



SAS[®] Event Stream Processing for Edge Computing 5.2: Deployment Guide

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

SAS® Event Stream Processing for Edge Computing 5.2: Deployment Guide

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

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

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

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

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

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

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

January 2019

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

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

5.2-P1:dplyedge0phy0lax

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	4
Java	5
GPU Requirements	6
User Accounts	6
Encryption and Authentication Options	7
Chapter 3 / Pre-installation Tasks	9
Save the Required Files	9
Create a Mirror Repository	9
Chapter 4 / Installation on Linux	13
Installation Overview	13
Complete the Installation	13
(Optional) Enable GPU Functionality	14
Chapter 5 / Installation with Docker	17
Installation Overview	17
Copy the Certificates for a Deployment with Docker	17
Create and Populate a Mirror Registry	18
Next Steps	20
Mount a Persistent Storage Volume and Apply the License	20
Encode and Apply the Product License	21
Change Default Settings	21
Enable TLS for the Docker Container	23
Stop a Container	24
Chapter 6 / Validating the Deployment	25
Verify the Installed Packages	25
Chapter 7 / Next Steps	29
Directory Structure and Permissions	29
Product Documentation	29
Chapter 8 / Managing Your Software	31
Overview	31
Upgrading Your Software	32
Updating Your Software	34
Chapter 9 / Uninstalling SAS Event Stream Processing	37
Uninstall SAS Event Stream Processing	37

Introduction

<i>Steps for a Successful Deployment</i>	1
Before You Begin	1
Step 1 — Prepare for the Deployment	1
Step 2 — Perform the Deployment	2
Step 3 — Validate the Deployment	2
<i>Contact SAS Technical Support</i>	2

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.

- 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.
- 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.

Step 1 — Prepare for the Deployment

- 1 Perform one of the following tasks:
 - To upgrade or update an existing deployment, go directly to [“Managing Your Software” on page 31](#).
 - To deploy a new instance of the software, continue with the following steps.
- 2 Go to [“System Requirements” on page 3](#) to learn about requirements for hardware, software, user accounts, and more.
- 3 Go to [“Pre-installation Tasks” on page 9](#) to prepare the environment before you deploy the software.

Step 2 — Perform the Deployment

Perform one of the following:

- Go to “[Installation on Linux](#)” on page 13 to complete the installation on a Linux machine.
- Go to “[Installation with Docker](#)” on page 17 to use the Docker image provided by SAS to create Docker containers that run the software.

Step 3 — Validate the Deployment

Go to “[Validating the Deployment](#)” on page 25 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.

System Requirements

Hardware Requirements	3
Hardware Requirements for SAS Event Stream Processing	3
Docker Requirements	4
Hardware Requirements to Support the Docker Container	4
Operating System Requirements	4
Supported Operating Systems	4
Linux Prerequisites	5
Additional Requirements for the ESP Server	5
SAS Support for Alternative Operating Systems	5
Java	5
GPU Requirements	6
User Accounts	6
Encryption and Authentication Options	7

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 2.1 Hardware Requirements for SAS Event Stream Processing

Item	Recommended Level*
CPU	Dual or quad-core x86_64 compatible processor. 64-bit ARMv8 chipsets are supported.
Memory	4 GB of available RAM
Disk Space	400 MB - 1.5 GB free space for the installation (depending on the optional components that were selected in the software order)

*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

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

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.

- Debian Linux 9 for 64-bit x86_64 and 64-bit ARMv8 chipsets
- 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 16.04

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 (Aarch64) chipsets.

- 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 5.x libraries for x86_64 chipsets have been built using gcc-4.4.7-16 on Red Hat Enterprise Linux Server 6.7 using libc-2.12.so, libstdc++.so.6.0.13, and libgcc_s-4.4.7-20120601.so.1.

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

SAS Support for Alternative Operating Systems

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

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 offers optional support for GPUs. To use a GPU for high-powered analytics calculations, you can install SAS Event Stream Processing for Edge Computing on one of the supported architectures.

The following components are required to enable GPU support:

- GPU with NVIDIA Pascal or Volta architecture
- 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 TX2 embedded computing module is required. NVIDIA JetPack L4T 3.1 for Ubuntu Linux is also required.

Note: On NVIDIA Jetson TX2, 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.

