the meta: section of `esp-properties.yml`.

6. For the customizations that you found in `security-properties.yml`, copy the changes from your diff file to the security: section of `esp-properties.yml`.

7. If your merged changes included enabling SSL but did not include updating the `server_cert_file` parameter, specify `server.pem` for the `server_cert_file` name in that same section of the file.

8. When you have completed the modifications, save `esp-properties.yml`.

9. Clean up unnecessary configuration files in order to avoid clashes. Change directories:
   
   ```
   cd /opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default
   ```

10. Delete the following files:
    - `connectors.excluded.rpmsave`
    - `esp-logger.xml.rpmsave`
    - `metatags.conf`
    - `metatags.conf.rpmnew` (an empty file that was created in order to preserve the contents of the actual file)
    - `security-properties.yml`
    - `security-properties.yml.rpmnew` (an empty file that was created in order to preserve the contents of the actual file)

   Note: Backup files are created only during an upgrade if customizations are detected.

---

### Exporting SAS Event Stream Processing Streamviewer Data

Starting with version 6.1, SAS Event Stream Processing Streamviewer is integrated with SAS Viya. By default, it uses the SAS Viya database, known as SAS Infrastructure Data Server and running on PostgreSQL. You also have the option to install SAS Event Stream Processing Streamviewer as a stand-alone application, in which case, it uses an H2 database.

Previous versions of SAS Event Stream Processing Streamviewer supported multiple database types. If you have data that you want to preserve from a previous
version of SAS Event Stream Processing Streamviewer, you must export it to a file. You can then import the file into SAS Event Stream Processing 6.2.

1 Open the SAS Event Stream Processing Streamviewer user interface from the following URL:

2 Supply your credentials to log on.

3 Click on the dashboard pane to export SAS Event Stream Processing Streamviewer data to a file. The Export Data window appears. This window shows the dashboard configuration in XML format.

4 Copy the XML configuration from the Export Data window to save it to a local file.

The file that was created by the export operation can be imported into any instance of SAS Event Stream Processing Streamviewer.

---

### Importing SAS Event Stream Processing Streamviewer Data

You can import SAS Event Stream Processing Streamviewer data directly from another running SAS Event Stream Processing Streamviewer instance or from a file. Perform the following steps:

1 Log on to the SAS Event Stream Processing Streamviewer user interface.

2 Click . The Import Data window appears.

3 Select whether to import data from a running SAS Event Stream Processing Streamviewer Server or from a File.

   If you select Server, complete the following fields and click Ok:

   - **Source Config URL**
     Enter the configuration URL of the server from which to import data. The URL is the host name and port of the SAS Event Stream Processing Streamviewer instance.

     **Note:** The SV_ENABLE_DATA_SHARING environment variable must be set to “true” on the source machine if it has SAS Event Stream Processing 6.2. For more information, see “Set Environment Variables” on page 21.

   - **Access Token**
     If the SAS Event Stream Processing Streamviewer instance is secure, enter the access token. Access tokens are not supported when SAS Event Stream Processing Streamviewer is running as a standalone application.

     If you select File, click Choose File to browse for the file to upload.
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-SAS_Viya_deployment_data.zip --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.
Synchronize the Mirror Repository

If you are using a mirror repository:

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

```
mirrormgr mirror diff --deployment-data path-to-SAS_Viya_deployment_data.zip --platform linux-distribution-value --path path-to-mirror-destination --latest
```

Note: Unless you add the `--platform` option, the `mirrormgr mirror diff` command returns the available files for all supported platforms of the products in the deployment. The `--platform` option filters out unwanted content from the output of the command.

2 Synchronize the deployment’s mirror repository with SAS’ mirror repository. Use the same options to update the mirror repository that you used to create the mirror repository.

