



SAS[®] Event Stream Processing: Using SAS[®] Event Stream Manager

2026.08

This document might apply to additional versions of the software. Open this document in [SAS Help Center](#) and click on the version in the banner to see all available versions.

<i>SAS Event Stream Manager Overview</i>	3
Understanding SAS Event Stream Manager	3
Access SAS Event Stream Manager	8
Using the Welcome Page	9
Install SAS Event Stream Manager as a Progressive Web App	9
<i>Understanding the User Interface</i>	10
Pages	10
Panels	12
Tiles	12
Windows	13
Toolbars	14
Arrange Information in Tables	15
<i>Working with Deployments</i>	16
Overview to Deployments	16
Create a Deployment	16
Open a Deployment	17
Edit Deployment Properties	17
Change the Production Status of a Deployment	18
Delete a Deployment	18
<i>Working with ESP Servers</i>	18

ESP Server That Runs in a Kubernetes Cluster	18
ESP Server That Runs on an Edge Server	19
Add ESP Servers That Run on Edge Servers	19
Edit an ESP Server That Runs on an Edge Server	22
Remove ESP Servers	23
Overview to Versioning	24
Managing Project Versions That Are Published Using File-Based Versioning	25
Overview of File-Based Versioning	25
View a Project	25
Manage Project Versions	27
Import a Project	31
Managing Project Versions That Are Published Using SAS Viya Platform Services	32
Overview of Managing Project Versions That Use SAS Viya Platform Services	32
View a Project	32
Delete a Project	33
Change the Production Status of a Project	34
Deploying Projects	34
Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster	34
Stop a Project That Is Running on an ESP Server That Runs in a Kubernetes Cluster ...	37
Load and Start a Project on an ESP Server That Runs on an Edge Server	37
Stop and Unload a Project That Is Running on an ESP Server That Runs on an Edge Server	39
Monitoring a Deployment	40
Monitor All Deployments	40
Monitor a Specific Deployment Whose Type Is "Cluster"	42
View Logs for a Deployment Whose Type Is "Cluster"	47
View Performance Information for a Deployment Whose Type Is "Cluster"	49
Monitor a Specific Deployment Whose Type Is "Edge"	51
View the ESP Server Log for a Deployment Whose Type Is "Edge"	53
View Sample Data	54
Update SAS Micro Analytic Service Modules	56
Monitor Unassigned ESP Servers	58
Monitor Jobs	58
Delete a Job	60
Rerun a Job	61
Stop a Running Job	61
Working with Project Containers	62
Project Containers	62
Creating Project Containers	64
Deploying Project Containers	70
Working with Job Templates	73
Job Template	73
Create a Job Template	74
Import a Job Template	75
Edit a Job Template	75
Deploy a Job Template	76
Export a Job Template	77
Delete a Job Template	77
Working with Filters	78
Filters	78

Working with Filters for a Specific Deployment	78
Managing Filters	80
Filter Expression Syntax	81
Working with Metering Servers	82
Metering	82
Add a Metering Server	82
Monitor a Metering Server	83
Managing Settings	84
Overview of Settings	84
Settings Window	84
ESP Settings Page	86
Global Settings	87
Settings for the User Session	93
ESP Operator	95
Tools	96
Examples	100
Overview of SAS Event Stream Manager Examples	100
Filter Example	101
Activity Tracker Example	103
GitHub Actions Workflow Example	116

SAS Event Stream Manager Overview

Understanding SAS Event Stream Manager

What Is SAS Event Stream Manager?

SAS Event Stream Manager is a web-based client that enables you to manage your SAS Event Stream Processing environment.

You can use SAS Event Stream Manager to perform the following tasks:

- deploy SAS Event Stream Processing projects into production environments and test environments
- monitor the health of your deployments, ESP servers, and running projects
- administer your deployments and manage change
- monitor your SAS Event Stream Processing metering servers

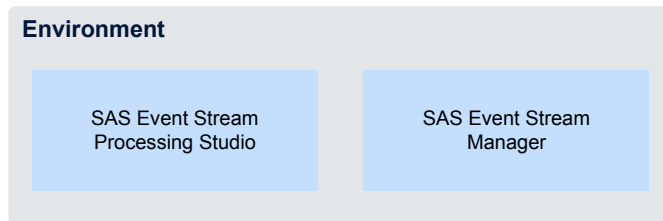
You can use SAS Event Stream Processing Studio to create the projects that you deploy to ESP servers using SAS Event Stream Manager.

SAS Event Stream Manager Instances

By default, a single instance of SAS Event Stream Manager is deployed. This instance supports the management of test and production environments in the same physical environment.

The following example shows a SAS Event Stream Processing environment that contains an instance of SAS Event Stream Processing Studio and an instance of SAS Event Stream Manager.

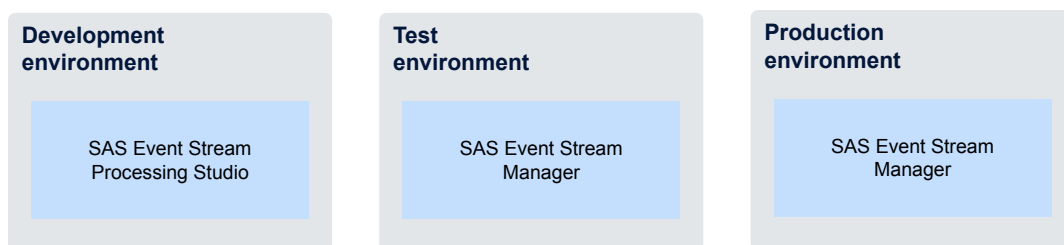
Figure 1 *An Example of an Environment with a Single SAS Event Stream Manager Instance*



However, it is recommended that a separated system with a minimum of two instances of SAS Event Stream Manager is deployed. A *separated system* contains separate test and production environments, with logical separation (separate namespaces) or physical separation (separate clusters) between the environments. Each environment has its own SAS Event Stream Manager instance.

The following example shows a development environment with an instance of SAS Event Stream Processing Studio, a test environment with an instance of SAS Event Stream Manager, and a production environment with an instance of SAS Event Stream Manager.

Figure 2 *An Example of a Separated System*



Here are some benefits of using a separated system:

- Reduced operational risk. In a separated system, it can be easier to prevent untested or experimental projects from affecting live services.
- Protection of data and access. You can limit who can do what in each environment. For example, you give developers access to work in the development and test environments, and allow only designated users to access the production environment. You can keep real customer data out of development and test, and use only masked or synthetic data in those environments.
- Support for reliable promotion and recovery. You promote changes in one direction: from development to test, and then to production. If a test fails, the failure does not affect production users. Rolling back a change is faster and safer when environments are separate.

When you are working with project containers, in addition to promoting project containers to higher promotion levels, you can demote project containers to a lower promotion level. A *project*

container is a lightweight container image that complies with the Open Container Initiative (OCI) and is designed to execute projects in a self-contained, immutable container without external software dependencies. For more information, see [“Working with Project Containers”](#).

- Alignment with good security practice. Separation of development, test, and production environments is a commonly recognized control in information security frameworks. This separation supports clear boundaries, least-privilege access, and change control.

Here are some examples of typical use cases for separation of development, test, and production environments:

- Regulated workloads. In industries such as financial services or health care, teams use separated environments to meet audit expectations for access control, change management, and data protection.
- Shared platforms. Central IT hosts multiple business projects in SAS Event Stream Processing. Separate environments keep team experiments from affecting other lines of business.
- Deploying the same assets to edge and cloud. Use a project container to facilitate the deployment of assets to multiple environments. You validate project containers in test, and then promote the same versioned artifacts to production in Kubernetes clusters or on edge devices.

The creation of additional SAS Event Stream Manager instances takes place when SAS Event Stream Processing is deployed. For more information about how SAS Event Stream Manager was deployed in your organization, contact your system administrator.

For more information about how SAS Event Stream Manager instances are created when SAS Event Stream Processing is deployed with other SAS Viya products, see [“Creating a Separated System” in SAS Event Stream Processing: Using SAS Event Stream Processing in a Kubernetes Environment](#). For more information about how SAS Event Stream Manager instances are created when standalone SAS Event Stream Processing is deployed, see [“Configure a Separated System for SAS Event Stream Manager” in SAS Event Stream Processing: Using SAS Event Stream Processing in a Kubernetes Environment](#).

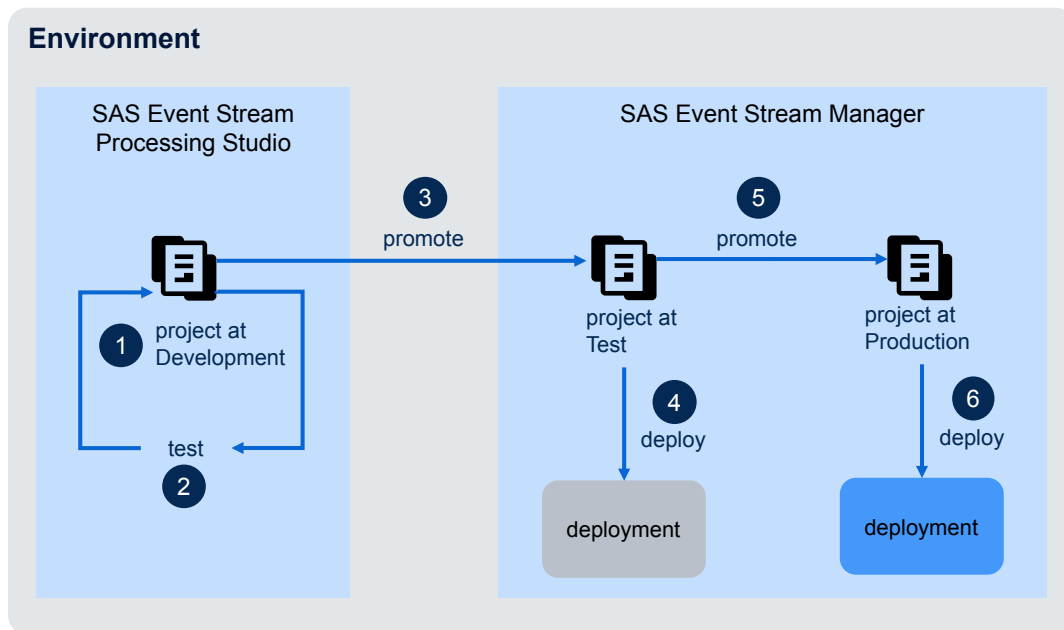
Overview of Using SAS Event Stream Manager to Deploy Projects

One of the main tasks that you can perform in SAS Event Stream Manager is to deploy SAS Event Stream Processing projects to SAS Event Stream Processing environments. This topic provides an outline of how a project that is created and tested in SAS Event Stream Processing Studio is made available to SAS Event Stream Manager and then deployed to deployments. In SAS Event Stream Manager, a *deployment* is a logical grouping of assets as a single unit for life cycle management and monitoring.

Deploying a Project When There Is a Single SAS Event Stream Manager Instance

The following diagram shows how a project is deployed when there is a single SAS Event Stream Manager instance.

Figure 3 Deploying a Project When There Is a Single SAS Event Stream Manager Instance



- 1 Use SAS Event Stream Processing Studio to create a project, import a project, or install an example project.
- 2 Run the project in test mode to verify that the model works as intended.
- 3 Promote the project from the Development promotion level to the Test promotion level. When SAS Event Stream Processing Studio and SAS Event Stream Manager are running in the same environment, promoting the project to Test also makes the project available in SAS Event Stream Manager.

Promotion levels help prevent projects that are meant for testing from accidentally being used for production (that is, being used in a live environment).

- 4 In SAS Event Stream Manager, deploy the project to a deployment that you are using for testing purposes, to check how the project performs. In SAS Event Stream Manager, a deployment is a logical grouping of assets as a single unit for life cycle management and monitoring.
For example, you can check metrics about consumption of resources in the Kubernetes cluster. If required, at this point you could return to SAS Event Stream Processing Studio to adjust the project and promote another project version to Test.
- 5 Promote the project from the Test promotion level to the Production promotion level.
- 6 Deploy the project to a deployment that you are using for production purposes.

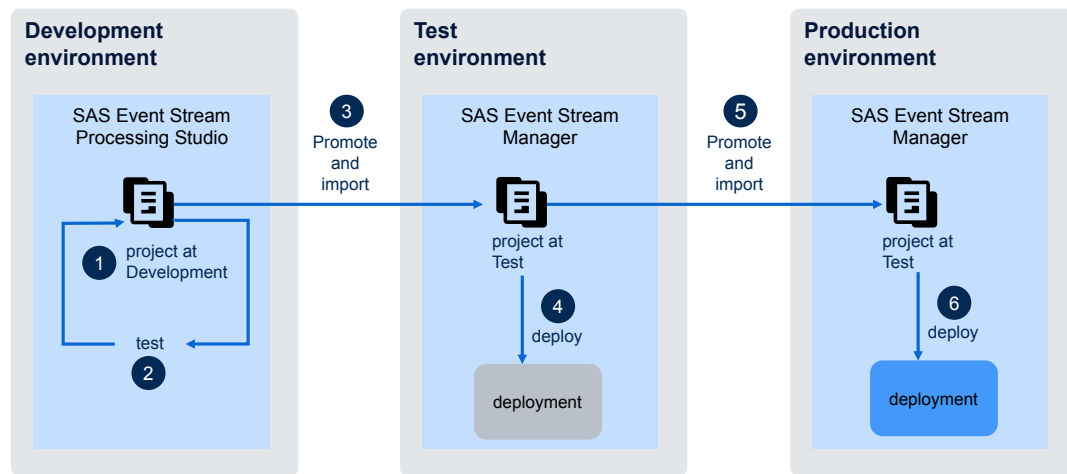
For more information about the steps in SAS Event Stream Processing Studio, see [“Example of Promoting a Project Version from Development to Test” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#). For more information about the steps in SAS Event Stream Manager, see [“Example of Promoting a Project Version from Test to Production”](#).

Deploying a Project in a Separated System

The following diagram shows an example of deploying a project in a separated system. The number of environments and SAS Event Stream Manager instances depends on how SAS Event Stream

Manager was deployed in your organization. For more information, contact your system administrator.

Figure 4 Example of Deploying a Project in a Separated System



- 1 In your development environment, use SAS Event Stream Processing Studio to create a project, import a project, or install an example project.
- 2 Run the project in test mode to verify that the model works as intended.
- 3 Promote the project from the Development promotion level to the Test promotion level, and then import the project to the SAS Event Stream Manager instance that is running in your test environment.
- 4 In SAS Event Stream Manager, deploy the project to check how it performs.
For example, you can check metrics about consumption of resources in the Kubernetes cluster. If required, at this point you could return to SAS Event Stream Processing Studio to adjust the project, promote another project version to Test, and import that project version to SAS Event Stream Manager.
- 5 Promote the project from the Test promotion level to the Production promotion level, and then import the project to the SAS Event Stream Manager instance that is running in your production environment.
- 6 Deploy the project. The project is now running in your production environment.

IMPORTANT By default, project versions are stored in a Git repository that is internal to SAS Event Stream Processing. To pass a project from development to test and then to production in a separated system, the project must be stored in a repository on a remote Git server. For more information, see [“File-Based Versioning and Git” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

If you use SAS Event Stream Manager to deploy a project container in a separated system, the SAS Event Stream Manager instances in the test and production environments must have a connection to the same container registry so that both instances can access the container image. For more information, see [“Add Container Registries”](#).

Production Assets

To prevent assets meant for testing from accidentally being used for production (that is, being used in a live environment), certain assets can be designated as production assets.

For example, when a project has been promoted to the Test promotion level, SAS Event Stream Manager does not regard that project as a production asset. In a separated system, you cannot use a SAS Event Stream Manager instance that is designated as a production instance to deploy that project.

In most cases, SAS Event Stream Manager automatically determines whether an asset is a production asset. For example, if a project has been promoted to the Production promotion level, SAS Event Stream Manager normally regards that project as a production asset.

In the following cases, you manually designate assets as production assets:

- a project XML file that is not part of a project package. For more information, see [“Change the Production Status of a Project”](#).
- a deployment whose type is Edge. A deployment’s type can be Cluster or Edge. When you deploy a project to a deployment whose type is Cluster, an ESP server is automatically created in the cloud. When you deploy a project to a deployment whose type is Edge, you are deploying a project to an ESP server that runs on an edge server (an edge device). SAS Event Stream Manager cannot infer whether the edge device is suitable for production use, so you must supply this information. For more information, see [“Change the Production Status of a Deployment”](#).

Access SAS Event Stream Manager

When SAS Event Stream Processing is deployed with other SAS products, by default only members of the SASAdministrators group can access SAS Event Stream Manager and the Event Stream Manager API. For more information, see [“Broaden Access to SAS Event Stream Manager Example”](#) in *SAS Viya Platform: Access to Functionality*.

This limitation also applies when SAS Event Stream Manager is accessed through other SAS functionality. Here are some examples:

- When you use SAS Model Manager to publish a model into SAS Event Stream Processing, the process creates a project container in SAS Event Stream Manager. For more information, see [“Publish a Model from SAS Model Manager into SAS Event Stream Processing”](#) in *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*. If you do not have access to SAS Event Stream Manager, the publish process fails.
- When you use the [SAS Event Stream Processing Data Source Plug-in for Grafana](#), Grafana contacts SAS Event Stream Manager to discover ESP server connection details. If you do not have access to SAS Event Stream Manager, the plug-in cannot access these details.

If you are using another SAS Event Stream Processing client or another application in the SAS Viya platform, you can access SAS Event Stream Manager by clicking  and selecting **Manage Streaming Analytics**. The list of applications that appears in the applications menu depends on how your system was installed and on your access permissions.

Otherwise, to access SAS Event Stream Manager:

- 1 Open the following URL:

`https://name-of-ingress-host/SASEventStreamManager`

The variable *name-of-ingress-host* specifies the name of the host that is configured for your Kubernetes cluster Ingress.

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

Note: If standalone SAS Event Stream Processing was deployed, your system might not be configured to require users to sign in. In this case, the page for signing in is not displayed. Skip this step.

For information about supported web browsers and whether the device that you use to access SAS Event Stream Manager meets requirements, see [“Client Requirements” in System Requirements for the SAS Viya Platform](#).

Using the Welcome Page

When you access SAS Event Stream Processing Studio or SAS Event Stream Manager for the first time, the **Welcome to SAS Event Stream Processing** page appears. Use the resources on the **Welcome to SAS Event Stream Processing** page to learn about SAS Event Stream Processing and the web-based clients.

The user menu on the application bar of SAS Event Stream Manager provides access to other resources, such as information about what’s new in each release. For more information, see [“Toolbars”](#).

Install SAS Event Stream Manager as a Progressive Web App

You can install SAS Event Stream Manager as a Progressive Web App (PWA). For more information, see [“Using SAS Event Stream Processing Web-based Clients as Progressive Web Apps” in SAS Event Stream Processing: Overview](#).

The ability to install the PWA is not available from within SAS Event Stream Manager. However, you can install the PWA from any other web app and access SAS Event Stream Manager from within the PWA.

To install the PWA:

- 1 Open any other SAS web app in a Chromium-based web browser (such as Chrome).
- 2 Do one of the following:
 - Click the appropriate icon at the end of the address bar, and then select **Install SAS**.

- Open the web browser's **More** menu and select **Install SAS**.

Note: The **Install SAS** option might be located under another menu option. For example, within the Microsoft Edge browser, it is located under **Apps**.

Note: The **Install SAS** option might be located under another menu option. For example, within the Microsoft Edge browser, it is located under **Apps**.

The web app is now installed as a desktop program named SAS. After one web application has been installed as a PWA, all other SAS applications are accessible in the PWA.

To uninstall the PWA:

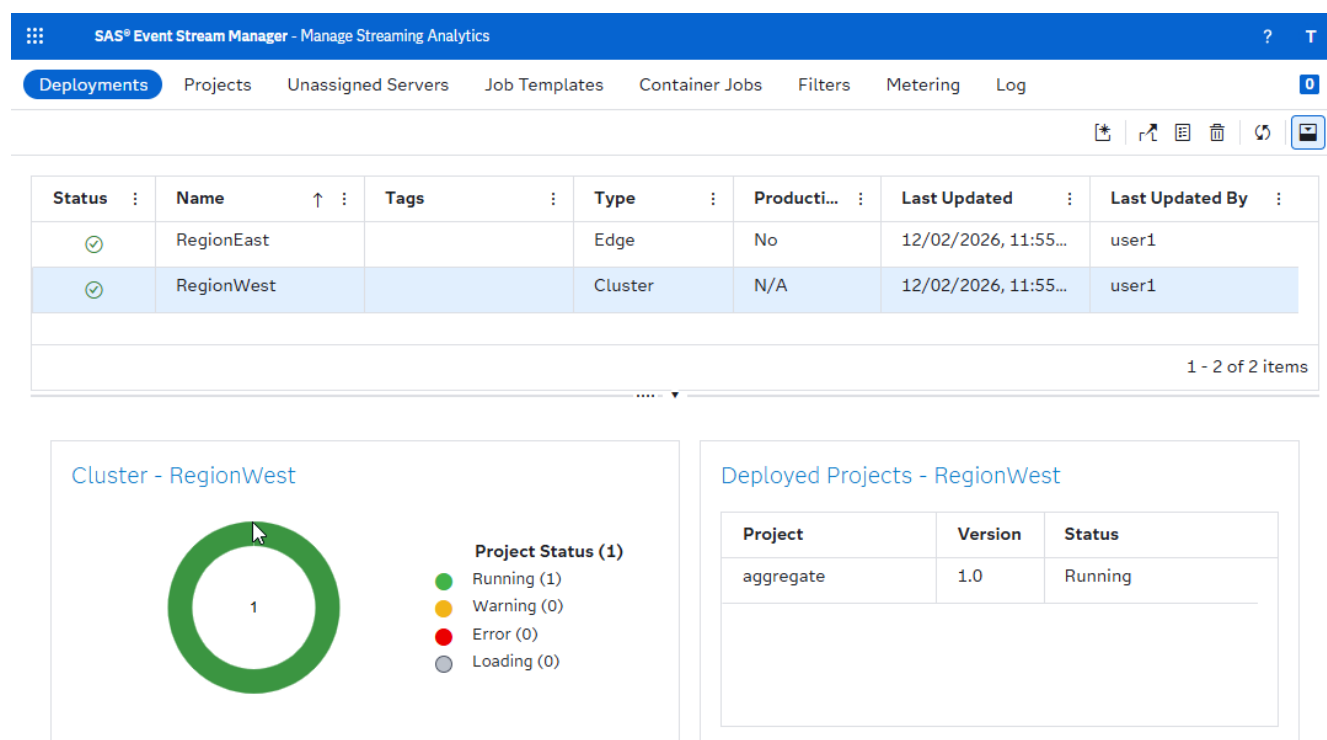
- 1 Open the PWA.
- 2 In the PWA menu, select **Uninstall SAS**.

Understanding the User Interface

Pages

A *page* is the highest-level container in the user interface. All other user interface elements are contained within the confines of a page.

Figure 5 Deployments Page with Two Deployments



SAS Event Stream Manager contains the following main pages:

- The **Deployments** page enables you to create deployments and then monitor and manage them.
- The **Projects** page enables you to view projects that are available in SAS Event Stream Manager.
- The **Unassigned Servers** page enables you to view available ESP servers that have been automatically registered with SAS Event Stream Manager and to register other ESP servers manually. ESP servers that already belong to a deployment are not displayed.
- The **Job Templates** page enables you to create and import job templates, which you can then deploy to create running jobs.
- The **Container Jobs** page lists jobs that have been initiated to create project containers. A *project container* is a lightweight container image.
- The **Filters** page enables you to manage filters that are available in SAS Event Stream Manager.
- The **Metering** page enables you to monitor the metering servers that track usage data.
- The **Log** page displays details of currently running jobs and historical jobs.

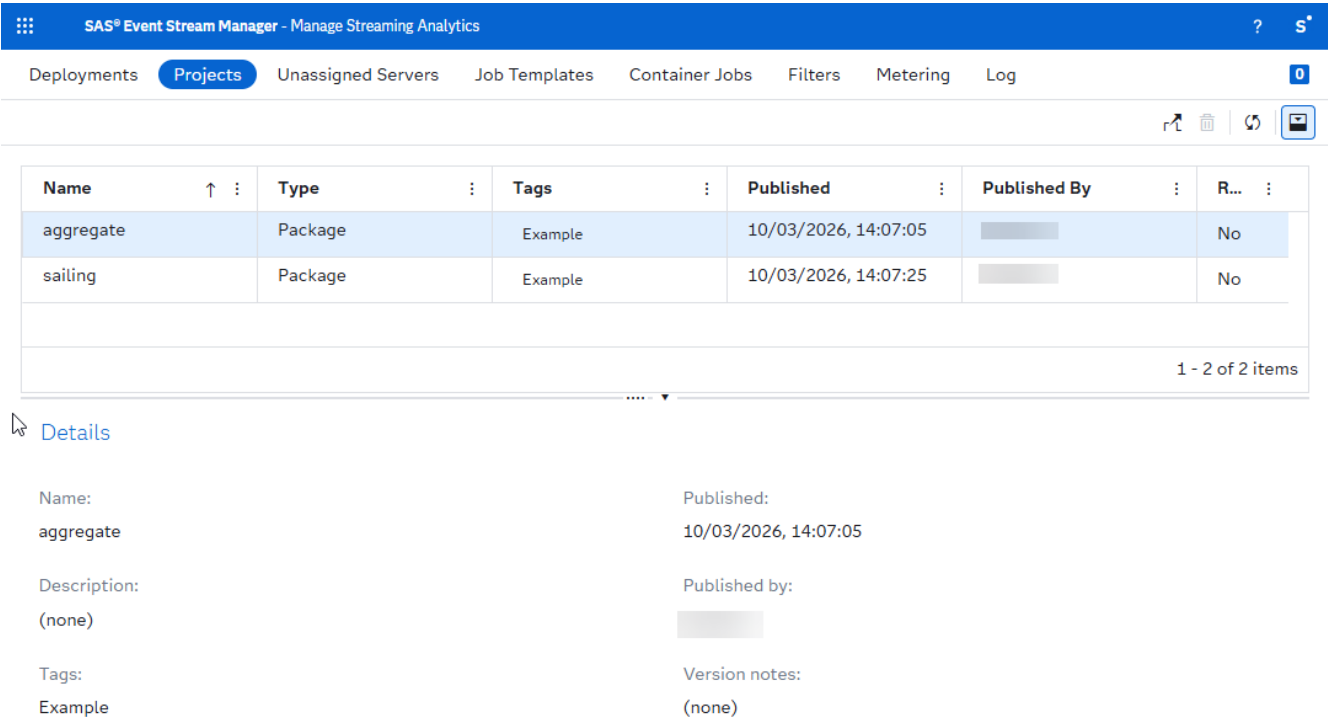
In addition, when you access SAS Event Stream Manager for the first time, the **Welcome** page appears. For more information, see [“Using the Welcome Page”](#).

For more information about the main tasks that you perform using SAS Event Stream Manager, see [“Overview of Using SAS Event Stream Manager to Deploy Projects”](#).

Panes

SAS Event Stream Manager pages contain *panes*. The following figure shows the **Projects** page, which contains a bottom pane with a *tile* called **Details**.

Figure 6 Example of a Page with a Pane



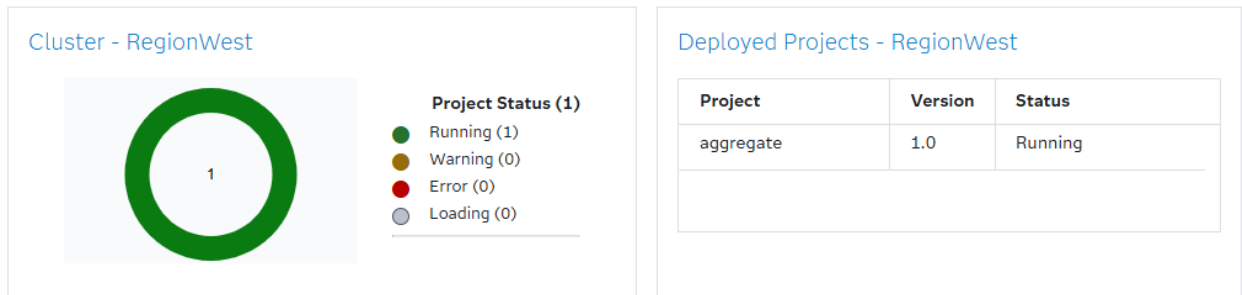
To resize a pane, drag a border that is marked with upward or downward.

To hide a pane, click . In some panes, you can alternatively click ▼ on the border to hide the pane. To display the pane again, click . Alternatively, click ▲ at the bottom of the user interface.

Tiles

A *tile* is a self-contained block of information that resides within a pane or sometimes directly on a page.

Figure 7 Example of Two Tiles within a Pane



Windows

A *window* is a floating user interface element that often appears as a result of a user action. Windows generally provide a means by which to perform an action. Closing a window returns you to the page from which the window was launched. The following figure shows a window that can create a new deployment in SAS Event Stream Manager.

