## Contents

### Chapter 1 / Getting Started
- SAS Viya on Cloud Foundry: Cheat Sheet .................................................. 1
- SAS Viya on Cloud Foundry: Scope .......................................................... 3
- SAS Viya on Cloud Foundry: Deployment Types ....................................... 3

### Chapter 2 / Bridging SAS 9 and SAS Viya
- Bridging SAS Viya and SAS 9: Overview .................................................. 5
- Bridging SAS Viya and SAS 9: How To ...................................................... 5
- Bridging SAS Viya and SAS 9: Licensing .................................................. 8
- Bridging SAS Viya and SAS 9: Concepts .................................................. 9

### Chapter 3 / Backup and Recovery
- Backup and Recovery: Overview .............................................................. 11
- Backup and Recovery: How To ............................................................... 11

### Chapter 4 / Configuration Properties
- Configuration Properties: Overview ....................................................... 15
- Configuration Properties: How To .......................................................... 15
- Configuration Properties: Reference ...................................................... 18

### Chapter 5 / Encryption
- Overview ................................................................................................... 37
- How To ...................................................................................................... 38

### Chapter 6 / Licensing
- Licensing: Overview .................................................................................. 41
- Licensing: How To .................................................................................... 41

### Chapter 7 / Monitoring
- Monitoring: How to (CAS Server Monitor) .............................................. 45

### Chapter 8 / SAS Cloud Analytic Services
- SAS Cloud Analytic Services: Overview .................................................. 49
- SAS Cloud Analytic Services: Operate ...................................................... 50
- SAS Cloud Analytic Services: How To (SAS Environment Manager) ........... 50
- SAS Cloud Analytic Services: How To (CAS Server Monitor) ...................... 51
- SAS Cloud Analytic Services: Concepts .................................................. 54
- SAS Cloud Analytic Services: Reference .................................................. 57

### Chapter 9 / Other Servers and Services
- Other Servers and Services: Overview ..................................................... 83
- Other Servers and Services: Operate ......................................................... 84
- Other Servers and Services: Add Service Instances .................................... 85

### Chapter 10 / Troubleshooting
- Troubleshooting ....................................................................................... 87
## Getting Started

### SAS Viya on Cloud Foundry: Cheat Sheet

The following table lists Cloud Foundry administrative functionality. For tasks that are the same on Cloud Foundry and Linux, the links go to the Linux documentation. For tasks that are different on Cloud Foundry and Linux, the differences are described in the table, or links are provided to the information that is specific to Cloud Foundry.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Notes, Exceptions, and References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>To determine which SAS components are managed by BOSH and which are managed by Cloud Foundry, see these diagrams.</td>
</tr>
</tbody>
</table>
| Authentication                 | LDAP connection information is in the BOSH manifest. For more information, see SAS Viya for Cloud Foundry: Deployment Guide.  
                                   For all other authentication topics, see SAS Viya Administration: Authentication.  
                                   For external credentials, Cloud Foundry supports these caslib types: Path based, DNFS, HDFS, LASR, Hadoop. For more information, see SAS Viya Administration: External Credentials. |
<p>| Backups                        | See Backup and Recovery on page 11.                                                              |
| CAS Server Monitor             | Some functionality on the CAS Server Monitor Nodes page should not be used for SAS Viya on Cloud Foundry. Do not attempt to add nodes, remove nodes, or terminate a server instance from the CAS Server Monitor (or with the addNode and removeNode CAS actions). Instead, use the appropriate BOSH command. See SAS Viya Administration: Using CAS Server Monitor. |
| Command-line interfaces        | See Command-line Interfaces.                                                                     |
| Configuration                  | For persisted changes to servers, services, and system properties, modify the manifest and redeploy. For temporary changes to services and system properties, use SAS Environment Manager. See Configuration Properties on page 15. |</p>
<table>
<thead>
<tr>
<th>Subject</th>
<th>Notes, Exceptions, and References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>See <em>SAS Viya Administration: Content Management</em> and <em>SAS Viya Administration: Promotion (Import and Export).</em></td>
</tr>
<tr>
<td>Data</td>
<td>The following paths must be accessible to all CAS users:</td>
</tr>
<tr>
<td></td>
<td>￭ CentOS: /dev/shm</td>
</tr>
<tr>
<td></td>
<td>￭ Ubuntu: /dev/shm and /run/shm</td>
</tr>
<tr>
<td></td>
<td>See Data Administration.</td>
</tr>
<tr>
<td>Encryption: Data in</td>
<td>The initial state depends on your site’s setup. Set up TLS at the load balancer, external to the SAS deployment. SAS does not install HAProxy or configure TLS internally. Because the traffic that is internal to the SAS deployment is HTTP, allow port 80 at the proxy and block port 80 at the firewall. Certificates are managed by BOSH and Cloud Foundry. Configure LDAPS connection. See <em>Configure SAS Viya to Connect to LDAPS Provider on page 38.</em> Use TLS with SAS/CONNECT. See <em>Sign On to SAS/CONNECT Using TLS on page 38.</em></td>
</tr>
<tr>
<td>Motion (TLS)</td>
<td></td>
</tr>
<tr>
<td>Encryption: Data at</td>
<td>See <em>Encryption in SAS Viya: Data at Rest.</em></td>
</tr>
<tr>
<td>Rest</td>
<td></td>
</tr>
<tr>
<td>Environment Manager</td>
<td>For Cloud Foundry installations, SAS Home does not include a SAS Environment Manager tile. You must access SAS Environment Manager by entering your equivalent of <a href="http://host/SASEnvironmentManager">http://host/SASEnvironmentManager</a> in your browser. Also for Cloud Foundry, the Dashboard and the Machines and Services views are not available. Otherwise, all functionality is as described in Using SAS Environment Manager.</td>
</tr>
<tr>
<td>Identities</td>
<td>See Identity Management.</td>
</tr>
<tr>
<td>Licenses</td>
<td>Software license information is in the BOSH manifest. To renew, paste license file contents into the manifest and redeploy. See <em>Licensing on page 41.</em></td>
</tr>
<tr>
<td>Logs</td>
<td>Access microservice logs from Cloud Foundry. Access CAS logs (and logs from other BOSH-managed virtual machines) from BOSH. Use the monit command to monitor the logs for services on stateful servers.</td>
</tr>
<tr>
<td>Mobile Devices</td>
<td>See <em>SAS Viya Administration: Mobile Devices.</em></td>
</tr>
<tr>
<td>Monitoring</td>
<td>You can use CAS Server Monitor to monitor CAS. See <em>Monitoring: How to (CAS Server Monitor) on page 45.</em></td>
</tr>
<tr>
<td>Servers and Services</td>
<td>Some functionality on the CAS Server Monitor Nodes page should not be used for SAS Viya on Cloud Foundry. Do not attempt to add nodes, remove nodes, or terminate a server instance from the CAS Server Monitor (or with the addNode and removeNode CAS actions). Instead, use the appropriate BOSH command. See <em>SAS Cloud Analytic Services on page 49</em> and <em>Other Servers and Services on page 83.</em></td>
</tr>
<tr>
<td>Themes</td>
<td>See Themes.</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>See <em>Troubleshooting on page 87.</em></td>
</tr>
<tr>
<td>Tuning</td>
<td>See <em>SAS Viya Administration: Tuning.</em></td>
</tr>
<tr>
<td>Updates</td>
<td>Software updates are handled by BOSH. All instances of a particular pet must be updated simultaneously. Microservices can be updated individually.</td>
</tr>
</tbody>
</table>
SAS Viya on Cloud Foundry: Scope

This document is intended for someone who is fluent in Cloud Foundry, familiar with your site’s Cloud Foundry implementation, and who has the necessary privileges for the Elastic Runtime to run Cloud Foundry command-line interface (cf CLI) commands.

Most operations tasks for SAS Viya on Cloud Foundry are performed using third-party tools such as BOSH commands and Cloud Foundry commands. Information about third-party tools is beyond the scope of SAS documentation.

This document provides following information:
- supporting facts that are specific to SAS Viya, including details, context, and references
- instructions for exceptions, where you can use a SAS component to perform an operations task

SAS Viya on Cloud Foundry: Deployment Types

Full Deployment

The full deployment type includes all of the software to which you are entitled. This is the default type of deployment. Contrast with programming-only deployment.
Programming-Only Deployment

The *programming-only deployment* type excludes SAS Home, most graphical user interfaces, and most services. This is the simplest and smallest type of deployment. Contrast with full deployment.

Note: If SAS Home is available (at your equivalent of http://host/SASHome), you do not have a programming-only deployment.

When `programming_only=true` is present in the sample manifest, only SAS Studio is deployed to the Cloud Foundry Elastic Runtime. Also, only BOSH virtual machines (VMs) for SAS Cloud Analytic Services (CAS) and SAS Foundation are set up. The other VMs present in a full deployment are not created: SAS Configuration Server (Consul), SAS Data Integration Server (PostgreSQL), SAS Message Broker (RabbitMQ), and so on.
Bridging SAS Viya and SAS 9: Overview

When a new release of SAS software becomes available, users typically either continue running their current version of SAS or they choose to upgrade and migrate to the new release. However, with SAS Viya, customers are expected to continue running their current version of SAS along with SAS Viya.

Bridging SAS Viya and SAS 9: How To

Start and Use SAS/CONNECT with SIGNON and RSUBMIT Statements

In Cloud Foundry environments, you can use SAS/CONNECT to connect SAS Viya to SAS 9. However, you cannot connect from SAS 9 to SAS Viya.

Use SAS/CONNECT to do the following:

- Start a connection from SAS Viya to SAS 9.
  
  Issue the SIGNON statement from a SAS/CONNECT client on SAS Viya to start a connection with the SAS/CONNECT server on SAS 9. For more information on the SIGNON statement, see the SAS/CONNECT User's Guide.

- Submit a block of statements from SAS Viya to SAS 9.
  
  Issue the RSUBMIT statement from a SAS/CONNECT client on SAS Viya to submit code to the SAS/CONNECT server on SAS 9. For more information on the RSUBMIT statement, see the SAS/CONNECT User's Guide.
Verify SAS/CONNECT between SAS Viya and SAS 9

To verify connections between SAS Viya and SAS 9, ensure the following:

- SAS/CONNECT is licensed in both environments.
- SAS/CONNECT spawner is running in the SAS 9 environment.
- The encodings of the client and server sessions are compatible. For more information, see Encoding Compatibility between SAS/CONNECT Client and Server Sessions on page 9.

For example, suppose that you have a Windows 2012 machine running SAS 9, with SAS Visual Analytics, SAS Visual Statistics, SAS Enterprise Miner, and SAS/CONNECT. The Windows machine is networked with a Cloud Foundry environment on which SAS Viya with SAS Visual Data Mining and Machine Learning are running. The following diagram shows the key elements used for testing SAS/CONNECT between SAS Viya and SAS 9. The SAS LASR Analytic Server is not used in testing. It is shown only to indicate its role as a functional predecessor for the SAS Cloud Analytics Services (CAS) server.

**Products Involved in Bridging SAS Viya and SAS 9**

To confirm SAS/CONNECT functionality between a SAS Viya session and a SAS 9 session, submit SAS/CONNECT statements from SAS Viya to SAS 9. For more information on SAS/CONNECT, see the SAS/CONNECT User’s Guide.

**Sample Code to Test SAS/CONNECT from SAS Viya to SAS 9**

Confirm that SAS/CONNECT works across environments by submitting code from SAS Viya to SAS 9.

This example uses SAS Studio in a SAS Viya environment to submit SAS/CONNECT statements to the SAS 9 environment. The SAS/CONNECT client in the SAS Viya environment submits a SIGNON statement to the SAS/CONNECT spawner in the SAS 9 environment, which, in turn, requests a connection to the SAS/CONNECT server in the SAS 9 environment. Once the SAS/CONNECT server is running, a connection is established between the machines in both environments.

The SAS Viya workspace server also allocates a CAS library in CAS. The CAS library can then download data directly from the SAS/CONNECT server in the SAS 9 environment. Code for the task to be performed is sent using RSUBMIT statements in an RSUBMIT code block.

The following diagram shows the servers involved and the initial flow of communication:
The sample code for this test is as follows:

```sas
/* Allocate CAS library named MYCAS as sasdemo */
%let caslibname = mycas;
LIBNAME &caslibname cas caslib=casuser;

/* Connect to SAS 9 server using credentials and */
/* rs-submit code so you can download the HEART data */
/* set from the remote SASHELP */
/* Change <myHost.myDomain.com> and <port> below */
/* to the host and port of the SAS/CONNECT spawner */

%let myserver=<myHost.myDomain.com> <port>;
SIGNON myserver user=sasdemo passwd="myPassword";
rsubmit;

/* Download SASHELP.HEART dataset from SAS 9 host */
/* and save the table to memory in CAS library MYCAS */
/* Then use PROC CONTENTS to view the file in memory */
PROC DOWNLOAD data=sashelp.heart out=mycas.heart94;
run;
endrsubmit;
PROC CONTENTS data=mycas.heart94;

/* Create basic statistics using PROC MDSUMMARY and */
/* then save the summary output table to CAS memory */
/* as heartsum94. */
PROC MDSUMMARY data=mycas.heart94;
GROUPBY deathcause;
```
VAR cholesterol systolic diastolic;
OUTPUT out=mycas.heartsum94;
run;

/* Use PROC DATASETS to view the saved tables residing */
/* in memory in the CAS library MYCAS. */
PROC DATASETS lib=mycas; run;
signoff;

In this test, you loaded data from the SAS 9 environment into CAS memory in the SAS Viya environment as a table. Then you performed some basic statistics on the table using a CAS based procedure, MDSUMMARY. Once the data is loaded into CAS, it is available for processing by in-memory analytics.

Note: The CAS library is local to the SAS Viya workspace server, as well as to the analytic code that is executed on the in-memory CAS tables in the SAS Viya environment. Code that is submitted to the SAS/CONNECT server in the SAS 9 environment using the RSUBMIT statement executes in the SAS 9 environment.

The output of PROC DATASETS for the CAS library MYCAS displays both the original table that was copied into CAS memory (HEART94) as well as the summary table that was generated in CAS memory using PROC MDSUMMARY.

These results indicate a successful use of SAS/CONNECT as a bridge between a SAS Viya environment and a SAS 9 environment.

---

**Bridging SAS Viya and SAS 9: Licensing**

To use SAS/CONNECT with your SAS Viya and SAS 9 environments, it must be licensed in both environments. SAS 9 customers should check with their SAS software administrator because many current SAS 9 customers already have licenses for SAS/CONNECT in their existing environments. SAS Visual Data Mining and Machine Learning in its default configuration on SAS Viya does not include licensing for the SAS/CONNECT software. For such installations, SAS/CONNECT for SAS Viya needs to be licensed separately.
Bridging SAS Viya and SAS 9: Concepts

SAS/CONNECT in Cloud Foundry Environments

You can use SAS/CONNECT to create a virtual one-way bridge between your SAS 9 environment and SAS Viya on Cloud Foundry environment. Data can flow from SAS 9 to SAS Viya but not from SAS Viya to SAS 9.

In Cloud Foundry implementations, the SAS/CONNECT server does not exist in the SAS Viya environment. The SAS/CONNECT server exists only in the SAS 9 environment. The SAS/CONNECT client does exist in the SAS Viya environment. Interactions that require SAS Viya to access a SAS/CONNECT server in SAS 9 can be completed, but interactions that require SAS 9 to access a SAS/CONNECT server in SAS Viya cannot. To move data from SAS 9 to SAS Viya, SAS Viya must pull the data from SAS 9, rather than SAS 9 pushing the data into SAS Viya. The connection enables features such as:

- transferring disk copies of data
- directly processing remote data sources and returning results locally
- running multiple independent processes asynchronously
- combining resources from multiple computers to work in parallel

While working in the SAS Viya programming environment, you can access data in your SAS 9 environment and transfer it directly into memory in SAS Viya. SAS/CONNECT supports all SAS releases, so you can move data from any SAS deployment to SAS Viya.

Encoding Compatibility between SAS/CONNECT Client and Server Sessions

To successfully use SAS/CONNECT programming services, make sure that the client and server session encodings are compatible. In the SAS Viya environment, the default session encoding is UTF-8. In the SAS 9 server environment, the default session encoding is LATIN1. Compatible encodings share a common character set. For example, client and server sessions that each use the UTF-8 encoding should be compatible with each other.