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

```
mirrormgr mirror --deployment-data path-to-SAS_Viya_deployment_data.zip --path path-to-mirror-destination --latest
```

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

3 If you are in a deployment without internet access, move the files from the machine where the connected mirror repository is located to the machine where the unconnected mirror repository is located.
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 zypper list-updates | grep "sas-"
```

Deployments with a Mirror Repository

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-SAS_Viya_deployment_data.zip --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-SAS_Viya_deployment_data.zip --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-SAS_Viya_deployment_data.zip --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:
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:

   ```bash
   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:

   ```bash
   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:**

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

3. **If you installed SAS Event Stream Processing Studio as a stand-alone application, stop the espvm service.**

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

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

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

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

   **Note:** This step is not required if you used the default method to install SAS Event Stream Processing Studio.

4. **If you installed SAS Event Stream Processing Streamviewer as a stand-alone application, stop the Streamviewer process:**

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

   Replace **host-name** with the host name of the machine where SAS Event Stream Processing Streamviewer is installed and running.
Replace `http-port` with the port number that you provided when you started SAS Event Stream Processing Streamviewer with the start-up script.

**Note:** This step is not required if you used the default method to install SAS Event Stream Processing and SAS Event Stream Processing Streamviewer.

5 Stop the metering server:

```bash
$DFESP_HOME/bin/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:

```bash
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:

```bash
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:

```bash
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:
   ```bash
   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. If you installed SAS Event Stream Processing Studio as a stand-alone application, stop the espvm service.
   ```bash
   sudo /etc/init.d/sas-viya-espvm-default stop
   ```
   Note: This step is not required if you used the default method to install SAS Event Stream Processing Studio.

3. If you installed SAS Event Stream Processing Streamviewer as a stand-alone application, stop the Streamviewer process:
   ```bash
   $DFESP_HOME/bin/dfesp_xml_client -url "http://host-name:http-port/exit"
   ```
   Replace `host-name` with the host name of the machine where SAS Event Stream Processing Streamviewer is installed and running.
   Replace `http-port` with the port number that you provided when you started SAS Event Stream Processing Streamviewer with the start-up script.
   Note: This step is not required if you used the default method to install SAS Event Stream Processing and SAS Event Stream Processing Streamviewer.

4. Stop the metering server:
   ```bash
   $DFESP_HOME/bin/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:
   ```bash
   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

Note: This step is not required if you used the default method to install SAS Event Stream Processing Studio.
4 If you installed SAS Event Stream Processing as a stand-alone application, stop the Streamviewer process:

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

Replace `host-name` with the host name of the machine where SAS Event Stream Processing is running.

Replace `http-port` with the port number that you provided when you started SAS Event Stream Processing with the start-up script.

**Note:** This step is not required if you used the default method to install SAS Event Stream Processing.

5 Stop the metering server:

```bash
$DFESP_HOME/bin/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 following command:

```bash
ansible-playbook update-only.yml
```

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:

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

**Note:** If you receive a message that resembles 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.
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 another machine. You have a backup of important files that might be required later, such as to update an existing order.

Review Code Examples for SAS Event Stream Processing

After the deployment has completed, you can find examples to help you write SAS Event Stream Processing applications here.

The examples include files for XML, Python, and Java, with a brief description of each example and its usage. SAS recommends that you copy the files 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
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.

Now refer to the product documentation, which 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. SAS Help Center is also available as a menu item that you can access from the login ID in each client user interface. 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 plan to use SAS Event Stream Manager to automate and monitor your environment, find a full set of instructions in Using SAS Event Stream Manager.

The product page also offers multiple tutorials and examples to help you learn to use the software. See: https://support.sas.com/en/software/event-stream-processing-support.html#tutorials.
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.

Use deploy-cleanup

Use the deploy-cleanup play to uninstall SAS Event Stream Processing and SAS Event Stream Manager.

1. Stop the ESP server. Change directories to the following location:
   
   ```
   cd /opt/sas/viya/home/SASEventStreamProcessingEngine/version/bin
   ```
   
   For version, substitute the version of the installed SAS Event Stream Processing software, such as 6.2.
2 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.

3 Stop the metering server:
   
   dfesp_xml_client -url "http://host-name:http-port/SASESP/exit"
   
   Replace host-name with the host name of the machine where the metering server is running.
   
   Replace http-port with the port number for the metering server. By default, it uses port 31001.

4 Navigate to the top level of the playbook in the sas_viya_playbook directory.

5 Here is the basic syntax for the command to run the playbook and deploy the software:
   
   ansible-playbook deploy-cleanup.yml
   
   Add an option based on the password requirements for the user ID that performs the command. Refer to the following table for the options:

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td>use the command as written</td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td>--ask-become-pass</td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td>--ask-pass</td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td>--ask-pass --ask-become-pass</td>
</tr>
</tbody>
</table>

   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

**Pre-installation Steps for a Production Deployment**
- Configure a Proxy Server
  - Overview
  - Using curl
  - Using yum
- Enable the Yum Cache
- Perform Additional Linux Tuning for a Production Deployment
  - Set the ulimit Values
  - Set the Semaphore Values
  - (SUSE Linux Only) Change the Maximum Number of Operating System Tasks
- Optional Configuration for Enhanced Security
  - Transport Layer Security
- Install a Multi-Machine or a Production Deployment
- (Optional) Configure Network Settings
  - Specify Network Settings for Selected Machines
  - Variables for network.conf
- Edit the Inventory File for a Multi-Machine Deployment
  - Specify the Machines in the Deployment
  - Assign the Target Machines to Host Groups
- Modify the vars.yml File
  - Set the Deployment Label
  - Set the Pre-deployment Validation Parameters
  - Specify Security Settings
  - Change the Repository Warehouse
  - (Optional) Specify JRE
  - (Optional) Disable SELinux Verification
- Specify the Path to Certificates
- Configure LDAP Settings
- Install the Software
  - Assessment Test
  - Deployment Command
  - Options
  - Run from a Directory Other Than the Default
  - Successful Playbook Execution
  - Retry a Failed Deployment
- Complete the Deployment
Pre-installation Steps 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 Chapter 3, “Pre-installation Tasks,” on page 15.

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:

```bash
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:

```bash
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:

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

### Enable the Yum Cache

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

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

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

```bash
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 multi-machine deployment.

For a production deployment, perform these steps in addition to those that are described in "Perform Linux Tuning" on page 23.
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

A PostgreSQL database is installed in order to store configuration data and SAS Event Stream Processing Studio projects. For optimal performance, the machine where the database is installed requires slightly different settings. In a multi-machine deployment, the machine where the database is installed is the deployment target for the [sasdatasvrc] host group.

Change Linux settings for each machine in your deployment:

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

2. Set the limit for open file descriptors:

   - For the machine where PostgreSQL will be deployed, use the nofile item to set the limit to 150000 for the sas user.

   ```
sas     -     nofile     150000
   ```

   **Note:** For a rapid deployment, all software is installed on the same machine.
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 recommended limit for all users.

*     -     nofile     150000

3 For machines where 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.

Note: For a rapid deployment, all software is installed on the same machine.

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 where 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 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 111.

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.
Install a Multi-Machine or a Production Deployment

SAS recommends installing the three SAS Event Stream Processing clients on a separate machine from the ESP server. Follow the steps in this section if you want to deploy SAS Event Stream Processing on multiple machines.

Complete the tasks in this section in order.

(Optional) Configure Network Settings

Note: If you are upgrading your deployment to SAS Viya 3.5, skip this task. The processes described here do not apply to an upgrade.

Specify Network Settings for Selected Machines

Your deployment might include machines that require you to specify certain network settings. For example, a machine might have multiple network interface cards (NICs), and you want to specify the addresses to which SAS Viya services should bind. You can configure network settings for the SAS Viya deployment by using network configuration files. If you do not use network configuration files, default values are used. Be aware that changing the network configuration after the software has been deployed is not supported. Although this task is optional, be sure of your decision if you decide to skip it.

IMPORTANT Before creating network configuration files, you must create a new directory in your playbook named host_vars.

For each machine in your deployment with multiple NICs, to create the appropriate network configuration files:

1. Create a new file in the host_vars directory and use the deployment alias for the machine (as it is listed in the inventory.ini file) as the file name.

