



SAS[®] Viya[®] 3.4 Administration: Docker Containers

SAS Viya for Containers: Cheat Sheet

This document is intended for administrators who are fluent in Docker or Kubernetes and who have the required privileges to run Docker or Kubernetes commands. Refer to the respective Docker or Kubernetes documentation for more information about running commands.

Table 1 SAS Viya for Containers: Cheat Sheet

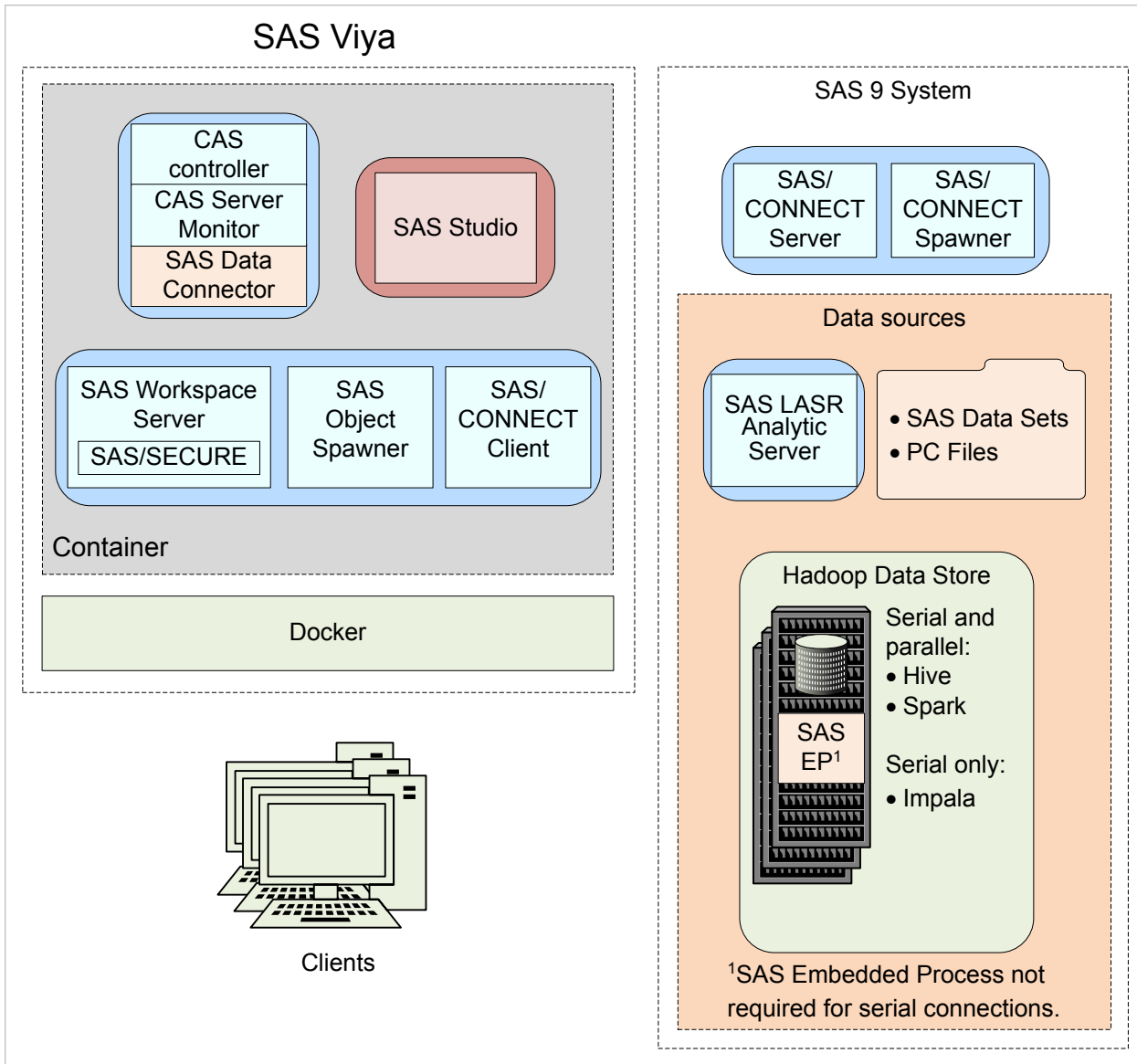
Subject	Notes and References
Architecture	See “SAS Viya for Containers: Architecture” on page 2.
Authentication	<p>By default, a containerized deployment of SAS Viya is secured by requiring host authentication to SAS Studio. Additional security, including LDAP authentication, requires a new container, which you can build based on the SAS Viya image in the Docker repository. Using this method, you can add an authentication layer, such as the Linux System Security Services Daemon (SSSD), and integrate the authentication scheme with LDAP.</p> <p>For more information, see SAS Viya for Containers: Deployment Guide</p> <p>For all other authentication topics, see SAS Viya Administration: Authentication.</p>
Backups	“SAS Viya for Containers: Backup and Recovery” on page 3
Bridging SAS 9.4 and SAS Viya	SAS Viya for Containers: Deployment Guide
CAS Configuration	“SAS Viya for Containers: SAS Cloud Analytic Services” on page 9
Data	The following path must be accessible to all CAS users: <code>/dev/shm</code> .
Licenses	<p>Software license information is provided by environment variable or on a file on disk. To renew, update the environment variable or the file on disk and redeploy.</p> <p>See “SAS Viya for Containers: Licensing” on page 4.</p>