Figure 8 Example of a Window

New Deployment [Close]

Name: *
Deployment1

Description:

Tags:

☐ Production deployment

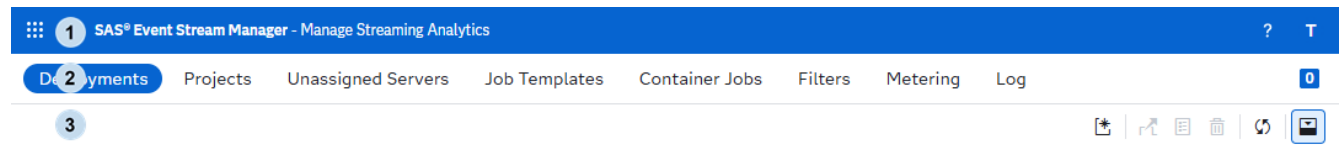
Deployment type: *
Cluster [Dropdown]

OK Cancel

Toolbars

There are three main *toolbars* in SAS Event Stream Manager, as shown in the following figure. For information about each toolbar, see the subsequent table.

Figure 9 SAS Event Stream Manager Toolbars

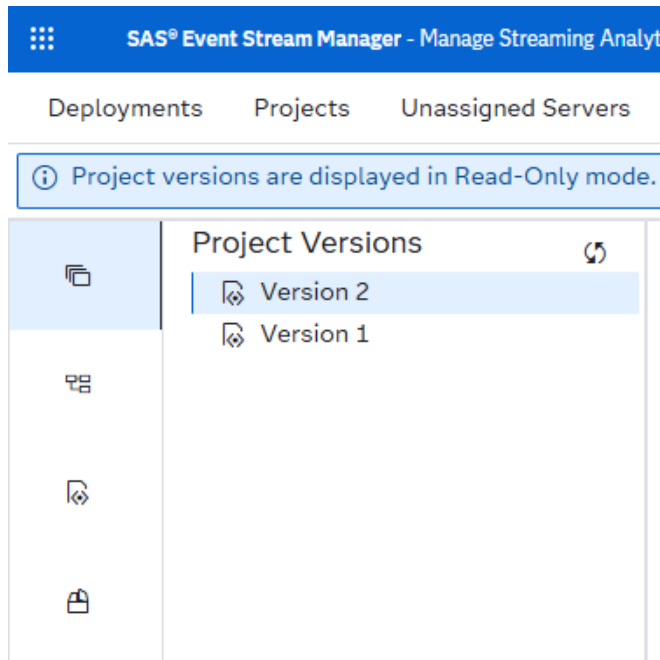


Item Number	Name	Description
1	Application bar	■ Displays the applications menu , which enables you to access other applications on the SAS Viya platform. The presence of the applications menu and the list of applications within it depends on how your system was installed and on your access permissions. ■ Provides access to Help. ■ Displays your user icon, which shows the first character of your name or user ID. Click the user icon to access the following functionality: □ View your display name or your user ID in full. If a display name has not been set, your user ID is displayed by default. If SAS Event Stream Manager has been configured so that you do not need to sign in with a user name and password, your user ID is not displayed. □ View information about what's new in each release, settings, and product information. □ Sign out from SAS Event Stream Manager (if applicable).
2	Navigation bar	■ Provides access to the main SAS Event Stream Manager pages: Deployments, Projects, Unassigned Servers, Job Templates, Container Jobs, Filters, Metering, and Log. ■ Provides access to each deployment, ESP server, project, job template, or job that is currently open. The navigation overflow menu button displays the total number of these pages that are currently open. An example is .
3	Toolbar or tabs	Includes buttons or tabs specific to the open page.

In addition to horizontal toolbars, SAS Event Stream Manager contains vertical toolbars. The following figure shows an example of a vertical toolbar. The toolbar in this example is visible when

you open a project. This toolbar enables you to switch between the project versions, the diagram, the XML code, and the project package.

Figure 10 Example of a Vertical Toolbar



Arrange Information in Tables

Sometimes a large amount of information is displayed in tables. To make it easier to work with a large amount of information, you can arrange this information in different ways.

You can sort the contents of many columns by ascending or descending order. To do this, click the heading of the column that you want to sort.

You can create filter criteria by which to display only a subset of information for a column. To create filter criteria, click  for the column that you want to apply filter criteria to, select **Filter**, and enter your filter criteria. The use of filter criteria is not available for some columns.

You can configure the columns that you want to display. To do this, click  in any column, select **Columns**, and deselect the columns that you do not want to appear.

You can re-order columns. To do this, click and hold the column heading and drag it to a different location.

Working with Deployments

Overview to Deployments

In SAS Event Stream Manager, a deployment is a logical grouping of assets as a single unit for life cycle management and monitoring. For more information about how a deployment is used to group assets, see [“Overview of Using SAS Event Stream Manager to Deploy Projects”](#).

Create a Deployment

- 1 On the **Deployments** page, click .

The New Deployment window appears.

- 2 In the **Name** field, enter a unique name for the deployment.
- 3 In the **Description** field, enter a description for the deployment. For example, you can enter the purpose of the deployment to enable users to differentiate between deployments with similar names. This description is displayed as a tooltip when you place the cursor over the deployment name in the Name column on the **Deployments** page.
- 4 In the **Tags** field, you can attribute one or more tags to the deployment.
To add a tag, enter text and press Enter. *Tags* are single-term descriptors for the deployment. Tags can be used to group and filter deployments. Tags cannot contain spaces. Duplicate tags are not permitted on a single deployment.
- 5 In the **Deployment type** drop-down menu, select **Cluster** if you want to deploy projects to ESP servers that run in a Kubernetes cluster. Or, select **Edge** if you want to deploy projects to ESP servers that run on edge servers.
- 6 If you set the **Deployment type** drop-down menu to **Edge**, then the **Production deployment** check box appears. Select the check box if you want to designate this deployment as being suitable for production purposes, as opposed to testing purposes. For more information, see [“Production Assets”](#).
- 7 Click **OK**.

Your deployment appears on a new page.

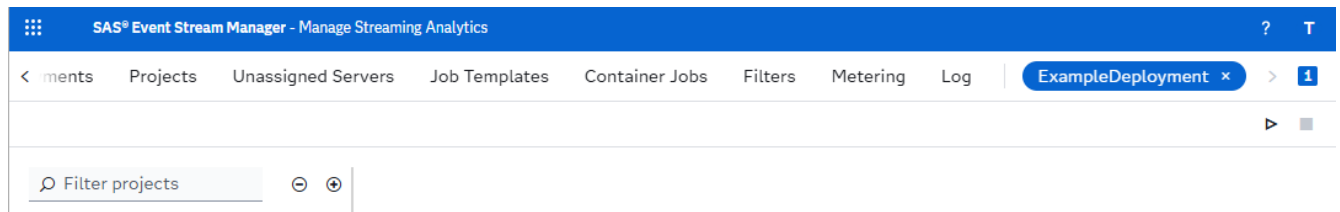
If you set **Deployment type** to **Cluster**, you do not need to add ESP servers to the deployment. An ESP server is created for you when you deploy a project. For more information, see [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#).

If you set **Deployment type** to **Edge**, you can now add ESP servers to your deployment. For more information, see [“Add ESP Servers That Run on Edge Servers”](#).

Open a Deployment

To open an existing deployment, on the **Deployments** page, select the deployment that you want to open and click . The deployment opens on a separate page.

Figure 11 Page for a Deployment Named *ExampleDeployment*



If the deployment contains running projects, you can monitor the status of the deployment. For more information, see the following topics:

- [“Monitor a Specific Deployment Whose Type Is “Cluster””](#)
- [“Monitor a Specific Deployment Whose Type Is “Edge””](#)

You can start a project in the deployment with these methods:

- If the deployment type is “Cluster” (the  icon is displayed adjacent to the page name), see [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#).
- If the deployment type is “Edge” (the  icon is displayed adjacent to the page name), you must add an ESP server to the deployment before you can start a project in the deployment. See [“Add ESP Servers That Run on Edge Servers”](#). There are two ways to start a project:
 - ☐ With user interface controls. For more information, see [“Load and Start a Project on an ESP Server That Runs on an Edge Server”](#).
 - ☐ With a job template. For more information, see [“Deploy a Job Template”](#).

Edit Deployment Properties

- 1 On the **Deployments** page, select the deployment that you want to edit and click .

The Deployment Properties window appears.

- 2 Edit the **Description** and **Tags** fields as required. When the deployment’s type is Edge, you can also select or deselect the **Production deployment** check box to indicate whether the deployment is suitable for production purposes. For more information about these fields, see [“Create a Deployment”](#).

Note: You cannot edit the **Name** field.

- 3 Click **OK**.

Change the Production Status of a Deployment

Certain assets can be marked as production assets. For more information, see [“Production Assets”](#).

When the deployment’s type is Edge, you can change its production status. On the **Deployments** page, right-click the deployment and select **Set as production** or **Remove as production**.

The Production column on the **Deployments** page changes to reflect your choice.

Delete a Deployment

You can delete a deployment if no ESP servers are associated with it. For more information, see [“Remove ESP Servers”](#).

To delete a deployment:

- 1 On the **Deployments** page, select the deployment that you want to delete.
- 2 Click  .
The Delete Deployment window appears.
- 3 Click **Delete**.

Working with ESP Servers

ESP Server That Runs in a Kubernetes Cluster

SAS Event Stream Manager enables you to deploy a project to a Kubernetes cluster. When a project is deployed to a Kubernetes cluster, an ESP server is created on demand in the Kubernetes cluster. When the project is stopped, the ESP server is deleted from the Kubernetes cluster.

When a project is deployed to a Kubernetes cluster, the ESP server that is created is automatically added to the selected deployment. You cannot edit the properties of the ESP server that was created for you.

You can deploy a project to a Kubernetes cluster only if SAS Event Stream Manager itself is running in a Kubernetes cluster. In this case, all deployments that are created in SAS Event Stream Manager are automatically associated with the Kubernetes cluster. On the page for each specific deployment in SAS Event Stream Manager, a user interface control is available so that you can start a project on an ESP server in the Kubernetes cluster.

Note: SAS Event Stream Manager is not a cluster management tool. SAS Event Stream Manager requests the Kubernetes cluster to create and delete ESP servers in the namespace of the current user. System administrators must use the `kubectl` command or other suitable tools to monitor and administer the Kubernetes cluster.

For more information about starting a project on an ESP server that runs in a Kubernetes cluster, see [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#).

ESP Server That Runs on an Edge Server

There are two ways to deploy a project to an ESP server that runs on an edge server: dynamically in the SAS Event Stream Manager user interface or by using a job template. For more information about deploying a project dynamically in the SAS Event Stream Manager user interface, see [“Load and Start a Project on an ESP Server That Runs on an Edge Server”](#). For more information about job templates, see [“Job Template”](#).

Before you can deploy a project to an ESP server that runs on an edge server, the ESP server must be registered with SAS Event Stream Manager. For more information, see [“Automatically Registered ESP Servers and Manually Registered ESP Servers”](#).

Add ESP Servers That Run on Edge Servers

Automatically Registered ESP Servers and Manually Registered ESP Servers

An ESP server must be registered with SAS Event Stream Manager in order for that ESP server to appear in the SAS Event Stream Manager user interface. An ESP server that runs on an edge server can be registered automatically or manually.

If an ESP server that runs on an edge server is not displayed in the SAS Event Stream Processing user interface, see the following topics to register the ESP server manually:

- [“Add a New ESP Server That Runs on an Edge Server to a Deployment”](#)
- [“Add a New ESP Server That Runs on an Edge Server without Adding It to a Deployment”](#)

In SAS Event Stream Manager, “a new ESP server” refers to an existing ESP server that has not been registered with SAS Event Stream Manager, as opposed to creating an ESP server.

Add a New ESP Server That Runs on an Edge Server to a Deployment

To manually register an ESP server that runs on an edge server and add the ESP server to a deployment in one action:

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

A separate page that displays the deployment opens.

- 2 Click  and select **New server**.

Note: The **Add an unassigned server** menu option, which is adjacent to the **New server** menu option, relates to adding a previously registered ESP server.

The ESP Server Properties window appears.

- 3 In the **Name** field, enter a unique name for the ESP server.
- 4 In the **Description** field, enter a description for the ESP server. For example, you can specify the purpose of this ESP server to enable users to differentiate between ESP servers with similar names. This description is displayed as a tooltip when you place the cursor over the ESP server's name in the Name column on the page for a specific deployment that includes this ESP server. Similar tooltips are displayed on the **Unassigned Servers** page for ESP servers that are not assigned to any deployment.
- 5 In the **Host** field, enter the host name of the server that the ESP server runs on.
- 6 In the **HTTP port** field, replace the default value with the port number that is used for HTTP administration requests and for the HTTP publish/subscribe server.

Note: This port is specified in the `-http` argument to the `dfesp_xml_server` command in SAS Event Stream Processing.

- 7 If required, change the setting for the **Authentication** field:
 - **None:** This is the default option.
 - **Kerberos:** This option is relevant only if the ESP server is configured to require authentication using Kerberos.
- 8 If required, select the **Connect using SSL** check box. Selecting this check box is relevant only if the ESP server is configured to require SSL encryption.
- 9 (Optional) Click **Test connection**. Testing the connection prevents the creation of an invalid ESP server connection and subsequent time-outs when the invalid connection is refreshed.
- 10 In the **Tags** field, you can attribute one or more tags to the ESP server.

To add a tag, enter text and press Enter. Tags can be used to group and filter ESP servers. *Tags* are single-term descriptors for the ESP server. Tags cannot contain spaces. Duplicate tags are not permitted on a single ESP server.

- 11 Click **OK**.

The ESP server appears in the table on the deployment page. To view more information about the ESP server, select it in the main table and view the information that appears in the tabs in the bottom pane. For more information about what is displayed, see [“Monitor a Specific Deployment Whose Type Is “Edge””](#).

Add a New ESP Server That Runs on an Edge Server without Adding It to a Deployment

You can add a new ESP server that runs on an edge server to SAS Event Stream Manager without adding it to a deployment. Adding a new ESP server to SAS Event Stream Manager could be useful if you want to list the ESP server in SAS Event Stream Manager but are not ready to add it to a deployment. The ESP server appears on the **Unassigned Servers** page rather than on the page for a specific deployment.

To manually register an ESP server that runs on an edge server without adding the ESP server to a deployment:

- 1 On the **Unassigned Servers** page, click .

The ESP Server Properties window appears.

- 2 Follow the instructions in [“Add a New ESP Server That Runs on an Edge Server to a Deployment”](#) starting with [Step 3](#).

The ESP server is displayed on the **Unassigned Servers** page.

Add an Unassigned ESP Server That Runs on an Edge Server to a Deployment

An unassigned ESP server is an ESP server that runs on an edge server and has been registered with SAS Event Stream Manager but has not been assigned to a deployment.

To assign an ESP server that runs on an edge server to a deployment:

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

A separate page that displays the deployment opens.

- 2 Click  and select **Add an unassigned server**.

Note: The **New server** menu option, which is adjacent to the **Add an unassigned server** menu option, relates to registering an ESP server manually.

The Assign ESP Servers window appears.

- 3 Select the ESP servers that you want to assign to the deployment.

If a large number of ESP servers is displayed, use the field at the top of the window to filter the ESP servers. Filtering makes it easier to select the ESP servers that you want to add to the deployment. For example, if you enter `test`, only ESP servers whose names contain that text are displayed.

- 4 Click **Assign**.

The ESP servers appear in the table on the deployment page. To view more information about a particular ESP server, select it in the main table and view the information that appears in the tabs in the bottom pane. For more information about what is displayed, see [“Monitor a Specific Deployment Whose Type Is “Edge””](#).

Edit an ESP Server That Runs on an Edge Server

To edit the properties of an ESP server that runs on an edge server:

- 1 Open the ESP server if it is not already open: on the page for a specific deployment or on the **Unassigned Servers** page, select the ESP server that you want to open, and click .

The Edit ESP Server Properties window appears.

- 2 Edit the fields as required:

- **Name:** Edit the name of the ESP server. The name must be unique.
- **Description:** Edit the description for the ESP server. For example, you can specify the purpose of this ESP server, to allow users to differentiate between ESP servers with similar names. This description is displayed as a tooltip when you place the cursor over the ESP server's name in the Name column on the page for a specific deployment that includes this ESP server. Similar tooltips are displayed on the **Unassigned Servers** page for ESP servers that are not assigned to any deployment.

- 3 If required, change the setting for the **Authentication** field:

- **None:** This is the default option.
- **Kerberos:** This option is relevant only if the ESP server is configured to require authentication using Kerberos.

- 4 If required, select the **Connect using SSL** check box. Selecting this check box is relevant only if the ESP server is configured to require SSL encryption.

- 5 In the **Tags** field, you can attribute one or more tags to the ESP server. To add a tag, enter text and press Enter. Tags can be used to group and filter ESP servers. *Tags* are single-term descriptors for the ESP server. Tags cannot contain spaces. Duplicate tags are not permitted on a single ESP server.

- 6 Click **OK**.

Remove ESP Servers

Remove an ESP Server That Runs on an Edge Server from a Deployment

Removing an ESP server that runs on an edge server from a deployment enables you to add that ESP server to another deployment.

If you remove an ESP server that has running projects, the running projects continue to run.

To remove ESP servers that run on edge servers from a deployment:

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

A separate page that displays the deployment opens.

- 2 Select the ESP servers that you want to remove. You can select multiple ESP servers by holding down the Ctrl key and using your mouse to select the ESP servers. You can also select all ESP servers by clicking .

- 3 Click .

The button is disabled if you select any ESP servers that are in a cluster.

The Remove ESP Servers window appears.

- 4 Select **Return the ESP servers to the list of unassigned servers**.

- 5 Click **OK**.

Remove an ESP Server That Runs on an Edge Server from SAS Event Stream Manager

Removing an ESP server that runs on an edge server from SAS Event Stream Manager means that SAS Event Stream Manager is not aware of that ESP server. However, the ESP server itself continues to exist.

Note: If the ESP server was registered automatically, you cannot use the SAS Event Stream Manager user interface to remove it.

You can remove an ESP server from SAS Event Stream Manager only if it was registered manually with SAS Event Stream Manager.

To remove ESP servers that are assigned to a deployment from SAS Event Stream Manager, follow the steps in [“Remove an ESP Server That Runs on an Edge Server from a Deployment”](#). Instead of

selecting **Return the ESP servers to the list of unassigned servers**, select **Remove the ESP servers permanently**.

To remove ESP servers that are not assigned to any deployment from SAS Event Stream Manager:

- 1 On the **Unassigned Servers** page, select the ESP servers that you want to remove.
- 2 Click .

The Remove ESP Servers window appears.

- 3 Confirm the deletion.

SAS Event Stream Manager is no longer aware of the ESP servers. However, the ESP servers themselves continue to exist.

Delete an ESP Server That Runs in a Kubernetes Cluster from a Deployment and from SAS Event Stream Manager

To delete an ESP server that runs in a Kubernetes cluster, stop the project that is running on the ESP server. Stopping the project deletes the ESP server from the Kubernetes cluster permanently. For more information, see [“Stop a Project That Is Running on an ESP Server That Runs in a Kubernetes Cluster”](#).

Overview to Versioning

You can use SAS Event Stream Processing Studio and SAS Event Stream Manager to manage multiple versions of a project.

The versioning functionality that is available in SAS Event Stream Processing Studio and SAS Event Stream Manager depends on how SAS Event Stream Processing was deployed.

When standalone SAS Event Stream Processing is deployed, file-based versioning is used to publish project versions. Project versions are stored in a Git repository that is internal to SAS Event Stream Processing and can be synchronized with an external Git repository. For more information, see [“Overview of File-Based Versioning” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

When SAS Event Stream Processing is deployed with other SAS products, project versions can be published using file-based versioning or SAS Viya platform services, depending on each project's type:

- When a project is part of a project package, project versions are published using file-based versioning.
- When the project is a project XML file, project versions are published using SAS Viya platform services. You can subsequently use SAS Event Stream Processing Studio to convert a project XML file to a project package that uses file-based versioning. For more information, see [“Convert a Project to Use File-Based Versioning” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

When SAS Viya platform services are used to publish project versions, the Files service and the File Store service are used to store project versions in a database. For more information, see [“Overview of Publishing Project Versions by Using SAS Viya Platform Services” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

Publishing projects by using SAS Viya platform services is not available when standalone SAS Event Stream Processing is deployed.

Managing Project Versions That Are Published Using File-Based Versioning

Overview of File-Based Versioning

When file-based versioning is used to manage project versions, project versions are stored in a Git repository that is internal to SAS Event Stream Processing and can be synchronized with an external Git repository. For an introduction to file-based versioning and a discussion of how SAS Event Stream Processing Studio and SAS Event Stream Manager interact to manage project versions, see [“Overview of File-Based Versioning” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

File-based versioning in SAS Event Stream Manager enables you to promote project versions from the Test promotion level to the Production promotion level. For more information, see [“Overview of Using SAS Event Stream Manager to Deploy Projects”](#).

IMPORTANT You can use job templates to deploy project XML files. You cannot use job templates to deploy project packages.

View a Project

Overview

The **Projects** page displays projects that are available in SAS Event Stream Manager.

You can create a project in SAS Event Stream Processing Studio or import a project to SAS Event Stream Processing Studio. When you use SAS Event Stream Processing Studio to publish a project (that is, to promote a project from Development to Test), it becomes visible in SAS Event Stream Manager. If you delete a published project in SAS Event Stream Processing Studio, it remains visible in SAS Event Stream Manager.

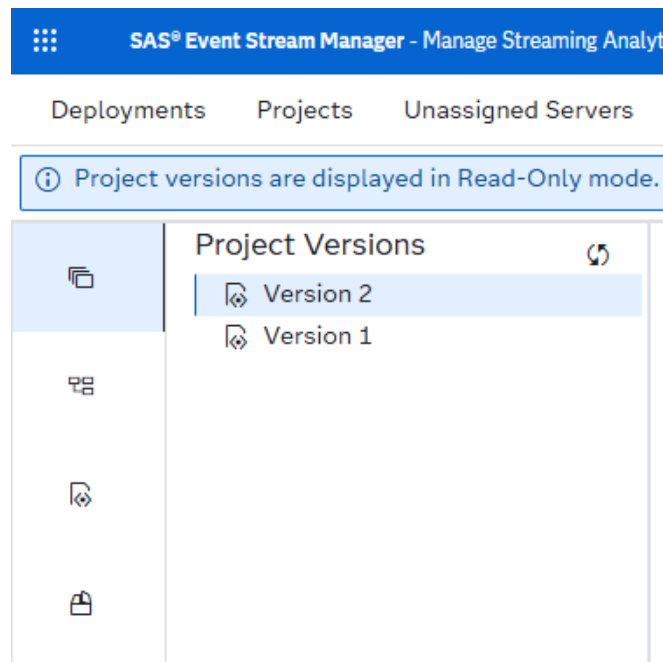
If your published project is not visible on the **Projects** page in SAS Event Stream Manager, click .

The Type column on the **Projects** page shows the project's type. **Package** indicates that a project package exists. **XML** indicates that the project XML file is not part of a project package.

The Remote Repository column shows whether a project is stored in a remote Git repository. For more information, see [“Create a Remote Repository for an Existing Project” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

To view or manage a project, double-click the project on the **Projects** page. The project page appears, with a vertical toolbar on the left, as shown in the following image.

Figure 12 Vertical Toolbar on the Project Page



Project Versions

Click  to view the **Project Versions** pane. Select the project version that you want to view. You can then perform the following actions:

- Use the right pane to manage the project. For example, you can promote a project version to Production. For more information, see [“Manage Project Versions”](#).
- Click the other buttons on the vertical toolbar to view more information about the currently selected version, as described in the following sections.

Diagram

Click  to view the project diagram, with the project's windows and edges. Each window can display icons that represent its state. For more information, see [“Window Icons” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

This is a read-only view. To edit the project, use SAS Event Stream Processing Studio.

XML

Click  to view the XML that is used to construct the model. You can collapse XML elements to restrict your view to the elements that you are interested in.

This is a Read-Only view. To edit the project, use SAS Event Stream Processing Studio.

Project Package

Click  to view the **Project Package** pane. Viewing the project package can be helpful if you want to check which files are included in the project package. The files that are displayed in the **Project Package** pane reflect the version that you selected in the **Project Versions** pane.

This is a Read-Only view. To make changes to the project package, use SAS Event Stream Processing Studio.

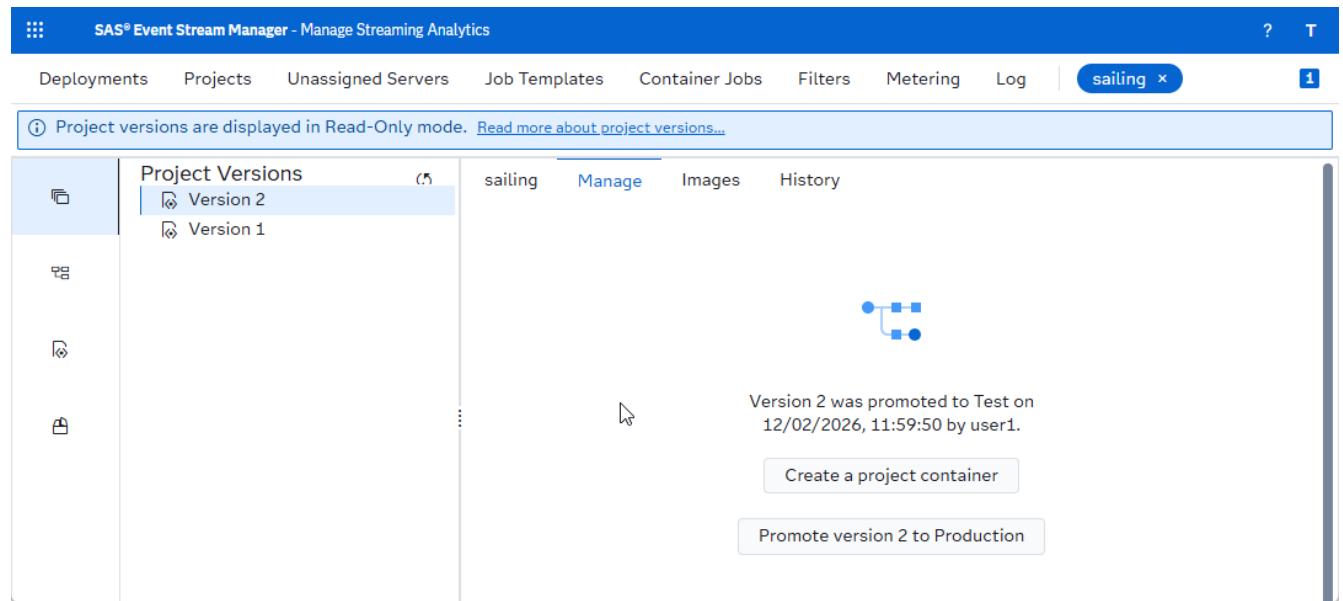
Manage Project Versions

This topic is relevant when project versions are published using file-based versioning. For an introduction to project versions, see [“Overview of File-Based Versioning”](#).

User Interface Controls

For information about how to access project versions in SAS Event Stream Manager, see [“View a Project”](#). The following figure shows information about project versions for a project.

Figure 13 Example of Viewing Project Versions



As shown in the preceding figure, a message in the user interface explains that in SAS Event Stream Manager, project versions are displayed in Read-Only mode. That is, if you want to edit the main content of projects, you need to use SAS Event Stream Processing Studio. You can use SAS Event Stream Manager to promote and deploy projects.

In SAS Event Stream Manager, the functionality that is available in the Project Versions pane and in the right pane is similar to those panes in SAS Event Stream Processing Studio. However, in SAS Event Stream Manager, you cannot work on project versions that belong to the Development promotion level. Use SAS Event Stream Processing Studio to work on project versions that belong to the Development promotion level.

The right pane displays details for the project version that is selected in the **Project Versions** pane:

- Use the **Manage** tab to perform the following actions:
 - Promote a project version to a different promotion level. For more information, see [“Example of Promoting a Project Version from Test to Production” on page 29](#).
 - Create a project container. For more information, see [“Create a Project Container” on page 67](#).

TIP The **Manage** tab also displays messages about the status of the selected project version. Here is an example: Version 1 was promoted to Test on <time> by <user>. You can use this information to learn which promotion level a project version has been promoted to.

- Use the **Images** tab to manage project containers that have been created for the selected project version. For example, you can promote and demote container images. For more information, see [“Deploy and Manage Project Containers” on page 71](#).
- Use the **History** tab to view the version history for the project.

Example of Promoting a Project Version from Test to Production

The following steps demonstrate a scenario for using SAS Event Stream Manager to promote a project version from Test to Production. You learn the following tasks:

- Testing a project version by deploying it to a deployment that you are using for testing purposes.
- Viewing sample data for a running project.
- Promoting a project version from Test to Production.
- Deploying a project version to a deployment that you are using for production purposes.

Note: This example assumes that the SAS Event Stream Processing environment contains a single SAS Event Stream Manager instance. When there are multiple SAS Event Stream Manager instances, after you promote the project to each promotion level you also need to import the project to the relevant SAS Event Stream Manager instance. Only the SAS Event Stream Manager instance that has access to the relevant promotion level can import the project. For more information, see [“Deploying a Project in a Separated System”](#). See also [“Import a Project”](#).

