Contents

Chapter 1 / Introduction .................................................................................. 1
  Steps for a Successful Deployment ............................................................. 1
  Contact SAS Technical Support .................................................................. 2

Chapter 2 / System Requirements .................................................................. 3
  Hardware Requirements ............................................................................... 3
  Docker Requirements ................................................................................. 4
  Operating System Requirements ................................................................. 5
  Java ............................................................................................................ 6
  GPU Requirements ..................................................................................... 6
  User Accounts ............................................................................................ 7
  Encryption and Authentication Options ...................................................... 8

Chapter 3 / Pre-installation Tasks .................................................................. 11
  Save the Required Files ............................................................................. 11
  Create a Mirror Repository ........................................................................ 12

Chapter 4 / Installation on Linux ................................................................... 15
  Installation Overview .................................................................................. 15
  Complete the Installation .......................................................................... 16
  (Optional) Enable GPU Functionality ......................................................... 18

Chapter 5 / Installation with Docker ............................................................... 19
  Installation Overview .................................................................................. 19
  Copy the Certificates for a Deployment with Docker ................................... 19
  Create and Populate a Mirror Registry ....................................................... 20
  Next Steps .................................................................................................. 23
  Mount a Persistent Storage Volume and Apply the License ....................... 23
  Encode and Apply the Product License ..................................................... 24
  Manage Run-Time Settings ....................................................................... 24
  Enable TLS for the Docker Container ....................................................... 27
  Start the Container and Deploy a Model .................................................... 28
  Stop a Container ....................................................................................... 29

Chapter 6 / Validating the Deployment ............................................................ 31
  Verify the Installed Packages ...................................................................... 31

Chapter 7 / Next Steps .................................................................................... 35
  Directory Structure and Permissions ......................................................... 35
  Product Documentation .............................................................................. 35

Chapter 8 / Managing Your Software .............................................................. 37
  Overview .................................................................................................... 37
  Upgrading Your Software ......................................................................... 38
  Updating Your Software .......................................................................... 42

Chapter 9 / Uninstalling SAS Event Stream Processing ................................. 45
  Uninstall SAS Event Stream Processing ..................................................... 45
Introduction

Steps for a Successful Deployment

Before You Begin

Use this guide to deploy SAS Event Stream Processing for Edge Computing. SAS Event Stream Processing products enable developers to create applications that can quickly process and analyze a large number of continuously flowing events in real time. SAS Event Stream Processing for Edge Computing is optimized for deployment to multiple targets on various device architectures.

- 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.

- You can deploy the software on machines that run a supported version of Linux or by creating a Docker container that includes the software. To use this guide successfully, you should have a working knowledge of the Linux operating system or experience with using Docker containers, respectively.

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 37.

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

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

3 Go to Chapter 3, “Pre-installation Tasks,” on page 11 to prepare the environment before you deploy the software.

Step 2 — Perform the Deployment

Perform one of the following:

- Go to Chapter 4, “Installation on Linux,” on page 15 to complete the installation on a Linux machine.
- Go to Chapter 5, “Installation with Docker,” on page 19 to use the Docker image provided by SAS to create Docker containers that run the software.

Step 3 — Validate the Deployment

Go to Chapter 6, “Validating the Deployment,” on page 31 to verify that your software has been installed properly.

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.
Hardware Requirements

The minimum requirement is a single machine. SAS recommends using an additional Linux machine to serve as a secure administrative host (a jump server) where a mirror repository is created. Otherwise, you can create the mirror repository on the same machine that is the target for the SAS Event Stream Processing deployment as long as it is running 64-bit Linux and meets the other system requirements.

Here are the standard specifications for a machine where SAS Event Stream Processing for Edge Computing is deployed:

<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></td>
<td>64-bit ARMv8 chipsets are supported.</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB of available RAM</td>
</tr>
</tbody>
</table>
**Recommended Level***

<table>
<thead>
<tr>
<th>Item</th>
<th>Recommended Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space</td>
<td>400 MB - 1.5 GB free space for the installation (depending on the optional components that were selected in the software order)</td>
</tr>
</tbody>
</table>

*The minimum requirements for an installation of SAS Event Stream Processing for Edge Computing are a single-core x86_64 or a 64-bit ARMv7 processor, 1 GB of available RAM, and 400 MB of free disk space. However, a minimum configuration is not recommended.

## Docker Requirements

You can use a Docker image that is available in the SAS software repository to create Docker containers that run SAS Event Stream Processing for Edge Computing.

Both Docker Community Edition (CE) and Docker Enterprise Edition (EE) are supported. SAS Event Stream Processing for Edge Computing has been tested with Docker 17.05.0-ce or later.

Check the Docker version on your machine by running the following command:

```
docker -v
```

The target machine for the installation must meet the following minimum requirements:

**Table 2.2 Hardware Requirements to Support the Docker Container**

<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 of SAS Event Stream Processing for Edge Computing, the Docker image does not include GPU support.
Operating System Requirements

The following requirements apply to all installations of SAS Event Stream Processing for Edge Computing. If you are using the SAS Docker image to create Docker containers running SAS Event Stream Processing for Edge Computing, all operating-system requirements are supplied by the image.

Supported Operating Systems

The following operating systems (64-bit versions) are supported:

Note: Docker deployments are supported only on the x86 architectures that are included in this list, with the exception of Windows. Docker deployments are not supported on Windows.

