SAS® Viya® 3.4 Administration: Configuration Properties

Configuration Properties: Overview **2**

Configuration Properties: How to Configure Services **3**
- Introduction 3
- Navigation 3
- Edit Configuration Instances 3
- Create Configuration Instances 3
- Review Default Configuration Values 4
- Set Time-out Interval for SAS Viya Web Applications 5
- Disable Opt-In Notifications 7

Configuration Properties: How to Configure SAS Studio 4.x **8**
- Update SAS Studio Configuration Properties 8
- Configuring Global Folder Shortcuts 9
- Set File Navigation Options 9

Operations **10**
- Automate Configuration Properties during Deployment (Ansible) 10

Configuration Properties: Concepts **11**
- What Is SAS Viya Configuration? 11
- Configuration Components 11
- How Configuration Definitions and Instances Are Displayed 11
- What Services Must Be Restarted? 13
- How SAS Viya Configuration Works 14

SAS Studio 5.x and 4.x Administration Differences **16**
- Overview 16
- Deployment Environments 17
- Authentication Differences 17
- Accessing SAS Content 17
- Improved Recovery 19
- Architectural Comparison 20

Configuration Properties: Troubleshooting **20**

Configuration Properties: Reference (Services) **20**
- Application Registry Service 20

Audit Service . . . 20
Authorization Service . . . 21
Backup Service . . . 22
Cache Services . . . 22
CAS Management Service . . . 25
Compute Service . . . 26
Notifications Service . . . 26
CAS Proxy Service . . . 26
Collections Service . . . 27
Configuration Service . . . 27
Cross Domain Proxy Service . . . 27
Device Management Service . . . 28
Identities Service . . . 28
Mail Service . . . 34
Maps Service . . . 35
Monitoring Service . . . 36
Report Data Service . . . 36
Report Packages Service . . . 39
Report Renderer Service . . . 40
Secret Manager Service . . . 41

Configuration Properties: Reference (Applications) . . . **42**
SAS Data Explorer . . . 42
SAS Data Studio . . . 43
SAS Infrastructure Data Server . . . 44
SAS Logon Manager . . . 44
SAS Studio 5.x . . . 50
SAS Studio 4.x . . . 51

Configuration Properties: Reference (System) . . . **53**
Commons REST Client . . . 53
Java Virtual Machine (JVM) . . . 54
Security . . . 54
Spring Boot Services . . . 56
zones . . . 57

Configuration Properties: Interfaces . . . **57**
Interfaces to CAS Administration . . . 57

Configuration Properties: Overview

You manage configuration properties for SAS Viya servers, services, and applications using the Configuration pages in SAS Environment Manager.

Note: A programming-only deployment does not use SAS Viya services and SAS Environment Manager.

You manage configuration properties for SAS Studio 4.x by modifying its configuration file. For more information, see How To Configure SAS Studio 4 on page 8.

Configuration Properties: How to Configure Services

Introduction

These instructions explain how to view and modify service configuration properties using SAS Environment Manager.

Navigation

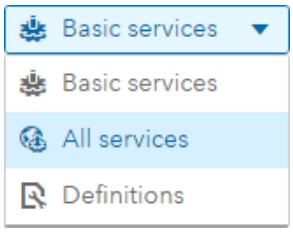
In the applications menu (☰), select **Administration** ⇨ **Manage Environment**. In the navigation bar, click 🔧.

The Configuration page is an advanced interface. It is available to only SAS Administrators.

Edit Configuration Instances

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

1 Using the drop-down list, choose **All services**.



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

3 Next to the configuration instance, click ✎.

4 In the Edit Configuration dialog box, change the value in one or more of the configuration property fields.

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

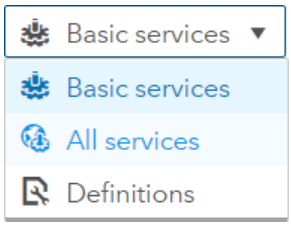
6 On a non-cloud platform, such as native Linux, some services require that you restart them when configuration changes are made. See "What Services Must Be Restarted?" on page 13.

Create Configuration Instances

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 must create a new configuration instance of logging.level for that service.

1 Using the drop-down list, choose **All services**.

4



2 Select a service for which you want to create a new configuration instance.

3 At the top of the content pane, click **New Configuration**.

4 In the Select Definition dialog box, select a configuration definition from which to create a new configuration instance.

5 In **Services**, make sure that the service displayed is the one for which you want to create a new definition. If the correct service is displayed, skip to Step 6.

Otherwise, do the following:

a Next to **Services**, click .

b In the Choose Services dialog box, highlight the service to which the configuration instance you are creating applies, and click .

c Remove any services for which you do not want to create a configuration instance, by highlighting the service and clicking .

d When you are finished, click **OK**.

6 Continue entering values. When you are finished, click **Save**.

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

7 On a non-cloud platform, such as native Linux, some services require that you restart them when configuration changes are made. See "What Services Must Be Restarted?" on page 13

Review Default Configuration Values

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

2 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

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

4 Most configuration definitions apply to only one service. In the New Configuration dialog box, if there is no edit icon ( ) next to the **Services** field, skip to Step 5.

Otherwise, do the following:

a Next to **Services**, click .

b In the Choose Services dialog box, highlight the service to which the configuration instance you are creating applies, and click .

c When you are finished, click **OK**.

5 Continue entering values. When you are finished, click **Save**.

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

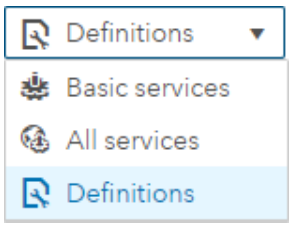
6 Repeat steps 2 – 5 for every configuration instance that is incomplete .

Set Time-out Interval for SAS Viya Web Applications

Using SAS Environment Manager, you can change the session time-out interval for one or more SAS Viya web applications. The session time-out interval is the specific period of time that a web application waits before it signs off users' inactive sessions.

Important: In SAS Studio 4.x, individual users set the session time-out interval up to a maximum that is defined by the administrator with `webdms.maxSessionTimeoutInHours`. For more information, see "SAS Studio 4.x" on page 51.

1 Using the drop-down list, choose **Definitions**.

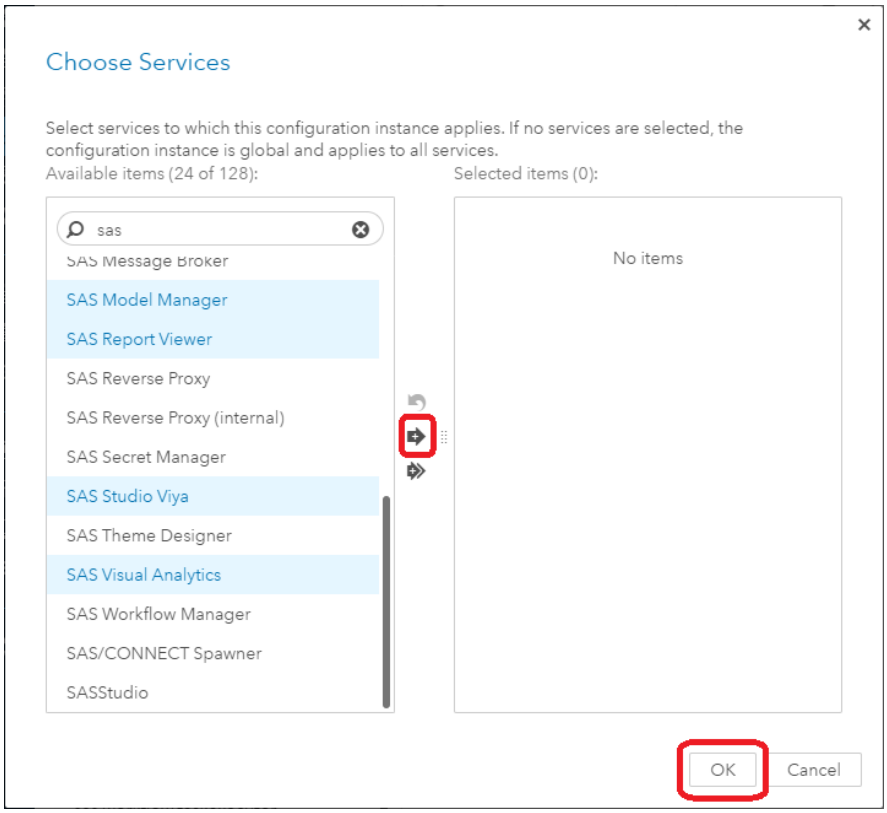


2 In the list of configuration definitions, select **server** and in top right corner of the view, click **New Configuration**.

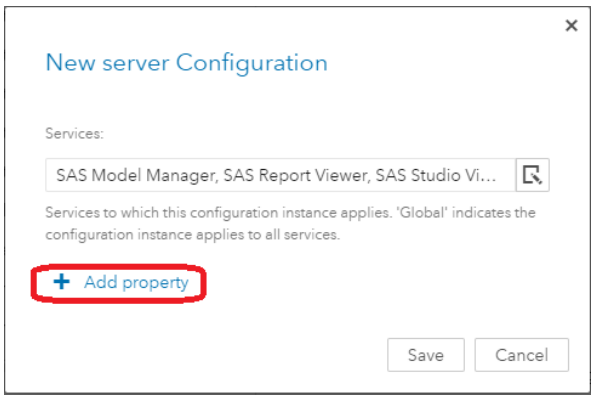
3 In the New server Configuration dialog box, click .

4 In the Choose Services dialog box, highlight one or more SAS Viya web applications, click , and click **OK**.

6

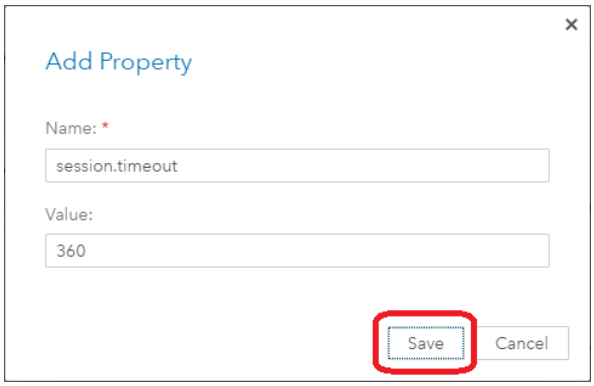


5 In the New server Configuration dialog box, click **Add property**.



6 In the Add property dialog box, in the **Name** field, enter the following Spring server property: `session.timeout`.

7 In the **Value** field, enter the number of seconds you want the SAS Viya web applications to wait before they sign off users' inactive sessions.



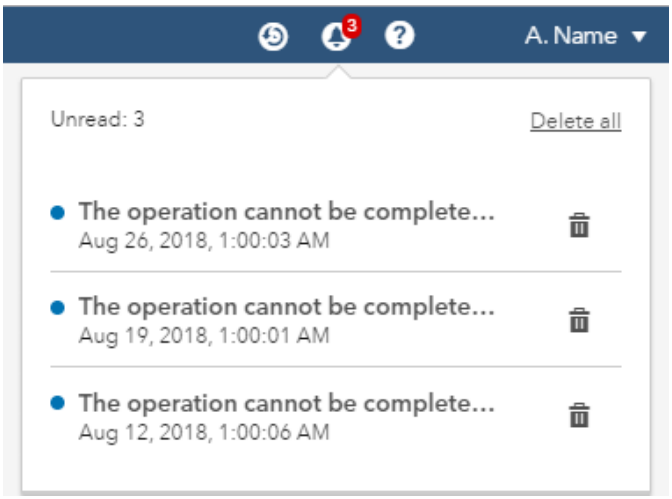
8 Click **Save**.

Your change takes effect for any new sign-ins to a SAS Viya web application.

Disable Opt-In Notifications

SAS Viya provides messages called *notifications* in various web applications. Users are alerted that one or more subscription-based (opt-in) notifications are available with a number circumscribed by a red circle next to the on the right side of the web application's banner. (The number indicates the number of unread notifications.)