Complete the following steps:

- 1 Complete the example steps in [“Example of Promoting a Project Version from Development to Test” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).
- 2 On the **Projects** page in SAS Event Stream Manager, double-click the `sailing` project to open it. On the **sailing** page, in the **Project Versions** pane, observe that **Version 1** is visible.
- 3 Create a deployment that you can use for testing purposes:
 - a On the **Deployments** page, click .
 - b In the New Deployment window, enter `sailingTest` as the deployment's name and click **OK**.
- 4 Deploy the project to the `sailingTest` deployment:
 - a On the **sailingTest** page, click .
 - The Load and Start Project in Cluster window appears.
 - b In the **Project** field, select **sailing**.
 - c In the **Version** field, **Version 1** is selected. There are no other versions to choose because in this example you created only one version at the Test promotion level.

TIP For information about the other settings in the Load and Start Project in Cluster window, see [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#).

- d Click **OK**.

The project is started.

5 View sample data:

- a On the **sailingTest** page, in the tree view in the left pane, expand the project so that the pod is displayed.
- b Right-click the pod and select **Open running project**.

A new page opens, where sample data is displayed using tables, with a tab for each window.

TIP For more information about the sample data page, see [“View Sample Data”](#).

- c Close the sample data page.
- d On the **sailingTest** page, click .
- e In the Stop Project and Delete ESP Server from Cluster window, click **Delete**.
- f Close the sailingTest deployment page.

6 Suppose that you want to promote Version 1 to Production:

- a Select the sailing project page on the main toolbar.
- b In the **Project Versions** pane, select **Version 1**.
- c On the **Manage** tab, click **Promote version 1 to Production**.
- d In the **Promote Version** window, click **OK**.

Version 1 now belongs to the Production promotion level.

7 Create a deployment that you can use for production purposes:

- a On the **Deployments** page, click .
- b In the New Deployment window, enter **sailingProduction** as the deployment's name.

8 Deploy the project to the sailingProduction deployment:

- a On the **sailingProduction** page, click .
- The Load and Start Project in Cluster window appears.
- b In the **Project** field, select **sailing**.
 - c In the **Version** field, **Version 1** is selected. There are no other versions to choose because in this example you created only one version at the Production promotion level.
 - d Click **OK**.

The project is started.

9 (Optional) View sample data for the project.

10 Stop the project.

Import a Project

This topic is relevant when your SAS Event Stream Processing environment contains multiple instances of SAS Event Stream Manager and you need to make a project available in another SAS Event Stream Manager instance. For more information, see [“Deploying a Project in a Separated System”](#).

After you promote a project to a higher promotion level you need to import the project to the relevant SAS Event Stream Manager instance. That is, promoting a project and importing a project are separate actions. Promoting a project means deciding that the project is suitable for use at the next promotion level. Importing a project makes the promoted project available in another SAS Event Stream Manager instance. Only the SAS Event Stream Manager instance that has access to the relevant promotion level can import the project. For more information, contact your system administrator.

Ensure that the following prerequisites are met:

- The project is stored in a remote Git repository.
- When you are importing a project that has a project container, the SAS Event Stream Manager instances in the test and production environments must have a connection to the same container registry so that both instances can access the container image.

To import a project:

- 1 On the **Projects** page, click .
- 2 Select **Import project from remote server**.
- 3 In the Import Project from Remote Git Server window, select a remote server, and then select the project that you want to import.
- 4 Click **Import**.

You can now open the project and use it. For example, when the project has a project container, you can select a project version in the **Project Versions** pane, and then use the **Images** tab in the right pane to validate or deploy the container image. For more information, see [“Deploy and Manage Project Containers”](#).

Managing Project Versions That Are Published Using SAS Viya Platform Services

Overview of Managing Project Versions That Use SAS Viya Platform Services

For information about circumstances in which it is possible to publish project versions by using SAS Viya platform services, see [“Overview to Versioning”](#).

When SAS Viya platform services are used to manage project versions, the Files service and the File Store service are used to store project versions in a database.

Publishing a project version from SAS Event Stream Processing Studio makes the project version available to SAS Event Stream Manager. You can then use SAS Event Stream Manager to manage the project: you can view the project and indicate whether the project is suitable for use in a production environment.

View a Project

Overview

The **Projects** page displays projects that are available in SAS Event Stream Manager.

You can create a project in SAS Event Stream Processing Studio or import a project to SAS Event Stream Processing Studio. If you publish the project using SAS Event Stream Processing Studio, it becomes visible in SAS Event Stream Manager.

If your published project is not visible on the **Projects** page in SAS Event Stream Manager, click .

If your project is still not visible, click  on the toolbar, and if the **Synchronize published projects** menu item is present, select it. This menu item is present only in certain circumstances.

It is possible to use SAS Event Stream Processing Studio or SAS Event Stream Manager to delete a published project. For more information, see [“Delete a Project”](#).

To view detailed information about a project, select the project on the **Projects** page and click . A page with five tabs appears.

Diagram Tab

The **Diagram** tab displays a graphical representation of the project's windows and edges. Each window can display icons that represent its state. For more information, see [“Window Icons” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

You can use the buttons on the toolbar to zoom in and out of the view, or to zoom to fit the view.

XML Tab

The **XML** tab displays a read-only view of the XML used to construct the model. You can collapse XML elements to restrict your view to the elements that you are interested in.

Details Tab

The **Details** tab displays general information about the project. This information was entered when the current version of the project was imported. For example, the **Details** tab displays the project description and version notes.

Versions Tab

The **Versions** tab lists the current version and any previous versions of the project.

To open a previous version, right-click it and select **Open project version**. A separate page appears where you can review the metadata and XML content relating to that version.

The major version is updated when you publish a new version of the project in SAS Event Stream Processing Studio. For example, the version is updated from 1.0 to 2.0.

The minor version is updated when you accept an update to a project that refers to a model that is stored in the SAS Model Manager common model repository. For example, SAS Event Stream Manager updates the project from version 1.0 to 1.1. For more information, see [“Update SAS Micro Analytic Service Modules”](#).

Files Tab

The **Files** tab lists analytics model files that are loaded from the SAS Model Manager common model repository and are executed through SAS Micro Analytic Service modules and the Calculate window.

Delete a Project

It is possible to delete a project XML file that was published to SAS Event Stream Manager by using SAS Viya platform services.

When an administrator deletes a published project from SAS Event Stream Processing Studio, the administrator can select an option to also delete the project from SAS Event Stream Manager. For more information, see [“Delete a Project” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

Any user of SAS Event Stream Manager can delete a project from SAS Event Stream Manager. The project is not deleted from SAS Event Stream Processing Studio. You or another user who has access to the project could publish the project again from SAS Event Stream Processing Studio to SAS Event Stream Manager, which would result in the project appearing again on the **Projects** page in SAS Event Stream Manager.

To delete a project from SAS Event Stream Manager, select it from the table on the **Projects** page and click . Then confirm the deletion. If the project contains SAS Micro Analytic Service stores, a message informs you that these stores are also deleted.

TIP To delete multiple projects simultaneously, press and hold Ctrl, select the projects that you want to delete, and then click .

Change the Production Status of a Project

Certain assets can be marked as production assets. For more information, see [“Production Assets”](#).

When you are working with a project XML file that is not part of a project package, you can change its production status. On the **Projects** page, right-click the project and select **Set as production** or **Remove as production**.

The Production column on the **Projects** page changes to reflect your choice.

Deploying Projects

Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster

For an introduction to deploying projects to a Kubernetes cluster, see [“ESP Server That Runs in a Kubernetes Cluster”](#).

This topic explains how to load and start a project by using the user interface controls on the page for a specific deployment. An alternative way to load and start a project is by including the `load-project` instruction in a job template. For more information, see [“load-project Element \(Cluster\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

To load and start a project with user interface controls:

- 1 Open the deployment if it is not already open: on the **Deployments** page, double-click the deployment that you want to open.

TIP You can use the Type column on the **Deployments** page to verify that the deployment type is "Cluster". When you have opened the deployment, you can confirm that the deployment type is "Cluster" by verifying that the page title displays the  icon adjacent to the deployment name.

- 2 Click .

The Load and Start Project in Cluster window appears.

- 3 In the **Project** field, select the project.

If the project that you are looking for does not appear in this field, check the project's promotion level. For example, in a separated system, you cannot use a SAS Event Stream Manager instance that is designated as a production instance to deploy a project that has not been promoted to the Production promotion level. For more information, see ["Production Assets"](#).

- 4 In the **Version** field, select the project version. The options available in this field include the latest minor version for each major version of the project.
- 5 When SAS Event Stream Processing is deployed with other SAS Viya platform products, use the **Continuous operation (does not stop and start automatically)** check box to indicate whether the ESP server that is created for your project is required to be in continuous operation. An ESP server that is not required to be in continuous operation might be stopped and started automatically in certain circumstances (for example, during SAS Viya platform system updates or if a non-production environment is stopped during the night for cost-saving purposes). For more information, see ["Understanding the Impact of Stopping and Starting ESP Servers" in SAS Event Stream Processing: Using SAS Event Stream Processing in a Kubernetes Environment](#).
- 6 When a project container has been created for the selected project version, use the **Images** table to select a container image to deploy. Select an image that meets your needs: for example, select an image that was created for a suitable ESP server release and architecture.

Container images are self-contained and immutable. For more information, see ["Project Containers" on page 62](#). In contrast, when no images are available, you are deploying a project that is stored on the Kubernetes persistent volume, using the standard ESP server image (`sas-event-stream-processing-server-app`).

IMPORTANT When a project container is created, SAS Event Stream Manager creates the image based on the specification but does not check that the image runs.

It is recommended that you do not deploy untested images to production environments. It is good practice to deploy an image to a test environment first to check that it runs as expected.

- 7 When your project is not part of a project package, ensure that input files are available in the expected location. If your project uses connectors that require a Kubernetes persistent volume, you must place input files in the location that has been configured for the Kubernetes persistent volume. The base location where SAS Event Stream Processing expects input files to be located

is `/mnt/data`, but the corresponding location where you need to place your input files depends on how your file storage has been set up.

If you require assistance, contact your system administrator.

- 8 (Optional) Adjust the values in the **Deployment settings** section. The **Deployment settings** section displays the settings that are used for creating an ESP server in the cluster. If the application can access previously used settings for the selected project, then those settings are displayed as default values. Otherwise, the project's computational requirements are used as default values. For more information, see ["Specify Computational Requirements" in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#). Contact your system administrator for advice.

- a Click Edit deployment settings.

The Deployment Settings window appears.

- b Specify the requested (initial) amount and the maximum amount to be used by the ESP server for each of the following resources:
 - Amount of memory.
 - Amount of CPU resource.
 - Amount of NVIDIA graphics processing unit (GPU) resource. The SAS Event Stream Manager user interface contains one field for setting both the requested amount and the maximum amount because these amounts must be the same.
 - Number of replicas, which are the number of ESP server instances created by autoscaling.

Note:

- The requested amount and maximum amount that you should set for memory and CPU resource depend on your circumstances. In many cases, the requested amount and the maximum amount can be set to the same value. However, to optimize the use of resources, you can set different values. Setting different values means that the Kubernetes pod that runs the project can be started with a lower amount of resource than the maximum amount that the pod might need with peak usage. However, if you expect to leave the project running for a long time, having a significant difference between the requested amount and the maximum amount might result in the Kubernetes cluster starting the pod before the maximum amount is reached. That is, the Kubernetes cluster might start a pod when the cluster can meet the minimum amount that you requested and you might not get the maximum amount of resource that you specify. Lack of sufficient resource might then result in the project stopping unexpectedly.
 - Your system administrator can set limits for the amount of memory and the amount of CPU resource that is allocated. If a value that you specify exceeds such a limit, then the value specified by your system administrator is used instead. The user interface displays an error message if you try to set a value that exceeds the value set by your system administrator.
-

- c Specify the target CPU utilization, as a percentage. This is the threshold at which another Kubernetes pod is created. This value is the average for all pods.
 - d Click **OK**.

- 9 Click **OK**.

The project appears in the left pane.

For information about how to monitor the status of your running project, see [“Monitor a Specific Deployment Whose Type Is “Cluster””](#).

If your environment includes Grafana and the SAS Event Stream Processing Data Source Plug-in for Grafana, you can use them to visualize events. For more information, see <https://github.com/sassoftware/grafana-esp-plugin>.

Stop a Project That Is Running on an ESP Server That Runs in a Kubernetes Cluster

When you stop a project that is running on an ESP server that runs in Kubernetes a cluster, the ESP server is deleted from the Kubernetes cluster.

You can stop a project by including the `unload-project` instruction in a job template. For more information, see [“unload-project Element \(Cluster\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#). You can also stop a project by using the user interface controls on the page for a specific deployment.

To stop a project with user interface controls:

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

TIP You can use the Type column on the **Deployments** page to verify that the deployment type is “Cluster”. When you have opened the deployment, you can confirm that the deployment type is “Cluster” by verifying that the page title displays the  icon adjacent to the deployment name.

- 2 In the left pane, select the project that you want to stop.
- 3 Click  on the toolbar.
The Stop Project and Delete ESP Server from Cluster window appears.
- 4 Click **Delete**.

The project is stopped and the ESP server on which it was running is deleted from the Kubernetes cluster.

Load and Start a Project on an ESP Server That Runs on an Edge Server

You can load and start projects by including the `load-project` and `start-project` instructions in a job template. For more information, see [“instructions \(Edge\)” in SAS Event Stream Processing: XML](#)

[Language Reference for Job Templates](#). You can also load and start projects by using the user interface controls on the page for a specific deployment. Loading and starting projects using these user interface controls is appropriate in many situations. However, for complex situations, using instructions in a job template might be more useful. For example, in the Activity Tracker example, where you collect user input about the selected device ID, using instructions in a job template is appropriate. For more information, see [“Overview of the Activity Tracker Example”](#).

Note: If your project uses a Python or Lua virtual environment and you want to deploy the project to an edge device, you must use a project container to deploy your project. Project containers include the files that pertain to selected virtual environments, which can help ensure that required dependencies are available in the edge environment. For more information, see [“When Is it Useful to Deploy Project Containers?”](#).

To load or start a project with user interface controls:

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

TIP You can use the Type column on the **Deployments** page to check that the deployment type is "Edge". When you have opened the deployment, you can confirm that the deployment type is "Edge" by checking that the page title displays the  icon adjacent to the deployment name.

- 2 In the main table, select the ESP servers on which you want to load or start a project. To select multiple ESP servers, hold down the Ctrl key and click the ESP servers. You can also select all ESP servers by clicking .
- 3 Click  and select **Load project**, **Start project**, or **Load and start project**, as required.

Note: The **Start project** and **Load and start project** menu items also start project connectors.

The Load Project window, the Start Loaded or Stopped Project window, or the Load and Start Project window appears.

- 4 Select the project, and if prompted for select the version, and click **OK**.

A new page appears, displaying details about the job. For more information, see [“View Job Details”](#).

For information about how to monitor the status of your running project, see [“Monitor a Specific Deployment Whose Type Is “Edge”](#)”.

Stop and Unload a Project That Is Running on an ESP Server That Runs on an Edge Server

You can stop and unload projects by including the `stop-project` and `unload-project` instructions in a job template. For an example of a job template like this, see [“Job Template for Stopping a Project”](#). For more information, see [“instructions \(Edge\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

You can also stop and unload projects by using the user interface controls on the page for a specific deployment.

To stop or unload a project with user interface controls:

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

TIP You can use the Type column on the **Deployments** page to verify that the deployment type is "Edge". When you have opened the deployment, you can confirm that the deployment type is "Edge" by verifying that the page title displays the  icon adjacent to the deployment name.

- 2 In the main table, select the ESP servers on which you want to stop or unload a project. To select multiple ESP servers, hold down the Ctrl key and click the ESP servers. You can also select all ESP servers by clicking .

- 3 Click  and select **Stop project**, **Unload project**, or **Stop and unload project**, as required.

The Stop Running Project window, the Unload Project window, or the Stop and Unload Running Project window appears.

- 4 Select the project and click **OK**.

A new page appears, displaying details about the job. For more information, see [“View Job Details”](#).

When a project has stopped, the **Projects** tab in the bottom pane of the **Deployments** page displays the project's status as stopped. When a project has been unloaded, it no longer appears in the **Projects** tab.

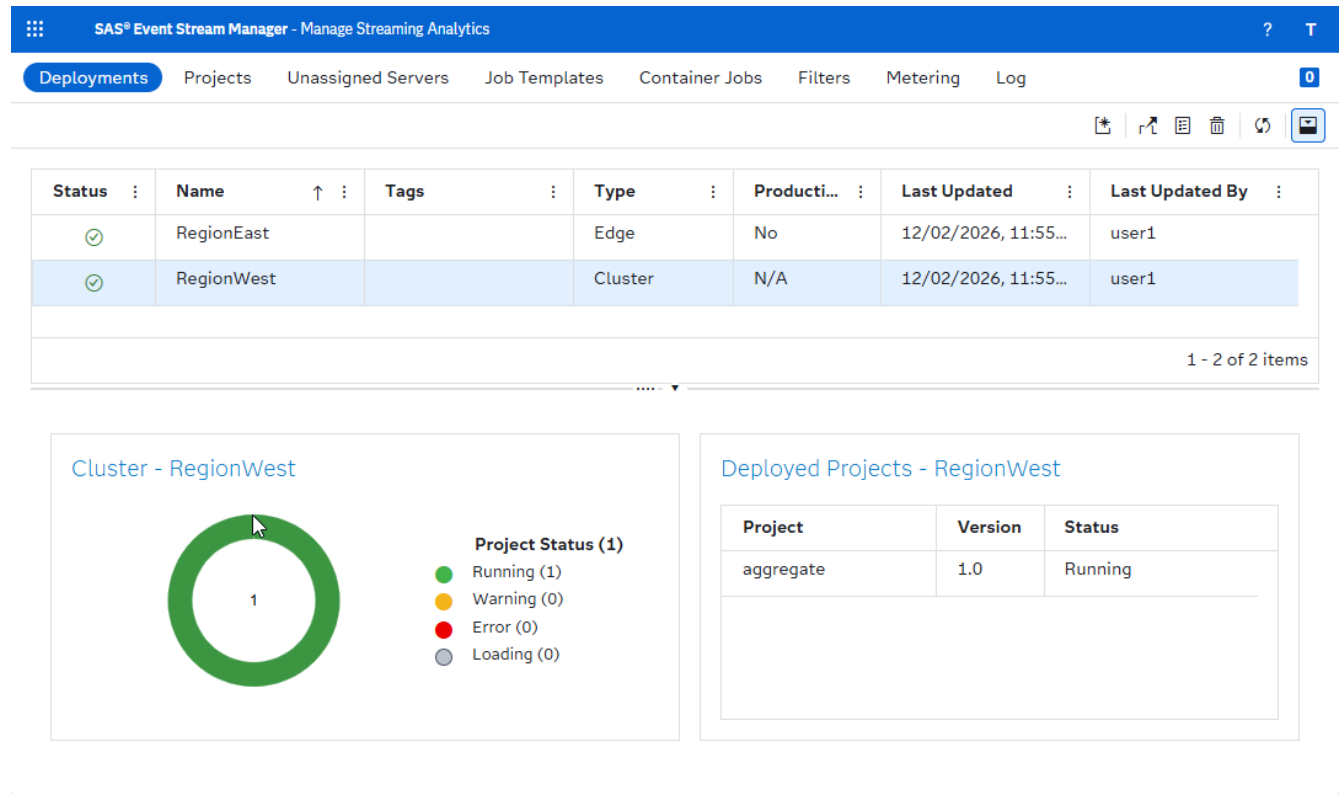
If a project is reported as missing, the Status column in the **Projects** tab displays the value `Missing` and the status icon is red (●). In this case, you might not be able to unload the project using an instruction. A project is reported as missing if the ESP server that it is running on is restarted. A project might be reported as missing if it was started, but never stopped, by SAS Event Stream Manager, and now is no longer available to the ESP server. To clean up a missing project, right-click the ESP server that contains the affected project and select **Clean up missing projects**. The affected project no longer appears in the **Projects** tab.

Monitoring a Deployment

Monitor All Deployments

The **Deployments** page displays any active deployments in your SAS Event Stream Processing environment.

Figure 14 Deployments Page with Two Deployments



The main table on the **Deployments** page displays the following information about each deployment:

- the status of the ESP servers associated with the deployment.
- the deployment's name. If the deployment has a description, it is displayed as a tooltip when you place the cursor over the deployment name.
- the tags attributed to the deployment. Tags can be used to group and filter deployments. You can attribute tags to a deployment by editing the deployment. For more information, see ["Edit Deployment Properties"](#).
- the deployment's type: whether the deployment contains ESP servers that run in a Kubernetes cluster or ESP servers that run on an edge server.

- whether a deployment whose type is Edge is a production deployment. This column is not applicable to deployments whose type is Cluster. For more information, see [“Production Assets”](#).
- the date on which the deployment was last updated.
- the user ID or user name of the account that last updated the deployment.

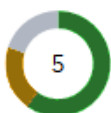
The Status column provides a summary of the condition of the ESP servers associated with the deployment. The condition of an ESP server is determined by comparing the state of its projects with their expected state. This information helps you focus on those deployments that have problems. The following icons can appear in the Status column:

Icon	Description
	Available (good) — The ESP servers associated with this deployment are available and do not have any projects in a warning or error state.
	If the deployment type is “Cluster”, the status icon has the following meaning: Crashed and restarting — At least one ESP server instance in the Kubernetes cluster has crashed and is restarting. If the deployment type is “Edge”, the status icon has the following meaning: Available (project error) — The ESP server is available, but at least one project is in an error state.
	If the deployment type is “Cluster”, the status icon has the following meaning: Insufficient resources — At least one instance of the ESP server in the Kubernetes cluster cannot start because there are no cluster nodes on which it can run. There are no instances that have crashed and are restarting. If the deployment type is “Edge”, the status icon has the following meaning: Available (unmanaged) — The ESP server is available and does not have any projects that are in an error state. However, it is running at least one unmanaged project (that is, a project that was not initiated from within SAS Event Stream Manager).
	Unavailable — The ESP servers associated with this deployment are not available.

You can arrange information in the table in several ways. For more information, see [“Arrange Information in Tables”](#).

Clicking a deployment in the main table populates the lower part of the **Deployments** page.

- The tile on the left has the title **Cluster** or **Edge**, depending on the deployment type. An icon similar to the following example is displayed:



If the deployment type is “Cluster”, the number of projects associated with the deployment is shown in the center of the icon. If the deployment type is “Edge”, the number of ESP servers associated with the deployment is shown instead.

If a deployment has more than one project or ESP server that are in different states, the icon in the Status column is displayed in multiple colors. These colors indicate the status of each of the projects or ESP servers associated with the deployment. For example, in the example above a deployment whose type is “Edge” has five ESP servers — three are in the state **Available** (good), one is **Available (unmanaged)**, and one is **Unavailable**.

- The **Deployed Projects** tile shows the projects that have been deployed as part of the selected deployment. If the deployment type is “Cluster”, this tile shows the name, version, and status of each project. If the deployment type is “Edge”, this tile shows the name and version of the project, tags attributed to the project, and the number of running instances of the project.

The project names and versions displayed in this tile correspond to the project names and versions displayed on the **Projects** page.

To view more details for a specific deployment, see the following topics:

- [“Monitor a Specific Deployment Whose Type Is “Cluster””](#)
- [“Monitor a Specific Deployment Whose Type Is “Edge””](#)

Monitor a Specific Deployment Whose Type Is “Cluster”

Overview

To view details for a specific deployment, select the deployment on the **Deployments** page and click . A separate page appears, displaying information about the projects associated with the deployment.

TIP You can use the Type column on the **Deployments** page to check that the deployment type is “Cluster”. When you have opened the deployment, you can confirm that the deployment type is “Cluster” by checking that the page title displays the  icon adjacent to the deployment name.

Click a project in the left pane to display a list of the Kubernetes pods that SAS Event Stream Processing has created for that project. When you select a project or a pod in the left pane, the right pane displays further information about the project or pod.

Figure 15 Example of a Page for a Specific Deployment Whose Type Is “Cluster”

The screenshot shows the SAS Event Stream Manager interface. The top navigation bar includes links for Deployments, Projects, Unassigned Servers, Job Templates, Container Jobs, Filters, Metering, and Log. A 'RegionWest' button is also visible. The left pane shows a list of projects under the 'Filter projects' search box. The selected project is 'esm--52egion-57est-sailing...'. The right pane displays the details for this project, including its name, restarts, source, status, and project health.

Details Log Performance

Name:
esm--52egion-57est-sailing-6fd5f47c5-pwtj6

Restarts:
0

Source:
(none)

Status:
Available (good)

Project Health:

Name	Value
connector_latency_avg_cq1/Boat4/Boat4	2501995
connector_latency_max_cq1/Boat4/Boat4	5003984

You can filter the projects in the left pane by entering text in the **Filter Projects** search box. The left pane then shows only those projects whose names match the text. This can be helpful if the deployment contains a large number of projects.

Note: The pod name is displayed in the left pane. The name consists of the prefix `esm` followed by the name of your deployment, the name of the project that was started in the cluster, and a unique identifier for the pod. These four parts are separated by hyphens. If the deployment name or the project name contains characters that do not conform to Kubernetes naming conventions, the resulting pod name contains changed characters. No characters are changed if the deployment name and the project name contain only lowercase letters (without diacritical marks), numbers, hyphens, or periods, and the names start with lowercase letters.

To collapse all projects and pods in the left pane, click . To expand all projects and pods in the left pane, click .

Monitor Projects

When you select a project in the left pane, the right pane displays the **Details**, **Kubernetes Metrics**, **Cluster Runtime Configuration**, and **Cluster Properties** tabs.

Some of the information that is displayed in the **Kubernetes Metrics** tab and the **Cluster Runtime Configuration** tab relates to settings that you can configure when you load and start a project:

- When you load and start a project by using the user interface controls, you can specify settings in the Deployment Settings window. For more information about the Deployment Settings window, see [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#).
- When you load and start a project by using a job template, you can specify settings in the job template. For more information about the relevant settings in a job template, see [“cluster-runtime-group Element” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

The information in the **Cluster Properties** tab relates to environment variables that can be specified in the **Global** section of the **Settings** page or in a job template. For more information about the **Settings** page, see [“Managing Settings”](#). For more information about specifying environment variables in a job template, see [“cluster-properties-groups \(Cluster\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

The **Details** tab displays the following information for the selected project:

- The project’s name.
- The tags attributed to the project. Tags can be used to group and filter projects. You can attribute tags to a project when you create or edit it in SAS Event Stream Processing Studio. For more information, see [“Create a Project” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).
- The time at which the project was loaded and started on the Kubernetes cluster.
- The project’s description.
- Whether the ESP server for this project is required to be in continuous operation. For more information, see [“Understanding the Impact of Stopping and Starting ESP Servers” in SAS Event Stream Processing: Using SAS Event Stream Processing in a Kubernetes Environment](#).
- The project’s status. The condition of a project is determined by comparing the state of the project with its expected state. This information helps you identify projects that have problems. The following status icons can appear in the left pane and in the **Details** tab in the right pane:

Icon	Description
	Starting in cluster — The project is starting in the Kubernetes cluster.
	Available (good) — The project is available and is not in a warning or error state.
	Crashed and restarting — At least one ESP server instance in the Kubernetes cluster has crashed and is restarting.
	Insufficient resources — At least one instance of the ESP server in the Kubernetes cluster cannot start because there are no cluster nodes on which it can run. There are no instances that have crashed and are restarting.
	Connector failure — At least one pod for the project has a failed connector. Expand the project in the left pane to identify the affected pods. Then for each affected pod use the Performance tab in the right pane to identify windows with failed connectors.

- The project's version number.
- The version notes that were entered when this version of the project was created.
- The user ID of the user who started the project

The **Kubernetes Metrics** tab displays the following information about the Kubernetes Horizontal Pod Autoscaler (HPA):

- The current CPU utilization, as a percentage. This value is the average for all pods.
Click **Refresh Metrics** to update this information. If the project is not running, the CPU utilization is displayed as 0%. When the project starts, after you click **Refresh Metrics** again, the message CPU usage is not determined yet is displayed. After a while, when you click **Refresh Metrics** again, the CPU utilization is displayed.

Note: This functionality relies on the Kubernetes Metrics Server component being available in your environment. If this component is not present, a message about the Kubernetes Metrics Server is displayed in the **Kubernetes Metrics** tab. Contact your system administrator.

- The target CPU utilization, as a percentage. This is the threshold at which another Kubernetes pod is created. This value is the average for all pods. You can set this value when you load and start a project.
- The length of time that the project has been running. Click **Refresh Metrics** to update this information.

The **Cluster Runtime Configuration** tab displays pod resource settings that are used when ESP servers are created in the Kubernetes cluster, for the selected project. You can set these values when you load and start a project. This tab displays the requested (initial) amount and the maximum amount to be used by ESP servers for each of the following resources:

- Amount of memory.
- Amount of CPU resource.
- Amount of NVIDIA graphics processing unit (GPU) resource.
- Number of replicas, which are the number of ESP server instances created by autoscaling.