- Debian Linux 9 for 64-bit x86_64 and 64-bit ARMv8 chipsets
- Microsoft Windows 10 Enterprise 2016 LTSC, Version 1607 and later
- Red Hat Enterprise Linux versions 6.7 (64-bit) and later within 6.x
- Red Hat Enterprise Linux version 7.1 (64-bit) and later within 7.x
- SUSE Linux Enterprise Server 12.2. OpenSUSE is not supported.
- Ubuntu Linux LTS versions 14 through 18.04
  SAS has tested Ubuntu 14 and later on 64-bit AMD chipsets.
  SAS has tested Ubuntu 16.x and later on either 64-bit AMD chipsets or 64-bit ARMv8 (Aarch6) chipsets.
  SAS has tested Ubuntu 18.04 LTS (64-bit) on NVIDIA Jetson products that include NVIDIA JetPack 4.2.1 or later.
- Wind River Linux 7 (Yocto Linux 1.7.2) and later for 64-bit x86_64

Note: For 64-bit ARMv8 chipsets, Debian Linux 9 for 64-bit x86_64 or Ubuntu Linux 16.04 or later is required.

Linux Prerequisites

The typical Linux installation includes all the packages and libraries that SAS requires. Problems can occur if default packages were removed from the base operating system.

Before you start the deployment, verify that the numactl package is installed. The libnuma library, which is included with numactl, is also required.
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.

Java

A Java Runtime Environment (JRE) version 1.8 must be installed on every machine in your deployment.

Only the JRE is required, not the full JDK.

If you are using the SAS Docker image to create Docker containers running SAS Event Stream Processing for Edge Computing, this requirement is supplied by the image.

GPU Requirements

SAS Event Stream Processing for Edge Computing 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 with a standard SAS Event Stream Processing for Edge Computing Linux deployment on x86_64 or 64-bit ARM chipsets:

- For x86_64, NVIDIA Pascal or a Volta-based Tesla class GPU is required.
- For 64-bit ARM, NVIDIA Jetson Nano, TX2, or AGX Xavier embedded computing module is required.

NVIDIA JetPack 4.2.1 is also required.

This configuration includes Ubuntu 18.04 LTS (64-bit).

**Note:** On supported NVIDIA Jetson devices, the user who is running the ESP server must be a member of the video group in order to access the GPU. The Linux administrator must add any users who will use the GPU for scoring streaming data to this group.

Unless you are deploying with an embedded computing module, a few post-deployment steps are required to enable GPU functionality. For more information, see *(Optional) Enable GPU Functionality* on page 18.

### User Accounts

If you are using the SAS Docker image to create Docker containers running SAS Event Stream Processing for Edge Computing, these requirements are provided by the image. You can skip this section.

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 to own SAS components during normal product operation, you must not delete it or change its name. It does not run as root.

The following table describes the predefined sas user account:
### 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.

**Note:** If you are using Docker to deploy SAS Event Stream Processing for Edge Computing, security options are different. For more information, see “Enable TLS for the Docker Container” on page 27.

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

Authentication and encryption apply to the following ESP server APIs:

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

- C, Java, or Python Publish/Subscribe API
- Connections that are created by a client that uses the C, Java, or Python Publish/Subscribe API to communicate with an ESP server
- Connections that are created by an adapter to communicate with an ESP server

For more information about enabling security for an ESP server, see *SAS Event Stream Processing: Security.*
Save the Required Files

When you order SAS software, SAS sends a Software Order Email (SOE) to your business or organization. Your SOE includes information about the software order, including a ZIP file attachment. To prepare for deployment of SAS Event Stream Processing for Edge Computing, perform the following steps:

1. Save the ZIP file, SAS_Viya_deployment_data.zip, to a directory on the computer where you will download the SAS software that you purchased.

2. Uncompress the ZIP file. The following files are copied to the local computer:
   - the license file (TXT file)
   - certificates that enable access to your software from SAS repositories (PEM files)

   These files are required to deploy the software.

3. Rename the license file that was supplied by the ZIP file as license.txt. This file name is required.

4. Save the SOE to a known location. It includes a list of software packages to which your order entitles you. When you deploy the software to your edge devices, you can refer to this list to verify that all the software has been installed.

All orders for SAS Event Stream Processing for Edge Computing include software packages for all the supported platforms. If you plan to use the SAS Docker image to create Docker containers running SAS Event Stream Processing for Edge Computing, the next steps are different. Skip to Chapter 5, “Installation with Docker,” on page 19 for deployment instructions.
Create a Mirror Repository

The steps in this section enable you to create a mirror repository and populate it with the software that you purchased from SAS. Create the mirror repository on a jump server from which you can deploy it to your edge devices.

SAS Mirror Manager Requirements

SAS Mirror Manager is a command-line utility for synchronizing a collection of SAS software repositories. Its primary use is to create and manage mirror repositories for software deployment. A mirror repository is required for SAS Event Stream Processing for Edge Computing.

Consider the requirements for your mirror repository:

- The computer that you will use as a jump server is running 64-bit Linux.
- The directory where the files will be downloaded has adequate space. For more information, see the Hardware Requirements on page 3.
- The default location for the mirror repository is the \espedge_repos\ directory of the installation user. Verify the available space in the default location or in the location that you select.
- The mirror repository must be installed in a location to which the target computers for the deployment have access through a web server or NFS.