Subject	Notes and References
Logs	To access logs, use one of the following commands: <ul style="list-style-type: none">■ <code>docker logs</code>■ <code>kubectl logs</code>
Monitoring	See “SAS Viya for Containers: Monitoring” on page 5.
Troubleshooting	See “SAS Viya for Containers: Troubleshooting” on page 11.
Updates	See SAS Viya for Containers: Deployment Guide.

SAS Viya for Containers: Architecture

SAS Viya for Containers includes a programming-only deployment. A programming-only deployment supports data scientists and programmers who use SAS Studio or direct programming interfaces such as Python or REST APIs. Understand that this type of deployment does not include SAS Drive, SAS Environment Manager, and the complete suite of services that are included with a full deployment.

- The supported CAS server mode is symmetric multi-processing (SMP).
- Multiple single instances of SAS Viya can be deployed on a single host.
- Hadoop is supported, but co-locating the CAS server with Hadoop is not supported.



SAS Viya for Containers: Backup and Recovery

Guidelines

Consider the following guidelines when implementing the backup and restore processes:

- Persistent storage attached to the containers can ensure that data sources that are not handled by the backup utilities can still be retained and recovered.
- Assets that retain information or the state between CAS sessions or SAS sessions (for example, CAS controls and caslib information) should be backed up on persistent disks. User home directories must also be NFS mounts and capable of being backed up through your organization's standard persistent disk backup procedures.

Retrieve a Copy of the Kubernetes Manifest

For backup and archival purposes, retrieve a copy of the running deployment manifest from the Kubernetes environment.

- 1 Make sure that `kubectl` is pointed to the correct environment.
- 2 Run the following command to get a list of the deployed pods:

```
kubectl get pods
```

Look for a pod that begins with `sas-programming`.

- 3 Run the following command:

```
kubectl get pod -o yaml name-of-pod > local-file-name
```

- 4 Copy the manifest file to an archive location for backup purposes.

SAS Viya for Containers: Licensing

Read This First

Information about SAS Viya configuration properties can be found in SAS Viya Administration: Licensing. Any differences specific to the Docker and Kubernetes platforms are described in this section.

- SAS Viya uses a single licensing file. The contents of the license file resides in an environment variable or in a file on disk that is mounted to the container.
You apply a new license to enable new products or to extend expiration dates on existing products.
- Both SAS Cloud Analytic Services (CAS) and SAS Foundation use the same license. During the deployment, a license is applied to both the CAS in-memory compute engine and the SAS Foundation compute engine.

Apply a New License

Perform one of the following steps to apply a new license.

For Docker environments:

- 1 Copy the new license file to the `sasinside` directory that was mounted to the running container. Make sure that it is named `license.sas`.
- 2 Run `docker restart sas-programming` to stop and start the running container, which applies the new license.
Note: To pick up a new license, the Docker container must be restarted.
- 3 Verify that the license has been applied to CAS and SAS Foundation. See the following topics:
 - [“View SAS Programming Run-Time License Information” in SAS Viya Administration: Licensing](#)
 - [“View SAS Cloud Analytic Services License Information” in SAS Viya Administration: Licensing](#)

For Kubernetes environments:

1 Use the `kubectl get configmaps` command to get the defined ConfigMap. Look for `sas-programming` in the resulting list.

2 Run the following command to get the `sp_configmap.yml` file:

```
kubectl get configmap -o yaml sas-programming > sp_configmap.yml
```

3 Edit the `sp_configmap.yml` file to add the contents of the new license file.

Important: When you insert the license information, ensure that you maintain the indentation from the license file.

