SAS® Viya® Platform: Using SAS® Viya® Monitoring for Kubernetes

1.2.47 - 1.2.54

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.

Introduction . . . 2

Performing Basic Tasks in OpenSearch Dashboards . . . 2
- Explore Log Messages with the Discover Application . . . 2
- Choose Your Dashboards . . . 4
- Filter Log Messages . . . 5
- Search Log Messages . . . 6
- Apply a Time Constraint . . . 6
- Filter Information in a Graph . . . 7
- Select Fields to Display . . . 8
- Save a New Search . . . 8
- Export Logs to a CSV File . . . 9

Performing Basic Tasks in Grafana . . . 11
- About the Welcome Dashboard . . . 11
- Change to Another Dashboard . . . 11
- Select a Time Interval . . . 11
- Filtering Dashboard Metrics . . . 12
- Learn about Other Grafana Dashboards . . . 13
- Learn More about How to Use Grafana Dashboards . . . 14

Using Grafana Alerting . . . 14
- About Grafana Alerting . . . 14

Monitoring SAS Compute Server, SAS Launcher Service, and SAS Workload Orchestrator . . . 15
- Examining Log Messages with OpenSearch Dashboards . . . 15

Monitoring Jobs Started by SAS Launcher Service ... 17

Monitoring SAS CAS Server ... **38**
About the SAS CAS Server Dashboard ... 38
The Dashboard User Interface ... 38
Ways to Use the SAS CAS Overview Dashboard ... 40

Exporting Logs with getlogs.py ... **40**
About the getlogs.py Script ... 40
General Syntax ... 41
Usage Examples ... 44

Accessing Observability Tools Via SAS Environment Manager ... **49**
Access the Log-Monitoring Application from SAS Environment Manager ... 49

Troubleshooting ... **49**
The Kubernetes-Proxy Dashboard Does Not Display Any Data ... 49
SAS Launched Jobs Dashboards Do Not Work ... 50

Introduction

The SAS Viya Monitoring for Kubernetes project is a monitoring solution for the SAS Viya platform that is built from highly regarded open-source observability tools. This document provides information about how to use two of these tools, OpenSearch Dashboards and Grafana, to monitor the SAS Viya platform.

The following information is also in this document:

- How to export logs by using the provided `getlogs.py` Python script.
- How to access OpenSearch Dashboards and Grafana from SAS Environment Manager.

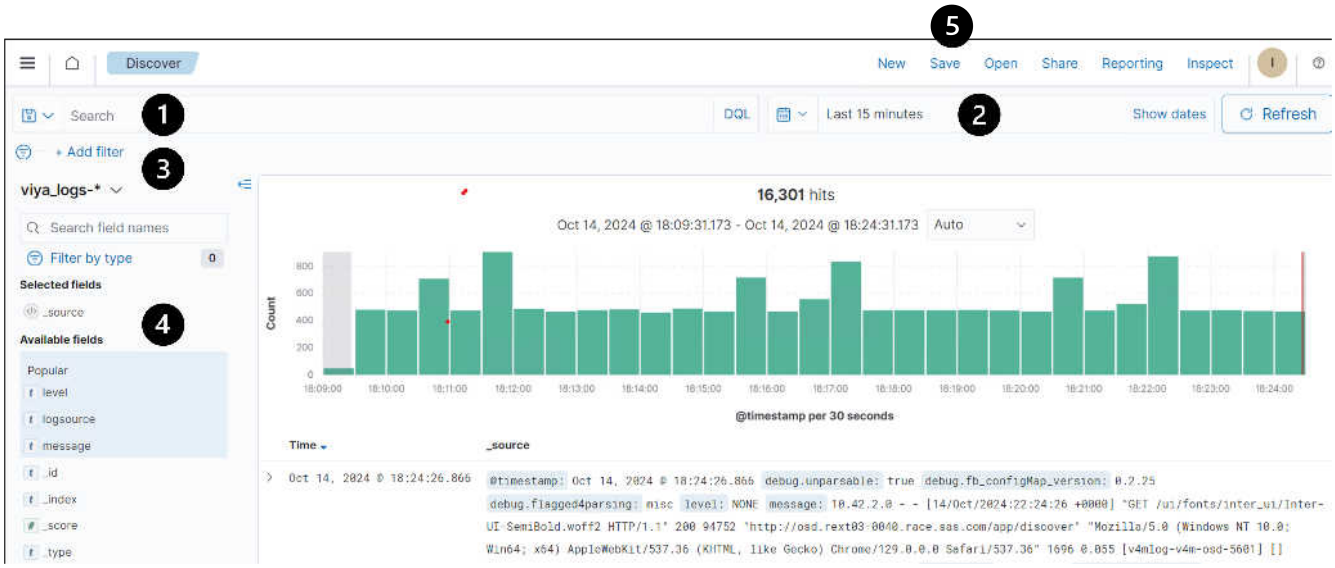
Performing Basic Tasks in OpenSearch Dashboards

Explore Log Messages with the Discover Application

The Discover application is designed for interactive exploration. By default, the application displays information from all log sources over a specified time period.

TIP The Discover application provides filtering tools and a large area for displaying log messages. These make the Discover application useful for troubleshooting specific issues.

To display the Discover application, click the ☰ and then click **Discover**.



1 The **Search** field.

To use Dashboard Query Language (DQL) in the **Search** field at the top, see Search log messages.

2 The time field and selection button.

To change the time period, see “Apply a Time Constraint”.

3 The **Add filter** button.

To define filters that are based on a specified source, message level, or specific text, see “Filter Log Messages”.

4 The **Available fields** selection view.

Select fields to display in the table.

5 The **Save** button.

To save the modifications so that you can return to that arrangement in the future, see “Save a New Search”.

IMPORTANT Before you save anything (dashboards, searches, views, and so on) in OpenSearch Dashboards, be sure to select and turn on the **Save as new** switch. This switch is located in the Save window. If you do not turn on this switch, you will overwrite the prior item even if you specify a new file name.

If, for example, you change a predefined dashboard or visualization and want to save your changes, you must turn on the **Save as new** switch to save your version. If you do not turn on the switch, the predefined dashboard or visualization is overwritten and, if you specified a new name, the file is renamed.

When predefined content is renamed or changed, other predefined content can fail. Because predefined content might rely on the renamed or changed content, a failure occurs when the expected content can no longer be located.

Choose Your Dashboards

About Your Dashboards Options

The OpenSearch Dashboards application provides dashboards that can display a combination of previously defined visualizations and tables to provide an overview of the logging activity. The dashboards are useful in routine monitoring of log messages and identifying the source of problems. See "OpenSearch Dashboards Provided by SAS" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes* for descriptions of the predefined dashboards.

You can create custom dashboards also. See "Create a Custom Dashboard" on page 4.

Select a Predefined Dashboard

Complete the following steps:

1 Click ☰ and then click **Dashboards**. The Dashboards application opens.

2 In the Dashboards application, click the name of a predefined dashboard. The dashboard opens.

Create a Custom Dashboard

Complete the following steps:

1 Click ☰ and then click **Dashboards**. The Dashboards application opens.

2 In the Dashboards application, click **Create dashboard**.

3 To add an existing visualization to the dashboard, click **Add an existing**.

Alternatively, you can click **Create new**. For information about creating visualizations, see the OpenSearch Dashboards documentation.

4 In the Add panels window, select one or more predefined visualizations.

5 Click outside the Add panels window to dismiss it. The selected visualizations appear on the dashboard.

Filter Log Messages

Visualizations, dashboards, and the Discover application in OpenSearch Dashboards display many log messages from many different sources. Because it can be difficult to find the log messages that you need to troubleshoot a problem, OpenSearch Dashboards enables you to define filters to display information only about selected messages (such as those from a specified source, a specified level, or specific text).

To define a filter:

1 On an OpenSearch Dashboards page that supports filtering (such as **Discover**, **Dashboards**, or **Visualize**), click **Add filter**.

2 In the Edit Filter window, click the **Field** field and select the field on which you want to filter.

> **TIP**
>
> - If you select a field that ends with the .keyword suffix, the **Value** field provides a list of possible values and autocompletion.
> - If you select a field that does not end with the .keyword suffix, the **Value** field enables you to enter partial strings to use in the filter.

3 Click the **Operator** field and select an operator, depending on how you want to filter the messages.

4 Click the **Value** field, and enter the value that you want to filter for or exclude, depending on the operator you selected.

> **TIP** Although the **Value** field supports autocompletion, response time can be slow, and results are sometimes incomplete if there are many possible values.

5 (Optional) If you want a custom label for the filter, click the **Create custom label** switch and enter a label in the **Custom label** field.

6 Click **Save** to apply the filter.

The chart and message table change to display messages that match the filter that you defined. If no messages occurred during the selected time period, you can change the time period displayed. See "Apply a Time Constraint" on page 6.

After you apply a filter, a breadcrumb appears at the top of the page that contains the filter criteria. Click on the breadcrumb text to edit the filter, include or exclude the filtered results, disable or re-enable the filter, or delete the filter.

Search Log Messages

Searching enables you to locate log messages that contain specific values, words, or phrases.

To search, enter a search query in the **Search** field at the top of a page that supports searching (such as the Discover application or a dashboard) to find log messages that match the query. By default, OpenSearch Dashboards searches all the fields in the index for the specified search terms and displays only the messages that contain the search terms.

To limit the search to a specific field, use this format for the search query:

```
field_name:search_term
```

> **TIP** If the search term contains spaces, you must enclose the search term in quotation marks (" "). For example, to search the message field for the words "fatal error", use the search query `message:"fatal error"`.

> **IMPORTANT** Before you save anything (dashboards, searches, views, and so on) in OpenSearch Dashboards, be sure to select and turn on the **Save as new** switch. This switch is located in the Save window. If you do not turn on this switch, you will overwrite the prior item even if you specify a new file name.
>
> If, for example, you change a predefined dashboard or visualization and want to save your changes, you must turn on the **Save as new** switch to save your version. If you do not turn on the switch, the predefined dashboard or visualization is overwritten and, if you specified a new name, the file is renamed.
>
> When predefined content is renamed or changed, other predefined content can fail. Because predefined content might rely on the renamed or changed content, a failure occurs when the expected content can no longer be located.

For more information about searching and using the Dashboards Query Language (DQL), see the OpenSearch documentation.

Apply a Time Constraint

By default, the pages in OpenSearch Dashboards display log messages from the past 15 minutes. You can change the time period according to your preference.

Set a Time Period Quickly

Complete the following steps to set a time period quickly:

1 Click  that is located next to the **Search** field.

2 Specify a time period or select one of the predefined periods in the **Commonly used** area.

3 Click **Apply**.

Set a Custom Time Period

Complete the following steps to set a custom time period:

1 Click in the time field to the right of the  button. The time field contains the start and end points of the time period.

2 You can select either an absolute or a relative value for the start and end points of the time period. An absolute value is a specific date and time. A relative value is a date and time relative to another value (such as **30 minutes ago** or **5 days from now**).

> **TIP**
>
> - Click the left side of the time field to specify the start point of the time period.
> - Click the right side of the time field to specify the end point of the time period.

Specify a value for the start or end point of a time period:

- To specify an absolute value, select the **Absolute** tab and select the date and time for your value.
- To specify a relative value, select the **Relative** tab. Enter the value into the first field (for example, **30**). Select the time increment from the second field (for example, **Minutes ago**).
- To set a value to now, select the **Now** tab and click the confirmation button.

3 Click **Update** to apply the time constraint.

If the page displays a bar chart of messages by time, you can select a bar to view information only about messages that occurred during the selected time. Click and drag to select multiple bars.

Filter Information in a Graph

OpenSearch Dashboards visualizations and dashboards can include graphs of log message frequency. You can interact with the graphs in a dashboard or visualization to filter the information in the graph and locate the information that you need.

Here are some of the interactions that are supported:

- Click and drag to select an area of a graph to display information about only the selected time period.
- Click on a bar in a bar chart to choose whether to filter based on the time period represented by the bar, the category variable (such as **logsource**) represented by the bar, or both.

Select Fields to Display

In the Discover application, you can select the fields that you want to include in the table of log messages. Each field is a column in the table. The **Time** and **_source** fields are displayed by default. The **_source** field is a concatenation of the most useful fields in the log message documents.

Note: The **Filter** pane is not available in the **Dashboards** application.

To add fields, follow these steps:

1. In the **Filter** pane to the left of the Discover view, hover the mouse pointer over the field name that you want to add. Click **Add field as column** +.

   The first field that you add replaces the **_source** field.

2. Repeat step 1 for every field that you want to add as a column in your table. Each field is added at the end of the table.

> **TIP** Use the **Search field names** search box to easily find fields.

> **TIP** If you previously added a field to the table, you can find that field at the top of the **Available fields** list under the **Popular** section.