2. Ensure that the following two lines are at the top of the machine-name.yml file:

```yaml
---
network_conf:
```

Note: Spacing and indentation are important in .yml files.
In the `machine-name.yml` file, you can configure variables that relate to IP addresses for use with SAS Viya, network binding addresses, and IP addresses for external use. For details about the variables, see “Variables for network.conf” on page 101.

Save and close `machine-name.yml` file.

Here is an example of a completed network configuration file:

```yaml
---
network_conf:
  SAS_HOSTNAME: sasmachine1.mycompany.com
  SAS_BIND_ADDR: 192.168.1.23
  SAS_SERVICE_ADDR: sasmachine1.mycompany.com
  SAS_SAN_DNS: "sasmachine1.mycompany.com localhost"
  SAS_SAN_IP: "127.0.0.1 192.168.1.23"
```

**Variables for network.conf**

When adding variables to `machine-name.yml` file, be sure to precede each new variable with two spaces.

Configuration variables that support IP addresses are divided into three groups:

- IP addresses used with SAS Viya
- TLS Certificate Parameters
- Binding variables for external ports

SAS services usually identify themselves based on the host's IP address. Bind addresses can be specified directly with SAS_BIND_ADDR. However, you might prefer to use the network name or CIDR addressing options. The following variables allow the same value to be used across all machines in a deployment or to account for changes to an IP address on the same machine over time.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS_BIND_ADDR</td>
<td>The IP address for services on this machine. The value must be an IPv4 address. If the variable is omitted, services bind to 0.0.0.0. Do not set this variable to 0.0.0.0. However, to use the default binding, remove this variable or set it to an empty string. This value allows the SAS Viya deployment to be restricted to a specific network on a machine with multiple NICs.</td>
</tr>
<tr>
<td>SAS_BIND_ADDR_IF</td>
<td>If SAS_BIND_ADDR is not set, this variable can be used to specify a network name. SAS_BIND_ADDR is set to the first primary IP address that is found for that network.</td>
</tr>
</tbody>
</table>
### Variable Definitions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS_BIND_ADDR_CIDR</td>
<td>If SAS_BIND_ADDR and SAS_BIND_ADDR_IF are not set, this variable specifies an address range using CIDR notation, such as 192.168.100.0/24. SAS_BIND_ADDR is set to the first primary IP address that is found that matches the address range. For more information about CIDR notation, see Classless IN-ADDR.ARPA delegation.</td>
</tr>
</tbody>
</table>

The following variables are other network binding options.

**Table A1.2 Other Network Binding Options**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS_HOSTNAME</td>
<td>The default host name for this machine. This value is used to connect to this machine or register services that are running on this machine. This value is also used as the Common Name for TLS certificates that are generated for this machine.</td>
</tr>
<tr>
<td>SAS_SAN_DNS</td>
<td>A list of alternative DNS names for this machine. This variable is used for DNS entries under Subject Alternative Name for all TLS certificates that are generated for this machine. If you specify alternative DNS names for this machine, you must also specify localhost (or the value that is used for loopback). The list must be enclosed in quotation marks (&quot;). Individual values should be delimited by a space.</td>
</tr>
<tr>
<td>SAS_SAN_IP</td>
<td>A list of alternative IP addresses for this machine. This variable is used for IP Address entries under Subject Alternative Name for all TLS certificates that are generated for this machine. If you specify alternative IP addresses for this machine, you must also specify 127.0.0.1 (or the value that is used for loopback). The list must be enclosed in quotation marks (&quot;). Individual values should be delimited by a space.</td>
</tr>
<tr>
<td>SAS_USE_IP_REGISTRATION</td>
<td>If the variable is set to a non-blank value, the value directs services to register with the IP address that is specified in SAS_BIND_ADDR instead of the name in SAS_HOSTNAME.</td>
</tr>
<tr>
<td>SAS_SERVICE_ADDR</td>
<td>Overrides the host name or address that is used to register services. This value is not typically set to a custom value.</td>
</tr>
</tbody>
</table>
There are a few ports that need public access and might require different network binding settings than those specified in the previous table. The following variables are used to specify public access ports.