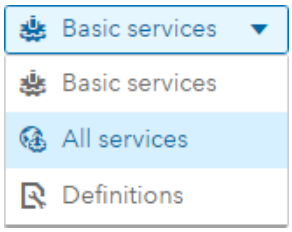
When you click the bell icon, a pop-up displays the notifications. The pop-up enables you to delete one or all of the notifications.



The SAS Viya Notifications service processes only events that are published by participating services and applications, such as SAS Visual Analytics and the Backup service.

To disable opt-in notifications, follow these steps:

1 Using the drop-down list, choose **All services**.



2 In the search field, enter **notifications**.

3 In list, select **Notifications service**.

> **TIP** At the top right side of the window, click ⌃ to easily locate the **Notifications service**.

4 In the content pane, locate **sas.notifications** and click ✎ on the right side of the window.

5 In the Edit sas.notifications Configuration dialog box, disable the **Enabled** slider and click **Save**.

Important: Disabling `sas.notifications.enabled`, turns off all subscription-based notification channels including subscription-based email notifications. To disable only subscription-based email notifications, disable `sas.notifications.delivery.mail.enabled`.

Configuration Properties: How to Configure SAS Studio 4.x

Update SAS Studio Configuration Properties

Note: SAS supplies two versions of SAS Studio, 4.x and 5.x. SAS Viya programming-only deployments use SAS Studio 4.x. For a comparison of the two SAS Studio versions, see "SAS Studio 5.1 and 4.4" in *What's New in SAS 9.4 and SAS Viya*.

Note: You set SAS Studio 5.x properties in the same way as the other SAS Viya applications and services. For more information, see "Configuration Properties: How to Configure Services".

To customize web application configuration properties for SAS Studio 4.x, edit init_usermods.properties, in the path appropriate for your operating system:

- Linux:

  /opt/sas/viya/config/etc/sasstudio/default/init_usermods.properties

- Windows:

  \ProgramData\SAS\Viya\etc\sasstudio\default\init_usermods.properties

Note: For sites that use Ansible: Ansible updates init_deployment.properties when it is run. Therefore, SAS Studio configuration changes that you make to init_usermods.properties are not overwritten by Ansible and are carried forward.

For a listing of configuration properties that you can update, see "SAS Studio 4.x" on page 51.

Changes take effect after you restart the web application. For more information, see instructions appropriate for your operating system.

> **TIP** Values that you specify in the init_usermods.properties file have precedence over corresponding values in other files. Unlike values in other files, values in the init_usermods.properties survive software upgrades.

Configuring Global Folder Shortcuts

In SAS Studio 4.x, you can create folder shortcuts from the **Server Files and Folders** section in the navigation pane. You might want to create global shortcuts for all the users at your site, so each user does not have to create these shortcuts manually.

1 In the init_usermods.properties file, specify a directory path for the `webdms.globalSettings` property.

By default, this directory path is:

- Linux:

  `/opt/sas/spre/home/SASFoundation/GlobalStudioSettings`

- Windows:

  `\Program Files\SAS\SPRE\SASFoundation\GlobalStudioSettings`

Important: If you choose to use this default, you must create the **`GlobalStudioSettings`** directory.

2 In an XML editor, create a shortcuts.xml file.

If you are trying to create a shortcut to a network location, here is the format of the shortcuts.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Shortcuts>
<Shortcut type="disk" name="network-location" dir="directory-path"/>
</Shortcuts>
```

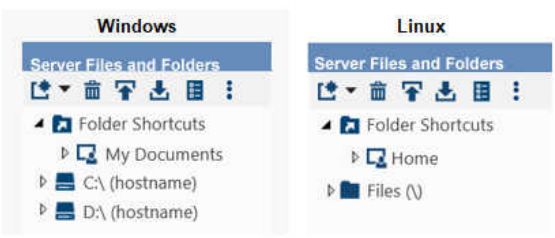
3 Save the shortcuts.xml file to the global settings directory.

Set File Navigation Options

For SAS Studio 4.x, the root directories that are available in the **Server Files and Folders** section depend on whether you specify to show the system root directory with the `webdms.showSystemRoot` property. For information, see "Update SAS Studio Configuration Properties".

- If `webdms.showSystemRoot` is set to True, the **Server Files and Folders** section displays the following information:
  - On Windows, the **Folder Shortcuts** folder includes a predefined shortcut to your **`My Documents`** folder.
  - On Linux, the **Folder Shortcuts** folder includes a predefined shortcut to your home directory. The **Files** folder is mapped to the system **`root`** directory for the server.

  Figure 1 *If the webdms.showSystemRoot Property is True*



- If `webdms.showSystemRoot` is set to False, the **Server Files and Folders** section displays an empty **Folder Shortcuts** folder and a **Files** folder.
  - On Windows, the **Files** folder is mapped to your **`My Documents`** folder.
  - On Linux, the **Files** folder is mapped to your `$HOME` directory.

Figure 2 *If the webdms.showSystemRoot Property is False*

Operations

Automate Configuration Properties during Deployment (Ansible)

You can deploy SAS Viya with configuration values that are customized to your site by running your Ansible playbook with sitedefault.yml. Using sitedefault.yml, enables you to provision multiple machines in the same manner, and prevents you from having to modify configuration values with an administration interface after deployment.

Note: It is extremely important that the initial values applied with sitedefault.yml are correct. After you set a value with sitedefault.yml, you cannot re-run sitedefault.yml to change that value. You can re-run sitedefault.yml only to set properties that have not already been set. To change properties set with sitedefault.yml, you must use the sas-bootstrap-config CLI directly, or use another administration interface, such as SAS Environment Manager.

To set configuration values using sitedefault.yml, follow these steps:

1 Sign on your Ansible controller with administrator privileges, and locate the file, **`/playbook/roles/consul/files/sitedefault_sample.yml`**.

2 Make a copy of sitedefault_sample.yml and name the copy, sitedefault.yml.

3 Using a text editor, open sitedefault.yml and add values that are valid for your site.

For information about the LDAP properties used in sitedefault.yml, see “sas.identities.providers.ldap” on page 29.

For information about the all the properties that can be used in sitedefault.yml, see “Configuration Properties: Reference (Services)” on page 20.

CAUTION! Some properties require passwords. If properties with passwords are specified in sitedefault.yml, you must secure the file appropriately. If you chose not to supply the properties in sitedefault.yml, then you can enter them using SAS Environment Manager. (Sign in to SAS Environment Manager as sasboot, and follow the instructions in “Configure the Connection to Your Identity Provider” in *SAS Viya for Linux: Deployment Guide*.)

4 When you are finished, save sitedefault.yml and make sure that it resides in the **`/playbook/roles/consul/files`** directory of the playbook.

5 Run your Ansible playbook using the sitedefault.yml file.

Here is an example:

```
ansible-playbook site.yml
```

For a complete list of playbook commands, see "Deploy the Software" in *SAS Viya for Linux: Deployment Guide*.

6 After the playbook is run, verify that the configuration values are successfully loaded into the configuration server by performing the following steps:

   a Verify that a copy of sitedefault.yml resides in `/viya/config/etc/consul.d/default/`.

   b Verify that config-kv-bulkload-sitedefault.json resides in `/viya/config/etc/consul.d/`.

   c View the configuration properties for a configuration definition such as, SAS Logon Manager, in SAS Environment Manager to verify that the specified values are present.

   For more information, follow the first five steps in "Edit Configuration Instances" on page 3.

Configuration Properties: Concepts

What Is SAS Viya Configuration?

From SAS Environment Manager, you can manage the configuration needs of the various SAS Viya services.

Configuration Components

A service's configuration consists of the following components:

- *configuration definition*: A schema that describes a type of configuration. You create configuration instances from a configuration definition. Some examples of configuration definitions are: jvm, spring, and sas.reportdata.

  Note: Configuration definitions that apply to one or a small set of services are referred to as service configuration definitions. System configuration definitions can apply to any service.

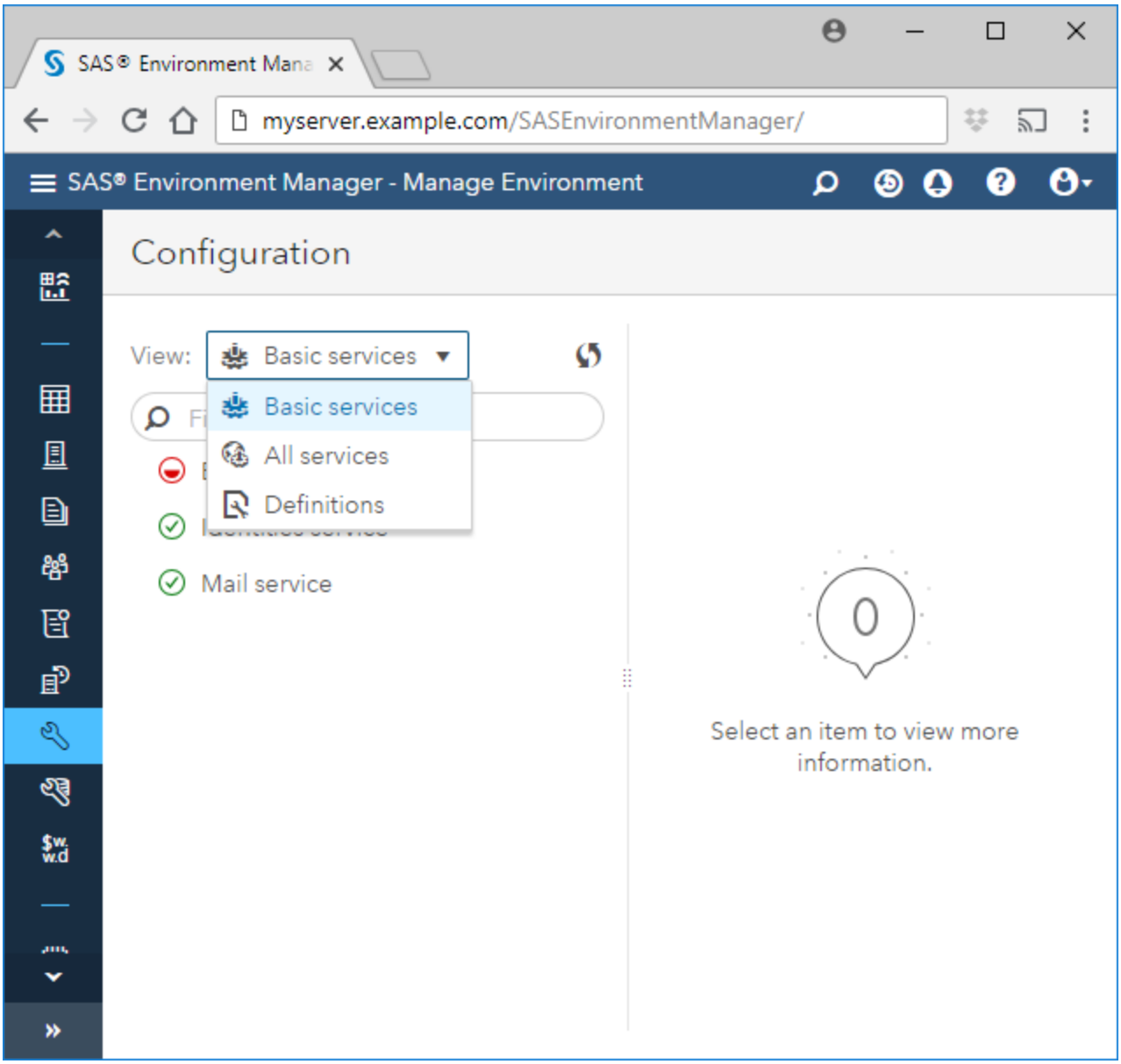
- *configuration instance*: A collection of name-value pairs (a property) that a service uses. (These name-value pairs can sometimes be nested.)

  Note: Certain configuration instances are required for a service to be able to run. See "Review Default Configuration Values" on page 4.

How Configuration Definitions and Instances Are Displayed

The Configuration window in SAS Environment Manager contains three views: **Basic services**, **All services**, and **Definitions**.

Figure 3 *Configuration Views in SAS Environment Manager*



The **All services** view lists all SAS Viya services that are currently deployed and those that an administrator has not manually stopped. A SAS Viya service can be affected by one or more configuration instances. Most services have a one-to-one relationship with a configuration instance. However, some services are associated with more than one configuration instance. It is important to note that some services do not have any configuration instances, but you can set configuration properties for any of these services.

The **Definitions** view lists all the SAS Viya configuration definitions.

The **Basic services** view contains those services with configuration properties for which SAS cannot create a reasonable default (for example, the machine name for your SMTP service). As an advanced topic, these configuration properties can be set in an initial deployment using sitedefault.yml. For more information, see "Automate Configuration Properties during Deployment (Ansible)" on page 10.

For a tenant within a multi-tenant environment, the services in the **Basic services** view can appear to be incomplete configuration instances (indicated by a half-filled red circle next to the configuration instance name). For security purposes, tenants cannot see configuration values that apply to other tenants. This means that the tenant administrator sees incomplete red icons, because administrators are not allowed to see a configuration

for an item that does not apply to them. For more information, see "Manage Tenants" in *SAS Viya Administration: Multi-tenancy*.

What Services Must Be Restarted?

On a non-cloud platform, such as native Linux, whenever a change is made to a Java virtual machine (JVM) configuration property (a Java option), any services that rely on that property must be restarted. For information about how to restart one or more services, see "General Servers and Services: Operate (Linux)" in *SAS Viya Administration: General Servers and Services*.

If you change configuration property values for any of the following services, you must restart the service:

- SAS Cache Locator
  - Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

    ```
    sudo systemctl restart sas-viya-cachelocator-default
    ```
  - Red Hat Enterprise Linux 6.x (or an equivalent distribution):

    ```
    sudo service sas-viya-cachelocator-default restart
    ```
- SAS Cache Server

  Important: If SAS Cache Server is restarted, then all services that are dependent on the cache server must be restarted. The list can vary depending on what SAS Viya offerings are deployed.

  > **TIP** To determine which services are dependent on SAS Cache Server, enter this command: `grep -H "Product-Name: Apache Geode" /opt/sas/viya/config/var/log/*/default/*.log`.

  - Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

    ```
    sudo systemctl restart sas-viya-cacheserver-default
    ```
  - Red Hat Enterprise Linux 6.x (or an equivalent distribution):

    ```
    sudo service sas-viya-cacheserver-default restart
    ```
- SAS Configuration Server (Consul)
  - Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

    ```
    sudo systemctl restart sas-viya-consul-default
    ```
  - Red Hat Enterprise Linux 6.x (or an equivalent distribution):

    ```
    sudo service sas-viya-consul-default restart
    ```
- SAS Message Broker (RabbitMQ)
  - Red Hat Enterprise Linux 7.x (or an equivalent distribution) and SUSE Linux Enterprise Server 12.x:

    ```
    sudo systemctl restart sas-viya-rabbitmq-server-default
    ```
  - Red Hat Enterprise Linux 6.x (or an equivalent distribution):

    ```
    sudo service sas-viya-rabbitmq-server-default restart
    ```
- SAS Infrastructure Data Server (PostgreSQL)

  For information, see "Operate a Cluster (Linux)" in *SAS Viya Administration: Infrastructure Servers*.

Note: You must be signed in to the machine where these services reside with sudo privileges to run these scripts.

See Also

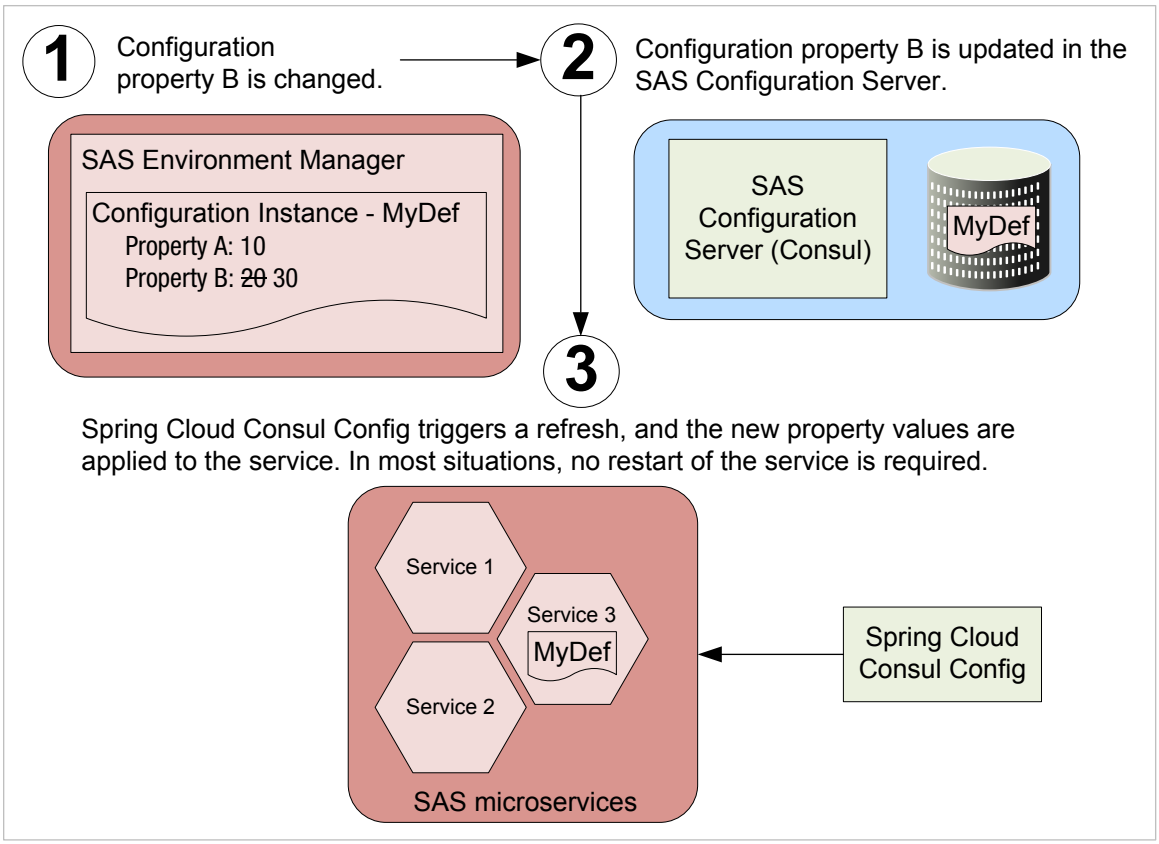
- “Start and Stop a Specific Server or Service” in *SAS Viya Administration: General Servers and Services*

How SAS Viya Configuration Works

Spring-Based Microservices

For SAS Viya, Spring-based microservices, property changes are made in SAS Environment Manager and stored in SAS Configuration Server. SAS Viya triggers a refresh, and the new property values are applied to the service. In most situations, no restart of the service is required.

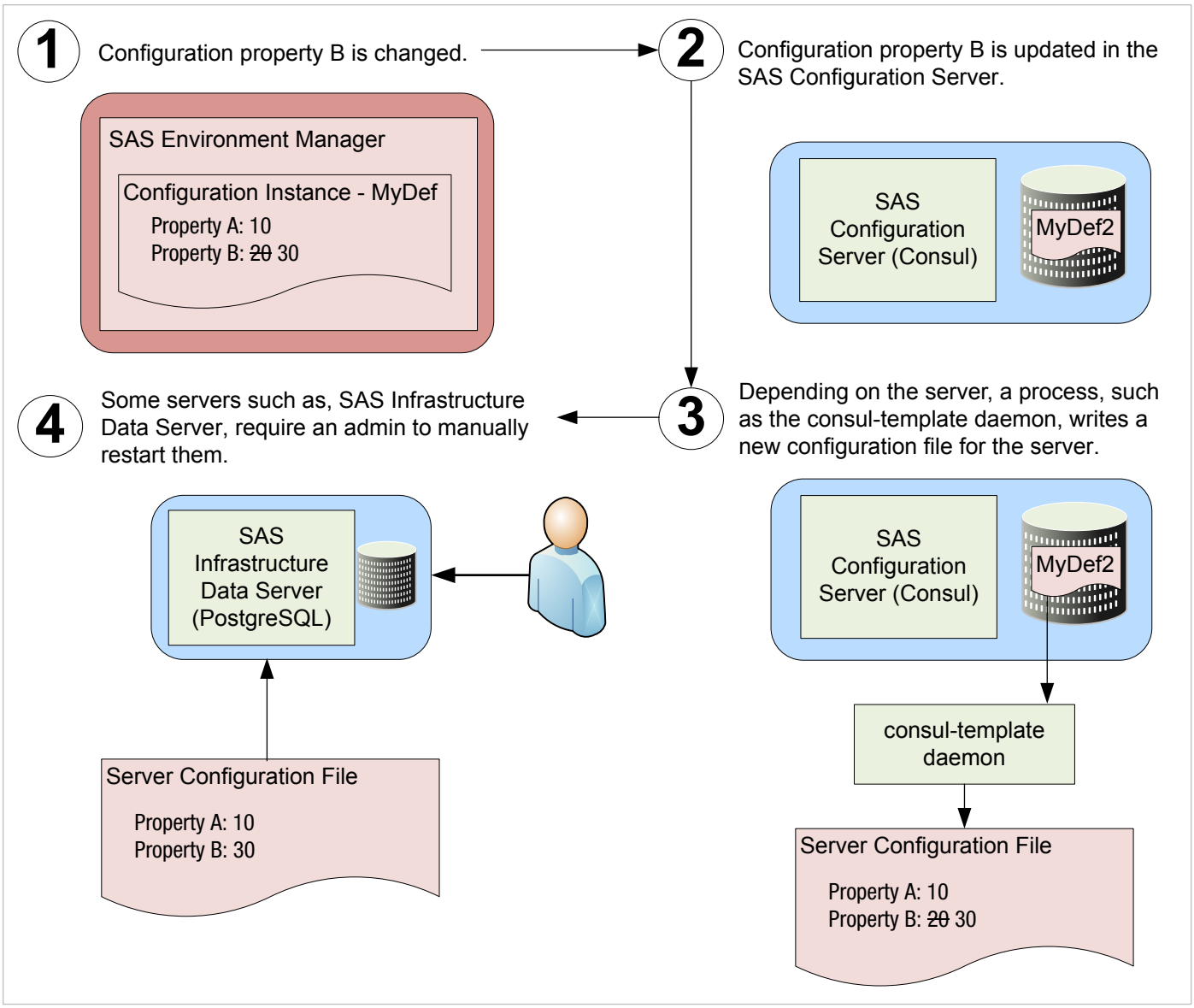
Figure 4 *How Configuration Properties Are Updated (Spring-Based Services)*



Non-Spring-Based Servers

For SAS Viya, non-Spring-based servers, property changes are made in SAS Environment Manager and stored in SAS Configuration Server. A tool, such as the consul-template daemon, extracts the configuration change from SAS Configuration Server and updates the appropriate service configuration file. Some servers, such as SAS Infrastructure Data Server, require you to manually restart them for their configuration changes to take effect.

Figure 5 *How Configuration Properties Are Updated (Non-Spring-Based Servers)*



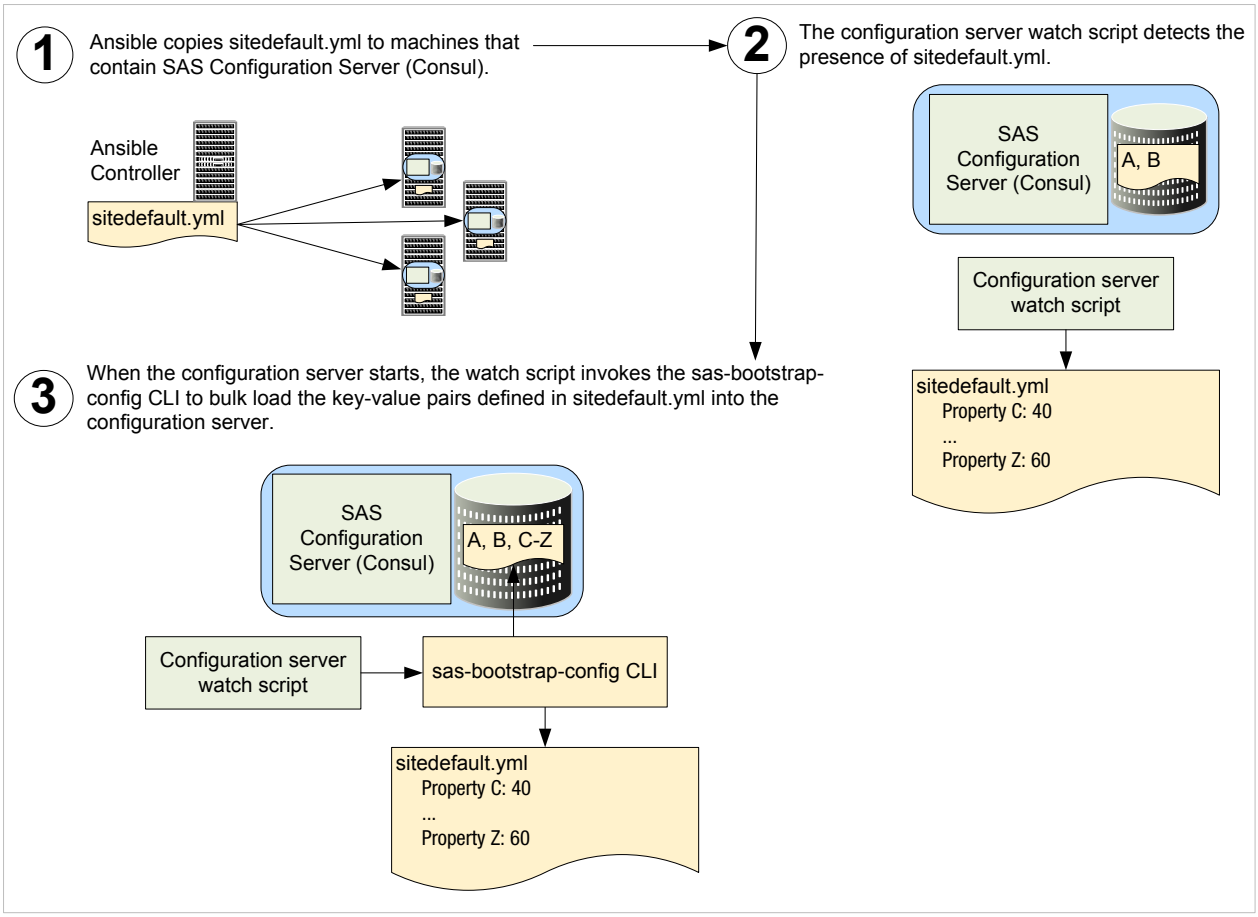
Bulk Loading of Configuration Values (sitedefault.yml)

You can deploy SAS Viya with configuration values that are customized for your site by running your Ansible playbook with sitedefault.yml. When sitedefault.yml is present in the playbook `roles/consul/files` directory, Ansible copies it to the machines that contain the SAS Configuration Server (Consul). When the configuration server starts, the watch script invokes the sas-bootstrap-config CLI to bulk load the key-value pairs that are defined in sitedefault.yml.

For more information, see "Automate Configuration Properties during Deployment (Ansible)" on page 10.

Note: The sas-bootstrap-config CLI uses a *check and set policy*. If a property currently exists in the configuration server, the CLI does not update the property. Therefore, it is extremely important that the initial values applied with sitedefault.yml are correct. After you set a value with sitedefault.yml, you cannot re-run sitedefault.yml to change that value. You can re-run sitedefault.yml only to set properties that have not already been set. To change properties set with sitedefault.yml, you must use the sas-bootstrap-config CLI directly, or use another administration interface, such as SAS Environment Manager.

Figure 6 How Configuration Properties Are Updated (Non-Spring-Based Services)



SAS Studio 5.x and 4.x Administration Differences

Overview

SAS Viya includes two releases of SAS Studio:

- SAS Studio 4.x

  the traditional version that has been available since the first release of SAS Viya.

- SAS Studio 5.x

  the new, microservices-based version, with a different interface. Studio 5.x integrates with other SAS Viya components (such as SAS Drive, Launcher Server, and Compute Server).

This section provides information about administration differences between the two versions of SAS Studio. For an overview of SAS Studio 5.x functionality, see "What's New in SAS Studio 5.1" in *SAS Studio: User's Guide*.

Deployment Environments

SAS Studio 4.x is used primarily in programming-only deployments. SAS Studio 4.x is an embedded web application that does not rely on external services. Therefore, SAS Studio 4.x has relatively few dependencies, such as a SAS Object Spawner and SAS Workspace Server.

SAS Studio 5.x is itself a microservice and provides functionality by relying on SAS Configuration Server, SAS Message Broker, SAS Infrastructure Data Server, and other microservices. For this reason, SAS Studio 5.x is available only in full deployments of SAS Viya.

Authentication Differences

SAS Studio 4.x uses host authentication such as Pluggable Authentication Modules (PAM) and Integrated Windows Authentication (IWA) to authenticate users. For more information, see "Authentication for Programming Interfaces" in *SAS Viya Administration: Authentication*.

SAS Studio 5.x authentication fully integrates with the features provided by SAS Logon Manager, and provides more authentication possibilities, such as the following:

- LDAP provider

  standard user name and password form.
- Kerberos or Integrated Windows Authentication (IWA)

  single sign-on from the client host to the visual interfaces.
- Security Assertion Markup Language (SAML) provider

  single sign-on from third-party provider.
- OAuth and OpenID connect provider

  single sign-on from third-party provider.
- Pluggable Authentication Modules (PAM)

  multi-factor authentication via third-party tools.
- SAS 9.4

  single sign-on from SAS 9.4.

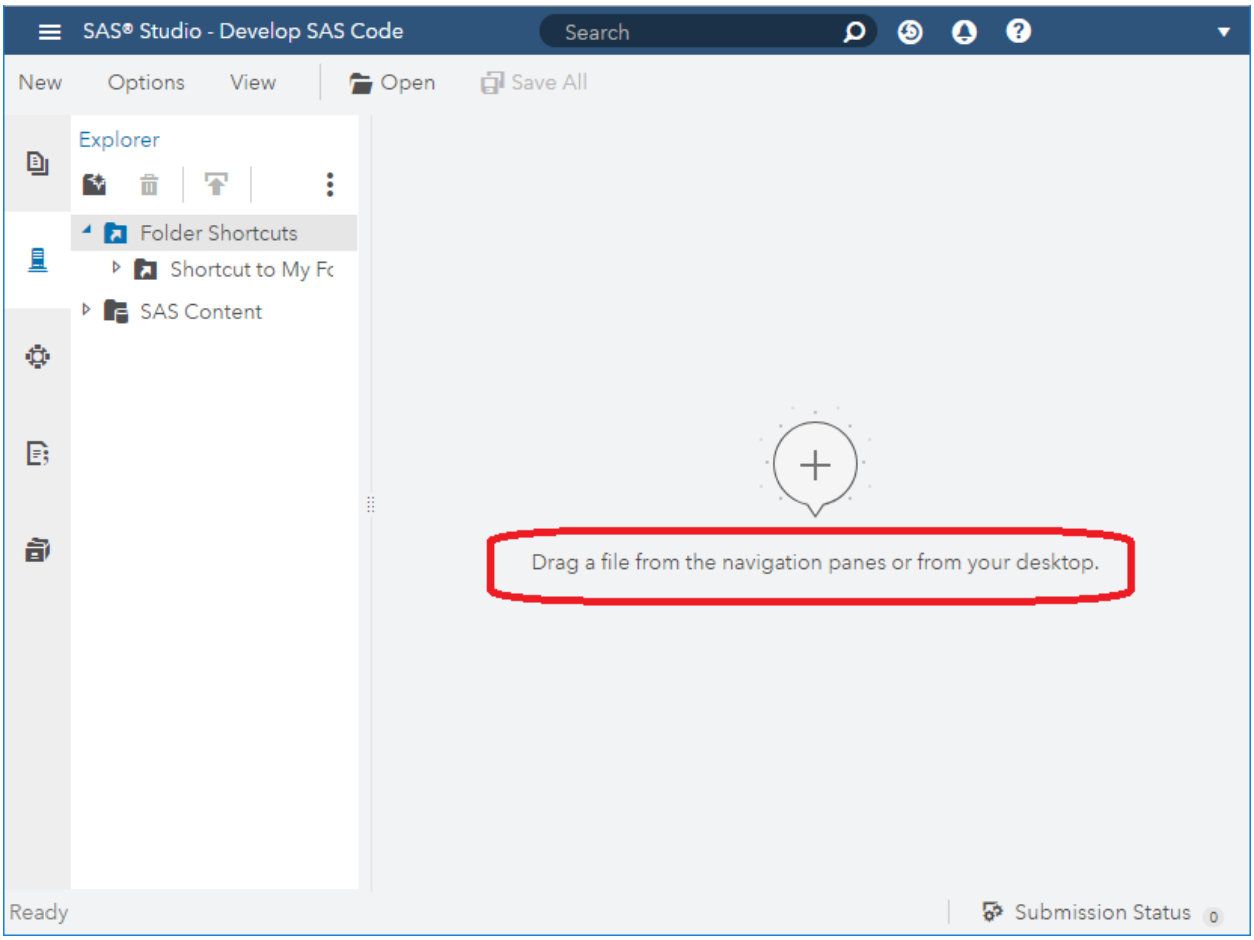
For more information, see "Authentication for Visual Interfaces" in *SAS Viya Administration: Authentication*.