Save a New Search

> **IMPORTANT** Before you save anything (dashboards, searches, views, and so on) in OpenSearch Dashboards, be sure to select and turn on the **Save as new** switch. This switch is located in the Save window. If you do not turn on this switch, you will overwrite the prior item even if you specify a new file name.
>
> If, for example, you change a predefined dashboard or visualization and want to save your changes, you must turn on the **Save as new** switch to save your version. If you do not turn on the switch, the predefined dashboard or visualization is overwritten and, if you specified a new name, the file is renamed.
>
> When predefined content is renamed or changed, other predefined content can fail. Because predefined content might rely on the renamed or changed content, a failure occurs when the expected content can no longer be located.

Click **Save** in the top right toolbar to save your current search and modified dashboard. Enter your title for your search and be sure to turn on the **Save as new** switch.

You can access your saved search at a later time by clicking **Open** in the top right toolbar.

The SAS Viya Monitoring for Kubernetes project comes with several saved searches. Click **Open** in the top right toolbar to view a list of saved searches.

For example, open the **Log Messages** saved search. It includes the following modifications:

- The `viya_logs-*` index pattern is used. The `viya_logs-*` index pattern contains log messages from most Kubernetes services in the cluster. (The other available index pattern is `viya_ops-*`. It contains log messages from the SAS Viya Monitoring for Kubernetes project.)
- The following fields are displayed in the table:
  - Time (this field is always included)
  - level
  - logsource
  - message
- The table is sorted in descending order of time, which is the default.

> **TIP** Use the **Log Messages** saved search in the Discover application as a starting point for your investigations and troubleshooting. Modify the time filter and add query filters to help identify problems.

Export Logs to a CSV File

OpenSearch Dashboards uses the search and filter criteria in a saved search to generate a CSV file.

Complete the following steps:

1. Click ☰ and then click **Discover**. The Discover application opens.
2. You can either create a new search or use a previously defined search.
   - **New search:** For information about creating a search, see the following topics:
     - “Search Log Messages” on page 6
     - “Filter Log Messages” on page 5
     - “Apply a Time Constraint” on page 6
   - **Previously defined search:** If you select an OpenSearch predefined search (such as the **Log Messages** search) or another search that must not be overwritten, you must save a copy of the search:
     1. Click **Save** in the menu bar.
     2. When the Save search window appears, click **Save as new search**.

     > **IMPORTANT** Be sure to turn on this switch. If you do not, the previously defined search is overwritten. All dashboards and visualizations that use the overwritten

search will be broken. The breakage will affect all users of the affected dashboards and visualizations.

3 In the **Title** field, enter a unique name for the copy of the search.

IMPORTANT Changing the name of the saved search does not create a new search. The change only renames the existing search. Therefore, make sure that you enable the **Save as new search** option.

4 Click **Save**.

With the copy of the search, you can adjust the search terms, add filters, and add a time constraint to further refine the search.

3 The CSV file does not include the **Time** column from the search results. To add the date and time to the CSV file, select **@timestamp** from the **Available fields** section in the left pane.

The **@timestamp** column is added to the far right in the table.

4 To move the **@timestamp** column to the left, hover the pointer over the **@timestamp** column heading to reveal buttons. Click **<<** to move the column to the left.

Repeat this step until the column is located to the right of the **Time** column. You cannot move the **@timestamp** column any farther left. Because the **Time** column is not generated to the CSV file, the **@timestamp** column is generated as the first column in the CSV file.

5 (Optional) Rearrange the other table columns to meet your needs. For each column that you want to move, hover the pointer over the column heading to reveal buttons. Click **<<** to move the column to the left or **>>** to move the column to the right.

6 To save the search settings and the table columns, click **Save** in the menu bar.

7 When the Save search window appears, click **Save**.

8 (Optional) If the search includes a relative time constraint, such as **last 15 minutes**, click **Refresh** immediately before the next step. This ensures that the results in the Discover application are generated shortly before the CSV file is generated.

9 In the menu bar, click **Reporting** ⇨ **Generate CSV**.

The CSV file is downloaded by your web browser. Open the CSV file in your preferred application such as Notepad++ or Microsoft Excel.

Note: When viewing the CSV file, remember the following considerations:

- If you made changes to the search after you saved the search, those changes are not in the CSV file.
- The **Time** column is not included in CSV file.
- Typically, the number of rows in the file is limited to 10,000.
- If the search includes a relative time constraint, such as **last 15 minutes**, the CSV file includes the log messages that were available within 15 minutes of the time of the CSV-file generation. That time range differs from when the search results were generated in the table in the Discover

application. To eliminate most of the time discrepancy, remember to click **Refresh** to refresh the Discover application immediately before generating the CSV file.

Performing Basic Tasks in Grafana

About the Welcome Dashboard

When you first start Grafana, the **Welcome to SAS Viya Monitoring for Kubernetes** dashboard is displayed. This dashboard provides the following resources:

- links to key dashboards for both Kubernetes cluster administration and SAS Viya administration
- a list of favorite and recently used dashboards
- links to support resources, including SAS and Grafana documentation, SAS Technical Support, and the project's GitHub site
- links for reporting and viewing project issues

Change to Another Dashboard

To change to a different dashboard, select a dashboard from either of these locations:

- the **Welcome to SAS Viya Monitoring for Kubernetes** dashboard
- the Application menu in the upper left corner. Select **Dashboards** for the list of the default set of dashboards

> **TIP** Select a label from the **Tags** column to return a subset of the dashboards.

For each dashboard, you can do the following tasks:

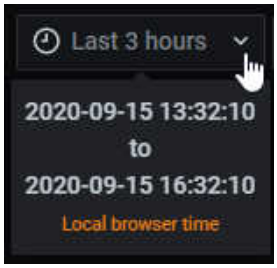
- "Select a Time Interval" on page 11
- "Filtering Dashboard Metrics" on page 12

Select a Time Interval

A time selector is located in the toolbar at the top of the dashboard.

Use this control to select the time period for which the metric data is displayed. The default value varies for dashboards. It ranges from 5 minutes to 3 hours.

Hover your pointer over the field to display the beginning and end of the selected time period.



Click on the control to specify a time period. You can specify time in the following ways:

- Select a specific beginning and ending time.
- Select a relative time range (such as **Last 6 hours** or **Last 7 days**).

Filtering Dashboard Metrics

How Filters Behave

A set of filters is available at the top of each dashboard that enable you to view only the data that you need to see. (Note that a Grafana dashboard does not require filters. However, the dashboards that are included in this project do have filters.)

When you apply a filter, this might affect the selections that are available for the remaining filters (from left to right). For example, if you use the **Job type** filter to view only sas-batch-job types, the **User** and **Job** filters operate only on the returned SAS batch jobs.

Available Filters

Here are some of the filters provided on the dashboards:

Application
selects an application from a list of all applications.

CAS Server (present only on the SAS CAS Overview dashboard)
selects a CAS server from a list of all CAS servers.

Datasource
selects a source of monitoring data from a list of all Prometheus data sources. This typically defaults to the appropriate value. You might need to modify this if you have defined multiple Prometheus data sources. An example is if you are using the same Grafana instance to visualize metric data from multiple clusters.

Instance
selects an instance from a list of all instances of the selected application. The application is selected via the **Application** filter.

Job
selects one or more jobs from a list of all active jobs.

Job Type
selects a type of job, such as sas-batch-jobs, compute-server, or sas-launcher-job.

Namespace
selects a namespace in the Kubernetes cluster. If you have multiple SAS namespaces on your cluster, you can view metrics for each namespace separately. Select **all** to view metrics for all the namespaces in the cluster.

Node (present only on the **Node Activity** dashboard)
selects a node in the Kubernetes cluster to view information about jobs running only on that node or **all** to view jobs on all nodes.

PG Cluster (present only on the PostreSQL dashboard)
selects a PG (PostgreSQL) cluster from a list of all PG clusters.

Queue
selects a SAS Workload Orchestrator queue to view information about jobs running on nodes that are associated with the selected queue.

Top (present only on the User Activity dashboard)
specifies whether you want to view metrics from the top 5, 10, 20, or 50 users.

User
selects one or more specific users.

Learn about Other Grafana Dashboards

The SAS Viya Monitoring for Kubernetes solution provides several Grafana dashboards to visualize the metric data that is collected by Prometheus. You can query, explore, or get alerts based on the metrics of your SAS Viya deployments and the Kubernetes cluster where you have deployed the SAS Viya Platform. These dashboards have been created by the SAS team, derived from Grafana public community dashboards, or they are from a third-party source. See "Grafana Dashboards Location and File Names" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes* in the Reference section for information about where the dashboards originate.

For descriptions of the available Grafana dashboards, see these topics:

- "Dashboards for SAS Viya Deployments" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*
- "Dashboards for Kubernetes Resources" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*
- "Dashboards for SAS Viya Monitoring for Kubernetes Solutions" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*

To add additional dashboards, see "Add More Grafana Dashboards" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

Learn More about How to Use Grafana Dashboards

To learn more about how to use Grafana dashboards, see Use Grafana Dashboards in the Grafana documentation.

Using Grafana Alerting

About Grafana Alerting

Grafana has its own built-in alerting service that is configurable through Grafana and enables you to do the following:

- create and manage Grafana-managed alerts and data source-managed alerts
- monitor alerts using dashboards and visualizations
- route alert notifications to a contact point or through notification policies

For more detailed information and instructions on using the alerting functionality, see Grafana Alerting.

Decide how you want to organize your alerts and notifications. See Alerting Best Practices for guidance on alerting management from Grafana documentation.

To access alerting, select **Alerting** from Grafana's sidebar.

> **TIP** Automate the provisioning of alerts, contact points and notification policies so that your alerting system will not have to be manually re-created every time you deploy your monitoring solution. See "Customize Alerts Managed by Grafana" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

Monitoring SAS Compute Server, SAS Launcher Service, and SAS Workload Orchestrator

Examining Log Messages with OpenSearch Dashboards

Considerations

> **IMPORTANT** It can be unclear whether the time shown by OpenSearch Dashboards is GMT or your local time zone. Be sure to determine the time zone used.

The filters in this section are designed for OpenSearch Dashboards, but they can be adapted for use in other tools. To create filters in OpenSearch Dashboards, see "Filter Log Messages" on page 5.

Before You Start

You must know one or both of the following items to filter log messages:

- The approximate time at which the issue occurred. See these topics:
  - "Determine When the Problem Occurred" in *SAS Viya Platform: Troubleshooting*
  - "Apply a Time Constraint" on page 6

  > **TIP** If nothing in the log messages appears connected to the issue, review events from five minutes before and five minutes after that time. Look for anything unusual, such as database errors or connection timeouts.

- The affected pod name. See these topics:
  - For ways to identify the Compute server pod name, see "Identify the Name of the Compute Server Pod That Failed to Start" in *SAS Viya Platform: Troubleshooting*.
  - For ways to identify the Launcher service pod name, see Examine the Launcher Service Logs and "Decipher the Compute Server Pod Name" in *SAS Viya Platform: Troubleshooting*.

Filter Event Log Messages

Display Kubernetes event log messages that are not Normal-level events.

```
logsource is KUBE_EVENT and level is not 'Normal'
```

Then, you can search the output for specific words or errors. See "Search Log Messages" on page 6.

Filter SAS Compute Server Log Messages

- Display log messages for all Compute server pods.

  ```
  kube.pod is 'sas-compute'
  ```

- Display log messages for a specified Compute server pod.

  ```
  kube.pod is 'pod_name'
  ```

  where *pod_name* is the Compute server pod name.

Filter SAS Workload Orchestrator Log Messages

- Display log messages that are for the SAS Workload Orchestrator pod and are not INFO-level events.

  ```
  kube.pod is 'sas-workload-orchestrator' AND level is not 'INFO'
  ```

- Display log messages that are for the SAS Workload Orchestrator pod and contain the Compute server pod name.

  ```
  kube.pod is 'sas-workload-orchestrator' AND message is 'pod_name'
  ```

  where *pod_name* is the Compute server pod name.

- Display log messages that are for the SAS Workload Orchestrator pod and contain the SAS Workload Orchestrator job ID that is associated with the Compute server. To learn about the job ID, see "Determine How the Compute Server Pod Was Created" in *SAS Viya Platform: Troubleshooting*.

  ```
  kube.pod is 'sas-workload-orchestrator' AND message is 'job_id'
  ```

  where *job_id* is the Compute server SAS Workload Orchestrator job ID.

Filter SAS Launcher Service Log Messages

- Display log messages for all Launcher service pods.

  ```
  kube.pod is 'sas-launcher'
  ```

- Display log messages for the Launcher service pod.

  ```
  kube.pod is 'pod_name'
  ```

  where *pod_name* is the Launcher service pod name.

Monitoring Jobs Started by SAS Launcher Service

Overview

The SAS Viya Monitoring for Kubernetes solution includes two Grafana dashboards that are designed to monitor jobs that are started by the SAS Launcher service:

- SAS Launched Jobs - Node Activity
- SAS Launched Jobs - User Activity

These dashboards enable you to monitor the following types of jobs in the SAS Viya platform:

- SAS Batch server (sas-batch-job)
- SAS Launcher service (sas-launcher-job)
- SAS Compute server (compute-server)

  Compute server jobs run programs for SAS Viya platform functions (such as jobs and flows) and SAS applications (such as SAS Studio).