Here is an example of the license information:

```
poac.SETINIT_TEXT: |
  PROC SETINIT RELEASE='V03';
  SITEINFO NAME=your-site-name'
  SITE=site-number OSNAME='LIN X64' RECREATE WARN=47 GRACE=45
```

4 Run the following command to update the ConfigMap.

```
kubectl replace -f sp_configmap.yml
```

5 Restart the container.

a Run the `kubectl get pods` command to get a list of the running pods. Look for the pod that begins with `sas-programming`.

b Run the following command to stop the current pod:

```
kubectl delete pod name-of-pod
```

Kubernetes creates a new pod that has the new license applied.

6 Verify that the license has been applied to CAS and SAS Foundation. See the following topics:

- [“View SAS Programming Run-Time License Information” in SAS Viya Administration: Licensing](#)
- [“View SAS Cloud Analytic Services License Information” in SAS Viya Administration: Licensing](#)

SAS Viya for Containers: Monitoring

Read This First

Generally, information about monitoring can be found in [SAS Viya Administration: Monitoring](#). Any differences specific to SAS Viya for Containers are described in this section.

Access CAS Server Monitor

Note: To access CAS Server Monitor, you must have an active CAS server session. For more information, see [“Access the Monitor” in SAS Viya Administration: SAS Cloud Analytic Services](#).

To log on to CAS Server Monitor, open a web browser and enter one of the following URLs:

- For a Docker deployment:

```
http://host-name-where-docker-is-running:port/cas-shared-default-http
```

- For a Kubernetes deployment:


```
http://path-to-ingress-controller:port/cas-shared-default-http
```

TIP About the port:

- If the port is 80, entering it is optional. If the port is something other than 80, you must enter it.
- To locate the correct port for a Docker deployment, run `docker ps` and locate the port on the Docker host that is mapped to port 80 in the container.
- For a Kubernetes deployment, the path to the Ingress controller is assumed to point to the service running on port 80. Depending on the Ingress controller configuration, the URL for Kubernetes can vary.

Monitor CAS Process Performance

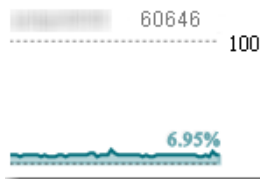
The CAS processes you can monitor with these steps correspond to SAS server processes. You can separately monitor each session that is started from the CAS server.

- 1 In CAS Server Monitor, beneath the Cloud Analytic Services banner, click .

- 2 Select **Add View** ⇒ **CAS Process CPU Usage**.

The **Process CPU Usage** panel on the window displays a set of histograms. There is one histogram for each machine and the corresponding CAS server process. The histogram in the upper left is the CAS controller node. If you are not an administrator, only the histogram for the CAS controller node is displayed.

Each histogram displays the percentage of CPU usage, from 0 to 100%.

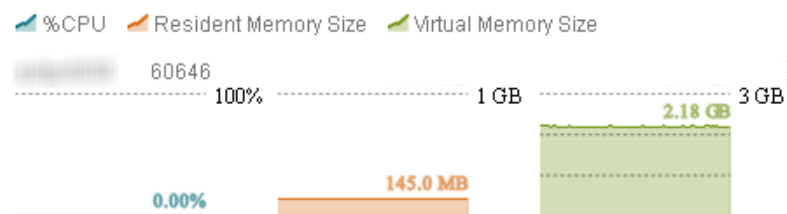




Use these histograms to note patterns of CPU usage among the CAS nodes.

- 3 Select **Add View** ⇒ **CAS Process Metrics**.

The **CAS Process Metrics** panel on the window displays a set of histograms. There is one set of three histograms for each machine and the corresponding CAS server process. If you are not an administrator, only the set of histograms for the CAS controller node is displayed.

Each set of histograms displays the percentage of CPU used, amount of resident memory used, and amount of virtual memory used for the CAS process.



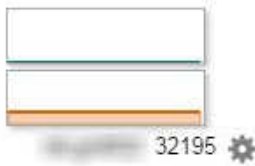
- 4 Click  if you want to stop metric collection. Click  to resume collection.

Monitor CPU Usage for a Session

1 In CAS Server Monitor, select  on the left side of the window.

2 Select **Add Session View** and select a session.

The panel for the session displays a set of histograms, with one histogram for each machine in the grid. If you are not an administrator, only the histogram for the CAS controller node is displayed. The top half of the histogram displays the percentage of CPU load used by the session, and the bottom displays the amount of resident memory used for the session.