Create and Populate a Mirror Repository

Use SAS Mirror Manager along with the optional extensions if you want to create a mirror repository with separate directories for each category of packages. The extensions for SAS Event Stream Processing for Edge Computing create and populate four directories for the SAS software to which your order entitles you.

If you decide to install all software, install the contents of the /basic directory first. Then proceed in the order specified here:

- /basic
- /analytics
- /astore
- /textanalytics
- /gpu

The last four directories provide optional additions to the functionality that is provided in the /basic directory.

To use SAS Event Stream Processing for Edge Computing extensions, take the following steps:
1 The Software Order Email (SOE) indicated that you should save the file attachments. If you have not already done so, save them now.

2 Download SAS Mirror Manager from the SAS Mirror Manager download site to the machine where you want to create your mirror repository. If you use Internet Explorer to download the Linux or Macintosh version, save the file as a .tgz file instead of a .gz file.

3 Uncompress the downloaded file.

4 Download the SAS Event Stream Processing for Edge Computing Extensions package, sas-edge-extension.tgz, from the Mirror Extension Website.

5 Uncompress sas-edge-extension.tgz in the same directory where you uncompressed SAS Mirror Manager.

6 Run the following command:
   
   edge_mirror.sh list-of-repositories

   The list of software repositories to which you are entitled is included in the SOE. For list-of-repositories, substitute the names of the software repositories that you want to create on the target computer. Separate the names of multiple repositories with spaces.

   You must include at least one repository in the command. Here is an example:

   edge_mirror.sh sas-espedge-100.0.0-aarch64_ubuntu_linux_16-apt-latest

   This edge_mirror script uses SAS Mirror Manager to download the specified software to repositories on your computer.

   The packages are placed in separate subdirectories under espedge_repos \repository-name. As a result, you can selectively deploy packages to your edge devices.

   A separate package is provided for each of the supported platforms or deployment architectures.

7 Now proceed to the Installation on Linux section of this guide.
Installation Overview

You have already performed a series of steps to create a mirror repository and to download the SAS Event Stream Processing for Edge Computing software from SAS repositories. After the initial download is complete, move the file structure to a web server or shared NFS mount.

You can use tools like rsync or scp to move the files. Here is an example command for rsync:

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

You can also use your preferred centralized management software or other procedures that are used at your enterprise to deploy SAS Event Stream Processing for Edge Computing to multiple targets from the local repository that you created.

After you have deployed the software to edge devices using your preferred method, complete the remaining steps in this section. You must apply the product license, set the required environment variables, and start the ESP server.
Complete the Installation

Apply the License

A valid license file is required in order to run any applications that use SAS Event Stream Processing for Edge Computing.

Your SOE contained a license file that you were instructed to save. Now you must apply the license file to the local machine by saving it to the license directory.

1. Locate the license file that you previously saved.

2. Copy the license file to the license directory.

   Run the following command:
   
   ```
   sudo cp license.txt /opt/sas/viya/home/SASEventStreamProcessingEngine/6.2/etc/license/
   ```

   Note: The file name `license.txt` is required.

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:
   ```
   export DFESP_METER_DATA=/opt/sas/viya/config/etc/ESPMeteredBilling/default
   ```
4. Run the basic command:
   ```
   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.

### Set the Required Environment Variables

You must set several environment variables before you start SAS Event Stream Processing for Edge Computing. For a shell that is used to invoke SAS Event Stream Processing for Edge Computing only, 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
```

### Start the ESP Server

To start the SAS processes and to verify that the software is running:

1. On the machine where you installed SAS Event Stream Processing for Edge Computing, run the following command to start the ESP server:
   ```
   $DFESP_HOME/bin/dfesp_xml_server -http 9900
   ```
   The `-http` argument runs the ESP server as a factory server that supports the creation of projects. In the example, port 9900 is used for HTTP administration requests.

   Other values that provide server start-up instructions are defined in the esp-properties.yml configuration file. For more information, see Server Configuration Properties.
The following INFO message is displayed:

```
Access control disabled (could not open permissions.yml, error: file not found)
```

The file that is referenced is required only to enable access control on the ESP server. You can ignore this message.

3 Validate the deployment. For more information, see Chapter 6, “Validating the Deployment,” on page 31.

---

(Optional) Enable GPU Functionality

The SAS GPU Reservation service helps SAS processes share resources and use GPUs that are available on a system. It is required on every machine where you want to use additional GPU functionality.

It is not necessary to perform the SAS software deployment again in order to add GPU functionality. After you have installed the software on your edge devices, to enable GPU functionality:

1 Using a user account that has sudoers privileges, log on to the machine where the GPU is installed.

2 If you have not already done so, install the contents of the /gpu directory that was created when you ran the SAS Mirror Manager Extensions.

3 Verify that the GPU Reservation service (sasgpud) has been installed in /etc/init.d.

4 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 service. If any requirements have not been met, you see an error message, and the service is not started.

5 Configure sasgpud so that it is automatically restarted when the device is restarted. For example, you can use systemd to set the service to start automatically.

Run this script again whenever a GPU device is added to or removed from the system.
### Installation Overview

This section describes steps to deploy the Docker container for SAS Event Stream Processing for Edge Computing. Using Docker to deploy the SAS Event Stream Processing for Edge Computing image makes it easy to manage your SAS software, including updates and upgrades.

