



SAS[®] Viya[®] 3.5 Administration: Authentication

3.5

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

Authentication: Overview	2
Authentication Options	2
Authentication: How To	3
Authentication Mechanisms	3
Session Management Using SAS Environment Manager	44
Additional Authentication Topics	46
Authentication: Concepts	57
Authentication Architecture	57
Authentication and SAS Viya Services	58
In-bound and Out-bound Authentication	58
Authentication Options	59
Concepts: Authentication Mechanisms	64
Additional Authentication Topics	81
Authentication: Guest Access (Linux)	83
About Guest Access	83
Enable Guest Access	84
Connect as Guest Users	86
Generate Custom Links to Reports	86
Disable Guest Access	87
Scenario: OIDC with Microsoft Entra ID (Linux Full Deployment)	89
Configure Microsoft Entra ID for OIDC	89
Configure OIDC Provider Properties for Microsoft Entra ID	90

Scenario: SAML with Microsoft Entra ID (Linux Full Deployment)	93
Configure Microsoft Entra ID for SAML	93
Configure SAML Provider Properties for Microsoft Entra ID	93
Configure the Enterprise Application in Microsoft Entra ID	97
Configure Cross-Origin Settings for SAML	98
Authentication: OIDC with ISAM Scenario (Linux Full Deployment)	99
Configure OIDC Provider Properties for ISAM	99
Configure OIDC Provider in ISAM	102
OIDC and ISAM	103
Authentication: OIDC with Okta Scenario (Linux Full Deployment)	105
Create the Web Application Using the Okta Admin Console	105
Configure OIDC Provider Properties for Okta	107
Authentication: Passwords	110
Update Account Passwords on Windows	110
Authentication: Reference	111
CAS Environment Variables for Clients	111
CAS Environment Variables for Administrators	112
Authentication: External Languages Package	114
Overview	114
External Languages Access Control Configuration	115
Sample Access Control File for the EXTLANG Package	119
Authentication: Troubleshooting	121

Authentication: Overview

SAS Viya user security comprises both user authentication and user identification. *Authentication* is the ability to prove that a user or application is genuinely who that person or what that application claims to be. *Identification* is the ability to identify a unique user of a system or a unique application that is running in the system.

When a user provides credentials in an attempt to log in to a SAS Viya user interface, those credentials are first validated; this is the authentication stage. If the credentials are valid, the user identity information is queried; this is the identification stage. Both stages must be successful in order for the user to log in.

SAS Viya provides a broad security framework that supports multiple third-party authentication options. The available options depend on the SAS Viya interface and operating system that are in use in your environment:

Table 1 *Authentication Options*

Type of Deployment	Operating System	Authentication Mechanism
full deployment	Linux	Users can be authenticated through SAS Logon Manager, using an identity and access management system (such as LDAP or SCIM); Kerberos; Security

Type of Deployment	Operating System	Authentication Mechanism
		<p>Assertion Markup Language (SAML), Pluggable Authentication Modules (PAM), or OAuth and OpenID Connect (OIDC).</p> <p>PAM can validate the user's credentials when accessing SAS Studio 5.2 (Basic) and CAS Server Monitor.</p> <p>Batch jobs submit credentials that require validation.</p>
	Windows	<p>Windows host authentication validates the user's credentials when accessing SAS Studio 5.2 (Basic) and CAS Server Monitor.</p> <p>Batch jobs submit credentials that require validation.</p> <p>Kerberos is the only supported authentication mechanism for SAS Viya visual interfaces and configuration of the middle tier environment.</p>
programming-only deployment	Linux	The only supported authentication mechanism is PAM.
	Windows	The only supported mechanism is Windows host authentication.

Authentication: How To

Authentication Mechanisms

Overview

Authentication mechanisms integrate SAS into your computing environment. External mechanisms that are supported by SAS Viya include direct LDAP authentication, which is referred to as either *LDAP* or as *authentication with an Identity and Access Management System (IAM)* in this guide. LDAP authentication is enabled by default. Other supported external mechanisms are host authentication, Kerberos, SAML, and OAuth 2.0 with OIDC.

PAM is used to extend host authentication on Linux. In a programming-only deployment, a pluggable authentication module validates the user's credentials. PAM enables the authentication of users accessing SAS Studio 5.2 (Basic) and CAS Server Monitor and of batch jobs that submit credentials.

Note: On Windows deployments, Windows host authentication validates the user's credentials when accessing SAS Studio 5.2 (Basic) and CAS Server Monitor, and for batch jobs. For SAS Viya visual interfaces and configuration of the middle-tier environment, Kerberos is the only authentication mechanism that is supported.

Configure the authentication mechanism that is appropriate for your environment and that is supported for your deployment type. For more information, see [“Concepts: Authentication Mechanisms”](#) on page 64.

Configure Kerberos (Linux Full Deployment)

The steps to configure Kerberos for Linux differ from the steps on Windows. To set up Kerberos for SAS Viya in a Windows environment, see [“Configure Kerberos \(Windows Full Deployment\)”](#) on page 17. To configure Kerberos on Linux, take the following steps:

- Verify prerequisites to make sure that certain conditions are met. See [“Verify Kerberos Prerequisites”](#) on page 4.
- Set up SAS Logon Manager and key SAS servers. See:
 - [“Configure Kerberos for SAS Logon Manager”](#) on page 6
 - [“Configure Kerberos for the CAS Server”](#) on page 8
 - [“Configure Kerberos for SAS Launcher Server”](#) on page 9
 - [“Configure Kerberos for SAS Object Spawner”](#) on page 11
- (Optional) Configure Kerberos constrained delegation. See [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 12.
- Perform the steps that are described in [“Configure Internet Options \(All Full Deployments\)”](#) on page 15. They apply to both Linux and Windows deployments.
- Configure your web browser for Kerberos. See [“Configure Microsoft Edge and Google Chrome to Use Kerberos”](#) on page 16 or [“Configure Mozilla Firefox to Use Kerberos”](#) on page 16.

Verify Kerberos Prerequisites

Before configuring Kerberos on Linux, verify the following items:

Note: These prerequisite components are usually configured by the Active Directory administrator.

- 1 Four service accounts exist in Active Directory.
- 2 A service principal name (SPN) for each of the service classes listed in [Table 2](#) is mapped to the service accounts from [Step 1](#).

Table 2 SAS Servers that Support Kerberos on Linux

Server	Service Class
SAS Logon Manager	HTTP

Server	Service Class
CAS server	sascas
SAS Launcher Server	sas-launcher
SAS Object Spawner	SAS

- a Verify that a mapping is already configured by running the `setspn -F -Q service_class/fully.qualified.hostname` command for each of the servers listed in [Table 2](#).

You should see output similar to the following:

```
CN=user-logon-name,OU=Service Accounts,OU=Domain
Controllers,OU=Servers,DC=EXAMPLE,DC=com
    service_class/fully.qualified.hostname
    service_class/HOSTNAME
```

Existing SPN found!

Note: The host name specifies the fully qualified domain name of the machine on which the server is running. If using unconstrained delegation, this service account must be trusted for unconstrained delegation, allowing delegation to all services. If using constrained delegation, then see [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 12.

If an SPN is not found, contact your IT support group for assistance with registering the SPN.

- b Verify that the service is linked to the service account by running the `setspn -L user-logon-name` command.

The value for `user-logon-name` is the same value that was specified for the common name (CN) from the previous command output, or as the `sAMAccountName` on the service account in Active Directory.

You should see output similar to the following:

```
Registered ServicePrincipalNames for CN=user-logon-name,OU=Service
Accounts,OU=Servers,
DC=EXAMPLE,DC=com:
    service_class/fully.qualified.hostname
    service_class/hostname
```

- 3 For the `sascas` service class, a user principal name (UPN) is required.

The matching account is used by the CAS Server to initialize Kerberos credentials for outbound authentication. UPNs are not required for the other service classes, but it is good practice to set them.

- 4 Verify that a keytab file has been generated by issuing the `ktutil rkt path-to-keytab-file.keytab list -e` command.

The following is sample output. Your keytab file is different.

Output 1 *Sample Linux Output*

```
slot KVNO Principal
- - -
  1   3   HTTP/<hostname>@<example>.com (aes256-cts-hmac-sha1-96)
```

For more information about the **ktutil** command, see the vendor documentation.

- 5 If the servers are accessed under aliases, an SPN must be added for each possible name used to reach the server. This applies to the HTTP service class, but could also apply to the sascas service class, if it is accessed directly by a client, such as SAS 9.4 or Python.

Note: It is possible to use a single service account for all four SPNs. In that case, all SPNs and the UPN for sascas must be assigned to the single service account.

Configure Kerberos for SAS Logon Manager

Specify properties to enable end users to log in to SAS Viya user interfaces using Kerberos authentication with SAS Logon Manager.

- 1 If you have not already done so, from SAS Environment Manager, add your user ID or an Active Directory group that contains the environment administrators, as a member of the SAS Administrators group. Then log off from SAS Environment Manager. For more information, see [“Add or Remove Custom Group Members” in SAS Viya Administration: Identity Management](#).

CAUTION

You must specify your personal user ID. Your user ID must be in your specified identity provider. It must match the user ID that you use to log on to your system. In addition, your user ID must be added to the SAS Administrators group because once Kerberos is configured, you can no longer sign in as the sasboot user.

- 2 Make sure that the keytab file is saved to a directory that is accessible to the user account that runs the SAS services.
- 3 Verify that the service principal name (SPN) is mapped to the user principal name (UPN).

```
setspn -F -Q HTTP/hostname.example.com
```


- 4 Configure the Kerberos authentication properties.
 - a From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
 - b In the **Definitions** list, select **sas.logon.kerberos**.
 - c In the top right corner of the window, click **New Configuration**.
 - d In the New sas.logon.kerberos Configuration window, enter the values for the following fields, based on your environment.

Table 3 Kerberos Configuration Fields, Default Values, and Descriptions

Configuration Field	Default Value	Description
Services	Global or SAS Logon Manager	By default, Global is selected and is the required value if you plan to configure Kerberos constrained delegation. For unconstrained delegation, choose SAS Logon Manager .
debug	On	Specifies whether to write debug messages in the log.
disableDelegationWarning	On	Specifies whether to display a warning message to users when Kerberos credential delegation is not properly configured.
holdOnToGSSContext:	On	This option is required to enable Kerberos delegation from SAS Logon Manager.
keyTabLocation	<i>file:///path-to-http-keytab-file</i>	Specifies the Uniform Resource Identifier (URI).
servicePrincipal	<i>principal-name-from-keytab</i> Note: If the environment includes multiple realms, this field should include the realm (for example, HTTP/ <i>fully.qualified.hostname@REALM</i>).	Issue the <code>ktab -l -k FILE:path-to-http-keytab-file.keytab</code> command.
spn	<i>service-principal-name</i>	Specifies the SPN, if it differs from the principal name in the keytab.

Configuration Field	Default Value	Description
stripRealmForGss	On	Specifies whether to remove the realm from the UPN.
impersonate	Off	Specifies whether to impersonate the user credentials using the Microsoft S4USelf extension to Kerberos for outgoing connections. Note: If using constrained delegation, set impersonate to On .

Note: Contact your administrator for the keytab location and the host name of the service principal.

- e Click **Save**.
- 5 Add Kerberos to the active profile.
 - a In the navigation pane, switch to the **All services** list and select **SAS Logon Manager**.
 - b In the **jvm** instance, click .
 - c Click **Add Property**.
In the **Name** field, specify `java_option_profiles`.
For the **Value**, specify `-Dspring.profiles.active=ldap,postgresql,kerberos`. Click **Save**.
 - d Click **Save** again.

- 6 Restart the SAS Logon Manager service.
For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

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

Configure Kerberos for the CAS Server

- 1 Create a keytab file for the CAS server.

The file is used to validate incoming user Kerberos tickets and generate server identity Kerberos tickets for access to Kerberized resources, such as Hadoop. By default, the keytab file should be saved in the `/etc/sascas.keytab` file and should be readable only by the CAS server. If you save the file in a different directory or use a different file name, set the `KRB5_KTNAME` environment

variable (for example, `env.KRB5_KTNAME = 'fully-qualified-filename'`) to the fully qualified file name. For more information, see [“CAS Environment Variables” in SAS Viya Administration: SAS Cloud Analytic Services](#).

- 2 Verify that the service principal name (SPN) is mapped to the principal name.

```
setspn -F -Q sascas/fully.qualified.hostname
```

- 3 If you changed the default principal name, set the `CAS_SERVER_PRINCIPAL` environment variable (for example, `env.CAS_SERVER_PRINCIPAL = 'principal-name'`).

By default, the CAS server uses the following Kerberos principal name: `sascas/fully-qualified-DNSname`. The CAS server searches for this principal in the keytab file.

- 4 Add the 'kerb' option to the `cas.provlist` configuration file option (for example, `cas.provlist = 'oauth.ext.kerb'`).

For more information about the configuration file option, see [“Configuration File Options” in SAS Viya Administration: SAS Cloud Analytic Services](#).

- 5 Enable the Kerberos option for authentication to CAS and to the Compute server.
 - a From SAS Environment Manager, navigate to the Launcher service configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).

- b In the **Definitions** list, select **sas.compute**.

- c Click .

- d In the Edit `sas.compute` Configuration window, select the **kerberos.enabled** option.

- e Click **Save**.

- 6 Restart the CAS controller.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

```
sudo systemctl restart sas-viya-cascontroller-default
```

Configure Kerberos for SAS Launcher Server

- 1 Create a keytab file for SAS Launcher Server to use.

- 2 Save the keytab file on the file system of any host where the SAS Launcher Server is running.

There is no default location where the keytab file should be saved, so it can be placed anywhere on the file system.

- 3 Make sure that the keytab file is accessible to the “sas” account, the Linux operating system account that runs the process for SAS Launcher Server.

- 4 Verify that the service principal name (SPN) is mapped to the principal name.

```
setspn -F -Q sas-launcher/fully.qualified.hostname
```

- 5 Complete the following steps as a user with root or sudo privileges:

- a Source the `consul.conf` file to add configuration values that use the SAS Security framework certificate truststore.

```
source SAS-Viya-configuration-directory/consul.conf
```

- b Run the `sas-bootstrap-config` script for the SAS Launcher Server keytab.

Note: Enter the command on a single line. Multiple lines are used for the command to improve readability.

```
SAS-Viya-home-directory/bin/sas-bootstrap-config --token-file SAS-Viya-configuration-
directory/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token kv write
--force --key config/launcher-server/global/keytab --value path-to-keytab-file
```

6 Set the SPN.

- If you are using an SPN-based keytab file, no additional steps are required. The default SPN for SAS Launcher Server is `sas-launcher/fully.qualified.hostname`.
- If you are using an alias (for example, when configuring a disaster recovery environment), the `SAS_SERVICE_PRINCIPAL` environment variable must be set to the alias SPN (for example, `sas-launcher/alias_fully.qualified.domain.name`).