If one session’s encoding is not compatible with the other session’s encoding, then SAS issues a note stating that data might not have been sent correctly. In this example, the SAS Viya client is signing on to a SAS 9 system:

```
signon host.9650 user=&user pwd=&pwd
NOTE: Remote signon to HOST.9650 commencing (SAS Release V.03.00M0P050516).
NOTE: FIPS validated AES encryption is being used to protect network traffic.
NOTE: The client session encoding utf-8 does not match the server session encoding latin1. This may produce errors when moving some character data.
Search "SAS/CONNECT Encoding Compatibility" for details.
NOTE: Unable to open SASUSER.PROFILE. WORK.PROFILE will be opened instead.
NOTE: All profile changes will be lost at the end of the session.
NOTE: Copyright(c) xxxx SAS Institute Inc., Cary, NC 27513-2414, U.S.A.
NOTE: SAS (r) Proprietary Software Version 7 (TSmm.xxx)
Licensed to SAS Institute Inc. Host Testing, Site 00000001.
NOTE: Remote signon to HOST.9650 complete.
```

If one session is using UTF-8 and the other session has an unknown, or unsupported, encoding, an error occurs and the connection is not made.
Backup and Recovery

Backup and Recovery: Overview

Use your Cloud Foundry tools to back up and recover your environment. The following guidelines should be considered when implementing the backup and restore processes:

- Scripts that call backup processes for SAS services that are deployed in BOSH (such as Postgres, Consul, and RabbitMQ) can be automated or run in an ad hoc manner.

- You can create BOSH errand jobs that run those scripts directly, or you can integrate the backup scripts into other standard processes that provide the backup capabilities for your other BOSH and Cloud Foundry services.

- Persistent storage attached to BOSH-managed virtual machines can provide a measure of safety to ensure that data sources that are not specifically handled by the backup utilities can still be retained and recovered.

  Note: CAS access controls and caslib information are best backed up through measures that back up data on persistent disks.

Backup and Recovery: How To

Retrieve a Copy of the BOSH Deployment Manifest

You need to retrieve a copy of the running deployment manifest from the BOSH director. This is useful for backup and archival purposes.

1. From the command line, target the BOSH director that hosts your SAS Viya deployment. For example:

   bosh target 10.10.3.3

2. Run `bosh deployments` to identify the name of your current SAS Viya deployment or deployments.
3 Run BOSH download manifest `<DEPLOYMENT-NAME>` `<LOCAL-FILE-NAME>` to download and save each BOSH deployment manifest. Copy the manifest file to an archive location for backup purposes.

Repeat this step for each SAS Viya deployment that you have in the BOSH/Cloud Foundry manifest file.

Back Up the SAS Infrastructure Data Server

SAS Viya uses the PostgreSQL relational database as its SAS Infrastructure Data Server. You can use standard PostgreSQL data backup procedures to back up and restore databases that are used by SAS Viya running on Cloud Foundry. For example, you could use the pgAdmin graphical user interface for manual backups, or you can integrate the `pg_dump` utility into scheduled scripts for automatic backups.

Note: SAS Viya for Cloud Foundry ships with PostgreSQL version 9.4.9.

To back up using the `pg_dump` utility:

- Run `pg_dump` to export the SAS SharedServices database:

  ```
  $ ./pg_dump -h `<PGPOOL-IP-ADDRESS>` -p `<PORT>` -U `dbmsowner` -d `SharedServices` > `/tmp/sas-viya-postgres.sql`
  ```

When backing up the SAS Infrastructure Data Server, consider the following points:

- You should run a version of `pg_dump` on the archival machine to avoid having to use secure shell to log into the BOSH VM.
- Use `bosh vms` to view the list of deployed VMs to identify the IP addresses used for the SAS Postgres and pgpool services.
- Make sure that you use the IP address for pgpool and the appropriate port (the default is 5432).
- When prompted for a password during connection, use the `postgres.password` property that is specified in the deployment manifest.
- Archive the exported database file per your organization's standards.

Back Up the SAS Configuration Server

SAS Viya uses Consul as its SAS Configuration Server. You can use standard Consul procedures to back up and restore the key/value configuration used by SAS Viya running on Cloud Foundry. You can integrate these backup procedures into your organization's standard archival practices.

Use Consul's KV API to back up the SAS Viya key/value configuration:

```bash
```

When backing up the SAS Configuration Server, consider the following points:

- Use `bosh vms` to view the list of deployed VMs to identify the IP address that is used for Consul.
- For the X-Consul-Token value, use the value of the `consul.tokens.management` property that is specified in the deployment manifest.
- Archive the exported JSON file that contains the key/value configurations per your organization's standards.

Back Up the SAS Message Broker

SAS Viya uses RabbitMQ as its SAS Message Broker. You can use standard RabbitMQ processes to back up and restore message broker definitions used by SAS Viya running on Cloud Foundry.
Use RabbitMQ’s API to back up the server definitions, including exchanges, queues, bindings, users, virtual hosts, permissions, and parameters:

```
```

When backing up the SAS Message Broker, consider the following points:

- Use `bosh vms` to view the list of deployed VMs to identify the IP address used for RabbitMQ.
- For the `<CLIENT-USERID>` and `<CLIENT-PASSWORD>` values, use the `rabbitmq.client.userid` and `rabbitmq.client.password` property values that are specified in the deployment manifest. If no values are set for those properties, use the default values `sasclient` for the userid and `sasclientpw` for the password.
- Archive the exported JSON file that contains the server definitions per your organization’s standards.

### Restore PostgreSQL Data

The text file containing SQL that is created by `pg_dump` can be read in by the `psql` utility. To restore a backup of the SAS Infrastructure Data Server:

- Run `psql` to import the SAS SharedServices database:
  ```bash
  $ ./psql -h <PGPOOL-IP-ADDRESS> -p <PORT> -U dbmsowner -d SharedServices < sas-viya-postgres.sql
  ```

When restoring PostgreSQL Data, consider the following points:

- You should run `psql` from the archival machine to avoid having to use secure shell to log into the BOSH VM.
- Make sure that you use the IP address for pgpool and the appropriate port (the default is 5432).
- When prompted for a password during connection, use the `postgres.password` property that is specified in the deployment manifest.

### Restore the Consul Configuration

Use a POST to the Consul's KV API to restore a backup of the SAS Viya key/value configuration:

```
```

When restoring the Consul configuration, consider the following points:

- For the `X-Consul-Token` value, use the value of the `consul.tokens.management` property that is specified in the deployment manifest.
- Iterate over the set of keys in the previously exported JSON file and POST the value to the appropriate key.

### Restore the RabbitMQ Definitions

Use a POST to RabbitMQ’s API to restore a backup of the SAS Viya message broker definitions:

```
curl -X POST -F "file=@/path/to/sas-viya-rabbitmq.json" \   http://<RABITTMQ-IP-ADDRESS>:15672/api/definitions -u <CLIENT-USERID>:<CLIENT-PASSWORD>
```

When restoring the RabbitMQ definitions, consider the following points:

- Make sure you POST using the `-F` field named `file`.
- For the `CLIENT-USERID` and `CLIENT-PASSWORD` values, use the `rabbitmq.client.userid` and `rabbitmq.client.password` property values that are specified in the deployment manifest. If no values are set for those properties, use the default values `sasclient` for the userid and `sasclientpw` for the password.
are set for those properties, use the default values `sasclient` for the userid and `sasclientpw` for the password.
Configuration Properties

Configuration Properties: Overview
For persisted changes to servers, services, and system properties, modify the manifest and redeploy.
For temporary changes to services and system properties, use SAS Environment Manager.

Configuration Properties: How To
Create and Edit Configuration Instances (Persistent)
You can persist configuration changes for the SAS Viya services by editing the BOSH manifest and redeploying.
Note: Contact your BOSH administrator for the necessary permissions to access the manifest and to run BOSH.
The SAS Viya service configuration properties are specified in the BOSH manifest under the consul instance under config.application.
Here is an example:

```yaml
- name: consul
  instances: 1 # DO NOT CHANGE
  jobs:
  - {name: consul, release: sas-bshconsul-<%= package_type %>
  - {name: load_consul_data, release: sas-bshconsul-<%= package_type %>
  vm_type: <%= vm_type %>
  stemcell: <%= stemcell %>
  networks:
```
- name: <%= private_network_name %>
  default: [dns, gateway]
- name: <%= public_network_name %>
  static_ips: [ <%= consul_ip %>]
properties:
  consul:
    tokens:
      management: <%= consul_tokens_management %>
  join_hosts:
    - <%= consul_ip %>
# For information to put in here, please review the deployment guide
key_value_data:
  config:
    application:
      sas:
        exportExcelRowLimit: 200000
        url:
          httpd: <%= servicesbaseurl %>
        oauth2:
          adminId: <%= oauth_admin_id %>
          adminSecret: <%= oauth_admin_secret %>

See also:
- SAS Viya for Cloud Foundry: Deployment Guide
- Edit Configuration Instances (Temporary)
- Reference

Edit Configuration Instances (Temporary)

Note: Using SAS Environment to edit configuration instances is temporary until another BOSH redeployment. To persist changes to servers, services, and system properties, modify the manifest and redeploy. For more information, see Create and Edit Configuration Instances (Persistent).

Note: Most SAS Viya applications and servers have a corresponding service in which you set their configuration property values.

1. Sign in to SAS Environment Manager as member of the SASAdministrators group and opt into the group.

2. From the SAS Environment Manager Dashboard, in the side menu (≡), under SAS Environment Manager, select Resources → Configuration.

3. In the navigation pane, using the select control, choose All services.

4. In the navigation pane, select a service whose configuration properties you want to change.

5. Next to the configuration instance, click 📋.
In the Edit Configuration dialog box, change the value in one or more of the configuration property fields.

When you are finished, click **Save**.

On non-cloud platforms, such as native Linux, some services require that you restart them when configuration changes are made. See “What Services Must Be Restarted?” in SAS Viya Administration: Configuration Properties.

See also:
- SAS Viya for Cloud Foundry: Deployment Guide
- Create and Edit Configuration Instances (Persistent)
- Reference

**Create Configuration Instances (Temporary)**

Note: Using SAS Environment to create configuration instances is temporary until another BOSH redeployment. To persist changes to servers, services, and system properties, modify the manifest and redeploy. For more information, see Create and Edit Configuration Instances (Persistent).

In some situations, you might decide to create a configuration instance. For example, if you want to configure a logging level for a service that is not already associated with logging.level, you need to create a new configuration instance of logging.level for that service.

1. Sign in to SAS Environment Manager as member of the SAS Administrators group and opt into the group.
2. From the SAS Environment Manager Dashboard, in the side menu (≡), under SAS Environment Manager, select Resources ➔ Configuration.
3. In the navigation pane, using the select control, choose Definitions.
4. In the navigation pane, select a configuration definition from which to create a new configuration instance.
5. At the top of the content pane, click  
6. In the New Configuration dialog box, in Services, enter one or more services for which the configuration instance applies.
7. Continue entering values. When you are finished, click **Save**.

**TIP** Properties with a red asterisk (*) are required to have a value.

On non-cloud platforms, such as native Linux, some services require that you restart them when configuration changes are made. See “What Services Must Be Restarted?” in SAS Viya Administration: Configuration Properties.
Review Default Configuration Values

1. Sign in to SAS Environment Manager as member of the SASAdministrators group and opt into the group.

2. From the Dashboard, in the top left corner of the window, click 🔍 and then Resources → Configuration.

3. In the top left corner of the window, make sure that Basic services is selected.

4. In Basic services list, select a service, application, or server whose configuration instance must be created.

   **TIP** Incomplete required configuration instances are marked with a half-filled red circle.

   ![identities] ![SASLogon]

5. On the right side of the window, next to the half-filled red circle, click 🔍.

6. In the New Configuration dialog box, in Services, enter one or more services for which the configuration instance applies.

7. Continue entering values. When you are finished, click Save.

   **TIP** You are required to provide a value for properties marked with a red asterisk (*).

8. When you are finished, click Save.

9. Repeat steps 5 – 8 for every configuration instance that is incomplete.

Configuration Properties: Reference

Configuration Properties: Reference (Services)

Application Registry Service

The Application Registry service registers applications to enable integration with SAS Home and with the Application Switcher (side menu).

sas.appregistry
   The set of configuration properties for the Application Registry service.

supplementalProperties
   The set of user-added, advanced properties.

Authorization Service

The authorization service provides the general authorization system. See SAS Viya Administration: General Authorization.

sas.authorization
   The set of configuration properties for the Authorization service.
rules.executorThreads (a supplemental property.)
specifies the number of threads that are available for bulk processing of authorization rules. The default value is 20. Modify this value only if you are directed to do so by SAS Technical Support.

remote (a supplemental property.)
disables enforcement in the general authorization system, if set to false. The default value is true. An administrator might temporarily disable authorization if rules that inadvertently prevent access are introduced. Do not disable authorization while the system is available to other users.

Cache Locator Service
The Cache Locator service returns polygon information for selected identifiers from a given table.

sas.cache
The set of configuration properties for the Cache Locator service.

ackSevereAlertThreshold
The number of seconds the distributed system waits after the ack-wait-threshold for an acknowledgment from a system member before sending a severe alert. A value of zero (0) disables this feature.

ackWaitThreshold
The number of seconds a distributed message waits for an acknowledgment from a system member before sending an alert.

conserveSockets
Allows sockets to be shared by a system member's threads.

deployWorkingDir
The working directory used when deploying JAR application files to distributed system members. This directory can be local and unique to the member or can be a shared resource.

disableAutoReconnect
Disables the ability of a system member to reconnect and reinitialize after it has been forced out of the distributed system.

disableNetworkPartitionDetection
Enables the distributed system to detect and handle splits in the distributed system. Splits are typically caused by a partitioning of the network (split brain) where the distributed system is running.

groups
The list of groups that this system member belongs to. Use a comma to separate group names.

locatorDiscoveryAttempts
The number of service discovery attempts allowed before a registered cache locator is found. A value of zero (0) allows for an unlimited number of attempts.

locatorWaitTime
The number of seconds that a system member waits for a locator to join the distributed system.

logLevel
The lowest diagnostic log level (TRACE, DEBUG, INFO, WARN, ERROR, and FATAL) that is processed. Log events whose levels are below the specified value are ignored.

memberTimeout
The number of milliseconds the distributed system waits before it determines that a system member has timed out.

CAS Management Service
The CAS Management service provides access to shared data for users and applications. The service also provides information about the SAS Viya system for operations such as monitoring and auditing.
sas.casmanagement
The set of properties that are used to configure private settings for the CAS Management service.

supplementalProperties
   The set of user-added, advanced properties.

sas.casmanagement.global
The set of properties that are used to configure global settings for the CAS Management service.

   - The set of properties used by applications to access shared data and analytics, such as map data.
     application.casServer
     The name of the CAS server used for application work.
     application.caslib
     The name of the caslib used for application data.

   - The set of properties used to identify the default CAS server for users.
     default.casServer
     The name of the default CAS server for users.
     default.caslib
     The name of the default caslib for users.

   - The set of properties used by the system for data produced during normal operation, such as audit records and monitoring data.
     system.casServer
     The name of the system CAS server.
     system.caslib
     The name of the system caslib.

Collections Service
The Collections service enables access to personal and shared collections.
sas.collections
   The set of configuration properties for the Collections service.
   supplementalProperties
   The set of user-added, advanced properties.

Configuration Service
sas.configuration
   The set of configuration properties for the Configuration service.
   forceWrite.enabled
   Enables writing to the persistence layer for every operation even when that operation made no changes.
   locking.enabled
   Enables locking between multiple instances of the SAS Configuration Service. Locking must be enabled when more than one SAS Configuration Service instance is present in the deployment.
   supplementalProperties
   The set of user-added, advanced properties.
Cross Domain Proxy Service
The Cross Domain Proxy service provides access to external web resources over HTTP.

sas.crossdomainproxy
The set of configuration properties for the Cross Domain Proxy service.

allowedDomains
The list of domains (a whitelist) that the cross-domain proxy is allowed to access. The value is a Java regular expression. Use the Or character (|) to separate multiple domains (for example, https://*\..sas\.com(:\d+)?/|https?://*\..foo\..bar\..org/).