Before you begin, be sure to verify that your target machine meets the requirements in “Docker Requirements” on page 4.

### Copy the Certificates for a Deployment with Docker

The software order email (SOE) is described in “Save the Required Files” on page 11. It 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. You have also renamed the license file that was supplied by the ZIP file as license.txt.

The steps for retrieving and copying the files are slightly different for a Docker deployment. To prepare for a Docker deployment of SAS Event Stream Processing for Edge Computing, perform the following steps on the machine where you have installed the Docker engine:

1. Verify that you have a supported version of Docker. For more information, see “Docker Requirements” on page 4.

2. Create a directory:
   ```
sudo mkdir -p /etc/docker/certs.d/
   ```

3. 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:
   ```
sudo mkdir -p /etc/docker/certs.d/ses.sas.download/
   ```

4. Copy the certificate files that were included in the ZIP file to the entitlement directory:
   ```
   Note: Specify each command on a single line. Multiple lines are used here to improve readability.
   ```
   ```
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.

5. Start or restart Docker:
   ```
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:

```
sudo groupadd docker
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:

```
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:

```
docker run -d -p 5000:5000 --restart=always --name registry-name registry: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.
For more information about using SAS Mirror Manager, see “Create a Mirror
Repository” on page 12.

7 Uncompress the downloaded file.

8 Download the SAS Event Stream Processing for Edge Computing Extensions
package, sas-edge-extension.tgz, from the Mirror Extension Website.
The optional Extensions package creates separate directories for each SAS
Event Stream Processing for Edge Computing package. It also provides a TXT
file that lists the contents of each directory.

9 Uncompress sas-edge-extension.tgz in the same directory where you
uncompressed SAS Mirror Manager.

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

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

11 Retrieve the SAS Event Stream Processing for Edge Computing base image
from the SAS software repository:

```
docker pull ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-
docker-latest/sas-espedge:version-numerical-identifier
```
docker inspect ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-
docker-latest/sas-espedge:version-numerical-identifier

For **version**, substitute the version of SAS Event Stream Processing for Edge Computing, such as 6.2.0. For **numerical-identifier**, substitute the unique numerical ID for each image.

**12 (Optional)** Retrieve the SAS Event Stream Processing for Edge Computing add-on image from the SAS software repository. This image includes the optional SAS Event Stream Processing Analytics and ASTORE support software:

```
donfig pull ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-
docker-latest/sas-espedge-analytics:version-numerical-identifier
donfig inspect ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-
docker-latest/sas-espedge-analytics:version-numerical-identifier
```

**13** Tag the base image:

```
donfig tag sas-repository/name-of-base-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-base-image**, substitute the name of the 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:

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

**14 (Optional)** Tag the add-on image, if you retrieved it:

```
donfig tag sas-repository/name-of-add-on-image/registry-host-name:port/tag
```

**15** Add the base image to your private registry:

```
donfig 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.

**16 (Optional)** Add the add-on image to your local registry:

```
donfig push registry-host-name:port/tag-for-add-on-image
```
Next Steps

After you have completed the steps in "Create and Populate a Mirror Registry" on page 20, your private registry has been populated with the SAS software to which your software order entitled you. You are now ready to deploy the container to your edge devices. Use the accepted software and processes at your enterprise to deploy the container from the local repository that you created to your target machines.

Choose one of the following methods for applying the product license:

- By default, SAS Event Stream Processing for Edge Computing looks for the license in an external storage volume that has been mounted as /data. For more information, see "Mount a Persistent Storage Volume and Apply the License" on page 23.

- The other method for applying the license requires that the license file be Base64-encoded. If you do not plan to use an external storage volume, follow the steps that are described in "Encode and Apply the Product License" on page 24.

Mount a Persistent Storage Volume and Apply the License

SAS Event Stream Processing for Edge Computing can be configured to use persistent external storage. The use of a storage volume is a recommended way to preserve settings in a configuration file. You can also store the SAS Event Stream Processing license file and provide it to enable the ESP server.

The sas user account, which is created automatically during the deployment process, is the default owner of the volume and is present within the container. You might be required to grant Read and Write permissions to the container so that data can be written to the external storage volume. SAS recommends using the docker run command to retrieve the UID of the sas user account that is present within the container.

To mount the volume and apply the product license:

1. Save the license file in the root location of the host directory. The file name license.txt is also required. The file format must be plain text.
   
   When the container starts up, the entry point will place the license in the required location.

2. Start the container. Here is an example:

   ```bash
docker run -v path-to-license-on-host:/data private-registry/host-name:port/sas-espedge-base:tag
   ```
Note: The /data directory path is required.

For path-to-license-on-host, substitute the directory path with the location of the license file in the volume mount.

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

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

### Encode and Apply the Product License

By default, SAS Event Stream Processing for Edge Computing looks for the file named license.txt in a storage volume that has been mounted as /data. This method of applying the product license requires the file to be in plain text format.

As an alternative, you can use a license.txt file that is Base64-encoded in order to authorize the ESP server. You must then export the ESP_LICENSE environment variable.

To use ESP_LICENSE to encode and apply the product license file:

1. Use Linux commands to set the variable and encode the license file. Here is an example:
   ```bash
   export ESP_LICENSE=$(base64 ./license.txt)
   ```

2. To configure the Docker container with the environment variable:

   ```bash
   docker run -e ESP_LICENSE=$ESP_LICENSE private-registry/host-name:port /sas-espedge-base:tag
   ```

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

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

   For tag, substitute the tag for the base SAS Event Stream Processing for Edge Computing image.

   The Docker entry point writes the value of the variable into a license file in the container for the SAS Event Stream Processing server to use.

### Manage Run-Time Settings

SAS Event Stream Processing for Edge Computing uses the configuration file esp-properties.yml to specify settings that affect container security, ESP server start-up
behavior, logging levels, and more. You can override the default settings when you start the container.

Commonly Used Settings

Key-value pairs that are used to define some product run-time default settings are stored in the configuration file esp-properties.yml. These default settings enable you to start using SAS Event Stream Processing immediately.

However, when you start the container, you can specify settings to override the default values that are assigned to the ESPENV environment variable. For more information, see “Override Default Settings” on page 26.

The following settings are the most commonly used to run SAS Event Stream Processing for Edge Computing. The syntax to use in order to override the default values is provided:

Table 5.1 Commonly Used Settings in the esp-properties.yml File

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description and Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>31415</td>
<td>Specifies the HTTP port for the ESP server’s HTTP REST API. Do not use a value that exceeds 65535. &lt;br&gt;alert{server.http=port}</td>
</tr>
<tr>
<td>License</td>
<td>commented out by default.</td>
<td>Specifies a full path to the product license file, which enables the software to run. You can use the ESP_LICENSE environment variable to set this location. &lt;br&gt;alert{ESP_LICENSE=pathname} &lt;br&gt;For more information, see Encode and Apply the Product License on page 24.</td>
</tr>
<tr>
<td>loglevel</td>
<td>commented out by default.</td>
<td>Sets the logging level for the ESP server. Here are the supported logging level values: off</td>
</tr>
<tr>
<td>model</td>
<td>commented out by default.</td>
<td>Specifies the URL to the XML model. Use the URL notation for files, specifying the full path to the XML model (file://path). &lt;br&gt;To specify a secure HTTP server, use ports. An example is http 8091s. &lt;br&gt;alert{server.model=file://model.xml}</td>
</tr>
</tbody>
</table>
### Property, Value, Description and Syntax

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description and Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>pubsub</td>
<td>31416</td>
<td>Specifies a port for the ESP server's publish and subscribe actions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>server.pubsub=port</td>
</tr>
<tr>
<td>single port mode</td>
<td>false</td>
<td>Specifies that a single port is used for ESP server publish/subscribe actions, rather</td>
</tr>
<tr>
<td></td>
<td></td>
<td>than a series of ephemeral ports.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>server.single_port_mode=false</td>
</tr>
</tbody>
</table>

For more information about esp-properties.yml, see Setting Up and Using the ESP Server.

### Override Default Settings

After the container starts, it looks in the esp-properties.yml configuration file for settings that affect the ESP server and other aspects of the environment. You override a default setting by passing it as a parameter in the container start-up command.

To override default settings:

1. If required, use SAS Mirror Manager to retrieve a list of available tags:
   ```bash
   run ./mirrormgr list remote docker tags --deployment-data --latest
   ```
2. Start the container with the values whose default settings you want to override. Here is an example that changes three default values:

   Note: Specify the command on a single line. Multiple lines are used here to improve readability. You can separate multiple values to override with commas, as shown here.

   ```bash
   docker run -e ESP_LICENSE="$ESP_LICENSE" -e ESPENV="server.http=44001,server.pubsub=54001,server.model=file:///model.xml" private-registry/repository-name/sas-espedge-base:tag
   ```

   For more information about the ESP_LICENSE environment variable, see “Encode and Apply the Product License” on page 24.

   For more information about the settings that are specified in this example, see “Commonly Used Settings” on page 25.

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

   For `repository-name`, substitute the name of the mirror repository.

   For `tag`, substitute the tag for the base SAS Event Stream Processing for Edge Computing image.

   Here is an example of the command to override the default logging level for the ESP server that is specified in esp-properties.yml:
Enable TLS for the Docker Container

If your ESP server has Transport Layer Security (TLS) enabled, you must specify several security parameters for the Docker container. Default security settings are provided in a single configuration file, esp-properties.yml. Any changes to the default values are passed in with an environment variable when you start the container.

If TLS is not enabled in your environment, skip this section.

1. Review the available settings and their default values in Security Configuration Properties.

2. Create an encryption key and a certificate. Here is an example that uses OpenSSL to create a self-signed certificate that expires in one year:

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

openssl req -x509 -newkey rsa:4096 -nodes -keyout key-file-name.pem -out cert-file-name.pem -days 365 -subj '/CN=127.0.0.1'
```

For `key-file-name`, substitute the name that you want to assign to your encryption key file.

For `cert-file-name`, substitute the name that you want to assign to your client certificate.

3. Concatenate the key and certificate in order to create the ESP server's certificate:

```
cat cert-file-name.pem key-file-name.pem > server-file-name.pem
```

For `cert-file-name`, substitute the name of your client certificate.

For `key-file-name`, substitute the name of your encryption key file.

For `server-file-name`, substitute the name that you want to assign to the new server certificate file.

4. Copy these three security files to the directory in your external volume where they will be available to the Docker container.