Monitor Host Performance

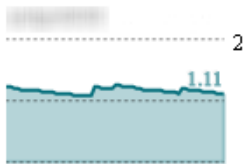
CAS Server Monitor displays histograms that enable you to view the CPU load and memory usage for all machines in your CAS server. Follow these steps:

1 In CAS Server Monitor, select  on the left side of the window.

2 To view the CPU load, select **Add View** ⇒ **Host CPU Load Average**.

The **Host CPU Load Average** panel on the window displays a set of histograms. There is one histogram for each machine in the CAS grid. If you are not an administrator, only the histogram for the CAS controller node is displayed.

Each histogram displays the CPU load on the machine, using the same format as the Linux `xload` command. Each division on the histograms represents one load average point. The highest point on each histogram is displayed to the right of the histogram.

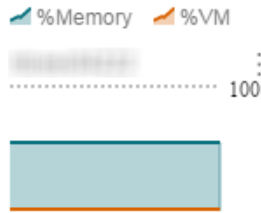


Use these histograms to note usage patterns among the CAS nodes. For example, if you notice that the load on a worker node machine is significantly and consistently higher than the load on other machines, you can use the **Show Processes** function to check for other running processes or defunct processes. See [“Monitor Process Information” in SAS Viya Administration: Monitoring](#) for instructions on this function.



3 To view the memory usage, select **Add View** ⇒ **Host Memory Usage**.

The **Host Memory Usage** panel on the window displays a set of histograms. There is one histogram for each machine in the CAS grid. If you are not an administrator, only the histogram for the CAS controller node is displayed.





Each histogram displays the percentage of memory used on the machine, from 0 to 100%. The percentage of memory used is displayed in green, at the top of the histogram. The percentage of virtual memory used is displayed in orange, at the bottom of the histogram.



Use these histograms to note patterns of memory usage among the CAS nodes. For example, if the memory usage is consistently high on a machine, its memory might need to be increased.



- 4 Click  if you want to stop metric collection. Click  to resume collection.

Monitor Process Information

- Perform one of these actions in CAS Server Monitor:
 - Select  on the left side of the window and open one of the views from the **Add View** or **Add Session View** menus. Click  to the right of a histogram. Select **Show Processes**.
 - Click  and select the **Nodes** tab. Click  on the right side of a node's row and select **Show Processes**.
- The Processes window appears. The window displays this information:
 - Metrics for the selected node, including uptime, number of processes, memory usage, CPU load, and file usage
 - A histogram of the CPU load for the node
 - A table containing the output from the `top` command for the selected node. The output includes metrics such as CPU usage, time, and threads for each process. If you are a SAS administrator, the window displays information about all processes. If you are not a SAS administrator, you can view information about only your own processes.

Change the Monitoring Display Options

When you are viewing the histograms in the **Resource Monitor** view in CAS Server Monitor, you can control how the histograms are displayed.

- To change how quickly the graph data is refreshed, move the slider next to the **Speed** label.
- To change the size of the histograms, move the slider next to the **Size** label.
- The default layout for a histogram view is a grid. To change to a single column, click the **column icon**  in the banner for a view. To return to a grid layout, click the **grid icon** .

To change the default view for the **Resource Monitor** view, select *userid* ⇌ **Settings** in the upper right of the CAS Server Monitor window. You can select a default monitor view and layout.

SAS Viya for Containers: SAS Cloud Analytic Services

Read This First

Information about SAS Cloud Analytic Services (CAS) can be found in [SAS Viya Administration: SAS Cloud Analytic Services](#). Any differences specific to SAS Viya for Containers are described in this section.

Adjust Caslib Management Privileges

Caslib management privileges are stored in a directory known as the `permstore`. For SAS Viya for Containers, the default `permstore` should be stored on persistent storage and attached to the container so that the `permstore` persists if the container is redeployed. For more information, see “[Adjust Caslib Management Privileges](#)” in [SAS Viya Administration: SAS Cloud Analytic Services](#).

CAS Configuration Options

For SAS Viya for Containers, CAS configuration options are specified as environment variables or are recorded in a file on disk that is mounted to the container.

Note: In the following examples, the CAS configuration option `MODE` is set. Because SAS Viya on Containers supports the CAS server for symmetric multi-processing (SMP) only, the `MODE` setting would always be `SMP`. The `MODE` setting is used here as an example of how a CAS configuration can be set.