- 1 Source the `consul.conf` file to add configuration values that use the SAS Security framework certificate truststore.

```
source SAS-Viya-configuration-directory/consul.conf
```

- 2 Run the `sas-bootstrap-config` script for the SAS Launcher Server for `SAS_SERVICE_PRINCIPAL`.

Note: Enter the command on a single line. Multiple lines are used for the command to improve readability.

```
SAS-Viya-home-directory/bin/sas-bootstrap-config --token-file SAS-Viya-configuration-
directory/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token kv
write --force --key config/launcher-server/global/environment/SAS_SERVICE_PRINCIPAL --
value sas-launcher/alias_FQDN
```

- If you are using a User Principal Name (UPN) based keytab file, the `SAS_SERVICE_PRINCIPAL` environment variable must match the UPN, `principal_name@REALM.COM`, where `principal_name` is the user account where the `sas-launcher` SPN is registered, and `REALM.COM` is the realm or domain.

- 1 Source the `consul.conf` file to add configuration values that use the SAS Security framework certificate truststore.

```
source SAS-Viya-configuration-directory/consul.conf
```

- 2 Run the `sas-bootstrap-config` script for the SAS Launcher Server for `SAS_SERVICE_PRINCIPAL`.


Note: Enter the command on a single line. Multiple lines are used for the command to improve readability.

```
SAS-Viya-home-directory/bin/sas-bootstrap-config --token-file SAS-Viya-configuration-
directory/etc/SASSecurityCertificateFramework/tokens/consul/default/client.token kv
write --force --key config/launcher-server/global/environment/SAS_SERVICE_PRINCIPAL --
value principal_name@REALM.COM
```

7 Restart SAS Launcher Server.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

```
sudo systemctl restart sas-viya-runlauncher-default
```

- 8 Enable the Kerberos option for authentication to SAS Compute Server.
 - a From SAS Environment Manager, navigate to the Launcher service configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
 - b In the **Definitions** list, select **sas.compute**.
 - c Click .
 - d In the Edit sas.compute Configuration window, select the **kerberos.enabled** option.
 - e Click **Save**.
- 9 To enable Kerberos constrained delegation, see [“Configure Kerberos Constrained Delegation in Active Directory” on page 12](#).

Configure Kerberos for SAS Object Spawner

You can enable Kerberos for direct connections from SAS Enterprise Guide 8.2 to SAS Object Spawner on SAS Viya. Complete the following steps:

- 1 Disable TLS for SAS Object Spawner in a Linux full deployment.
 - a Edit the *SAS-Viya-configuration-directory/etc/spawner/default/spawner_usermods.sh* file.
 - b Add the following line to the bottom of the file:


```
spawner_options="${spawner_options//--ssl*/}"
```
 - c Restart SAS Object Spawner.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

```
sudo systemctl restart sas-viya-spawner-default
```
 - d Verify your changes by looking at the output from the following command:


```
ps -ef |grep objsp
```

The command should no longer include the following options:

 - -sslpvtkeyloc
 - -sslcertloc
 - -sslpvtkeypassfile
 - -ssllalistloc
- 2 Edit the *SAS-Viya-configuration-directory/etc/spawner/default/spawner_usermods.sh* file.
 - a Add the following lines before the existing USERMODS line:


```
CMD_OPTIONS=-sspi
export KRB5_KTNAME=/opt/sas/sas.keytab
```
 - b Update the existing USERMODS line to the following:

```
USERMODS="$JREOPTIONS $CMD_OPTIONS"
```

- c To configure a custom SPN, add the following lines before the existing USERMODS line:

```
SAS_SERVICE_PRINCIPAL=MYSAS/my_sas_server.com
export SAS_SERVICE_PRINCIPAL
```

Note: The connection profile in SAS Enterprise Guide 8.2 must be updated with the custom SPN.

- d Save your changes.

- 3 Restart SAS Object Spawner.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

```
sudo systemctl restart sas-viya-spawner-default
```

Configure Kerberos Constrained Delegation in Active Directory

Note: If you are configuring Kerberos unconstrained delegation, you can skip these steps.

- 1 As the Windows domain administrator, navigate to **System and Security** ⇒ **Administrative Tools** ⇒ **Active Directory Users and Computers** to access the properties window for the relevant account and grant the delegation privilege.
- 2 For SAS Logon Manager, add all Kerberos protected services for the servers listed in [Table 4 on page 12](#):

Table 4 *Kerberos-Protected Servers*

Server	Service Class
CAS Server	sascas
SAS Launcher Server	sas-launcher
SAS Object Spawner	SAS

- a Select the account to which the HTTP service class is defined.
- b Right-click the name and select **Properties**.
- c In the Properties window, select the **Delegation** tab.
- d On the **Delegation** tab, select the **Trust this user for delegation to specified services only** and the **Use any authentication protocol** check boxes. Then click **Add**.
- e In the Add Services window, click **Users or Computers**.
- f In the Select Users or Computers window, complete the following for each of the Kerberos protected service classes (sascas and sas-launcher):

- 1 In the **Enter the object names to select** text box, enter the domain account to which the service class is defined from [Table 4 on page 12](#). Then click **Check Names**.
 - 2 If the name is found, click **OK**. Otherwise, check the spelling of the name and enter it again.
 - 3 Repeat the previous two steps to select additional SPNs for the service class.
 - 4 When you are done, click **OK**.
- g In the Add Services window, click **OK**.
- h In the Properties window, click **OK**.
- 3 For the CAS server:
- a Select the account to which the service class is defined.
 - b Right-click the name and click **Properties**.
 - c In the Properties window, select the **Delegation** tab.
 - d On the **Delegation** tab, select the **Trust this user for delegation to specified services only** and **Use any authentication protocol** check boxes. Then click **Add**.
 - e In the Add Services window, click **Users or Computers**.
 - f In the Select Users or Computers window, complete the following for the Kerberos protected services that the server accesses:
 - 1 In the **Enter the object names to select** text box, enter the account for the Kerberos protected service that the CAS server accesses, such as Microsoft SQL Server. Then click **Check Names**.
 - 2 If the name is found, click **OK**. Otherwise, check the spelling of the name and enter it again.
 - 3 Repeat the previous two steps to select additional SPNs for the sascas service.
 - 4 When you are done, click **OK**.
 - g In the Add Services window, click **OK**.
 - h In the Properties window, click **OK**.
- 4 For SAS Launcher Server:
- a Select the account to which the service class is defined.
 - b Right-click the name and click **Properties**.
 - c In the Properties window, select the **Delegation** tab.
 - d On the Delegation tab, select the **Trust this user for delegation to specified services only** and **Use any authentication protocol** check boxes. Then click **Add**.
 - e In the Add Services window, click **Users or Computers**.
 - f In the Select Users or Computers window, complete the following for the Kerberos protected services that the server accesses:

- 1 In the **Enter the object names to select** text box, enter the account for the Kerberos protected service the SAS Compute Server accesses. This should include sascas, as well as any other services, such as Microsoft SQL Server. Then click **Check Names**.
 - 2 If the name is found, click **OK**. Otherwise, check the spelling of the name and enter it again.
 - 3 Repeat the previous two steps to select additional SPNs for the sas-launcher service.
 - 4 When you are done, click **OK**.
- g In the Add Services window, click **OK**.
- h In the Properties window, click **OK**.

Note: No additional configuration of SAS Object Spawner is required to support constrained delegation.

Note: It is not necessary to restart the SAS services. However, Windows client users need to log off and log back on to their desktop. Or, from the command prompt, they can use the `klint purge` command to clear their Kerberos credentials cache. Windows regenerates the Kerberos credentials as necessary.

Validate Kerberos Configuration

All users are authenticated using OAuth 2.0 and OpenID Connect. Complete the following steps to verify that Kerberos is configured correctly:

- 1 Check the CAS log to see how the non-delegated user authenticated to CAS by running the following command:

```
cat /var/log/sas/viya/cas/default/* |grep non_delegated_user|grep authenticated|tail -1
```

- 2 Look for output similar to the following:

```
2018-06-12T11:03:35,376 INFO [00002846] <non_delegated_user> local MAIN NoUser [tkidentgss.c:741]
- User <non_delegated_user>@<domain_name> successfully authenticated using the OAuth authentication
provider.
```

On Linux systems, delegation occurs only for users who are in the CASHostAccountRequired custom group. Users with delegated Kerberos credentials are also authenticated with the Kerberos authentication provider to delegate their identity to CAS. To validate Kerberos for the delegated user, complete the following steps:

- 1 Check the CAS log to see how the delegated user authenticated to CAS.

On Linux, run the following command:

```
cat /var/log/sas/viya/cas/default/* |grep delegated_user|grep kerberos|tail -1
```

- 2 Look for output similar to the following:

```
2018-06-12T11:03:35,376 INFO [00002846] <delegated_user> local MAIN NoUser [tkident.c:741] - User
<delegated_user> successfully authenticated using the Kerberos authentication provider.
```

Configure Internet Options (All Full Deployments)

The settings in this section apply to full deployments with Kerberos authentication configured.

Configure Security Settings

- 1 In the Windows Control Panel, open Internet Options.
- 2 In the Internet Properties window, select the **Security** tab.
- 3 Select **Local intranet**, and then click **Sites**.
- 4 In the Local intranet window, configure the intranet domain settings.
 - a Verify that the following options are selected:
 - **Include all local (Intranet) sites not listed in other zones**
 - **Include all sites that bypass the proxy server**
 - b Click **Close**, and then click **OK**.
- 5 Configure intranet authentication.
 - a In the **Security level for this zone** area, click **Custom level**.
 - b In the Security Settings - Local Intranet Zone window, scroll to the **User Authentication** section, select **Automatic Logon only in Intranet Zone**, and click **OK**.

Configure Connection Settings

If your site uses a proxy server, follow these steps:

- 1 In the Internet Properties window, select the **Connections** tab.
- 2 Click **LAN settings**.
- 3 In the Local Area Network (LAN) Settings window, verify that the proxy server address and port number are correct.
- 4 Click **Advanced**.
- 5 In the Proxy Settings window, verify that the correct domain names are entered in the **Exceptions** field. Then click **OK**.
- 6 Click **OK**.

Configure Integrated Windows Authentication

- 1 In the Internet Properties window, select the **Advanced** tab.
- 2 Scroll to the **Security** section, and verify that **Enable Integrated Windows Authentication** is selected.
- 3 Click **OK** and restart your computer to activate the changes.

Configure Microsoft Edge and Google Chrome to Use Kerberos

Configure User Delegation for Microsoft Edge

Complete the following steps after configuring Integrated Windows Authentication:

- 1 Open the Windows registry editor.
- 2 Add the following REG_SZ keys:
 - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge\AuthServerAllowlist
Specifies which servers to enable for integrated authentication. Set the value to the SAS Web Server host name: *hostname.example.com*.
 - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge\AuthNegotiateDelegateAllowlist
Specifies servers to which Microsoft Edge can delegate. Set the value to the SAS Web Server host name: *hostname.example.com*.

Configure User Delegation for Google Chrome

By default, Chrome disables the delegation of Kerberos credentials. The Windows registry must be updated. Microsoft recommends performing a system backup before editing the registry. Complete the following steps to enable Kerberos delegation after configuring Integrated Windows Authentication:

- 1 Open the Windows registry editor.
- 2 Add the following REG_SZ keys:
 - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\AuthServerAllowlist
Specifies the servers that should be allowed for integrated authentication. Set the value to the SAS Web Server host name: *hostname.example.com*.
 - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\AuthNegotiateDelegateAllowlist
Specifies servers to which Chrome can delegate. Set the value to the SAS Web Server host name: *hostname.example.com*.

.....
Note: You might also need to add Google and Chrome under Policies.

Configure Mozilla Firefox to Use Kerberos

Configure Kerberos for Firefox

- 1 From a browser window, navigate to `about:config`.
- 2 Click **I accept the risk!** to accept the security warning.
- 3 In the **Search** field, enter `network.negotiate`.
- 4 Double-click the `network.negotiate-auth.trusted-uris` Preference Name, enter `http://hostname.example.com`, in the **Enter string value** field, and then click **OK**.

.....
Note: The values in the **Enter string value** field are comma-separated.

Configure User Delegation for Firefox

- 1 From a browser window, navigate to `about:config`.
- 2 Click **I accept the risk!** to accept the security warning.
- 3 In the **Search** field, enter `network.negotiate`.
- 4 Double-click the `network.negotiate-auth.delegation-uris` Preference Name, enter `http://hostname.example.com` in the **Enter string value** field, and then click **OK**.

Configure Kerberos (Windows Full Deployment)

To configure Kerberos on Windows, you must take the following steps:

- Verify prerequisites to make sure that certain conditions are met. See [“Verify Kerberos Prerequisites” on page 17](#).
- Set up SAS Logon Manager and SAS Object Spawner. See [“Configure Kerberos for SAS Logon Manager” on page 19](#) and [“Configure Kerberos for SAS Object Spawner” on page 22](#).
- (Optional) Configure Kerberos constrained delegation. See [“Configure Kerberos Constrained Delegation for SAS Launcher Server and SAS Object Spawner” on page 23](#) and [“Configure Kerberos Constrained Delegation in Active Directory” on page 24](#).
- Perform the steps that are described in [“Configure Internet Options \(All Full Deployments\)” on page 26](#). They apply to both Linux and Windows deployments.
- Configure your web browser for Kerberos. See [“Configure Microsoft Edge and Google Chrome to Use Kerberos” on page 27](#) or [“Configure Mozilla Firefox to Use Kerberos” on page 28](#).

Verify Kerberos Prerequisites

Before configuring Kerberos, make sure that the following items exist:

Note: These prerequisite components are usually configured by the Active Directory administrator.

- 1 Three service accounts exist in Active Directory.
- 2 The `cas` account requires the following properties:
 - Membership in the local Administrators groups on the machine where the CAS Server is installed.
 - Has the following privileges:
 - Log on as a service
 - Replace a Process Level Token
 - The recommended account name is `cas`. However, the name must be unique for the equivalent user on the domain. The maximum length of the name is 20 characters.
 - This account requires a password. If the password expires, the CAS service no longer starts.
- 3 A service principal name (SPN) for each of the service classes listed in [Table 5](#) is mapped to the service accounts from [Step 2](#).

Table 5 Servers That Support Kerberos on Windows

Server	Service Class
SAS Logon Manager	HTTP
CAS server	sascas
SAS Object Spawner	SAS

- a Verify that a mapping is already configured by running the `setspn -F -Q service_class/fully.qualified.hostname` command for each of the servers listed in [Table 5](#).

You should see output similar to the following:

```
CN=user-logon-name,OU=Service Accounts,OU=Domain
Controllers,OU=Servers,DC=EXAMPLE,DC=com
    service_class/fully.qualified.hostname
    service_class/HOSTNAME
```

Existing SPN found!

Note: The host name specifies the fully qualified domain name of the machine where the server is running.