**TIP** SAS recommends that you escape dot (.) characters in regular expressions with a slash (/) (for example, *\.sas\.com). Also, add a trailing forward slash (/) with each domain (for example, *\.sas\..com/).

allowSystemDomains
Enables the list of trusted domains (if any) required by SAS. If the cross-domain proxy is denied access to one or more of these domains, certain SAS features are disabled. (This list of trusted domains is displayed in the property description in SAS Environment Manager.)

sas.crossdomainproxy.system
The set of system configuration properties for the Cross Domain Proxy service.

excludeRequestHeaders
The list of header names (a blacklist) which the cross-domain proxy excludes from requests sent to a destination URL. The value is a Java regular expression. Use the Or character (|) to separate multiple header names (for example, cookie|Authorization).

maxPooledConnectionsPerRoute
The maximum number of pooled connections per route. (This value must be a positive integer.)

maxPooledConnections
The maximum number of total pooled connections. (This value must be a positive integer.)

connectionTimeoutInMinutes
The number of minutes allowed before the connection to the HTTP client times out. A value of zero (0) specifies no time-out.

Device Management Service
The Device Management service provides the means to maintain the server's device blacklist and whitelist tables, including controlling which security model is in place. See SAS Viya Administration: Mobile Devices.

sas.devicemanagement
The set of configuration properties for the Device Management service.

offlineLimitDays
The number of days before the mobile application goes off-line.

passcodeAttempts
The number of passcode attempts before the user is locked out of the mobile application.

passcodeTimeoutMinutes
The number of minutes before the passcode expires on the mobile application.

whitelistSupportEnabled
Enables whitelist support for mobile device security on the server.
Home Service

The Home service supports the functionality of the SAS Home application.

sas.homeservice
   The set of configuration properties for the Home service.
      supplementalProperties
         The set of user-added, advanced properties.

Identities Service

The Identities service retrieves information about identities (users or groups) from your identity provider. It also enables the creation and management of custom groups. This functionality is documented in SAS Viya Administration: Identity Management. Here are the configuration properties for the Identities service:

sas.identities
   The set of properties that are used to configure global settings for the Identities service.

cache.enabled
   Enables identities information to be cached. Caching is enabled by default.

cache.providerPageLimit
   The number of identities to process in a given request when loading the cache. The default value is 1000.

cache.cacheRefreshInterval
   The refresh interval for the identities cache.

   Note: Do not set cache.cacheRefreshInterval below 20 minutes. Doing so might have a significant impact on your overall system, especially on the LDAP and SAS Infrastructure Data (PostgreSQL) servers.

   Use the following conventions to specify the unit of time for the refresh interval:

   - d - refers to days (for example, 6d).
   - h - refers to hours (for example, 6h).
   - m - refers to minutes (for example, 6m).
   - s - refers to seconds (for example, 6s).
   - ms - refers to milliseconds (for example, 6ms).

defaultProvider
   The default provider. The default value is local. (For this release, SAS recommends that you do not change this value.)

sas.identities.providers.ldap
   The set of properties that are used to configure your LDAP provider.

pagedResults
   Enables the LDAP server to use pagination when processing requests. Pagination is enabled by default.

pageSize
   The number of identity requests per page to be processed by the LDAP server (if pagination is enabled). The default value is 500.

distinguishedName
   The LDAP property that is associated with the distinguishedName field.
sas.identities.providers.ldap.connection

The set of properties that are used to configure your LDAP provider.

host
   The host name of the LDAP server to connect to.

password
   The password for logging on to the LDAP server.

pool.enabled
   Enables pooling of LDAP connections. Pooling is enabled by default.

pool.evictionTimePeriodMillis
   The number of milliseconds that the idle-object evictor thread sleeps between runs. If the value is non-positive, the idle object evictor thread does not run. The default value is 240000.

pool.idleTimeMillis
   The minimum amount of time in milliseconds that objects can sit idle in the pool before becoming eligible for eviction by the idle-object evictor, if present. The default value is 480000.

pool.maxActive
   The maximum number of active connections of a given type (either read-only or read-write) that can be allocated from the pool at the same time. The default value is 8.

pool.maxIdle
   The maximum number of active connections of a given type (either read-only or read-write) that can remain idle in the pool without extra connections being released. For no limit, specify a non-positive value. The default value is 8.

pool.whenExhaustedAction
   An integer that indicates the behavior when the pool is exhausted. Valid values are: 0 (fail), 1 (block), or 2 (grow).

pool.testOnReturn
   Enables validation of objects before they are returned to the pool. Disabled by default.

pool.maxSize
   The maximum number of active connections of all types that can be allocated from the pool at the same time. For no limit, specify a non-positive value. The default value is \(-1\).

pool.minIdle
   The minimum number of active connections of a given type (either read-only or read-write) that can remain idle in the pool without extra connections being created. To create no extra connections, specify zero. The default value is 0.

pool.testOnBorrow
   Enables validation of objects before they are borrowed from the pool. If an object fails validation, it is dropped from the pool and an attempt is made to borrow another object. This option is enabled by default.

pool.maxWait
   The maximum amount of time in milliseconds that the pool waits for a connection to be returned before throwing an exception. For no limit, specify a non-positive value.

pool.testWhileIdle
   Enables validation of objects by the idle-object evictor, if present. If an object fails validation, it is dropped from the pool. This option is enabled by default.

port
   The port for connecting to LDAP.

Note:
When connecting via LDAP, port values are set to 389. When connecting via Lightweight Directory Access Protocol over SSL (LDAPS), port values are set to 636.

**url**
The URL for connecting to LDAP.

The default is:
```
url: ldap://${sas.identities.providers.ldap.connection.host}:${sas.identities.providers.ldap.connection.port}
```

When the host and port properties have been specified, the `url` must be changed if you are connecting via the LDAPS protocol.

**userDN**
The distinguished name (DN) of the user account for logging on to the LDAP server (for example, `cn=AdminUser, dn=example, dn=com`).

---

**sas.identities.providers.ldap.group (Field Mappings)**

The set of properties that are used to configure the mapping of group-level fields in your LDAP provider to group-level fields in SAS. For each of the following SAS fields, you specify the corresponding field in your LDAP provider. The default values are valid for most implementations of Microsoft Active Directory. For other LDAP providers, you must provide different values for some fields.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default (valid for most implementations of Microsoft Active Directory)</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountId</td>
<td>The field in the LDAP provider that is used to populate the group’s ID.</td>
<td>sAMAccountName</td>
</tr>
<tr>
<td>createdDate</td>
<td>The field in the LDAP provider that is used to populate the group’s account created date.</td>
<td>whenCreated</td>
</tr>
<tr>
<td>description</td>
<td>The field in the LDAP provider that is used to populate the group’s description.</td>
<td>description</td>
</tr>
<tr>
<td>member</td>
<td>The field in the LDAP provider that is used to populate the members of the group.</td>
<td>member</td>
</tr>
<tr>
<td>memberOf</td>
<td>The field in the LDAP provider that is used to populate the groups that this group is a member of.</td>
<td>memberOf</td>
</tr>
<tr>
<td>modifiedDate</td>
<td>The field in the LDAP provider that is used to populate the group’s account last modified date.</td>
<td>whenChanged</td>
</tr>
<tr>
<td>name</td>
<td>The field in the LDAP provider that is used to populate the group’s name.</td>
<td>displayName</td>
</tr>
<tr>
<td>objectClass</td>
<td>The object class value to use when loading groups.</td>
<td>group</td>
</tr>
</tbody>
</table>

---

**sas.identities.providers.ldap.group (Additional Properties)**

The set of properties that are used to configure information for retrieving group information from your LDAP provider.

**baseDN**
The point from which the LDAP server searches for groups (for example, `ou=groups, dc=example, dn=com`).
The filter for customizing results that are returned when groups are queried (for example, (objectClass=group)).

You can create a custom filter to exclude identities whose accounts are disabled or expired, or to exclude objects that represent computer resources rather than actual groups. If you have a large number of groups, using a custom filter can improve performance and reduce memory requirements. In addition, user management tasks can be performed more efficiently if only relevant identities are listed in SAS Environment Manager.

The filter that is used to find a group account. The default filter is:

${sas.identities.providers.ldap.group.accountId}={0}.

The following properties specify the mapping of user-level fields in your LDAP provider to user-level fields in SAS. For each of the following SAS fields, you specify the corresponding field in your LDAP provider. The default values are valid for most implementations of Microsoft Active Directory. For other LDAP providers, you must provide different values for some fields.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default (valid for most implementations of Microsoft Active Directory)</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountId</td>
<td>The field in the LDAP provider that is used to populate the user's ID.</td>
<td>sAMAccountName</td>
</tr>
<tr>
<td>address.country</td>
<td>The field in the LDAP provider that is used to populate the user's country.</td>
<td>co</td>
</tr>
<tr>
<td>address.locality</td>
<td>The field in the LDAP provider that is used to populate the user's city.</td>
<td>l</td>
</tr>
<tr>
<td>address.postalCode</td>
<td>The field in the LDAP provider that is used to populate the user's postal code.</td>
<td>postalCode</td>
</tr>
<tr>
<td>address.region</td>
<td>The field in the LDAP provider that is used to populate the user's region or state.</td>
<td>region</td>
</tr>
<tr>
<td>address.street</td>
<td>The field in the LDAP provider that is used to populate the user's street address.</td>
<td>street</td>
</tr>
<tr>
<td>createdDate</td>
<td>The field in the LDAP provider that is used to populate the user's account created date.</td>
<td>whenCreated</td>
</tr>
<tr>
<td>description</td>
<td>The field in the LDAP provider that is used to populate the user's description.</td>
<td>description</td>
</tr>
<tr>
<td>emailAddress.other</td>
<td>The field in the LDAP provider that is used to populate the user's alternate email address.</td>
<td>otherMailbox</td>
</tr>
<tr>
<td>emailAddress.work</td>
<td>The field in the LDAP provider that is used to populate the user's work email address.</td>
<td>mail</td>
</tr>
<tr>
<td>memberOf</td>
<td>The field in the LDAP provider that is used to populate the groups that this user is a member of.</td>
<td>memberOf</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Default (valid for most implementations of Microsoft Active Directory)</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>modifiedDate</td>
<td>The field in the LDAP provider that is used to populate the user’s account last modified date.</td>
<td>whenChanged</td>
</tr>
<tr>
<td>name</td>
<td>The field in the LDAP provider that is used to populate the user’s name.</td>
<td>displayName</td>
</tr>
<tr>
<td>objectClass</td>
<td>The type of user objects that are being searched for.</td>
<td>organizationalPerson</td>
</tr>
<tr>
<td>phone.business</td>
<td>The field in the LDAP provider that is used to populate the user’s work phone number.</td>
<td>telephoneNumber</td>
</tr>
<tr>
<td>phone.businessFax</td>
<td>The field in the LDAP provider that is used to populate the user’s work fax number.</td>
<td>facsimileTelephoneNumber</td>
</tr>
<tr>
<td>phone.home</td>
<td>The field in the LDAP provider that is used to populate the user’s home phone number.</td>
<td>homePhone</td>
</tr>
<tr>
<td>phone.mobile</td>
<td>The field in the LDAP provider that is used to populate the user’s mobile phone number.</td>
<td>mobile</td>
</tr>
<tr>
<td>phone.pager</td>
<td>The field in the LDAP provider that is used to populate the user’s pager number.</td>
<td>pager</td>
</tr>
<tr>
<td>title</td>
<td>The field in the LDAP provider that is used to populate the user’s title.</td>
<td>title</td>
</tr>
</tbody>
</table>

**sas.identities.providers.ldap.user (Other Properties)**

The set of properties that are used to configure additional information for retrieving user information from your LDAP provider.

**baseDN**

The point from which the LDAP server searches for users.

**objectFilter**

The filter for customizing results that are returned when querying users.

You can create a custom filter to exclude identities whose accounts are disabled or expired, or to exclude objects that represent computer resources rather than actual users. If you have a large number of users, using a custom filter can improve performance and reduce memory requirements. In addition, user management tasks can be performed more efficiently if only relevant identities are listed in SAS Environment Manager.

Here is an example of a filter that excludes identities that represent computers and identities that are inactive. This filter is compatible with Microsoft Active Directory.

```
(&(objectCategory=person)(objectClass=user)(!(userAccountControl: 1.2.840.113556.1.4.803:=2)))
```

For OpenLDAP, the filter `(objectclass=person)` excludes identities that represent resources other than users.

**searchFilter**

The filter used for locating a user account in the LDAP provider so that the user can make a connection using an ID and password.
The default filter is \${sas.identities.providers.ldap.user.accountId}={0}.

Maps Service

The Maps service returns polygon information for selected identifiers from a given table.

`sas.maps`

The set of configuration properties for the Maps service.

`defaultOSMCommunicationProtocol`

The protocol (HTTP, HTTPS) used for default Open Street Map servers.

`localEsriServicesUrl`

The URL to the local Esri map services. The URL consists of a protocol, host, port, and path (for example, http://myserver:6080/arcgis/services/).

*Note:* If your on-premises Esri servers use a different network domain than your SAS Viya system, then you must add the necessary map URLs to the whitelist of domains that the cross-domain proxy is allowed to access. For more information, see `allowedDomains` on page 21.

`useArcGISOnlineMaps`

Enable access to background maps from the Esri ArcGIS Online catalog.

The set of custom configuration properties for Open Street Map server settings.

`customOSM.servers`

A comma-separated list of servers, with paths to tiles (for example, http://myhost1.myorg.com/tiles/, http://myhost2.myorg.com/tiles/).

`customOSM.maxResolution`

The maximum resolution in meters per pixel of each map tile. The value must be a decimal number.

`customOSM.numResolutions`

The number of tile levels configured on the tile servers. The value must be a positive integer.

Monitoring Service

The monitoring service provides information about the machines and services in your environment. See *SAS Viya Administration: Monitoring*.

`sas.monitoring`

The set of configuration properties for the Monitoring service.

`cacheTimeFrame`

The number of milliseconds allocated to clean up the metric cache.

`collectorInclude`

The list of metric collectors (cas, consul, postgres, springboot) that are started by the Monitoring service. Use a space to separate multiple collectors.

`collectorInterval`

The number of milliseconds that the collector waits to collect the metrics.

`resourceFreshInterval`

The number of milliseconds that the resource finder waits to re-discover a resource in the environment.

Report Data Service

The Report Data service retrieves data from reports.

`sas.reportdata`

The set of configuration properties for the Report Data service.
comparisonEpsilon
The number in E notation that is the variability allowed when comparing floating point numbers for equality.

decisionTreeResponseCardinalityLimit
The maximum cardinality of a dependent variable allowed to run a decision tree.

decisionTreePredictorCardinalityLimit
The maximum cardinality of independent variables allowed to run a decision tree.

defaultInteractiveDrillDepth
The number of interactive drill levels included in the offline data for report viewers.

maxTiesToIncludeOnRank
The maximum number of ties allowed for a rank.

defaultMaxCellsProduced
The maximum number of data cells delivered for each query result to report viewers.

enableResultCache
Enable report result caching.

exportExcelRowLimit
The maximum number of rows allowed for export files formatted for Excel.

exportExcelColumnLimit
The maximum number of columns allowed for export files formatted for Excel.

exportTSVandCSVRowLimit
The maximum number of rows allowed for tab- and comma-separated export files.

exportTSVandCSVColumnLimit
The maximum number of columns allowed for tab- and comma-separated export files.

executorExpirationIntervalMinutes
The number of minutes before inactive data executor sessions are closed.

modelingClassCardinalityLimit
The maximum number of class values allowed to run on fit models.

modelingGroupByCardinalityLimit
The maximum number of group by values allowed to run on fit models.

resultCacheErrorExpirationSeconds
The number of seconds before the error cases for a report result are removed from the cache.

resultCacheTimeToLiveSeconds
The number of seconds before a report result is removed from the cache.

socketTimeoutSubscribeMillis
The number of milliseconds allowed for executing subscribe data queries.

socketTimeoutLiveCancellableMillis
The number of milliseconds allowed for executing live, cancelable data queries.

socketTimeoutLiveNonCancellableMillis
The number of milliseconds allowed for executing live, non-cancelable data queries.

tempCacheTimeToIdleSeconds
The number of seconds allowed for a client to retrieve temporary files of result data before they are removed from the cache.

xmlParserPoolSize
The number of XML parsers to be instantiated during application start-up.

supplementalProperties
The set of user-added, advanced properties.
Report Package Service

The Report Package service executes reports to generate corresponding “report packages.” The report package includes the report.xml, CSS style sheets, images, CSV data files, and so on, that are required to render the report.

sas.reportpackages
The set of configuration properties for the Report Package service.

backgroundThreadMonitorSecs
The frequency in seconds that the background thread monitor runs. A value of zero indicates that the monitor is disabled.

imageDefaultMaxBytes
The maximum number of bytes for an image. Larger images are scaled down, unless ‘noscale’ is specified in the report.

subscribeConcurrentRequestLimit
The maximum number of reports packages that can be generated concurrently per user.

useProxyServiceForExternalImages
The Cross Domain Proxy service used to retrieve the external images in the report.

xmlParserPoolSize
The number of XML parsers to be instantiated during application start-up.

packageExpirationTime
The number of seconds allowed before the report package is deleted from the cache.

supplementalProperties
The set of user-added, advanced properties.