5. Set an environment variable to specify the locations of the security files:

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

```


For location-of-files, substitute the directory where you copied the three files in the previous step.

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

For repository-name, substitute the name of the mirror repository.

For tag, substitute the tag for the base SAS Event Stream Processing for Edge Computing image.

6 Test the encrypted connection to the ESP server. Run the following commands at the client:

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

```
curl --tlsv1.2 --ssl-reqd --cacert ./cert-file-name.pem --capath ./ --key-key-file-name.pem -kv https://127.0.0.1:31415/SASESP/server
```

For cert-file-name, substitute the name of your client certificate.

For key-file-name, substitute the name of your encryption key file.

Results from the curl command alert you to any problems with the configuration process.

---

Start the Container and Deploy a Model

The ESPENV environment variable enables you to start the container and pass a model to the ESP server for execution. You must first save the model that you want to deploy in an accessible location.

The ESPENV environment variable uses the settings that you defined in the $DFESP_HOME/etc/esp-properties.yaml file. For more information about these settings, see "Commonly Used Settings" on page 25.

Run the following command on a single line. Multiple lines have been used here for improved readability:

```
docker run -v $(pwd)/:data -v $DOCKER_TEST_CONFIG_DIR:/docker-test - e ESP_LICENSE="$ESP_LICENSE" - e ESPENV="${ESPENV},server.http=$PORT_HTTP,server.pubsub=$PORT_PUBSUB,server.model=file" $DOCKER_IMAGE_NAME
```

For file, use syntax like the following: file://model.xml for the location of the model XML file.
Stop a Container

You can use Docker commands to manage the SAS Event Stream Processing containers.

1. List all containers that are running:
   
   ```
   docker ps
   ```

2. Stop a container:

   ```
   docker stop container-ID
   ```

   For `container-ID`, substitute the ID of the container that you want to stop.
Validating the Deployment

Verify the Installed Packages

The steps that you can take to verify that all of your software has been installed are different if you installed on Red Hat Enterprise Linux or another supported Linux platform.

Validate the SAS RPM Packages

Note: If you are using Docker to deploy SAS Event Stream Processing for Edge Computing, you can skip this section.

When you deployed on Red Hat Enterprise Linux or Yocto Linux, your SAS software was delivered in RPM (Red Hat Package Manager) packages. You can check these packages for errors after you have installed them.

To obtain a list of all SAS Event Stream Processing for Edge Computing 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-6.2.0-build-identifier.x86_64 package, run the following command:

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

Note: Run the above commands for each host on which you have deployed SAS Event Stream Processing for Edge Computing.
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:

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

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

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

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

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

The error indicators are shown in the following format:

```bash
SMDLUGT c
```

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

```bash
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, since they are likely to change and therefore are unlikely to verify successfully.

---

## Validate the SAS Packages on Ubuntu

When you install SAS Event Stream Processing for Edge Computing on Ubuntu platforms, the software is copied to your machine in DEB packaging format. You can use the dpkg and debsums utilities to validate the deployment. Perform the following steps to check the SAS packages for errors after you have completed the installation:

1. **On the machine where you installed the SAS packages from the SAS repositories, run the following commands to verify the packages:**
   ```bash
   dpkg -l | grep -i sas-
   ```
   A list of installed packages is returned.

2. **Validate the MD5 checksums of all files that have them. For files that do not have MD5 checksums, use dpkg to verify them. Run the following commands to install debsums and to perform validation:**
   ```bash
   sudo apt-get install debsums
debsums $(dpkg -l | awk '{print $2}' | grep -e "^sas-*")
dpkg -V sas-espbase
   ```

3. **Check the return codes for errors. Here is an example of the output showing typical return codes:**
   ```bash
   ??5?????? /usr/share/doc/sas-tkdq/copyright
   ```
   A ‘?’ character indicates that the check could not be performed. For example, permissions on the file might not allow a check.
A ‘.’ character indicates that the check passed.

An alphanumeric character indicates that a specific check failed.

A ‘5’ in the third position indicates the MD5 checksum verification. A ‘5’ is typically followed by a space, an attribute character, and then another space before the path to the file whose checksum was verified.
Directory Structure and Permissions

After you install SAS Event Stream Processing for Edge Computing, 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. For more information, see “User Accounts” on page 7.

Product Documentation

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

The next step is to consult the product documentation. The product documentation is included in SAS Help Center. A link to all SAS Event Stream Processing documentation is available on the SAS Event Stream Processing product page. 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.

SAS has provided examples to help you write SAS Event Stream Processing applications. You can find them on the SAS Support website 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.
If you plan to use SAS Event Stream Manager, review the user documentation. For a full set of instructions about how to manage SAS Event Stream Processing deployments, see Using SAS Event Stream Manager.
Managing Your Software

Overview

SAS Event Stream Processing for Edge Computing supports both upgrades and updates. Each process is distinct and separate from the other.

If you are using a PDF version of this guide, go to the Deployment Guides web page at https://support.sas.com/en/documentation/install-center/viya/deployment-guides.html and verify that you have the latest version of the deployment documentation before you start the upgrade or update process. The release date of each document is located in the bottom right corner of the front page.

What Is an Upgrade?

An upgrade adds significant changes or improvements to your deployed software. You will need a new software order to upgrade your deployed software.