Note: For CAS, the SPN must be registered on the service account that is running the server. If using unconstrained delegation, then this service account must be trusted for unconstrained delegation, allowing delegation to all services. If using constrained delegation, then see [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 24.

If an SPN is not found, then contact your information technology support group for assistance with registering the SPN.

- b Verify that the service is linked to the service account by running the `setspn -L user-logon-name` command.

The value for `user-logon-name` is the same one identified in the common name (CN) from the previous command output, or as the `sAMAccountName` on the service account in Active Directory.

You should see output similar to the following:

```
Registered ServicePrincipalNames for CN=user-logon-name,OU=Service
Accounts,OU=Servers,
DC=EXAMPLE,DC=com:
    service_class/fully.qualified.hostname
    service_class/hostname
```

- 4 For the `sascas` service class, a user principal name (UPN) is required.

The matching account is used by the CAS Server to initialize Kerberos credentials for outbound authentication. UPNs are not required for the other service classes, but it is good practice to set them.

- 5 Verify that a keytab file has been generated by issuing the `ktab.exe -l -k FILE:path-to-keytab-file.keytab` command.

The following is sample output. Your keytab file is different.

Output 2 Sample Windows Output

```
Keytab name: <filename>.keytab
KVNO    Principal
-      -
1      HTTP/<hostname>@<example>.com
```

For more information about the **ktab** command, see the vendor documentation.

- 6 If the servers are accessed under aliases, an SPN must be added for each possible name used to reach the server. This applies to the HTTP service class, but could also apply to the sascas service class, if it is accessed directly by a client, such as SAS 9.4 or Python.
- 7 Verify that the computer machine object must be trusted for delegation. SAS Launcher Server runs under the local system account on the machine it is deployed on and registers its own SPN. Therefore, a sas-launcher service class is not required.
 - For **unconstrained** delegation, the server on which SAS Launcher Server is running must be marked in Active Directory as trusted for delegation to any service.
 - For **constrained** delegation, the server on which SAS Launcher Server is running must be marked in Active Directory as trusted for delegation to specified services.

Note: If the computer is not marked as trusted, it cannot use the user's Kerberos ticket to access remote file systems, nor can it launch CAS sessions under user identity.

Note: It is possible to use a single service account for all three SPNs. In that case, all SPNs and the UPN for sascas must be assigned to the single service account.

Configure Kerberos for SAS Logon Manager

Now specify properties to enable end users to log in to SAS Viya user interfaces using Kerberos authentication with SAS Logon Manager.

- 1 If you have not already done so, from SAS Environment Manager, add your user ID or an Active Directory group that contains the environment administrators, as a member of the SAS Administrators group. Then log off from SAS Environment Manager. For more information, see [“Add or Remove Custom Group Members” in SAS Viya Administration: Identity Management](#).

CAUTION

You must specify your personal user ID. Your user ID must be in your specified identity provider. It must match the user ID that you use to log on to your system. In addition, your user ID must be added to the SAS Administrators group because once Kerberos is configured, you can no longer sign in as the sasboot user.

- 2 Make sure that the keytab file is saved to a directory that is accessible to the user account that runs the SAS services.

- 3 Verify that the service principal name (SPN) is mapped to the user principal name (UPN).

```
setspn -F -Q HTTP/hostname.example.com
```


- 4 Configure the Kerberos authentication properties.
- a From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
 - b In the **Definitions** list, select **sas.logon.kerberos**.
 - c In the top right corner of the window, click **New Configuration**.
 - d In the New sas.logon.kerberos Configuration window, enter the values for the following fields, based on your environment.

Table 6 Kerberos Configuration Fields, Default Values, and Descriptions

Configuration Field	Default Value	Description
Services	Global or SAS Logon Manager	By default, Global is selected and is the required value if you plan to configure Kerberos constrained delegation. For unconstrained delegation, select SAS Logon Manager .
debug	On	Specifies whether to write debug messages in the log.
disableDelegationWarning	On	Specifies whether to display a warning message to users when Kerberos credential delegation is not properly configured.
holdOnToGSSContext:	On	This option is required to enable Kerberos delegation from SAS Logon Manager.
keyTabLocation	file:///path-to-http-keytab-file Note: You must use forward slashes, even on Windows systems	Specifies the Uniform Resource Identifier (URI).

Configuration Field	Default Value	Description
	(for example, file:///c:/ path-to-http-keytab-file	
servicePrincipal	principal-name-from-keytab Note: If the environment includes multiple realms, this field should include the realm (for example, HTTP/fully.qualified.hostname@REALM).	On Windows, issue the ktab.exe -l -k FILE:<userSuppliedValue>path-to-http-keytab-file</userSuppliedValue>.k eytab command from the directory where Java is installed on your machine.
spn	service-principal-name	Specifies the SPN, if it differs from the principal name in the keytab.
stripRealmForGss	On	Specifies whether to remove the realm from the UPN.
impersonate	Off	Specifies whether to impersonate the user credentials using the Microsoft S4USelf extension to Kerberos for outgoing connections. Note: If using constrained delegation, set impersonate to On .

Note: Contact your administrator for the keytab location and the host name of the service principal.

- e Click **Save**.
- 5 Add Kerberos to the active profile.
 - a In the navigation pane, switch to the **All services** list and select **SAS Logon Manager**.
 - b In the **jvm** instance, click .
 - c Click **Add Property**.
In the **Name** field, specify `java_option_profiles`.

For the **Value**, specify `-Dspring.profiles.active=ldap,postgresql,kerberos`. Click **Save**.

d Click **Save** again.

- Restart the SAS Logon Manager service. In Windows Services Manager, right-click **SAS Logon Manager service** and select **Restart**.

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

Note: Once Kerberos is enabled on Windows, a browser running on the same machine where the services are deployed cannot connect to SAS Viya visual interfaces.

Configure Kerberos for SAS Object Spawner

You can enable Kerberos for direct connections from SAS Enterprise Guide 8.2 to SAS Object Spawner on SAS Viya. Complete the following steps:

- Edit the `SAS-Viya-configuration-directory\etc\spawner\default\spawner_usermods.bat` file.

- Update the existing USERMODS line to the following:

```
Set USERMODS_OPTIONS=-sspi
```

- Uninstall and re-install the Windows Server for SAS Object Spawner, by running the following commands:

```
SAS-Viya-configuration-directory\etc\spawner\default\spawner.bat remove
SAS-Viya-configuration-directory\etc\spawner\default\spawner.bat install
```

- If your SAS servers use DNS aliases, manual registration of the SPNs is necessary in order to support the Kerberos connections. By default, the SAS Object Spawner service runs as a local system. A best practice is to set this service to run under a service account.

Although the account must have administrator privileges, system administrators can configure delegation to a single account in order to enable access to third-party resources (such as network shares).

When using a service level account to run the object spawner service, you can configure the necessary SPNs with the following commands:

```
setspn -A SAS/computerNetbios -u domain\ObjectSpawnerServiceAccount
setspn -A SAS/computerFullname -u domain\ObjectSpawnerServiceAccount
setspn -A SAS/computerFullAlias -u domain\ObjectSpawnerServiceAccount
setspn -A SAS/computerShortAlias -u domain\ObjectSpawnerServiceAccount
```

If you choose to run SAS Object Spawner under a service level account, that account must be a Windows administrator on the spawner's host and must have the following Windows user rights **Adjust memory quotas for a process**, **Replace a process level token**, and **Act as part of the Operating System**. These user rights assignments are part of the local security policy for the Windows computer that hosts the spawner.

CAUTION

SPNs must be unique among the objects that are in the same Active Directory forest. A given SPN must not be assigned to more than one object. If a duplicate SPN is found in response to a TGS request, the Key Distribution Center sends an error to the client that the principal was not found.

- 5 To configure a custom SPN, it must be registered with a domain user account. Then the SAS Object Spawner service must be configured to run under that domain user account.

Note: The connection profile in SAS Enterprise Guide 8.2 must be updated with the custom SPN.

- 6 To enable Kerberos constrained delegation, see [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 24.

Configure Kerberos Constrained Delegation for SAS Launcher Server and SAS Object Spawner

Note: If you are configuring Kerberos unconstrained delegation, you can skip these steps.

- 1 Set the `SAS_CONSTRAINED_DELEG_ENABLED` environment variable.
 - a As the local administrator, in the Windows Control Panel, open System.
 - b Click the **Advanced system settings** link.
 - c In the System Properties window, click **Environment Variables**.
 - d In the Environment Variables window, in the **System variables** section, click **New**.
 - e In the New System Variable window, in the **Variable name:** field, enter `SAS_CONSTRAINED_DELEG_ENABLED`. In the **Variable value:** field, enter 1. Then click **OK**.

Note: The variable value can be set to any value.

 - f In the Environment Variables window, click **OK**.
 - g In the System Properties window, click **OK**.
- 2 In Windows Services Manager, restart SAS Launcher Server, SAS Object Spawner, and the CAS server.
 - Right-click the **SAS Runlauncher Service** and select **Restart**.
 - Right-click the **SAS Object Spawner service** and select **Restart**.
 - Right-click **SAS Cloud Analytic Services** and select **Restart**.
- 3 To enable Kerberos constrained delegation, see [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 24.

Note: Once constrained delegation is enabled, all SAS Compute Server processes and all CAS server session process run as the same user ID as SAS Launcher Server and the CAS server,

respectively. The internal threads of each process impersonate the client user ID. Therefore, they have the same rights and privileges as the client user.

Configure Kerberos Constrained Delegation in Active Directory

Note: If you are configuring Kerberos unconstrained delegation, you can skip these steps.

- 1 As the Windows domain administrator, navigate to **System and Security** ⇒ **Administrative Tools** ⇒ **Active Directory Users and Computers** to access the properties window for the relevant account and grant the delegation privilege.

- 2 For SAS Logon Manager, add all Kerberos protected services for the servers listed in [Table 7 on page 24](#):

Table 7 *Kerberos Protected Servers*

Server	Service Class
CAS server	sascas
SAS Launcher Server	sas-launcher
SAS Object Spawner	SAS

- a Select the account to which the HTTP service class is defined.
 - b Right-click the name and select **Properties**.
 - c In the Properties window, select the **Delegation** tab.
 - d On the **Delegation** tab, select the **Trust this computer for delegation to specified services only** and the **Use any authentication protocol** check boxes. Then click **Add**.
 - e In the Add Services window, click **Users or Computers**.
 - f In the Select Users or Computers window, complete the following for each of the Kerberos protected service classes (sascas and sas-launcher):
 - 1 In the **Enter the object names to select** text box, specify the domain account to which the service class is defined from [Table 7 on page 24](#). Then click **Check Names**.
 - 2 If the name is found, click **OK**. Otherwise, check the spelling of the name and enter it again.
 - 3 Repeat the previous two steps to select additional SPNs for the service class.
 - 4 When you are done, click **OK**.
 - g In the Add Services window, click **OK**.
 - h In the Properties window, click **OK**.
- 3 For the CAS server:

- a Select the account to which the service class is defined.
 - b Right-click the name and click **Properties**.
 - c In the Properties window, select the **Delegation** tab.
 - d On the **Delegation** tab, select the **Trust this user for delegation to specified services only** and **Use any authentication protocol** check boxes. Then click **Add**.
 - e In the Add Services window, click **Users or Computers**.
 - f In the Select Users or Computers window, complete the following for the Kerberos protected services that the server accesses:
 - 1 In the **Enter the object names to select** text box, enter the account for the Kerberos protected service that the CAS server accesses, such as Microsoft SQL Server. Then click **Check Names**.
 - 2 If the name is found, click **OK**. Otherwise, check the spelling of the name and enter it again.
 - 3 Repeat the previous two steps to select additional SPNs for the sascas service.
 - 4 When you are done, click **OK**.
 - g In the Add Services window, click **OK**.
 - h In the Properties window, click **OK**.
- 4 For SAS Launcher Server:
- a Select the account to which the service class is defined.
 - b Right-click the name and select **Properties**.
 - c In the Properties window, select the **Delegation** tab.
 - d On the **Delegation** tab, select the **Trust this computer for delegation to specified services only** and **Use any authentication protocol** check boxes. Then click **Add**.
 - e In the Add Services window, click **Users or Computers**.
 - f In the Select Users or Computers window, complete the following for the Kerberos protected services that the server accesses:
 - 1 In the **Enter the object names to select** text box, enter the account for the Kerberos protected service the SAS Compute Server accesses. This should include sascas, as well as any other services, such as Microsoft SQL Server. Then click **Check Names**.
 - 2 If the name is found, click **OK**. Otherwise, check the spelling of the name and enter it again.
 - 3 Repeat the previous two steps to select additional SPNs for the sas-launcher service.
 - 4 When you are done, click **OK**.
 - g In the Add Services window, click **OK**.
 - h In the Properties window, click **OK**.

Note: If a SAS Viya application needs to access a resource in a universal naming convention (UNC) path, the computer object where the SAS service is running must be trusted for delegation to the applicable common internet file system (CIFS) service class. This requirement enables Windows to use server message block (SMB) to access UNC paths. SMB runs under the local system computer account.

For example, if the Compute service is running on HostA and needs to assign a libname to a path on HostB, the computer object for HostA must define delegation to trust CIFS for HostB.

Validate Kerberos Configuration

All users are authenticated using OAuth 2.0 and OpenID Connect. Complete the following steps to verify that Kerberos is configured correctly:

- 1 Check the CAS log to see how the non-delegated user authenticated to CAS. Navigate to the `C:\ProgramData\SAS\Viya\var\log\cas\default` directory and view the contents of the `cas_date_hostname` file.
- 2 Look for output similar to the following:

```
2018-06-12T11:03:35,376 INFO [00002846] <non_delegated_user> local MAIN NoUser [tkidentgss.c:741]
- User <non_delegated_user>@<domain_name> successfully authenticated using the OAuth authentication
provider.
```

On Windows systems, users are automatically delegated. Users with delegated Kerberos credentials are also authenticated with the Kerberos authentication provider to delegate their identity to CAS. To validate Kerberos for the delegated user, complete the following steps:

- 1 Check the CAS log to see how the delegated user authenticated to CAS.
On Windows, navigate to the `C:\ProgramData\SAS\Viya\var\log\cas\default` directory and view the contents of the `cas_date_hostname` file
- 2 Look for output similar to the following:

```
2018-06-12T11:03:35,376 INFO [00002846] <delegated_user> local MAIN NoUser [tkident.c:741] - User
<delegated_user> successfully authenticated using the Kerberos authentication provider.
```

Configure Internet Options (All Full Deployments)

The settings in this section apply to full deployments with Kerberos authentication configured.

Configure Security Settings

- 1 In the Windows Control Panel, open Internet Options.
- 2 In the Internet Properties window, select the **Security** tab.
- 3 Select **Local intranet**, and then click **Sites**.
- 4 In the Local intranet window, configure the intranet domain settings.
 - a Verify that the following options are selected:

- **Include all local (Intranet) sites not listed in other zones**
 - **Include all sites that bypass the proxy server**
- b Click **Close**, and then click **OK**.
- 5 Configure intranet authentication.
 - a In the **Security level for this zone** area, click **Custom level**.
 - b In the Security Settings - Local Intranet Zone window, scroll to the **User Authentication** section, select **Automatic Logon only in Intranet Zone**, and click **OK**.