Accessing SAS Content

Microservices are stateless and do not directly consume resources exposed by the underlying operating system, including the file system. They are designed to be running in a cloud or a containerized environment that might not even have a file system. SAS Studio 5.x follows the same pattern: when you open or save a file, you do not have any access to the back-end file system. So how do you access your programs? There are at least two ways.

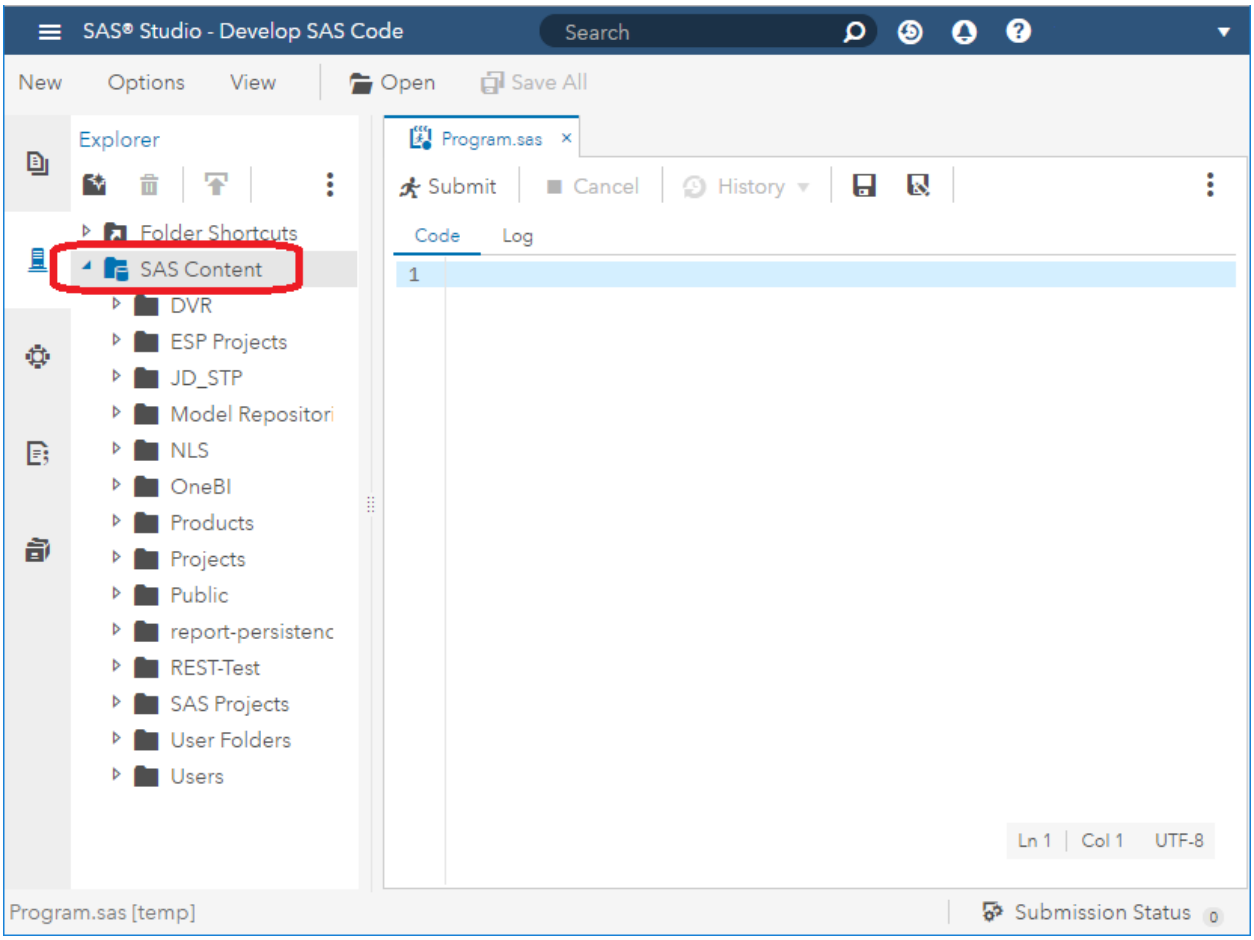
- Drag and drop:

  If your program file is available in your client, you can simply drag and drop it in the SAS Studio window. SAS Studio automatically opens the file.



- **SAS Content** folder:

  All files, including program and data, can be read from and written to folders inside **SAS Content**. **SAS Content** is available from the **Explorer** pane.



SAS Content is shared across all SAS Viya applications and managed by the SAS Drive web application.

Important:

Although any file can be written to or uploaded to folders in **SAS Content**, you cannot access data (.sas7bdat files) from SAS Content using libraries (because libraries cannot be assigned to **SAS Content** folders). Only data that can be read using a fileref can be read from **SAS Content**: CSV, TXT, XLSX, SAS, CTM (tasks), and CTK (task templates). For more information, see "Working with Data" in *SAS Studio: User's Guide*.

Improved Recovery

Previous releases of SAS Studio can be clustered for scalability and high availability, providing multiple instances of the web application. However, before SAS Studio 5.x, end-user sessions were bound to a specific instance known as a "sticky session."

To understand the disadvantages of a sticky session, consider the following example. Suppose that you have two SAS Studio 4.x instances running on server1 and server2. When you sign in to SAS Studio 4.x, the front-end load balancer directs you to server2. Until you sign out, your session remains on server2. If server2 abnormally shuts down—even though server1 is unaffected—your session is lost and you must sign out, possibly losing your work. When you sign in again, the load balancer redirects you to the available server, in this case, server1.

There is also another, more subtle issue. Each instance of SAS Studio 4.x can access only an object spawner and workspace servers running locally on the same machine. Suppose that all SAS Studio 4.x instances are running fine, but the object spawner on server2 goes down. Even if your SAS Studio 4.x session is unaffected, you are unable to perform any work. If you decide to sign out and back in, there is a chance that you will again be routed to server2, because the load balancer continues to see SAS Studio 4.x running on server2.

SAS Studio 5.x solves all these issues. Because it is a stateless microservice, Studio 5.x end-user sessions are not bound to any specific instance. If one instance stops abnormally, SAS Studio 5.x continues working using another microservice instance. Also, to run SAS code, SAS Studio 5.x does not use an object spawner and workspace server. Instead, Studio 5.x submits code to a compute server started by a launcher server. The compute and launcher server infrastructure ensures that there are no dependencies to a specific server machine. Therefore, SAS Studio 5.x can use a launcher and compute server located on any machine.

Architectural Comparison

To understand the architectures of SAS Studio 4.x and 5.x compare these diagrams:

- full deployment (SAS Studio 5.x)
- programming-only deployment (SAS Studio 4.x)

Configuration Properties: Troubleshooting

Service fails to start

Explanation:

A service might fail to start for many reasons. One error that can cause start failure is if you recently modified a configuration instance on which the service depends, and either the property name is misspelled or its value is incorrect.

Resolution:

Check any recently modified configuration instances, looking for a property misspelling or incorrect value. To do this, from the Configuration window in SAS Environment Manager, choose **Definitions**. Select the name of the configuration definition that you modified and look for any problems.

Configuration Properties: Reference (Services)

Application Registry Service

The Application Registry service registers applications to enable integration with SAS Drive 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.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

Audit Service

The Audit service provides a framework for reporting on audit events.

sas.audit.archive
The set of properties that are used to archive audit records.

enabled
Enables archiving of audit records.

batchSize
The number of audit records to process in a single batch during an archive request.

scanSchedule
The schedule that determines when an archive request is initiated.

localRetention
The retention period for persisting audit records within the service.

storageType
The external storage mechanism to use for archiving audit records. Must be set to 'none' or 'local'.

storage.local.destination
The file location to use when 'storageType' is set to 'local'.

sas.audit.record
The set of properties that are used to control how audit events are recorded.

type
Customizable properties to control which type of audit events are recorded.

application
Customizable properties to control which application-specific audit events are recorded.

Authorization Service

The Authorization service provides the general authorization system.

sas.authorization
The set of configuration properties for the Authorization service.

maxAncestryCacheSize
Specifies the maximum number of ancestors to cache per object. The default value is `1000`. The cache enhances performance in container-based inheritance.

reshareEnabled
Specifies whether resharing is enabled. When set to `Off`, prevents downstream sharing by users who lack Secure access to the object that is being shared. The default value is `On`. See "Sharing: Details for Administrators" in *SAS Viya Administration: General Authorization*.

Note: If sharingEnabled is `Off`, setting this property has no effect.

Note: You cannot enable or disable downstream sharing on a per-object or per-identity basis. The ability to perform downstream sharing is controlled only by this deployment-wide configuration property.

sharingEnabled
Specifies whether sharing is enabled. When set to `Off`, prevents all sharing. The default value is `On`. See "Sharing: Details for Administrators" in *SAS Viya Administration: General Authorization*.

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.

warnOnCycles (a supplemental property.)
Prevents cyclic warnings from being written to the authorization service log, if set to **`false`**. The default value is **`true`**. An administrator might temporarily set this property to **`false`** if cyclic warnings are causing the log to grow rapidly. After cyclic rules are corrected, set this property back to **`true`**.

Backup Service

The Backup service manages the backup and recovery of configuration information and user-created content in a SAS deployment.

sas.deploymentbackup
The set of configuration properties for the Backup service.

agentType
Important: This property is deprecated. Do not use this property.

The type of communication (messaging or SSH) that is used between the Backup service and the Backup agent.

jobTimeout
The number of minutes a backup job or a restore job is allowed to run before the job is marked 'Failed.'

retentionPeriod
The number of days that backups are stored before they are removed from the backup vault.

scheduledBackupAllowed
Important: This property is deprecated. Do not use this property.

Allows scheduled backups to run. In this release, the default value is false and cannot be changed.

sharedVault
A network location where all the backups are stored. This location should be accessible to the user identity installing deploymentBackup and backup-agent services.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

vaultLocation
Important: This property is deprecated. Do not use this property.

The location where all backups are stored. In a multi-machine deployment, the install user must have Read and Write permissions on every machine.

Cache Services

The Cache services provide other microservices the ability to distribute cached data across instances. The Cache services consist of both a Cache Locator and a Cache Server.

sas.cache.default

Important: The configuration definition, `sas.cache.default`, has been deprecated. Use instead, `sas.cache.config`.

sas.cache.config

The set of properties that provide customization for caching.

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 `0` (zero) 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.

criticalHeapPercentage
: The percentage of Java old generation heap. When reached, prevents Write operations of cached data from occurring.

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 it can be a shared resource.

disableAutoReconnect
: Disables the ability of a system member to reconnect and re-initialize after it has been forced out of the distributed system.

diskStoreDir
: The directory used when region data is overflowed to disk. This directory can be local and unique to the member or can be a shared resource.

diskStoreOpLogSize
: The maximum size for a region overflow file when region data is overflowed to disk. The value is specified in Megabytes.

diskStoreSize
: The maximum amount of data on disk when region data is overflowed. The value is specified in megabytes.

distributedCache
: Specifies that the cache should be distributed. Valid values are **`true`** or **`false`**. SAS Cache Server requires the value **`true`**'.

enableNetworkPartitionDetection
: 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.

evictionHeapPercentage
: The percentage of Java old generation heap when reached, causes cached data to overflow to disk.

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 `0` (zero) 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
: Indicates 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.

maxConnections
: The maximum number of connections to pool when connected to a cache server.

membershipPortRange
: The port range used when selecting ephemeral ports for members of the distributed system. Values are **`32768`** to **`61000`**.

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

mode
The mode of operation to use when connecting to the cache servers. Valid values are **client** or **local**. SAS Cache Server requires **local**.

overflowEnabled
Allows cached data to overflow to disk in low memory situations.

persistentEnabled
Allows cached data to persist to disk.

pingInterval