When monitoring these jobs, be aware that your environment might use queues to manage workload. The queues are managed by SAS Workload Orchestrator. If queues are configured, their names are available as filters. See "Filter by a Specified Queue" on page 25.

Note: To understand how SAS Launcher service works with SAS Compute server, see "How Compute Server Works" in *SAS Viya Platform: Programming Run-Time Servers*.

See Also

- SAS Viya Platform: Programming Run-Time Servers
- SAS Viya Platform: Workload Management
- Troubleshooting Strategies

Considerations

IMPORTANT These dashboards require SAS Workload Orchestrator. It is enabled by default when SAS Viya platform is deployed. Make sure that SAS Workload Orchestrator is enabled in your deployment. See "Dashboards That Require a License or Enablement" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes* for details.

When you use these dashboards, remember the following considerations:

- Because of the nature of metric data, Grafana reports what the data tells it. Grafana cannot state specifications about your environment. For example, Grafana does not know or report how much memory or how many CPUs each node has in your environment. You must know that information. You can identify this information in either of the following ways:
  - Use the steps in "Nodes in Your Kubernetes Cluster" in *SAS Viya Platform: Troubleshooting*.
  - Use these Grafana dashboards that are also provided by SAS Viya Monitoring for Kubernetes:
    - Node Exporter - Nodes
    - Perf/Node Utilization Detail
- The objects in the dashboards use data from a set of recording rules that are provided with SAS Viya Monitoring for Kubernetes. Each of these rules aggregates a set of Kubernetes metrics to return a single metric value. The metrics recording rules are run every ten seconds.

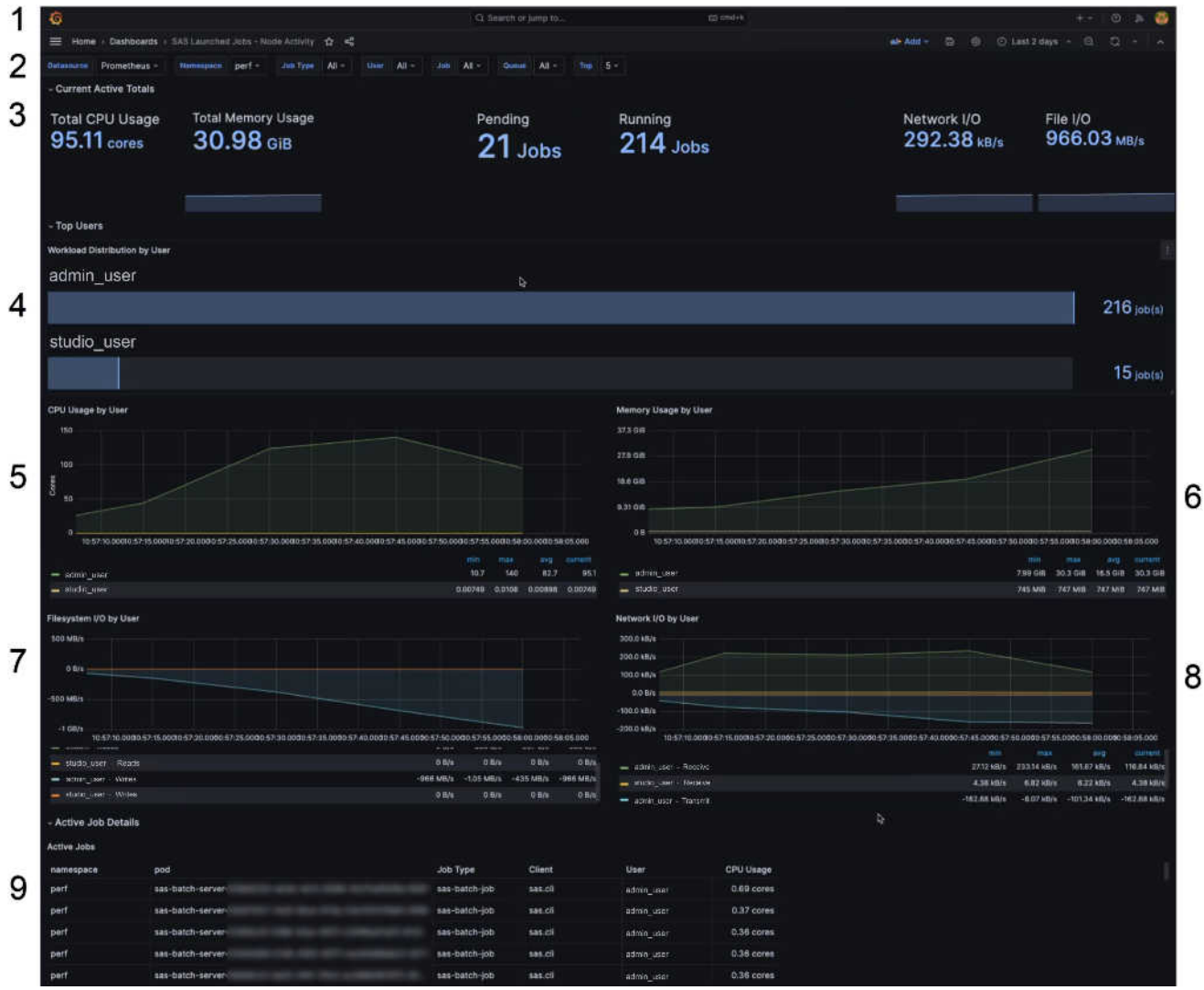
  Note: Because of timing issues with the Kubernetes queries, there might be occasional inconsistencies between the values displayed by various items in the dashboards.

  See "Recording Rules Provided for SAS Workload Manager" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes* for more information about the recording rules that are used by these dashboards.

The Dashboard Interface

The following figure displays a SAS Launched Jobs dashboard and describes the elements of its user interface.

Figure 1 *The SAS Launched Jobs Dashboard User Interface*



1 A toolbar of general options across the top of the dashboard. These options include the **Refresh** button and the time range and time picker. See "Filter by Time Range" on page 20.

2 A toolbar of filters below the general toolbar. By using these filters, you can focus on specific metric data to facilitate troubleshooting and performance tuning. See "Filtering Dashboard Data" on page 20.

3 The **Current Active Totals** row. It reports data for the currently selected state of the dashboard. See "View Current Active Totals" on page 28.

4 The **Workload Distribution** chart varies depending on the currently viewed dashboard:

- (Node Activity dashboard) The **Workload Distribution by Node** chart. This bar chart provides a quick way to evaluate a node's share of the workload.
- (User Activity dashboard) The **Workload Distribution by User** chart. This bar chart provides a quick way to evaluate a user's share of the workload.

See "View Workload Distribution" on page 28.

5 The **CPU Usage** chart varies depending on the currently viewed dashboard:

- (Node Activity dashboard) The **CPU Usage by Node** chart. This chart shows how the SAS Launcher service jobs use the CPUs on the selected nodes or all nodes across a selected range of time. See "Identify a Node with Saturated or Under-Used Cores (CPUs)" on page 30.
- (User Activity dashboard) The **CPU Usage by User** chart. This chart shows CPU processing across a range of time by user. See "Identify a User's CPU Processing Needs" on page 35.

6 The **Memory Usage** chart varies depending on the currently viewed dashboard:

- (Node Activity dashboard) The **Memory Usage by Node** chart. This chart shows how the SAS Launcher service jobs use memory on the selected nodes or all nodes across a selected range of time. See "Identify a Node with a Memory Leak" on page 30.
- (User Activity dashboard) The **Memory Usage by User** chart. This chart shows how each user uses memory across a selected range of time. See "View Memory Use by User" on page 36.

7 The **Filesystem I/O** chart varies depending on the currently viewed dashboard:

- (Node Activity dashboard) The **Filesystem I/O by Node** chart. This chart shows each node's Read and Write activity in the file system for the selected SAS Launcher service pods across a selected range of time. See "Monitor File System I/O Activity by Node" on page 31.
- (User Activity dashboard) The **Filesystem I/O by User** chart. This chart shows the read and write activity in the file system for the selected SAS Launcher service pods across a selected range of time by user. See "Monitor File System I/O Activity by User" on page 37.

8 The **Network I/O** chart varies depending on the currently viewed dashboard:

- (Node Activity dashboard) The **Network I/O by Node** chart. This chart shows each node's network transmission activity for the selected SAS Launcher service pods across a selected range of time. See "Monitor Network I/O Activity by Node" on page 32.
- (User Activity dashboard) The **Network I/O by User** chart. This chart shows network transmission activity for the selected SAS Launcher service pods across a selected range of time by user. See "Monitor Network I/O Activity by User" on page 37.

9 The **Active Jobs** table is located on both the Node Activity and User Activity dashboard. This table displays all of the jobs in the selected time range. See "View Active Job Details" on page 33.

Filtering Dashboard Data

> **TIP** To narrow any filter selection list, enter a partial name. Sometimes entering the last few characters of a name can quickly show your selection.

Filter by Time Range

When you troubleshoot, narrowing the time range of your data is important. The current dashboard time range and time picker is located on the general toolbar. It is labeled with a small clock icon.

1 Click the down arrow to the right of the time selection.

2 To set an absolute time, complete these steps:

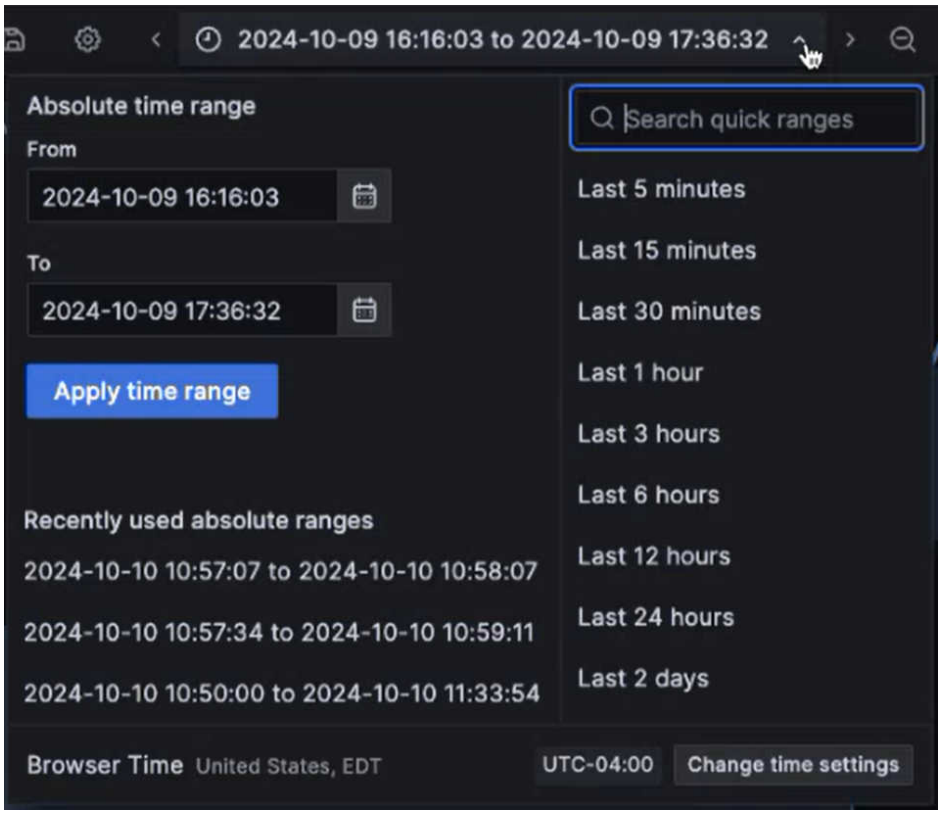
a In the **Absolute time range** area, click the calendar button next to the **From** field and select the start of the time range.

b Click the calendar button next to the **To** field and select the end of the time range.

c Click **Apply time range**.

3 To set a relative time range, click a selection under the **Search quick ranges** field. If none of the selections meets your needs, you can search for another in the field or specify your own relative time ranges (such as *now-2h* or *now-1h*).

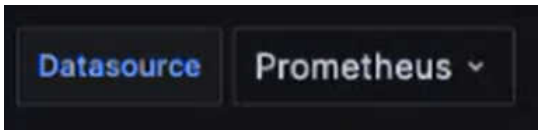
The window collapses into the toolbar and your selection is displayed in the toolbar for easy reference. The data in the dashboard is filtered by your selection.



Filter by Data Source

The **Datasource** filter is located on the filter toolbar.