The **Cluster Properties** tab displays the environment variables that are used when ESP servers are created in the Kubernetes cluster, for the selected project. These environment variables can be specified in the **User Session** section of the **Settings** page or in a job template. Environment variables that are specified in a job template override environment variables that are specified on the **Settings** page. You can use the **Cluster Properties** tab to check which of these environment variables are in use for the selected project. The tab also displays the value of the MYUSERID environment variable. For information about the MYUSERID environment variable, see [“Using Tokens for Predefined Variables” in SAS Event Stream Processing: XML Language Reference for Event Stream Processing Models](#).

Monitor Pods

When you select a pod in the left pane, the right pane displays the **Details**, **Log**, and **Performance** tabs.

The **Details** tab displays the following information for each pod:

- The name of the pod.
- How many times the pod has been restarted.
- The name of the Kubernetes cluster where the pod is located.
- The pod's status. The condition of a pod is determined by comparing the state of the pod with its expected state. This information helps you identify pods that have problems. The following status icons can appear in the left pane and in the **Details** tab in the right pane:

Icon	Description
	This icon indicates one of the following statuses. Place the cursor over the icon to display the status. ■ Starting — The pod is starting. ■ Unknown — The pod's status is unknown.
	Running — The pod is running and is not in a warning or error state.
	This icon indicates one of the following statuses. Place the cursor over the icon to display the status. ■ Project error — There is a problem with the project that the pod is trying to run. ■ Insufficient resources — The pod in the Kubernetes cluster cannot start because there are no cluster nodes on which it can run. ■ Cluster pod is in a crash loop — The pod in the Kubernetes cluster is crashing and is restarting repeatedly. ■ Cluster pod is unable to load an image — The pod in the Kubernetes cluster is unable to load a container image. ■ Cluster error — At least one ESP server instance in the Kubernetes cluster has crashed and is restarting.
	Connector failure — The pod has at least one failed connector. Use the Performance tab to identify windows with failed connectors.

- The health of the project that the pod was created for. Various metrics are displayed, such as the current state of connectors. For more information about these metrics, see [“Understanding Project Health Metrics” in SAS Event Stream Processing: Troubleshooting](#).

The **Log** tab displays log messages. For more information, see [“View Logs for a Deployment Whose Type Is “Cluster””](#).

The **Performance** tab provides various statistics about performance and can help you identify windows that have performance issues or failed connectors. For more information, see [“View Performance Information for a Deployment Whose Type Is “Cluster””](#).

Perform Actions on Projects

On the page for a specific deployment whose type is “Cluster”, you can perform the following actions on projects associated with your deployment:

- Load and start projects. For more information, see [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#).
- View sample data for running projects. For more information, see [“View Sample Data”](#).
- Stop projects. For more information, see [“Stop a Project That Is Running on an ESP Server That Runs in a Kubernetes Cluster”](#).

View Logs for a Deployment Whose Type Is “Cluster”

Access the Logs

- 1 Open the deployment whose logs you want to view: on the **Deployments** page, select the deployment and click .

TIP You can use the Type column on the **Deployments** page to check that the deployment type is “Cluster”. When you have opened the deployment, you can confirm that the deployment type is “Cluster” by checking that the page title displays the  icon adjacent to the deployment name.

- 2 In the left pane, select a project and then select a Kubernetes pod.
- 3 In the right pane, click the **Log** tab.

About the Log Tab

SAS Event Stream Manager displays recent log messages for the pod that you selected, including messages that were logged before you accessed the **Log** tab. The messages are updated in real time. Up to 1,000 lines are displayed.

Select the Source of the Log Messages

Use the **Log** drop-down list to select the source of the log messages:

- **ESP server** - Messages from the ESP server log. These messages relate to the ESP server that was created in the Kubernetes cluster when you deployed the project. TRACE-level messages are

not displayed. Some message details are not displayed in the user interface. To view the complete message text, including TRACE-level messages, export the log for the Kubernetes pod and view its contents. For more information, see [“Export the Logs”](#).

- **ESP operator** - Messages from the ESP operator log. These messages relate to all deployments in SAS Event Stream Manager. The ESP operator is a Kubernetes operator, a software extension that uses custom resources to manage applications. For more information about Kubernetes operators, see the [relevant Kubernetes documentation](#). This option enables you to view Kubernetes messages that might otherwise need to be accessed by a system administrator by using the `kubectl` command. If the pod does not start and you cannot access log messages in the **Log** tab, you can access the ESP operator log on the **Settings** page instead. For more information, see [“ESP Operator”](#).
- **Console** - Standard output (stdout) messages that are created by the ESP server but are not included in the ESP server log.
- **Pod events** - Messages that are created by the Kubernetes cluster for a specific pod. This option enables you to view Kubernetes messages that might otherwise need to be accessed by a system administrator by using the `kubectl` command.

Filter Log Messages

To filter log messages by message type, select one or more of the following options in the **Log** tab:

- All - Shows all log message types, except TRACE-level messages from the ESP server log.
- Informational - Shows general log messages that are not warnings or errors.
- Warning - Shows only warning messages.
- Fatal and Error - Shows only error messages, including fatal errors and normal errors.

If you select **Console** or **Pod events** from the **Log** drop-down list, the ability to filter log messages is not available.

Clear the Log Tab

To clear the contents of the **Log** tab, click  **Clear Log**.

Export the Logs

To export all logs to text files, click  **Export All Logs**. A ZIP file that contains the log files is exported to your computer. The location into which the ZIP file is exported might vary depending on your browser's configuration.

The following log files are available:

- **pod.log** - The full log for the Kubernetes pod. The log contains the complete text of all ESP server messages, including TRACE-level messages (if TRACE-level logging is enabled), and all console messages.

TIP If the exported Kubernetes pod log does not contain the level of logging detail that you want to view, you can change the logging properties on the **User Session** section of the **Settings** page. For example, you could change the logging level for a particular logger from INFO to TRACE. For more information, see [“Settings for the User Session”](#).

- **previous pod.log** – This file is present only when the Kubernetes pod has been restarted. The file contains log entries for the original pod.
- **operator.log** – The log for the ESP operator.

View Performance Information for a Deployment Whose Type Is “Cluster”

When you suspect that your model has performance issues, viewing performance information can help you identify the windows that are causing the issues.

To view performance information:

- 1 Open the deployment that contains the project whose performance you want to view: on the **Deployments** page, select the deployment and click .

TIP You can use the Type column on the **Deployments** page to check that the deployment type is “Cluster”. When you have opened the deployment, you can confirm that the deployment type is “Cluster” by checking that the page title displays the  icon adjacent to the deployment name.

- 2 In the left pane, select a project and then select a Kubernetes pod.
- 3 In the right pane, click the **Performance** tab.

The following fields are displayed at the top of the page:

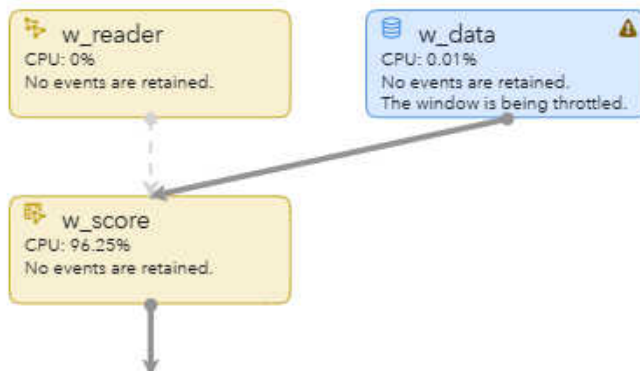
- The **System memory** field displays how much memory is available on the Kubernetes node.
- The **Virtual memory** field displays how much memory is allocated to the project (that is, the Kubernetes pod). This is not a hard limit. As resident size grows, the ESP server requests more virtual memory.
- The **Resident memory** field displays how much memory the project (the Kubernetes pod) actually consumes.
- The **Limit for memory** field displays how much memory you specified the project (the Kubernetes pod) is allowed to use. You can specify a limit for memory in the Deployment Settings window, which is accessed from the Load and Start Project in Cluster window.

The read-only project diagram on this page displays summary information for each window in the model, including CPU usage and event count. When a window is stateless, no events are retained.

The  and  icons can be displayed in the windows, depending on the severity of issues. For

example, the  icon is displayed when a window is throttled (a connector is blocked because data arrives faster than the ESP server can process it) or when a connector fails.

Figure 16 Example of Performance Information in the Diagram



When you select a window in the diagram, the tabs below the diagram display further information for that window:

- The **Health** tab displays various metrics for the window, such as the amount of time the window has been throttled and the current state of the window's connectors. For more information about these metrics, see ["Understanding Project Health Metrics" in SAS Event Stream Processing: Troubleshooting](#).
- The **Events** tab displays events in the latest event block that has passed through the model. For example, if each event block contains one event, then this tab displays one event at a time.

The CPU usage information for each window is created by inspecting a time period and checking what percentage of that time the CPU is busy. For example, if a one-second period is inspected and the CPU is busy for the entire one-second period, then 100% CPU usage is reported. If a one-second period is inspected and the CPU is busy for half of that time, then 50% CPU usage is reported.

When a window uses more CPU resource than other windows in the model, this does not necessarily indicate a problem in the model's design. The windows that are designed to handle the most events in your model are also the windows that are likely to use more CPU resource than windows that handle fewer events. Similarly, when a window is throttled, the project is managing its intake of events. Throttling that occurs in a Source window can prevent throttling from occurring in subsequent windows.

When you use the Load and Start Project in Cluster window to specify deployment settings, take care to specify an appropriate amount of CPU resource. Specifying an insufficient amount of CPU resource for a resource-intensive model might result in warnings about CPU usage and throttling. The warnings might indicate an insufficient CPU resource allocation rather than a problem with in the model's design.

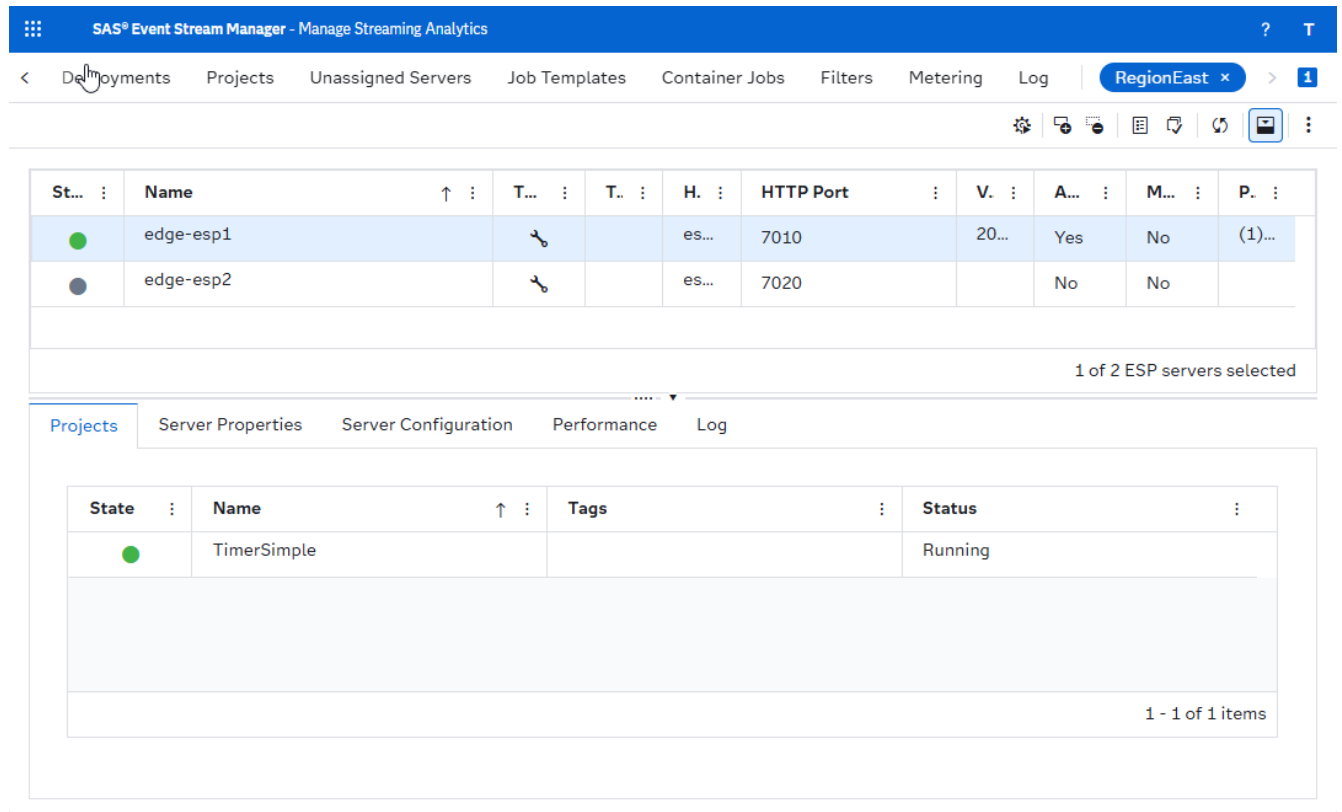
Monitor a Specific Deployment Whose Type Is “Edge”

To view more details for a specific deployment, select the deployment on the **Deployments** page and click . A separate page appears, displaying information about the ESP servers assigned to the deployment and the projects running on those ESP servers.

TIP You can use the Type column on the **Deployments** page to check that the deployment type is “Edge”. When you have opened the deployment, you can confirm that the deployment type is “Edge” by checking that the page title displays the  icon adjacent to the deployment name.

Clicking the name of an ESP server in the main table populates the bottom pane with tabs that contain further information about that ESP server.

Figure 17 Example of a Page for a Specific Deployment Whose Type Is “Edge”



The screenshot shows the SAS Event Stream Manager interface. The top navigation bar includes links for Deployments, Projects, Unassigned Servers, Job Templates, Container Jobs, Filters, Metering, and Log. The current page is titled "SAS® Event Stream Manager - Manage Streaming Analytics" and shows the "RegionEast" deployment. The main table displays two ESP servers:

St...	Name	T...	T..	H.	HTTP Port	V.	A...	M...	P.
	edge-esp1			es...	7010	20...	Yes	No	(1)...
	edge-esp2			es...	7020		No	No	

Below the table, it indicates "1 of 2 ESP servers selected". The bottom pane shows the "Projects" tab for the selected server, displaying a single project:

State	Name	Tags	Status
	TimerSimple		Running

The bottom right of the Projects tab indicates "1 - 1 of 1 items".

The main table displays the following information for each server or device defined as an ESP server:

- The ESP server's status.

- The ESP server's name.
- The ESP server's type.

Icon	Description
	The ESP server was automatically registered with SAS Event Stream Manager. For more information about automatically registered ESP servers, see “Automatically Registered ESP Servers and Manually Registered ESP Servers” .
	The ESP server was manually registered with SAS Event Stream Manager. For more information about manually registered ESP servers, see “Automatically Registered ESP Servers and Manually Registered ESP Servers” .

- The tags attributed to the ESP server. Tags can be used to group and filter ESP servers. You can attribute tags to an ESP server by editing the ESP server. For more information, see [“Edit an ESP Server That Runs on an Edge Server”](#).
- The host on which the ESP server is running.
- The port for HTTP administration requests and for the HTTP publish/subscribe server.
- The SAS Event Stream Processing version installed on the host on which the ESP server is running.
- Whether a SAS Event Stream Processing Analytics license is installed on the host. To deploy a project that contains SAS Event Stream Processing Analytics windows, an appropriate license must be available.
- Whether SAS Event Stream Processing has been enabled to count the number of events that are processed on the ESP server.
- A count of projects with different statuses. An example is (4) 2 Running, 1 Loaded, 1 Stopped.

The Status column provides a summary of the ESP server's condition. The condition of an ESP server is determined by comparing the state of its projects with their expected state. This information helps you identify ESP servers that have problems. The following icons can appear in the Status column:

Icon	Description
	Available (good) — The ESP server is available and does not have any projects in a warning or error state.
	Available (project error) — The ESP server is available, but at least one project is in an error state.
	Available (unmanaged) — The ESP server is available and does not have any projects that are in an error state. However, it is running at least one unmanaged project (that is, a project that was not initiated from within SAS Event Stream Manager).

Icon	Description
	Unavailable — The ESP server is not available.

You can arrange information in the table in several ways. For more information, see [“Arrange Information in Tables”](#).

Clicking the name of an ESP server populates the bottom pane with tabs that contain information relating to that ESP server:

- The **Projects** tab displays projects that are running on the ESP server.
The status icon provides information that is similar to the status icon in the main table on the page for a specific deployment: the status of a project is determined by comparing its state with its expected state. For example, if the status and the expected status do not match, the icon is red and its tooltip shows the message `Unexpected status`.
- The **Server Properties** tab displays identifying information about the ESP server. You can click **Edit properties** to change most of these properties.
- The **Server Configuration** tab displays information about connector types, streaming analytics, and whether SAS Event Stream Processing has been enabled to meter the number of events that are processed on the ESP server.
- The **Performance** tab provides information about memory use. It also provides information about CPU use for each project and window.
- The **Log** tab displays log messages for the ESP server. For more information, see [“View the ESP Server Log for a Deployment Whose Type Is “Edge”](#)”.

On this page you can also perform actions on projects associated with your deployment:

- Load and start projects. For more information, see [“Load and Start a Project on an ESP Server That Runs on an Edge Server”](#).
- View sample data for running projects. For more information, see [“View Sample Data”](#).
- Stop and unload projects. For more information, see [“Stop and Unload a Project That Is Running on an ESP Server That Runs on an Edge Server”](#).

View the ESP Server Log for a Deployment Whose Type Is “Edge”

Access the Log

If an ESP server is assigned to a deployment, you can use SAS Event Stream Manager to view the ESP server log.

To view the ESP server log:

- 1 Open the deployment to which the ESP server is assigned: on the **Deployments** page, select the deployment and click .

TIP You can use the Type column on the **Deployments** page to check that the deployment type is "Edge". When you have opened the deployment, you can confirm that the deployment type is "Edge" by checking that the page title displays the  icon adjacent to the deployment name.

- 2 In the main table, select the relevant ESP server.
- 3 In the bottom pane, click the **Log** tab.
- 4 If the **Enable** button is displayed in the **Log** tab, click it to enable logging on the ESP server. By default, logging is disabled on each ESP server that runs on an edge server.

About the Log Tab

SAS Event Stream Manager displays a real-time view of the ESP server log. Historical log messages are not displayed.

SAS Event Stream Manager displays messages that were logged after you opened the deployment in SAS Event Stream Manager and selected the ESP server.

If you select another ESP server or close the page for the specific deployment, log messages that were displayed in SAS Event Stream Manager are not retained. If you subsequently return to viewing the same deployment and select the same ESP server, only new log messages are displayed.

If you want to keep viewing log messages for an ESP server, keep the page open for the specific deployment.

You can clear the messages in the **Log** tab by clicking **Clear log**. The ESP server continues to log messages, and these new messages appear in the **Log** tab.

Filter Log Messages

To filter log messages by message type, select one or more of the following options in the **Log** tab:

- All – Shows all log message types.
- Informational – Shows general log messages that are not warnings or errors.
- Warning – Shows only warning messages.
- Fatal and Error – Shows only error messages, including fatal errors and normal errors.

View Sample Data

To view sample data for a running project:

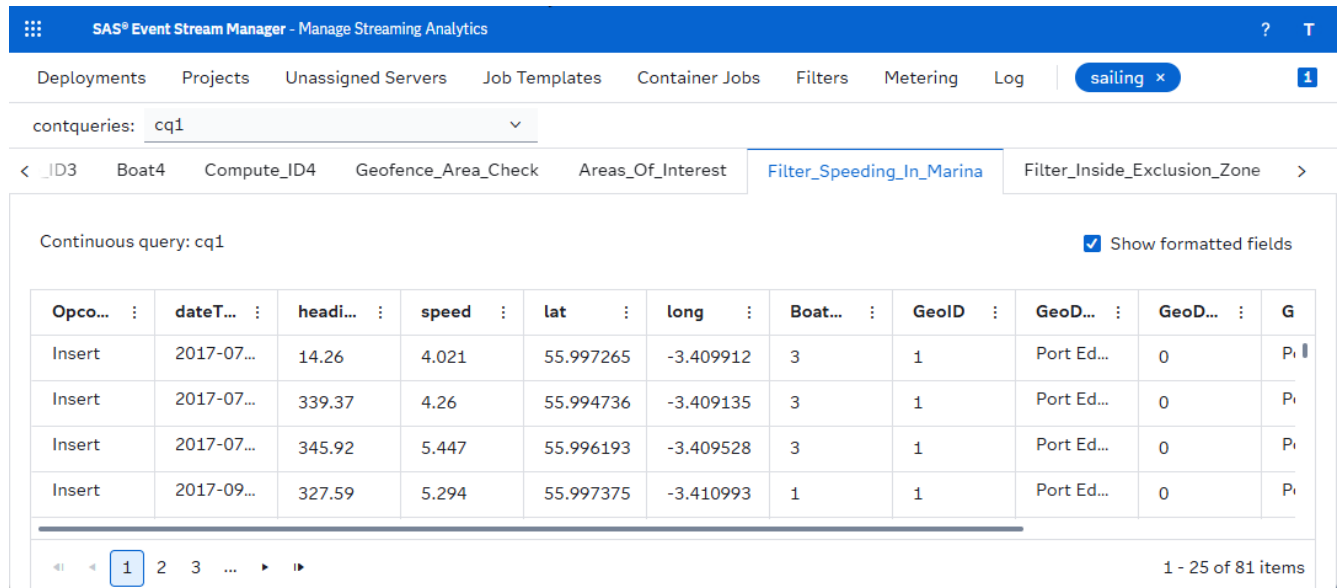
- 1 On the **Deployments** page, select a deployment that has a running project and click .

TIP You can use the **Running Projects** tile on the **Deployments** page to identify deployments that have running projects. For more information, see [“Monitor All Deployments”](#).

A page that displays information about the selected deployment appears.

- 2 If the deployment type is “Cluster”:
 - a In the left pane, expand the relevant project.
A list of the Kubernetes pods for the selected project is displayed.
 - b Right-click a pod and select **Open running project**.
If the project has multiple pods, the sample data is different depending on which pod you select, because events that flow through the project are distributed across multiple pods.
- 3 If the deployment type is “Edge”:
 - a In the main table, select the ESP server that contains the running project.
Tabs appear in the bottom pane.
 - b Right-click a running project in the **Projects** tab in the bottom pane, and select **Open running project**.

A new page opens, where sample data is displayed using tables, with a tab for each window. The following figure shows an example:



The screenshot shows the SAS Event Stream Manager interface. The top navigation bar includes tabs for Deployments, Projects, Unassigned Servers, Job Templates, Container Jobs, Filters, Metering, and Log. A search bar contains 'cq1'. Below the navigation bar, there are tabs for different queries: _ID3, Boat4, Compute_ID4, Geofence_Area_Check, Areas_Of_Interest, Filter_Speeding_In_Marina (selected), and Filter_Inside_Exclusion_Zone. The main content area displays a table for the continuous query 'cq1'. The table has columns: Opco..., dateT..., headi..., speed, lat, long, Boat..., GeoID, GeoD..., GeoD..., and G. The table contains four rows of data, each starting with 'Insert'. The bottom of the interface shows a pagination bar with '1' selected, indicating the first of 81 items.

Opco...	dateT...	headi...	speed	lat	long	Boat...	GeoID	GeoD...	GeoD...	G
Insert	2017-07...	14.26	4.021	55.997265	-3.409912	3	1	Port Ed...	0	Pi
Insert	2017-07...	339.37	4.26	55.994736	-3.409135	3	1	Port Ed...	0	Pi
Insert	2017-07...	345.92	5.447	55.996193	-3.409528	3	1	Port Ed...	0	Pi
Insert	2017-09...	327.59	5.294	55.997375	-3.410993	1	1	Port Ed...	0	Pi

Note: SAS Event Stream Manager displays a real-time view of the data. If your data source contains a limited number of events (for example, the data source is a CSV file rather than a stream of data), the table will be empty after all the events have been displayed.

Beginning in SAS Event Stream Processing 2023.01, when a project contains a large number of windows, some windows might not be displayed when you view sample data. Source windows are prioritized for being displayed.

You can arrange information in the table in several ways. For more information, see [“Arrange Information in Tables”](#).

You can use the **Show formatted fields** check box to choose whether data appears exactly as it was received from the ESP server or with additional formatting. Here are some examples of additional formatting that is applied when the check box is selected:

- Dates are shown as Coordinated Universal Time (UTC) in ISO 8601 format. An example is 2018-11-30T13:33:47.000Z. If you clear the check box, dates appear in UNIX Epoch time, as this is the format in which the data is received from the ESP server.
- A dot is used as a separator in certain types of numerical data, rather than another separator, such as a comma. If you clear the check box and your locale is set to a locale that uses another separator, that separator is displayed instead of a dot.
- Opcodes are displayed using their localized names if your locale is not set to an English-language locale. If you clear the check box, opcodes are always shown in English, as this is how the data is received from the ESP server.

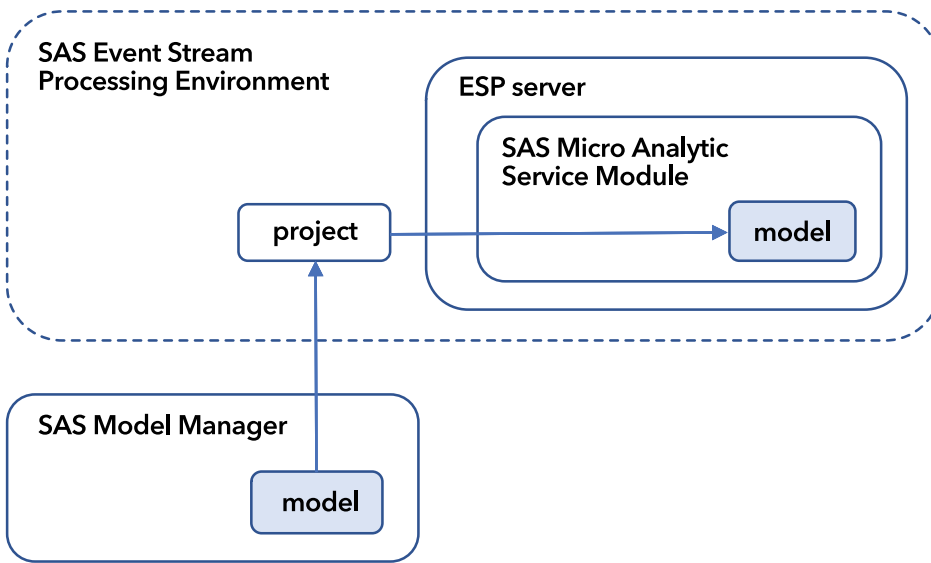
Update SAS Micro Analytic Service Modules

This topic is relevant when a model has been automatically imported into an ESP project. For more information, see [“Overview of SAS Model Manager Integration” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

About SAS Micro Analytic Service Modules

SAS Event Stream Processing projects can refer to models that are stored in the SAS Model Manager common model repository. When a project is deployed, the model is retrieved from the SAS Model Manager common model repository and written to the Kubernetes pod that runs the ESP server. SAS Micro Analytic Service modules are used to accommodate such models. The following figure shows the process.

Figure 18 Model Is Retrieved from SAS Model Manager

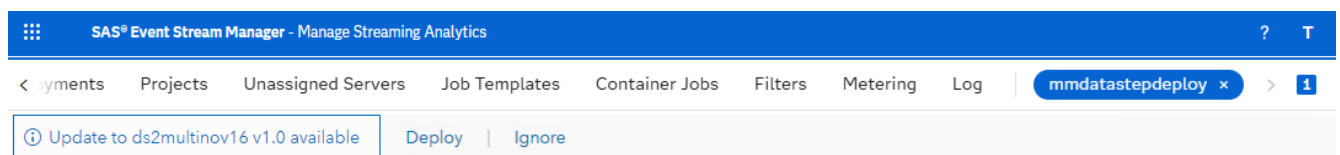


These models can be subsequently updated if a new champion model is declared. If a running project in SAS Event Stream Manager refers to a model stored in the SAS Model Manager common model repository, you can update the running project to reflect updates to the model.

Accept Updates to SAS Micro Analytic Service Modules

If an update is available, a notification bar is displayed on the page for the affected deployment. The following figure shows an example. You can update all projects that are running the specified version of the model, or you can select the projects that you want to update.

Figure 19 Notification Bar with an Update for a SAS Micro Analytic Service Module



To hide the notification for a specific update temporarily, click **Ignore**. The notification for that update is hidden until you return to the **Deployments** page.

To accept the update for one or more projects that are running the specified version of the model:

- 1 Click **Deploy** on the notification bar.

The Update SAS Micro Analytic Service Modules window appears.

- 2 (Edge environment only) Click **All projects running this version** or **Select projects**, as required.

If you clicked **Select projects**, a drop-down list is displayed. Use this drop-down list to select the projects that you want to update. The drop-down list displays all the projects that reference the updated model. For each project, the ESP server that the project is currently deployed to is shown in brackets.

- 3 Click **Update**.

SAS Event Stream Manager fetches and deploys the new champion model for the selected projects.