First, refer to Chapter 2, “System Requirements,” on page 3 in case requirements have changed. Perform pre-installation tasks and then follow the steps in “Upgrading Your Software” on page 38 to install the software.
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 Your Software

Overview

An upgrade adds significant changes or improvements to your deployed software. An upgrade might require changes to the deployed software’s configuration.

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

If you deployed SAS Event Stream Processing for Edge Computing 6.1 in a Docker container, stop the container and then follow the steps in Chapter 5, “Installation with Docker,” on page 19 to deploy a new container.

Prepare to Upgrade

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

With SAS Event Stream Processing for Edge Computing 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. For more information about this new file, see “Commonly Used Settings” on page 25.

To perform an upgrade, you will use SAS Mirror Manager to download software into a mirror repository. To install only a subset of the software packages that are available, you can use the edge_extension.sh script.

1 Check for configuration files that have been customized.

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. Run the following commands:
Note: Specify each diff command on a single line. Multiple lines are used here for improved readability.

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 consult this file when you are ready to migrate the configuration changes to esp-properties.yml. For more information, see “Merge Configuration Changes into New File” on page 41.

3 Verify that your environment meets the requirements that are listed in Chapter 2, “System Requirements,” on page 3. System requirements for RAM, CPU, and disk space might change with each release.

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

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

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

5 When performing an upgrade, you receive a new Software Order Email (SOE) from SAS. Save the SOE to a known location.

It includes a list of software packages to which you are entitled. When you deploy the software to your edge devices, you can refer to this list to verify that all the software has been installed.

6 Extract the license file (license-build-identifier.txt) from the SAS_Viya_deployment_data.zip file that is attached to your SOE. Save it to a known location with the required file name, license.txt.

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

8 Uncompress the downloaded file.

9 Download the SAS Event Stream Processing for Edge Computing Extensions package, sas-edge-extension.tgz, from the Mirror Extension Website.

10 Uncompress sas-edge-extension.tgz in the same directory where you uncompressed SAS Mirror Manager.

11 To refresh your mirror repository with the newer versions of the SAS software:

```bash
edge_mirror.sh list-of-repositories
```

The list of software repositories to which you are entitled is included in the SOE. For list-of-repositories, substitute the names of the software repositories that you
want to create on the target computer. Separate the names of multiple repositories with spaces.

You must include at least one repository in the command. Here is an example:

```
edge_mirror.sh sas-espedge-100.0.0-aarch64_ubuntu_linux_16-apt-latest
```

This edge_mirror script uses SAS Mirror Manager to download the specified software to repositories on your computer.

The packages are placed in separate subdirectories under `espedge_repos/repository-name`. As a result, you can selectively deploy packages to your edge devices.

A separate package is provided for each of the supported platforms or deployment architectures.

---

**Stop Servers**

1. Change to the following directory:
   ```
cd /opt/sas/viya/home/SASEventStreamProcessingEngine/version/bin
   ```
   For `version`, substitute the version of SAS Event Stream Processing for Edge Computing that is already installed, such as 6.1.

2. To stop the ESP server:
   ```
   dfesp_xml_client -url "http://host:port/SASESP/server/state?value=stopped" -put
   ```
   Replace `host:port` with the host name and port of the machine where the ESP server is running.

3. Stop the metering server if it is running in your environment:
   ```
   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.

---

**Note:** The metering server is not required for SAS Event Stream Processing for Edge Computing, but it is required for SAS Event Stream Processing on Linux or Windows.

---

**Upgrade SAS Software**

You have already downloaded the packages into your mirror repository using Mirror Manager. Now, you can install these packages on your target machine.

1. To install the packages that you have downloaded into your repository, run the upgrade commands that are appropriate for your operating system.
You can install a subset of the software that you have downloaded. To install selected packages, run the upgrade commands for the packages that you want.

2 Locate the license file that you previously extracted from the ZIP file that was attached to your SOE and was saved.

3 Copy the license file to the license directory:

```
sudo cp license.txt /opt/sas/viya/home/
SASEventStreamProcessingEngine/version/etc/license/
```

**Note:** The file name `license.txt` is required. For `version`, substitute the version of SAS Event Stream Processing for Edge Computing that is already installed, such as 6.1.

4 On the machine where you installed SAS Event Stream Processing for Edge Computing, run the following command to start the ESP server:

```
$DFESP_HOME/bin/dfesp_xml_server -http 9900
```

The `-http` argument runs the ESP server as a factory server that supports the creation of projects. In the example, port 9900 is used for HTTP administration requests.

5 The following INFO message is displayed:

```
Access control disabled (could not open permissions.yml, error: file not found)
```

The file that is referenced is required only to enable access control on the ESP server. You can ignore this message.

---

**Merge Configuration Changes into New File**

In “Prepare to Upgrade” on page 38, you were instructed to compare versions of SAS Event Stream Processing configuration files and save a file reflecting the differences between them. Now merge the newer information in the configuration files 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.

1 Open the file that contains the results of the diff commands that you saved in “Prepare to Upgrade” on page 38.

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`. 

---

**Upgrading Your Software**

41
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.

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

Clean up unnecessary files in order to avoid clashes. Change directories:

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

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 only created during an upgrade if customizations are detected.

After the software has been upgraded, validate the deployment. See Chapter 6, “Validating the Deployment,” on page 31.

If you deployed a subset of the available software packages, you can safely delete any packages that are not included in the /analytics, /astore, or /gpu subfolders of your mirrored package repository after you have completed the upgrade process.

### 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 using the same software order and the same tools that you used to perform the initial installation.

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

#### Prepare to Update

These steps apply to most deployments of SAS Event Stream Processing for Edge Computing. To update a deployment that uses Docker, skip to Step 4.

To prepare to update your deployment:
1 Change to the following directory:

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

For `version`, substitute the version of SAS Event Stream Processing for Edge Computing that is already installed, such as 6.1.

2 Stop the ESP server:

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

Replace `host:port` with the host name and port of the machine where the ESP server is running.

3 Stop the metering server if it is running in your environment:

```bash
    dfesp_xml_client -url "http://host-name: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.