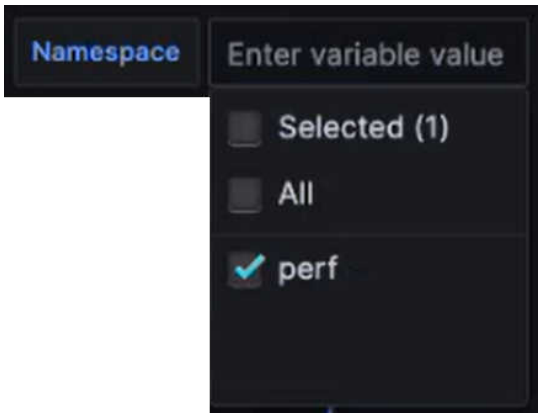
SAS Viya Monitoring for Kubernetes imports data from Prometheus into Grafana. The **Datasource** filter defaults to the **Prometheus** selection.



Filter Data by a Specified Namespace

The **Namespace** filter is located on the filter toolbar.

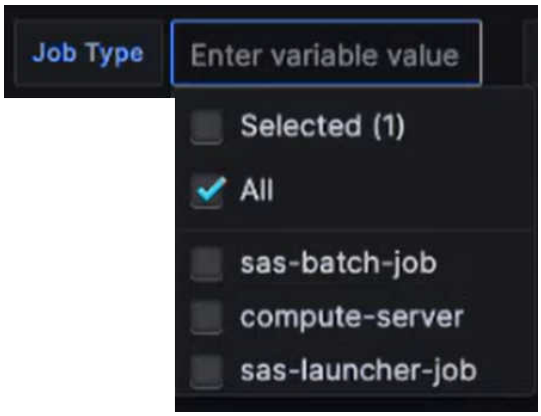
If your Kubernetes environment has more than one namespace, you can filter the data by that value. In the **Namespace** filter, you can scroll down the list to select a specific namespace. Or, you can enter part of the namespace name in the **Enter variable value** field. The filter defaults to **All**. In this example, the environment has one namespace called `perf`.



Filter by SAS Launcher Service Jobs, SAS Batch Server Jobs, or SAS Compute Server Jobs

The **Job Type** filter is located on the filter toolbar.

If you are investigating specific types of jobs, use the **Job Type** filter to narrow the data shown by the job type.



The following names are the default names for the job types. However, job types can have other names that can appear in the filter. For details, see "Decipher the Compute Server Pod Name" in *SAS Viya Platform: Troubleshooting*.

- For SAS Launcher service jobs, select **sas-launcher-job**.
- For SAS Batch server jobs, select **sas-batch-job**.
- For SAS Compute server jobs, select **compute-server**.

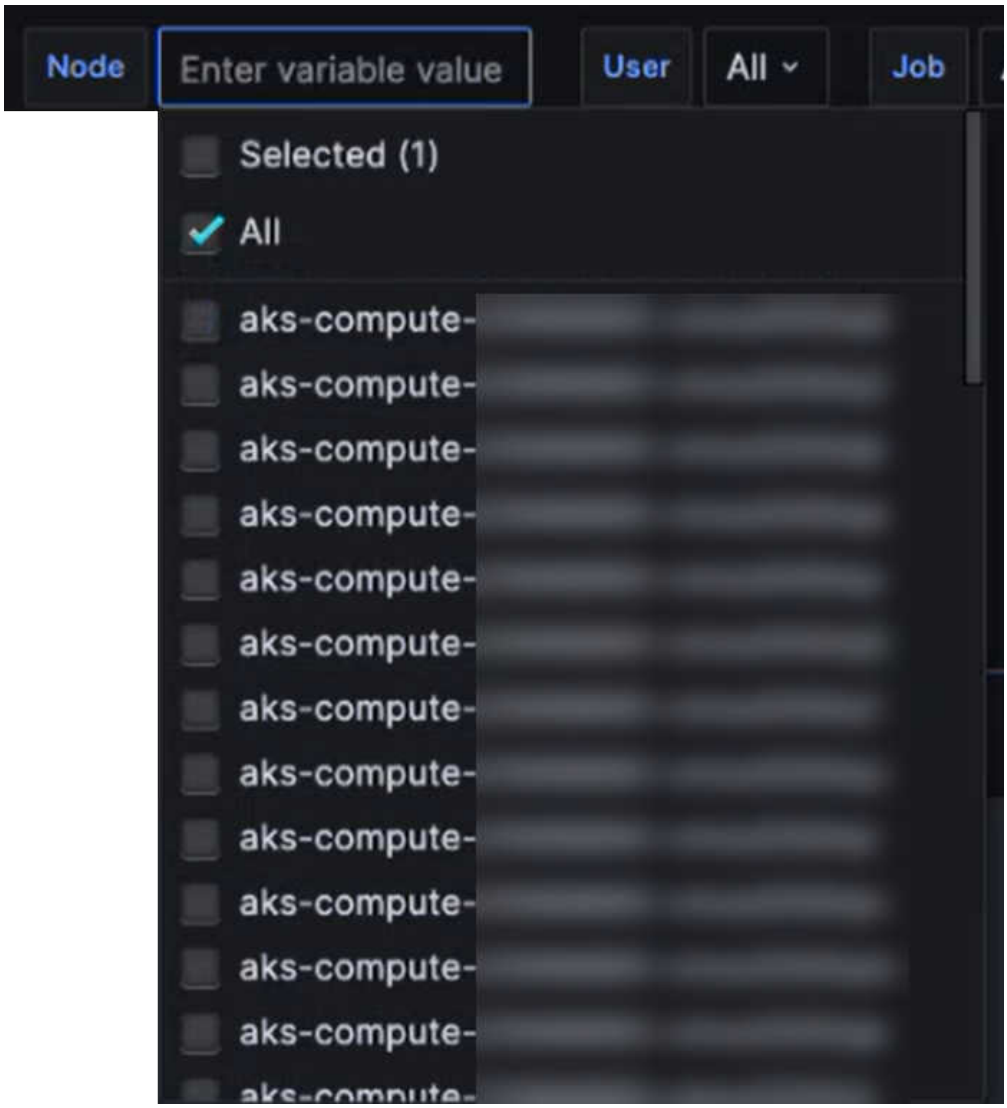
The filter defaults to **All**.

Filter Data by a Specified Node

Note: The **Node** filter is available in the filter toolbar on the SAS Launched Jobs - Node Activity dashboard only.

A Kubernetes cluster can have many nodes. Use the **Node** filter to narrow the data by one or more nodes. In the **Node** filter, you can scroll down the list to select a specific node. Or, you can enter part of the node name in the **Enter variable value** field. The filter defaults to **All**, which is all of the nodes on which the jobs ran.

> **TIP** To narrow the selection list, enter a partial node name (such as characters at its end).

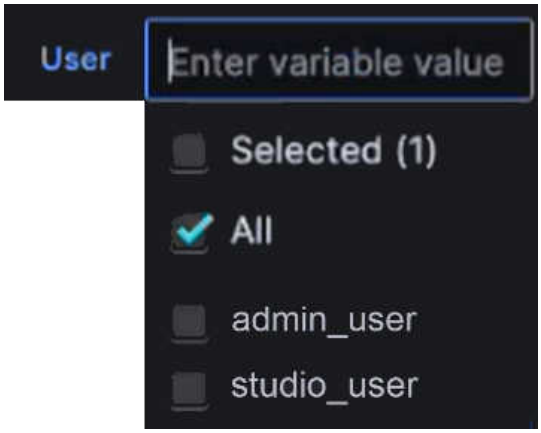


Filter by a Specified User

The **User** filter is located on the filter toolbar.

You can filter the data by one or more users in the environment. This filter can help you diagnose problems reported by a specific user. In this example, two users are shown. In the **User** filter, you can scroll down the list to select a specific user. Or, you can enter part of the user name in the **Enter variable value** field. The filter defaults to **All**.

> **TIP** To narrow the selection list, enter a partial user name.

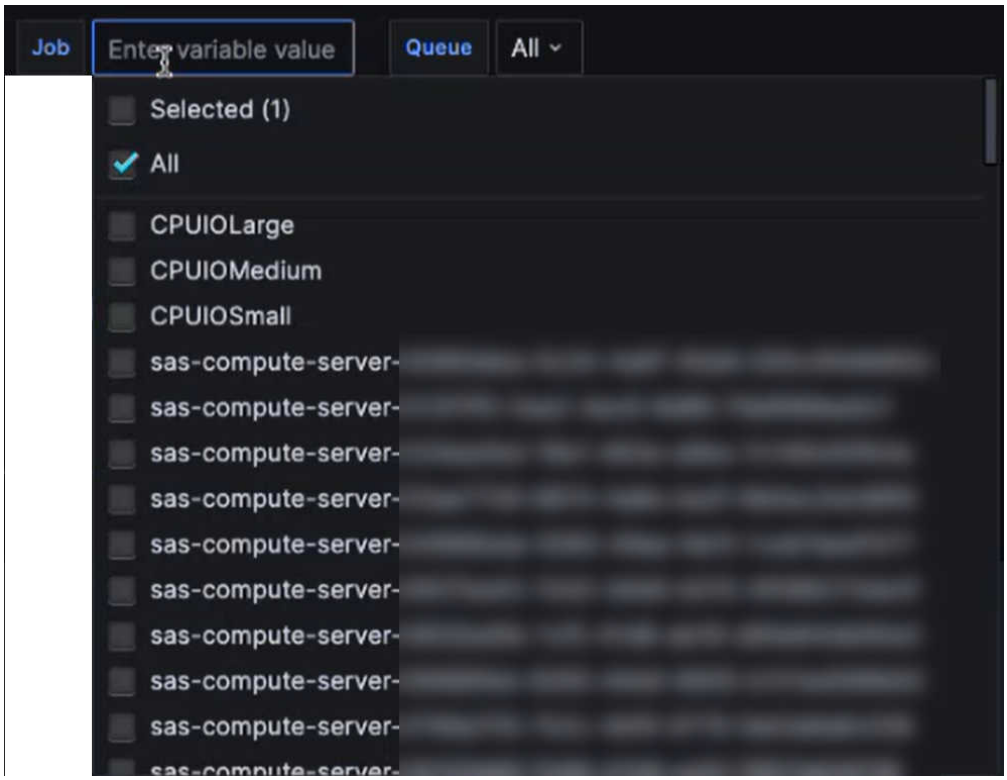


Filter Data by a Specified Job

The **Job** filter is located on the filter toolbar.

When troubleshooting, you might want to investigate data associated with a specific job. In the **Job** filter, you can scroll down the list to select a specific job. Or, you can enter part of the job name in the **Enter variable value** field. The filter defaults to **All**.

> **TIP** To narrow the selection list, enter a partial job name (such as the hash at the end).

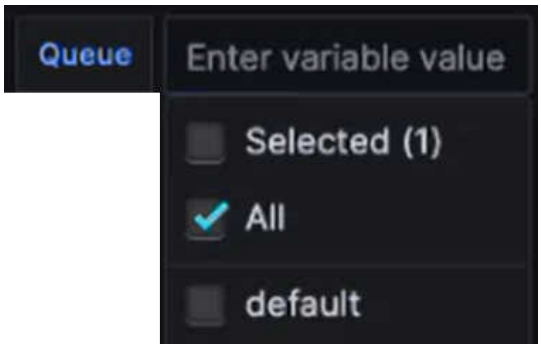


Filter by a Specified Queue

The **Queue** filter is located on the filter toolbar.

If SAS Workload Orchestrator manages multiple queues in your environment, you can filter the data by queue. In the **Queue** filter, you can scroll down the list to select a specific queue. Or, you can enter part of the queue name in the **Enter variable value** field. The filter defaults to **All**. This example shows one queue named `default`.

> **TIP** The queue names displayed are the same queue names seen in on the **Workload Orchestrator** page in SAS Environment Manager. See "Workload Orchestrator Page" in *SAS Environment Manager: User's Guide*.



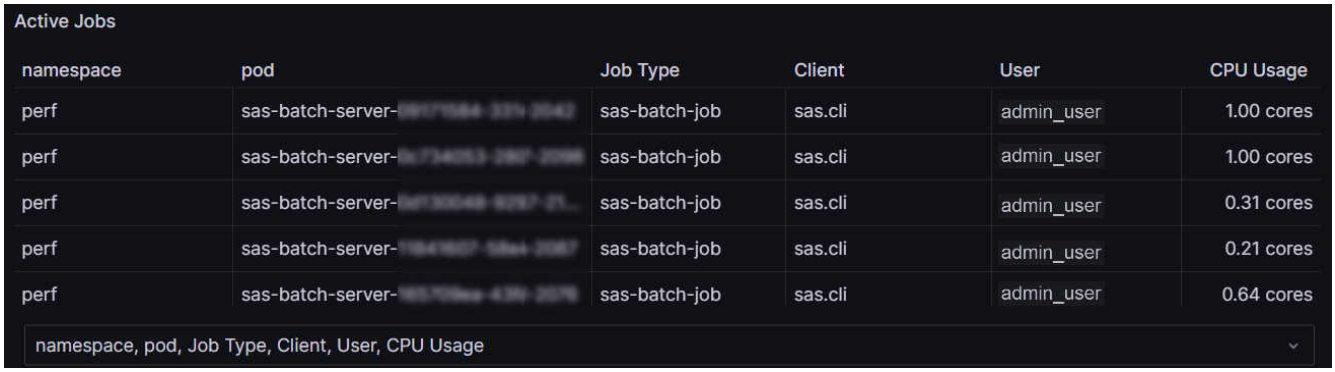
Filtering by queue enables you to evaluate the metric data for queues that manage different workloads. Here are examples:

- A high-priority queue for large jobs that require intensive processing.
- A low-priority queue that processes small jobs.