Configure Connection Settings

If your site uses a proxy server, follow these steps:

- 1 In the Internet Properties window, select the **Connections** tab.
- 2 Click **LAN settings**.
- 3 In the Local Area Network (LAN) Settings window, verify that the proxy server address and port number are correct.
- 4 Click **Advanced**.
- 5 In the Proxy Settings window, verify that the correct domain names are entered in the **Exceptions** field. Then click **OK**.
- 6 Click **OK**.

Configure Integrated Windows Authentication

- 1 In the Internet Properties window, select the **Advanced** tab.
- 2 Scroll to the **Security** section, and verify that **Enable Integrated Windows Authentication** is selected.
- 3 Click **OK** and restart your computer to activate the changes.

Configure Microsoft Edge and Google Chrome to Use Kerberos

Configure User Delegation for Microsoft Edge

Complete the following steps after configuring [Internet Options on page 15](#) and Integrated Windows Authentication:

- 1 Open the Windows registry editor.
- 2 Add the following REG_SZ keys:
 - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge\AuthServerAllowlist
Specifies which servers to enable for integrated authentication. Set the value to the SAS Web Server host name: hostname.example.com.
 - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge\AuthNegotiateDelegateAllowlist
Specifies the servers to which Microsoft Edge can delegate. Set the value to the SAS Web Server host name: hostname.example.com.

Configure User Delegation for Google Chrome

By default, Chrome disables the delegation of Kerberos credentials. The Windows registry must be updated. Microsoft recommends performing a system backup before editing the registry. Complete the following steps to enable Kerberos delegation after configuring Integrated Windows Authentication:

1 Open the Windows registry editor.

2 Add the following REG_SZ keys:

\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\AuthServerAllowlist

Specifies which servers should be allowed for integrated authentication. Set the value to the SAS Web Server host name: *hostname.example.com*.

\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome

\AuthNegotiateDelegateAllowlist

Specifies which servers Chrome can delegate to. Set the value to the SAS Web Server host name: *hostname.example.com*.

.....
Note: You might also need to add Google and Chrome under Policies.

Configure Mozilla Firefox to Use Kerberos

Configure Kerberos

1 From a browser window, navigate to `about:config`.

2 Click **I accept the risk!** to accept the security warning.

3 In the **Search** field, enter `network.negotiate`.

4 Double-click the **network.negotiate-auth.trusted-uris** Preference Name, enter `http://hostname.example.com`, in the **Enter string value** field, and then click **OK**.

.....
Note: The values in the **Enter string value** field are comma-separated.

Configure User Delegation

1 From a browser window, navigate to `about:config`.

2 Click **I accept the risk!** to accept the security warning.

3 In the **Search** field, enter `network.negotiate`.

4 Double-click the **network.negotiate-auth.delegation-uris** Preference Name, enter `http://hostname.example.com` in the **Enter string value** field, and then click **OK**.

Configure OAuth and OIDC (Linux Full Deployment)

Overview

The following sections provide details for configuring OpenID Connect (OIDC) in a single tenant environment. For information about configuring OAuth and OIDC in a multi-tenant environment, see [“About Identity Management and Authentication in Multi-tenancy” in SAS Viya Administration: Multi-tenancy](#).

Configuration of OIDC typically follows the following pattern:

- 1 [“Configure the OIDC IdP” on page 29](#)
- 2 [“Configure SAS Viya with Information about the OIDC IdP” on page 29](#)

Configure the OIDC IdP

Instructions are provided for configuring OIDC with the following identity providers (IdPs):

- [ISAM Scenario on page 99](#)
- [Okta Scenario on page 105](#)

Configure SAS Viya with Information about the OIDC IdP

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 In the **Definitions** list, select **sas.logon.oauth.providers**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.logon.oauth.providers Configuration window, enter values for the required fields, based on your environment. The following table provides guidance about the information needed for the listed fields:

Table 8 OAuth Configuration Fields, Default Values, and Descriptions

Configuration Field	Default Value	Description
addShadowUserOnLogin	On Note: This option should always be on.	Specifies that a local shadow user should be added once authentication is successful. This field is required.
attributeMapping.userName	By default, the value is email.	Specifies the attribute from the provider, which contains the user name. The value specified is

Configuration Field	Default Value	Description
		used by the <i>scopes</i> option. This field is required.
authUrl	No default value	Specifies the URL to the authorization endpoint of the third-party. This field is required.
discoveryUrl	No default value	Specifies the URL that is used to discover the provider and obtain information that is needed to interact with it. This field is optional. If a value is not specified, you must enter values for the issuer field and either the tokenKey or tokenKeyUrl field.
emailDomain	<i>domain_name1, domain_name2, domain_name3</i>	Specifies a comma-separated list of email domains of users that can sign on with this provider. It is used with identity provider (IdP) discovery. This field is optional.
issuer	<i>https://hostname/auth/realms/realm_name</i>	Specifies the principal that issued the token, specified as a case-sensitive string or URI. This value must match the issue claim in the token. This field is optional if you specify a value for the discoveryUrl field.
linkText	The default value is "Use your corporate credentials".	Specifies the text that should be displayed on the sign-in page. This field is optional.
name	No default value	Specifies the <i>redirect_uri</i> that is provided in the App registration that is created in the Okta

Configuration Field	Default Value	Description
		portal. This field is required. IMPORTANT Do not include a period or other special characters in the value specified for the name.
relyingPartyId	<i>account-name-OAuth</i>	Specifies the client ID that is registered with the provider. This field is required.
relyingPartySecret	No default value	Specifies the secret that is registered with the provider for the client ID. This field is optional.
scopes	The list should contain openid .	This option depends on what is defined for the <i>attributeMapping.userName</i> . The scope tells the provider which fields to retrieve from the provider. Depending on the provider, they might need to include a scope to get back the user name field. This field is required.
showLinkText	On	Specifies that the link text should be shown on the sign-in page. This field is required.
tokenKey	No default value	Specifies the HMAC key or RSA public key that is used to sign ID tokens. This field is optional if you specify a value for the discoveryUrl field. Note: If a value for the discoveryUrl field is not specified, either the tokenKey or tokenKeyUrl field must be specified.

Configuration Field	Default Value	Description
tokenKeyUrl	No default value	Specifies the URL to obtain the signing key. This field is optional if you specify a value for the discoveryUrl field. Note: If a value for the discoveryUrl field is not specified, either the tokenKey or tokenKeyUrl field must be specified.
tokenUrl	No default value	The URL to obtain tokens from the provider. This value is required.
type	By default, the value is <code>oidc1.0</code> .	Specifies the protocol type. This field is required. Note: SAS Viya requires an <code>id_token</code> in the authorization response from the provider. However, some providers return an <code>id_token</code> when the scope in the authorization request is <code>openid</code> and <code>response_type=token</code> . For those providers, use type <code>oauth2.0</code> .

5 Click **Save**.

6 Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

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

See Also

[“Authentication: OIDC with ISAM Scenario \(Linux Full Deployment\)” on page 99](#)

Configure IdP Discovery for OIDC

IdP discovery enables you to use multiple identity providers (IdPs) in your environment. IdP discovery uses the domain of the authenticating user's email address to automatically select the IdP to use.

You can complete the following steps to enable IdP discovery:

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 In the **Definitions** list, select **sas.logon.zone**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.logon.zone Configuration window, enable the *idpDiscovery.enabled* option.

Note: When the **idpDiscovery.enabled** option is enabled and a user enters an email address with a domain that matches one of the values that is specified in the **emailDomain** configuration field, the provider is used.

- 5 Click **Save**.
- 6 If you see an error that refers to a violation of a “Content Security Policy directive,” your program requires a change to the content security policy setting for SAS Logon Manager. Follow the steps that are described in [“Update the Content Security Policy” on page 48](#) in order to update the content security policy value for the SAS Logon Manager service. Append the host name of the IdP, such as “https://login.microsoftonline.com”, to the value of sas.common.web.security, as in this example:

```
default-src 'self'; style-src 'self'; font-src 'self' data:; frame-ancestors 'self';
form-action 'self' https://login.microsoftonline.com;
```

Note: Depending on your configuration, you might also need to add 'unsafe-inline' to the default-src definition.

See Also

[“IdP Discovery for OIDC and SAML” on page 81](#)

Configure SAML (Linux Full Deployment)

Overview

Before configuring the Security Assertion Markup Language (SAML) in a single tenant environment, you must generate an RSA private key in PKCS#1 format and a certificate. You can generate this yourself or use an existing one (for example, the private key and certificate used by the httpd server). For more information, see [“Generate a JWT Signing Key” in *Encryption in SAS Viya: Data in Motion*](#). Configuration for SAML typically follows this pattern:

- 1 “Configure SAS Viya as a SAML Service Provider” on page 34
- 2 “Configure the SAML IdP – Relying Party Configuration” on page 36
- 3 “Configure SAS Viya with Information about the SAML IdP” on page 36

For information about configuring SAML in a multi-tenant environment, see “[About Identity Management and Authentication in Multi-tenancy](#)” in *SAS Viya Administration: Multi-tenancy*.

Note: By default, SAS Viya allows only same-origin requests. Authentication requests from the SAML IdP might be seen as cross-origin. Therefore, the origin of the SAML provider might need to be added. For more information, see “[Configure Cross-Origin Resource Sharing](#)” on page 46.

Configure SAS Viya as a SAML Service Provider

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see “[Edit Authentication Configuration Instances](#)” on page 44.
- 2 In the **Definitions** list, select **sas.logon.saml**.

Note: If you change any of the sas.logon.saml properties, the new metadata must be provided to the Relying Party in the federated service. If it is not, the SAML connections might fail.

- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.logon.saml Configuration window, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:

Table 9 SAML Configuration Fields and Descriptions

Configuration Field	Description
entityBaseURL	The external URL for the SAS Logon web application in SAS Viya (for example, <code>https://hostname.example.com/SASLogon</code>). This field is required.
entityID	The unique ID that represents the service provider that is included in protocol messages between relying parties. Change from the default value that is pre-populated. This field is required.
maxAuthenticationAge	Specifies the maximum time (in seconds) between users’ initial authentication with the IdP and processing of an authentication statement. The default value is 864000. This field is optional.

Configuration Field	Description
serviceProviderCertificate	Paste a copy of the PEM-encoded (base64) certificate, which is used by the service provider. This field is required.
serviceProviderKey	Paste a copy of the PEM-encoded (base64) key, which is used by the service provider. This field is required.
serviceProviderKeyPassword	Provide the password for the service provider, or leave blank if there is no password. This field is optional.
setProxyParams	IMPORTANT This field should not be modified. The value should remain Off .
signatureAlgorithm	Specifies the algorithm for SAML signatures. Acceptable values are SHA1, SHA256, and SHA512. The default value is SHA256. This field is optional.
signMetaData	Specifies whether the local service provider should sign the metadata. This field is required.
signRequest	Specifies whether the local service provider should sign the SAML requests. This field is required.
socket.connectionManagerTimeout	Specifies the amount of time (in milliseconds) before the connection pooling times out for HTTP requests for SAML metadata. The default value is 10000. This field is optional.
socket.soTimeout	Specifies the amount of time (in milliseconds) before the read times out for HTTP requests for SAML metadata. The default value is 10000. This field is optional.
wantAssertionSigned	Specifies whether the assertions should be signed. This field is required.

5 Click **Save**.

6 Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

Configure the SAML IdP – Relying Party Configuration

You can either configure the relying party trust or supply the required information to your information technology support group, in order for them to add the relying party trust. Here is an overview of the steps to perform, if you configure the relying party trust. The steps might vary, depending on which tool you use for configuration.

- 1 If the IdP requires it, configure Transport Layer Security (TLS), if it has not already been configured. For more information, see [“Update Apache HTTP Server TLS Certificates and Cryptography” in Encryption in SAS Viya: Data in Motion](#).
- 2 Download the application metadata.xml file, which contains information about the service provider, or provide the `https://hostname/SASLogon/saml/metadata` link to your information technology support group.
- 3 Request that your information technology support group configure a relying party in the IdP.

Configure SAS Viya with Information about the SAML IdP

- 1 Complete the following steps in SAS Environment Manager:
 - a In the **Definitions** list, select **sas.logon.saml.providers**.
 - b In the top right corner of the window, click **New Configuration**.
 - c In the New sas.logon.saml.providers Configuration window, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:

Table 10 SAML External Provider Configuration Fields, Default Values, and Descriptions

Configuration Field	Default Value	Description
addShadowUserOnLogin	On Note: This option should always be set to On .	Add a local shadow user upon successful authentication. This field is optional.
assertionConsumerIndex	0	The index of the assertion consumer service to use from IdP metadata. This field is optional
authnContext	No default value	The comma-separated list of authentication contexts that are included in SAML

Configuration Field	Default Value	Description
		requests to the IdP. This field is optional.
emailDomain	No default value	Specifies a comma-separated list of email domains for users that can sign on with the SAML provider. It is used with IdP discovery. This field is optional.
idpMetadata	No default value	The metadata XML content. This can be useful if manual changes need to be made to the IdP metadata. This field is required.
linkText	Use your corporate credentials	The hyperlink to display on the sign-in page. This field is optional.
metadataTrustCheck	Off	Specify whether to trust the IdP certificate. This field is required.
name	No default value	Specifies a unique name for this provider. This field is required. IMPORTANT Do not include periods, spaces, or other special characters in the value specified for the name.
nameID	The field is populated with a default value.	The nameID format to expect for the user name. Verify with your information technology support group that the value is correct. This field is required.
showSamlLoginLink	On	Determines whether a link should be displayed on the sign-in page for this IdP. This field is required.

Configuration Field	Default Value	Description
skipSslValidation	Off	Specifies whether to skip the TLS validation of the certificate. This field is optional.

d Click **Save**.

- 2 Edit the *SAS-Viya-configuration-directory/etc/sysconfig/sas-javaesntl/sas-java-services-security* file, and uncomment the highlighted line in the following block:

```
if [ -f $truststore ]; then
    export java_global_option_truststore="-Djavax.net.ssl.trustStore=$truststore"
    export java_global_option_truststore_password="-
Djavax.net.ssl.trustStorePassword=changeit"
fi
```

- 3 Restart the SAS Logon Manager service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

Configure Cross-Origin Settings for SAML

The SAML protocol uses the `POST` binding to send a SAML response from the identity provider (IdP) back to the service provider (SP). The web browser treats this as a cross-origin request because it is initiated from the IdP. Security settings on the server must specifically instruct the web browser to send cookies to the SP. If the SP receives the SAML response without the session cookie, it cannot link it to the original SAML request and fails with an error.