---

**Note:** The metering server is not required for SAS Event Stream Processing for Edge Computing, but it is required for SAS Event Stream Processing on Linux or Windows.

---

4 You should already have SAS Mirror Manager and the Extensions package from the initial software deployment. For more information about how to download this software again, see “Create a Mirror Repository” on page 12. For a Docker deployment, see “Create and Populate a Mirror Registry” on page 20.

5 To use SAS Mirror Manager with the Extensions to set up and populate your mirror repository:

```bash
    edge_mirror.sh list-of-repositories
```

The list of software repositories to which you are entitled is included in the software order email (SOE) that you received from SAS. For `list-of-repositories`, substitute the names of the software repositories that you want to update on the target machine. Separate the names of multiple repositories with spaces.

You must include at least one repository in the command. Here is an example:

```bash
    edge_mirror.sh sas-espedge-100.0.0-aarch64_ubuntu_linux_16-apt-latest
```

The `edge_mirror` script uses SAS Mirror Manager to download the specified software to repositories on your machine.

The packages are placed in separate subdirectories under `espedge_repos \repository-name`. As a result, you can selectively deploy packages to your edge devices.

A separate package is provided for each of the supported platforms or deployment architectures.
Update SAS Software

To update your deployment, a few additional steps are required. You have already refreshed the packages in your mirror repository using SAS Mirror Manager. Now, you can copy these packages to your target machine.

For a Docker deployment, redeploy a new container based on the image that you downloaded to your private registry.

1. To update the software on the target system with the newer software in your mirror repository, run the package update commands that are appropriate for your operating system.

2. On the machine where you installed SAS Event Stream Processing for Edge Computing, run the following command to start the ESP server:

   ```bash
   $DFESP_HOME/bin/dfesp_xml_server -http 9900
   ```

3. The following INFO message is displayed:

   Access control disabled (could not open permissions.yml, error: file not found)

   The file that is referenced is required only to enable access control on the ESP server. You can ignore this message.

4. Start the metering server if it was running in your environment.

   Note: The metering server is not required for SAS Event Stream Processing for Edge Computing, but it is required for SAS Event Stream Processing on Linux or Windows.

5. After the software has been updated, validate the deployment. See Chapter 6, “Validating the Deployment,” on page 31.
Uninstalling SAS Event Stream Processing

Note: If you are using Docker to deploy SAS Event Stream Processing for Edge Computing, you can skip this section.

Use the appropriate tool for your platform in order to uninstall your SAS software. The steps provided are specific to the supported Linux platforms.

Uninstall from Red Hat Enterprise Linux or Yocto Linux

This example illustrates how you can remove the SAS Event Stream Processing software from most varieties of Linux. Perform the following steps to uninstall SAS Event Stream Processing for Edge Computing from Red Hat Enterprise Linux or Yocto Linux machines:

1. Remove the license file:
   ```bash
   sudo rm /opt/sas/viya/home/SASEventStreamProcessingEngine/6.2/etc/license/license.txt
   ```

2. Uninstall the SAS Event Stream Processing software. Run the following command:
   ```bash
   rpm -e $(rpm -qg SAS)
   ```

   This command also removes any SAS products that were deployed on the system in addition to your SAS Event Stream Processing software. It also removes all residual packages, including configuration files.
Uninstall from SUSE Linux

This example illustrates how you can remove the SAS Event Stream Processing software from SUSE Linux.

1. Remove the license file:
   
   ```
   sudo rm /opt/sas/viya/home/SASEventStreamProcessingEngine/6.2/etc/license/license.txt
   ```

2. Use Zypper to uninstall the SAS Event Stream Processing software. Run the following command:
   
   ```
   sudo zypper remove $(rpm -qg SAS)
   ```

   This command also removes any SAS products that were deployed on the system in addition to your SAS Event Stream Processing software. It removes all residual packages, including configuration files. However, you will not see feedback from the command to notify you that the software has been removed.

Uninstall from Ubuntu Linux

This example illustrates how you can remove the SAS Event Stream Processing software from Ubuntu Linux.

1. Remove the license file:
   
   ```
   sudo rm /opt/sas/viya/home/SASEventStreamProcessingEngine/6.2/etc/license/license.txt
   ```

2. Uninstall the SAS Event Stream Processing software by running the following command:

   ```
   sudo apt-get remove --purge $(sudo dpkg -l | grep SAS | grep "^rc" | awk '{print $2}' | tr \"\n\" ' ')
   ```

   This command also removes any SAS products that were deployed on the system in addition to your SAS Event Stream Processing software. It also removes all residual packages, including configuration files.