SAS Event Stream Manager updates the SAS Micro Analytic Service store with the model content and then replaces the SAS Micro Analytic Service module that each selected project is using. The schemas for the existing windows in the project are not altered during this operation. The new champion model must contain function input and output variables that are compatible with the existing schema.

The minor version of the project is updated. For example, the project version is updated from 1.2 to 1.3. The new version is also available in SAS Event Stream Processing Studio.

Monitor Unassigned ESP Servers

You can use the **Unassigned Servers** page to view all ESP servers that are registered with SAS Event Stream Manager and that can be added to a deployment. ESP servers that are already assigned to deployments are not displayed.

The information about the status and properties of ESP servers is similar to the information about the page for a specific deployment. For more information, see [“Monitor a Specific Deployment Whose Type Is “Edge””](#).

ESP servers that run in a Kubernetes cluster are not usually displayed on the **Unassigned Servers** page. However, if you run a project in test mode in SAS Event Stream Processing Studio, the ESP server that is created in the Kubernetes cluster for that project is displayed on the **Unassigned Servers** page for the duration of the test.

In some situations, the  button is enabled on the toolbar so that you can stop a project that is running on an ESP server that runs in a Kubernetes cluster. This action also deletes the ESP server from the Kubernetes cluster.

Monitor Jobs

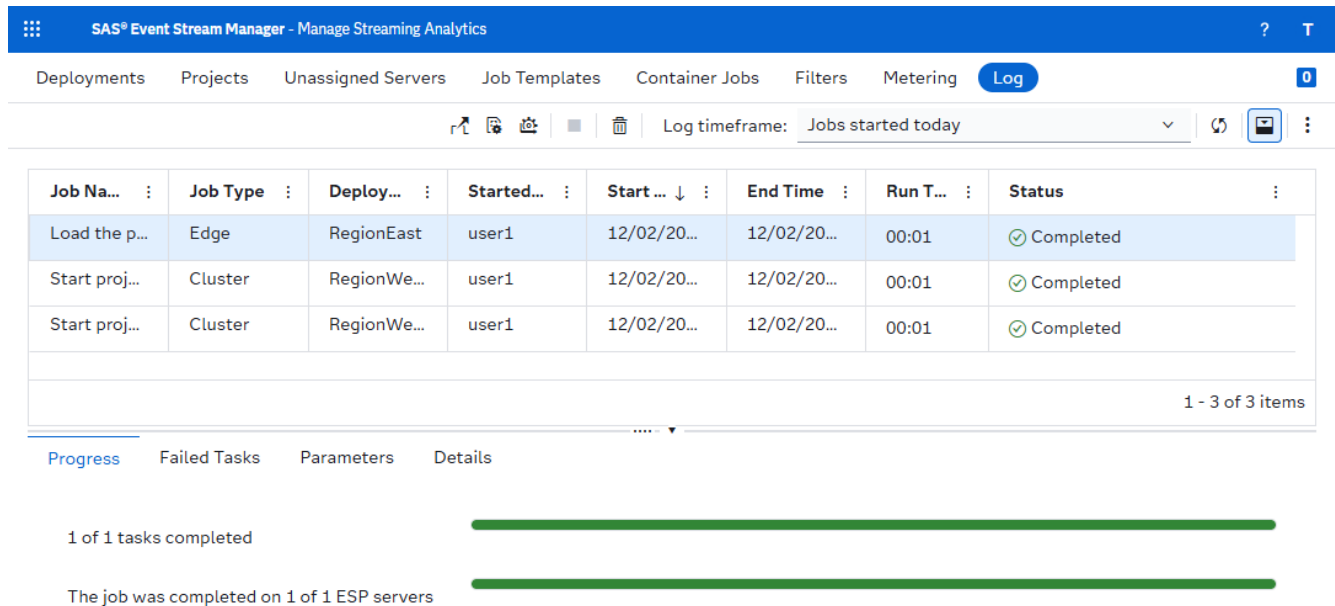
View All Jobs

The **Log** page displays the status of active and historical jobs, except for any jobs that you have deleted from the page. You can use this information to help resolve problems with your deployments. The following information is displayed for each job:

- The job name. The job name reflects the action that created the job. Here is an example of a job name: Start project “ExampleProject” 1.0. If the job was created because a job template was deployed, the name of the job template is displayed as the job name.
- The job type. The job type can be “Cluster” or “Edge”.
- The name of the deployment that the job relates to.
- The user who performed the action that created the job.

- The time at which the job was created.
- The time at which the job was completed
- The time it took for all the tasks in this job to be completed (whether they were completed successfully or not).
- The status of the job.

Figure 20 Log Page



If you navigate away from the **Log** page and subsequently return to it, the page is not refreshed. To refresh the page, click .

If the table contains too much information, you can arrange the information in several ways to find the jobs that you are interested in. For more information, see [“Arrange Information in Tables”](#). In addition, you can use the drop-down menu that has the text **Jobs started today** to select a time period. You can also delete jobs from the table. For more information, see [“Delete a Job”](#).

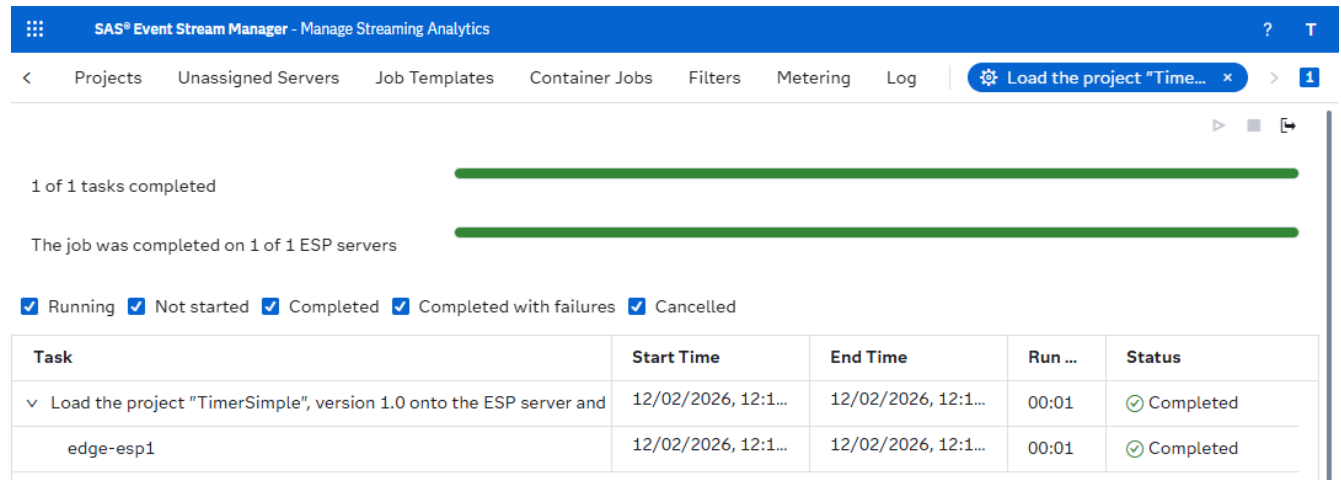
If the job type is “Edge”, you can view additional information for the job. Select the job in the table at the top of the page. The bottom pane displays the following information:

- The **Progress** tab shows how many tasks were completed and whether tasks were completed on all relevant ESP servers.
- The **Failed Tasks** tab lists tasks that were not completed.
- The **Parameters** tab is relevant only if the job was created because a job template was deployed. This tab lists parameters in the job template along with the value entered by the user for each parameter.
- The **Details** tab shows a description of the job and other basic information.

View Job Details

When you deploy a job template, a new page that contains job details appears. A page that contains job details also appears when the deployment type is "Edge" and you load, start, stop, or unload a project with user interface controls. To display the page again, select the job in the main table on the **Log** page and click . The following figure shows an example:

Figure 21 Open Job



The top of the page shows a summary and contains information about how many tasks were completed. When the deployment type is "Edge", the page also shows whether tasks were completed on all relevant ESP servers.

The table on this page shows all tasks for the job, in the order in which the instructions are executed. You can expand items in the **Task** column to view the required level of detail.

You can use the following check boxes to filter the contents of the table: **Running**, **Not started**, **Completed**, **Completed with failures**, and **Cancelled**.

Delete a Job

Deleting a job removes it from the main table on the **Log** page. Deleting jobs can be useful if the table contains jobs that are no longer relevant.

Deleting a job affects only the information that appears in the table; it does not affect job execution. Nevertheless, you cannot delete a job that is still running.

1 Do one of the following:

- To delete a specific job from the **Log** page, select the job and click .
- To delete several jobs, click  and select **Delete all canceled jobs**, **Delete all failed jobs**, or **Delete all successful jobs**.

The Delete Job window appears.

- 2 Confirm the deletion.

Rerun a Job

If a job fails to complete successfully, you can run it again. For example, a job might not complete successfully because an ESP server was unavailable or a data file was not present. After you have resolved the problem, you can rerun the job quickly from the page for that specific job rather than deploying the job template again. You can also rerun jobs that completed successfully.

Rerunning a job creates a new job, as opposed to starting the original job again.

To rerun a job:

- 1 Open the job if it is not already open: on the Log page, select the job and click .

- 2 Click .

The Rerun Job window appears.

- 3 In the **Strategy** field, select your preferred option:

- **Rerun all instructions**
- **Rerun all instructions on servers that failed to complete successfully**
- **Save failed servers to a filter**

Select this option to create a filter that references the failed ESP servers. If you select this option, you must enter a name in the **Filter name** field. You can then reference this filter in another job template that addresses errors or cleans up ESP servers. For example, consider a situation where a project was loaded but a connector was not started because files were not available. A job template that addresses this error might include the `start-connectors` instruction. Consider another situation where a project could not be loaded because it had already been loaded. A cleanup job template might contain the `unload-project` and `stop-project` instructions.

If the original job template needs to be run instead, use the **Rerun all instructions on servers that failed to complete successfully**.

For more information about how you can use filters, see [“Filters”](#).

- 4 If required, select **Skip failed instructions**.
- 5 Click **OK**.

SAS Event Stream Manager reruns the job. The result appears on a new page and also on a new row in the table on the **Log** page.

Stop a Running Job

Stopping a running job can be helpful if, for example, you want to stop a job that is taking a long time to complete. Stopping a running job means canceling the execution of instructions that have not yet

been executed. That is, when you stop a running job, the instruction that is currently being processed is executed, but any subsequent instructions will not be executed.

Stopping a running job does not stop a running project. For more information, see the following topics:

- [“Stop a Project That Is Running on an ESP Server That Runs in a Kubernetes Cluster”](#)
- [“Stop and Unload a Project That Is Running on an ESP Server That Runs on an Edge Server”](#)

To stop a running job:

- 1 Select the job in the main table of the **Log** page and click .
The Cancel Job window appears.
- 2 Click **OK**.

Working with Project Containers

Project Containers

Overview of Project Containers

A *project container* is a lightweight container image that complies with the Open Container Initiative (OCI) and is designed to execute projects in a self-contained, immutable container without external software dependencies. It can be deployed in different environments: in cloud environments, in edge environments, and on premises.

Projects consist of resources that need to be managed, versioned, and deployed as a whole. A project package contains a project XML file and files that the project references. Examples of such files include input files, analytic stores, ONNX models, custom connectors, Python or Lua code files, plug-ins, and parameter files. For more information, see [“Project Package” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#). When you use SAS Event Stream Manager to create a project container, a project package and the minimal software components that are required to run the project are combined into a container image.

Because project containers that are created in SAS Event Stream Manager are OCI-compliant images, they can be deployed in any container deployment flow in your organization's environment.

Benefits of Project Containers

Here are some benefits of project containers:

- enable flexible deployment of images. The same container can run in cloud environments, in edge environments, and on premises.

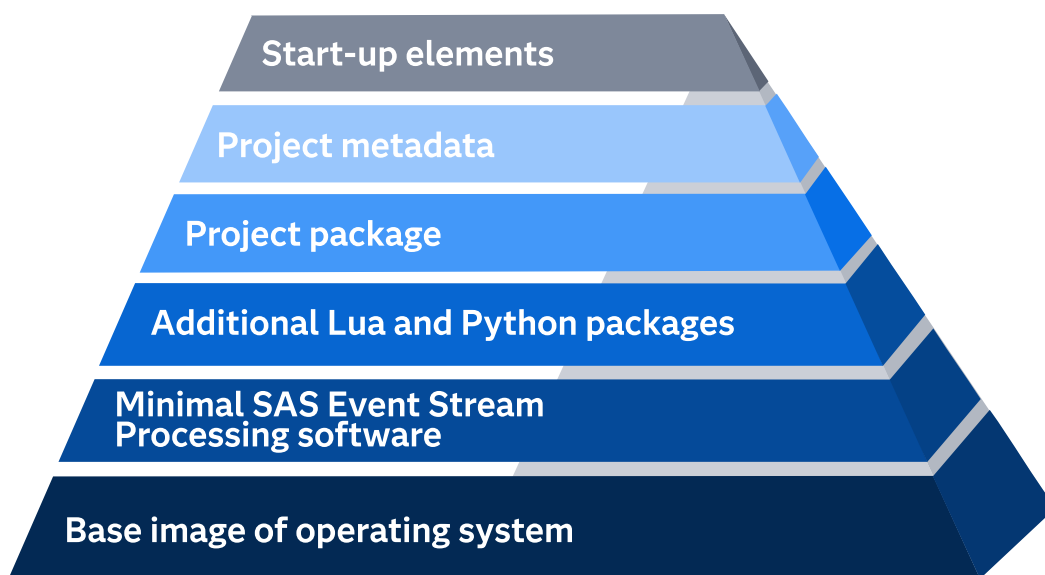
- are self-contained. A container can run without the need to deploy SAS Event Stream Processing or the SAS Viya platform in the environment.
- include only the software that is needed by the project, which reduces the size of the container. For example, projects might not need all the analytics capabilities that SAS Event Stream Processing provides. Libraries that are related to unused analytics capabilities are not included in the image.
- are immutable, which provides consistency. For example, when you deploy an image to a test environment and then deploy the same image to a production environment, the content that is deployed is exactly the same. Similarly, you can deploy the same image to cloud environments, to edge environments, and on premises.
- provide isolation. Containers are isolated from each other and from the host system, which can make security easier to manage.
- provide an easy way to create images by using graphical user interface controls in SAS Event Stream Manager.
- are OCI-compliant container images. That is, they conform to industry standards for container formats and runtimes.

For more information about scenarios in which project containers might be used, see [“Deploying Project Containers”](#).

Structure of Project Containers

When you use SAS Event Stream Manager to create a project container, the following layers are present in the container image.

Figure 22 *Layers in a Container Image*



- The bottom layer contains the base image of an operating system. Other layers are built on top of this layer. The base image contains the minimal operating system components that are required to run SAS Event Stream Processing. The base image is based on Red Hat Enterprise Linux for x86-64 architectures or Arm64 architectures. For more information about the Red Hat Enterprise

Linux release that is used in the base image, see [“Node Requirements” in *System Requirements for the SAS Viya Platform*](#).

- The next layer contains minimal SAS Event Stream Processing software. That is, the minimum libraries that are required to run the project. When you use SAS Event Stream Manager to create an image, the image generation process determines which libraries are required to run the project in question.

The SAS Event Stream Processing release that is used for building the image corresponds to the SAS Event Stream Manager release that you use to create the project container.

- The next layer can contain additional Python libraries and Lua modules that are referenced by a virtual environment. Your project can use the built-in virtual environment, which contains a default set of packages, or a virtual environment that was created by your system administrator. Alternatively, if your project does not contain Python or Lua code, or uses core Python and Lua functionality only, you can set the project to not include Python or Lua dependencies. This option can result in faster image creation as well as a smaller container image. For more information, see [“Update Project Properties” in *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*](#).
- The next layer contains the project package (that is, the project XML file and other files that the project references).
- The next layer contains project metadata. This layer also contains the commands for creating the image.
- The top layer contains environment variables that are required to run the project.

Creating Project Containers

Prepare Your Project

Ensure that your project is suitable for creating a project container:

- 1 In SAS Event Stream Processing Studio, create a project that is part of a project package. Alternatively, when you have an existing project, create a project package for it. For more information, see the following topics:
 - [“Create a Project” in *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*](#)
 - [“Create a Project Package for an Existing Project” in *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*](#)
- 2 Add content to your project as usual and run the project in test mode to check that it works as expected.
- 3 Promote the project to the Test promotion level. For more information, see [“Manage Project Versions” in *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*](#).

If your SAS Event Stream Processing environment has a single SAS Event Stream Manager instance, promoting a project to the Test promotion level makes the project available on the **Projects** page in SAS Event Stream Manager.

- 4 If there are multiple SAS Event Stream Manager instances, in addition to promoting the project to Test, you must import the project to the relevant SAS Event Stream Manager instance to make the project available on the **Projects** page. For more information, see the following topics:
 - For more information about how the number of SAS Event Stream Manager instances affects how you promote projects, see [“Overview of Using SAS Event Stream Manager to Deploy Projects”](#).
 - For more information about how to import a project when there are multiple SAS Event Stream Manager instances, see [“Import a Project”](#).

In addition, before you can create a project container a system administrator must have made a target container registry available. For more information, see [“Add Container Registries”](#).

Obtain Source Images

Before you can create a project container, you must have access to the following source images:

- Operating system base image (`sas-espccr-base`)
- Components image (`sas-espccr-components`): required SAS Event Stream Processing components and system components
- ESP server image (`sas-event-stream-processing-server-app`): the ESP server runtime
- BuildKit image (`sas-buildkit`): the container build service used by SAS Event Stream Manager

When you create a project container for use in Red Hat Enterprise Linux 8 for x86 architectures, SAS Event Stream Manager can pull these source images directly from the SAS Container Registry.

When you create a project container for use in Arm64 architectures, you must first create the Arm64 images and store them in your organization's source container registry. For more information, see [“Manage Arm64 Images”](#).

Storing source images in your organization's source container registry might also be relevant when you want to use an x86 architecture or an Arm64 architecture, and you want to customize an image. For example, assume that you want to create a project container for use in an x86 architecture. SAS provides the operating system base image, which contains the system libraries that are required for SAS Event Stream Processing to run. However, you require other libraries to be present in the base image, so you customize the base image by adding specific drivers.

Note: Customized x86 source images must be derived from images that are obtained from the SAS Container Registry.

Images must have a tag. The tag can be any existing tag (such as `latest`) for that image in the source container registry.

Create a ConfigMap

Note: Creating a ConfigMap is required when you use Arm64 images. Otherwise, creating a ConfigMap is optional.

When you create a project container, you use fields in the SAS Event Stream Manager user interface to specify the location of each source image in the SAS Container Registry or in your organization's source container registry.

Alternatively, you can create and apply a Kubernetes ConfigMap that specifies default locations for the source images. SAS Event Stream Manager reads the default values from the ConfigMap. When you create a project container, you do not need to fill in details of source images in the SAS Event Stream Manager user interface.

Here is example YAML code for a ConfigMap file:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ecr-default-source-images
  namespace: ecr
data:
  ecr-default-source-images.json: |
    {
      "2025.03": {
        "x86_64": {
          "sas-esp-base": "cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-esp-base:stable-2025.03",
          "sas-esp-components": "cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-esp-components-x86_64:stable-2025.03",
          "sas-event-stream-processing-server-app": "cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-event-stream-processing-server-app:stable-2025.03",
          "sas-buildkit": "cr.sas.com/viya-4-x64_oci_linux_2-docker/api-test-base-images/sas-buildkit:stable-2025.03"
        },
        "2025.02": {
          "x86_64": {
            "sas-esp-base": "cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-esp-base:stable-2025.02",
            "sas-esp-components": "cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-esp-components-x86_64:stable-2025.02",
            "sas-event-stream-processing-server-app": "cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-event-stream-processing-server-app:stable-2025.02",
            "sas-buildkit": "esplireg.azurecr.io/api-test-base-images/sas-buildkit:latest"
          },
          "arm64": {
            "sas-esp-base": "esplireg.azurecr.io/api-test-base-arm-images/sas-esp-base-aarch64:latest",
            "sas-esp-components": "esplireg.azurecr.io/viya-4-aarch64_oci_linux_2-docker/sas-esp-components-aarch64:stable-2025.02",
            "sas-event-stream-processing-server-app": "esplireg.azurecr.io/api-test-base-arm-images/sas-event-stream-processing-server-app-aarch64:latest",
            "sas-buildkit": "esplireg.azurecr.io/api-test-base-arm-images/sas-buildkit-aarch64:latest"
          }
        }
      }
    }
```

In the example, two SAS Event Stream Processing releases are specified: 2025.02 and 2025.03.

For the 2025.02 release, both x86_64 and arm64 architectures are specified.

In each case, four source images are specified. In some cases the source image is located in `cr.sas.com` (the SAS Container Registry), and in some cases the source image is located in another source registry.

To specify your organization's source container registry in a ConfigMap, add the secret of your registry in an existing secret in your environment:

- When standalone SAS Event Stream Processing is deployed, add the secret to `cr-secret`.
- When SAS Event Stream Processing is deployed with other SAS Viya platform products, add the secret to `sas-image-pull-secrets`.

The name of the ConfigMap must be `ecr-default-source-images`.

Use the following command to apply the ConfigMap:

```
kubectl -n {k8sNamespace} apply -f {pathToYaml}
```

For example:

```
kubectl -n esi apply -f /home/users/sas/ecrOnAz129/ecr-default-source-images.yaml
```

Create a Project Container

When you create a project container, a container image is created and placed in a target container registry.

To create a project container:

- 1 On the **Projects** page, double-click the project to open it.
- 2 In the **Project Versions** pane, select the project version that you want to include in the project container.
- 3 Click **Create a project container**.
- 4 In the New Project Container window, if required, adjust the name of the image that is created.
- 5 Enter a description for the image. This information can be useful to other users of the image.
- 6 The **ESP server release** field displays the release that is included in the project container, for executing the project.
- 7 Select the architecture: **Intel** (x86-64) or **Arm64**.

Note: When you want to use an Arm64 architecture, you must have already created the Arm64 source images and stored them in your organization's source container registry. For more information, see ["Manage Arm64 Images"](#).

The value of the **Name of destination image** field reflects your choice.

- 8 When the project requires a graphics processing unit (GPU), select the GPU that should be used: **OpenVino** or **NVIDIA**.
- 9 If required, in the **Tags** field, enter identifying keywords that describe the image.
- 10 Select the target container registry in which the image is to be stored.

If a suitable container registry is not listed, a system administrator needs to make a target container registry available. For more information, see [“Add Container Registries”](#).

- 11 If required, adjust the name of the repository subdirectory in which the image is to be stored.

The value of the **Target repository URL** field reflects the details that you specified.

- 12 If required, click **Advanced options** to perform any of the following tasks: select a different source container registry and specify image details in the source container, include the **test_files** folder in the image, and specify a different ESP server release.

Note: When a ConfigMap has been applied, values that you specify in the Advanced Options window override values from the ConfigMap.

- a Select the check box for the source image whose dependencies you want to modify:

- Operating system base image (sas-espccr-base)
- Components image (sas-espccr-components)
- ESP server image (sas-event-stream-processing-server-app)
- BuildKit image (sas-buildkit)

- b Enter the name of the image in the source container registry.

- c Enter the image's tag. The tag can be any existing tag (such as `latest`) for that image in the source container registry.

- d Select the source container registry that contains the source image.

If a suitable container registry is not listed, a system administrator needs to add it. For more information, see [“Add Container Registries”](#).

- e If required, enter the name of the repository subdirectory from which the source image is to be pulled.

The value of the **Source image URL** field reflects the details that you specified.

- f Repeat steps a-e for any other source images whose details you want to modify.

- g If required, select the **Include the “test_files” directory in the image** check box. The **test_files** directory of a project package is intended for storing input files that are used for testing your project. By default, this directory is not included in the image. Transient test files might not be useful in an image that is deployed to a production environment. Including test files increases the size of the image.

- h If required, edit the **ESP server release** field. Specifying a different release overwrites the value that is displayed in the New Project Container window. Ensure that you specify an ESP server release that is compatible with the release of SAS Event Stream Manager that you are using.

- i Click **OK** to close the Advanced Options window.

- 13 Click **OK** to close the New Project Container window.

- 14 The **Container Jobs** page is displayed. Check the status of the job for your image. When the image's status is shown as , the image is available in the target container registry.

The **Pull command** field on the **Details** tab shows the Docker command for pulling the image from the target container registry. You can view log messages that are related to the creation of the project container on the **Log** tab of the **Container Jobs** page. For more information, see [“Use the Container Jobs Page”](#).

When the image has been created, you can perform actions such as deploying the image, setting the image’s validation status, and promoting the image to the next promotion level. For more information, see [“Deploy and Manage Project Containers”](#).

Use the Container Jobs Page

The table on the **Container Jobs** page lists jobs that have been initiated to create project containers. Use the toolbar buttons to monitor these jobs.

Button	Description
	Creates a project container. For more information, see “Create a Project Container” . When you create a project container, a job appears in the table.
	Deletes a job from the table. Use this button when you no longer want to monitor a job and do not want to keep it for reference. Note: This button stops SAS Event Stream Manager from monitoring a job. It does not cancel the creation of a container image and does not delete any container image that has already been created in the container registry.
	Archives a job. Use this button when you no longer want to monitor a job but want to keep it for reference. When you are viewing archived jobs, click this button to unarchive the selected job.
	Shows archived jobs. When you click this button, the table shows archived jobs only. Click the button again to view unarchived jobs.

The following status icons can appear in the table:

Icon	Description
	QUEUED. The job to create a project container is waiting to be processed.
	IN_PROGRESS. The creation of the project container is in progress.
	COMPLETE. The container image is available in the target container registry.
	FAILED. The creation of the project container failed.

When you select a job in the table, the following information is displayed in the bottom pane:

- The **Details** tab provides basic information about the job. For example, this tab shows which options were selected in the New Project Container window. The **Pull command** field shows the Docker command for pulling the image from the target container registry.
- The **Log** tab displays log messages that are related to creation of the project container. The Kubernetes pod that creates the image has containers that are running tasks. Use the **Log** drop-down list to view log messages about these tasks. Messages are available while the image is being created. Afterward, messages are removed from the **Log** tab.

Deploying Project Containers

When Is it Useful to Deploy Project Containers?

Here are example scenarios in which you might choose to deploy container images:

- You have different Kubernetes clusters for development, testing, and production purposes. You use SAS Event Stream Processing in the development cluster to create a project and run it in test mode. When you are satisfied with the project, you create a project container and deploy it to the test cluster for testing purposes. Finally, you deploy the project container to the production cluster. The test and production clusters do not need to have a deployment of SAS Event Stream Processing or the SAS Viya platform.
- You have a requirement to run SAS Event Stream Processing on premises due to security and privacy reasons. However, the development of projects can happen in a cloud environment by using dummy data.

For example, in your project, you might use an MQTT connector in which the host name is provided using an environment variable. You create and test the project in a development environment, and then create a project container. You pass the project container to other team members for use in test and production environments. The team deploys the project container but provides a different host name for the MQTT connector in those environments.

- You want to run a project on an edge device. Running a project container on an edge device is similar to deploying a SAS Event Stream Processing for Edge Computing container. However, you might prefer to use a project container because the size of the container might be smaller because minimal SAS Event Stream Processing software is included in the image. You might also prefer to use a project container when the project depends on other files or libraries. A project container can include additional Python libraries or Lua modules referenced by a virtual environment. When your edge environment does not provide all of the dependencies that are available in the environment where SAS Viya platform is running, using a project container can make the deployment of projects to edge devices more predictable. For more information about requirements for running a project container outside SAS Event Stream Manager, see [“Deploy a Project Container Using Other Tools”](#).
- You create a project container and can deploy it to any environment in which running a container image is supported and suitable tools are available.

Deploy and Manage Project Containers

After you have created a project container, you can perform actions such as deploying the container image, setting the image's validation status, and promoting the image to the next promotion level. For example, you can promote an image from the Test promotion level to the Production promotion level. This is similar to promoting a project version from Test to Production.

IMPORTANT It is recommended that you not deploy untested images to production environments. It is good practice to deploy an image to a test environment first to check that it runs as expected.

Note: If you use SAS Event Stream Manager to deploy a project container in a separated system, the SAS Event Stream Manager instances in the test and production environments must have a connection to the same container registry, so that both instances can access the container image. For more information, see ["Add Container Registries"](#).

To deploy and manage images:

- 1 On the **Projects** page, double-click the project to open it.
 - 2 In the **Project Versions** pane, select the project version whose images you want to manage.
 - 3 In the right pane, click the **Images** tab.
 - 4 Select an image and use the toolbar within the **Images** tab to perform the desired actions:
 - To create another project container for the same project version, click . For more information, see ["Create a Project Container"](#).
 - To deploy the selected image, click . In the Load and Start Project in Cluster window, select the deployment to which you want to deploy the image. If required, you can edit the deployment settings. For more information, see ["Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster"](#).
 - To set the image's status, click . SAS Event Stream Manager does not validate images. Setting the image status enables you to indicate whether an image meets your organization's criteria for a valid image.
 The default value is **Validation required**. Use this status to indicate that your organization has not tested whether the image works as expected.
 Use the **Valid image** status to indicate that the image meets your organization's criteria for a valid image. After you select this status, the  button is enabled and you can promote the image.
 Use the **Invalid image** status to indicate that the image does not meet your organization's criteria for a valid image.
- To promote an image to the next promotion level, click .
 - To demote an image to the previous promotion level, click .

When you demote an image that is already at the lowest promotion level, the image is deleted from SAS Event Stream Manager. However, this action does not delete the container image from the container registry or its respective Kubernetes cluster resources.

Deploy a Project Container Using Other Tools

After you have created a container image, you can use SAS Event Stream Manager to deploy it. For more information, see [“Deploy and Manage Project Containers”](#). Alternatively, you can use your own tools to deploy the container image to your environment.

IMPORTANT It is recommended that you do not deploy untested images to production environments. It is good practice to deploy an image to a test environment first to check that it runs as expected.

The infrastructure into which you plan to deploy an image must meet the following conditions:

- The infrastructure contains an OCI engine or a Kubernetes environment that can run OCI-compliant images.
- The infrastructure reflects the settings that you select when you create the image. For example, if you indicate that the image should use Arm64 and an NVIDIA GPU, the environment in which you deploy the image must contain them.

The project container requires the following parameters to run:

- an `ESPEMV_LICENSE` environment variable that points to a valid SAS Event Stream Processing license.
- (Optional) a folder that is mapped to the `/data` folder in the container.

In project packages, the `ESP_PROJECT_OUTPUT` environment variable can be used to refer to the `output` folder of the project package. For more information, see [“Reference Files That Are Included in a Project Package”](#) in *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*. In a project container, the `ESP_PROJECT_OUTPUT` environment variable points by default to an internal folder called `/data`. To read the contents of the `/data` folder from outside the container, you must map a folder that is outside the container to the `/data` folder.

- any settings or properties that the project relies on. For example, your project might require some environment variables to be set in a certain way for the project to run successfully.

Working with Job Templates

Job Template

A *job template* is an XML document that contains a set of instructions used to create a job. That is, a job template outlines the steps required to execute a project on an ESP server. When you deploy a job template, a job is created.

The job template also specifies what input you must supply when you deploy the job template. For example, when you deploy a job template that contains an instruction to load a project, SAS Event Stream Manager prompts you to select the project that you want to load.

There are two types of job templates:

- A job template whose type is “Cluster” is used for starting and stopping projects and ESP servers in a Kubernetes cluster.

On the Job Templates page in the SAS Event Stream Manager user interface, these job templates are identified in the Type column as “Cluster”. In the job template XML code, these job templates are identified by setting the `job template` element’s `type` attribute to `cluster`.

- A job template whose type is “Edge” is used for interacting with ESP servers that run on edge servers. These job templates can load, start, modify, stop, and unload projects.

On the Job Templates page in the SAS Event Stream Manager user interface, these job templates are identified in the Type column as “Edge”. In the job template XML code, if the `job template` element does not include the `type` attribute, the job template is regarded as a job template whose type is “Edge”.

Depending on the job template’s type, a job template might contain the following elements:

- strings that display localized output
- user parameters that allow a user to enter data into the deployment and customize a job template when it is deployed
- enumerations that restrict user input when used with user parameters
- initializations that replace placeholders to alter a project when it is deployed
- filters that specify a subset of available ESP servers
- instructions that execute or modify projects
- failure instructions that attempt to return the system to normal operation after an instruction has failed to execute

For more information about the elements that are valid in a job template whose type is “Cluster” and in a job template whose type is “Edge”, see [SAS Event Stream Processing: XML Language Reference for Job Templates](#).

In a job template whose type is “Edge”, if the job template contains user parameters, you are prompted to enter each of the user parameters when you deploy the job template. SAS Event Stream Manager displays a list of projects and ESP servers that are available. The string values from a job template’s localizable strings are used to construct labels used when specifying user parameters. After you have selected your user parameters, SAS Event Stream Manager produces and executes a job derived from the job template and the user parameters that you entered. As the job executes, you receive progress updates. The string values from the localizable strings are used to display labels for each instruction.

For examples of job templates, see [“Overview of SAS Event Stream Manager Examples”](#).

As previously explained, job templates contain instructions that execute projects. Some of these actions can be performed only by using instructions in a job template. An example is modifying a running project, which is possible in a job template whose type is “Edge”. However, most actions can also be performed with controls in the SAS Event Stream Manager user interface:

- [“Load and Start a Project on an ESP Server That Runs in a Kubernetes Cluster”](#)
- [“Stop a Project That Is Running on an ESP Server That Runs in a Kubernetes Cluster”](#)
- [“Load and Start a Project on an ESP Server That Runs on an Edge Server”](#)
- [“Stop and Unload a Project That Is Running on an ESP Server That Runs on an Edge Server”](#)

Create a Job Template

- 1 On the **Job Templates** page, click .

The Create New Job Template window appears.

- 2 In the **Name** field, enter a unique name for the job template.
- 3 In the **Description** field, you can enter the purpose of the job template. For example, you can enter the purpose of the job template to enable users to differentiate between job templates with similar names. This description is displayed as a tooltip when you place the cursor over the job template’s name in the Name column on the **Job Templates** page.
- 4 In the **Tags** field, you can attribute one or more tags to the job template.
To add a tag, enter text and press Enter. *Tags* are single-term descriptors for the job template. Tags can be used to group and filter job templates. Tags cannot contain spaces. Duplicate tags are not permitted on a single job template.
- 5 In the **Template type** drop-down list, select the job template’s type. For more information about template types, see [“Job Template”](#).
- 6 In the **Version notes** field, you can add notes to further describe the job template.
- 7 Click **OK**.

Your new job template appears on the **Job Templates** page.

You must now add content to your job template to make it functional. For more information, see [“Edit Job Template Content”](#).

Import a Job Template

- 1 On the **Job Templates** page, click  and select **Import job template**.

The Import Job Template window appears.

- 2 In the **File** field, click **Browse** and navigate to the location of your job template.

Job templates must be in XML format.

- 3 In the **Tags** field, you can attribute one or more tags to the job template.

To add a tag, enter text and press Enter. *Tags* are single-term descriptors for the job template. Tags can be used to group and filter job templates. Tags cannot contain spaces. Duplicate tags are not permitted on a single job template.

- 4 In the **Version notes** field, you can add notes to further describe the job template.

Adding notes is especially useful if you want to track changes between versions of the same job template within SAS Event Stream Manager.

- 5 Click **OK**.

When successfully imported, the job template appears on the **Job Templates** page.

Note: If the job template fails validation, an error message appears. You can use the information in the message to correct the issue, and then import a revised job template.

Edit a Job Template

Edit Job Template Content

- 1 On the **Job Templates** page, select the job template that you want to edit and click .

A separate page appears and displays the XML that defines the job template.

- 2 Edit the content of the job template as required. For more information, see [“Overview of Job Template Content” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

The following table describes the buttons on the XML editor toolbar:

Icon	Action	Keyboard Shortcut
	Reverts your previous change.	Ctrl+Z

Icon	Action	Keyboard Shortcut
	Reverts the effects of the undo action.	Ctrl+Y
	Searches for specific text.	Ctrl+F
	Formats manually entered XML code.	Not available

- 3 Click . Alternatively, you can click  and select **Save template as** to save your changes as a new job template.

Edit Job Template Properties

- 1 On the **Job Templates** page, select the job template that you want to edit and click .
A separate page appears and displays the XML that defines the job template.
- 2 Click .
The Job Template Properties window appears.
- 3 Edit the **Tags** field as required. For more information about this field, see [“Create a Job Template”](#).

Note: You cannot edit the **Name** and **Description** fields.

- 4 Click **OK**.

Deploy a Job Template

- 1 On the **Job Templates** page, select a job template and click .
The Job Template window appears.
- 2 In the **Deployment** field, select the deployment to which you want to deploy the job template.
If you are deploying a job template whose type is “Cluster”, only deployments whose type is “Cluster” are displayed in the **Deployment** field.
If you are deploying a job template whose type is “Edge”, only deployments whose type is “Edge” are displayed in the **Deployment** field. When you select a deployment, the Job Template window is refreshed to display additional fields, which reflect the `parameters` element of the job template that you are deploying. Complete these additional fields.

Note: By default, SAS Event Stream Manager deploys the latest version of the project. To deploy a different version, select a value from the **Project version** drop-down list. You cannot specify the minor version number — instead, SAS Event Stream Manager always uses the latest minor version of the specified major version.

- 3 Click **OK**.

SAS Event Stream Manager creates a job from the job template that you deployed. The result of running the job is displayed on a new page. For more information, see [“View Job Details”](#).

You can also monitor the deployment:

- [“Monitor a Specific Deployment Whose Type Is “Cluster””](#)
- [“Monitor a Specific Deployment Whose Type Is “Edge””](#)

For information about stopping a running job, see [“Stop a Running Job”](#).

Export a Job Template

- 1 On the **Job Templates** page, select the job template that you want to export.
- 2 Click  and select **Export job template**.

The job template is exported to your computer. The location of the exported job template might vary depending on your browser's configuration.

Delete a Job Template

You can delete a job template if there are no running jobs that reference it.

To delete a job template:

- 1 On the **Job Templates** page, select the job template that you want to delete.
- 2 Click .

The Delete Job Template window appears.

- 3 Confirm the deletion.

Working with Filters

Filters

Filtering is available only for deployments whose type is “Edge”.

SAS Event Stream Manager enables you to search for ESP servers that match certain criteria and save those criteria as a filter that you can reference in job templates. For example, you might want to deploy a project to ESP servers that have the tag `primary`. If the number of ESP servers that have this tag changes, a filter that references this tag still finds any matching ESP servers.

A filter contains a filter expression. The filter expression specifies the search criteria for ESP servers.

You can specify that a filter is available only to a specific deployment or that a filter is available for use with all deployments. For example, this means that the filters available to a deployment called `Deployment1` include filters specific to that deployment as well as filters available to all deployments. However, filters specific to another deployment called `Deployment2` are not available to `Deployment1`.

In addition, the SAS Event Stream Manager user interface can show the result of applying a filter. This means that for a given deployment, the user interface lists the ESP servers that match the filter criteria. This functionality is different from applying a filter. A filter is applied when a job template that references the filter is deployed.

You can reference filters from the `server-filters` and `server-filter-selector` elements in job templates. You can also specify a filter expression directly in the `server-filters` element. For more information about the `server-filters` element, see [“server-filters \(Edge\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#). For more information about the `server-filter-selector` element, see [“server-filter-selector” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

In addition to creating filters yourself, there is one situation where SAS Event Stream Manager constructs a filter expression that you can then save as a filter. If a job fails to execute on some ESP servers, SAS Event Stream Manager enables you to save a filter that references those failed servers. For more information, see [“Rerun a Job”](#).

Working with Filters for a Specific Deployment

When you open a deployment, you can work with filters that are available for that specific deployment.

To work with filters that are available for use with all deployments, use the Filters page instead. For more information, see [“Managing Filters”](#).

Show the Result of Applying a Filter

Showing the result of applying a filter means showing only those ESP servers that match a selected filter. This functionality is different from applying a filter. A filter is applied when a job template that references the filter is deployed.

- 1 Open the deployment if it is not already open: on the **Deployments** page, select the deployment that you want to open and click .

A separate page opens to display the deployment.

- 2 Click  on the toolbar and select **Advanced Filtering**.

The **Advanced Filtering** area appears on the page.

- 3 In the drop-down list that displays the text `Enter a filter expression`, either select an existing filter or enter a filter expression. For more information, see [“Filter Expression Syntax”](#).

- 4 Click **Apply** to see what the result of applying this filter would be.

The table on the page changes to display only those ESP servers that match the filter expression.

You can save your filter for further use. For more information, see [“Save a Filter”](#).

To display all ESP servers in the table again, click **Clear**.

Save a Filter

- 1 Complete the steps in [“Show the Result of Applying a Filter”](#).

- 2 Click **Save**.

The Filter Properties window appears.

- 3 Edit the fields as required:

- **Name:** The name of the filter. The name must be unique. To rename an existing filter, enter a new name.

To overwrite an existing filter, select the filter name from the drop-down list. The contents of all fields, except for **Filter expression**, are replaced with the values from the selected existing filter.

- **Description:** A description of the purpose of the filter.

- **Tags:** One or more tags attributed to the filter.

To add a tag, enter text and press Enter. Tags can be used to group and sort filters. *Tags* are single-term descriptors for the filter. Tags cannot contain spaces. Duplicate tags are not permitted on a single filter.

- **Filter expression:** A valid expression that defines the filter criteria. For more information, see [“Filter Expression Syntax”](#).

- **Scope:** Select **One deployment** or **Any deployment**. If you select **One deployment**, the filter is assigned to the current deployment only. You cannot assign the filter to a different

deployment on this page. This functionality is available only when you edit the filter on the **Filters** page. For more information, see [“Edit a Filter”](#).

- 4 Click **OK**.

Managing Filters

The **Filters** page enables you to manage filters that are available in SAS Event Stream Manager. This includes filters that are available for use only with specific deployments as well as filters that are available for use with all deployments.

Sort Filters

You can arrange the information in the table on the **Filters** page in several ways. However, you might find it particularly helpful to sort the table using the Usage column, so that filters available to different deployments or to all deployments are listed together.

For more information about sorting tables, see [“Arrange Information in Tables”](#).

Create a Filter

The **Filters** page enables you to create a filter by entering a filter expression. However, you cannot check the result of applying your filter on this page. This functionality is available only when you open the page for a specific deployment and enter a filter expression there. For more information, see [“Working with Filters for a Specific Deployment”](#).

- 1 Click .

The Filter Properties window appears.

- 2 Edit the fields as required:

- **Name:** The name of the filter. The name must be unique.
- **Description:** A description of the purpose of the filter.
- **Tags:** One or more tags attributed to the filter.

To add a tag, enter text and press Enter. Tags can be used to group and sort filters. *Tags* are single-term descriptors for the filter. Tags cannot contain spaces. Duplicate tags are not permitted on a single filter.

- **Filter expression:** A valid expression that defines the filter criteria. For more information, see [“Filter Expression Syntax”](#).
- **Scope:** Select **One deployment** and then select the desired deployment, or select **Any deployment**.

- 3 Click **OK**.

Edit a Filter

- 1 Click .

The Filter Properties window appears.

- 2 Edit the fields as required. For more information about how to use the fields in this window, see [“Create a Filter”](#).
- 3 Click **OK**.

Delete a Filter

- 1 Click .

The Delete Filter window appears.

- 2 Click **Delete**.

Filter Expression Syntax

Filtering is available only for deployments whose type is “Edge”.

Here are examples of supported filter expressions:

- `hostname='server1234.example.com'`

Finds ESP servers that are running on the host `server1234.example.com`.

- `label~'Test*'`

Finds ESP servers whose names begin with the characters `Test` and are followed by any other characters. For example, this filter finds an ESP server with the name `Test_1` but not `ESP_Test_1` or `test_1`.

Note: If the name of the ESP server contains an apostrophe, enclose the filter term in double quotation marks, and escape the apostrophe with a backslash or an apostrophe. For example, the filter `label~"SiteA\'s*"` finds an ESP server with the name `SiteA's_ESP_server`.

- `((tags='tag1') | (tags='tag2'))`

Finds ESP servers that have the tag `tag1` or `tag2` or both.

- `((tags='tag1') & (tags='tag2'))`

Finds ESP servers that have the tags `tag1` and `tag2`.

- `((tags='tag1') & (tags='tag3') & !(tags='tag5'))`

Finds ESP servers that have the tags `tag1` and `tag3`, but do not have the tag `tag5`.

- `metering=true`
Finds ESP servers whose hosts have metering enabled.
- `version='1.1'`
Finds ESP servers that run with a specific version of SAS Event Stream Processing. Replace `1.1` with the relevant version.
- `ssl=true`
Finds ESP servers with SSL authentication.
- `analyticsLicensed=true`
Finds ESP servers whose hosts have a SAS Event Stream Processing Analytics license installed.

Working with Metering Servers

Metering

SAS Event Stream Manager enables you to monitor your metering servers. This ensures that your production ESP servers are in compliance with the terms of your software license.

You must run at least one metering server to track usage data (that is, event counts) on your production ESP servers.

If SAS Event Stream Processing is running in a Kubernetes cluster, a metering server named `sas-event-stream-processing-metering-app` is created automatically. This metering server monitors all ESP servers that run in the Kubernetes cluster and records all events that occur on those ESP servers.

You do not need to track events on development servers because they do not contribute to the event volume assigned to your software license.

Add a Metering Server

This topic is about adding a metering server if SAS Event Stream Processing is running on an edge server. For more information about metering in a Kubernetes cluster, see [“Metering”](#).

In order to monitor an ESP metering server, you must add it to the list of defined metering servers in SAS Event Stream Manager.

To add a metering server:

- 1 On the **Metering** page, click .

The Metering Server Properties window appears.

- 2 In the **Host** field, enter the host name of the server containing the metering server.
- 3 In the **Metering port** field, either accept the default value (31001) or replace it with the network port defined on the metering server.
- 4 Click **OK**.

The metering server is displayed on the **Metering** page, along with any other metering servers that were added previously.

Monitor a Metering Server

View Metering Server Details

To view details about a metering server, select the required metering server on the **Metering** page.

If SAS Event Stream Processing is running in a Kubernetes cluster, a metering server named `sas-event-stream-processing-metering-app` is listed on the **Metering** page. This metering server monitors all ESP servers that run in the Kubernetes cluster and records all events that occur on those ESP servers.

Note: If the metering server that you want to monitor is not listed, you must add it to SAS Event Stream Manager. For more information, see [“Add a Metering Server”](#).

The Events tile displays each license associated with the selected metering server. It also displays the total number of events recorded for that license for each month or year in the selected time period.

To change the time period, click the drop-down list and select the required time period. For example, select **Show yearly totals**, **Show last 12 months**, or **Show monthly breakdown for year**. The event information is shown for the selected time period.

You can click  to refresh the results in the **Events** tile to reflect any further metered events that have taken place after you selected the metering server.

Expanding each license in the Events tile enables you to view the total number of events for each host that uses the license. For metering servers that run with SAS Event Stream Processing 6.1 or a later release, each host can be expanded to view events for individual servers.

Export Detailed Events Information

You can export detailed information about metering server events to a comma-separated values (CSV) file. The CSV file contains the following events information, with separate rows for each unique combination: license, host, ESP server, server type, event date, event time (for each hour), continuous query, project, and window. The total number of events for the specific hour is displayed in the final column.

To export events information:

- 1 On the Events tile, use the **Metering timeframe** drop-down list to select a time period.
- 2 (Optional) Select a row in the table to specify the license, host, or server that you want to export detailed event information for. If you do not select a row in the table, detailed event information is exported for all licenses, hosts, and servers on the metering server.

Note: All rows within a group are included in the export file. For example, if you select a license, the export file contains details of all hosts that use that license and all servers on each of those hosts.

- 3 Click .
 - 4 In the Export Metering Data for All Licenses window, move the months for which you want to export metering data from the **Months** field to the **Selected months** field.
 - 5 Edit the file name as required and click **OK**.
- The CSV file containing the exported data is downloaded to your computer.

Group Metering Servers

You can arrange the information in the table on the **Metering** page in several ways. However, you might find it particularly helpful to group the table by the **Host** column, so that all ESP metering servers on a particular host are grouped together.

For more information, see [“Arrange Information in Tables”](#).

Managing Settings

Overview of Settings

Use the Settings window to edit user preferences or customize accessibility settings for all SAS web applications. For more information, see [“Settings Window”](#).

Use the **ESP Settings** page to adjust how certain features of SAS Event Stream Processing Studio and SAS Event Stream Manager work. For more information, see [“ESP Settings Page”](#).

Settings Window

Use the Settings window to edit user preferences or customize accessibility settings for all SAS web applications. Changing these settings does not impact other users.

To access these settings, click the user button on the application bar, and select **Settings**.

General

The **General** section includes settings that enable users to change the appearance of the web applications, enable warning and information messages to be displayed, and choose a profile picture. Here are the settings:

- You can change the appearance of the web applications by using the **Theme** setting. The default theme is set by an administrator. The theme specifies the collection of colors, graphics, and fonts that appear in the application.

Select a theme from the drop-down list to change the look of the applications. The theme change takes effect immediately.

SAS themes:

High Contrast

Presents a black background with high-contrast foreground elements to meet the needs of users with low vision.

Dark

Presents an all-dark color palette for a better low-light experience.

Light

Presents a light and clean color palette. The Light theme is the default application theme.

- If you want messages to display that you previously asked not to display, click **Reset Messages**. By default, all warning and information messages are displayed.

Note: This setting relates to messages from other SAS products, rather than messages from SAS Event Stream Processing Studio or SAS Event Stream Manager.

- You can select a profile picture to display in the application bar, as well as in other places within the application that use profile pictures. The default is the first initial or character of your first name.

Note: When standalone SAS Event Stream Processing is deployed, you cannot select a profile picture.

Click **Choose image** and then select an image file to upload. The file size of the image can be up to 1 MB. The valid file types are BMP, GIF, JPEG, JPG, and PNG.

To remove an existing profile picture, click the down arrow and select **Remove image**.

Region and Language

The **Region and Language** section enables users to specify the locale for regional formats and sorting.

The **Locale for regional formats and sorting** setting specifies the locale that is used for sorting data and formatting values such as dates, times, numbers, and currency. By default, the browser locale is used. For example, you might use this setting to display the user interface in English while using a different locale, such as Japanese, to sort and format dates, times, numbers, and currency.

Note: Select the *Browser locale* option only if your browser's locale is one of the supported locales listed in the **Locale for regional formats and sorting** setting. This avoids errors when sorting or formatting dates, times, numbers, and currency.

TIP Changes take effect after you sign out and sign back in.

Notifications

The **Notifications** section enables users to adjust how notifications from other SAS products are displayed.

Select **Show a pop-up message** to display a pop-up message when a notification arrives from another SAS product.

Note: When standalone SAS Event Stream Processing is deployed, the **Notifications** section is not available.

Accessibility

Several settings in the **Accessibility** section can assist people who rely on assistive technologies.

- Select **Adjust the display duration for pop-up notifications** to specify how long temporary pop-up notification messages are displayed. Increasing the amount of time that notifications are displayed can help users discover and read messages that disappear automatically.
- Select **Invert application colors** to make the user interface easier to see for users with sensitivity to certain bright colors. You can also use the Ctrl+` (Ctrl+back quote) keyboard shortcut to invert the application colors.
- Select **Display tooltips when using the keyboard to navigate** to enable keyboard users to access tooltips. When this option is selected, putting keyboard focus on a control also displays the tooltip on the screen. You can also select the location in the browser window to display the tooltip. By default, the tooltip is displayed in the bottom right corner of the browser window.
- The focus indicator is an outline that indicates which user interface component is active. You can make the focus indicator easier to see by selecting **Customize the focus indicator** and adjusting its color, thickness, and opacity.

ESP Settings Page

The **ESP Settings** page enables you to complete tasks such as adjusting how certain features of SAS Event Stream Processing Studio and SAS Event Stream Manager work.

To access the **ESP Settings** page, click the user button on the application bar, and select **Settings**.

The content that is available on the **ESP Settings** page in SAS Event Stream Processing Studio is different from the content that is available on the **ESP Settings** page in SAS Event Stream Manager. The content in each application relates to tasks that you carry out in these applications.

The content that is available to you on the **ESP Settings** page depends on whether you are a member of the administrator group. By default, the administrator group is SASAdministrators.

Global Settings

Overview of Global Settings

The **Global** section of the **ESP Settings** page enables you to modify settings that affect all users of the application. An exception to this is the **Theme** setting, which applies only to your browser.

Note: Only members of the administrator group can access the **Global** section. By default, the administrator group is SASAdministrators.

Restart the Client-Configuration Server

Under certain circumstances, you might need to restart the client-configuration server. For more information, see [“Understanding and Managing the Client-Configuration Server” in SAS Event Stream Processing: Using SAS Event Stream Processing in a Kubernetes Environment](#).

IMPORTANT Restarting the client-configuration server affects all users of the application.

- 1 On the **ESP Settings** page, in the **Global** section, select **Global Settings for ESP Servers**.

Note: Only members of the administrator group can access the **Global Settings for ESP Servers** page. By default, the administrator group is SASAdministrators.

The right pane displays the **Global Settings for ESP Servers** page.

- 2 Click .

The **Status** field displays the following message: Restarting the client-configuration server.

The restart is complete when the **Status** field displays the following message: The client-configuration server is running.

Include or Exclude Connectors

You can use the **Connectors** tab on the **Global Settings for ESP Servers** page to modify ESP server properties for connectors. These ESP server properties control which connectors are available to users of SAS Event Stream Processing Studio in the Connector Configuration window.

Note: Only members of the administrator group can access the **Global Settings for ESP Servers** page. By default, the administrator group is SASAdministrators.

TIP To access the Connector Configuration window:

- 1 Open a project in SAS Event Stream Processing Studio. Select a Source window or add a new Source window.
- 2 In the right pane, expand **Input Data (Publisher) Connectors or Subscriber Connectors**.
- 3 Click . The **Connector Configuration** window appears. The **Connector type** field displays available connector types.

For information about alternative ways to control which connectors are displayed, see [“Excluding Connectors” in SAS Event Stream Processing: Connectors and Adapters](#).

IMPORTANT Any changes to the **Global Settings for ESP Servers** page affect all users of the application. To change settings for the user session instead, on the **ESP Settings** page, in the **User Session** section, select **ESP Server Properties**.

The process for including or excluding connectors includes an automatic restart of the client-configuration server.

The **Connectors** tab enables you to change ESP server properties for connectors, but not other ESP server properties. To change other ESP server properties for all users of the application, select the **Properties** tab.

To include or exclude connectors:

- 1 On the **ESP Settings** page, in the **Global** section, select **Global Settings for ESP Servers**.
The right pane displays the **Global Settings for ESP Servers** page.
- 2 Select the **Connectors** tab.
- 3 Select check boxes for the connectors that you want to be available in the Connector Configuration window, and clear check boxes for connectors that you do not want to be available.
- 4 Click .
The Save Changes window appears.
- 5 Click **Save**.

The client-configuration server is restarted. Your changes are complete when the **Status** field displays the following message: The client-configuration server is running.

You can also reset all ESP server properties for connectors to their defaults.

- 1 Click .

The Reset ESP Server Properties for Connectors window appears.

- 2 Click **Reset**.

The client-configuration server is automatically restarted.

Specify Environment Variables and ESP Server Properties for All Users

You can use the **Environment Variables** tab and the **Properties** tab on the **Global Settings for ESP Servers** page to add, edit, or delete environment variables and ESP server properties for all users. The changes take effect immediately and all new ESP servers that are created in the Kubernetes cluster use your latest settings. If you want your changes to be applied to the client-configuration server, you must restart the client-configuration server.

Note: Only members of the administrator group can access the **Global Settings for ESP Servers** page. By default, the administrator group is SASAdministrators.

IMPORTANT Any changes to the **Global Settings for ESP Servers** page affect all users of the application. To change settings for the user session instead, on the **ESP Settings** page, in the **User Session** section, select **Environment Variables** or **ESP Server Properties**.

To specify environment variables and ESP server properties for all users:

- 1 On the **ESP Settings** page, in the **Global** section, select **Global Settings for ESP Servers**.
The right pane displays the **Global Settings for ESP Servers** page.
- 2 Select the **Environment Variables** tab or the **Properties** tab.
- 3 In the table, specify settings as required:
 - To add a setting, click . A window appears, prompting you to specify the name of the setting and the value of the setting.
 - To edit a setting, click the setting's **Value** field and specify a new value. Then click . A window appears, prompting you to confirm that you want to save changes.
 - To delete a setting, click the row that contains the setting and then click . A window appears, prompting you to confirm the deletion.

The changes take effect immediately and all new ESP servers that are created in the Kubernetes cluster use your latest settings.

- 4 If you want your changes to be applied to the client-configuration server, click .

The **Status** field displays the following message: Restarting the client-configuration server.

The restart is complete when the **Status** field displays the following message: The client-configuration server is running.

Specify Connections to Remote Git Servers

The **Remote Git Server Connections** page enables you to add, update, and delete connections to remote Git servers. After remote Git servers have been defined, you can synchronize project data from the Git repository that is internal to SAS Event Stream Processing to remote Git repositories. For more information, see [“Overview of File-Based Versioning” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

Note: Only members of the administrator group can access the **Remote Git Server Connections** page. By default, the administrator group is SASAdministrators.

CAUTION

Specifying a remote Git server connection is a task that must be performed by a system administrator. This task involves specifying a Git access token and specifying a topic to organize repositories. Because project data is stored on a remote Git server, you must ensure that backup processes are in place for the remote Git server. Not having backups in place could result in data loss. Use only SAS Event Stream Processing to interact with remote Git repositories. Interacting directly with a repository that stores a SAS Event Stream Processing project on your remote Git server could result in data loss.

Note: To use external services such as GitHub, GitLab, and Gitea with SAS Event Stream Processing, the Kubernetes cluster administrator must ensure that there is access from the cluster to the appropriate IP addresses.

To specify a connection to a remote Git server:

- 1 On the **ESP Settings** page, in the **Global** section, select **Remote Git Server Connections**.
The right pane displays the **Remote Git Server Connections** page.
- 2 Click .

The Add Remote Git Server Connection window appears.

- 3 Enter a name to describe the connection. This name will be displayed elsewhere in the user interface. For example, when a user creates a project and chooses to store it in a remote repository, the name that you enter in the Add Remote Git Server Connection window is displayed as one of the remote Git servers on which the user can store the project.
- 4 Select the server type:
 - **GitHub**
 - **GitHub Enterprise Cloud:** GitHub Enterprise that is hosted by GitHub on GitHub.com.
 - **GitHub Enterprise Server:** GitHub Enterprise that is self-hosted.

- **GitLab**

- **Gitea**

- 5 If you selected GitLab, GitHub, or Gitea as the server type, specify who has visibility of the repositories that are created on this Git server.
 - If you are using GitLab, you can set visibility to **Private**, **Internal**, or **Public**. For more information about the meaning of these settings in GitLab, see [GitLab documentation](#).
 - If you are using GitHub or Gitea, you can set visibility to **Private** or **Public**. For more information about the meaning of these settings in GitHub, see [GitHub documentation](#).
 - 6 Enter the server URL. If you selected GitHub or GitHub Enterprise Cloud as the server type, the URL is entered for you.
 - 7 Enter your user name.
 - 8 Enter the value of your access token instead of the name of the access token. For example, an access token for GitLab might have the following format: `glpat-xxxxxxxxxxxxxxxxxxxx`. (In this example, some characters have been changed to `x`.)
 - If you are using GitLab, the access token must be a personal access token. The access token's scope must include `api`. For more information about creating access tokens in GitLab, see [GitLab documentation](#).
 - If you are using GitHub, the access token must be a classic access token. The access token's scope must include `repo`. For more information about creating access tokens in GitHub, see [GitHub documentation](#).
 - If you are using Gitea, the access token's scope must include `repo` and `user`. For more information about access tokens in Gitea, see [Gitea documentation](#).
 - 9 When the server type is Gitea, you can enable Git Large File Storage (LFS). For more information, see ["File-Based Versioning and Git Large File Storage" in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#). To configure your remote Git server connection to use Git LFS:
 - a Contact your Gitea administrator and obtain the value of the `LFS_JWT_SECRET` field from the Gitea `app.ini` file. For more information about the `LFS_JWT_SECRET` field in Gitea, see [Gitea documentation](#).
 - b Enter the value in the **Gitea LFS secret** field in the Add Remote Git Server Connection window.
- IMPORTANT** If Gitea restarts, then the value of the `LFS_JWT_SECRET` field is regenerated. You must update the value in the **Gitea LFS secret** field.
- 10 If you selected GitLab, GitHub, or Gitea as the server type, you can enter the name of a topic. Here is an example: `esp-project-package`. Topics are labels that are used to organize repositories. Specifying a topic is optional.
 - The topic that you specify here is used when a project is stored in a remote repository. That is, when a remote repository is created to store a project, the repository has the topic that you specify here.

- The topic that you specify here is also used when a project is imported from a remote repository. That is, when you import a project from a remote Git server, only those repositories that have the specified topic are listed as available repositories.
- For more information about topics in GitLab, see [GitLab documentation](#). For more information about topics in GitHub, see [GitHub documentation](#).

11 Click **Test connection** to check that your credentials work.

12 Click **OK**.

When an access token expires, if a new access token is created you can edit the connection to update the value of the access token: click  and enter the new access token. You cannot edit other connection properties.

To delete a connection, click .

When projects are synchronized with repositories on remote Git servers, projects can be deleted from the persistent volume to free up storage space. For more information, see [“Clean Up Projects from the Persistent Volume”](#).

Add Container Registries

The **Container Registries** page enables you to add, update, and delete connections to repositories in container registries. Container registries are needed when you want to use the functionality for project containers. For more information, see [“Project Containers”](#).

Note: When SAS Event Stream Processing is deployed with other SAS products, only members of the administrator group can access the **Container Registries** page. By default, the administrator group is SASAdministrators. When standalone SAS Event Stream Processing is deployed, any user can access this page.

Use the **Container Registries** page to add connections to the following registries:

- your organization's source container registries (that is, registries from which you want to pull base images of operating systems)

Note: You can obtain base images from the SAS Container Registry. However, when you use SAS Event Stream Manager to create a project container, SAS Event Stream Manager cannot pull base images directly from the SAS Container Registry. As a result, base images must be stored in your organization's source container registry.

- your organization's target container registries (that is, registries in which you want to store container images that you create using SAS Event Stream Manager)

Note: The following registries are supported: Azure Container Registry and Harbor.

When SAS Event Stream Processing is deployed with other SAS products and you use Harbor, Transport Layer Security (TLS) is supported. For more information, see [“Certificate Generators” in SAS Viya Platform Encryption: Data in Motion](#). When standalone SAS Event Stream Processing is deployed, you can use Harbor but TLS is not supported.

CAUTION

Adding container registries is a task that must be performed by a system administrator.

To specify a connection to a container registry:

- 1 On the **ESP Settings** page, in the **Global** section, select **Container Registries**.
The right pane displays the **Container Registries** page.
- 2 Click .
- 3 In the Add Container Registry window, enter a name to describe the connection. This name is displayed elsewhere in the user interface. For example, when a user is creating an image and selects a target registry, the name that you enter in the Add Container Registry window could be displayed as one of the options.
- 4 Enter a description of the container registry.
- 5 Enter the URL of the container registry.
- 6 Enter the name of a repository within the container registry for use as the default repository.
- 7 Enter the user name and password for the container registry, and then click **OK**.

When you have added connections to at least one source container registry and one target container registry, you can then create project containers. For more information, see [“Creating Project Containers”](#).

If you need to edit the details of an existing connection, click .

To delete a connection, click .

Settings for the User Session

Overview of Settings for the User Session

The **User Session** section of the **ESP Settings** page enables you to change event stream processing settings for your user session.

You can add, edit, reset, and delete the value of specific environment variables and specific ESP server properties (including logging properties) that are used when an ESP server is created in a Kubernetes cluster. An ESP server is created when you deploy a project to a Kubernetes cluster in SAS Event Stream Processing Studio or SAS Event Stream Manager, or test a project in SAS Event Stream Processing Studio.

IMPORTANT Any changes to these settings affect the current user only. To change settings for all users of the application, on the **ESP Settings** page, in the **Global** section, select **Global Environment Variables** or **Global ESP Server Properties**.

The changes that you make in the **User Session** sections on the **ESP Settings** pages of SAS Event Stream Processing Studio and SAS Event Stream Manager are independent of each

other. To use the same settings in both applications, you must configure the settings in both applications.

Specify Environment Variables and ESP Server Properties for the User Session

To adjust environment variables and ESP server properties (including logging properties):

- 1 On the **ESP Settings** page, in the **User Session** section, select **Environment Variables** or **ESP Server Properties**.

The right pane displays the **Environment Variables** page or the **ESP Server Properties** page.

- 2 In the table, specify settings as required:
 - To add a setting, click . The Add Environment Variable or Add ESP Server Property window appears. Use this window to specify the name of the setting and the value of the setting.
 - To edit a setting, click the setting's **Value** field and specify a new value. Then click . The Save Changes window appears, prompting you to confirm that you want to save changes.
 - To delete a setting, click the row that contains the setting and then click . The Delete Environment Variable or Delete ESP Server Property window appears, prompting you to confirm the deletion.
 - To reset all settings to default values, click . The Reset Environment Variables or Reset ESP Server Properties window appears, prompting you to confirm the resetting.

Here is additional information about some properties that are present on the **ESP Server Properties** page by default:

- `server.disableTrace` – When the value of this setting is false, you can enable trace-level logging for specific windows in a project. Open a project in SAS Event Stream Processing Studio and click  to view settings for the continuous query. In the **Debugging** section, you can select the windows for which you want to enable trace-level logging.
- `server.maxws-queue` – Use this property to specify the maximum number of megabytes that the ESP server keeps in a WebSocket event queue before it starts to drop old events. You do not normally need to alter the value of this property. However, if you see warnings in the ESP server log about events being discarded, consider increasing the value of this property or slowing the input rate of events. For more information, see [“Property to Set the Size of Event Queues” in SAS Event Stream Processing: Using SAS Event Stream Processing in a Kubernetes Environment](#).

Specify Whether to Run Projects as the Current User

When standalone SAS Event Stream Processing is deployed, projects always run in the Kubernetes cluster as the system user. You cannot change this.

When SAS Event Stream Processing is deployed with other SAS products, in some circumstances it is possible to specify how projects run in the Kubernetes cluster:

- When a project is part of a project package, the project always runs in the Kubernetes cluster as the system user.
- By default, project XML files run in the Kubernetes cluster with the access rights and permissions of the current user. You can adjust this setting to select which group's permissions are used for the current user or to run project XML files as the system user instead. For more information about situations in which you might want to run projects as the system user, see [“Understanding File Access” in SAS Event Stream Processing: Troubleshooting](#).

To specify how project XML files should run in the Kubernetes cluster:

- 1 On the **ESP Settings** page, in the **User Session** section, select **ESP Server Properties**.
The right pane displays the **ESP Server Properties** page.
- 2 Select the desired setting:
 - **Run project XML files as the system user.** By default, the system user is the sas user account.
 - **Run project XML files as the current user.** The group whose permissions are used is the group ID (GID) from the current user's LDAP record. When several groups are available, you can select the **Override group** check box, and then select a different group.

Note: These settings affect your user session only. They do not apply globally.

- 3 Click .

ESP Operator

Overview

The **ESP Settings** page provides an alternative way to access the ESP operator log. This can be useful if a pod does not start and you cannot access log messages in test mode in SAS Event Stream Processing Studio or on the page for a specific deployment in SAS Event Stream Manager.

Access the ESP Operator Log

On the **ESP Settings** page, in the **ESP Operator** section, select **ESP Operator Log**.

The right pane displays messages from the ESP operator log.

Filter Log Messages

To filter log messages by message type, select one or more of the following options:

- All – Shows all log message types.
- Informational – Shows general log messages that are not warnings or errors.

- Warnings – Shows only warning messages.
- Fatal and Error – Shows only error messages, including fatal errors and normal errors.

Clear the ESP Operator Log Page

To clear the contents of the **ESP Operator Log** page, click  **Clear Log**.

Export the ESP Operator Log

You can export the ESP operator log to a text file. Click  **Export All Logs**. The log is exported to your computer. The location of the log that you exported might vary depending on your browser's configuration.

Search within Log Messages

Use the search field to search the log messages for a specific term. The search field works in conjunction with the log filters. For example, if you click **Warning** and then search for a specific term, the search finds occurrences of that term within warning messages only. The search is not case sensitive.

Tools

Overview of Tools

The **Tools** section of the **Settings** page enables you to encrypt a password in an adapter, clean up project files from the persistent volume, delete SAS Micro Analytic Service stores, and manage Arm64 images.

Encrypt a Password in an Adapter

Some adapters require you to enter a password in the connection's configuration details. SAS Event Stream Processing Studio and SAS Event Stream Manager enable you to encrypt passwords in adapters.

- To encrypt the password in a connector, use the Connector Configuration window in SAS Event Stream Processing Studio. For more information, see [“Encrypt a Password in a Connector” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).
- To encrypt the password in an adapter, including the alternative transports for RabbitMQ and Solace, use the password encryption tool on the **ESP Settings** page in SAS Event Stream Processing Studio or SAS Event Stream Manager.

To use the password encryption tool on the **ESP Settings** page:

- 1 On the **ESP Settings** page, in the **Tools** section, select **Encrypt Password**.
- 2 Select the adapter.
- 3 Enter the user ID.
- 4 Enter the password that you want to encrypt and click **Encrypt password**.
- 5 Click **Copy** to copy the encrypted password to the clipboard.
- 6 Use the copied password when you connect to the data source.

For example, when you are using the MQTT adapter, use the encrypted password as the value for the `mqttpassword` key. Depending on the adapter, you might need to set other relevant keys too. For example, for the MQTT adapter, you also need to set the `mqttpasswordencrypted` key to `true`.

Here is another example. For the Database adapter, use the encrypted password to create the system data source name (DSN).