To fix the error, complete the following steps:

- 1 In SAS Environment Manager, edit the CORS configuration instance. For more information, see [“Edit Configuration Instances” in SAS Viya Administration: Configuration Properties](#).
- 2 Select **sas.common.web.security.cors**.
- 3 Create or edit the **sas.common.web.security.cors** configuration.
 - a Select **SAS Logon Manager** from the list of services.
 - b Set **allowedOrigins** to the specific Origin header value from the SAML IdP.

You can determine this value by turning on developer tools in the browser and looking at the HTTP request headers on the first request back to SAS Viya after authenticating with the IdP. For Azure, this is usually `https://login.microsoftonline.com`.

Note: Do not use wildcards.

- c Click **Save**.
- 4 Create or edit the **sas.common.web.security.cookies** configuration.
- a Select **SAS Logon Manager** from the list of services.
 - b Set **sameSite** to **None**.
 - c Click **Save**.
 - d Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

Configure IdP Discovery for SAML

IdP discovery enables you to use multiple IdPs in a SAML environment. IdP discovery uses the domain of the authenticating user's email address to automatically select the IdP to use.

You can complete the following steps to enable IdP discovery:

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 In the **Definitions** list, select **sas.logon.zone**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.logon.zone Configuration window, enable the *idpDiscovery.enabled* option.

Note: When the **idpDiscovery.enabled** option is enabled and a user enters an email address with a domain that matches one of the values specified in the **emailDomain** configuration field, the provider is used.

- 5 Click **Save**.
- 6 If you see an error that refers to a violation of a “Content Security Policy directive,” your program requires a change to the content security policy setting for SAS Logon Manager. Follow the steps that are described in [“Update the Content Security Policy for SAML” on page 40](#) (below) in order to update the content security policy value for the SAS Logon Manager service. Append the host name of the IdP, such as “https://login.microsoftonline.com”, to the value of sas.common.web.security, as in this example:

```
default-src 'self' 'unsafe-inline'; style-src 'self'; font-src 'self' data:; frame-ancestors 'self'; form-action 'self' https://login.microsoftonline.com;
```

See Also

[“IdP Discovery for OIDC and SAML” on page 81](#)

Update the Content Security Policy for SAML

If the SAML identity provider (IdP) uses a `POST` binding, changes are required to the content security policy that is used by SAS Logon Manager. Follow the steps that are described in [“Update the Content Security Policy” on page 48](#) in order to update the Content-Security-Policy setting for the SAS Logon Manager service.

A few changes are needed for a SAML IdP. Include `'unsafe-inline'` in the `default-src`. In addition, append the URL of the IdP to the `form-action`. This URL should not be enclosed in quotation marks. Here is an example:

```
default-src 'self' 'unsafe-inline'; style-src 'self'; font-src 'self' data:; frame-ancestors 'self'; form-action 'self' https://URL-of-IdP
```

The value for `https://URL-of-IdP` is unique to each IdP. Some testing might be required to find the correct value.

TIP If you see an error when you attempt to log in to the SAS Viya platform, you might be able to determine this URL by examining the error in the developer tools console.

Configure Single Sign-On with Automatic Redirect

The following section provides instructions on how to configure single sign-on (SSO) so that it automatically redirects to an already configured third-party SAML or OIDC provider. SSO with either SAML or OIDC enables end users to access the SAS Viya web applications without having to interact with SAS Logon Manager.

Note: This configuration does not affect the command-line interfaces, nor does it affect SAS Studio 5.2 (Basic).

- 1 Add an entry in the `emailDomain` option for the SAML or OIDC provider. For SAML, this option is under the `sas.logon.saml.providers` definition. For OIDC, this option is under the `sas.logon.oauth.providers` definition.

Note: The string entered in the `emailDomain` option does not have to be the actual email domain for the end users. You can specify any string (for example, `matchme.com`) because the email domain is configured in the Apache HTTP Server. Therefore, this configuration works well when the end users have many different email domains and when the SAS administrator does not know the email domains of the end-users.

- 2 Configure the `login_hint` option.
 - a Add `login_hint` to the authorize request.

This requires a change to the Apache HTTP Server. Add a rewrite rule to the configuration to set the `login_hint` option on the requests to `/SASLogon/oauth/authorize`. In most cases,

the SAS Viya environment is accessed over HTTPS. This means that the required changes should be made in the SSL configuration file. For Red Hat Enterprise Linux this would be the `/etc/httpd/conf.d/ssl.conf` file. The new content is added to the end of the file before the closing `</VirtualHost>` tag.

Note: For deployments that are using HTTP, this should be placed in a new `.conf` file. The Apache HTTP Server processes the `.conf` files in alphabetical order and this one needs to occur before the proxy configuration in `proxy.conf`, so it should be named accordingly (for example, `login_hint.conf`). Adding the redirect rule in both places is also supported. Restart the Apache HTTP Server after making any changes to the configuration.

- b Enable `login_hint` for all authorize requests that are made to SAS Logon Manager, using one of the following methods:
 - To add `login_hint` to all authorize requests, use the following example commands:

```
# SSO for SAS Viya set login_hint option
RewriteEngine On
RewriteCond "%{QUERY_STRING}" !login_hint
RewriteRule "SASLogon/oauth/authorize" "/SASLogon/oauth/authorize?
login_hint=email_domain" [QSA,PT]
```

Note: Enter `/SASLogon/login` in the web browser to access the sign-in page directly and to sign in using LDAP credentials or the `sasboot` account.

The SAS Logon Manager sign-in page is not displayed for any users.

- To enable the SAS Logon Manager sign-in page when accessed from a specific IP address, use the following example commands:

```
# SSO for SAS Viya set login_hint option except for specific IP address
RewriteEngine On
RewriteCond "%{QUERY_STRING}" !login_hint
RewriteCond expr "!" -R 'ip_address'"
RewriteRule "SASLogon/oauth/authorize" "/SASLogon/oauth/authorize?
login_hint=email_domain" [QSA,PT]
```

Replace `ip_address` with the value for your environment.

- To enable the SAS Logon Manager sign-in page to be available to more than a single IP address, change the `RewriteCond` statement to use CIDR notation:

```
# SSO for SAS Viya set login_hint option except for specific IP Range
RewriteEngine On
RewriteCond "%{QUERY_STRING}" !login_hint
RewriteCond expr "!" -R 'ip_address/24'"
RewriteRule "SASLogon/oauth/authorize" "/SASLogon/oauth/authorize?
login_hint=email_domain" [QSA,PT]
```

Replace `ip_address` with the CIDR value for your environment.

- To add conditions to the `RewriteRule`, you can change the behavior based on the browser rather than on the IP address. To exclude a web browser, use the following:

```
# SSO for SAS Viya set login_hint option except for browser_type
RewriteEngine On
RewriteCond "%{QUERY_STRING}" !login_hint
```

```

RewriteCond "%{HTTP_USER_AGENT}" "!.*browser_type.*"
RewriteRule "SASLogon/oauth/authorize" "/SASLogon/oauth/authorize?
login_hint=email_domain" [QSA,PT]

```

To exclude Google Chrome, specify **Edg.** in place of *browser_type*. To exclude Microsoft Edge, specify **Chrome**.

Configure Operating-System Authentication with PAM (Linux)

You can use a pluggable authentication module to enable users of SAS Viya visual interfaces to access operating-system resources, such as file systems. Such access requires user accounts that are stored on the operating system (host accounts), whereas the visual interfaces require user accounts in LDAP or the equivalent.

During the deployment, default PAM configuration files are installed for both the CAS server and SAS Studio.

- 1 As a user with root authority, edit the *SAS-Viya-configuration-directory/etc/pam.d/service* file. For the CAS server, *service* is *cas*. For SAS Studio, *service* is *sasauth*.

The following information is displayed for the CAS server:

```

$ vi /etc/pam.d/cas
#%PAM-1.0
auth    include    password-auth
account include    password-auth
password include    password-auth
session include    password-auth

```

The following information is displayed for SAS Studio:

```

$ vi /etc/pam.d/sasauth
#%PAM-1.0
auth    include    password-auth
account include    password-auth

```

- 2 Make any modifications to the file that are necessary for your environment.
- 3 Save the file and exit.

Configure Authentication Options with SAS 9.4

Configure the SAS 9.4 Deployment

- 1 Log on to SAS Management Console and navigate to **Plug-ins** ⇒ **Application Management** ⇒ **Configuration Manager**.
- 2 Right-click **SAS Application Infrastructure** and select **Properties**.
- 3 Click **Advanced**, and then set the following property value:

ServiceUrl.Allowed

Specifies the address to where tickets should be sent on SAS Viya. The format of the address should be similar to the following: `http://hostname/SASLogon/**`.

Note: For SAS deployments prior to SAS 9.4M3, the *ServiceUrl.Allowed* property is not required.

- 4 Click **OK**.
- 5 Restart all instances of SASServer1 to pick-up the new property.

Configure the SAS Viya Deployment

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 In the **Definitions** list, select **sas.logon.sas9**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.logon.sas9 Configuration window, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:

Table 11 SAS 9.4 Configuration Fields and Descriptions

Configuration Field	Description
autoLink	<p>Specifies whether to automatically open the link to SAS 9.4 when the sign-in page is displayed. This field is optional.</p> <p>Note: If the <i>autoLink</i> property is enabled, the SAS Viya Logon Manager sign-in page is not displayed. End users are automatically redirected to SAS Logon Manager in SAS 9.4 to authenticate. End users cannot use the LDAP provider or equivalent.</p>
enabled	<p>Specifies whether to enable users to sign in using SAS 9.4 credentials. This field is required.</p>
linkText	<p>Specifies the hyperlink to display on the sign-in page. This field is optional.</p> <p>Note: By default, the end user is presented with a link at the bottom of the standard SAS Logon Manager sign-in page in SAS Viya. The text of the link is controlled by the <i>linkText</i> property. This default behavior means that end users can choose to authenticate using either SAS 9.4 or the LDAP provider or equivalent.</p>

Configuration Field	Description
sas9LogonUrl	Specifies the URL of SAS Logon Manager in SAS 9.4 (for example, <code>https://SAS9_hostname/SASLogon</code>). This field is required.
showLinkText	Specifies whether to display the link text on the sign-in page. This field is optional.
single.signOn.enabled	Specifies whether to redirect to SAS 9.4 for single sign-on. This field is required.
single.signOut.enabled	Specifies whether the local sign-out should also sign the user out of SAS 9.4. This field is optional.
viyaLogonUrl	Specifies the URL of SAS Logon Manager in SAS Viya (for example, <code>https://SASViya_hostname/SASLogon</code>). This field is required.

5 Click **Save**.

6 Restart the SAS Logon Manager Service:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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


Session Management Using SAS Environment Manager

Overview

The following sections provide information about customizing SAS Logon Manager and the user's session experience.

Edit Authentication Configuration Instances

Customize SAS Logon Manager settings using SAS Environment Manager. The following steps are common to most configuration procedures:


- 1 Log on to SAS Environment Manager, using your user ID or the ID of a user who is a member of the SAS Administrators group.
- 2 In the applications menu (☰), select **Administration** ⇨ **Manage Environment**.
- 3 In the navigation bar, click .
- 4 Select **Definitions** from the drop-down menu.

See Also

- [SAS Viya Administration: Configuration Properties](#)

Customize Sign-in, Sign-out, and Session Time-out Content


You can configure customized content that is displayed when users of SAS web applications sign in, sign out, or the session reaches the time-out interval. To enable the display of customize content, follow these steps:

- 1 In the **Definitions** list, select **sas.logon.custom**.
- 2 In the top right corner of the window, click .
- 3 In the New sas.logon.custom Configuration window, specify the URI that contains the custom content that you want to display. Here are the available fields:
 - **login**
 - **logout**
 - **timedout**

For a description of the properties, see “sas.logon.custom” in [SAS Viya Administration: Configuration Properties](#).

- 4 Click **Save**.

Customize Concurrent Sign-in Sessions

- 1 In the **Definitions** list, select **sas.logon.sessions**.
- 2 In the top right corner of the window, click .
- 3 In the New sas.logon.sessions Configuration window, you can set the following properties:

`maxConcurrentSessions`


Set this property to limit users to a certain number of concurrent sessions.

`rejectNewSessionsIfMaxExceeded`

When sessions are limited, the default behavior is to cause an existing session to expire and grant a new session to the user attempting to authenticate. To override this behavior and prevent a new session from being granted, set this property to *true*.

- 4 Click **Save**.

Configure the HTTP Session Time-out Interval

- 1 In the **Definitions** list, select **server**.
- 2 In the top right corner of the window, click .
- 3 In the New server Configuration window, complete the following:
 - a Select **SAS Logon Manager** from the **Services** drop-down list.
 - b Click **+**.
 - c In the **Name** field, enter `session.timeout`.
 - d In the **Value** field, enter the amount of time a session has to be idle before it times out, in seconds.
 - e Click **Save**.
- 4 Click **Save**.
- 5 Restart all services to reflect the new time-out interval. For more information, see [“Start and Stop All Servers and Services”](#) in *SAS Viya Administration: General Servers and Services*.

Disable Sign-ins

As a SAS administrator, you can disable sign-ins through operating system firewall rules or using LDAP. This disables new sessions, ends current sessions, and prevents others from using the deployment. For more information, see the appropriate documentation for your operating system.

Additional Authentication Topics

Configure Cross-Origin Resource Sharing

By default, SAS Viya allows only same-origin requests. If cross-origin requests are needed, complete the following steps:

- 1 In SAS Environment Manager, edit the CORS configuration instance. For details, see [“Edit Authentication Configuration Instances”](#) on page 44.
- 2 Select **sas.common.web.security.cors**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New `sas.common.web.security.cors` Configuration window, specify values that correspond to your environment. For a description of each field, see [“sas.common.web.security.cors”](#) in *SAS Viya Administration: Configuration Properties*.

Note: The specified value for the **allowedOrigins** field must be a comma-delimited list of URIs or an asterisk (*) to accept all origins. Partial wildcards are not supported (for example, `https://*.example.com`).

5 Click **Save**.

6 Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

Bypass SAS Logon Manager

Configuration options are available to enable users to bypass SAS Logon Manager when they authenticate. These settings are useful in a SAML or OIDC environment. When you configure one of these settings for your IdP, users are automatically redirected to your third-party SAML or OIDC IdP.

SAS Environment Manager includes the following configuration options under `sas.logon.zone`:

- **defaultIdentityProvider** - provides the name of the identity provider to which user logins are automatically redirected.
- **idpDiscovery.enabled** - discovers the appropriate external identity provider to which to redirect user logins by matching an email domain.

If you set the value of **defaultIdentityProvider** to match the value of either **sas.logon.oauth.providers.name** or **sas.logon.saml.providers.name**, users are automatically redirected from their browser to the associated third-party IdP, such as Microsoft Entra ID, bypassing SAS Logon Manager. If a user session is already established with the third-party IdP, the user is rapidly logged in to the SAS Viya web applications. If the user does not have an existing session with the IdP, the user is prompted to log in to the IdP and is then provided access to the SAS Viya web applications without requiring an additional login step.