- Ubuntu Linux 14.04 (64-bit)

A few post-deployment steps are required to enable GPU functionality. For more information, see [“\(Optional\) Enable GPU Functionality” on page 14](#).

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:

Account Name and Group	Parameters	Purpose
sas; member of sas group	<p>Non-login service account without user restrictions.</p> <p>No password; can add password after installation if desired.</p> <p>Password does not expire.</p> <p>The default user name is required until the installation is complete. Any post-installation changes to this account do not prevent future software updates.</p>	<p>Required for the installation.</p> <p>The installation process sets user and group ownership permissions on all of the installation files. This user must exist to enable ownership.</p> <p>After the installation has completed, this user account enables required components to run.</p>

Sudoers privileges are not required after the installation to run SAS Event Stream Processing for Edge Computing. 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.

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 23](#).

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

Authentication and encryption apply to the following ESP server APIs:

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

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

Pre-installation Tasks

<i>Save the Required Files</i>	9
<i>Create a Mirror Repository</i>	9
SAS Mirror Manager Requirements	10
Create and Populate a Mirror Repository	10

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 [“Installation with Docker” on page 17](#) 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.

Note: The process for creating a mirror repository for SAS Event Stream Processing 5.2 is different from the one used in previous versions. If you are familiar with earlier versions of the software, you should not assume any similarities with the process used by those versions.

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`
- `/gpu`

The last three 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” on page 13](#) section of this guide.

Installation on Linux

<i>Installation Overview</i>	13
<i>Complete the Installation</i>	13
Apply the License	13
Set the Required Environment Variables	14
Start the Software	14
<i>(Optional) Enable GPU Functionality</i>	14

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/5.2/etc/license/
```

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

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/5.2
export LD_LIBRARY_PATH=$DFESP_HOME/lib:/opt/sas/viya/home/SASFoundation/sasexe
```

Start the Software

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.

- 2 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 [“Validating the Deployment” on page 25](#).

(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 `sasgpd` 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 with Docker

<i>Installation Overview</i>	17
<i>Copy the Certificates for a Deployment with Docker</i>	17
<i>Create and Populate a Mirror Registry</i>	18
<i>Next Steps</i>	20
<i>Mount a Persistent Storage Volume and Apply the License</i>	20
<i>Encode and Apply the Product License</i>	21
<i>Change Default Settings</i>	21
Default Container Settings	21
<i>Enable TLS for the Docker Container</i>	23
<i>Stop a Container</i>	24

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 9](#). 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. In addition, 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 registry-name:2
```

For *registry-name*, substitute the name of your private registry. The default name is **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 9.

- 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:

Note: Specify each command on a single line. Multiple lines are used here to improve readability. For the repository and package names, substitute the names that are listed in the output from the Mirror Manager `list remote docker tags` command. Each name includes a multiple-digit numerical identifier.

```
docker pull ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-docker-latest/
sas-espedge:5.2.0-numerical-identifier

docker inspect ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-docker-latest/
sas-espedge:5.2.0-numerical-identifier
```

- 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:

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

```
docker pull ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-docker-latest/
sas-espedge-analytics:5.2.0-numerical-identifier

docker inspect ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-docker-latest/
sas-espedge-analytics:5.2.0-numerical-identifier
```

- 13 Tag the base image:

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

```
docker 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:

```
docker tag ses.sas.download/espedge-100.0.0-x64_redhat_linux_7-docker-latest/
sas-espedge-base:5.2.0-numerical-identifier localhost:5000/sas-espedge-base
```

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

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

- 15 Add the base image to your private 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.

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

```
docker 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 18, 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 your site’s accepted software and processes 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 20.
- 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 21.

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:

```
docker run -v path-to-license-on-host:/data private-registry/host-name:port/sas-esp-edge-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:

```
export ESP_LICENSE=$(base64 ./license.txt)
```