**Table A1.3  Variables for External Ports**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SASEXTERNAL_BIND_ADDR</td>
<td>The IP address for public services on this machine. If the value is omitted, public services bind to 0.0.0.0. Do not set this variable to 0.0.0.0. However, to set the default binding, remove this variable or set it to an empty string.</td>
</tr>
<tr>
<td>SASEXTERNAL_BIND_ADDR_IF</td>
<td>If SASEXTERNAL_BIND_ADDR is not set, this variable can be used to specify a network name. SASEXTERNAL_BIND_ADDR is then set to the first primary IP address that is found for that network.</td>
</tr>
<tr>
<td>SASEXTERNAL_BIND_ADDR_CIDR</td>
<td>If SASEXTERNAL_BIND_ADDR and SASEXTERNAL_BIND_ADDR_IF are not set, this variable specifies an address range using CIDR notation.</td>
</tr>
<tr>
<td>SASEXTERNAL_HOSTNAME</td>
<td>This variable is used to register public services that are running on this machine. The value is used as the Common Name for TLS certificates that are generated for public services on this machine.</td>
</tr>
<tr>
<td>SASEXTERNAL_SAN_DNS</td>
<td>A list of alternative DNS names for this machine. This variable is used for DNS entries under Subject Alternative Name for TLS certificates that are generated for external services on this machine. If you specify alternative DNS names for this machine, you must also specify localhost (or the value that is used for loopback). The list must be enclosed in quotation marks (*). Individual values should be delimited by a space.</td>
</tr>
<tr>
<td>SASEXTERNAL_SAN_IP</td>
<td>A list of alternate IP addresses for this machine. This variable is used for IP Address entries under Subject Alternative Name for TLS certificates that are generated for external services on this machine. If you specify alternative IP addresses for this machine, you must also specify 127.0.0.1 (or the value that is used for loopback). The list must be enclosed in quotation marks (*). Individual values should be delimited by a space.</td>
</tr>
<tr>
<td>SASEXTERNAL_SERVICE_ADDR</td>
<td>Overrides the host name or address that is used to register external services.</td>
</tr>
</tbody>
</table>
Edit the Inventory File for a Multi-Machine Deployment

You can use the same Ansible playbook to install SAS Event Stream Processing on a single machine (for a “rapid” deployment) or on multiple machines (for a “production” deployment). If you have already performed a rapid deployment, you can modify your playbook to add deployment targets or to change some variable settings. Perform the tasks in this section to set up your inventory file to install SAS Event Stream Processing components on separate machines.

Specify the Machines in the Deployment

Ansible uses an inventory file to identify 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 inventory 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 enables 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>
<thead>
<tr>
<th>Component of the Deployment Target Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployTarget</td>
<td>specifies the alias that is used by Ansible to refer to the physical machine definition. The default alias is <code>deployTarget</code>. 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 <code>ansible-controller</code>.</td>
</tr>
<tr>
<td>ansible_host</td>
<td>specifies any resolvable address for the target host, such as the IP address or fully qualified domain name.</td>
</tr>
<tr>
<td>ansible_user</td>
<td>specifies the user ID that is used by Ansible to connect to each of the remote machines and to run the deployment. Must have root or sudo permissions.</td>
</tr>
</tbody>
</table>
### Component of the Deployment Target Reference

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ansible_ssh_private_key_file</td>
</tr>
</tbody>
</table>

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.

If you are using Ansible locally (on the same machine where you are deploying SAS software), do not revise the deployment target definition. The following example specifies the deployment target to be used when SAS software will be deployed on a separate machine from the Ansible controller:

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

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

```bash
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
```

---

### Assign the Target Machines to Host Groups

The second section in the inventory file is used to assign deployment targets to each host group. The host group assignment list contains a mapping of the installable groups of software and the machines on which they will be deployed. Host groups 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. You can specify the machines on which a host group will be deployed by listing them under the host group name. More than one host group can be deployed on a machine.