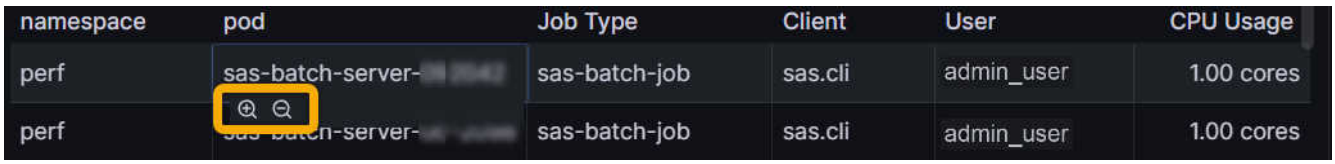
Filter by Pod

This filter is not on the filter toolbar by default. You access this filter via the **Active Jobs** table.

1 Scroll down to the **Active Jobs** table.

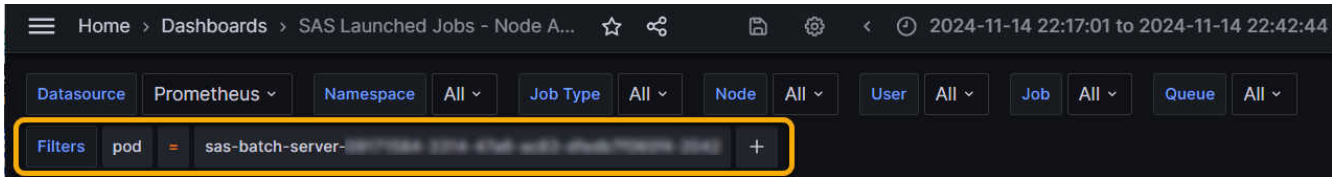


2 Hover over the **pod** cell by which you want to filter. The **Filter for value** and **Filter out value** buttons appear.



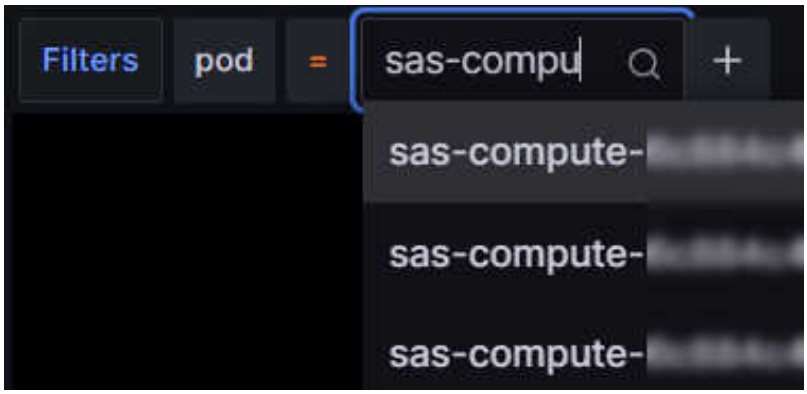
> **TIP** Each cell in this table except **CPU Usage** provides these filter buttons.

3 Select  (**Filter for value**). The **Filters** option appears in the filter toolbar. The option is set to filter on the selected pod.



The dashboard charts refresh to show the metric data for the specified pod.

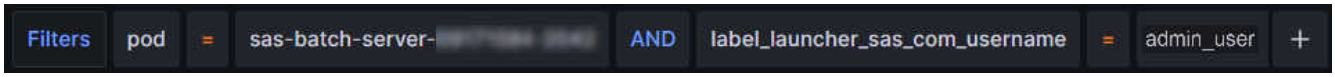
4 You can change the pod selection in the filter. Click the pod name. The field expands into a list. You can enter part of a pod name to filter the list.



The dashboard charts refresh to show the metric data for the selected pod. However, if the selected pod is not in the data, the charts do not have any metric data to show.

You can further refine the data in the following ways:

- You can also add additional filters to this option. Hover over another cell in the **Active Jobs** table to either filter for another value or filter out another value. In the following example, a filter has been added for *admin_user*.

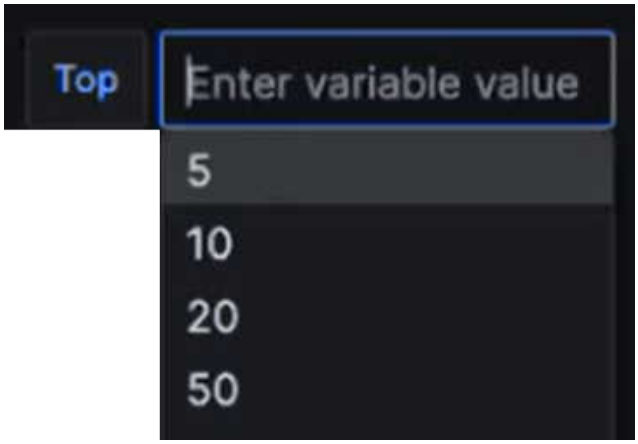


- Adjust the time range.
- Drill down into the charts.

Filter by a Specified Number of Users with the Most Activity

Note: The **Top** filter is available the filters toolbar on the SAS Launched Jobs - User Activity dashboard only.

You can filter the data by a specific number of users who have the most activity. For example, to view the data associated with the five users with the most activity, select **5** in the **Top** filter.



View At-a-Glance Data

View Current Active Totals

The **Current Active Totals** area contains a set of charts that provides an overview of performance that varies depending on the dashboard:

- **Node Activity dashboard:** Shows an overview of node performance.
- **User Activity dashboard:** Shows an overview of SAS Launcher pod performance.

The information shown is current in that moment in time within the selected time range. If you change the time range, such as narrow your time selection, the charts update to show the current values as calculated for the new time range.

Each chart in this area displays two types of information:

- the most recently collected value for the metric
- a general visual representation of metric trend

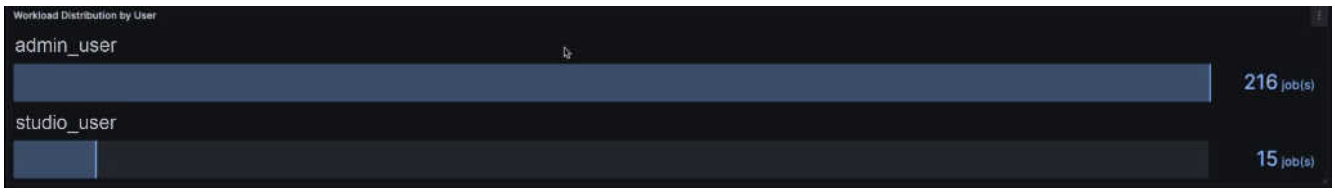
These charts update depending on how you apply filters to the dashboard.

View Workload Distribution

This bar chart provides a quick way to evaluate how the workload is distributed across nodes or users.

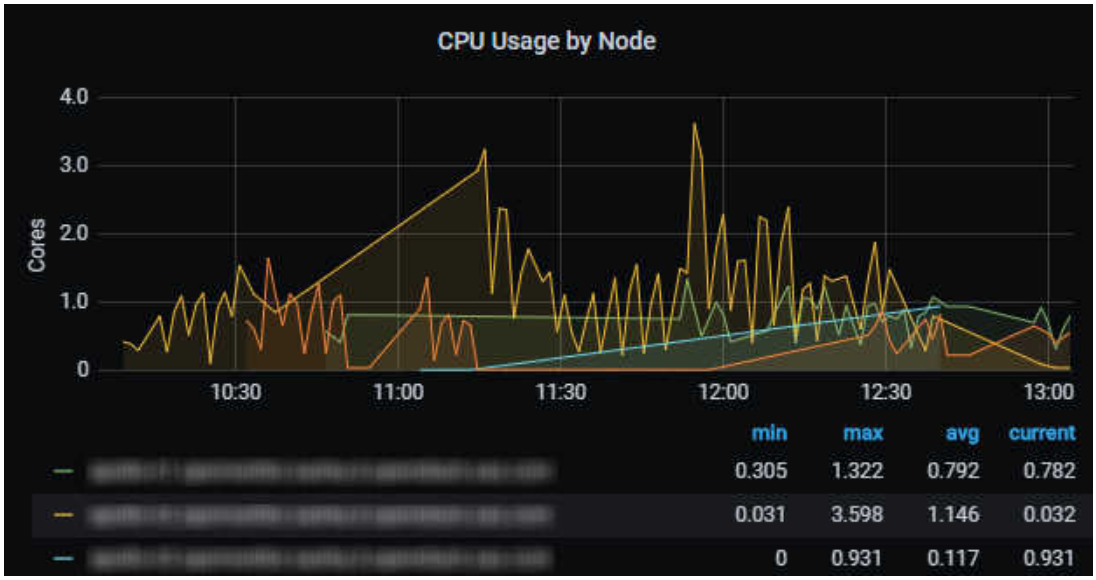
Figure 2 *The Workload Distribution by Node Chart*

Figure 3 *The Workload Distribution by User Chart*



General Tips about How to Use the Charts

Both SAS Launched Jobs dashboards display multiple line charts. Here is an example:



These line charts have the following functionality:

- The line for each node is color coded.
- A table of all the nodes is displayed below the chart. The table scrolls to shows all of the nodes. For each node, the table displays its color, name, minimum, maximum, average, and current values of use.
- Hover over any given point in the chart and a pop-up displays information that applies to the data point.
- Drill down into the view of the chart by narrowing the time range. Doing so can make it easier to see details. When you drill down in one chart, the other charts update their view to the new selection as well.

  To select a new time range, left-click in the chart at one end of the range and drag the mouse pointer to other end. Release the left mouse button and the chart updates.

  Note: When you view CPU use in a small enough range of time, the Y-axis shows millicores (1/1000 of a single CPU or core). These are slices of a CPU and indicate that the CPU is under used.

- In the table, click on a column heading to sort the data by that column.
- (Node Activity dashboard) In the table, click on a node to view only its data in the chart.

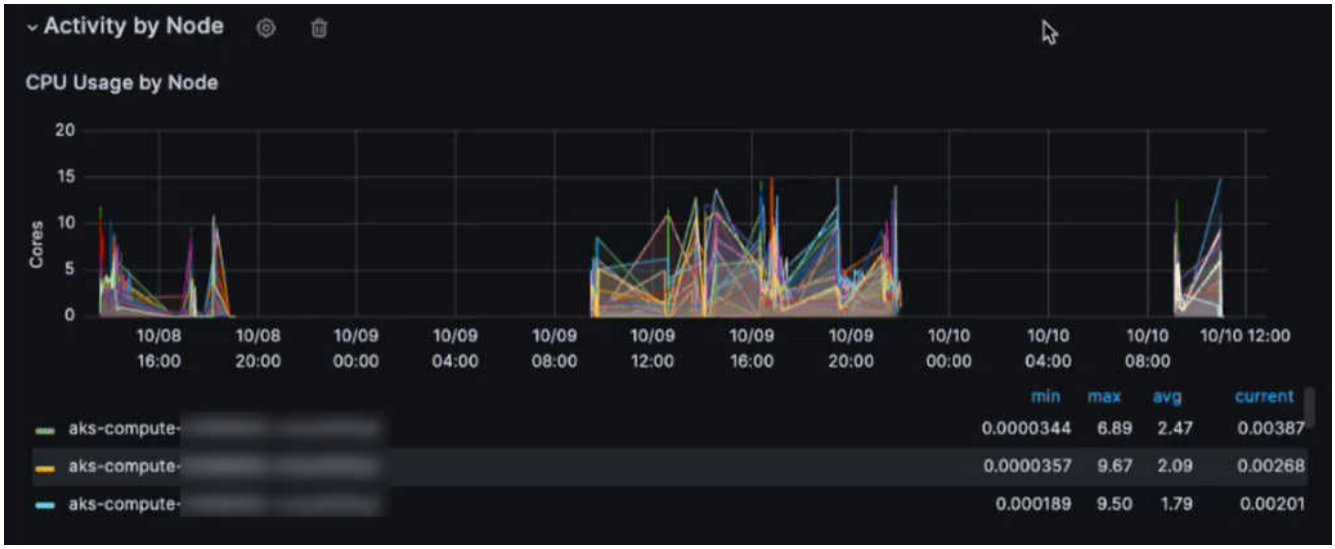
  (User Activity dashboard) In the table, click on a user to view only their data in the chart.

Identify a Node with Saturated or Under-Used Cores (CPUs)

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **CPU Usage by Node** chart is located on the Node Activity dashboard. This chart shows how the SAS Launcher service jobs use the CPUs on the selected nodes or all nodes across a selected range of time. Time is measured on the X axis of the chart. Cores are measured on the Y axis.