The ping interval for the cache client to check the availability of servers in milliseconds.

retryAttempts
The number of retry attempts for operations if a time-out or exception occurs.

subscriptionEnabled
Configures the client to register with the cache server for subscription events.

tcpPort
The TCP port a member of the distributed system listens on for cache communications.

Cache Locator Service

The Cache Locator service provides discovery information to SAS Viya microservices for the purpose of forming a distributed data cache. SAS Cache Locator is based on the open-source Apache Geode project.

sas.cache.locator
The set of properties that provide customization for the Cache Locator service.

host
The host where the service is deployed.

hostnameForClients
The external host name of the cache locator if different from the local bind address or host name.

port
The port registered for the cachelocator-listener.

retryCount
The number of attempts the service makes to register the cachelocator-listener.

retryPeriod
The amount of time between registration attempts for the cachelocator-listener.

timeout
The amount of time this service waits to start the locator process.

timeoutInterval
The amount of time between attempts checking for the start of the locator process.

Cache Server Service

The Cache Server service hosts long-lived data regions (a cache) and serves the contents to SAS Viya microservices. Like SAS Cache Locator, SAS Cache Server is based on the open-source Apache Geode project.

sas.cache.server

The set of properties that provide customization for the cache server.

autoStartup
: Specifies whether the cache server should be started automatically on start-up.

host
: The host where the service is deployed.

hostnameForClients
: The external host name of the cache server if different from the local bind address or host name.

maxTimeBetweenPings
: The maximum time in milliseconds between messages or a ping from a cache client.

port
: The port registered for the cache server.

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.

  Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

sas.casmanagement.global

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

The sas.casmanagement.global configuration instance applies to all SAS Viya servers and services (global).

- 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.

Compute Service

The Compute service enables clients to submit SAS programs and stored procedures in the form of jobs for processing. The SAS Compute Server implements the Compute service. See "SAS Compute Server and Compute Service" in *SAS Viya Administration: Programming Run-Time Servers*.

sas.compute

The set of properties used to configure the compute and related servers.

domain.default
: The default authentication domain to use for looking up host credentials.

kerberos.enabled
: Authenticate to compute servers using Kerberos.

serviceAccount.default:
: The default service account that should be used to run jobs on the host.

Notifications Service

The Notifications service stores and retrieves system and application event notifications. There are two types of notifications, subscription-based (also known as opt-in notifications) and directed (explicit notifications).

You can manage subscription-based notifications using the Notifications service. Directed notifications are managed by the SAS applications that use them (such as SAS Workflow Manager).

sas.notifications

The set of properties used to control subscription-based (opt-in) notifications.

enabled
: Enables opt-in notifications on all channels.

  Important: Turning off `sas.notifications.enabled`, turns off all subscription-based notification channels, such as notifications in SAS Viya web applications' banner and email notifications.

sas.notifications.delivery

The set of properties used to control subscription-based (opt-in) notifications on specific delivery channels.

mail.enabled
: Enables opt-in notification delivery to the email channel.

  Important: To enable subscription-based notifications to email, `sas.notifications.enabled` must be on (TRUE).

CAS Proxy Service

The set of configuration properties for the CAS Proxy service.

jobExecutionProvider

Configurable values for the CAS language (CASL) job execution provider job.

caslJESExpiresAfter
The amount of time after job completion before the job execution service deletes the job. Specify time in W3C XML duration format (for example, PT5M = 5 minutes). A null value indicates that job is not deleted.

caslJESHeartbeatInterval
The heartbeat value (in seconds) to use with a CASL job execution provider job. A zero or negative value indicates that the heartbeat is not checked.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

Configuration Service

The Configuration service manages changes to the configuration of services. See "Configuration Properties: Concepts" on page 11.

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.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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*.

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.

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. For detailed information about this functionality, see *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.

Important: The set of properties for `sas.identities.providers.ldap` are global settings that apply to all LDAP configurations. For multi-tenancy, these global settings apply to the provider and all tenants.

membershipCacheRefreshInterval
Specifies the interval that is used to refresh the membership cache for the LDAP provider. The default value is **`6h`**.

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`**.

primaryGroupMembershipEnabled
Enables processing of primary group memberships (used only for posixGroup schema).

sas.identities.providers.ldap.connection

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

anonymousBind
Defines whether Read-Only operations are performed using an anonymous (unauthenticated) context.

customEnvironmentProperties
The set of user-added, advanced properties for configuring the LDAP client environment.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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

password
The password for logging on to the LDAP server. If **anonymousBind** is enabled, specify a value of `none`.

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.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.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.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.testOnReturn
Enables validation of objects before they are returned to the pool. This option is disabled by default.

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.

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

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 TLS (LDAPS), port values are set to 636.

startTLS.mode
When using StartTLS, which mode to enable. The possible values are none (default) or simple.

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`). If **anonymousBind** is enabled, specify a value of **`none`**.

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.

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
accountId	The field in the LDAP provider that is used to populate the group's ID.	sAMAccountName
createdDate	The field in the LDAP provider that is used to populate the group's account created date.	whenCreated
description	The field in the LDAP provider that is used to populate the group's description.	description
member	The field in the LDAP provider that is used to populate the members of the group.	member
memberOf	The field in the LDAP provider that is used to populate the groups that this group is a member of. Set `memberOf` to **`none`** if the LDAP group attribute is not a fully qualified DN value.	memberOf
modifiedDate	The field in the LDAP provider that is used to populate the date on which the group's account was last modified.	whenChanged
name	The field in the LDAP provider that is used to populate the group's name.	displayName
objectClass	The object class value to use when loading groups.	group

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.

Note: The Identities service does not process referrals.

baseDN
The point from which the LDAP server searches for groups (for example, `ou=groups,dc=example,dn=com`).

distinguishedName
The field in the LDAP provider that is used to populate the group's distinguished name value.

Note: If your LDAP server does not support an explicit distinguished name attribute (for example, OpenLDAP), you must set this property to `none`.

objectFilter
: 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.

searchFilter
: The filter that is used to find a group account. The default filter is `${sas.identities.providers.ldap.group.accountId}={0}`.

sas.identities.providers.ldap.user (Field Mappings)

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.

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
accountId	The field in the LDAP provider that is used to populate the user's ID.	sAMAccountName
address.country	The field in the LDAP provider that is used to populate the user's country.	co
address.locality	The field in the LDAP provider that is used to populate the user's city.	l
address.postalCode	The field in the LDAP provider that is used to populate the user's postal code.	postalCode
address.region	The field in the LDAP provider that is used to populate the user's region or state.	region
address.street	The field in the LDAP provider that is used to populate the user's street address.	street
createdDate	The field in the LDAP provider that is used to populate the user's account created date.	whenCreated
description	The field in the LDAP provider that is used to populate the user's description.	description
emailAddress.other	The field in the LDAP provider that is used to populate the user's alternate email address.	otherMailbox
emailAddress.work	The field in the LDAP provider that is used to populate the user's work email address.	mail

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
emailAddress.sms	The field in the LDAP provider that is used to populate the user's SMS email address.	
memberOf	The field in the LDAP provider that is used to populate the groups that this user is a member of.	memberOf
modifiedDate	The field in the LDAP provider that is used to populate the date on which the user's account was last modified.	whenChanged
name	The field in the LDAP provider that is used to populate the user's name.	displayName
objectClass	The type of user objects that are being searched for.	organizationalPerson
phone.business	The field in the LDAP provider that is used to populate the user's work phone number.	telephoneNumber