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 an example of host group assignment:
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.

```ini
[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 facts when editing the inventory file:

- SAS recommends that you 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.

- The machine where you intend to install SAS Event Stream Manager must be specified in both the [viprESM] and [consul] host groups.
- The playbook installs the SAS Event Stream Processing metered billing server on the machine target for the [Operations] host group. The ESP server is installed on the machine target for the [espServer] host group. SAS recommends installing them 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:

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

To prepare an inventory file for a multi-machine deployment:

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.

   ```sh
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 software is being deployed and the user account that controls Ansible.

   If you are using Ansible on a separate host, 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:

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

3. Save and close the inventory.ini file.

Now take the steps that are described in “Modify the vars.yml File” on page 107 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 requirements. Note that all entries in the vars.yml file are case-sensitive. Modifying these settings is not required for a “rapid” deployment of SAS Event Stream Processing.
Open the file from the `sas_viya_playbook` directory, and use your preferred text editor to modify it. If you used the recommended location for uncompressing your playbook, the file is located at `/sas/install/sas_viya_playbook-vars.yml`.

---

**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: "esp-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 [https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html#ansible](https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html#ansible).*

---

**Verify System Requirements**

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

1. Verify that SELinux mode for each machine is disabled. The value should be set to `permissive`, which disables SELinux.

   This check is performed by default. You can deploy SAS Viya in an environment where SELinux is enabled. However, some additional configuration is required. For more information about the requirements to run with SELinux enabled, see "Configure SELinux" on page 22.

   You can temporarily set SELinux to `permissive` until the deployment has completed. However, if you instead deploy with SELinux enabled (that is, set to `enforcing`), you must add a key-value pair that disables playbook checks for
For each machine, verify the nofile and nproc settings for the install user.

Note: For more information about setting ulimits, see “Set the ulimit Values” on page 24.

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:
   ```yaml
   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:
   ```yaml
   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.

(Optional) Disable SELinux Verification

If SELinux is active in your environment, and you want to deploy SAS Viya with SELinux enabled, add a key-value pair.

1. Open the vars.yml file if it is not already open.
2. Add the following statement at the bottom of the file:
   ```yaml
   VERIFY_SELINUX: false
   ```
3. Save and close the vars.yml file.

For more information about SELinux, see “Configure SELinux” on page 22.
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.

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.

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

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

   - 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.
Configure LDAP Settings

The sitedefault.yml file, which is in the /roles/consul/files directory in the playbook, is used to configure authentication for the SAS Event Stream Processing client applications. After the initial deployment, you cannot simply modify sitedefault.yml to change an existing value and deploy the software again. You can modify sitedefault.yml only to set property values that have not already 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. Locate the sitedefault_sample.yml file, located on the same machine as the playbook. 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 modify sitedefault.yml.

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

Check your work carefully.

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

Install 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:

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

Before running the command, ensure that you are at the top level of the playbook in the sas_viya_playbook directory.

Add an option based on the password requirements for the user ID that performs the command:
Table A1.5  Command Options Based on Password Requirements

<table>
<thead>
<tr>
<th>Password Requirements</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not require passwords</td>
<td>use the command as written</td>
</tr>
<tr>
<td>Requires a sudo password only</td>
<td>--ask-become-pass</td>
</tr>
<tr>
<td>Requires an SSH password only</td>
<td>--ask-pass</td>
</tr>
<tr>
<td>Requires both a sudo and an SSH password</td>
<td>--ask-pass --ask-become-pass</td>
</tr>
</tbody>
</table>

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 [https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html#ansible](https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-operating-systems.html#ansible).

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:

`ansible-playbook site.yml [ option ]`

Add an option based on the password requirements for the user ID that performs the command, using the table in “Assessment Test” on page 112.

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 basic command, 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 basic command:

`ansible-playbook site.yml [ option ]`
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, restart the deployment using the appropriate commands that are described in “Assessment Test” on page 112 and any appropriate options.

Be aware that failures can occur if there are port conflicts.

Complete the Deployment

When you have installed the SAS software and resolved any issues, you are ready to perform some final configuration tasks.

Proceed to Chapter 5, “Post-installation Tasks,” on page 39 for instructions on completing your SAS Event Stream Processing deployment.