Note: When you view CPU use in a small enough range of time, the Y-axis shows millicores (1/1000 of a single CPU or core). These are slices of a CPU and indicate that the CPU is under used.



In this example, each node has 16 cores or CPUs. (You must know how many cores are available for each of your nodes.) Each node shows spikes of CPU use. Quick spikes are typical when processing SAS Compute server, SAS Launcher service, SAS Batch server, and SAS Workload Orchestrator.

However, if a node spikes into a flat line at the top of its available cores (in this example, 16), then it is peak flat-lined and the CPUs are saturated. When this happens, the affected node does not have enough CPUs to process the workload. Typically, the solution is to upgrade to a node with more CPUs or run the CPU-intensive job during downtimes.

If the problem persists after adding CPUs, contact SAS technical support.

Identify a Node with a Memory Leak

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **Memory Usage by Node** chart is located on the Node Activity dashboard. This chart shows how the SAS Launcher service jobs use memory on the selected nodes or all nodes across a selected range of time. Time is measured on the X axis of the chart. Memory is measured on the Y axis.

In this example, each node has 128 GiB of memory. (You must know how much memory is available for each of your nodes.) Each node shows spikes of memory use, but all of the spikes are much lower than 128 GiB. These spikes are typical when processing SAS Compute server, SAS Launcher service, SAS Batch server, and SAS Workload Orchestrator.

However, if a node spikes into a flat line at the top of its available memory, then it is peak flat-lined. When this happens, the affected node does not have enough memory to process the workload. Typically, this indicates a memory leak. Contact SAS technical support.

Observe and Tune Your Environment

You can use the **CPU Usage by Node** and **Memory Usage by Node** charts to observe a test environment during stress tests. You can learn which nodes are over- or underused when processing your organization's workloads. When you have this information, you can tune the performance of your system before moving to a production environment.

After you move the system to a production environment, use these charts to review your organization's workload to confirm that your organization's usage is as expected. If users start to encounter unexpected errors, review these dashboards to see if the affected nodes have saturated CPUs or memory use.

You can also use these charts to identify whether a specific SAS Launcher service pod is using too many resources on the system.

See Also

- *SAS Viya Platform Administration: Tuning*

Monitor File System I/O Activity by Node

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

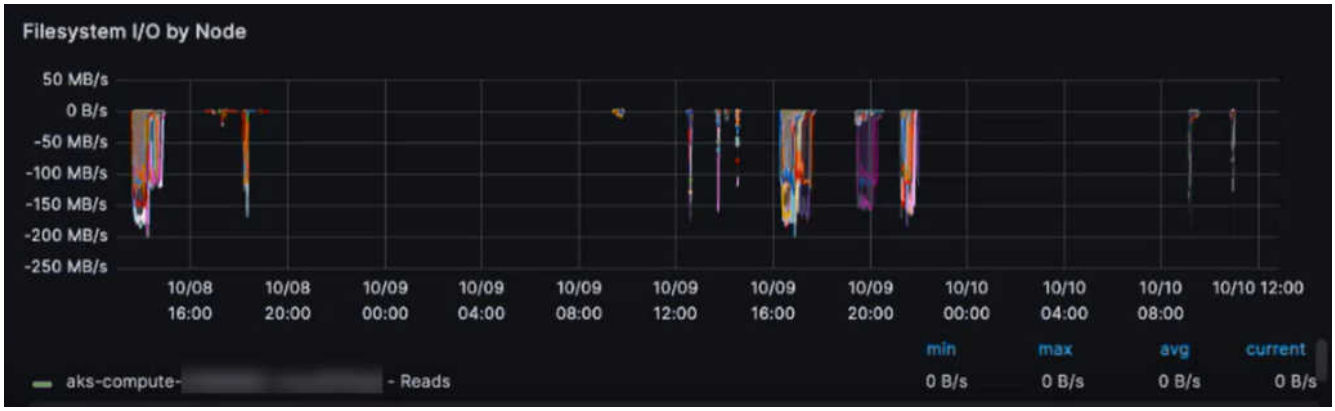
The **Filesystem I/O by Node** chart is located on the Node Activity dashboard. This chart shows each node's Read and Write activity in the file system for the selected SAS Launcher service pods across a

selected range of time. Time is measured on the X axis of the chart. The amount of data read or written is measured on the Y axis.

- Negative values (below zero) show Write activity. The value is the amount of data written to the file system.

  Typically, most of the activity shown is Write activity. If you are monitoring SAS Batch server, the chart displays a lot of write activity because the batch server is writing out file sets and data from each batch job.

- Positive values show Read activity. The value is the amount of data read from the file system.

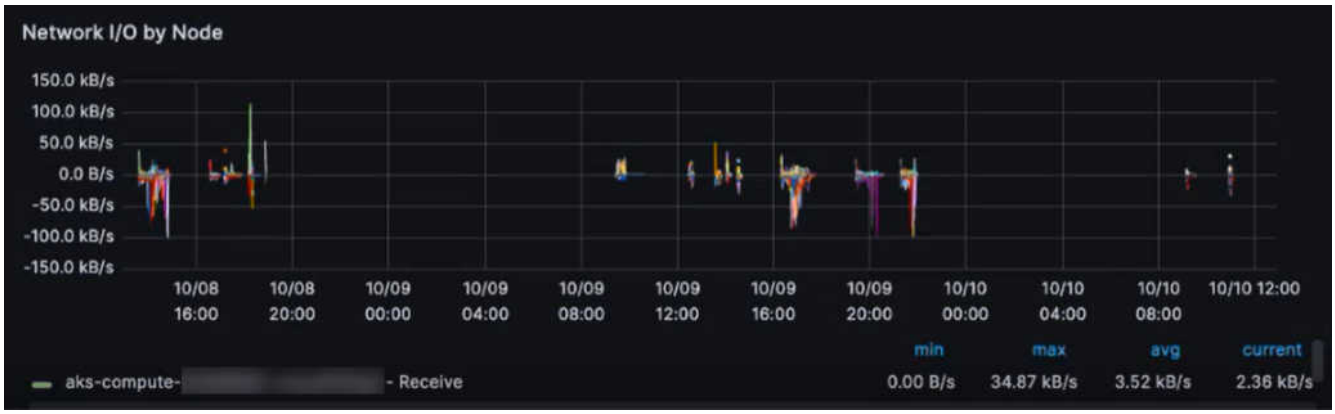


Monitor Network I/O Activity by Node

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **Network I/O by Node** chart is located on the Node Activity dashboard. This chart shows each node's network transmission activity for the selected SAS Launcher service pods across a selected range of time. Time is measured on the X axis of the chart. The amount of data transmitted is measured on the Y axis.

- Negative values (below zero) show data that the node is transmitting across the network. The value is the amount of data transmitted across the network.
- Positive values show data that the node is receiving from the network. The value is the amount of data received from the network.



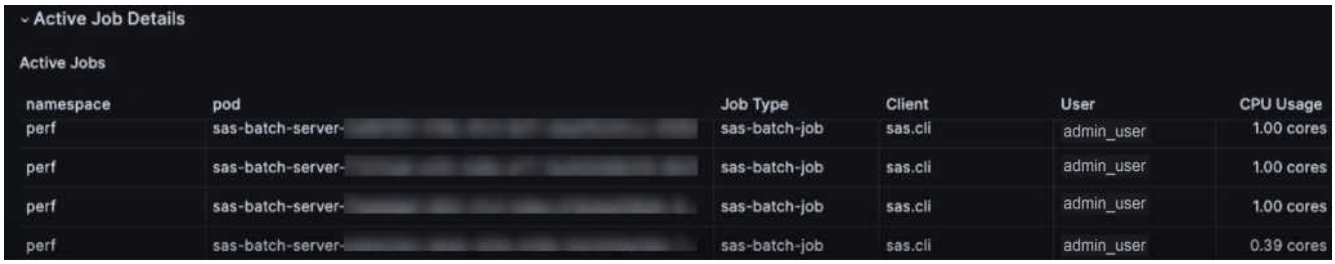
In this example, each node has 150 kB/s total for transmitting and receiving data. None of the spikes shown total more than 150 kB/s. If there were spikes that exceeded the node's network I/O limit, it could indicate congested network traffic and other network issues. Contact your organization's team that manages your networks.

View Active Job Details

Note: For more ways to use the **Active Jobs** table, see "Filter by Pod" on page 26.

The **Active Jobs** table is located on both the Node Activity and User Activity dashboard. Scroll down to the bottom of each dashboard to see the table. This table displays all of the jobs in the selected time range. The table lists the namespace, pod name, job type, client, user, and CPU usage for each job.

> **TIP** The job type and the pod name indicate whether the job belongs to SAS Compute server, SAS Launcher service, or SAS Batch server.

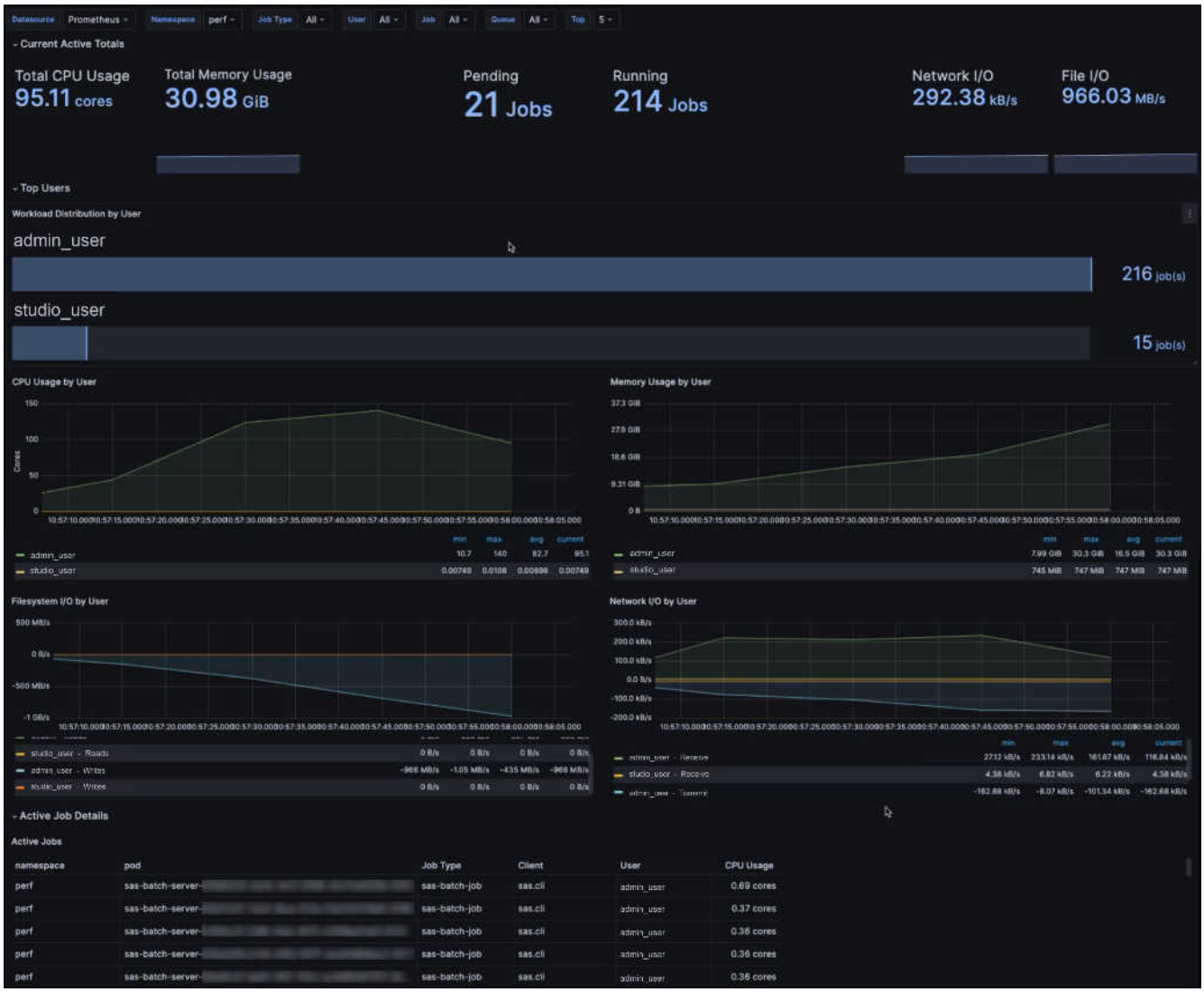


You can click on any cell in the table to filter the data displayed by the entire dashboard. For example, If you click on a job name in the table, the data on the dashboard is filtered by the job.

View Metric Data by User

The SAS Launched Jobs - User Activity dashboard displays the metric data by user instead of by node. The dashboard displays the users who are using the system during the selected time range.