Report Renderer Service

The Report Renderer service creates PDF documents from report packages.

sas.reportrenderer
The set of configuration properties for the Report Renderer service.

directory
The directory where reports are copied when running in debug mode.

cacheDuration
The number of seconds allowed before rendered reports are deleted from the cache.

timeoutMillis
The number of milliseconds allowed before the rendering process times out.

workingDirectory
The working directory used for building rendered reports.

supplementalProperties
The set of user-added, advanced properties.

Configuration Properties: Reference (Applications)

SAS Home

Here are the configuration properties for SAS Home:
sas.home
The set of configuration properties for SAS Home.
supplementalProperties
The set of user-added, advanced properties.

SAS Logon Manager
SAS Logon Manager provides OAuth2 and OpenID Connect services for authentication, and a user interface for sign in, sign out, and other related functions. See “Authentication: Overview” in SAS Viya Administration: Authentication. Here are the configuration properties for SAS Logon Manager:

cors
The set of properties that are used to configure Cross-Origin Resource Sharing (CORS) security for SAS Logon Manager (SASLogon) only. (Use sas.commons.web.security.cors to configure CORS for other services.)

Note: Modifying one of these property values requires you to restart one or more SAS Viya services. For more information, see “General Servers and Services: Operate” in SAS Viya Administration: General Servers and Services.

default.allowed.uris
The comma-separated list of URIs that are allowed by default to be called from another origin. The value can contain regular expressions.
default.allowed.origins
The comma-separated list of origins that are allowed by default. The list can contain regular expressions.
default.allowed.headers
The comma-separated list of HTTP headers that are allowed by default in cross-origin requests.
default.allowed.methods
The comma-separated list of HTTP methods that are allowed by default in cross-origin requests.
default.allowed.credentials
Require that cross-origin requests must be made using credentials.
default.max_age
The maximum number of seconds that the response to the preflight request can be cached without sending another preflight request.

xhr.allowed.uris
The comma-separated list of URIs allowed in XMLHttpRequest (XHR) requests called from another origin. The value can contain regular expressions.

taxr.allowed.origins
The comma-separated list of origins allowed in XHR requests. The value can contain regular expressions.

taxr.allowed.headers
The comma-separated list of HTTP headers allowed in XHR cross-origin requests.

taxr.allowed.methods
The comma-separated list of HTTP methods allowed in XHR cross-origin requests.

taxr.allowed.credentials
Require that XHR cross-origin requests must be made using credentials.

taxr.max_age
The maximum number of seconds that the response to the XHR preflight request can be cached without sending another preflight request.
sas.logon.callback
The set of properties that are used to configure the whitelist of URIs for trusted applications.

allowed.uris
The comma-delimited list of URIs that users can be redirected to after signing in following a time-out or logoff.

sas.logon.custom
The set of properties that are used to provide custom content that is included on the Sign In to SAS page.

login
The URI of the custom content included on the Sign In to SAS page.

logout
The URI of the custom content included on the Sign In to SAS page when users sign out of the system.

timedout
The URI of the custom content included on the time-out page.

sas.logon.groups
The set of properties that are used to customize lookup of group authorities.

recursion.enabled
Allow recursive lookups of authorities for groups assigned to users.

requiresRecursion
The comma-separated list of groups that require a recursive lookup to determine externally assigned authorities.

sas.logon.guest
The set of properties that are used to configure guest access to the system.

Note: Some SAS products, such as SAS Visual Analytics, do not support guest access.

enabled
Enable anonymous guest access. (Default is false.)

username
The user name of the guest user account. (Default is webanon.)

password
The password for the guest user account.

email
The email address of the guest user.

allowedUris
The comma-separated list of URIs that allow guest access.

disallowedUris
The comma-separated list of URIs that disallow guest access.

groups
The comma-separated list of groups in which guest users are automatic, implicit members.

sas.logon.headers
The set of properties that are used to configure HTTP response headers.
content-security-policy
   The string used in the Content-Security-Policy header.

sas.logon.initial
The set of properties that are used to initially configure the system.
Note: Modifying one of these property values requires you to restart one or more SAS Viya services. For more information, see “General Servers and Services: Operate” in SAS Viya Administration: General Servers and Services.

user
   The user name for the initial user account.

password
   The password for the initial user account.

passwordResetLifetime
   The number of milliseconds for which the password reset code is valid after restart.

redirectUri
   The URI to which the initial user should be redirected after resetting the password.

sas.logon.kerberos
The set of properties that are used to enable sign-ins using Integrated Windows Authentication (IWA).
Note: Modifying one of these property values requires you to restart one or more SAS Viya services. For more information, see “General Servers and Services: Operate” in SAS Viya Administration: General Servers and Services.

servicePrincipal
   The name of the service principal in the keytab.

keyTabLocation
   The URL of the keytab file (for example, file:///opt/sas/viya/conf/etc/my_keytab).

stripRealmForGss
   Removes the @... from the end of the user name.

debug
   Enables the debug mode of the JAAS Kerberos login module.

sas.logon.oauth.providers.default
The set of OAUTH provider properties that are used to enable sign-ins using an external provider. Modifying one of these property values requires you to restart SAS Logon Manager. For more information, see SAS Viya Administration in SAS Help Center.

authUrl
   The URL to the authorization endpoint.

tokenUrl
   The URL to the token endpoint.

tokenKey
   The HMAC key or RSA public key used to sign tokens.

tokenKeyUrl
   The URL to obtain the token key.

emailDomain
   The email address domain of users that can sign on with this provider.
issuer
   The principal that issued the token, as a case-sensitive string or URI.

linkText
   The text that should be displayed on the sign-in page for this provider.

relyingPartyId
   The client ID registered in the provider.

relyingPartySecret
   The secret registered in the provider for the client ID.

scopes
   The comma-delimited list of scopes for the authorization request.

addShadowUserOnLogin
   Adds a local shadow user upon successful authentication.

showLinkText
   Shows the link text on the sign-in page.

type
   Either 'oidc1.0' or 'oauth2.0'.

attributeMappings.user_name
   The attribute claim to use as the user name.

sas.logon.oauth2.jwt
   The set of properties that are used to secure JSON web tokens with RSA digital signatures or hashed message authentication codes (HMACs).

signingKey
   Either a Base64-encoded RSA private key that is used to digitally sign tokens, or a simple passphrase for HMACs.

certificateChain
   Note: This property is not used in SAS Viya 3.2.
   The X.509 public key certificate or certificate chain in Base64-encoded format, corresponding to the RSA keys that are used to digitally sign tokens. Leave blank if using HMACs.

certificateThumbprint
   Note: This property is not used in SAS Viya 3.2.
   The SHA-1 thumbprint of the X.509 certificate corresponding to the RSA keys used to digitally sign tokens. Thumbprints must be Base64, URL-encoded. Leave blank if using HMACs.

sas.logon.oauth2.token
   The set of properties that are used to enable sign-ins using OAuth 2.0 tokens.

assumableScopes
   The groups that have an elevated level of access that users should be prompted to assume when signing into their session. A user must already be a member of one or more of these groups to receive a prompt.

sas.logon.saml
   The set of Security Assertion Markup Language (SAML) properties that are used to enable sign-ins using an external identity provider.

Note: Modifying one of these property values requires you to restart one or more SAS Viya services. For more information, see “General Servers and Services: Operate” in SAS Viya Administration: General Servers and Services.
entityBaseURL
The URL of the application where SAML assertions are accepted, (for example: https://example.com/SASLogon).

setProxyParams
Allows the base URL to reside behind an HTTP proxy.

entityID
The entity ID of the service provider.

serviceProviderKey
The PEM-encoded, RSA private key that is used by the service provider.

serviceProviderKeyPassword
The password for the private key.

serviceProviderCertificate
The PEM-encoded, X.509 certificate that is used by the service provider.

wantAssertionSigned
Specifies that the assertions must be signed.

signMetaData
Specifies that the local service provider should sign metadata.

signRequest
Specifies that the local service provider should sign SAML requests.

socket.connectionManagerTimeout
The number of milliseconds before the connection pooling times out for HTTP requests for SAML metadata.

socket.soTimeout
The number of milliseconds before the read times out for HTTP requests for SAML metadata.

sas.logon.saml.providers.default
The set of Security Assertion Markup Language (SAML) identity provider properties that are used to enable sign-ins using an external provider.

idpMetadata
The identity provider metadata or the URL to the metadata.

metadataTrustCheck
Specifies that the identity provider certificate must be trusted.

assertionConsumerIndex
The index of the assertion consumer service to use from identity provider metadata. The value must be a positive integer.

nameID
The default name ID format.

linkText
The hyperlink to display on the sign-in page.

showSamlLoginLink
Displays a link to the identity provider on the sign-in page.

sas.logon.sessions
The set of properties that are used to configure how concurrent sessions are handled.

maxConcurrentSessions
The maximum number of allowed concurrent sessions. A value of -1 allows an unlimited number of sessions.
rejectNewSessionsIfMaxExceeded
   Rejects new sessions if the maximum number of sessions is exceeded. If false, when the maximum number
   of sessions is reached, an existing session is invalidated to allow a new one to be created.

Configuration Properties: Reference (System)

Security
   The following are properties to configure security.

   sas.commons.web.security
   The set of properties that are used to configure web security.

   content-security-policy-enabled
      Sends the Content-Security-Policy header in HTTP responses to prevent injection attacks.

   content-security-policy
      The string used for the Content-Security-Policy HTTP header.

   x-content-type-options-enabled
      Sends the X-Content-Type-Options header in HTTP responses.

   x-content-type-options
      The custom value for the X-Content-Type-Options header.

   x-frame-options
      The string used for the X-Frame-Options HTTP header. Entering 'disable' prevents the header from being
      returned.

   x-xss-protection-enabled
      Sends the X-XSS-Protection header in HTTP responses.

   x-xss-protection
      The string used for the X-XSS-Protection header.

   sas.commons.web.security.cors
   The set of properties that are used to configure Cross-Origin Resource Sharing (CORS) security. (Use cors
to configure CORS for SAS Logon Manager, SASLogon.)

   allowedOrigins
      The comma-separated list of origins that are allowed by default. The list can contain regular expressions.

   allowedHeaders
      The comma-separated list of HTTP headers that are allowed by default in cross-origin requests.

   allowedMethods
      The comma-separated list of HTTP methods that are allowed by default in cross-origin requests.

   allowCredentials
      Allows credentials to be used in cross-origin requests.

Spring Boot Services
   Here is the list of third-party, Spring Boot services that you can configure. For a list of the valid property names
   and descriptions, see https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-
   properties.html.

Endpoints
   The set of properties that are used to configure Spring Actuator endpoints.
Flyway
   The set of properties that are used to configure Spring Flyway integration.

Liquibase
   The set of properties that are used to configure Spring Liquibase integration.

Logging
   The set of properties that are used to configure logging.

Logging.Level
   The set of properties that are used to configure logging levels.

Management
   The set of properties that are used to configure Spring application management.

Multipart
   The set of properties that are used to configure Spring multipart handling.

Security
   The set of properties that are used to configure Spring security.

Server
   The set of properties that are used to configure the embedded Spring server.

Shell
   The set of properties that are used to configure the Spring remote shell.

Spring
   The set of properties that are used to configure other Spring features.
Encryption

Overview

Encryption Coverage

SAS provides encryption in two contexts for Cloud Foundry:

- Data in motion is data that is being transmitted to another location. Data is most vulnerable while in transit. Sensitive data in transit should be encrypted. You can protect all traffic in transit between servers and clients. This document covers encrypting data in motion.

  Note: Certificates are managed using BOSH and Cloud Foundry.

- Data at rest is data stored in databases, file servers, endpoint devices, and various storage networks. This data can be on-premises, virtual, or in the cloud. This data is usually protected in conventional ways by firewalls. Numerous layers of defense are needed, and encrypting sensitive data is another layer. See Encryption in SAS Viya: Data at Rest.

When you deploy SAS Viya 3.2 on Cloud Foundry, the following encryption information applies to the installation:

- Cloud Foundry environments must use the BOSH-managed system store of certificates used for TLS. If you are using the BOSH trusted certificates, any virtual machine started by BOSH will have the certificates installed in the system stores. Contact your security administrator to get the certificate information that you need.

  Note: If you have your own CA certificates or are using self-signed certificates, you must deploy them to the BOSH virtual machines.

- SAS does not install HAProxy or configure TLS internally. You can set up TLS for Cloud Foundry at the load balancer, external to the SAS deployment.

- You can configure LDAPS. See Configure SAS Viya to Connect to LDAPS Provider on page 38.

- You can sign on to SAS/CONNECT using TLS. See Sign On to SAS/CONNECT Using TLS on page 38.

  Note: All discussion of TLS is also applicable to the predecessor protocol, Secure Sockets Layer (SSL).
How To

Sign On to SAS/CONNECT Using TLS

Use SAS/CONNECT as a bridge to access data across environments. In Cloud Foundry environments, you can use SAS/CONNECT to connect SAS Viya to SAS 9. However, you cannot connect from SAS 9 to SAS Viya.

Note: SAS/CONNECT spawners do not run in Cloud Foundry. From the Cloud Foundry environment, a Viya SAS/CONNECT client can use the SIGNON statement to sign on to other SAS/CONNECT spawners to create SAS/CONNECT servers in the SAS 9 environment. They can also sign on using the SASCMD SIGNON commands for same-machine SAS/CONNECT servers.

For more information, see Bridging SAS 9 and SAS Viya on page 5.

You can use TLS to secure the SAS/CONNECT bridge between SAS Viya and SAS 9. You need to use the TLS/SSL system options and environment variables when you sign on to SAS/CONNECT. These options are described in Encryption in SAS 9.4 and Encryption in SAS Viya 3.2: Data in Motion.

Cloud Foundry environments must use the BOSH-managed system store of certificates used for TLS. If you are using the BOSH trusted certificates, any virtual machine started by BOSH will have the certificates installed in the system stores. Contact your security administrator to get the certificate information that you need.

Note: If you have your own CA certificates or are using self-signed certificates, you must deploy them to the BOSH virtual machines.

From Cloud Foundry, a SAS Viya client can sign on to a SAS/CONNECT spawner and secure that connection using TLS as follows:

```plaintext
options netencryptalgorithm=SSL;
%let myserver=<myHost.myDomain.com> <port>;
SIGNON myserver user=sasdemo passwd="password";
```

Configure SAS Viya to Connect to LDAPS Provider

Lightweight Directory Access Protocol (LDAP) connections can be established in a TLS session so that all data that is sent between the LDAP client and LDAP server is encrypted. LDAP over TLS is known as LDAPS.

To connect to an LDAPS provider, SAS Viya needs access to the CA certificate used by the LDAPS provider. Certificates are managed using BOSH for Cloud Foundry. If you are using the BOSH trusted certificates, any virtual machine started by BOSH will have the certificates installed in the system stores. Contact your security administrator to get the certificate information that you need.