As an alternative, the setting **idpDiscovery.enabled** is appropriate for environments with multiple IdPs. When **idpDiscovery.enabled** is set to **true**, users are prompted for their email address when they log in. SAS Logon Manager compares the text string after the "@" with the values configured for **emailDomain** in any OIDC or SAML providers in the environment. If a string match is found, SAS Logon Manager redirects the browser to the corresponding third-party IdP. Otherwise, SAS Logon Manager displays the standard login form.

Typically, you would not set both options under `sas.logon.zone`. Set the value to **defaultIdentityProvider** if you have a single OIDC or SAML IdP. If you have multiple OIDC or SAML IdPs, set the value to **idpDiscovery.enabled** so that users are authenticated based on their email address.

If you enable IdP discovery, you might see this error:

```
Refused to send form data to 'https://example.domain.com/SASLogon/login/idp_discovery' because it violates the following Content Security Policy directive: "form-action 'self'".
```

The error indicates that your program requires a change to the content security policy setting (CSP) for SAS Logon Manager. Follow the steps that are described in [“Update the Content Security Policy” on page 48](#) in order to update the CSP value for the SAS Logon Manager service. Append the host name of the OIDC or SAML IdP, such as “https://*.microsoftonline.com”, to the value of `sas.common.web.security`, as in this example:

```
default-src 'self'; style-src 'self'; font-src 'self' data:; frame-ancestors 'self';
form-action 'self' https://*.microsoftonline.com;
```

See Also

- [“IdP Discovery for OIDC and SAML” on page 81](#)
- [“Configure IdP Discovery for SAML” on page 39](#)


Update the Content Security Policy


The default content security policy that applies to SAS Viya is set up to be as secure as possible. In certain situations, you might need to adjust the content security policy settings. One of these settings, the HTTP content security policy `form-action` directive, restricts the URLs that can receive form submissions from a specified context. If you are calling or embedding SAS Viya endpoints from another application, you might need to add your endpoints to the `form-action` directive for the SAS Logon Manager service. You might also need to add the root context for your SAS Viya deployment. An example of adding the root context would resemble this format:

```
https://my.deployment.com/
```

Typically, the same values should be specified for the `allowed_origins` parameter of the [Cross-Origin Resource Sharing \(CORS\) configuration](#).

It is important to preserve all the existing settings in the content security policy and to add to it only where necessary. Because the content security policy differs slightly among services, you should never apply a new configuration globally; do not specify the `Global` option in SAS Environment Manager. Instead, you should update the content security policy for one service at a time, as needed. Take the following steps:

- 1 From SAS Environment Manager, navigate to the configuration definitions.
- 2 In the **Definitions** list, select **sas.common.web.security**.
- 3 In the properties window of the Definitions view, find the service to which you want to apply an adjusted content security policy value.
 - If the service is listed in the properties window, click  in order to edit the configuration.
 - If the service is not listed, in the top right corner of the window, click **New Configuration**.

In the New `sas.common.web.security` Configuration window, click , and select the service to which this configuration should be applied.

- In the Edit `sas.common.web.security` window, locate the **content-security-policy** value. This string determines the composition of the Content-Security-Policy HTTP header. Click in the field in order to append values to the string.

Valid values are listed in the Mozilla MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>.

IMPORTANT Each SAS Viya service potentially has its own content security policy that is specific to its unique requirements. Before modifying the content security policy, determine the current value that is used by the service and append additional values to the existing value.

If SAS Environment Manager is showing an existing configuration for the service, you can simply edit the configuration and make your changes. If the service does not have an existing configuration for the content security policy, you can determine the current value by calling the service from a web browser with developer tools enabled and examining the HTTP request headers.

IMPORTANT For the form-action directive only, if you specify a list of values, use spaces to separate them rather than commas.

- (For SAML environments only) If the SAML identity provider uses a `POST` binding, additional parameters are required. For more information, see “[Update the Content Security Policy for SAML](#)” on page 40.
- Click **Save**.
- Restart the affected services.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

```
sudo systemctl restart sas-viya-service-name-default
```

On Windows, in Windows Services Manager, right-click the service and select **Restart**.

Note: It might take several minutes to restart the services.

The CAS HTTP server uses its own content security policy that cannot be modified by default and is not controlled by the definition of **sas.common.web.security**. You can override the content security policy setting for the CAS HTTP server.

Register an OAuth Client ID

You might need to register a client for API access to the SAS Viya platform. For example, to use a SCIM identity provider, you must register an OAuth client for the identity provider and use the client credentials to generate an access token that grants access to APIs. The full task involves registering a new client ID and setting it up to obtain tokens using client credentials. You need to register a client only once.

You can register a client manually or by using the SAS Viya platform CLI. For a multi-tenant deployment, register the client on each tenant where it will be used.

Register a Client Manually

- 1 Obtain a token to register the new client ID and secret.

The example below uses a password grant, but other options are available. Use an account that is a member of the SASAdministrators group:

```
SASLOGON_URL=https://host-name.example.com; \
export BEARER_TOKEN=`curl -sk -X POST \
"${SASLOGON_URL}/SASLogon/oauth/token" \
-u "sas.cli:" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d 'grant_type=password&username=myusername&password=mypassword' | awk -F: '{print $2}'|awk -F\" '{print $2}'`
```

Note: The initial line of the curl command must be entered on one line. It is shown on more than one line for improved readability.

- 2 Verify that the command executed correctly by running the following command:

```
echo "The registration access-token is: " ${BEARER_TOKEN}
```

Output should be similar to the following:

```
The registration access-token is: eyJhbGciOiIhZi0leQhpfd_nZcVE9stGwqjJ4WzR2doHEd
```

- 3 Use the token to register the new client ID and secret.

The following curl command registers a client to obtain tokens using a password grant. Run this example command in a single-tenant environment:

Note: For a listing of the other grant parameters, see [Table 12 on page 51](#).

```
curl -k -X POST "${SASLOGON_URL}/SASLogon/oauth/clients" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $BEARER_TOKEN" \
-d '{
  "client_id": "client-id",
  "client_secret": "client-secret",
  "authorities": ["uaa.none"],
  "authorized_grant_types": ["client_credentials"]
}'
```

Note: The initial line of the curl command must be entered on one line. It is shown on more than one line for display purposes only.

The output should be similar to the following:

```
{"scope":["uaa.none"],"client_id":"your_client_id","resource_ids":["none"],
"authorized_grant_types":["client_credentials"],"autoapprove":[],
"authorities":["uaa.none"],"lastModified":1673983218606,"required_user_groups":[]}
```

For multi-tenant deployments, run the command with the highlighted header included:

```
curl -k -X POST "${SASLOGON_URL}/SASLogon/oauth/clients" \
-H "Content-Type: application/json" \
```

```
-H "Authorization: Bearer $BEARER_TOKEN" \
-H "X-Identity-Zone-Id: tenant" \
-d '{
  "client_id": "client-id",
  "client_secret": "client-secret",
  "authorities": ["uaa.none"],
  "authorized_grant_types": ["client_credentials"]
}'
```

The output for a multi-tenant deployment should be similar to the following:

```
{ "scope": ["uaa.none"], "client_id": "your_client_id", "resource_ids": ["none"],
  "authorized_grant_types": ["client_credentials"], "autoapprove": [],
  "authorities": ["uaa.none"], "lastModified": 1673983218606, "required_user_groups": [] }
```

Table 12 Parameters and Descriptions

Parameter	Type	Constraints	Description
client-id	String	Required	Specifies a unique client identifier.
client-secret	String	Required if the client uses the authorization_code or client_credentials grant types	Specifies a secret string used for authenticating as this client.
scope	Array	Optional (defaults to "uaa.none")	Specifies the list of scopes that are allowed for the client to obtain on behalf of users, when using any grant type other than "client_credentials". For most SAS Viya APIs, "openid" and "uaa.user" are sufficient. For client applications that use only the grant type "client_credentials" and therefore do not act on behalf of users, use the default scope "uaa.none".
authorities	Array	Optional (defaults to "uaa.none")	Specifies the groups that tokens get using the client_credentials grant_type.
authorized_grant_types	Array	Required	Specifies the list of grant types that can be used to obtain a token with

Parameter	Type	Constraints	Description
			this client. The types include authorization_code, password, implicit, and client_credentials.
redirect_uri	Array	Optional	Specifies the allowed URI pattern for redirect during authorization. Wildcard patterns can be specified using the Ant-style pattern.
autoapprove	[Boolean, Array]	Optional	Specifies the scopes that do not require user approval. Boolean values true and false apply to all scopes, otherwise a list can be provided.
access_token_validity	Number	Optional	Specifies the time, in seconds, to access token expiration after it is issued.
refresh_token_validity	Number	Optional	Specifies the time, in seconds, to refresh token expiration after it is issued.
allowedproviders	Array	Optional	Specifies a list of origin keys (aliases) for IdPs to which the client is limited. Null implies any IdP is allowed.
name	String	Optional	Specifies a human readable name for the client.
token_salt	String	String	Specifies a random string used to generate the client's revocation key. Change this value to revoke all active tokens for the client.

Register a Client Using the CLI

Use the `sas-admin` CLI to register an OAuth client and enable access to SAS Viya APIs.

The oauth plug-in to the `sas-admin` CLI is required. For more information, see [“Command-Line Interface: Preliminary Instructions” in SAS Viya Administration: Using the Command-Line Interfaces](#).

- 1 Log in to the SAS Viya CLI using an account that is a member of the SASAdministrators group. Here is an example that uses a profile:

```
sas-admin --profile profile-name auth login
```

- 2 Register the client by running the oauth plug-in to the `sas-admin` CLI.

The following command registers a client to obtain tokens using a password grant. Run the command without the `--grant-password` flag in order to display all of the grant parameters. For a listing of the other parameters, see [Table 12 on page 51](#).

```
https://host-name.example.com
sas-admin --sas-endpoint ${SASLOGON_URL} oauth register-client \
  --grant-password \
  --id client-id --secret client-secret
```

Note: Change *host-name* to specify the host name for the endpoint in your environment. The value for *client-secret* should be a unique secret value for registering the client. In a multi-tenant deployment, the SASLOGON_URL should include the tenant.

The output should be similar to the following:

```
Registering new client ' client-id ' with grant_type password...
client-id has been registered as a client of the SAS environment at https://hostname.example.com.
Clients can use client-id and client-secret to obtain access tokens.
OK
```

Obtain an Access Token

SAS Viya APIs have a security layer that requires an access token. The token authenticates the user and also contains the group memberships of the user, which is used to make authorization decisions and determine whether the user can access the endpoint or resource.

You can obtain an access token using an authorization code, a password, or client credentials.

Obtain an Access Token Using an Authorization Code

During this method, the end user is not required to give their password credentials to the client application. Instead, the user goes to SAS Logon Manager to get the client application an authorization code, which it can use to obtain an access token.

Note: A prerequisite is that the client ID is registered with the *authorization_code*, *grant_type*, and *redirect_uri* `urn:ietf:wg:oauth:2.0:oob`.

- 1 The end user approves access to get the authentication code.

- a Enter the following URL in your web browser and substitute the host name and client ID from your environment:

```
https://localhost/SASLogon/oauth/authorize?client_id=client-id&response_type=code
```

The SAS sign-in page is displayed.

- b Log on with your SAS user credentials.
 - c On the Authorize Access window, select the **openid** option, and any other required groups. Then click **Authorize Access**.
 - d On the Authorization Code window, copy the authorization code.
- 2 The client application gets an access token using the authorization code.

- a Run the following command:

```
curl -k https://localhost/SASLogon/oauth/token \
  -H "Accept: application/json" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=authorization_code&code=auth-code" \
  -u "client-id:secret"
```

Note: The access token is created once and needs to be refreshed when the token expires.

- b Create environment variables for the tokens, since they might be long:

```
export ACCESS_TOKEN="access_token"
export REFRESH_TOKEN="refresh_token"
```

- 3 Use the access token to call SAS Viya APIs. Once you have the access token, you can call any API, as long as your access token is valid and the user has access to the endpoint.

```
curl -k https://localhost/folders/folders?filter=isNull(parent)
  -H "Authorization: Bearer $ACCESS_TOKEN"
```

- 4 Use the refresh token to get a new access token. A prerequisite is that the client ID is registered with the *refresh_token* grant_type.

Here is an example:

```
curl -k https://localhost/SASLogon/oauth/token
  -H "Accept: application/json" \
  -H "Content-Type: application/x-www-form-urlencoded"
  -u "myclientid:myclientsecret"
  -d "grant_type=refresh_token&refresh_token=$REFRESH_TOKEN"
```

The access token is new, and the refresh token remains static. Use the new token for future REST calls. Make sure to replace the ACCESS_TOKEN variable with the new token. The access token has a default timespan that determines when it expires. Most applications deal with expiring and refreshing tokens programmatically. If you want to change the default expiry of an access token in SAS, see [“sas.logon.jwt” in SAS Viya Administration: Configuration Properties](#).

Obtain an Access Token Using Password Credentials

During this method, the end user gives their password credentials to the client application.

Note: A prerequisite is that the client ID is registered with the *password* grant_type.

- 1 Get an access token using password credentials. Here is an example:

```
curl https://localhost/SASLogon/oauth/token \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=password&username=username&password=password" \
  -u "client-id:secret"
```

- 2 Copy the value of the *access_token* attribute from the JavaScript Object Notation (JSON) in the response and save it.

Obtain an Access Token Using Client Credentials

Clients might obtain an access token using the client's credentials alone. This type of access token is not associated to any end user and is scoped according to the authorities registered to the client. This topic is covered in the next section. An end user is not involved in this flow, so there is no resource owner except the client itself.

Note: A prerequisite is that the client ID is registered with the *client_credentials*, *grant_type*, and any authorities that are required.

- 1 The client requests an access token using the client credentials.

```
curl -k https://localhost/SASLogon/oauth/token \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=client_credentials" \
  -u "client-id:secret"
```

- 2 SAS Logon Manager authenticates the client and issues an access token.


Enable Fallback Authentication

SAS Logon Manager in SAS Viya 3.5 supports fallback authentication when Kerberos is enabled. End users whose browser cannot authenticate with Kerberos will "fall back" to the default login page, where they can authenticate with a user name/password combination instead.

However, with some versions of SAS Logon Manager, fallback itself might fail because of a default content security policy setting. This setting was introduced in a 2022 SAS Viya hot fix. The updated content security policy can prevent the fallback script from running as expected. Instead of displaying the login form, SAS Logon Manager displays a blank page and attempts to redirect to an IdP for authentication.

In order to resolve the issue and enable fallback authentication, take the following steps:

- 1 From SAS Environment Manager, navigate to the configuration page. For more information, see ["Edit Authentication Configuration Instances" on page 44](#).
- 2 On the Configuration page, select **Definitions** from the drop-down list.
- 3 In the **Definitions** list, select **sas.common.web.security**.
- 4 In the top right corner of the window, click **New Configuration**.

- 5 In the New sas.common.web.security Configuration window, click  to edit the **Service** field, which specifies Global by default.