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

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

```
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.

Change Default Settings

When the container starts up, it uses the following default settings for the environment variables that enable SAS Event Stream Processing for Edge Computing operations:

Table 5.1 Default Container Settings

Environment Variable	Value	Description
DFESP_CONFIG	<code>/opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default</code>	<p>Enables advanced options.</p> <p>Replaces the default location for configuration values. To use non-default logs, metering, and MAS stores, you must copy files from the default location to the full path that is specified for this variable. When this variable is set, the ESP servers use the settings that are specified in the alternative location.</p> <p>The file <code>esp-logger.xml</code> must also be stored in the specified location.</p> <p>Note: The default location does not exist until the container has been started.</p>
ESP_HTTP_PORT	31415	Specifies the port for the ESP server's HTTP REST API. The value cannot exceed 65535.

Environment Variable	Value	Description
ESP_LICENSE	None	Specifies the Base64-encoded content of the product license. By default, SAS Event Stream Processing for Edge Computing looks for license.txt in an external volume that is mounted as <code>/data</code> . If it is found, the license file is then copied to <code>\$DFESP_HOME/etc/license</code> . The file name license.txt is required.
ESP_LOGGING_LEVEL	None	Sets the initial logging level. Supported logging levels are <code>off</code> <code>trace</code> <code>debug</code> <code>info</code> <code>warn</code> <code>error</code> <code>fatal</code> .
ESP_MODEL_PATH	None	Specifies the URL to the XML model. Use the URL notation for files, specifying the full path to the XML model (<code>file://path</code>). To specify a secure HTTP server, use <code>ports</code> . An example is <code>http 8091s</code> .
ESP_PUBSUB_PORT	31416	Specifies a port for the ESP server's publish and subscribe actions.
SASLOGON_SERVERPORT	None	Specifies the fully qualified domain name and port of the SAS Logon server for your environment in the format <code>fully-qualified-domain-name:port</code> .

You can change the default settings for these environment variables. The Docker entry point writes the value of each variable into a configuration file in the container for the ESP server to use.

For more information about the ESP_LICENSE variable, see [“Encode and Apply the Product License” on page 21](#).

For the commands that set SAS Event Stream Processing for Edge Computing environment variables, use the following substitutions in the examples that are provided:

- For *fully-qualified-domain-name*, substitute the URL to your ESP server.
- For *port*, substitute the port where the ESP server listens.
- For *private-registry*, substitute the name of your private registry.
- For *XML-namespace*, substitute the namespace of your ESP server schema.
- For *tag*, substitute the Docker tag that is assigned to the container.

To set values for the other environment variables that affect SAS Event Stream Processing settings:

- 1 You can use the DFESP_CONFIG variable to specify an alternative location for your configuration data, such as a `connectors.excluded` file. To change the default location for configuration files, first update the configuration file on the local host. Then mount the alternative directory when you start Docker. Here is an example:

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

```
docker run -e DFESP_CONFIG=/config -v path-to-alternative-configuration-directory-on-host:/config:
/config private-registry/host-name:port /sas-espedge-base:tag
```

- 2 You can use the ESP_HTTP_PORT variable to set the HTTP port for the XML server's HTTP REST API. In the following example, it is set to \$ESP_HTTP_PORT:

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

```
docker run -e ESP_HTTP_PORT=$ESP_HTTP_PORT\private-registry/XML-namespace
/sas-espedge-base:tag
```

- 3 You can use the `ESP_LOGGING_LEVEL` variable to set the logging level for ESP server logs. To set the level to one of the options that are listed in the table:

Note: This example shows the debug option.

```
docker run -e ESP_LOGGING_LEVEL=debug private-registry/XML namespace/sas-espedge-base: tag
```

- 4 You can use the `ESP_MODEL_PATH` variable to set the location of the XML model. In the following example, it is set to `$ESP_MODEL_PATH`:

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

```
docker run -e ESP_MODEL_PATH=$ESP_MODEL_PATH\private-registry/XML-namespace
/sas-espedge-base:tag
```

- 5 You can use the `ESP_PUBSUB_PORT` variable to set the HTTP port for the XML server's publish and subscribe operations. In the following example, it is set to `$ESP_PUBSUB_PORT`:

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

```
docker run -e ESP_PUBSUB_PORT=$ESP_PUBSUB_PORT\private-registry/XML-namespace
/sas-espedge-base:tag
```

- 6 You can use the `SASLOGON_SERVERPORT` variable to set the location where the SAS Logon server is listening.

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

```
docker run -e SASLOGON_SERVERPORT=fully-qualified-domain-name:port private-registry/XML-namespace
/sas-espedge-base:tag
```

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 in a security configuration file named **security-properties.yml**.

For more information about the security configuration file, see [Security Configuration File](#) in *SAS Event Stream Processing: Security*.

- 1 Using your preferred text editor, verify that the following configuration is included in the `security-properties.yml` file:

```
security:
  pubsub_ssl_enabled: false
  http_ssl_enabled: true
  trust_selfsigned: true
  server:
    auth: null
    server_cert_file: server.pem
  client:
    auth: null
    ca_cert_file: cert.pem
```

- 2 Save the file in the `/data` directory of the external volume that you have mounted. For more information, see [“Mount a Persistent Storage Volume and Apply the License”](#) on page 20.
- 3 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.

- 4 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.

- 5 Copy these three security files to the directory where they will be available to the Docker container.
- 6 Set three environment variables to specify the location of the security files:

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

```
docker run -v location-of-files/:/data -p 127.0.0.1:31415:31415 -e ESP_LICENSE=$ESP_LICENSE
-e "ESP_MODEL_STRING=file:///data/model.xml" -e DFESP_SEC_YAML="/data/security-properties.yml"
-e ESP_SERVER_CERT="/data/server.pem" -e ESP_CLIENT_CACERT="/data/cert.pem"
```

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

- 7 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.

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	25
Validate the SAS RPM Packages	25
Validate the SAS Packages on Ubuntu	27

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-5.1.0-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:

```
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:

```
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:

```
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, 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:

```
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:

```
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:

```
??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.

Next Steps

<i>Directory Structure and Permissions</i>	29
<i>Product Documentation</i>	29

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.

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 5.2: Overview*, which provides an introduction to product features and explains how to proceed with creating event stream processing models and incorporating them into applications.

Managing Your Software

Overview	31
What Is an Upgrade?	31
What Is an Update?	31
Upgrading Your Software	32
Overview	32
Prepare to Upgrade	32
Stop Servers	33
Upgrade SAS Software	33
Updating Your Software	34
Overview	34
Prepare to Update	34
Update SAS Software	35

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.

The installation process for previous releases of SAS Event Stream Processing for Edge Computing has changed in version 5.2. See “[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 32 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.

Prepare to Upgrade

The process to install SAS Event Stream Processing for Edge Computing has changed with the 5.2 version of the software. 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 Verify that your environment meets the requirements that are listed in [“System Requirements” on page 3](#). System requirements for RAM, CPU, and disk space might change with each release.

- 2 (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:

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

- 3 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.

- 4 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**.
- 5 Download SAS Mirror Manager from the [SAS Mirror Manager download site](#) to the machine where you want to create your mirror repository.
- 6 Uncompress the downloaded file.
- 7 Download the SAS Event Stream Processing for Edge Computing Extensions package, `sas-edge-extension.tgz`, from the [Mirror Extension Website](#).
- 8 Uncompress `sas-edge-extension.tgz` in the same directory where you uncompressed SAS Mirror Manager.
- 9 To refresh your mirror repository with the newer versions of the SAS software:

```
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/5.2/bin
```

- 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/5.2/etc/license/
```

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

- 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.

- 6 Start the Metering server if it was running in your environment.

For more information about enabling metering, see [Using the Metering Server in SAS Event Stream Processing: Using the ESP Server](#).

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.

- 7 After the software has been upgraded, validate the deployment. See [“Validating the Deployment” on page 25](#).

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:

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

- 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.

- 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 9](#). For a Docker deployment, see [“Create and Populate a Mirror Registry” on page 18](#).
- 5 To use SAS Mirror Manager with the Extensions to set up and populate your mirror repository:

```
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:

```
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:

```
$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.

For more information about enabling metering, see [Using the Metering Server](#) in *SAS Event Stream Processing: Using the ESP Server*.

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 [“Validating the Deployment”](#) on page 25.

Uninstalling SAS Event Stream Processing

Uninstall SAS Event Stream Processing	37
Uninstall from Red Hat Enterprise Linux or Yocto Linux	37
Uninstall from SUSE Linux	37
Uninstall from Ubuntu Linux	38

Uninstall 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:

```
sudo rm /opt/sas/viya/home/SASEventStreamProcessingEngine/5.2/etc/license/license.txt
```

- 2 Uninstall the SAS Event Stream Processing software. Run the following command:

```
rpm -e $(rpm -qq 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/5.2/etc/license/license.txt
```

- 2 Use Zypper to uninstall the SAS Event Stream Processing software. Run the following command:

```
sudo zypper remove $(rpm -qq 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/5.2/etc/license/license.txt
```

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

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

```
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.