```
DSN=Oracle Wire Protocol;uid=exampleuserid;pwd=pkrrt08hAlzcgkUUr05zRhgzjePLR0HgqQYWtyLr5W70yUjT3Wf1Eo72VGd4i6s3;hostname=example_DB_server_name.com;servicename=example.process_name"
```

For more information about the connection details for adapters, see [“Adapters by Source Language” in SAS Event Stream Processing: Connectors and Adapters](#).

For more information about the connection details for the alternative transports for RabbitMQ and Solace, see [“Using Alternative Transports for Java Clients” in SAS Event Stream Processing: Publish/Subscribe API Reference](#).

Clean Up Projects from the Persistent Volume

When projects are synchronized with repositories on remote Git servers, projects can be deleted from the persistent volume to free up storage space. The projects remain on the remote Git servers and are restored to the persistent volume if the projects are reopened.

Note: Only members of the administrator group can access the **Clean Up Persistent Volume** page. By default, the administrator group is SASAdministrators.

To delete projects from the persistent volume:

- 1 On the **ESP Settings** page, in the **Tools** section, select **Clean Up Persistent Volume**.
Only projects that are synchronized with repositories on remote Git servers and that are not running are listed on this page.
- 2 Select the projects that you want to delete and click .
- 3 In the Clean Up Persistent Volume window, click **Yes**.

Delete SAS Micro Analytic Service Stores

This topic is relevant for SAS Micro Analytic Service stores that are associated with project XML files or that were previously associated with project XML files. This topic is not relevant for project packages.

On the **Settings** page, in the **Tools** section, the **SAS Micro Analytic Service Stores** page contains two tables.

The first table shows project XML files that have SAS Micro Analytic Service stores. Here is an example of a situation in which you might want to delete SAS Micro Analytic Service stores that are associated with a project: when there are several versions of the same project, you might want to delete stores for earlier project versions to save storage space.

The second table shows orphaned SAS Micro Analytic Service stores. These stores no longer belong to any project XML files.

Here are situations in which SAS Micro Analytic Service stores are automatically deleted and there are no orphaned stores:

- A member of the administrator group uses SAS Event Stream Processing Studio to delete a project XML file that has SAS Micro Analytic Service stores and selects the option to also delete the project from SAS Event Stream Manager. By default, the administrator group is SASAdministrators.
- Any user of SAS Event Stream Manager uses SAS Event Stream Manager to delete a project XML file that has SAS Micro Analytic Service stores.

Here are examples of situations in which you might want to delete orphaned stores:

- When a user who is not an administrator uses SAS Event Stream Processing Studio to delete a project, stores that are associated with the project are not automatically deleted. The stores are orphaned.
- When the automatic deletion of stores has failed for some reason.
- When a user deletes a SAS Micro Analytic module from a project XML file, the reference to the SAS Micro Analytic Service store is deleted, but the store is not. The store is orphaned.

To delete SAS Micro Analytic Service stores:

- 1 On the **Settings** page, in the **Tools** section, select **SAS Micro Analytic Service Stores**.
- 2 Select the stores that you want to delete and click .

A window appears, prompting you to confirm the deletion.

The deletion takes place in the background and can take some time because an ESP server is created to perform the deletion. The ESP server runs with the access rights and permissions of the current user. When the deletion is complete, a confirmation message is displayed.

Manage Arm64 Images

When you are working with project containers, you can use the **Manage Arm64 Images** page to create Arm64 source images. For more information about source images that are needed to build project containers, see [“Obtain Source Images”](#). The **Manage Arm64 Images** page creates two of the four Arm64 source images that you require. Use a ConfigMap to specify default locations for the other two images, as detailed in the following steps.

Note: Only members of the administrator group can access the **Manage Arm64 Images** page. By default, the administrator group is SASAdministrators.

- 1 To use project containers in an Arm64 environment, an Arm-based node pool must be available in the Kubernetes cluster in which SAS Event Stream Manager is running. Set the node pool properties to suit your requirements, except for the label and the taint, which must be set as follows:
 - For the label, add a key called `arm-key` and set its value to `arm-value`.
 - For the taint, add a key called `arm-key` and set its value to `arm-value`. Set the effect to `NoSchedule`.
- 2 Create and apply a Kubernetes ConfigMap that specifies default locations for the `sas-espcr-components-aarch64` and `sas-buildkit-aarch64` source images. For general information about ConfigMaps, see [“Create a ConfigMap”](#).

Here is example YAML code for a ConfigMap file. Replace `2025.04` with the relevant ESP server release.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ecr-default-source-images
  namespace: ecr
data:
  ecr-default-source-images.json: |
    {
      "2025.04": {
        "arm64": {
          "sas-espcr-components": "cr.sas.com/viya-4-aarch64_oci_linux_2-docker/sas-espcr-components-aarch64:stable-2025.04",
          "sas-buildkit": "cr.sas.com/viya-4-aarch64_oci_linux_2-docker/sas-buildkit:stable-2025.04"
        }
      }
    }
```

- 3 In SAS Event Stream Manager, on the **Settings** page, in the **Tools** section, select **Manage Arm64 Images**.
- 4 To create an Arm64 image, your organization must have a container registry where you can store the image. If you have not previously added connections to container registries, an error is displayed. Click **Create a container registry** to configure a connection, and then return to the **Manage Arm64 Images** page. For more information, see [“Add Container Registries”](#).
- 5 Click **Create source images required for Arm64-based project containers**.
- 6 In the New Arm64 Images window, specify the relevant ESP server release. The ESP server release must be the same as in the ConfigMap that you created in step 1.
- 7 Select the container registry in which the image is to be stored.
- 8 If required, specify a repository subdirectory in which the image is to be stored. Specifying a repository subdirectory enables you to group images together.
- 9 Click OK.

The `sas-espcr-base-aarch64` and `sas-event-stream-processing-server-app-aarch64` source images are created. You can check the progress of the image creation by checking the **Job status** field on the **Manage Arm64 Images** page.

- 10 When the `sas-esp-cr-base-aarch64` and `sas-event-stream-processing-server-app-aarch64` source images have been created, update the ConfigMap to include their locations. Here is an example of YAML code for the ConfigMap file. Replace `2025.04` with the relevant ESP server release. Replace `<your.container.registry.url>` and `<your.repository.subdirectory>` with values that you specified in the New Arm64 Images window.

```
ecr-default-source-images.json: |
{
  "2025.04": {
    "arm64": {
      "sas-esp-cr-components": "cr.sas.com/viya-4-aarch64_oci_linux_2-docker/sas-
esp-cr-components-aarch64:stable-2025.04",
      "sas-buildkit": "cr.sas.com/viya-4-aarch64_oci_linux_2-docker/sas-
buildkit:stable-2025.04"

      "sas-esp-base": "<your.container.registry.url>/
<your.repository.subdirectory>/sas-esp-cr-base-aarch64:stable-2025.04"
      "sas-event-stream-processing-server-app": "<your.container.registry.url>/
<your.repository.subdirectory>/sas-event-stream-processing-server-app-
aarch64:stable-2025.04"
    }
  }
}
```

Examples

Overview of SAS Event Stream Manager Examples

The following examples demonstrate how to use SAS Event Stream Manager:

- The Filter example demonstrates how to deploy a project to a Kubernetes cluster. For more information, see [“Overview of the Filter Example”](#).
- The Activity Tracker example demonstrates how to deploy a project using a job template whose type is “Edge”. To follow this example, you need an ESP server that runs on an edge server. For more information, see [“Overview of the Activity Tracker Example”](#).
- The GitHub Actions Workflow example demonstrates how to automate the promotion of a SAS Event Stream Processing project to a test environment using a GitHub Actions workflow. For more information, see [“Overview of the GitHub Actions Workflow Example”](#).

Filter Example

Overview of the Filter Example

This example demonstrates how you can use SAS Event Stream Manager to deploy a project to a Kubernetes cluster. The project used in this example filters temperature readings. For more information about the project, see the `filter` example in [SAS Event Stream Processing Studio Examples GitHub](#).

The instructions assume that you are using a Kubernetes environment. That is, you are not testing the project on an ESP server that runs on an edge server.

By following this example, you learn how to do the following tasks. Some of these tasks are performed in SAS Event Stream Processing Studio and some tasks are performed in SAS Event Stream Manager.

- install an example
- make a project available to SAS Event Stream Manager
- create a deployment
- deploy a project
- stop a project

Example Steps

Make the Project Available to SAS Event Stream Manager

- 1 Install the example:
 - a In SAS Event Stream Processing Studio, on the **Projects** page, click **View example projects**.
 - b In the SAS Event Stream Processing Examples window, find the Filter example and click **Install example**.

The `filter` project appears on the **Projects** page in SAS Event Stream Processing Studio.

- 2 Double-click the `filter` project to open it. Explore the project diagram by clicking each window and reading the contents of the **Description** field in the right pane.
- 3 Make the `filter` project available to SAS Event Stream Manager:
 - a Click  on the left toolbar.
 - b In the right pane, click **New version**.
 - c In the New Version window, click **OK**.

- d In the right pane, click **Promote version 1 to Test**.
- e In the Promote Version window, click **OK**.

Version 1 appears in the **Project Versions** pane under **Test**.

- 4 In SAS Event Stream Manager, go to the **Projects** page and verify that the `filter` project is present. If the project is not visible, click .

Create a Deployment

- 1 On the **Deployments** page, click .
- The New Deployment window appears.
- 2 In the **Name** field, enter `filter`.
 - 3 In the **Description** field, enter `A deployment for filtering events`.
 - 4 In the **Tags** field, enter `filter` and press Enter.
 - 5 In the **Deployment type** drop-down menu, select **Cluster**.
 - 6 Click **OK**.

The `filter` deployment appears on a new page.

Deploy the Project

- 1 Click **Load and Start Project**.
- The Load and Start Project in Cluster window appears.
- 2 In the **Project** field, select `filter`.
 - 3 Click **OK**.

The `filter` project appears in the left pane. Expand `filter` in the left pane to view the Kubernetes pod that is created for the project. If the project was deployed successfully and the pod was created successfully, their statuses are displayed as . The right pane displays further information about the project and the pod.

View Sample Data

- 1 In the left pane, right-click the pod for the `filter` project and select **Open running project**.
- Sample data for the running project appears on a new page named `filter`.
- 2 Select each tab, in turn, to view the results for each window:
 - The **SourceWindow** tab displays all temperature readings.
 - The **Filter** tab displays temperature readings greater than 75°F.

Stop the Project

- 1 On the page for the `filter` deployment, select the `filter` project in the left pane.

- 2 Click .

The Stop Project and Delete ESP Server from Cluster window appears.

- 3 Click **Delete**.

Activity Tracker Example

Overview of the Activity Tracker Example

This example demonstrates how you can use a job template whose type is “Edge” in SAS Event Stream Manager to deploy a project to an edge server. The project used in this example processes Global Positioning System (GPS) data from activity tracker devices that were worn by players in a football match.

The job template in this example loads and starts a project. When the job template is deployed, the job template also prompts the user to select a device ID. A placeholder in the project is then replaced with the device ID that is selected by the user. The output of some of the windows in the project reflects the user’s choice.

Note: To follow this example, you must have an ESP server that runs on an edge server. You cannot deploy the job templates in this example to an ESP server that runs in a Kubernetes cluster. You can view available ESP servers that run on an edge server on the **Unassigned Servers** page in SAS Event Stream Manager.

By following this example, you learn how to do the following tasks. Some of these tasks are performed in SAS Event Stream Processing Studio and some tasks are performed in SAS Event Stream Manager.

- make the project available to SAS Event Stream Manager
- create a deployment
- use a job template to deploy a project
- monitor the deployment
- stop a project

Prepare the Example Files for Use

- 1 If you have not already done so, obtain the `/EndtoEndExamples/activitytracker` directory from <https://github.com/sassoftware/esp-studio-examples> and save it on your computer.

The `/EndtoEndExamples/activitytracker` directory contains the following files:

- `activity_tracker_project.xml` is the project used in the example.
- `activity_tracker_job_template.xml` is the job template used in this example.

- **activitytracker.csv** contains GPS data from activity tracker devices that were worn by players in a football match.
- 2 Save the **activitytracker.csv** file to the server where SAS Event Stream Processing is running.

Note: For the purposes of this example, you cannot upload the **activitytracker.csv** file to a project package.

Example Steps

Make the Project Available to SAS Event Stream Manager

- 1 Import the **activity_tracker_project.xml** file to SAS Event Stream Processing Studio. For more information, see [“Import a Project” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).

The ActivityTracker project appears on the **Projects** page in SAS Event Stream Processing Studio.

- 2 Double-click the ActivityTracker project to open it. Explore the project diagram by clicking each window and reading the contents of the **Description** field in the right pane.
- 3 Adjust the DeviceOutput window to point to the location of the **activitytracker.csv** file on your system:
 - a Click the DeviceOutput window.
 - b In the right pane, click .
 - c Expand **Input Data (Publisher) Connectors** and click .

The Connector Configuration window appears.

 - d In the **Fsname** field, adjust the path to the **activitytracker.csv** file as required.
 - e Click **OK**.

- 4 Click .
- 5 Publish the ActivityTracker project, to make it available in SAS Event Stream Manager. For more information, see [“Publish a Version of a Project” in SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#).
- 6 In SAS Event Stream Manager, go to the **Projects** page and verify that the ActivityTracker project is present. If the project is not visible, click .

Import the Job Template to SAS Event Stream Manager

- 1 On the **Job Templates** page, click  and select **Import job template**.
The import Job Template window appears.
- 2 In the **File** field, navigate to the **activity_tracker_job_template.xml** file.

- 3 Click **OK**.

The ActivityTracker job template appears on the **Job Templates** page.

Note: To explore the job template's contents, double-click the ActivityTracker job template. For more information, see [“Activity Tracker Job Template”](#).

Create a Deployment

- 1 On the **Deployments** page, click .
- The Deployment Properties window appears.
- 2 In the **Name** field, enter **ActivityTracker**.
- 3 In the **Deployment type** drop-down menu, select **Edge**.
- 4 Click **OK**.

The ActivityTracker deployment appears on a new page.

Add an ESP Server

You must add an ESP server to the ActivityTracker deployment.

Note: Ensure that you have an ESP server that you can use for this example. To follow the example, you need an ESP server that runs on an edge server. You cannot deploy the job templates in this example to an ESP server that runs in a Kubernetes cluster. You can view available ESP servers that run on edge servers on the **Unassigned Servers** page. If your ESP servers were not automatically registered with SAS Event Stream Manager and do not appear on the **Unassigned Servers** page, you can register an ESP server manually. For more information, see [“Automatically Registered ESP Servers and Manually Registered ESP Servers”](#).

To add an ESP server to the ActivityTracker deployment:

- 1 Click  and select **Add an unassigned server**.
- The Assign ESP Servers window appears.
- 2 Select the ESP server that you want to assign to the deployment
- 3 Click **Assign**.

The ESP server appears in the table on the ActivityTracker deployment page.

Deploy the Job Template

- 1 On the **Job Templates** page, select the ActivityTracker job template and click .
- The ActivityTracker window appears.
- 2 Complete the fields as follows:
 - **Deployment:** Select **ActivityTracker**.

- **Project:** Select **ActivityTracker**.
- **Latest available version:** This field is automatically set to 1.1.
- **ESP server:** Select your ESP server.
- **Device ID:** Select **accde822**.

This field enables you to filter events from the CSV file. The output from some of the windows in the project reflects your choice. In these windows, only events that match the specified device ID are included in the output.

Note: The **Deployment** field always appears in this window. The other fields appear because they are specified in the job template parameters.

3 Click **OK**.

The **ActivityTracker** page appears. When the `Load Project` task has completed successfully, its status is displayed as .

Monitor the Deployment

- 1 On the **ActivityTracker** deployment page, select your ESP server from the main table.
The **ActivityTracker** project appears in the bottom pane.
- 2 Right-click the `accde822_1` project and select **Open running project**.
The **accde822_1** page appears.
- 3 Select each tab, in turn, to view the results for each window:
 - The **DeviceOutput** tab displays a snapshot of input events in real time.
 - The **LowSatFilter** tab displays events for which the number of satellites that is available to activity tracker devices is fewer than four. Events for all devices are displayed.
 - The **DeviceFilter** tab lists only events that match the selected device ID, `accde822`.
 - The **AggregatedOutput** tab lists only events that match the selected device ID, `accde822`. Details such as the number of satellites available, latitude, and longitude are not displayed. Instead, results of the following calculations are displayed: minimum speed, maximum speed, average speed, and standard deviation.

To see results for the other devices, deploy the project again and select a different device ID.

Stop the Project

When you have finished exploring this example, stop the project:

- 1 On the **ActivityTracker** deployment page, select your ESP server from the main table.
- 2 Click  and select **Stop and unload project**.
- 3 Select the `accde822_1` project and click **OK**.

A new page appears and tasks are displayed for stopping and unloading the project. When each task has completed successfully, the status for each task is displayed as .

An alternative way to stop the project is to use another job template file that is supplied with this example, `stop_job_template.xml`. Deploying this job template stops the selected project and unloads it from the ESP server. For more information, see “[Job Template for Stopping a Project](#)”. To stop the project using another job template:

- 1 On the **Job Templates** page, click  and select **Import job template**.

The Import Job Template window appears.

- 2 From the **File** field, select the `stop_job_template.xml` file.

- 3 Click **OK**.

- 4 Select the `Remove a Running Project v.1` job template and click .