There are two ways to add certificates to a Java application in Cloud Foundry.

Note: These instructions assume that you have knowledge of how to use Cloud Foundry Java buildpacks. For more information, see Java Buildpack.

- Use the trusted certificates provided by BOSH. These certificates are enabled by default in more recent buildpack releases. The results are the promotion of the OS-level trusted certificates provided by BOSH into the application container and into the Java truststore. For more information, see Java Custom Buildpacks and BOSH: Configuring Trusted Certificates.
- Create a custom buildpack that includes the certificates in the Java truststore. You need to have deployed your elastic run-time environment with BOSH trusted certificates to be able to promote the certificates. Refer to BOSH Custom Trusted Certificate Support.

Configure LDAPS using the following steps.
Note: Only LDAP identity providers are supported. These instructions assume that you have basic familiarity with LDAP administration.

1 Log on to your machine as a user with root, SAS Admin, or sudo privileges.

2 If the Certificate Authority (CA) certificate that you are using is already in the default Java truststore (referenced in the default Java buildpack), perform the following tasks:

   - Verify that the default Java buildpack is listed first in the BOSH manifest file. Use the following command to list the buildpacks:
     
     ```
     cf buildpacks
     ```

   - Add the port (by default LDAPS is 636) and protocol (LDAPS) to the BOSH manifest file or edit the identities settings using SAS Environment Manager. You can also use the port value 3269 (Global Catalog) for LDAPS.

     Set the `sas.identities.providers.ldap.connection` variables. For an example, see Consul Settings.

     Edit the identities settings using SAS Environment Manager. See Step 3.

     If the CA certificate being used is a site-signed certificate, add that site-signed certificate to a custom Java buildpack.

     - Make sure that your custom Java buildpack is either listed first or that the custom buildpack is being referenced in the Diego cell section of the BOSH manifest file.

     Use the following command to list the buildpacks:

     ```
     cf buildpacks
     ```

     Use the following command to see whether the buildpack is being referenced from the Diego cell section of the manifest file:

     ```
     cf runtime diego cell
     ```

     - Add the port (by default LDAPS is 636) and protocol (LDAPS) to the BOSH manifest file or edit the identities settings using SAS Environment Manager. You can also use the port value 3269 (Global Catalog) for LDAPS.

     Set the `sas.identities.providers.ldap.connection` variables. For an example, see Consul Settings.

     Edit the identities settings using SAS Environment Manager. See Step 3.

3 Use SAS Environment Manager to set the configuration property `sas.identities.providers.ldap.connection`. Specify an LDAPS port number (by default LDAPS is 636) and specify LDAPS in the `url` field. You can also use the port value 3269 (Global Catalog) for LDAPS.

   a If the Configuration page of SAS Environment Manager is not already displayed, select Resources ➔ Configuration from the side menu.

   b Select Basic Services from the list, and then select the Identities service from the list of services.

   c In the `sas.identities.providers.ldap.connection` section, click . In the Edit `sas.identities.providers.ldap.connection` Configuration window, do the following:

      i Update values for the port field, adding an LDAPS port value (636 or 3269). Update the url field to specify LDAPS. For the remaining fields, review the default values and make changes as necessary. The default values are appropriate for most sites.

      ii Click Save.

For additional configuration instructions, see Configure the Connection to Your Identity Provider. For details about the `sas.identities.providers.ldap.connection` property, see Configuration Properties: Reference (Services) on page 18.
If needed, restart the SAS Logon Manager service by running the following command:

```
sudo service sas-viya-saslogon-default restart
```

**Note:** It might take several minutes to restart SAS Logon Manager.

If needed, restart the Identifies service.

```
sudo service sas-viya-identities-default restart
```
Licensing

Licensing: Overview

SAS Viya uses a single licensing file. Both SAS Cloud Analytic Services (CAS) and SAS Foundation use the same license.

During installation, a license is applied to both the CAS in-memory compute engine and the SAS Foundation compute engine. You apply a new license to enable new products or to extend expiration dates on existing products.

The contents of the license file resides in the BOSH manifest.

Licensing: How To

View SAS Foundation License Information

1 Open a web browser and sign in to SAS Studio with administrator privileges.
   
   Here is an example:
   
   https://mysasserver.example.com/SASStudio

2 In the Code tab, enter the following command:
   
   proc setinit; run;

3 Click .

   You should see output similar to the following:
proc setinit;
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

View SAS Cloud Analytic Services License Information

1. Open a web browser and sign in to CAS Server Monitor with CAS Administrator privileges.

   Here is an example:
   https://my_controller.example.com:8877

2. Sign in to CAS Server Monitor with a user ID that has CAS Administrator privileges.

3. In to CAS Server Monitor, beneath the Cloud Analytic Services banner, click

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

4. On the System State page, select Controller.

   You should see output similar to the following:

   ![System State Screenshot]

   Note: The license file path depicted in the figure above is for SAS Viya running on Linux.

Locate My New License File

Your new license file resides in the same directory where you have saved and uncompressed the .tgz file, sent to you by SAS.

The license file is named: SASViyaVrelease-number_order-number_Linux_x86-64.txt.

Here is an example: SASViyaV0301_09JNF2_Linux_x86-64.txt.
Apply New Licenses

Use BOSH to apply a new SAS license when your current license has expired, or when you are adding new SAS products to your deployment. BOSH applies your new license to the CAS controller and also to SAS Foundation.

1. Move the current license file into a backup location.

2. Using a text editor, open your new SAS license file, and copy its contents into the paste buffer.
   
   For information about where to locate your new license file and how to identify it, see Locate My New License File on page 42.

3. Open your BOSH manifest file, and locate the line that contains `setinit:`.

   ```
   setinit:
   text: |
   TODO - Copy the contents of the license in the Software Order Email to here.
   Make sure the contents of the file are lined up under the 'x' in 'text' above.
   ```

4. In the manifest, overwrite the lines starting with `TODO` with the license content that you copied in Step 2.

   **Note:** Make sure that the license contents begins under the `x` in the word `text` as shown in the following example.

   ```
   setinit:
   text: |
   PROC SETINIT RELEASE='V03';
   SITEINFO NAME='My Site'
   SITE=12345678 OSNAME='LIN X64' RECREATE WARN=55 GRACE=45
   BIRTHDAY='23JUN2017'D EXPIRE='30SEP2017'D PASSWORD=111111111;
   CPU MODEL=' ' MODNUM=' ' SERIAL=' ' NAME=CPU000;
   CPU MODEL=' ' MODNUM=' ' SERIAL='+9999' NAME=CPU001;
   CPU MODEL=' ' MODNUM=' ' SERIAL='+9999' NAME=CPU002;
   EXPIRE 'PRODNUM000' '18AUG2017'D / CPU=CPU000 CPU001 CPU002;
   ```

5. Set the BOSH deployment to reference SAS-Viya-manifest.yml.

   ```
   bosh deployment SAS-Viya-manifest.yml
   ```

6. Deploy the software.

   ```
   bosh deploy
   ```

7. Verify that your SAS Foundation license has been renewed by following the steps in View SAS Foundation License Information on page 41.

8. Verify that your SAS Cloud Analytic Services license has been renewed by following the steps in View SAS Cloud Analytic Services License Information on page 42.
Monitoring

Access CAS Server Monitor
To log on to CAS Server Monitor, open a web browser and enter the URL http://controller-machine:8777 in the address field.

You must have an active CAS Server session in order to access CAS Server Monitor.
For more information, see “Access the Monitor” in SAS Viya Administration: Using CAS Server Monitor.

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%.

<table>
<thead>
<tr>
<th>qstgrd008</th>
<th>60646</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>6.95%</td>
</tr>
</tbody>
</table>
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**

1 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**. This option is available only if you are an administrator.

- Click ■ and select the **Nodes** tab. Click •• on the right side of a node’s row and select **Show Processes**.

2 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 the process owner, the window displays information about all processes. If you are not the process owner, you can view information about your own processes.
Change the Monitoring Display Options

When you are viewing the histograms in the Grid 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 Grid 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 Cloud Analytic Services: Overview

SAS Cloud Analytic Services (CAS) is a server that provides the cloud-based, run-time environment for data management and analytics with SAS. CAS uses a combination of hardware and software where data management and analytics take place on either a single machine or as a distributed server across multiple machines.
SAS Cloud Analytic Services: Operate

Like the other SAS Viya servers, SAS Cloud Analytic Services (CAS) runs in a BOSH virtual machine and is operated and managed through BOSH in a typical manner.

For more information, contact your site’s Cloud Foundry administrator.

SAS Cloud Analytic Services: How To (SAS Environment Manager)

View and Modify Server Properties

To change black and whitelist settings and adjust membership in the superuser role, follow these steps:

Note: The CAS server properties described in this topic are stored in the CAS permstore file. For SAS Viya on Cloud Foundry, the default permstore is stored on the BOSH persistent disk. Therefore, the permstore persists when redeploying, as long as the deployment is not deleted before being redeployed with BOSH.

1. Sign in to SAS Environment Manager as member of the SASAdministrators group and opt into the group.
2. From the SAS Environment Manager Dashboard, in the side menu ( ), under SAS Environment Manager, select Data.
3. In View, select Servers.
4. Right-click the CAS server whose properties you want to access, and select Assume the Superuser role.
5. Right-click the CAS server whose properties you want to access, and select Properties.
6. In Paths List, click to select to enforce a blacklist, whitelist, or neither list.

   If you select the blacklist or whitelist, you can populate the list with paths.

   Note: By default, the SAS Viya install and various configuration directories are on the blacklist.

7. To save any changes, click Save. Otherwise, click Cancel.
8. In Superuser Role Membership, click to grant users the superuser role.
9. To save any changes, click OK. Otherwise, click Cancel.
10. When you are finished, click Close.
11. Click Relinquish at the bottom right of the window to relinquish the Superuser role.

Adjust Caslib Management Privileges

To adjust caslib management privileges for a particular CAS server in SAS Environment Manager, follow these steps:
Note: Caslib management privileges are stored in the CAS permstore file. For SAS Viya on Cloud Foundry, the default permstore is stored on the BOSH persistent disk. Therefore, the permstore persists when redeploying, as long as the deployment is not deleted before being redeployed with BOSH.

1. Sign in to SAS Environment Manager as member of the SAS Administrators group and opt into the group.

2. From the SAS Environment Manager Dashboard, in the side menu (≡), under SAS Environment Manager, select Data.

3. In View, select Servers.

4. Right-click the CAS server caslib management privileges that you want to adjust, and select Assume the Superuser role.

5. Right-click the CAS server whose caslib management privileges that you want to adjust, and select Properties.


7. For the identities listed, choose to enable (or disable) the ability to add and delete session and global caslibs.

   Regardless of access controls, the Superuser can add and manage all caslibs.

   Note: This display shows directly granted privileges. Indirectly granted privileges and denials of privileges are not reflected in this display.

8. To save any changes, click Save. Otherwise, click Cancel.

9. When you are finished, click Close.

10. Click Relinquish at the bottom right of the window to relinquish the Superuser role.


---

**SAS Cloud Analytic Services: How To (CAS Server Monitor)**

**View CAS Controller and System Information**

1. Sign in to CAS Server Monitor with a valid user ID and password.

2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click 🔍.

3. On the System State page, make sure that Controller is selected.

**View CAS Server Configuration**

To use CAS Server Monitor to view the current list of CAS Server options and their values, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.

2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click 🔍.

3. On the Configuration page, make sure that CAS Configuration is selected.
View CAS Start-up Options and Environment Variables

You can use CAS Server Monitor to view the option used when a CAS server was started and to see the current list of CAS environment variables and their values.

To view CAS start-up options and environment variable values, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click.

View Information about CAS Nodes

CAUTION! Some functionality on the CAS Server Monitor Nodes page should not be used for SAS Viya on Cloud Foundry. Do not attempt to add nodes, remove nodes, or terminate a server instance from the CAS Server Monitor (or with the addNode and removeNode CAS actions). Instead, use the appropriate BOSH command.

1. Sign in to CAS Server Monitor.
2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click.
4. In the Nodes table, you can view information about all the nodes in your analytics cluster.
5. To view information about processes running on a particular node, do the following:
   - Next to the node that you want to view process information about, click and select Show Processes.

View User Session Information

You can use CAS Server Monitor to view information about a user’s session, such as connection port, length of connection time, and so on.

To view user session information, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click.

Cancel CAS User Session

To cancel your CAS server session, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click.
4. At the end of the row for the session that you want to cancel, click and select Cancel Session.
**Terminate CAS User Session**

**TIP** Terminate a session only after having tried canceling a session. Using terminate might not release resources (for example, mapped memory and memory involving database connections, and so on).

To terminate your CAS server session, follow these steps:

1. Sign in to CAS Server Monitor with a valid user ID and password.
2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click.
4. At the end of the row for the session that you want to terminate, click and select Terminate Session.

**Adjust Caslib Management Privileges**

To enable non-administrators to add global caslibs, follow these steps:

**Note:** Caslib management privileges are stored in the CAS permstore file. For SAS Viya on Cloud Foundry, the default permstore is stored on the BOSH persistent disk. Therefore, the permstore persists when redeploying, as long as the deployment is not deleted before being redeployed with BOSH.

1. Sign in to CAS Server Monitor with a valid user ID and password that has administrator privileges.
2. In CAS Server Monitor, beneath the Cloud Analytic Services banner, click.
3. On the Configuration page, select Access Controls.
4. In the Caslibs list, select Global Caslib Creation.

**TIP** If the Global Caslib Creation caslib is not listed, you are not signed in as an administrator.

5. In the upper right, click Edit.
6. In the Edit Access Controls window, adjust values as needed.

<table>
<thead>
<tr>
<th>Intent</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable all users to add global caslibs.</td>
<td>In the existing row for Authenticated Users, select the Grant radio button.</td>
</tr>
<tr>
<td>Enable a group to add global caslibs.</td>
<td>Click Add Row. Select Group, enter the group name, and select the Grant radio button.</td>
</tr>
<tr>
<td>Enable an individual user to add global caslibs.</td>
<td>Click Add Row. Select User, enter the user name, and select the Grant radio button.</td>
</tr>
</tbody>
</table>

7. Click OK to save your changes.
8. Under Access Controls, review the results of your changes.
9. Verify that users who should be able to add global caslibs can do so.
Here are details:

- User and group names that you enter are not validated.
- Regardless of access controls, administrators can add and manage all caslibs.
- For the special caslibs (Global Caslib Creation and Session Caslib Creation), the only available value in the Activity column is Manage Access. The special caslibs are protected by role requirements, not by the ManageAccess permission. Granting or denying the ManageAccess permission on the special caslibs affects only the ability of non-administrators to manage other caslibs.
- If you want to restrict the ability to manage session caslibs, select Session Caslib Creation in the Caslibs list. Add direct denials as needed.

**SAS Cloud Analytic Services: Concepts**

**CAS Controller**

Controller is one of two roles that can be assigned to a host for SAS Cloud Analytic Services. For both server architectures, distributed and single-machine, one machine is assigned the controller role. When the server starts, the controller process is started. This process is sometimes referred to as the server controller. The controller accepts a connection from a client.

**Single-machine CAS Server**

The single-machine architecture uses symmetric multiprocessing (SMP). The functionality for a single-machine server is nearly identical to MPP, except that there is no cluster communication. In this architecture, the server acts as a controller. Before a client connects, the server listens on a port for connections.

After a client connects, a session is created and the session connects back to the client. (This is identical to the method that is performed by a CAS server that uses MPP.)

*Single-machine CAS Server*
CAS Workers

When a server is running in massively parallel processing (MPP) mode, in addition to a controller, the server also has multiple machines that are assigned the worker role.

The controller parses out work to each worker node. Each worker node sends the results of its computations back to the controller.

Distributed CAS Server

CAS can be deployed on a cluster of machines. This massively parallel processing (MPP) architecture is appropriate for analyzing large data sets. Analysis proceeds on tables that are already made available to the server (loaded) or on tables that are gathered or created by the server on demand.

Session Processes

When a user connects to the server with a client, the server starts a session process for the user. Afterward, the client communicates with the session process.

A server running in symmetric multiprocessing mode (SMP mode) consists of a controller only, and the server starts a session controller process only. It is the session controller process that operates on rows of data.