A search window displays.

- 6 Search for and select the **SAS Logon Manager** service. Click **OK**.
- 7 Scroll down to the **content security policy**. In the field, specify the following:


```
default-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self';
font-src 'self' data:; frame-ancestors 'self'; form-action 'self';
```
- 8 Click **Save**.
- 9 Restart the SAS Logon Manager service.

Create an Authinfo File

The authinfo file supplies a user name and password that is sent to CAS for authentication. For information about how to create an authinfo file, see [Create an Authinfo File](#).

Configure the SameSite Attribute

- 1 From SAS Environment Manager, navigate to the configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 In the **Definitions** list, select **sas.common.web.security.cookies**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.common.web.security.cookies Configuration window:
 - a Specify **Global** in the **Services** field.
 - b Specify a value for the SameSite property. Valid values are listed in [Table 13](#):

Table 13 Possible SameSite Property Values

Value	Description
Unset	The same-site cookie attribute is not set. This is the default value.
Strict	The browser does not send the cookie in any cross-site requests.
Lax	The browser sends the cookie in same-site requests and cross-site top-level GET requests.
None	The same-site cookie attribute is set and the cookie is sent in cross-site requests.

Value	Description
	Note: In order for SAS Visual Analytics SDK to work properly, the SameSite property must be set to None .

c Click **Save**.

5 Restart the SAS Logon Manager Service.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

Authentication: Concepts

Authentication Architecture

In a full deployment, authentication services are provided by SAS Logon Manager. SAS Logon Manager is based on the Cloud Foundry User Account and Authentication (UAA) server. The security architecture is built around Open Authorization (OAuth) and OpenID Connect (OIDC). By default, authentication is performed via a Lightweight Directory Access Protocol (LDAP) provider. Authentication support is also available for Kerberos, OAuth 2.0 with OIDC, and Security Assertion Markup Language (SAML).

Note: For Windows deployments, Kerberos is the only supported authentication mechanism for SAS Viya visual interfaces and configuration of the middle tier environment.

In a programming-only or a full deployment, host authentication is supported on both Linux and Windows systems. On Linux systems, you can configure the host to use only pluggable authentication modules (PAM).

The ability to identify users enables them to be authenticated. SAS recommends using an Identity and Access Management (IAM) system to synchronize identities among all SAS Viya user interfaces. In this guide, the term *IAM* refers to the system that makes users and groups available to SAS Viya by means of an LDAP server or through SCIM provisioning.

See Also

[“About Identity Management and Authentication in Multi-tenancy” in SAS Viya Administration: Multi-tenancy](#)

Authentication and SAS Viya Services

The following table lists the key services that are used in authentication in SAS Viya:

Table 14 SAS Viya Services

Service Name	Description
SAS Logon Manager	Provides both an end-user interface for authentication and internal authentication to other services. Enables single sign-on within the SAS Viya environment between services. Enables single sign-on to the SAS Viya environment through configuration of third-party software.
Identities service	Provides the user and group information to other services. Reads user and group information from the IAM system.
Authorization service	Provides authorization information to other services.
Launcher Service	Provides connection and authentication services. Resolves the credentials that are used when authenticating to the SAS Launcher Server.
CAS server	Authenticates end users launching CAS sessions by way of the CAS controller.
SAS 9.4	Supports several mechanisms for coupling authentication with SAS 9.4.
SAS Studio 5.2 (Basic)	Leverages the SAS Object Spawner to authenticate users accessing SAS Studio 5.2 (Basic).

In-bound and Out-bound Authentication

In-bound Authentication

In-bound authentication is the authentication of the end user to the environment. In-bound authentication provides an internal OAuth token and group membership information in the OAuth token. If Kerberos authentication is used, a delegated Kerberos credential is also stored.

The client browser must be configured to delegate credentials. Otherwise, an error message is displayed. To prevent the message, set the **disableDelegationWarning** option. For more information, see [the steps to configure the Kerberos authentication properties on page 6](#).

Out-bound Authentication

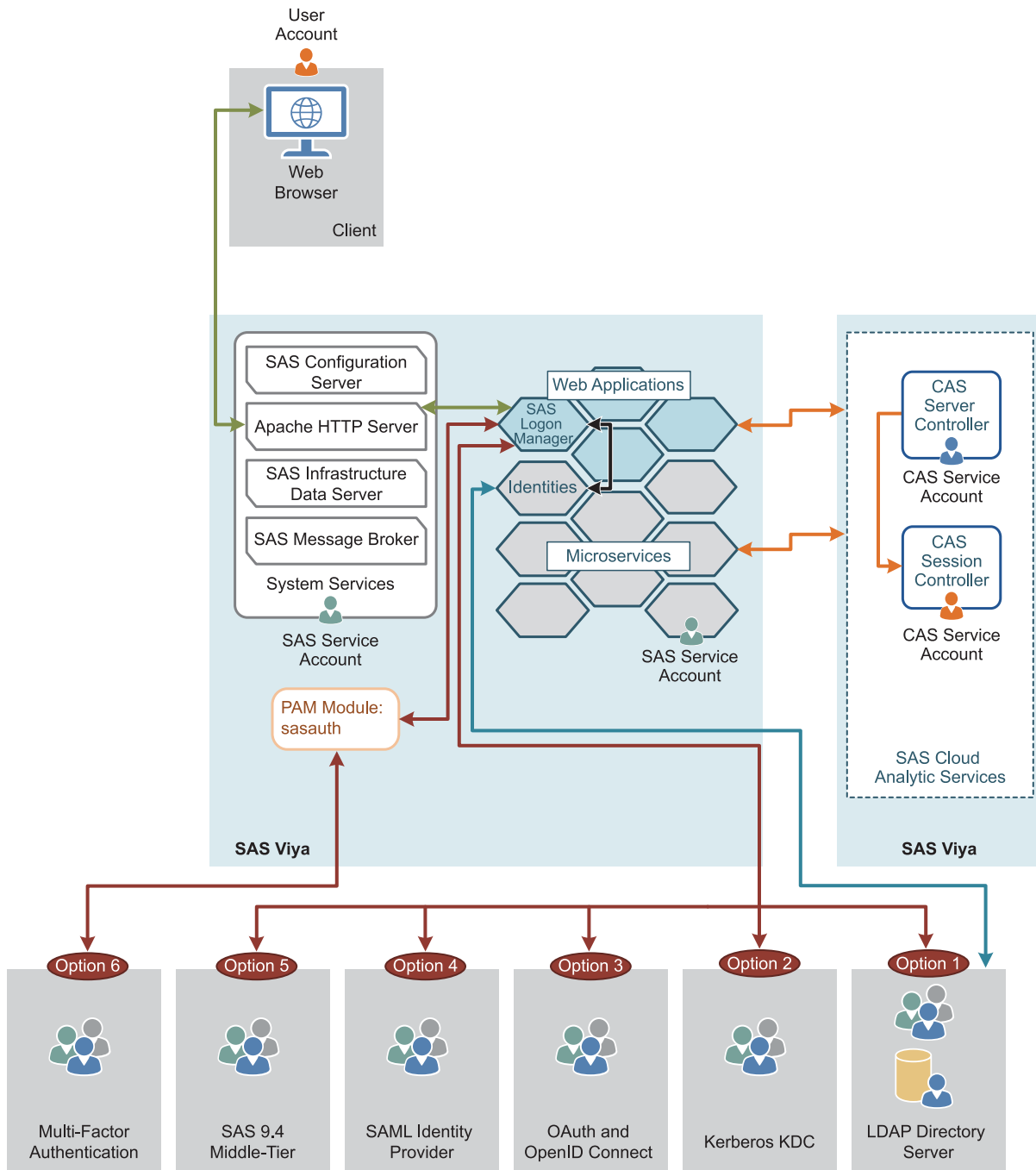
Out-bound authentication is the authentication of the SAS process to a downstream process. Out-bound authentication occurs after the end user is initially authenticated to SAS Logon Manager. Out-bound authentication occurs to the CAS server, SAS Compute Server (through SAS Launcher Service), and then onto external resources, such as Secured Hadoop environments.

Authentication Options

Authentication for Visual Interfaces

With visual interfaces, users are authenticated through SAS Logon Manager. SAS Logon Manager is a web application that handles all authentication requests for SAS web applications and is accessed via the Apache HTTP Server.

The following figure shows how a user is authenticated on Linux to SAS Logon Manager and the supported authentication mechanisms.



The following protocols are available for you to configure for authentication:

- The first option is a Lightweight Directory Access Protocol (LDAP) provider. This is the default configuration. In this configuration, SAS Logon Manager displays a sign-in page and submits the entered credentials to LDAP. The identities service verifies users in LDAP. For more information, see [“LDAP Authentication \(Full Deployment\)”](#) on page 64.

Note: Rather than using an LDAP identity provider, you can instead configure an IAM system that uses the System for Cross-domain Identity Management (SCIM) to provide identity information to the identities service.

- The second option is Kerberos. In this configuration, SAS Logon Manager uses SPNEGO to authenticate users against the Kerberos Key Distribution Center (KDC). The identities service verifies users in your IAM system. For more information, see [“Kerberos Authentication \(Full Deployment\)”](#) on page 64.

Note: CAS server sessions run as the end user only when using Kerberos delegation. On Linux systems, the user must be a member of the CASHostAccountRequired custom group. On Windows systems, users are automatically delegated.

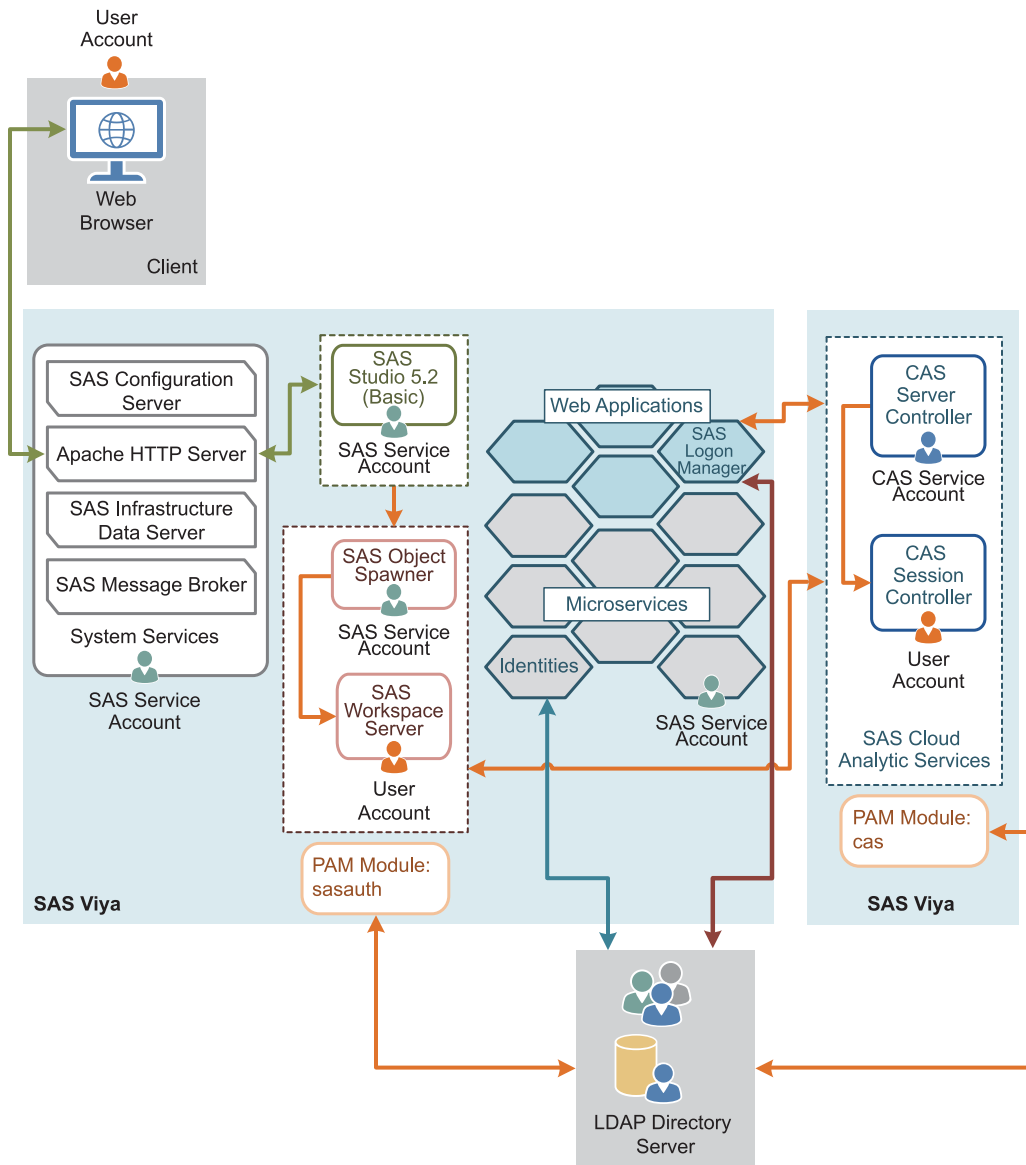
- The third option is OAuth 2.0 and OIDC. In this configuration, SAS Logon Manager uses OAuth 2.0 and OIDC to authenticate users. The identities service verifies users in your IAM system. For more information, see [“OAuth and OIDC Authentication \(Linux Full Deployment\)”](#) on page 74.
- The fourth option is Security Assertion Markup Language (SAML). In this configuration, SAS Logon Manager uses a SAML provider to authenticate users. The identities service verifies users in your IAM system. For more information, see [“SAML Authentication \(Linux Full Deployment\)”](#) on page 75.
- The fifth option is SAS 9.4. In this configuration, SAS Logon Manager supports single sign-on and single sign-off with SAS 9.4. The identities service verifies users in your IAM system. For more information, see [“SAS 9.4 Authentication”](#) on page 77.
- The sixth option is pluggable authentication module (PAM) to support multi-factor authentication. In this configuration, SAS Logon Manager uses the operating system PAM stack. The identities service verifies users in your IAM system. For more information, see [“PAM Authentication \(Linux\)”](#) on page 79.

With all six options, the connection to the CAS server is performed using internal OAuth tokens that are generated by SAS Logon Manager. In most cases, the session that is started by the CAS controller runs on the operating system as the same user who launched the CAS operating system service. This defaults to the cas account.

Authentication for Programming Interfaces

Overview of Programming Interfaces

The following figure shows how a user is authenticated on Linux while using programming interfaces.



In a deployment with programming interfaces, the user's credentials are sent to SAS Studio through the Apache HTTP Server. Then SAS Object Spawner uses pluggable authentication module (PAM) configuration files on the host to validate the user ID and password. The user ID and password can correspond to a local account on the host or, depending on the PAM configuration, an account in the IAM system. Once the user is authenticated, SAS Workspace Server is started. The PAM configuration file for SAS Studio is named `sasauth` and includes the password module.