phone.businessFax	The field in the LDAP provider that is used to populate the user's work fax number.	facsimileTelephoneNumber
phone.home	The field in the LDAP provider that is used to populate the user's home phone number.	homePhone
phone.mobile	The field in the LDAP provider that is used to populate the user's mobile phone number.	mobile
phone.pager	The field in the LDAP provider that is used to populate the user's pager number.	pager
title	The field in the LDAP provider that is used to populate the user's title.	title

sas.identities.providers.ldap.tenancy

The set of specific, multi-tenancy properties that can be used when implementing an LDAP provider.

Important: The purpose of `sas.identities.providers.ldap.tenancy` is for a default tenant configuration for a single, shared LDAP server. These properties are not used when customizing the structures for each tenant.

groupRdn
: The relative distinguished name group (RDN) value.

tenantKey
: The default value (OU) for tenantKey.

userRdn
: The relative distinguished name user (RDN) value.

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.

Note: The Identities service does not process referrals.

baseDN
: The point from which the LDAP server searches for users.

distinguishedName
: The field in the LDAP provider that is used to populate the user's distinguished name value.

  Note: If your LDAP server does not support an explicit distinguished name attribute (for example, OpenLDAP), you must set this property to `none`.

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}`.

Mail Service

The Mail service provides a client the ability to send email to a configured SMTP server using a REST API.

For more information, see "Configure the Connection to the Mail Service" in *SAS Viya for Linux: Deployment Guide*.

sas.mail
: The set of configuration properties for the Mail service.

  allowAllSenders
  Provides the ability to override restriction on the 'from' mail address allowed to send mail.

  fromAddress
  Default 'from' mail address to use when mail is sent directly from a service. (The default is `noreplies@company.com`.)

  fromPersonalName
  Default personal name to use when mail is sent directly from a service. (The default is `Service`.)

  host
  The mail server host (machine).

password
The optional password for connecting to the mail server.

port
The mail server port. (The default is 25.)

properties
Optional properties set on the remote mail server.

sizeLimit
The maximum size of mail sent to the configured mail server (in megabytes).

username
The optional user name for connecting to the mail server.

Maps Service

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

SAS Viya supports several third-party map services. Whether the map service uses HTTP or secured HTTP (HTTPS), depends on the following:

- Open Street Map

  Controlled by the `defaultOSMCommunicationProtocol` configuration property.

- ArcGIS Online

  Controlled by SAS Visual Analytics, which always uses secure HTTP (HTTPS).

- local Esri server

  Controlled by the URL that is entered for the server *protocol*://*host-name*:*port*/*path*. The protocol that is entered is used.

  Important: SAS Viya currently supports only token-based authentication for Esri. For example, an Esri server configured for Integrated Windows Authentication (IWA) is incompatible with SAS Viya.

sas.maps
The set of configuration properties for the Maps service.

defaultOSMCommunicationProtocol
The protocol (HTTP, HTTPS) that is used for the default Open Street Map servers.

localEsriServicesRequiresAuthentication
Indicates that the local Esri map services URL requires an authentication token for access.

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/rest/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 27.

supplementalProperties
The set of user-added, advanced properties.

> **Note:** Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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.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.

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

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.

Report Data Service

The Report Data service retrieves data from reports.

sas.reportdata.system
The set of system configuration properties for the Report Data service.

Note: In a multi-tenant configuration, sas.reportdata.system properties apply to all tenants.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

executorExpirationIntervalMinutes
The number of minutes before inactive data executor sessions are closed if they have no active queries.

executorForceExpirationIntervalMinutes
The number of minutes before inactive data executor sessions are forced closed even if they have active queries.

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

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.

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

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

sas.reportdata.properties
The set of configuration properties for the Report Data service.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

comparisonEpsilon
The number in E notation that is the variability allowed when comparing floating point numbers for equality.

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

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

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

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.

ignoreMissingValuesInCountDistinct
Ignore missing values in count distinct.

maxTiesToIncludeOnRank
The maximum number of ties allowed for a rank.

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.

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

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

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

A map of the maximum result rows values for the supported visual types.

maxRowsLookup.bubble
The maximum result rows for a bubble visual.

maxRowsLookup.buttonBar
The maximum result rows for a button bar visual.

maxRowsLookup.crossTab
The maximum result rows for a multidimensional table visual.

maxRowsLookup.customContent
The maximum result rows for a custom content.

maxRowsLookup.dropdown
The maximum result rows for a drop-down control.

maxRowsLookup.dualAxisTimeSeries
The maximum result rows for a dual axis time series visual.

maxRowsLookup.geoBubble
The maximum result rows for a geo bubble visual.

maxRowsLookup.geoContour
The maximum number of result rows for a geo contour visual.

maxRowsLookup.geoHeatmap
The maximum result rows for a geo heat map visual.

maxRowsLookup.geoRegion
The maximum result rows for a geo region visual.

maxRowsLookup.geoScatter
The maximum result rows for a geo scatter visual.

maxRowsLookup.graphDefault
The maximum result rows for a default graph visual.

maxRowsLookup.heatbox
The maximum result rows for a heat box visual.

maxRowsLookup.heatmap
The maximum result rows for a heat map visual.

maxRowsLookup.kpi
The maximum result rows for a kpi visual.

maxRowsLookup.list
The maximum result rows for a list visual.

maxRowsLookup.listTable
The maximum result rows for a list table visual.

maxRowsLookup.scatter
The maximum result rows for a scatter visual.

maxRowsLookup.textInput
The maximum result rows for a text input control.

maxRowsLookup.timeSeries
The maximum result rows for a time series visual.

maxRowsLookup.treeMap
The maximum result rows for a treemap visual.

maxRowsLookup.wordCloud
The maximum result rows for a word cloud visual.

sas.reportdata.debug
The set of debug configuration properties for the Report Data service.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

Report Packages Service

The Report Packages service executes reports to generate corresponding *report packages*. A 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.system
The set of system configuration properties for the Report Packages service.

Note: In a multi-tenant configuration, sas.reportpackages.system properties apply to all tenants.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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

packageExpirationTime
The amount of time in seconds after which the report package expires from the cache.

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

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

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

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

highContrastTheme
The name of the theme to be used when reports are displayed in high contrast.

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 report packages that can be generated concurrently per user.

subscribeConcurrentRequestLimitGuest
The maximum number of report packages that can be generated concurrently for the Guest user.

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

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

Report Renderer Service

The Report Renderer service creates PDF documents from report packages.

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

Note: In a multi-tenant configuration, sas.reportrenderer.system properties apply to all tenants.

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

workingDirectory
Override the working directory used for building rendered reports.

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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

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

footerContentFormatted
The HTML formatted footer to be included on each PDF rendered page.

webContentRendererLink
The URL to a service that is set up to provide images for printing of web content.

Note: In order for web content to be included when you print a report, you must set up an application to provide images for the web content. A third-party tool such as Rendertron (https://github.com/GoogleChrome/rendertron) is one option that can be installed and configured to generate such images dynamically using Headless Chrome.

The Report Renderer service supports URLs with the following substitution tokens: `__WebContentURL__`, `__ImageWidth__`, and `__ImageHeight__`.

`_WebContentURL_` is a token that the Report Renderer service substitutes with the web content URL found in the report. The Report Renderer service substitutes `__ImageWidth__`, and `__ImageHeight__` with the requested image size, as allocated by the report layout.

Note: `webContentRendererLink` supports only standard web content (that is, content that does not require data or authentication). For this reason, data-driven content and SAS stored processes are not supported and cannot be rendered when printing reports in this release.

You must add this domain to the cross-domain proxy whitelist using the `sas.crossdomainproxy.allowedDomains` property. For more information, see “sas.crossdomainproxy”.

Here is an example:

```
http://my-server.example.com:3000/screenshot/__WebContentURL__?
width=__ImageWidth_&height=__ImageHeight__
```

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

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

supplementalProperties
The set of user-added, advanced properties.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

Secret Manager Service

The Secret Manager service manages certificates generated by and stored by HashiCorp Vault. Vault provides a secure interface to secrets. These connections are secured by a Public Key Infrastructure (PKI) based on HashiCorp Vault, which is configured by SAS. The certificates are all signed by a Vault-generated root CA and intermediate certificate.

sas.vault

The set of configuration properties used to configure SAS Secret Manager (Vault).

Important: When modifying `sas.vault.*_ttl` properties, you must not violate precedence rules or SAS Secret Manager does not start. For more information, see "TTL Precedence Rules" in *SAS Viya Administration: Infrastructure Servers*.

certificate_role
Role of the certificates.

certificate_role_allow_any_name
The common name represents the name protected by the TLS certificate. Any name is allowed for the common name. The certificate is valid only if the requested host name matches the certificate common name. It consists of a single host name (for a single-name certificate) or a wildcard name (for a wildcard certificate, *.example.com).

certificate_role_key_bits
The key-length (in bits) of the certificate generated from Vault.

certificate_role_key_type
The certificate encryption algorithm. RSA or Elliptic-Curve (EC) can be specified.

certificate_role_key_usage
A comma-separated list that defines the allowed uses of a generated certificates. You can specify one or all of the following:

```
DigitalSignature,KeyAgreement,KeyEncipherment
```

certificate_role_max_ttl
The maximum length of time in hours that a Vault-issued certificate lasts before expiring. Must be greater than the **`certificate_role_ttl`**.

certificate_role_ttl
The default length of time that a Vault-issued certificate lasts before expiring.

intermediate_ca
An intermediate certificate is a subordinate certificate issued by the trusted root specifically to issue end-entity server certificates.

intermediate_ca_common_name
The common name (CN) for the Vault-issued intermediate certificate authority (CA) certificate. For SAS Viya, the name is SAS Viya intermediate CA. The CN identifies the host name associated with the certificate.

intermediate_ca_desc
The name for the Vault-issued intermediate CA certificates and SAS Viya Intermediate CA.

intermediate_ca_key_bits
The key-length (in bits) of the intermediate certificate generated from Vault.

intermediate_ca_max_ttl
The maximum length of time in hours that a Vault-issued intermediate certificate lasts before expiring. Must be greater than the **`intermediate_ca_ttl`**.

intermediate_ca_ttl
The default length of time that a Vault-issued intermediate certificate lasts before expiring.

root_ca
A public key certificate that identifies a root CA. A root certificate is the top-most certificate of the tree. The private key is used to sign other certificates.

root_ca_common_name
The common name for the Vault-issued root CA certificate and SAS Viya Root CA. The CN identifies the host name associated with the certificate.

root_ca_desc
The name for the Vault-issued Root CA and SAS Viya Root CA.

root_ca_key_bits
The key-length (in bits) of the root certificate generated from Vault.

root_ca_max_ttl
The maximum length of time in hours that a Vault-issued root CA certificate lasts before expiring. Must be greater than the **`root_ca_ttl`** value.

root_ca_ttl
The default length of time that a Vault-issued root CA certificate lasts before expiring.

systems
The time in hours that secrets and tokens are managed.

system_max_lease_ttl
The maximum amount of time (in hours) that Vault-issued secrets and tokens are valid. This value must be larger than the **`vault_token_default_lease_ttl`** value for the token configuration instance.

tokens
The time in hours that secrets and tokens are valid.

vault_token_default_lease_ttl
The default length of time (in hours) that Vault-issued tokens are valid.

Note: Changes to this value take effect after running the Ansible renewal playbook.

Configuration Properties: Reference (Applications)

SAS Data Explorer

SAS Data Explorer enables you to discover data and copy it to a CAS server.

sas.dataexplorer
The set of configuration properties for SAS Data Explorer.

casSessionImportNumNodes
The number of nodes on which to start CAS sessions when running import jobs. A value of `0` means all nodes.

casSessionInteractiveNumNodes
The number of nodes on which to start CAS sessions used for browsing data sources. A value of `0` means all nodes.

filterAvailableTab
Turn the toggle switch on to automatically populate the **Available** and **Data Sources** tabs with tables only from the default caslib for your site. The default is `off`.

By default, the **Available** and **Data Sources** tabs display all tables that have been loaded to memory, from any CAS server to which you have access. If some nodes are slow to respond to queries from the **Available** and **Data Sources** tabs, your browser might freeze while it is waiting for a response. If slow performance persists, an administrator can set an option so that the **Available** and **Data Sources** tabs are automatically populated with tables only from the default caslib for your site.

jobExecutionProvider
Configurable values for the SAS Data Explorer job execution provider job:

- **heartbeatInterval**. Default is `300`. The maximum number of items that can be submitted in a single import.
- **scheduledJobExpiresAfter**. Not set by default. The amount of time after a scheduled job completion before the job execution service deletes the job. Specify time in W3C XML duration format (for example, PT5M = 5 minutes). A null value indicates that job is not deleted.

maxImportQueueSize
The maximum number of items that can be imported with the **Import All** option on the **Import** tab. The default is `100`F.

availableTabEntryTimeoutMS
Specifies how long (in seconds) Data Explorer waits for a given library to report back its list of available tables. The default value is 10 seconds. If the request does not come back within the allotted time, the library is not included in the available tables list. This setting affects the list of available tables in the **Available** tab and the `Data Sources` tab.

supplementalProperties
The set of user-added, advanced properties.

> **Note:** Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

SAS Data Studio

SAS Data Studio provides a way for you to prepare data, including data transformations.

sas.datastudio
The set of configuration properties for SAS Data Studio.

casSessionNumNodes
The number of nodes on which to start a SAS Data Studio CAS session (0 means all nodes).

casSessionTimeout
The number of seconds to set as the CAS session time-out for sessions that SAS Data Studio creates.

interactiveJobExpiresAfter
The amount of time after an interactive data plan job completes before the job execution service deletes the job. Specify time in W3C XML duration format (for example, PT5M = 5 minutes). A null value indicates that job is not deleted.

saveTableJobExpiresAfter
The amount of time after a data plan job, which saves a table, completes before the job execution service deletes the job. Specify time in W3C XML duration format (for example, PT5M = 5 minutes). A null value indicates that job is not deleted.

SAS Infrastructure Data Server

SAS Infrastructure Data Server is based on PostgreSQL version 9 and is configured specifically to support SAS software. See "Overview" in *SAS Viya Administration: Infrastructure Servers*. Here is the list of SAS Infrastructure Data Server configuration definitions that consist of third-party PostgreSQL and pgpool-II configuration properties.

sas.dataserver.common

The set of properties that are common to a cluster (that is, both to pgpool-II and to PostgreSQL nodes).

For a list of the valid PostgreSQL property names (configuration parameters) and descriptions, see https://www.postgresql.org/docs/9.1/static/runtime-config.html.

sas.dataserver.conf

The set of properties that are used to set up the SAS Infrastructure Data Server node database configuration file, postgresql.conf.

For a list of the valid PostgreSQL property names (configuration parameters) and descriptions, see https://www.postgresql.org/docs/9.1/static/runtime-config.html.

sas.dataserver.hba

The set of properties that are used to set up the SAS Infrastructure Data Server node host-based authentication file, pg_hba.conf.

For a list of the valid PostgreSQL property names (authorization records) and descriptions, see https://www.postgresql.org/docs/9.1/static/auth-pg-hba-conf.html.

sas.dataserver.pool

The set of properties that are used to set up the pgpool-II node configuration file, pgpool.conf.

For a list of the valid pgpool-II property names (key-value pairs) and descriptions, see http://www.pgpool.net/docs/pgpool-II-3.5.4/doc/pgpool-en.html#pgpool_conf.

sas.dataserver.pool.hba

The set of properties that are used to set up the pgpool-II node host-based authentication file, pool_hba.conf.

For a list of the valid pgpool-II property names (key-value pairs) and descriptions, see http://www.pgpool.net/docs/pgpool-II-3.5.4/doc/pgpool-en.html#hba.

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:

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.

assumable
: Specifies groups that have an elevated level of access that the user must approve at sign-in in order to assume those groups in a session.

approvalExpirySeconds
: When approving or denying access to a third-party application, specifies the number of seconds that the approval or denial should be remembered.

groupLookupRequired
: Requires groups to be determined for authentication to succeed.

recursion.enabled
: Allows 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.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 (Linux)” in *SAS Viya Administration: General Servers and Services*.

reset.enabled
: Displays a password reset link for the initial user account at start-up.

user
: The user name for the initial user account.

passwordResetLifetime
: The number of milliseconds for which that 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.jasig.cas

The set of properties that are used to enable sign-ins using a Java in Administration Special Interest Group (Jasig) central authentication server (CAS) provider.

enabled
Enable sign-ins using CAS.

single.signOn.enabled
Enable single sign-on.

single.signOut.enabled
Local sign-out should sign user out of the CAS server also.

single.signOut.logoutParameterName
The parameter name that identities a back channel single logoff request from the CAS server.

keepalive.enabled
Keep the CAS server session active by obtaining proxy tickets.

keepalive.proxyTicketUrl
The URL to obtain proxy tickets for keepalive purposes.

sasLogonUrl
The URL of the SAS Logon Manager (for example: https://sas.example.com/SASLogon).

casServerUrl
The URL for the CAS server.

service.authenticateAllArtifacts
Process all tickets, including proxy tickets.

service.sendRenew
Force the server to authenticate the user again, even if the user has previously authenticated.

service.artifactParameter
The request parameter to look for when attempting to see whether a ticket was sent from the server.

service.serviceParameter
The request parameter to send the server for this service.

validate.renew
Determine whether the ticket validation request should include a renew.

validate.encoding
Encoding of the ticket validation response from the server.

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

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

autoLink
Automatically open the link to this provider when the login page is displayed.

sas.logon.jwt

The set of properties that are used to configure how JSON web tokens are issued.

signingKey
Either a Base64-encoded RSA private key that is used to digitally sign tokens, or a simple passphrase for HMACs. Enter a value only if you want to override the system-generated RSA private key.

issuer.uri
The URI of the application, for the issuer claim in tokens (for example, https://example.com/SASLogon).

claims.exclude
The comma-separated list of claims that should be excluded from the JSON web token.

policy.accessTokenValiditySeconds
The default number of seconds that access tokens are valid for after being issued in the default zone.

policy.refreshTokenValiditySeconds
The default number of seconds that refresh tokens are valid for after being issued in the default zone.

policy.global.accessTokenValiditySeconds
The default number of seconds that access tokens are valid for after being issued in all zones.

policy.global.refreshTokenValiditySeconds
The default number of seconds that refresh tokens are valid for after being issued in all zones.

refresh.restrictGrant
Grant refresh tokens only to clients with a scope of refresh_token for offline access.

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 (Linux)" in *SAS Viya Administration: General Servers and Services*.

servicePrincipal
The name of the service principal in the keytab.

spn
The HTTP service principal name (SPN), if different than the principal name 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.

holdOnToGSSContext
Enables Kerberos delegation to SAS Cloud Analytic Services.

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

disableDelegationWarning
Disables the warning message displayed to users if they are unable to perform Kerberos credential delegation from their browser to the SAS Logon Manager.

sas.logon.oauth.providers.external_oauth

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 comma-delimited list of possible email address domains 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.pam

The set of properties that are used to enable sign-ins using PAM.

enabled
Enables sign-in using PAM.

serviceName
The service name in the PAM configuration.

sas.logon.provider.guest

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

Apply configuration only to this tenant (provider)
When this property is set to `off`, the configuration applies to all tenants, including the provider. Each tenant can override the configuration from within its own environment.

Note: **Apply configuration only to this tenant (provider)** is available only to provider tenants in a multi-tenant environment.

enabled
Enable anonymous guest access to web applications.

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 (Linux)” 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.

CAUTION! Do not modify setProxyParams. The value should remain off (false).

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.

signatureAlgorithm
The algorithm for SAML signatures. The accepted values are SHA1, SHA256, and SHA512.

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.external_saml

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.

addShadowUserOnLogin
Adds a local shadow user upon successful authentication. If set to `false`, users must be pre-created in the database to log on.

skipSslValidation
: Skips Transport Layer Security (TLS) validation of the certificate.

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

sas.logon.sas9

The set of properties that are used to enable sign-ins using SAS 9.4 and later.

autoLink
: Automatically open the link to SAS 9 when the login page is displayed.

enabled
: Enable sign-ins using SAS 9 credentials.

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

sas9LogonUrl
: The URL of the SAS 9 Logon Manager (for example, https://sas9.sas.example.com/SASLogon).

showLinkText
: Show the link text on the sign-in page.

single.signOn.enabled
: Redirect to SAS 9 for single sign-on.

single.signOut.enabled
: Local sign-out should sign user out of SAS 9 also.

viyaLogonUrl
: The URL of the SAS Viya Logon Manager (for example, https://viya.sas.example.com/SASLogon).

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.

sas.logon.tenancy

The set of properties that are used to configure multi-tenancy.

bootstrap.enabled
: Automatically configure identity zones and LDAP when tenants are onboarded or access policy is changed.

autoUpdateLdapConfiguration
: Automatically update all identity zones' LDAP configurations when the provider LDAP configuration is changed.

SAS Studio 5.x

SAS Studio is a development application for SAS that you access through your web browser. SAS Studio 5.x relies on the SAS Viya service layer.

Note: SAS Studio 5.x is installed with a *full deployment*.

sas.studiov
The set of configuration properties for SAS Studio 5.x.

allowDownload
Allow users to download data.

allowUpload
Allow users to upload data.

longPollingHoldTimeSeconds
The maximum number of seconds to wait for a message from the client.

maxUploadSize
The maximum file size (bytes) allowed for upload.

SAS Studio 4.x

SAS Studio is a development application for SAS that you access through your web browser. SAS Studio 4.x relies on an embedded web application that is part of SAS Viya.

Note: SAS Studio 4.x is installed with a *programming-only deployment*.

For more information, see "Update SAS Studio Configuration Properties" on page 8.

Table 1 *SAS Studio: Configuration Properties*

Property	Default Value	Description
sasstudio.appserver.https.keystorefile	(blank)	Specifies the keystore file to use for HTTPS.
sasstudio.appserver.https.keystorepass	(blank)	Specifies the keystore password to use for HTTPS.
sasstudio.appserver.https.port	(blank)	Specifies the port to use for HTTPS.
sasstudio.appserver.port	(blank)	Specifies the port to use for HTTP.
webdms.allowBackgroundSubmit	true	Specifies whether the **Background Submit** option is available when you right-click a .sas file in the navigation tree in the SAS Studio workspace.
webdms.allowFolderShortcuts	true	Specifies whether you can create folder shortcuts in the user interface.
webdms.batchSubmissionResultsRetentionPeriod	24	Specifies the number of hours to keep the output files from a background submission.
webdms.customPathRoot	(blank)	Specifies a path that determines the root node in the Folders tree. **Note:** You can use `<userid>` substitution for a directory path (for example, `/home/<userid>`).
webdms.defaultEncoding	UTF-8	Specifies the default SAS encoding method.

Property	Default Value	Description
webdms.defaultVVN	ANY	Specifies the default value for the VALIDVARNAME option.
webdms.globalSettings	`/opt/sas/spre/home/SASFoundation/GlobalStudioSettings`	Specifies the directory location for global XML files.
webdms.longPollingHoldTimeSeconds	30	Specifies the maximum number of seconds to wait for a message from the client.
webdms.maxNumActiveBatchSubmissions	3	Specifies the maximum number of active background jobs for the current SAS Studio user.
webdms.maxNumActiveBatchSubmissionsSystem	24	Specifies the maximum number of background jobs that can be submitted for a given instance of SAS Studio across all users.
webdms.maxSessionTimeoutInHours	1	Specifies the maximum number of hours a user can specify for the session time-out value in preferences.
webdms.maxUploadSize	10485760 (10MB)	Specifies the number of bytes allowed for file upload. **Note:** Large files can take a long time to load. If you have a large amount of content to upload, divide your content into smaller files if possible.
webdms.showSystemRoot	true	Specifies that the system root location be displayed in the Folders tree. **Note:** Set the value to false when the LOCKDOWN statement or the LOCKDOWN system option is used. For more information, see "References " in *SAS Viya Administration: Programming Run-Time Servers*.
webdms.studioDataParentDirectory	(blank)	Specifies the location of SAS Studio preferences, snippets, my tasks, and more. This preference is specific to the local computer. The default value is blank. An administrator must mount a shared location to access data from any workspace server session. **Note:** You can use `<userid>` substitution for a directory path (for example, `/home/<userid>`).

Property	Default Value	Description
webdms.workspaceServer.allowGetRecordCount	true	Specifies whether to retrieve all of the rows for database tables. If you set this property to `false`, performance improves, but you might not see all rows of the table. For example, for large tables, total rows and filtered rows appear as Unavailable in the user interface. If the table has fewer than 100 rows or you scroll to the last page of the table, the values for the total rows and filtered rows are shown.
webdms.workspaceServer.cacheTableRow	true	Specifies whether to cache the rows from database tables to improve performance. If you use caching, the row count could be wrong if you modify the table. You must click **Refresh** to remove the value from the cache and to force a re-query of the database. If correct row count is more important than performance improvement, set this property to `false` to disable caching.
webdms.workspaceServer.hostName	localhost	Specifies the host to use to connect to the workspace server.
webdms.workspaceServer.largeTableRows	50,000	Specifies the maximum number of rows to display in the table viewer. If the number of rows in the table is unknown or greater than the value specified for the `webdms.workspaceServer.largeTableRows` property, the following behavior occurs: - SAS Studio displays a warning that sorting could take a long time. - SAS Studio does not generate a list of distinct values to select from when SAS Studio filters the data.
webdms.workspaceServer.port	8591	Specifies the port to use to connect to the workspace server.

Configuration Properties: Reference (System)

Commons REST Client

The following are properties to configure the commons REST client library, a library that all SAS Viya microservices incorporate.

sas.commons.rest.client

The set of configuration properties for the commons REST client library.

Bypass HTTP proxy
: Enables requests to be routed directly to the service rather than through the HTTP proxy. Changing **Bypass HTTP proxy** requires you to restart all SAS Viya services.

Java Virtual Machine (JVM)

The set of properties (Java options) that are used to configure the Java Virtual Machine when it is launched. Each JVM property defined in SAS Environment Manager corresponds to a single Java option.

To define service or global options for the JVM, follow the steps listed in "Create Configuration Instances" on page 3.

Note: Creating or 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 (Linux)" in *SAS Viya Administration: General Servers and Services*.

When adding each JVM property, remember these guidelines:

- For the list of the valid Java options and descriptions, see http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html.
- The property name for each Java option that you add must start with the string, java_option_ (for example, java_option_xmx).
- The property value is a single Java command-line option (for example, -Xmx512m).
- When the property names match, Java options specified at the service level override global Java options.
- Matching a Java option's property name (with a value consisting of a zero-length string) is the only way to disable Java option values.
- There is no control over the order that the JVM processes Java options.

Note: If you are using SAS Studio version 4.x, you must modify the JVM options in these configuration files:

- **`/opt/sas/viya/home/SASStudio/bin/appserver.sh`**
- `/opt/sas/viya/home/SASStudio/bin/sas.sasstudio.host`

Security

The following are properties to configure web security.

sas.commons.web.security

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

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

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

x-content-type-options
: The string used for the X-Content-Type-Options header for unsecured endpoints.

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

x-frame-options
: The string used for the X-Frame-Options HTTP header. A restart is required to pick up changes to this property.

x-frame-options-enabled
Sends the X-Frame-Options header in HTTP responses. A restart is required to pick up changes to this property.

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

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

sas.commons.web.security.cors

The set of properties that are used to configure Cross-Origin Resource Sharing (CORS) security. By default, CORS is enabled. For more information about CORS, see CORS support in Spring Framework.

allowCredentials
Allows credentials to be used in cross-origin requests. By default, this property is set to **On**.

allowedHeaders
The comma-separated list of HTTP headers that are allowed, by default, in cross-origin requests. Specify an asterisk ('*') to match any header.

allowedOrigins
The comma-separated list of origins that are allowed by default. The list can contain regular expressions. Specify an asterisk ('*') to match any origin.

allowedMethods
The comma-separated list of HTTP methods that are allowed, by default, in cross-origin requests. Specify an asterisk ('*') to match any method.

sas.commons.web.security.csrf

The set of properties that are used to configure Cross-Site Request Forgery (CSRF) security. By default, CSRF is enabled. To disable it, create a new configuration for the security definition. Specify the property name as **`enable-csrf`** and the value as **`false`**. For more information, see "Create Configuration Instances" on page 3.

SAS Viya protects against CSRF using the following:

- Synchronizer Tokens: Randomly generated tokens that are associated with the user's current session. CSRF is checked only on requests with authenticated sessions, and is always skipped on GET, HEAD, TRACE, and OPTIONS requests. For more information, see Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet.
- Header Checking: A filter that checks that the HTTP Referer header has the host and port of the requested URI or matches an optional whitelist of URIs that is configured as a comma-separated list in the *sas.commons.web.security.csrf.allowedUris* property.

For more information about CSRF, see Common application properties.

allowedReferers
This property is currently not supported and should be left blank.

allowedUris
The comma-separated list of referer URIs that are allowed by default. The list must contain regular expressions.

failIfNoHeaders
Blocks requests if both the Origin and Referer headers are absent.

sas.security

The set of configuration properties that are used to configure security for SAS Viya servers and services. The sas.security configuration instance applies to all SAS Viya servers and services (global).

network.databaseTraffic.enabled
: Toggle security for database traffic.

network.sasData.enabled
: Toggle security for other SAS information.

network.serverControl.enabled
: Toggle security for serverControl.

network.web.enabled
: Toggle security for web-based traffic.

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 Common application properties.

CAUTION! When adding a property, be extremely careful. Entering the wrong property name or an invalid data type can cause SAS Viya to become inoperable.

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.

zones

The set of properties that are used to configure zone information for multi-tenancy. Modifying one of these property values requires you to restart the service.

internal.hostnames
: The comma-separated list of internal host names that are used to access the provider zone, or that are used in a subdomain to access other zones.

> **TIP** Be sure to specify the base host name without any tenant prefixes.

Configuration Properties: Interfaces

There are several interfaces that you can use to manage configuration properties for SAS Viya servers, services, and applications. The following table lists these interfaces and the shading indicates the relative amount of SAS Viya configuration that each covers:

Table 2 *Interfaces to CAS Administration*

◒	Ansible	A software orchestration tool that provides a straightforward approach to deploying and provisioning SAS Viya. You can set configuration property defaults at installation.
◕	SAS Environment Manager	A graphical enterprise web application that enables you to modify and view SAS Viya configuration properties.
●	Command-line interface	A command-line interface that enables you to manage SAS Viya configuration properties.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. Copyright © 2018, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. January 2020 3.4-P1:calconfig

THE POWER TO KNOW®