In a distributed server (MPP mode), a session process is created on each machine in the cluster. These processes are sometimes referred to as the session controller and session worker processes.

Even though the sessions have their own operating system processes, the server processes must continue to run. When the server process terminates, the session processes also terminate.
**Paths List**

From a CAS server, all access to file system paths (host and HDFS directories) is through caslibs. To limit the paths that are available to non-administrators when they created or edit a caslib, use one of the following approaches:

- Create a blacklist of paths that should not be available.
- Create a whitelist of paths that should be available.

**Note:** Black and whitelists do not apply to HDFS.

Here are key points:

- Paths must be absolute.
- All subdirectories of each specified path are affected.
- Paths list constraints do not affect access to existing caslibs.
- If you do not define a blacklist or whitelist, no paths list constraints are in effect.
- Paths list constraints do not apply to users who assume the Superuser role or the Data role.
- Only users who assume the Superuser role for a server can see and manage that server’s paths list.

**Note:** Access to third-party databases is not affected by a server’s blacklist or whitelist.

**Caslib Management Privileges**

**Caslib Management Privileges**

<table>
<thead>
<tr>
<th>Task</th>
<th>Who Can Perform the Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add global casibs.</td>
<td>Superusers and Data administrators. Users who have global caslib management privileges.</td>
</tr>
<tr>
<td>Add session casibs.</td>
<td>Superusers and Data administrators. Users who have session caslib management privileges.</td>
</tr>
<tr>
<td>Delete global casibs.</td>
<td>Superusers and Data administrators. Users who have global caslib management privileges can delete any global caslib for which they have the ReadInfo and ManageAccess permissions.</td>
</tr>
<tr>
<td>Delete session casibs.</td>
<td>Superusers and Data administrators. Users who have session caslib management privileges can delete any session caslib for which they have the ReadInfo and ManageAccess permissions.</td>
</tr>
<tr>
<td>Adjust caslib management privileges.</td>
<td>Superusers and Data administrators.</td>
</tr>
</tbody>
</table>

* Global caslib management privileges correspond to the ManageAccess permission on the _GLOBAL caslib. Session caslib management privileges correspond to the ManageAccess permission on the _SESSION caslib.

**Note:** Data administrators are displayed in CAS Server Monitor only.
**SAS Cloud Analytic Services: Reference**

**Configuration File Options**

**How Do I Use CAS Configuration File Options?**

SAS Cloud Analytic Services (CAS) configuration file options are specified in the BOSH manifest under the `cas_controller` instance under `cas.config`. The BOSH deployment automatically shares CAS properties from the CAS controller job with the CAS worker jobs.

**Note:** When specifying a CAS configuration file option, the `cas.` prefix is not needed. The BOSH deployment process adds the prefix for you.

Here is an example:

```yaml
- name: cas_controller
  instances: 1 # DO NOT CHANGE
  jobs:
    <% if programming_only == false %>
    - {name: consul, release: sas-bshconsul-<%= package_type %>}
    <% end %>
    - {name: install_config_nfs, release: sas-bshnfs-<%= package_type %>}
    - {name: install_config_sssd, release: sas-bshsssd-<%= package_type %>}
    - {name: install_java, release: sas-bshjava-<%= package_type %>}
    - {name: install_numa, release: sas-bshnuma-<%= package_type %>}
    <% if stemcell == "ubuntu" %>
    - {name: install_packages, release: sas-bshpkginst-<%= package_type %>}
    <% end %>
    - name: cas_controller
      release: sas-bshvdmmlcas-<%= package_type %>
      provides:
        controller: {as: cas_controller}
      vm_type: <%= vm_type %>
      stemcell: <%= stemcell %>
      persistent_disk_type: <%= disk_type %>
      networks:
        - name: <%= private_network_name %>
          default: [dns, gateway]
        - name: <%= public_network_name %>
          static_ips: [<%= cas_controller_ip %>]
      properties:
        cas:
          key: <%= cas_key %>
          config:
            gcport: 5580
            mode: <%= cas_mode %>
        <% if programming_only == false %>
            servicesbaseurl: <%= servicesbaseurl %>
        <% end %>
        <% if cas_mode == "mpp" %>
            initialworkers: <%= cas_worker_count %>
        <% end %>
      consul:
```
server: false
join_hosts:
- <%= consul_ip %>

### Configuration File Options Reference

cas.ADDFMTLIB='*table-name*.sashdat <table-name>.sashdat <...>>'
specifies the format libraries to add and promote during server start-up prior to any sessions starting.

*table-name*.sashdat is a persisted format library that was previously saved using the CAS Mysess SAVEFMTLIB statement. Use the CAS statement SAVEFMTLIB to persist a format library to a caslib or a table.

Begin and end the string of one or more table names with single quotation marks. Separate multiple library names with a whitespace character.

**Valid in** CAS configuration file

**Category** Formats

**Note** During session start up, each format library that is identified by `cas.addfmtlib` is automatically added to the format search for the session.

**See** cas.FMTCASLIB on page 62

**Example**
cas.addfmtlib='mylib1.sashdat mylib2.sashdat mylib3.sashdat'

cas.APPTAG='tag-string'
specifies an arbitrary string to prefix to log messages.

Using `apptag` helps determine which log messages are associated with an application.

**Valid in** CAS statement SESSOPTS option

**Category** Session

**Default** No `tag-string`

**Note** The CAS server uses `apptag` when writing to its log.

**See** cas.LOGCFGLOC

**Example**
apptag='my_app'

cas.CMPOPT='optimization-value <optimization-value <...>>' | 'all' | 'none'
specifies the type of code generation optimizations to use in the SAS language compiler.

- `optimization-value`

  specifies the type of optimization that the SAS compiler is to use. Specify one or more of the following as a space-delimited list enclosed in quotation marks:

  - 'extramath' | 'noextramath'

    specifies whether the compiler is to retain or remove the extra mathematical operations that do not affect the outcome of a statement.

  - 'funcdifferencing' | 'nofuncdifferencing'

    specify `funcdifferencing` to calculate numeric-differencing derivatives for user-defined functions. Specify `nofuncdifferencing` to calculate analytic derivatives for user-defined functions.
- 'guardcheck' | 'noguardcheck'
  specifies whether the compiler checks for array boundary problems.

Note: noguardcheck is set when cmpopt is set to 'all' or 'none'.

- 'misscheck' | 'nomisscheck'
  specifies whether to check for missing values in the data.

- 'precise' | 'noprecise'
  specify precise to handle exceptions at the operation boundary. Specify noprecise to handle exceptions at the statement boundary.

  'all'
  specifies that the compiler is to optimize the machine language code by using the noextramath, nomisscheck, noprecise, noguardcheck, and nofuncdifferencing optimization values.

Note: ‘all’ cannot be specified with other values.

  'none'
  specifies that the compiler is not set to optimize the machine language code by using the extramath, misscheck, precise, noguardcheck, and funcdifferencing optimization values.

Note: ‘none’ cannot be specified with other values.

Valid in CAS statement SESSOPTS option

CAS configuration file

Category Action

Default noextramath, nofuncdifferencing, noguardcheck, nomisscheck, and noprecise

Note If the data contains a significant amount of missing data, specify misscheck to optimize the compilation. Otherwise, specify nomisscheck.

Example In this example, the SAS compiler is set to retain the extra mathematical operations, check for missing values, and handle exceptions at an operation boundary:
cas.cmpopt='extramath misscheck precise'

cas.COLLATE='mva' | 'uca'

Specifies the collating sequence for sorting.

mva specifies SAS client collating. uca specifies a locale-appropriate collating sequence.

Valid in CAS statement SESSOPTS option

CAS configuration file

Category Sort

Default 'uca'

Example cas.collate='mva'

cas.COLOCATION='none' | 'hdfs'

specifies whether to create a personal caslib (hdfs) at CAS server start-up.

A server started in MPP mode defaults to hdfs because it assumes it is co-located with Hadoop. Specify none for the server running in MPP mode not to create a personal caslib at start-up.
Valid in | CAS configuration file
---|---
Category | Caslib
Default | `cas.colocation='hdfs'`
Requirement | Used with `cas.mode='mpp'` and `cas.hdfsuserloc`.
Example | In this example, the CAS server is running in MPP mode and is not co-located with Hadoop. At start-up, the CAS server does not create a personal caslib for the user ID under which the server is run. 
| `cas.colocation='none'`

**cas.DATASTEPFMTERR***true | false**

Corresponds to the FMTERR in SAS. Specifies how the DATA step reacts when a format is not available. When `true`, the DATA step writes an error and stops. When `false`, the DATA step uses `$w` or `BEST12` instead of the unavailable format. (The unavailable format is still associated with variables in the output table.)

Valid in | CAS statement SESSOPTS option
---|---
Category | DATA Step
Default | True
Note | The values `true` and `false` are case sensitive.
See | FMTERR System Option
Example | In this example, the DATA step uses `$w` or `BEST12` instead of the unavailable format.
| `cas.datastepfmterr=false`

**cas.DATASTEPMSGSUMLEVEL**='all' | 'none' | 'put'

Specifies the DATA step message summary level. When the DATA step runs on multiple threads, the same message can be generated on each thread. This option controls the summary level of duplicate messages.

- **'all'**
  The first occurrence of all message and put statements are sent to the client when they occur. Duplicate occurrences of all message and put statements are summarized and sent to the client when the DATA step exits. This is the default.

- **'none'**
  All message and put statements from every thread are written to the client log. No summarization occurs.

- **'put'**
  The first occurrence of all message and put statements are sent to the client. Duplicate occurrences of messages are summarized and sent to the client when the DATA step exits. Put statements are not summarized; rather, they are sent to the client when they occur.

Valid in | CAS statement SESSOPTS option
---|---
Category | DATA Step
Default | All
In this example, all message and put statements from every thread are written to the client log. No summarization occurs.

```
cas.datastepmsgsumlevel='none'
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cas.datastepreplacetable</code></td>
<td><code>true</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td><strong>Valid in</strong></td>
<td>CAS statement SESSOPTS option</td>
<td>CAS configuration file</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>true</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The values <code>true</code> and <code>false</code> are case sensitive.</td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td><code>cas.datastepreplacetable=true</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cas.dqlocale</code></td>
<td><code>locale-code</code></td>
<td>specifies the default locale to use for data quality (DQ) operations, using the five-letter SAS Quality Knowledge Base (QKB) ISO locale code.</td>
</tr>
<tr>
<td><strong>Valid in</strong></td>
<td>CAS statement SESSOPTS option</td>
<td>CAS configuration file</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>In this example, the default locale for DQ operations is French Canadian:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>cas.dqlocale='fr_CA'</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cas.dqsetuploc</code></td>
<td><code>QKB-name</code></td>
<td>specifies the name of the default SAS Quality Knowledge Base (QKB) to use for data quality (DQ) operations.</td>
</tr>
<tr>
<td><strong>QKB-name</strong></td>
<td>is the absolute path to a SAS Quality Knowledge Base.</td>
<td></td>
</tr>
<tr>
<td><strong>Valid in</strong></td>
<td>CAS statement SESSOPTS option</td>
<td>CAS configuration file</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td><code>dqSetupLoc='/opt/sashome/SASQualityKnowledgeBases/en/my_qkb'</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cas.elastic</code></td>
<td><code>true</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td><strong>Valid in</strong></td>
<td>CAS configuration file</td>
<td></td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td><strong>Requirement</strong></td>
<td>Used with <code>cas.gcport</code>.</td>
<td></td>
</tr>
</tbody>
</table>
Supports CAS servers running MPP.

Note The values true and false are case sensitive.

See cas.GCPORT

Example In this example, the CAS controller allows new worker nodes to join the analytic cluster:
cas.elastic=true
cas.ElastiCSsl=true | false
When elasticssl=true, new machines are allowed to join the cluster when the CAS controller can authenticate their identity.
Authentication data is contained in a key file that was used to start the CAS server. Any machine that can access this key can join the CAS controller as a worker node.

Valid in CAS configuration file
Category Security
Default false
Requirement Used with cas.gcport and cas.keyfile.
Supports CAS servers running MPP.
Note The values true and false are case sensitive.
See cas.GCPORT and cas.KEYFILE
Example cas.elasticssl=true

cas.EVENTDS='event-data-set'
specifies one or more event objects that define custom date events.
event-data-set specifies the name of a data set that contains event definitions. You can use a one-level name or a two-level name, such as libref.dataset. When specifying multiple names, separate each name with a space.
Enclose event-data-set in single quotation marks.

Valid in CAS statement SESSOPTS option
CAS configuration file
Category Input Control
Example cas.eventds='mydataset'

cas.FMTCASLIB='caslib'
specifies the caslib where persisted format libraries can be found.
Enclose caslib in single quotation marks.

Valid in CAS statement SESSOPTS option
CAS configuration file
Category Formats
Default: ‘formats’

Note: Set by the system administrator.

See: cas.ADDFMTLIB on page 58

Example: cas.fmtcaslib='formats'

cas.GCPORT=port
specifies the network port that is used on a distributed server for communication between the controller and its worker nodes.

The commonly configured port is 5580.

Valid in: CAS configuration file

Category: Network

Default: 0 (random port in the range 32678–61000)

Supports: CAS servers running MPP.

See: cas.HTTPPORT and cas.PORT

Example: cas.gcport=5580

cas.HDFSUSERLOC='/hdfs-path/%USER'
For CAS servers running in MPP mode, specifies that the server create a personal caslib for each user at session start-up time in the specified HDFS path.

'hdfs-path/%USER' refers to a directory named for the user’s user ID under the specified HDFS path.

Enclose hdfs-path in single quotation marks.

Valid in: CAS configuration file

Category: Data

Requirement: cas.MODE='mpp' on page 69 and cas.COLOCATION='hdfs' on page 59

Example: In this example, the user’s caslib directory is a subdirectory named for the user ID under /user:

    cas.hdfsuserloc='/user/%USER'

cas.HTTPPORT=port | port-range
The port (or range of ports) that SAS Cloud Analytic Services listens to for HTTP communication.

The commonly configured port is 8777.

Valid in: CAS configuration file

Category: Network

Default: 0 (random port)

Note: If the first port in the range is already taken, CAS tries the next port until it finds a port that is free.

See: cas.HTTPPORTMAX, cas.GCPORT, and cas.PORT

Examples: cas.httpport=8777
**cas.httpport=8777-9000**

**cas.HTTPPORTMAX=maximum-port-range**

Specifies the maximum port range that SAS Cloud Analytic Services listens to for HTTP communication.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Network</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Range</td>
<td>0–65535</td>
</tr>
<tr>
<td>See</td>
<td>cas.HTTPPORT</td>
</tr>
<tr>
<td>Example</td>
<td>cas.httpport=8777-9000</td>
</tr>
</tbody>
</table>

**cas.INITIALWORKERS='n'**

Specifies the number of CAS worker nodes that must join the analytic cluster before CAS begins processing user connections.

cas.initialworkers enables administrators to establish an expected cluster size for configurations, where it is typical for all or most worker nodes to join elastically.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Server</td>
</tr>
<tr>
<td>Default</td>
<td>-1</td>
</tr>
<tr>
<td>Range</td>
<td>-1 to 32767</td>
</tr>
<tr>
<td>Requirement</td>
<td>cas.elastic must be set to true.</td>
</tr>
<tr>
<td>Notes</td>
<td>A value of zero indicates that the controller does not wait for any worker nodes to join the cluster before it begins to establish user connections.</td>
</tr>
<tr>
<td></td>
<td>A value of -1 indicates that the controller waits for the number of workers that is specified in the machine list file.</td>
</tr>
<tr>
<td>See</td>
<td>cas.ELASTIC and cas.MACHINELIST</td>
</tr>
<tr>
<td>Example</td>
<td>In this example, the CAS controller waits for 16 workers to join the analytic cluster before it begins processing user connections.</td>
</tr>
<tr>
<td></td>
<td>cas.initialworkers='16'</td>
</tr>
</tbody>
</table>

**cas.INTERVALDS='interval-1=libref.dataset-name-1 <interval-2=libref.dataset-name-2 ...>'**

Specifies one or more interval-name=value pairs, where the value is the name of a data set that contains user-defined intervals.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Input Control</td>
</tr>
<tr>
<td>See</td>
<td>&quot;INTERVALDS= System Option&quot; in SAS Viya System Options: Reference</td>
</tr>
<tr>
<td>Example</td>
<td>intervals='subsid1=subsid.storeHours'</td>
</tr>
</tbody>
</table>
cas.JREOPTIONS='(JRE-option <JRE-option> <...>)'
specifies the Java Virtual Machine (JVM) options that SAS Cloud Analytic Services uses at start-up. Separate JRE options with a whitespace character. Enclose any paths in quotation marks.

For the list of the valid Java options, and what they do, see [http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html](http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html)

Valid in CAS configuration file
Category Java
Default (null)
Example In the following example, the initial and maximum sizes of the memory allocation pool are set to 256 and 1024MB, respectively. Also, the log4j configuration file path and Java classpath are set:
cas.jreoptions = '(-Xms256m -Xmx1024m -Dlog4j.configuration=' .. ' -Djava.class.path=' .. env.CAS_HOME .. '/lib/base/base-tkjni.jar')'

---

cas.KEYFILE='pathname'
identifies to the CAS controller the path and filename to the X.509 digital certificate file that is used to start the server. The certificate must be signed by a CA that is trusted by the CAS server.

Enclose pathname in single quotation marks.

Valid in CAS configuration file
Category Security
Requirement Used with cas.elasticssl and cas.mode='mpp'.
Supports CAS servers running MPP.
See cas.GCPORT and cas.ELASTIC
Example cas.keyfile='/opt/TKGrid/certs/controller.pem'

---

cas.LIFETIME=minutes
indicates the duration, in minutes, that a server remains running.

Valid in CAS configuration file
Category Administration
Default 0
Example In the following example, the server shuts itself down in 120 minutes:
cas.lifetime=120

---

cas.LOCALE='POSIX-locale-string'
specifies the locale to use for sorting and formatting.

Valid in CAS statement SESSOPTS option
CAS configuration file
Category Localization
Default 'en_US'
See valid POSIX locale strings
**Example**  
```
cas.locale='fr_FR'
```

**cas.LOGCFGLOC**=`pathname`

specifies the path to the SAS logging facility logging configuration file.

Enclose *pathname* in single quotation marks.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
<td>Log</td>
</tr>
</tbody>
</table>

See [cas.APPTAG on page 58](#)

**Example**  
```
cas.logcfgloc='/opt/sas/cas1/etc/logconfig.xml'
```

**cas.LOGFLUSHTIME**=-1 | 0 | *number*

specifies the log flush time, in milliseconds.

- **-1**
  - flushes logs after each action completes.

- **0**
  - flushes logs as they are produced.

- **number**
  - flushes logs in *number* milliseconds.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
<td>Log</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>100</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>-1–86400</td>
</tr>
</tbody>
</table>

**Example**  
In the following example, the CAS server writes buffered lines to the log every 500 milliseconds:
```
cas.logflush=500
```

**cas.MACHINELIST**=`path/machine-list-file`

identifies the path and filename on the controller machine that contains the list of machines in the CAS analytics cluster.

Enclose *path* in single quotation marks.

*machine-list-file* contains all of the machines in the analytics cluster in the form:

```
<fully-qualified-domain-name controller | worker>
```

Place each machine on a separate line. For example:

```
my_machine01.example.com controller
my_machine02.example.com worker
my_machine03.example.com worker
my_machine04.example.com worker
my_machine05.example.com worker
```

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
<td>Log</td>
</tr>
</tbody>
</table>
### cas.MAXSESSIONS='n'

specifies the maximum number of concurrent sessions. Users who can assume an administrative role are not subject to the limit.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Server</td>
</tr>
<tr>
<td>Default</td>
<td>5000</td>
</tr>
<tr>
<td>Range</td>
<td>0–100000</td>
</tr>
<tr>
<td>Notes</td>
<td>Specifying zero (0) indicates that there is no session limit. This option cannot be changed after the system initializes.</td>
</tr>
<tr>
<td>Example</td>
<td>In this example, the maximum number of concurrent CAS sessions is 1,000: cas.maxsessions='1000'</td>
</tr>
</tbody>
</table>

### cas.MAXTABLEMEM=number | /[number k | m | g | t] |

specifies the maximum amount of physical memory to allocate for a table.

- **number**
  - specifies the maximum amount of physical memory, in bytes, to allocate for a table.
- '/[number k | m | g | t]'  
  - specifies the maximum amount of physical memory to allocate for a table in a unit other than bytes: k (kilobytes), m (megabytes), g (gigabytes), and t (terabytes).

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS statement SESSOPTS option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Caslib</td>
</tr>
<tr>
<td>Default</td>
<td>16M</td>
</tr>
<tr>
<td>Note</td>
<td>After this threshold is reached, the server uses temporary files and operating system facilities for memory management.</td>
</tr>
<tr>
<td>Example</td>
<td>In this example, the CAS server can allocate up to 32MB of physical memory for a table: cas.maxtablemem='32m'</td>
</tr>
</tbody>
</table>

### cas.MEMORYSIZE=number | /[number k | m | g | t] |

specifies the maximum amount of physical memory to allocate for the CAS cgroup. This limit also applies to the YARN request, when cas.USEYARN is specified.

- **number**
specifies the maximum amount of physical memory, in bytes, to allocate for the CAS cgroup and the
YARN request.

- *[number k | m | g | t]*
  
  specifies the maximum amount of physical memory to allocate for the CAS CGroup and the YARN
  request in a unit other than bytes: k (kilobytes), m (megabytes), g (gigabytes), and t (terabytes).

Valid in  CAS configuration file
Category  Administration
Default  0
See  cas.USEYARN on page 73

"Linux Cgroup" in SAS Cloud Analytic Services: Fundamentals

Example  In the following example, the maximum amount of physical memory allocated for the CAS cgroup
and the YARN request is 256GB:

```
cas.memorysize='256g'
```

cas.MESSAGELEVEL='all' | 'default' | 'error' | 'none' | 'note' | 'warning'

specifies the log message level.

Valid in  CAS statement SESSOPTS option
Category  CAS configuration file
Default  'all'
See  cas.MESSAGELEVEL on page 68

Example  cas.messagelevel='default'

cas.METRICS=true | false

causes CAS server metrics information to be displayed (true) or not displayed (false) in the SAS log.

When cas.metrics=true, you see information similar to the following displayed in the SAS log:

```
NOTE: Action 'nobs' used (Total process time):
NOTE:       real time               2.100185 seconds
NOTE:       cpu time                0.010999 seconds (0.52%)
NOTE:       total nodes             6 (192 cores)
NOTE:       total memory            1.11T
NOTE:       memory                  7.00K (0.00%)
The analytic server processed the request in 2.100185 seconds.
```

Valid in  CAS statement SESSOPTS option
Category  Log
Default  false
Note  The values true and false are case sensitive.
See  CASLIB Statement
In the following example, CAS server metrics information is not displayed in the SAS log:

```bash
cas.metrics=false
```

`cas.MODE='smp' | 'mpp'`

forces a server to be started in symmetric multiprocessing mode (`smp`) or in massively parallel processing mode (`mpp`).

**Category** Administration

**Interaction**

Used in MPP mode with `cas.machinelist`.

**Note**

The server returns an error when `cas.mode='smp'` is specified for a server with a valid machine list.

**Example**

In the following example, the CAS server is forced to start in massively parallel processing mode (MPP).

```bash
cas.mode='mpp'
```

`cas.NWORKERS=number`

specifies the number of worker nodes associated with this session.

**Valid in**

- CAS statement `SESSOPTS` option
- CAS configuration file

**Category** Administration

**Default** 0

**Range** 0–5000

**Example**

```bash
cas.nworkers=8
```

`cas.OAUTHSIGNINGCERTIFICATE='path-to-public-key-file.pem'`

contains the full path to the PEM file that contains the public key. The file must exist and be readable by the CAS process or CAS does not start.

**Valid in**

- CAS configuration file

**Category** Security

**Example**

```bash
cas.oauthsigningcertificate='/opt/keys/mypublickey.pem'
```

`cas.OAUTHSIGNINGKEY='key-string'`

contains the text of the key used to sign OAuth tokens.

**Valid in**

- CAS configuration file

**Category** Security

**Example**

```bash
cas.oauthsigningkey='zAc8GqF23Fu85e7qz7ZN2U42RhfVJ3WpwPACoE3Z7kBwASldowUaiy1lyTt5jkRb4J50vUPVWkAReR7YP15jC'
```

`cas.ONELOG=true | false`

specifies that all server logging is written to a single file. When `cas.onelog=false`, each worker node creates its own log file.

- The main controller and main worker processes each create their own log files.
Valid in: CAS configuration file

Category: Log

Default: false

Interaction: cas.onelog=true is used with cas.logcfgloc and cas.loghost.

Note: The values true and false are case sensitive.

Example: In this example, each CAS worker node creates a log file:

```
cas.onelog=false
```

**cas.PERMSTORE=‘path’**

specifies the path to a directory where the CAS server stores permissions.

Enclose path in single quotation marks.

The server saves its caslib and access control information to the cas.permstore directory periodically and when it shuts down.

Each subsequent time that the server starts, caslib and access control information is initialized from the server’s cas.permstore location.

**CAUTION! Backups of access controls are not automatically performed.** It is strongly recommended that you periodically back up each CAS server’s stored access control and caslib information. In particular, it is important to create a backup after you modify access controls or add, delete, or modify global caslibs. See SAS Viya Administration: Backup and Recovery.

Valid in: CAS configuration file

Category: Access Control

Note: Each CAS server should have its own cas.permstore location. To minimize the potential for network timing issues, it is recommended that each cas.permstore location be on the controller machine and not on a network file system. The server creates a directory with the name of the fully qualified DNS name of the machine that the main controller is running on in the specified permstore directory. Do not directly edit the files in a cas.permstore location.

Example: cas.permstore=’/var/vcap/store/cas_controller/permstore’

**cas.PORT=port**

specifies the port to which the CAS server listens.

The maximum allowable port number is 65535. If you do not specify a valid port, then the server listens on a random port in the range 32678-61000.

The commonly configured port is 5570.

Valid in: CAS configuration file

Category: Network

See: cas.GCPORT and cas.HTTPPORT

Example: cas.port=5570

**cas.PROVLIST=‘ext’ | ‘oauth’**

specifies the authentication providers that the CAS server uses to authenticate incoming user connections.

- ‘ext’
The external provider provides support for an external PAM authentication method when root access is required for authentication.

- ‘oauth’

  OAuth provider is always loaded (even when not listed) to support REST endpoints and communications between CAS worker nodes and the controller.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Security</td>
</tr>
<tr>
<td>Default</td>
<td>oauth</td>
</tr>
<tr>
<td>Note</td>
<td>The CAS server configures the specified providers and uses each in order until an authenticated connection is successful.</td>
</tr>
</tbody>
</table>
| Example          | In this example, an external provider provides support for an external PAM authentication method. Although not specified, OAuth is always loaded to support REST endpoints and communications between CAS worker nodes and the controller. cas.provlist='ext'

`cas.REMOVEDNODECANCELCANCELTIMEOUT='interval'`

when quiescing sessions in preparation for moving data from nodes that are being removed, the time that CAS will wait for long-running actions to complete before cancelling them.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Server</td>
</tr>
<tr>
<td>Default</td>
<td>120 seconds</td>
</tr>
<tr>
<td>Note</td>
<td>A value of zero indicates that a cancel request should never be sent.</td>
</tr>
<tr>
<td>Example</td>
<td><code>cas.removeNodeCancelTimeout='600'</code></td>
</tr>
</tbody>
</table>

`cas.REMOVEDNODEKILLTIMEOUT='interval'`

when quiescing sessions in preparation for moving data from nodes that are being removed, the time that CAS will wait for a cancelled action to stop before killing its session.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>CAS configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Server</td>
</tr>
<tr>
<td>Default</td>
<td>15 seconds</td>
</tr>
<tr>
<td>Note</td>
<td>A value of zero indicates that the sessions should never be killed.</td>
</tr>
<tr>
<td>See</td>
<td><code>cas.REMOVEDNODECANCELCANCELTIMEOUT on page 71</code></td>
</tr>
<tr>
<td>Example</td>
<td><code>cas.removeNodeKillTimeout='300'</code></td>
</tr>
</tbody>
</table>

`cas.RESOLVWORKERADDRESS=true | false`

specifies how CAS list node actions return CAS worker node host names.

When `true`, CAS list node actions attempt to return the list of worker node host names. If the directory name service (DNS) lookup is unresponsive, CAS cancels the lookup and resolveworkeraddress is automatically set to `false`.

When `false`, list node actions return only the IP address of elastically added nodes.
Setting `cas.RESOLVEWORKERADDRESS` to `false` ensures that the analytic cluster is less impacted by an unresponsive DNS configuration. However, some output is displayed as IP addresses instead of host names.

**Valid in**: CAS configuration file  
**Category**: Server  
**Default**: True  
**See**: `cas.ELASTIC`  
**Example**: `resolveworkeraddress=false`

**`cas.SERVICESBASEURL='URL'`**  
 specifies the URL that enables CAS server to authenticate and to use SAS Viya services. `URL` points to the deployed SAS Viya web services.  
When set, `cas.SERVICESBASEURL` creates a hybrid authentication environment where username-password authentication is converted to an OAuth token and CAS can fetch groups from SAS Viya services and use features such as the credentials vault.

**Valid in**: CAS configuration file  
**Category**: Security  
**Notes**: `URL` must match the reverse proxy server host name and port to enable CAS to communicate with SAS Viya web services.  
Enclose `URL` in single quotation marks.

**Example**: `cas.servicesbaseurl='http://company.example.com'`

**`cas.SUBSETSESSIONCOPIES=number-of-blocks`**  
 specifies the number of extra block copies made for failover in either of the following scenarios:  
- a session is smaller than the full server.  
- CAS reads blocks of an HDFS remotely.

**Valid in**: CAS configuration file  
**Category**: Administration  
**Default**: 0  
**Example**: `cas.subsetsessioncopies=3`

**`cas.TIMEOUT=seconds`**  
 specifies the SAS Cloud Analytic Services session time-out in seconds for a new or existing session.

**Valid in**:  
CAS statement `SESSOPTS` option  
CAS configuration file  
**Category**: Session  
**Default**: In order of descending precedence:  
1. CAS statement `TIMEOUT=` option value, if specified
2. SAS system option CASTIMEOUT=, if you explicitly set it in SAS to a value greater than 0
3. 60

Range 0–31536000

Notes The session time-out starts when the number of connections to the session becomes zero and no actions are executing.

If a connection is established before the time-out expires, the time-out is canceled. Otherwise, the session is automatically terminated when the time-out expires.

When set to 0, the session is terminated immediately when the connection count becomes zero.

See “CASTIMEOUT= System Option” in SAS Cloud Analytic Services: Language Reference

Example cas.timeout=100

cas.TIMEZONE=offset
specifies the time zone offset, in hours, from UTC.

Valid in CAS statement SESSOPTS option

CAS configuration file

Category Session

Default -1

Range -1–23

Example cas.timezone=5

cas.USERLOC='%HOME' | 'pathname%/USER'
specifies that the CAS server create a personal caslib for each user at session start-up time in the specified location.

' %HOME ' equates to the user’s operating system $HOME directory.

'pathname%/USER' refers to a directory named for the user’s user ID under the specified file system path.

Enclose pathname in single quotation marks.

Valid in CAS configuration file

Category Caslib

Examples In this example, the personal caslib directory is the user’s operating system $HOME directory:
cas.userloc='%HOME'

In this example, the user’s personal caslib directory is named for his or her user ID and is located under /local:
cas.userloc='/local/%USER'

cas.USEYARN=true | false
adds a reservation request to YARN for CAS memory size.

The memory limit for the YARN request is set with cas.memorysize.

Valid in CAS configuration file
Category: Administration

Default: false

See: cas.MEMORYSIZE on page 67

“Linux Cgroup” in SAS Cloud Analytic Services: Fundamentals

Example: cas.useyarn=true

Grouped by Categories

Access Control Options
- cas.PERMSSTORE on page 70

Action Options
- cas.CMPOPT on page 58

Administration Options
- cas.SUBSETSESSIONCOPIES on page 72
- cas.ELASTIC on page 61
- cas.LIFETIME on page 65
- cas.MACHINELIST on page 66
- cas.MEMORYSIZE on page 67
- cas.MODE on page 69
- cas.NWORKERS on page 69
- cas.USEYARN on page 73

Caslib Options
- cas.COLOCATION on page 59
- cas.MAXTABLEMEM on page 67
- cas.USERLOC on page 73

Data Options
- cas.HDFSUSERLOC on page 63

Data Quality Options
- cas.DQLOCALE on page 61
- cas.DQSETUPLOC on page 61

DATA Step Options
- cas.DATASETPFMTERR on page 60
- cas.DATASETPMSGSUMLEVEL on page 60
Formats Options
- cas.ADDFMTLIB on page 58
- cas.FMTCASLIB on page 62

Input Control Options
- cas.EVENTDS on page 62
- cas.INTERVALDS on page 64

Java
- cas.JREOPTIONS on page 65

Localization
- cas.LOCALE on page 65

Log Options
- cas.LOGCFGLOC on page 66
- cas.LOGFLUSHTIME on page 66
- cas.MESSEAGELEVEL on page 68
- cas.METRICS on page 68
- cas.ONELOG on page 69

Network Options
- cas.GCPORT on page 63
- cas.HTTPPORT on page 63
- cas.HTTPPORTMAX on page 64
- cas.PORT on page 70

Security Options
- cas.ELASTICSSL on page 62
- cas.KEYFILE on page 65
- cas.OAUTHSIGNINGCERTIFICATE on page 69
- cas.OAUTHSIGNINGKEY on page 69
- cas.PROVLIST on page 70
- cas.SERVICESBASEURL on page 61

Server Options
- cas.INITIALWORKERS on page 64
- cas.MAXSESSIONS on page 67
CAS Environment Variables

Where Do I Set CAS Environment Variables?
SAS Cloud Analytic Services (CAS) environment variables are specified in the BOSH manifest under the `cas_controller` instance under `cas.environment`. The BOSH deployment automatically shares CAS environment variables from the CAS controller job with the CAS worker jobs.

Note: When specifying a CAS environment variable, the `env.` prefix is not needed. The BOSH deployment process adds the prefix for you.

Here is an example:

```yaml
- name: cas_controller
  instances: 1 # DO NOT CHANGE
  jobs:
    <% if programming_only == false %>  
    - {name: consul, release: sas-bshconsul-<%= package_type %>} 
    <% end %>
    - {name: install_config_nfs, release: sas-bshnfs-<%= package_type %>} 
    - {name: install_config_sssd, release: sas-bsshssd-<%= package_type %>} 
    - {name: install_java, release: sas-bshjava-<%= package_type %>} 
    - {name: install_numa, release: sas-bshnuma-<%= package_type %>} 
    <% if stemcell == "ubuntu" %>  
    - {name: install_packages, release: sas-bshpkginst-<%= package_type %>} 
    <% end %>
    - name: cas_controller 
      release: sas-bshvdmmlcas-<%= package_type %>
      provides:
        controller: {as: cas_controller}
      vm_type: <%= vm_type %>
      stemcell: <%= stemcell %>
      persistent_disk_type: <%= disk_type %>
      networks:
        - name: <%= private_network_name %>
          default: [dns, gateway]
        - name: <%= public_network_name %>
          static_ips: [<%= cas_controller_ip %>]
      properties:
        cas:
```

- cas.REMOVEDONECANCELCALLTIMEOUT on page 71
- cas.REMOVEDONETEKILLTIMEOUT on page 71
- cas.RESOLVEWORKERADDRESS on page 71

Session Options
- cas.APPTAG on page 58
- cas.TIMEOUT on page 72
- cas.TIMEZONE on page 73

Sort Options
- cas.COLLATE on page 59
CAS Environment Variables Reference

Note: For information about SAS Cloud Analytic Services TLS environment variables, see “CAS TLS Environment Variables” in Encryption in SAS Viya: Data in Motion.

env.CAS_ACTION_THREAD_NICE='niceness-priority'
specifies the niceness priority for the CPU intensive threads that do CAS action processing.

Use when the CAS server has to share CPU resources with other processes, and when the CAS server is incorrectly detecting disconnected worker nodes.

Valid in: server configuration file and cas.settings file
Category: Environment
Default: 0
Range: 0–19
See: The man page for the Linux nice command.
Example: env.CAS_ACTION_THREAD_NICE='1'

env.CAS_DISK_CACHE='path [[:path] ...]'
specifies the disk paths to cache data.

Delimit multiple paths with a colon (:).

Valid in: server configuration file and cas.settings file
Category: Data
Restriction: Do not set to /tmp.
Tip: There is an advantage to using multiple physical disks. When using multiple threads, mapping files can occur concurrently if multiple disks are used. Hadoop also uses this method. Therefore, there is an advantage to using a set of disks that map to both hadoop_data and CAS_DISK_CACHE directories.
Example: env.CAS_DISK_CACHE = '/data/disk1:/data/disk2'

env.CAS_ENABLE_REMOTE_SAVE
when defined, specifies whether CAS saves blocks on remote HDFS worker nodes.
Valid in server configuration file and cas.settings file

Category Data

Example env.CAS_ENABLE_REMOTE_SAVE

env.CAS_HEARTBEAT_LOST_TIMEOUT='interval'
specifies the interval (in seconds) since the last heartbeat received from a CAS worker node before the controller treats the node as lost.

Smaller intervals detect machines that silently leave the network more quickly. Larger intervals are more tolerant of machines that might be exceptionally overloaded.

Valid in server configuration file and cas.settings file

Category Environment

Default 60 seconds

Range 60 – (no upper limit) seconds

Example CAS_HEARTBEAT_LOST_TIMEOUT='300'

env.CAS_INSTALL='install-path'
specifies the installation directory for CAS.

Valid in server configuration file and cas.settings file

Category Environment

Example env.CAS_INSTALL='/var/vcap/packages/cas-dmml/*/opt/sas/viya/home/SASFoundation'

env.CAS_LICENSE='path/license-file'
specifies the path and filename that contains the CAS license.

Valid in server configuration file and cas.settings file

Category Administration

Example env.CAS_LICENSE='/var/vcap/packages/cas-dmml/opt/sas/viya/config/etc/cas/default/license.sas'

env.CAS_REMOTE_HADOOP_PATH='SASHDAT-executables-directory'
specifies the path to the plugin location when CAS is using an HDFS caslib to a remote HDFS cluster.

Valid in server configuration file and cas.settings file

Category Environment

Default If not set, defaults to $HADOOP_HOME/bin

Note Might be needed to accommodate a non-standard Hadoop plugin.

Example env.CAS_REMOTE_HADOOP_PATH='$HADOOP_HOME/bin'

env.CAS_VIRTUAL_HOST = 'host-name'
The external host or machine name for the controller.

Use env.CAS_VIRTUAL_HOST when an external HTTP client needs to use an external address that differs from the actual host name known by the operating system. A common use is when the controller machine is behind a proxy server.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>env.CAS_VIRTUAL_HOST='my_machine'</strong></td>
<td>Use this environment variable when external HTTP clients must reach the CAS controller through a proxy. This identifies the path portion of the URL for the proxy.</td>
</tr>
<tr>
<td><strong>env.CAS_VIRTUAL_PATH='URL-path-suffix'</strong></td>
<td>Use this environment variable when external HTTP clients must reach the CAS controller through a proxy. This identifies the path portion of the URL for the proxy.</td>
</tr>
<tr>
<td><strong>env.CAS_VIRTUAL_PORT=port</strong></td>
<td>The external port number for the controller. Use env.CAS_VIRTUAL_PORT when an external HTTP client needs to use a port that differs from the actual port that is local to the controller machine. A common use is when the controller machine is behind a proxy server.</td>
</tr>
<tr>
<td>**env.CAS_VIRTUAL_PROTOCOL='http</td>
<td>https'**</td>
</tr>
<tr>
<td>**env.CASCLIENTDEBUG=true</td>
<td>false**</td>
</tr>
<tr>
<td><strong>env.HADOOP_HOME='path'</strong></td>
<td>specifies the standard HADOOP_HOME variable used by Hadoop.</td>
</tr>
</tbody>
</table>
**env.HADOOP_NAMENODE**: `machine-name [:machine-name]`

Identifies which machines in the Hadoop cluster are NameNodes. There can be up to two Hadoop NameNodes. Separate machine names with a colon (:). *Machine-name* can be a name, fully qualified domain name, or an IP address for a machine.

- **Valid in**: server configuration file and cas.settings file
- **Category**: Data
- **Example**: `env.HADOOP_NAMENODE='my_namenode1:my_namenode2'`

**env.TKTXTANIO_BINDAT_DIR**: `install-path`

Specifies the installation directory for SAS linguistic binary files required to perform text analysis.

- **Note**: This environment variable is valid only on native operating systems such as Linux.
- **Note**: TKTGDat.sh contains the SAS linguistic binary files required to perform text analysis in SAS LASR Analytic Server with SAS Visual Analytics and to run PROC HPTMINE and HPTMSCORE with SAS Text Miner.

- **Valid in**: cas.settings file
- **Category**: Data
- **Example**: `env.TKTXTANIO_BINDAT_DIR='/opt/sas/viya/home/SASFoundation/utilities/TKTGDat'`

**LD_LIBRARY_PATH**: `path [:path] ...`

Specifies the path to search for additional shared libraries.

- **CAUTION!** SAS Cloud Analytic Services ignores any instance of LD_LIBRARY_PATH found in the server configuration file. Set LD_LIBRARY_PATH in the cas.settings file only.

- **Valid in**: cas.settings file
- **Category**: Data
- **Example**: `export LD_LIBRARY_PATH=/var/my_libs:/share/groups_libs:$LD_LIBRARY_PATH`

**Grouped by Categories**

- **Administration Variables**
  - `env.CAS_LICENSE` on page 78

- **Data Variables**
  - `env.CAS_DISK_CACHE` on page 77
  - `env.HADOOP_HOME` on page 79
  - `env.HADOOP_NAMENODE` on page 80
  - `env.CAS_ENABLE_REMOTE_SAVE` on page 77
  - `env.TKTXTANIO_BINDAT_DIR` on page 80
  - `LD_LIBRARY_PATH` on page 80
Environment Variables
- env.CAS_INSTALL on page 78
- env.CAS_REMOTE_HADOOP_PATH on page 78

Log Variables
- env.CASCLIENTDEBUG on page 79

Network Variables
- env.CAS_VIRTUAL_HOST on page 78
- env.CAS_VIRTUAL_PATH on page 79
- env.CAS_VIRTUAL_PORT on page 79
- env.CAS_VIRTUAL_PROTOCOL on page 79
Other Servers and Services

Other Servers and Services: Overview

Servers

Aside from SAS Cloud Analytic Services, SAS Viya contains these other servers:

- SAS Workspace Server and Object Spawner

  SAS Object Spawners interact with SAS by creating a server process for each client connection. SAS Workspace Servers are initialized by the SAS Object Spawner. An object spawner runs on the same machine as the workspace server, listens for requests, and launches the servers as necessary.

  The SAS Workspace Server enables client programs to access SAS libraries, to perform tasks by using the SAS language, and to retrieve results. Each workspace server process is owned by the client user that made the server request.

- SAS Configuration Server (Consul)

  SAS Configuration Server is based on HashiCorp’s Consul. Consul is a distributed, highly available registry that contains service configuration data and availability and overall performance (health) information. Configuration data resides in SAS Configuration Server as key-value pairs. This data is used by SAS Viya microservices at start-up to load default values and to discover any service dependencies.

- SAS Infrastructure Data Server (PostgreSQL)

  SAS Infrastructure Data Server is used for transactional storage by SAS middle-tier software. It is also used by some SAS solutions software for user content such as reports, custom groups, comments, authorization rules, selected source definitions, attachments, and user preferences. The server is configured specifically to support SAS software, and is based on PostgreSQL version 9.

- pgpool-II

  SAS provides pgpool-II (version 3) open-source software to enable you to manage PostgreSQL clusters. The pgpool-II software resides and operates between SAS Infrastructure Data servers and clients. All data connections and database requests are routed through the pgpool service.

- SAS Message Broker (RabbitMQ)
SAS Message Broker is an integral part of the event-driven architecture in which SAS Viya services participate. SAS uses a set of event APIs that are dependent on Spring Integration and Spring AMQP for interacting with the message broker. The AMQP-compliant message broker that SAS uses is Pivotal’s RabbitMQ. The SAS event APIs provide a layer of abstraction between the message broker and its clients. The SAS event APIs also prevent code from breaking, which could result if SAS changed its third-party message broker from RabbitMQ to another third-party message broker in the future.

- SAS Cache Server (Geode Server)
  SAS uses Apache Geode Server to provide access to SAS applications in the cloud. Geode pools memory, CPU, and network or local resources across multiple processes to manage application objects. An in-memory data management system, Geode Server provides reliable asynchronous event notifications and guaranteed message delivery.

- SAS Cache Locator (Geode Locator)
  Geode locators provide both discovery and load balancing services. You configure clients with multiple locator services. The locators maintain a dynamic list of member servers.

Like typical BOSH applications, all these servers also run on BOSH virtual machines and are operated and managed through BOSH.

Properties for the servers are maintained through the BOSH manifest.

**Services**

SAS Viya contains several services often referred to as *microservices*. A microservice is a small service that runs in its own process and communicates with a lightweight mechanism (HTTP).

Some of the services in SAS Viya are: Authorization, Backup, CAS Management, and so on. Use SAS Environment Manager to see the complete list of SAS Viya services. For more information, follow the first three steps in “Edit Configuration Instances” in SAS Viya Administration: Configuration Properties.

Note: A programming-only deployment does not use most SAS Viya services.

---

**Other Servers and Services: Operate**

**Servers**

Like typical BOSH applications, SAS Viya servers also run on BOSH virtual machines and are operated and managed through BOSH.

Note:

Whenever you restart the SAS Viya servers, you must also restart the SAS Viya services.

For more information, contact your site’s Cloud Foundry administrator.

**Services**

SAS Viya services run in the Cloud Foundry Elastic Runtime.

Sign in to your Cloud Foundry secure administrative host (jump server) from where you can run the necessary Cloud Foundry command line interface (cf CLI) commands to stop, start, and restart SAS Viya services.

To obtain a list of SAS Viya services, refer to the list of servers and services in SAS Environment Manager. Follow the first three steps in “Edit Configuration Instances” in SAS Viya Administration: Configuration Properties. (In Step 3 choose Definitions instead of All Services.)
To restart all SAS Viya services, run the following command from your jump server:

```
for i in $(cf a | awk '{print $1}' | tail -n +5); do cf restart ${i}; done
```

For more information, consult with your site's Cloud Foundry administrator.

---

**Other Servers and Services: Add Service Instances**

Although SAS Viya 3.2 on Cloud Foundry does not support multiple instances of servers running on the BOSH director, you can add multiple instances of services that run on the Elastic Runtime.

**Note:** Any changes to applications that are not included in the manifest are lost the next time you redeploy with BOSH.

Sign in to your Cloud Foundry secure administrative host (jump server) from where you can run the following command: `cf scale application-name -i number-of-instances`.

For more information about how to use the Cloud Foundry command-line interface (cf CLI), see your Cloud Foundry documentation.

To add additional CAS workers, see [SAS Viya 3.2 for Cloud Foundry: Deployment Guide](#).
Troubleshooting

SAS Studio appears to be unresponsive.
Explanation:
Performing a BOSH recreate on the CAS controller while a CAS session is still running might be the cause.
Resolution:
SAS Studio eventually responds.
In the future, stop all CAS sessions before issuing the BOSH recreate command.

ERROR: Contact your SAS Installation Representative to obtain your updated SAS Installation Data (SID) file, which includes SETINIT information.
Explanation:
Your SAS Viya license has expired.
Resolution:
Follow the steps in Apply New Licenses on page 43.

SAS Logon Manager fails to start.
Explanation:
When the SAS Viya servers running in the BOSH virtual machines (VMs) are shut down and then restarted automatically, the geodelocator process is not able to properly de-register from consul.
Resolution:
Stop and restart the SAS Cache Server (also known as the Geode Server).