In the following example, two users are on the system during the time range: admin_user and studio_user. The admin_user is running SAS Batch server jobs. The studio_user is running SAS Compute server jobs via SAS Studio.



Reviewing metric data by user can enable you to evaluate the types of jobs that each user runs. If you identify users who run large process-intensive jobs, you can create a queue to handle those jobs. Small, quickly processed jobs can remain in another queue. If necessary, you can create a queue for a specific user.

See Also

- *SAS Viya Platform: Workload Management*

Identify a User's CPU Processing Needs

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **CPU Usage by User** chart is located on the User Activity dashboard. This chart shows CPU processing across a range of time by user. Time is measured on the X axis of the chart. Cores are measured on the Y axis.

Note: When you view CPU use in a small enough range of time, the Y-axis shows millicores (1/1000 of a single CPU or core). These are slices of a CPU and indicate that the CPU is under used.

In this example, each node has 16 cores or CPUs. (You must know how many cores are available for each of your nodes.) The selected user is studio_user. This chart enables you to identify users who run large process-intensive jobs. You can create a queue to handle those jobs.

View Memory Use by User

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **Memory Usage by User** chart is located on the User Activity dashboard. This chart shows how each user uses memory across a selected range of time. Time is measured on the X axis of the chart. Memory is measured on the Y axis.

In this example, each node has 128 GiB of memory. (You must know how much memory is available for each of your nodes.) The selected user is studio_user. This user's memory use is well below the 128 GiB limit.

Monitor File System I/O Activity by User

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **Filesystem I/O by User** chart is located on the User Activity dashboard. This chart shows the read and write activity in the file system for the selected SAS Launcher service pods across a selected range of time by user. Time is measured on the X axis of the chart. The amount of data read or written is measured on the Y axis.

- Negative values (below zero) show write activity. The value is the amount of data written to the file system.

  Typically, most of the activity shown is write activity. If you are monitoring SAS Batch server, the chart displays a lot of write activity because the batch server is writing out file sets and data from each batch job.
- Positive values show read activity. The value is the amount of data read from the file system.

This chart can help identify a user (and their job) that writes frequently or excessively to the file system.

Monitor Network I/O Activity by User

Note: Be sure to review "General Tips about How to Use the Charts" on page 29.

The **Network I/O by User** chart is located on the User Activity dashboard. This chart shows network transmission activity for the selected SAS Launcher service pods across a selected range of time by user. Time is measured on the X axis of the chart. The amount of data transmitted is measured on the Y axis.

- Negative values (below zero) show data that the user's jobs are transmitting across the network. The value is the amount of data transmitted across the network.
- Positive values show data that the user's jobs are receiving from the network. The value is the amount of data received from the network.

As you become familiar with the usage profiles in your environment, you are able to spot trends for both regular and excessive usage.

Monitoring SAS CAS Server

About the SAS CAS Server Dashboard

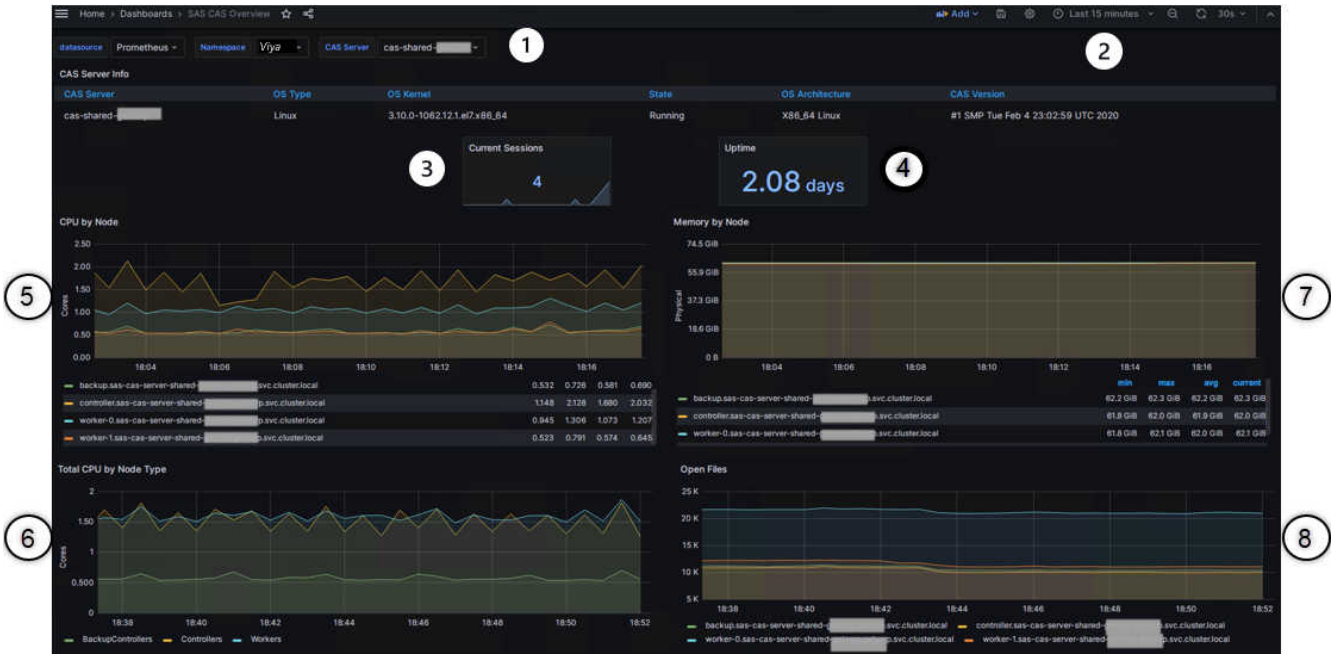
The SAS CAS Overview Dashboard enables you to monitor the health of your CAS server. It displays CPU usage and memory usage as open files for a selected CAS server.

For an overview of the CAS Server and what is monitored, see "Monitoring SAS CAS Server" in *SAS Viya Platform: Troubleshooting*.

Note: If you enabled log-enhanced Grafana dashboards, the dashboard is called SAS CAS Overview with Logs. It is identical to the dashboard shown here with the additional capability of showing log messages below the metric charts. The generated log messages that are available are from the selected CAS server during the selected time period. See "Display Log Messages in Grafana Dashboards" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

The Dashboard User Interface

In Grafana, navigate to the SAS CAS Overview dashboard. See "Change to Another Dashboard" on page 11.



1 SAS Server list: if you have more than one CAS server, select the CAS server that you want to monitor from the drop-down arrow next to **CAS Server**.

2 Time Range list: modify your time period to the appropriate time interval for monitoring. Use the time selector that is located on the toolbar at the top of the dashboard to change the time interval that is displayed in the panel.

Note: Click on a point in time on a graph in a panel and drag your mouse to a different point in time to zoom in.

3 **Current Sessions**: indicates the number of sessions that are connected to the CAS server.

4 **Uptime**: indicates the amount of time that the CAS server has been running.

5 **CPU by Node**: indicates the CPU activity of CAS server pods over the specified time period. The lines represent the distinct CAS server role and the names are displayed below the graph: controller, backup controller, and one or more workers (MPP environment).

6 **Total CPU by Node Type**: indicates the total CPU activity of CAS server pods over a specified time period. The CAS server worker pods are combined and displayed as one line.

7 **Memory by Node**: indicates the memory that is consumed by the CAS server pods over the specified time period. The lines represent the distinct CAS server role and the names are displayed below the graph: controller, backup controller, and one or more workers (in an MPP environment).

8 **Open Files**: indicates the number of open files on the pod. This includes active input/output resources linked to the Operating System, running executables or scripts, and files for CAS tables. Knowing this critical data ensures optimal resource utilization since open files mean resource consumption such as kernel data structures and memory.

Ways to Use the SAS CAS Overview Dashboard

Here are some suggestions for using the SAS CAS Overview Dashboard:

- View the dashboard regularly to gain an understanding of baseline metrics and what is healthy. Metrics are collected at regular intervals that are frequent enough to understand usage, but not so frequent as to cause performance issues.
- View the dashboard to see if there is anything out of the ordinary. This could include spikes in CPU or memory, or a CAS server being down.
- Use the dashboard for endurance or performance testing within a specific time period.
- Look for trends or patterns to understand what has happened in the system at this time.
- After seeing irregularities in the dashboard, record the time and look at logs for anomalies. Start with the main CAS controller log, then logs for any running sessions. Alternatively, take a screenshot and continue to monitor to see if there are similar occurrences.

Note: If you find that CAS sessions are killed with out-of-memory (OOM) errors, see "Session Terminated by cgroup OOM Error or CAS Pod Evicted by Host OOM Error" in *SAS Viya Platform: SAS Cloud Analytic Services*.

See Also

"SAS Cloud Analytic Services: Troubleshooting" in *SAS Viya Platform: SAS Cloud Analytic Services*

Exporting Logs with getlogs.py

About the getlogs.py Script

There might be situations when you need to collect log messages and send them to a file. For example, you might need to send log messages that are collected during a specific time period, or for a specific pod, to SAS Technical Support to help diagnose a problem. SAS Viya Monitoring for Kubernetes includes the `getlogs.py` script that enables you to export log messages from the OpenSearch database.

The `getlogs.py` script connects to an OpenSearch API and generates an OpenSearch domain-specific language query (DSL) to the API to retrieve logs.

The `getlogs.py` script supports two methods for connecting to your OpenSearch Database:

- "Connect to OpenSearch with Kubernetes Port-Forwarding" on page 44
- "Connect to OpenSearch with Direct Link to Host" on page 45

User-supplied parameters enable you to construct a query in DSL format. For example, you can specify parameters to return log messages in the following ways:

- return log messages during a specified time period
- return log messages for a source or logging level
- return log messages that contain a specified word in the message field
- return log messages to a file on disk

The `getlogs.py` script is found in the **`logging/bin`** subdirectory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

See "General Syntax" on page 41 for the parameters that the `getlogs.py` script uses.

Note: In order to run the `getlogs.py` script, you must have installed:

- Python 3.11 or a later version
- OpenSearch library:

```
pip install opensearch-py
```

General Syntax

`python3 logging/bin/getlogs.py` *`[arguments]`* `where:`

- you are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.
- *[arguments]* are user-supplied parameters. The available parameters are listed below.

Connection Parameters

- `--user` or `-u`

  The user name that is used to connect to OpenSearch. The default is $OSUSER.

- `--password` or `-p`

  The password that is used to connect to OpenSearch. The default is $OSPASSWD.

- `--port-forward` or `-pf`

  Use this option if you connect to the OpenSearch API through port-forwarding. You must create the $KUBECONFIG environment variable as `getlog.py` uses this value to port-forward. You do not need to include the `--host` and `--port` options (or set the corresponding environment variables). However, you do need to provide user credentials through the `--user` and `--password` options (or set the corresponding environment variables).

- `--host` or `-ho`

The host name where OpenSearch is deployed. The default is $OSHOST. Do not include 'https://' in the host name.

- `--port` or `-po`

  The port number that is used for the OpenSearch instance. The default is $OSPORT.

- `--disable-ssl` or `-nossl`

  If you use this option, SSL is not used to connect to the database.

For usage examples, see "Connect to OpenSearch with Kubernetes Port-Forwarding" on page 44, and "Connect to OpenSearch with Direct Link to Host" on page 45.

File Output and Query Output Parameters

- `--out-file` or `-o`

  Use this option to create a file to which the query results are written. If this option is not used, the results are written to `STDOUT`. The file format is CSV unless you specify `json` on the `--format` parameter.

  > **IMPORTANT** The `getlogs.py` script functions with only files that are in the current working directory, in order to prevent path traversal attacks. Any parameters that use user-provided file paths require the selected files to be in the same working directory as the `getlogs.py` script.

- `--format` or `-fo`

  Use this option to specify the format of the file that you create with the `--out-file` parameter. Use lowercase `json` or `csv`. The default format output is CSV.

- `--force` or `-f`

  Use this option to overwrite results if the file that is specified already exists in the current working directory.

- `--fields` or `-fi`

  Use this option to specify the columns that you want to include in your query output. The default fields are `@timestamp`, `level`, `kube.pod`, `message`, and `_id`. If you use this option, you must include any of the default fields that you want to include, with the exception of `_id`. The `_id` field is always included in the output as it is a unique identifier that is associated with each log message that is captured.

  Note: If a matching log message does not have the specified field, a NULL value is used as a placeholder.

- `--maxrows` or `-m`

  Use this option to specify the maximum number of log messages to return. By default, a maximum of 10 log messages that match the selection criteria are returned.

- `--show-query` or `-sh`

  Use this option to view a query in JSON format that is submitted to the OpenSearch API.

- `--save-query` or `-sq`

  Use this option to save the DSL query that is generated in order to reuse it at a later time. Specify a file name, but do not include the file extension in the name. The query is saved as a JSON file in the current working directory.

  > **IMPORTANT** The `getlogs.py` script functions with only files that are in the current working directory, in order to prevent path traversal attacks. Any parameters that use user-provided file paths require the selected files to be in the same working directory as the `getlogs.py` script.