To specify a CAS configuration option as an environment variable:

- Prefix the CAS configuration option with `CASCFG_` and capitalize the option name. In the following examples, the `MODE` option is prefixed with `CASCFG_` and is set to `smp`.
- For Docker environments, use the Docker run command to set the CAS configuration option. Here is an example:

```
docker run \
  --detach \
  --rm \
  --env CASCFG_MODE=smp \
  --hostname sas.viya.programming \
  --name sas-viya-programming \
  sas-viya-programming
```

- For Kubernetes environments, an addition is made to the manifest file (`sas-analytics.yml`). Here is an example:

```
- name: CASCFG_MODE
  value: "smp"
```

To specify the CAS configuration options in a file that is mounted to the container:

- 1 In the directory where you unpacked the `SAS_Viya_deployment_data.zip` file, run Docker to create a directory named `sasinside`, and then create a file in that directory named `casconfig_usermods.lua`.
- 2 Edit the `casconfig_usermods.lua` file to specify the CAS configuration options, such as `cas.mode='smp'`, and then save the file.
- 3 Perform one of the following steps:

- For Docker environments, run the container to link the location of the `casconfig_usermods.lua` file to where the software expects it.

```
docker run \
--detach \
--rm \
--volume ${PWD}/sasinside:/sasinside \
--hostname sas.viya.programming \
--name sas-viya-programming \
sas-viya-programming
```

- For Kubernetes environments, edit the `sas-analytics.yml` file to mount the volume. In the following example, an NFS server is used.

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: sas-programming
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: sas-programming
    spec:
      containers:
        - name: sas-programming
          <content removed intentionally>
          volumeMounts:
            - name: sasconfig
              mountPath: /path/to/sasinside
      volumes:
        - name: sasconfig
          nfs:
            server: sasconfig.company.com
            path: "/vol"
```

Note: To pick up configuration changes, a user must connect to a new session of CAS.

CAS Environment Options

For SAS Viya for Containers, CAS environment options are specified as environment variables or are recorded in a file on disk that is mounted to the container.

To specify a CAS environment option as an environment variable:

- Prefix the CAS environment option with `CASENV_` and capitalize the option name. In the following examples, the `ADMIN_USER` option is prefixed with `CASENV_` and is set to `sasdemo`.
- For Docker environments, use the Docker run command to set the CAS environment option. Here is an example:

```
docker run \
--detach \
--rm \
--env CASENV_ADMIN_USER=sasdemo \
--hostname sas.viya.programming \
--name sas-viya-programming \
sas-viya-programming
```

- For Kubernetes environments, an addition is made to the manifest file (sas-analytics.yml). Here is an example:

```
- name: CASENV_ADMIN_USER
  value: "sasdemo"
```

To specify the CAS environment options in a file that is mounted to the container:

- 1 In the directory where you unpacked the SAS_Viya_deployment_data.zip file, run Docker to create a **sasinside** directory, and then create a file in that directory named **casconfig_usermods.lua**.
- 2 Edit the casconfig_usermods.lua file to specify the CAS configuration options, such as *Env.ADMIN_USER='sasdemo'*, and then save the file.
- 3 Perform one of the following steps:
 - For Docker environments, run the container to link the location of the casconfig_usermods.lua file to where the software expects it.

```
docker run \
--detach \
--rm \
--volume ${PWD}/sasinside:/sasinside \
--hostname sas.viya.programming \
--name sas-viya-programming \
sas-viya-programming
```

- For Kubernetes environments, edit the sas-analytics.yml file to mount the volume. In the following example, an NFS server is used.

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: sas-programming
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: sas-programming
    spec:
      containers:
        - name: sas-programming
          <content removed intentionally>
          volumeMounts:
            - name: sasconfig
              mountPath: /path/to/sasinside
      volumes:
        - name: sasconfig
          nfs:
            server: sasconfig.company.com
            path: "/vol"
```

Note: To pick up configuration changes, a user must connect to a new session of CAS.

SAS Viya for Containers: Troubleshooting

Expired license

Explanation:

If you see the following message, your license has expired:

```
ERROR: Contact your SAS Installation Representative to obtain your updated SAS
Installation Data (SID) file, which includes SETINIT information.
```

Resolution:

See [“SAS Viya for Containers: Licensing”](#) on page 4.

File not found message displays**Explanation:**

The following error is expected and does not impact the running of the system because Transport Layer Security is not turned on:

```
ERROR: File was not found. File =/opt/sas/viya/config/etc/
SASSecurityCertificateFramework/private/cas/shared/default/encryption.key.
System Error Code = 2.
```

Resolution:

Ignore the error.

User password is not recognized for Jupyter Notebook**Explanation:**

Jupyter Notebook can be added to a Docker container. Token-based authentication can be enabled for Jupyter Notebook by setting a token for the JUPYTER_TOKEN option. During a subsequent logon, a user can continue to use the token, or a custom password can be set. If a custom password is set but the Docker container is not stopped and restarted, then the password is not recognized.

Resolution:

After a custom password is set for Jupyter Notebook, stop and restart the container.

```
docker stop sas-programming
./launchsas.sh
```