The Job Template ‘Remove a Running Project’ v.1 window is displayed.

- 5 Complete the fields as follows:

- **Deployment:** Select **ActivityTracker**.
- **ESP server:** Select your ESP server.
- **Project to unload:** Select `accde822_1`.

Note: If you selected a different device ID when exploring the example, select the appropriate device ID instead.

- 6 Click **OK**.

The **Remove a Running Project** page appears.

A new page appears and displays the tasks for stopping and unloading the project. When each task has completed successfully, the status for each job is displayed as .

Activity Tracker Job Template

Overview

The following topics describe an example job template, `activity_tracker_job_template.xml`, that loads and starts a SAS Event Stream Processing project.

This job template is used as part of the Activity Tracker example, where it is used together with other resources to create a deployment and deploy a job. For more information, see “[Overview of the Activity Tracker Example](#)”.

The complete job template file is available in the `/EndtoEndExamples/activitytracker` directory in <https://github.com/sassoftware/esp-studio-examples>.

The documentation that follows describes how the job template is constructed. An outline of the job template is created first, and then detail is added to each of the high-level elements in the job template. The complete job template XML is displayed at the end. For more information, see “[The Complete File](#)”.

Job Template Outline

Here is the outline for the activity_tracker_job_template.xml file:

```
<job-template
description-localization-id="jobDescription"
id="ActivityTracker"
localization-id="jobName"

  <localization-strings default-language="en-us">
    <language id="en-us">
      <string id="jobName">ActivityTracker</string>
      <string id="jobDescription">Deploy the
        ActivityTracker project</string>
    </language>
  </localization-strings>

  <enumerations>
  </enumerations>

  <parameters>
  </parameters>

  <initialization>
  </initialiaization>

  <server-filters>
  </server-filters>

  <instructions>
  </instructions>

</job-template>
```

The top-level element, job-template, has the following attributes:

Attribute	Purpose
description-localization-id	A reference to a string value for the job template description used in SAS Event Stream Manager.
id	A unique identifier given to the job template.
localization-id	A reference to a string value for the job template name used in SAS Event Stream Manager.

The localization-strings element in the job template outline stores the string identifier and values for each supported language. Within the localization-strings element, there is a language element whose id attribute is set to en-us (United States English). This language element contains string elements. The jobName and jobDescription strings are referenced by the localization-id and description-localization-id attributes in the job-template element at the beginning of the file. When the job template is imported and then displayed in SAS Event Stream Manager, these values are displayed as metadata about the job template.

The `enumerations` element in the job template outline is populated with elements that specify user input options.

The `parameters` element in the job template outline is populated with elements to gather the user input that is required for executing instructions.

The `initialization` element in the job template outline is populated with elements to alter the SAS Event Stream Processing projects as the projects are deployed.

The `server-filters` element in the job template defines a filter for ESP servers. When the job template is deployed, the filter resolves to a specific ESP server.

The `instructions` element in the job template outline guides SAS Event Stream Manager through the process of creating or modifying a deployment. It is populated with an instruction to start a SAS Event Stream Processing model.

Adding an Enumeration

An *enumeration* is a finite list of options that is presented to the user to restrict input. The enumeration is referenced from the `parameters` element of the job template. In this example, the user is limited to selecting from three device IDs. The enumeration uses string identifiers for the localized string values that are displayed in SAS Event Stream Manager during execution.

The XML `enumerations` element looks like this:

```
<enumerations>
  <enumeration id="edge_ids_list">
    <enumeration-value id="accde822" localization-id="accde822"/>
    <enumeration-value id="15189327" localization-id="15189327"/>
    <enumeration-value id="cc7cf418" localization-id="cc7cf418"/>
  </enumeration>
</enumerations>
```

The `enumeration` element has an attribute called `id`, which is a unique identifier given to this enumeration.

Each `enumeration-value` child element has the following attributes:

Attribute	Purpose
<code>id</code>	A unique identifier given to the enumeration value.
<code>localization-id</code>	A reference to a string value for the enumeration value when it is displayed in SAS Event Stream Manager.

Adding the load-project Instruction

Instructions tell SAS Event Stream Manager what to do with assets in a deployment. In this example, a `load-project` instruction is added to load and then start a SAS Event Stream Processing project.

Like other SAS Event Stream Manager instructions, the `load-project` instruction uses string identifiers for the localized string values that are displayed in SAS Event Stream Manager during execution. It also has XML attributes for parameters to identify required input for the project to load and identify the ESP server, as well as other additional parameters. These parameters can be

specified as literal values in the XML or as references to user-defined input parameters. In this example, user-defined input parameters are specified.

The XML `instructions` element looks like this:

```
<instructions>
  <load-project
    id="load-project"
    localization-id="load-project"
    project-id="{transformed-project.id}"
    project-version="{project.version}"
    running-project-name="{edge_id}_{project.version}"
    server-filter="filter1"
    start="true" />
</instructions>
```

The XML element name, `load-project`, is the instruction name. In this example, the XML line has been split to make it more readable. In a functional job template, it is more likely to be on a single line.

The `load-project` instruction has the following attributes:

Attribute	Purpose
<code>id</code>	A unique identifier given to the instruction. All instructions require this attribute, and its value must be unique in the job template.
<code>localization-id</code>	A reference to a string value for the instruction when it is displayed in SAS Event Stream Manager.
<code>project-id</code>	A unique identifier is created internally for each SAS Event Stream Processing project that has been published in SAS Event Stream Processing Studio. This value is specified by using a variable placeholder indicated by the use of the braces, { and }. The variable in this case is <code>{transformed-project.id}</code>, which is explained in the <code>initialization</code> section.
<code>project-version</code>	Each project that has been published in SAS Event Stream Processing Studio is versioned. In this example, the value of the <code>project-version</code> attribute is specified by using a variable placeholder indicated by the use of the braces, { and }. The variable in this case is <code>{project.version}</code>, which is referenced and explained in the <code>parameters</code> section. For information about alternative ways of specifying the value of the <code>project-version</code> attribute, see “load-project (Edge)” in SAS Event Stream Processing: XML Language Reference for Job Templates.
<code>running-project-name</code>	A name for the running project that will be created in SAS Event Stream Processing. This value must be unique to the running ESP server that will execute the project. This value is specified by using variable placeholders indicated by the use of the braces, { and }.

Attribute	Purpose
	In this case, two variables are specified with a literal underscore (<code>_</code>) to create a project name, <code>{edge_id}_{project.version}</code> . These variables are explained in the <code>parameters</code> section.
<code>server-filter</code>	A reference to a filter for ESP servers. A filter resolves to a set of ESP servers to which the project is deployed. The filter is defined in the <code>server-filters</code> element within the job template.
<code>start</code>	The value of this attribute tells SAS Event Stream Manager to start running the SAS Event Stream Processing project after it has been loaded. In this case, the value is <code>true</code>. If it was <code>false</code>, an additional instruction, <code>start-project</code>, would be required to start the project.

Note: The `load-project` instruction can also take the `project-name` attribute, but in this example the `project-id` attribute is used instead. For more information, see [“load-project \(Edge\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

Adding Parameters for the load-project Instruction

When SAS Event Stream Manager reads a job template in preparation for executing instructions, it first looks to resolve the parameters defined in the `parameters` element.

Given the content of the `load-project` instruction, parameters that identify the following items must be resolved:

- the project itself
- the ESP server to deploy the project onto
- a value for the name of the running project

Three selectors are added to the `parameters` element:

```
<parameters>
  <project-selector id="project"
    localization-id="project"
    required="true" />
  <server-selector id="server"
    localization-id="server"
    required="true" />
  <enum-selector id="edge_id"
    localization-id="edge_id"
    required="true"
    source="edge_ids_list" /></parameters>
```

In this example, each selector element is split onto several lines to make it more readable. In a functional job template, it is more likely to be on a single line.

The `localization-id` attributes are discussed later. (For more information, see [“Adding String References”](#).)

The attributes for the selectors are similar. Each of the attributes is described as follows:

Attribute	Purpose
id	A unique identifier given to the user parameter.
localization-id	A reference to a string value for this parameter when it is displayed in SAS Event Stream Manager.
required	If set to <code>true</code> , the user must provide an input value.
source	This attribute is used only for <code>enum-selector</code> . It provides information about the source of the data. The value must be the name of a defined enumeration.

The `project-selector` element provides the user with a drop-down list that contains the SAS Event Stream Processing projects that have been published in SAS Event Stream Processing Studio. When a value is selected, the `id` and `version` properties of the selected project can be referenced throughout the job template. You can reference these properties by using the `id` attribute, with either a `.id` or `.version` suffix.

The `server-selector` element provides the user with a drop-down list that contains the ESP servers that have been added to SAS Event Stream Manager. When a value is selected, the `id` and `version` properties of the selected project can be referenced throughout the job template. You can reference these properties by using the `id` attribute, with either a `.id` or `.version` suffix.

The `enum-selector` element provides the user with a drop-down list that contains values from the referenced enumeration. That is, this selector provides a list of device IDs.

Adding String References

The `load-project` instruction and the three selectors within the `paramaters` element all specify `localization-id` attribute values that need to be added to the `localization-strings` element of the job template. Each `id` attribute must be associated with a string value. These string values are used by SAS Event Stream Manager when prompting for input or when showing progress or error messages when executing a job created from the job template.

The string references are added under the `language id="en-gb"` element from the outline XML defined earlier. The string references are as follows:

```
<string id="project">Project</string>
<string id="server">ESP server</string>
<string id="load-project">Load Project</string>
<string id="edge_id">Device ID</string>
<string id="accde822">accde822</string>
<string id="15189327">15189327</string>
<string id="cc7cf418">cc7cf418</string>
```

Adding the project-transformation Instruction

An instruction to replace placeholders when a project is deployed must be added to the `initialization` element. Replacing the placeholders is achieved by using the `project-transformation` instruction.

The XML `initialization` element looks like this:

```
<initialization>
```

```

<project-transformations>
  <project-transformation id="transformed-project"
    project-id="{project.id}"
    project-version="{project.version}">
    <placeholder id="edge_id">{edge_id}</placeholder>
  </project-transformation>
</project-transformations>
</initilization>

```

Variables used for {project} are resolved at execution when the user selects a project. The placeholder {edge_id} is replaced with the device ID that is selected by the user.

Adding a Filter

A `server-filters` element must be added to define a filter for ESP servers. In this example, the filter resolves to the ESP server that is selected by the user when the job template is deployed. However, in a more complicated example, the filter might resolve to a larger set of ESP servers, and the project might be deployed to more than one ESP server.

The XML `server-filters` element looks like this:

```

<server-filters>
  <server-filter
    filter-expression="objectId='{server.id}'"
    id="filter1" />
</server-filters>

```

The filter expression contains the {server.id} variable. This variable references the `server-selector` element within the `parameters` element.

In this example, the `server-filters` element is added between the `initialization` and `instructions` elements.

The Complete File

Here is the complete `activity_tracker_job_template.xml` file:

```

<job-template description-localization-id="jobDescription"
  id="ActivityTracker"
  localization-id="jobName">
  <localization-strings default-language="en-us">
    <language id="en-us">
      <string id="jobName">ActivityTracker</string>
      <string id="jobDescription">Deploy the ActivityTracker
        project</string>
      <string id="project">Project</string>
      <string id="server">ESP server</string>
      <string id="load-project">Load Project</string>
      <string id="edge_id">Device ID</string>
      <string id="accde822">accde822</string>
      <string id="15189327">15189327</string>
      <string id="cc7cf418">cc7cf418</string>
    </language>
  </localization-strings>
  <enumerations>
    <enumeration id="edge_ids_list">
      <enumeration-value id="accde822" localization-id="accde822" />
      <enumeration-value id="15189327" localization-id="15189327" />
    </enumeration>
  </enumerations>
</job-template>

```

```

        <enumeration-value id="cc7cf418" localization-id="cc7cf418" />
    </enumeration>
</enumerations>
<parameters>
    <project-selector id="project"
        localization-id="project"
        required="true" />
    <server-selector id="server"
        localization-id="server"
        required="true" />
    <enum-selector id="edge_id"
        localization-id="edge_id"
        required="true"
        source="edge_ids_list" />
</parameters>
<initialization>
    <project-transformations>
        <project-transformation id="transformed-project"
            project-id="{project.id}"
            project-version="{project.version}">
            <placeholder id="edge_id">{edge_id}</placeholder>
        </project-transformation>
    </project-transformations>
</initialization>
<server-filters>
    <server-filter filter-expression="objectId='{server.id}'"
        id="filter1" />
</server-filters>
<instructions>
    <load-project id="load-project"
        localization-id="load-project"
        project-id="{transformed-project.id}"
        project-version="{project.version}"
        running-project-name="{edge_id}_{project.version}"
        server-filter="filter1"
        start="true" />
</instructions></job-template>

```

Job Template for Stopping a Project

This topic describes a simple job template, **stop_job_template.xml**, that stops a running project and unloads it from the ESP server.

This job template is used as part of the Activity Tracker example. For more information, see [“Overview of the Activity Tracker Example”](#).

The job template file is available in the `/EndtoEndExamples/activitytracker` directory in <https://github.com/sassoftware/esp-studio-examples>.

Here is the content for the `stop_job_template.xml` file:

```

<?xml version="1.0"?>
<job-template id="clean-server" localization-id="jobName"
    description-localization-id="jobDescription">
    <localization-strings default-language="en-us">
        <language id="en-us">

```

```

    <string id="jobName">Remove a Running Project</string>
    <string id="jobDescription">Unload a project from a
      specified ESP server</string>
    <string id="target-server">ESP server</string>
    <string id="project-to-stop">Project to unload</string>
    <string id="stop-project">Stop Project</string>
    <string id="unload-project">Unload Project</string>
  </language>
</localization-strings>
<parameters>
  <server-selector id="server"
    localization-id="target-server" />
  <running-project-selector id="project-to-stop"
    localization-id="project-to-stop"
    required="true" />
</server-selector>
</parameters>
<server-filters>
  <server-filter id="filter1"
    filter-expression="objectId='{server.id}'" />
</server-filters>
<instructions>
  <stop-project id="stop-project"
    localization-id="stop-project"
    server-filter="filter1"
    running-project-name="{project-to-stop.name}" />
  <unload-project id="unload-project"
    localization-id="unload-project"
    server-filter="filter1"
    depends-on="stop-project"
    running-project-name="{project-to-stop.name}" />
</instructions>
</job-template>

```

The `localization-strings` element stores the string identifier and values for each supported language. Within the `localization-strings` element, there is a `language` element whose `id` attribute is set to `en-us` (United States English). This language element contains `string` elements. The `jobName` and `jobDescription` strings are referenced by the `localization-id` and `description-localization-id` attributes in the `job-template` element at the beginning of the file. When the job template is imported and then displayed in SAS Event Stream Manager, these values are displayed as metadata about the job template.

The `parameters` element contains selectors that gather the user input required to execute instructions.

The `server-filters` element defines a filter for ESP servers. The `{server.id}` variable references the `server-selector` element within the `parameters` element. In this example, the filter resolves to the ESP server selected by the user when the job template is deployed. However, in a more complicated example the filter might resolve to a larger set of ESP servers.

The `instructions` element contains instructions for stopping the running project specified by the user and unloading it from the ESP server. For more information about the `stop-project` instruction, see [“stop-project \(Edge\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#). For more information about the `unload-project` instruction, see [“unload-project \(Edge\)” in SAS Event Stream Processing: XML Language Reference for Job Templates](#).

GitHub Actions Workflow Example

Overview of the GitHub Actions Workflow Example

This example demonstrates how you can create a GitHub Actions workflow that automatically promotes a SAS Event Stream Processing project to a test environment when a tag for a new major version is pushed to your repository. For more information, see [GitHub Actions: Events that trigger workflows](#).

The workflow authenticates with SAS Logon, calls the [Event Stream Processing Studio API](#) to promote the project version, and then uses the [Event Stream Manager API](#) to refresh and deploy the project in the target test deployment.

Use Case

Teams that develop SAS Event Stream Processing projects need a reliable, repeatable way to move a project from development into a shared test environment after each major release. Manual deployments are error-prone and slow. A tag-triggered GitHub Actions workflow automates every deployment step: authentication, version promotion, project refresh, and project start.

This example implements the development-to-test promotion stage. When a tag for a new major version is pushed to the repository, the `CI_New_Version` workflow starts automatically and executes the following steps:

- 1 Checks out the repository.
- 2 Extracts the major version number (for example, 4 from `v4.1`).
- 3 Logs the detected version and the project name.
- 4 Obtains the bearer token from SAS Logon.
- 5 Promotes the project version in SAS Event Stream Processing Studio to the Test promotion level.
- 6 Refreshes the table on the **Projects** page in SAS Event Stream Manager.
- 7 Starts the project in the test deployment.

No manual intervention is required after the tag is pushed.

Workflow File

In this example, you create a workflow file called `new_version_workflow.yml` that must be committed to the `.github/workflows/` directory of your project repository.

The file contains the tag trigger, the SAS Logon authentication step, the SAS Event Stream Processing Studio promotion call, and the SAS Event Stream Manager refresh and deployment calls.

Set Up the GitHub Actions Workflow

Before You Start

Check Prerequisites

Confirm that the following prerequisites are met:

- You have Admin or Write permissions for the GitHub repository that contains your ESP project.
- A GitHub Actions runner is registered on the repository. The workflow uses `runs-on: self-hosted`, but you must update this value to match the available runners for your repository, such as `runs-on: ubuntu-latest`.
- You have a SAS Logon client ID whose credentials are Base64-encoded in the format `clientId:clientSecret` for the `LOGON_AUTHZ` secret. For more information, see [“Authentication: Overview” in SAS Viya Platform: Authentication](#).
- A SAS user account exists with permission to promote projects in SAS Event Stream Processing Studio and to start deployments in SAS Event Stream Manager.
- The project that you want to promote already exists in SAS Event Stream Processing Studio. Its name must match the repository name exactly, because the workflow uses `{{ github.event.repository.name }}` as the project identifier.

Configure Repository Secrets and Variables

Before committing the workflow file, configure the following repository secrets and variables:

- 1 In the project repository, select **Settings** ⇒ **Secrets and variables** ⇒ **Actions**.
- 2 Select **New repository secret** and create the following secrets:

Name	Value
LOGON_AUTHZ	Base64-encoded <code>clientId:clientSecret</code> credentials for your SAS Logon OAuth client. For more information about how to register a client token for use by the workflow file, see https://developer.sas.com/docs/rest-apis/getting-started/authentication .
CLIENT_USERNAME	The SAS user name that is used to authenticate with SAS Logon.
CLIENT_PASSWORD	The password for <code>CLIENT_USERNAME</code> .

CAUTION

Never paste credentials directly into a workflow file Reference credentials by using the `{{ secrets.<NAME> }}` syntax.

- 3 Click the **Variables** tab and create the following variables:

Name	Value
DEV_TEST_HOST	The base URL for the SAS Viya platform in your organization (for example, <code>https://my-viya-server.example.com</code>). The SAS Viya platform must be running and reachable at this URL.
TEST_DEPLOYMENT_NAME	The name of the SAS Event Stream Manager deployment to deploy the project into (for example, <code>test-deployment</code>). The deployment must already exist in SAS Event Stream Manager.

Create the Workflow File

Create the file `.github/workflows/new_version_workflow.yml` in your project repository with the following content.

Note: You can make the file more secure by using an OpenID Connect (OIDC) workflow. Replace the Resource Owner Password Credentials (ROPC) grant with OIDC so that the workflow does not need to store a long-lived password in GitHub secrets.

```
# Workflow that automatically promotes a project to test when a new major tag is
detected in a commit to the repository.
name: CI_New_Version
run-name: On new major version (${{ github.ref_name }})
on:
  push:
    tags:
      - 'v[0-9]+.1'    # Matches v1.1, v4.1, v10.1, and so on
  workflow_dispatch:  # Allows manual re-runs from the Actions tab
jobs:
  build:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v4
      - name: Get version
        id: getVersion
        uses: actions/github-script@v6
        with:
          script: |
```

```

        core.setOutput('majorVersion', "${{ github.ref_name }}".substring(1, "${{ github.ref_name }}".indexOf('.'));
    - name: Workflow information
      run: echo 'Detected a new version "$
    {{ steps.getVersion.outputs.majorVersion }}" of project "$
    {{ github.event.repository.name }}" with tag "${{ github.ref_name }}"'
    # Authenticate with SAS Logon to obtain a bearer token
    - name: SAS Logon token request
      id: tokenRequest
      uses: fjogeleit/http-request-action@v1
      with:
        url: '${{ vars.DEV_TEST_HOST }}/SASLogon/oauth/token'
        method: 'POST'
        customHeaders: '{"Content-Type":"application/x-www-form-urlencoded","Authorization":"Basic ${{ secrets.LOGON_AUTHZ }}"}'
        data: '{"grant_type":"password","username":"$
    {{ secrets.CLIENT_USERNAME }}","password":"${{ secrets.CLIENT_PASSWORD }}"}'
        ignoreSsl: "true"
        maskResponse: "true"
    - name: SAS Logon response
      run: echo "Token Retrieved"
    # Promote the project to test level via SAS Event Stream Processing Studio
    - name: Promote project to Test
      id: promoteProjectRequest
      uses: fjogeleit/http-request-action@v1
      with:
        url: '${{ vars.DEV_TEST_HOST }}/SASEventStreamProcessingStudio/v4/projects/${{ github.event.repository.name }}/promotions'
        method: 'POST'
        customHeaders: '{"Content-Type":"application/json","Accept":"application/json, application/vnd.sas.esp.project-version-history+json, application/vnd.sas.error+json","Authorization":"Bearer $
    {{ fromJson(steps.tokenRequest.outputs.response).access_token }}"}'
        data: '{"notes":"Automatic promotion to Test","tags":["QA","CI"],"version":$
    {{ steps.getVersion.outputs.majorVersion }}}'
        ignoreSsl: "true"
        timeout: 10000
    - name: Promote project response
      run: echo "Promoted project with response $
    {{ fromJson(steps.promoteProjectRequest.outputs.status) }}"
    # List projects in SAS Event Stream Manager to trigger refresh of projects
    - name: Refresh projects
      id: refreshProjectsRequest
      uses: fjogeleit/http-request-action@v1
      with:
        url: '${{ vars.DEV_TEST_HOST }}/SASEventStreamManager/v4/projects'
        method: 'GET'
        customHeaders: '{"ContentType":"application/json","Accept":"application/json,application/vnd.sas.collection+json","Authorization":"Bearer $
    {{ fromJson(steps.tokenRequest.outputs.response).access_token }}"}'
        ignoreSsl: "true"
    - name: Deploy project response
      run: echo "Refreshed projects"
    # Wait for projects to refresh asynchronously
    - name: Sleep for 30 seconds
      run: sleep 30

```

```

# Start project in test SAS Event Stream Manager test deployment
- name: Deploy project
  id: deployProjectRequest
  uses: fjogeleit/http-request-action@v1
  with:
    url: '${{ vars.DEV_TEST_HOST }}/SASEventStreamManager/v4/deployments/${{ vars.TEST_DEPLOYMENT_NAME }}/clusterServers'
    method: 'POST'
    customHeaders: '{"Content-Type":"application/json","Accept":"application/json, application/vnd.sas.esp.job-representation+json,application/vnd.sas.error+json","Authorization":"Bearer ${{ fromJson(steps.tokenRequest.outputs.response).access_token }}"}'
    data: '{"deploymentName":"${{ vars.TEST_DEPLOYMENT_NAME }}","projectName":"${{ github.event.repository.name }}"}'
    ignoreSsl: "true"
    timeout: 60000
  - name: Deploy project response
    run: echo "Deployed project with response ${{ fromJson(steps.deployProjectRequest.outputs.status) }}"

```

The `on.push.tags` value `v[0-9]+.1` limits the trigger to tags that identify major version releases intended for the development-to-test promotion stage:

Tag Example	Milestone
v4.1	Development to test (this workflow)
v4.2	QA environment
v4.3	Staging or production environment

The `workflow_dispatch` trigger enables you to run the workflow manually in the GitHub user interface, which is useful for testing the configuration before you push a real tag. To do this, click **Actions** ⇒ **CI_New_Version** ⇒ **Run workflow**.

Workflow Steps Reference

Get Version

The `Get version` step uses the `actions/github-script` action to extract the major version number from the tag. For example, from the tag `v4.1`, the action produces the output `4`. This output is stored as `majorVersion` and referenced in later steps as `${{ steps.getVersion.outputs.majorVersion }}`.

The major version number is passed in the promotion request body (`"version": 4`) to tell SAS Event Stream Processing Studio which version of the project to promote.

SASLogon Token Request

The `SASLogon Token Request` step uses the `fjogeleit/http-request-action` action to POST to the SAS Logon OAuth `/token` endpoint by using the Resource Owner Password Credentials (ROPC) grant. A successful response contains an `access_token` that is used as a bearer token in all subsequent API calls. For more information, see [fjogeleit/http-request-action](#).

Setting `maskResponse: "true"` prevents the token from appearing in the workflow run log.

Promote Project to Test

The `Promote project to Test` step issues a POST request to the Event Stream Processing Studio API's [POST /projects/{projectName}/promotions](#) endpoint. The request body passes three values:

- `version` - the integer for a major version that is extracted in the `Get Version` step
- `notes` - a human-readable description of the promotion (for example, "Automatic promotion to Test")
- `tags` - the labels `QA` and `CI`, which are applied to the version entry in SAS Event Stream Processing Studio

Refresh Projects

The `Refresh projects` step issues a GET request to the Event Stream Manager API's [GET /projects](#) endpoint. This call causes SAS Event Stream Manager to synchronize its internal project list with the latest promotions from SAS Event Stream Processing Studio, making the newly promoted version available for deployment.

Deploy Project

The `Deploy project` step issues a POST request to the Event Stream Manager API's [POST /deployments/{deploymentName}/clusterServers](#) endpoint for the target deployment. SAS Event Stream Manager starts the project within the specified deployment. The `timeout` value is 60000 milliseconds (60 seconds) to allow sufficient time for the project to start.

Test the Workflow and View the Results

Verify the Results

To trigger the workflow, open the project in SAS Event Stream Processing Studio and use the **Project Versions** pane to create a new version.

After the new version is created, verify the results in GitHub Actions and on the SAS Viya platform:

- 1 In the project repository, select **Actions**.
- 2 Confirm that the `CI_New_Version` workflow appears with the run name `On new major version (v1.1)`.
- 3 Open the run and expand each step to confirm that it succeeded:
 - **Get version:** Confirm that no errors are reported.
 - **Workflow information:** Confirm that the log shows the correct project name and tag.
 - **SAS Logon response:** Confirm that the log shows `Token Retrieved`.
 - **Promote project response:** Confirm that the log shows an HTTP 2xx status code.
 - **Deploy project response** (Refresh projects): Confirm that the log shows `Refreshed projects`.

- **Deploy project response** (Deploy project): Confirm that the log shows an HTTP 2xx status code.
- 4 In SAS Event Stream Processing Studio, confirm that a new project version was created:
 - a Open the project and click  on the left toolbar. Confirm that a new version is listed in the **Project Versions** pane.
 - b Click the **History** tab in the right pane and confirm that the project version has the tags QA and CI, and the comment Automatic promotion to Test.
 - 5 In SAS Event Stream Manager, open the test deployment and confirm that the project appears and is in a running state.

Note: If required, you can refresh the **Project Versions** pane in SAS Event Stream Processing Studio and the deployment page in SAS Event Stream Manager to see the latest status.

The running project in the test deployment indicates that the promotion was successful and that the new version of the project is available for testing by the team. When testing is complete, manually stop the project in SAS Event Stream Manager.

Troubleshooting

Symptom	Likely Cause	Action
Workflow does not start after pushing a tag.	The tag name does not match the pattern, or the workflow file is not on the default branch.	Verify the tag format (v followed by digits, followed by .1) and confirm that the workflow file is committed to the default branch. Check the runner status at Settings ⇒ Actions ⇒ Runners .
The SASLogon Token Request step fails with an HTTP 401 response.	LOGON_AUTHZ is incorrect, or the user credentials are wrong.	Verify that LOGON_AUTHZ is a correctly Base64-encoded clientId:clientSecret string. Verify that CLIENT_USERNAME and CLIENT_PASSWORD are correct.
The Promote project to Test step fails with an HTTP 404 response.	The project does not exist in SAS Event Stream Processing Studio, or its name does not match the repository name.	Confirm that a project whose name exactly matches the repository name exists in SAS Event Stream Processing Studio.
The Promote project to Test step fails with an HTTP 409 response.	The version was already promoted.	The process continues to the next step.

Symptom	Likely Cause	Action
The Deploy project step fails with an HTTP 404 response.	TEST_DEPLOYMENT_NAME does not match an existing SAS Event Stream Manager deployment.	Verify the deployment name in SAS Event Stream Manager and update the TEST_DEPLOYMENT_NAME variable.
The Deploy project step times out.	The project takes longer than 60 seconds to start.	Increase the timeout value in the Deploy project step.
The Refresh project step fails with an HTTP 403 response.	The user account has permission to use the Event Stream Processing Studio API but not the Event Stream Manager.	Ensure that the user account has appropriate permissions. For more information, see “Access SAS Event Stream Manager” .