- `--query-file` or `-q`

  Use this option to specify a saved query that is in the current working directory. The script submits the query from the file. All other query parameters are ignored.

Query Search Parameters

Note: Separate multiple values on parameters with spaces. Do not use single quotation marks in arguments.

- `--namespace` or `-n`
- `--namespace-exclude` or `-nx`

  Use these options to filter log messages from SAS Viya deployment namespaces or Kubernetes namespaces to include in the output or exclude from the output.

- `--pod` or `-p`
- `--pod-exclude` or `-px`

  Use these options to filter log messages from pods to include in the output or exclude from the output.

- `--container` or `-c`
- `--container-exclude` or `-cx`

  Use these options to filter log messages from containers to include in the output or exclude from the output.

- `--logsource` or `-s`
- `--logsource-exclude` or `-sx`

  Use these options to filter log messages from log sources to include in the output or exclude from the output.

- `--level` or `-l`
- `--level-exclude` or `-lx`

  Use these options to filter log messages from message levels to include in the output or exclude from the output. Message levels are: `NONE`, `INFO`, `WARNING`, `ERROR`, `FATAL`, `TRACE`, and `DEBUG`.

- `--search` or `-se`

Use this option to search for words or phrases in the message field. Use quotation marks to surround your text string. The text search is case insensitive.

- `--start` or `-st`

  Use this option to modify the start date and time for which logs are requested. The default is 1 hour prior to the current date and time. Use this format: Y-M-D H:M:S. Here is an example: 2023-02-16 10:00:00.

- `--end` or `-en`

  Use this option to modify the end date and time for which logs are requested. The default is the current date and time. Use this format: Y-M-D H:M:S. Here is an example: 2024-01-16 10:00:00.

  The `--start` and `--end` parameters use the industry standard ISO-8601 Y-M-D H:M:S format, where:

  - Y is a four-digit year
  - M is a two-digit month
  - D is a two-digit day
  - H is a two-digit hour

    The two-digit hour uses the 24-hour clock where midnight is 00, noon is 12, and 11 PM is 23.
  - M is a two-digit minute
  - S is a two-digit second

Note: By default, log messages are saved on a rotating three-day cycle in the OpenSearch database. The retention period is configurable. See "Adjust the Retention Period for Log Messages" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

Usage Examples

Connect to OpenSearch with Kubernetes Port-Forwarding

The `getlogs.py` script supports connecting to OpenSearch using Kubernetes Port-Forwarding. This is especially useful when OpenSearch is not exposed from outside of the cluster. To use port-forwarding, set a $KUBECONFIG environment variable to reference the `kubeconfig` file, which grants access to your cluster. The `getlogs.py` script uses that file to port-forward the OpenSearch service to a locally available port on your machine and connect to OpenSearch. Perform the following steps:

1 Set your KUBECONFIG environment variable:

```
export KUBECONFIG=directory_path_to_kubeconfig_file
```

2 Use the following parameters on `getlogs.py`:

- `--user` (`-u`)
- `--password` (`-p`)
- `--port-forward` (`-pf`)

The `--port-forward` argument is necessary. The script uses the value in your KUBECONFIG environment variable to port-forward and connect to the OpenSearch API in the specified namespace. Because of this, `--host` and `--port` are not necessary, but `--user` and `--password` are still required to authenticate and connect to the database.

Here is an example:

Note: The following example assumes that you are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

```
python3 logging/bin/getlogs.py --pf --user username_for_OpenSearch_dashboards
--password password_for_OpenSearch_dashboards
```

Note: Alternatively, set connection information in environment variables. By default, the last 10 log messages are returned in CSV format.

```
#Set env. variables. $OSHOST and $OSPORT are not needed when connecting through port-fowarding.
export OSUSER=username_for_OpenSearch_dashboards
export OSPASSWD=password_for_OpenSearch_dashboards
export KUBECONFIG=directory_path_to_kubeconfig_file

python3 logging/bin/getlogs.py -pf
```

Connect to OpenSearch with Direct Link to Host

If your OpenSearch instance is accessible from outside of the cluster with Ingress or NodePorts, use the following parameters to connect to OpenSearch:

- `--user` (`-u`)
- `--password` (`-p`)
- `--host` (`-ho`)
- `--port` (`-po`)

Here is an example:

Note: The following example assumes that you are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

```
python3 logging/bin/getlogs.py --user username_for_OpenSearch_dashboards
--password password_for_OpenSearch_dashboards --host host_where_OpenSearch_runs
--port port_of_OpenSearch_API
```

Note: Alternatively, set connection information in environment variables and run `getlogs.py` without arguments. By default:

- The script uses $OSUSER, $OSPASSWD, $OSHOST, and $OSPORT environment variables.
- The last 10 log messages are returned in CSV format.

```
export OSUSER=username_for_OpenSearch_dashboards
export OSPASSWD=password_for_OpenSearch_dashboards
export OSHOST=host_where_OpenSearch_runs
export OSPORT=port_of_OpenSearch_API

python3 logging/bin/getlogs.py
```

See Also

- The Kubernetes Ingress configuration sample file information in "About the Sample Files" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*
- "Configure Access to OpenSearch Dashboards via NodePorts" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*

Create a File with WARNING and ERROR Log Messages

Note: This example makes the following assumptions:

- You are connecting to OpenSearch through Kubernetes port-forwarding. See "Connect to OpenSearch with Kubernetes Port-Forwarding" on page 44 for more information.
- You are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

```
python3 logging/bin/getlogs.py -pf -l WARNING ERROR -o WarnErrMsgs -fo json --force
```

Where:

- The `-pf` parameter signifies that a KUBECONFIG file is used to port-forward the OpenSearch service and connect. The $KUBECONFIG, $OSUSER, and $OSPASSWRD environment variables must exist.
- The `-l WARNING ERROR` parameter searches for WARNING and ERROR values in the level field and returns those rows.
- The `-o WarnErrMsgs` parameter writes the returned log messages to a file in the current working directory that is named WarnErrorMsgs.
- The `-fo json` parameter writes the WarnErrMsgs file in JSON format. (The default format type is CSV.)
- The `--force` parameter overwrites a same-named file if it exists in the current working directory.

Modify Fields and Number of Rows to Output

Note: This example makes the following assumptions:

- You are connecting to OpenSearch through Kubernetes port-forwarding. See "Connect to OpenSearch with Kubernetes Port-Forwarding" on page 44 for more information.

- You are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

```
python3 logging/bin/getlogs.py -pf --fields @timestamp level message kube.host --
maxrows 20 --pod sas-cas-server-default-controller
```

Where:

- The `-pf` parameter signifies that a KUBECONFIG file is used to port-forward the OpenSearch service and connect. The $KUBECONFIG, $OSUSER, and $OSPASSWRD environment variables must exist.
- The `--fields @timestamp level message kube.host` parameter identifies the fields that are included in the output. The default set of fields that are included are @timestamp, level, kube.pod, message, and _id. If you want a different set of fields to be included, use the `--fields` parameter. The id field is always included in the output.
- The `--maxrows 20` parameter specifies that a maximum of 20 rows are included in the output.
- The `--pod sas-cas-server-default-controller` specifies that you want log messages from the cas-server-default-controller pod to be included in the output.

Show Query and Save Query to a File

Note: These examples make the following assumptions:

- You are connecting to OpenSearch through Kubernetes port-forwarding. See "Connect to OpenSearch with Kubernetes Port-Forwarding" on page 44 for more information.
- You are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

```
#create env. variable with Viya namespace.
export VIYA_NS=Viya_namespace
```

```
python3 logging/bin/getlogs.py -pf --pod sas-logon* --show-query --save-query
saslogonQuery
```

Where:

- The `-pf` parameter signifies that a KUBECONFIG file is used to port-forward the OpenSearch service and connect. The $KUBECONFIG, $OSUSER, and $OSPASSWRD environment variables must exist.
- The `--pod sas-logon*` parameter returns log messages for the SAS Logon Manager service (or any pod that begins with `sas-logon`).
- The `--show-query` parameter displays the query in the terminal (STDOUT).
- The `--save-query saslogonQuery` saves the query to a file that is named saslogonQuery in the current working directory. The format type is JSON.

```
python3 logging/bin/getlogs.py -pf --level INFO WARNING ERROR --namespace $VIYA_NS --
show-query --save-query ViyaLogsQuery -o ViyaLogMsgs.csv -f
```

Where:

- The `-pf` parameter signifies that a KUBECONFIG file is used to port-forward the OpenSearch service and connect. The $KUBECONFIG, $OSUSER, and $OSPASSWRD environment variables must exist.
- The `--level INFO WARNING ERROR` parameter returns log messages that have a logging level of INFO, or WARNING, or ERROR.
- The `--namespace` *`$VIYA_NS`* parameters return log messages that are in the $VIYA_NS namespace.
- The `--show-query` parameter displays the query in the terminal (STDOUT).
- The `--save-query ViyaLogQuery` parameter saves the query to a file that is named ViyaLogsQuery.json in the current working directory.
- The `-o ViyaLogMsgs.csv` parameter saves the requested log messages to a file that is named ViyaLogMsgs.csv in the current working directory.
- The `-f` parameter overwrites the results if the file that is specified on the `-o` parameter already exists in the current working directory.

Request Log Messages from Last 24 Hours

Note: This example makes the following assumptions:

- You are connecting to OpenSearch through Kubernetes port-forwarding. See "Connect to OpenSearch with Kubernetes Port-Forwarding" on page 44 for more information.
- You are in the root directory of your local copy of the GitHub repository. See "Copy the Repository" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes*.

Note: Change the Y-M-Y H:M:S formatted date and time on the `--start` parameter to 24 hours prior to the current date and time before you run the command.

```
#create env. variable with Viya namespace.
export VIYA_NS=Viya_namespace

   python3 logging/bin/getlogs.py -pf -o LogMsgsOutput.json -fo json -f --search 'Error
   Proxying' --fields @timestamp level message kube.pod  --start 2024-01-07 00:00:00 --
   save-query ProxyErrorQuery -m 40
```

Where:

- The `-pf` parameter signifies that a KUBECONFIG file is used to port-forward the OpenSearch service and connect, and the $KUBECONFIG, $OSUSER, and $OSPASSWRD environment variables exist.
- The `-o` parameter pipes the output to a file that is named LogMsgsOutput.json in the current working directory.
- The `-fo` parameter formats the log file in JSON.
- The `-f` parameter overwrites the file's content if the file already exists.
- The `--search` parameter searches for log messages that have "Error Proxying" text anywhere in the message field. It is case insensitive.

- The `--fields` parameter prints the `timestamp`, `level`, `message`, and `kube.pod` fields of the log messages that are returned.
- The `--start 2024-01-07 00:00:00` parameter starts the log search on the date that you supply in the following format: YYYY-MM-YY HH:MM:SS.
- The `--save-query` parameter saves the query in a file that is named ProxyErrorQuery.json in the current working directory.
- The `--maxrows 40` parameter specifies that a maximum of 40 rows are included in the output.

Accessing Observability Tools Via SAS Environment Manager

Access the Log-Monitoring Application from SAS Environment Manager

To access your application after configuration in SAS Environment Manager, see "Access Your Log-Monitoring Application" in *SAS Environment Manager: User's Guide*.

Troubleshooting

The Kubernetes-Proxy Dashboard Does Not Display Any Data

Problem:

The Grafana dashboard for Kubernetes-Proxy does not display any data.

Explanation:

The kube-proxy metrics are used in the Kubernetes-Proxy dashboard. Some clusters are deployed with the kube-proxy metrics listening address set to 0.0.0.0:10249. This address prevents Prometheus from collecting metrics. Therefore, the dashboard does not display any data.

It is also possible that the cluster does not use kube-proxy. To determine whether a cluster uses kube-proxy, enter the following command:

```
kubectl -n kube-system get pods | grep proxy
```

If no results are returned, kube-proxy is not used in the cluster.

Solution:

To enable collection of the kube-proxy metrics, complete the following steps:

1 To edit the YAML file of the configMap for kube-proxy, enter the following command:

```
kubectl edit cm -n kube-system kube-proxy
```

Note: This command opens the system default editor.

2 Locate `metricsBindAddress` in the file. If it is set to `0.0.0.0:10249`, replace that setting with **`127.0.0.1:10249`**.

3 To re-create the pods so that they use the new configuration, enter the following command:

```
kubectl delete po -n kube-system -l k8s-app=kube-proxy
```

SAS Launched Jobs Dashboards Do Not Work

Problem:

The following dashboards do not have metrics:

- SAS Launched Jobs — Node Activity
- SAS Launched Jobs — User Activity

Explanation:

Both of these dashboards require that either SAS Workload Orchestrator is enabled or that the feature has a license in your SAS Viya platform deployment.

Solution:

See "Dashboards That Require a License or Enablement" in *SAS Viya Platform: Deploying SAS Viya Monitoring for Kubernetes* for details.