SAS Workspace Server connects to the CAS environment using the user ID and password that were used to start SAS Workspace Server. However, if the `AUTHINFO=` option is specified, it is used to find credentials to connect to CAS. For more information about the `AUTHINFO=` option, see [AUTHINFO= SAS system option](#).

The CAS controller uses its own PAM configuration to validate the user's credentials and launch the session process as the user. The PAM configuration file for CAS is named `cas`, and it also includes the password module.

The CAS controller uses the user ID and password to obtain an internal OAuth token from SAS Logon Manager. The user ID and password must be valid in the LDAP provider that is configured for SAS Logon Manager. Otherwise, CAS cannot obtain an OAuth token, and the session fails.

PAM for SAS Studio (`sasauth`), PAM for CAS (`cas`), and SAS Logon Manager should all use the same or equivalent identity providers. Otherwise, attempts to connect might fail. SAS recommends synchronizing user identities using an IAM system.

Programming Interfaces with Symmetric Multiprocessing CAS Server

In a symmetric multi-processing (SMP) environment, a CAS server consists of a controller and a worker that run on a single machine. Authentication proceeds as follows:

- 1 The SAS Viya user connects to SAS Studio 5.2 (Basic) and enters a user name and password on the sign-in page. SAS Studio is proxied by the Apache HTTP Server.
- 2 SAS Studio 5.2 (Basic) passes the user name and password to SAS Object Spawner to start the SAS Workspace Server for the user.
- 3 SAS Object Spawner uses the PAM configuration that is defined in `/etc/pam.d/sasauth` to validate the user name and password. It launches SAS Workspace Server as the end user.
- 4 The SAS Studio user submits code to start a CAS session. SAS Workspace Server passes the user name and password to the CAS controller.
- 5 The CAS controller connects to SAS Logon Manager to obtain an OAuth token by presenting the user's user name and password.
- 6 SAS Logon Manager validates the user name and password. SAS Logon Manager also connects to the identities service to obtain group information for inclusion in the OAuth token.
- 7 The identities service connects to the LDAP provider with a simple BIND operation. It uses stored credentials for a service account and regularly connects to refresh the cache of users and groups, which is stored in SAS Infrastructure Data Server.
- 8 SAS Logon Manager returns the OAuth token to the CAS controller.
- 9 The CAS controller uses the PAM configuration in `/etc/pam.d/cas` to validate the user name and password and launches the CAS controller as the end user.

See Also

- [“Single-machine CAS Server” in SAS Viya Administration: SAS Cloud Analytic Services](#)
- [“Multiple CAS Servers” in SAS Viya Administration: SAS Cloud Analytic Services](#)

Programming Interfaces with Massively Parallel Processing CAS Servers

In a massively parallel processing (MPP) environment, a distributed CAS server consists of one controller, one or more workers, and optionally, one or more backup controllers. Each component runs on a separate machine. The authentication process for MPP CAS is essentially the same as for SMP CAS, with the following key differences:

- Initial communication between the CAS controller and CAS workers occurs using SSH.
- Ongoing communication does not use SSH.
- A worker process is launched on each CAS worker as the end user.
 - The CAS controller authenticates the end user with PAM.

- ❑ The CAS controller generates an internal identity token after authenticating the end user.
- ❑ The internal identity token is used to launch the CAS worker processes.
- ❑ PAM is not used on the CAS worker nodes.

See Also

[“Multiple CAS Servers” in SAS Viya Administration: SAS Cloud Analytic Services](#)

Concepts: Authentication Mechanisms

LDAP Authentication (Full Deployment)

Overview of LDAP

In SAS Viya, LDAP can be used for both identifying and authenticating users. Third-party LDAP server implementations are supported, including Microsoft Active Directory and OpenLDAP.

How It Works in SAS Viya

LDAP is the default authentication mechanism for SAS Viya. By default, SAS Logon Manager performs form-based LDAP authentication. When users log in by supplying credentials in the login form, SAS Logon Manager authenticates them by making a direct connection to an LDAP provider.

For all authentication methods, SAS Viya requires information about the user accounts that can access SAS Viya and their group memberships. An IdP serves as the source of information about user identities with LDAP authentication. The method that is used to supply this information to SAS Viya can be LDAP or SCIM. The identities service obtains and manages this information for SAS Logon Manager.

For information about configuring LDAP, see [Configure the Connection to Your Identity Provider](#).

Kerberos Authentication (Full Deployment)

Overview of Kerberos

Kerberos is a network authentication protocol that is used to verify user or host identity. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a service (and vice versa) across an insecure network connection. During Kerberos authentication, a user's credentials (user ID and password) are not sent over the network. Instead, both the client and the service use the credentials that were supplied as a key in an encryption algorithm to encrypt the message that is sent between the client and the service. If the client sends an encrypted message, and the service uses the same key to decrypt the message, it is proven that the credential is known without having to transmit the credentials.

In SAS Viya, the visual interfaces are SAS Environment Manager and CAS Server Monitor. SAS Environment Manager can be enabled to support Kerberos authentication. Conversely, CAS Server Monitor does not support Kerberos authentication.

Key Terms

Table 15 Term Definitions

Term	Definition
Client	An application that is attempting to connect to and access a resource, on behalf of a user. Resources include reports that are viewed, services that are accessed, and databases that are queried. In SAS Viya, the client is the web browser.
Service	A service, or server, that hosts a resource the user wants to connect to. The service must be able to validate the service tickets presented by the client.
Key Distribution Center	A trusted third party within Kerberos that verifies the authenticity of the client and service. Both the client and service must trust the KDC. In addition, end users and services must register with the KDC.
Service Principal Name	A unique name that is used to identify a web service that is running on a server. Before a service principal name (SPN) can be used, it must be registered. Every web service that uses Kerberos authentication needs to have an SPN set for it so that clients can identify the server on the network. An SPN usually matches the pattern of <code>HTTP/hostname.example.com</code> .
Keytab File	A file containing pairs of Kerberos principals and encrypted keys. The keys are associated with a password for the principal. The principals are SPNs. Keys can use different encryption algorithms. For a single principal, you might have several entries that correspond to each encryption type.
Ticket-granting ticket	An encrypted identification file that is valid for a limited amount of time. After a user is authenticated, this file is granted to a user for data traffic protection by the KDC. The TGT file contains the session key, its expiration date, and the user's IP address.

How It Works in SAS Viya

When you configure SAS Logon Manager for Kerberos authentication, the default LDAP provider is no longer used for authentication to SAS Logon Manager. Kerberos provides users with single sign-on capabilities from the browser on their desktop. Single sign-on allows users to access the SAS Viya visual interfaces without being prompted to enter their credentials.

For information, see [“Configure Kerberos \(Linux Full Deployment\)”](#) on page 4.

Integrated Windows Authentication

Integrated Windows Authentication (IWA) uses Kerberos authentication and is a Microsoft technology that is used in an environment where users have Windows domain accounts. With IWA, the credentials are hashed before being sent across the network. The client browser proves its knowledge of the password through a cryptographic exchange with the web application server. When IWA is used in conjunction with Kerberos, IWA enables the delegation of security credentials. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection.

Kerberos Authentication with CAS Scenarios

Different scenarios in which a user's credentials are used to access a Hadoop environment that is secured by Kerberos are supported. The following table provides an overview of each use case and links to additional information.

Table 16 *Kerberos Scenarios*

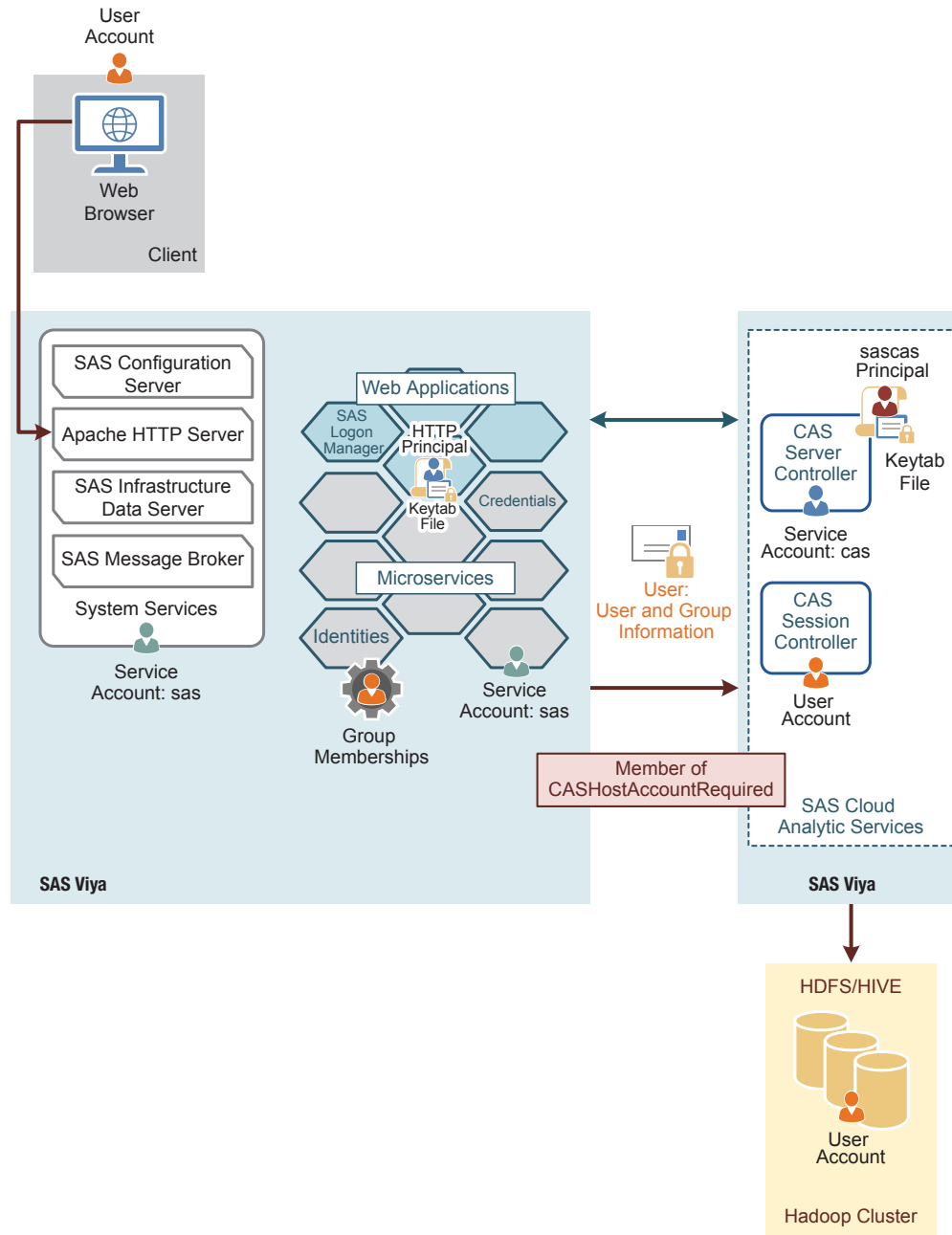
End-User Client	Connection to SAS Viya	Who Runs CAS Session	Connection to Hadoop
SAS Viya visual interface See “Kerberos in SAS Viya Visual Interface (Delegation)” on page 67.	Kerberos (delegation)	End user	End user
SAS Viya visual interface See “Kerberos in SAS Viya Visual Interface (Outbound from CAS)” on page 68.	Kerberos	Service account (cas)	Service account (sascas)
SAS Viya visual interface See “Kerberos in SAS Viya Visual Interface (Leveraging Stored Credentials)” on page 69.	Kerberos	End user	End user
SAS Viya programming interface See “Kerberos in SAS Viya Programming Interface with User Credentials” on page 69.	User ID and password	End user	End user
SAS 9.4 See “Kerberos in SAS 9.4 with User Credentials” on page 70.	User ID and password	End user	End user
SAS 9.4 See “Kerberos in SAS 9.4 with Delegation” on page 71.	Kerberos (delegation)	End user	End user
SAS 9.4	One-time password	Service account (cas)	Service account (sascas)

End-User Client	Connection to SAS Viya	Who Runs CAS Session	Connection to Hadoop
-----------------	------------------------	----------------------	----------------------

See “Kerberos in SAS 9.4 with One-Time Password” on page 71.

Kerberos in SAS Viya Visual Interface (Delegation)

The following figure illustrates this scenario:



Kerberos delegation to CAS, or user delegation, is a feature that allows a SAS Viya application to reuse the end-user credentials to access Kerberized systems. Delegation allows a server to forward a

user's credentials to the CAS server where they can be used to access other Kerberized services, such as Hadoop. By default, user delegation is not enabled and must be configured.

In this scenario, membership in the CASHostAccountRequired group notifies CAS that the user session needs to be launched under the user's operating system account and that Kerberos delegation needs to take place. The user's delegated Kerberos ticket can then be used for access to Kerberized services as himself or herself.

Note: On Windows, user sessions are launched under the user identity. However, in some situations, this is not possible. A session can also be launched under the CAS service account if a session cannot be launched under the user identity and the user requesting the session is able to assume the Superuser role.

See Also

- [“Configure Kerberos for the CAS Server” on page 8](#)
- [“The CASHostAccountRequired Custom Group” in *SAS Viya Administration: Identity Management*](#)

Kerberos in SAS Viya Visual Interface (Outbound from CAS)

Note: This scenario is currently not supported on Windows.

In this scenario, the inbound authentication is a mechanism other than Kerberos, but the outbound authentication is performed using Kerberos. For inbound authentication, you can use any authentication mechanism that is supported by SAS Logon Manager, such as SAML or OAuth and OIDC.

A registered principal for the CAS server is required. This is *not* the account that is running the CAS server. The default principal name is `sascas/cas_controller_hostname`. An alternate principal name must be specified using the `CAS_SERVER_PRINCIPAL` environment variable. In addition, the principal must be mapped to a valid Hadoop user and permission granted in Hadoop.

A Kerberos keytab file for the service account is also required. It should contain only the credentials of the service account. The default location for the keytab is `/etc/sascas.keytab`. An alternative location must be defined using the `KRB5_KTNAME` environment variable.

Here is a list of the implications of outbound Kerberos with any other authentication mechanism used by SAS Logon Manager:

- Access to the secured Hadoop cluster is as the principal provided to CAS. No end-user credentials are available for access to Hadoop.
- Kerberos credentials can be automatically renewed by SAS. CAS initializes the credentials using the keytab.
- Authorizations that are set in SAS Environment Manager still apply to the end user. These authorizations are the only permissions applied since all access to Hadoop is as the service account.

Kerberos in SAS Viya Visual Interface (Leveraging Stored Credentials)

Note: This scenario is currently not supported on Windows.

In this scenario, the inbound authentication is a mechanism other than Kerberos. Stored credentials are used for outbound authentication, which is performed using Kerberos. For inbound authentication, you can use any authentication mechanism that is supported by SAS Logon Manager, such as SAML or OAuth and OIDC.

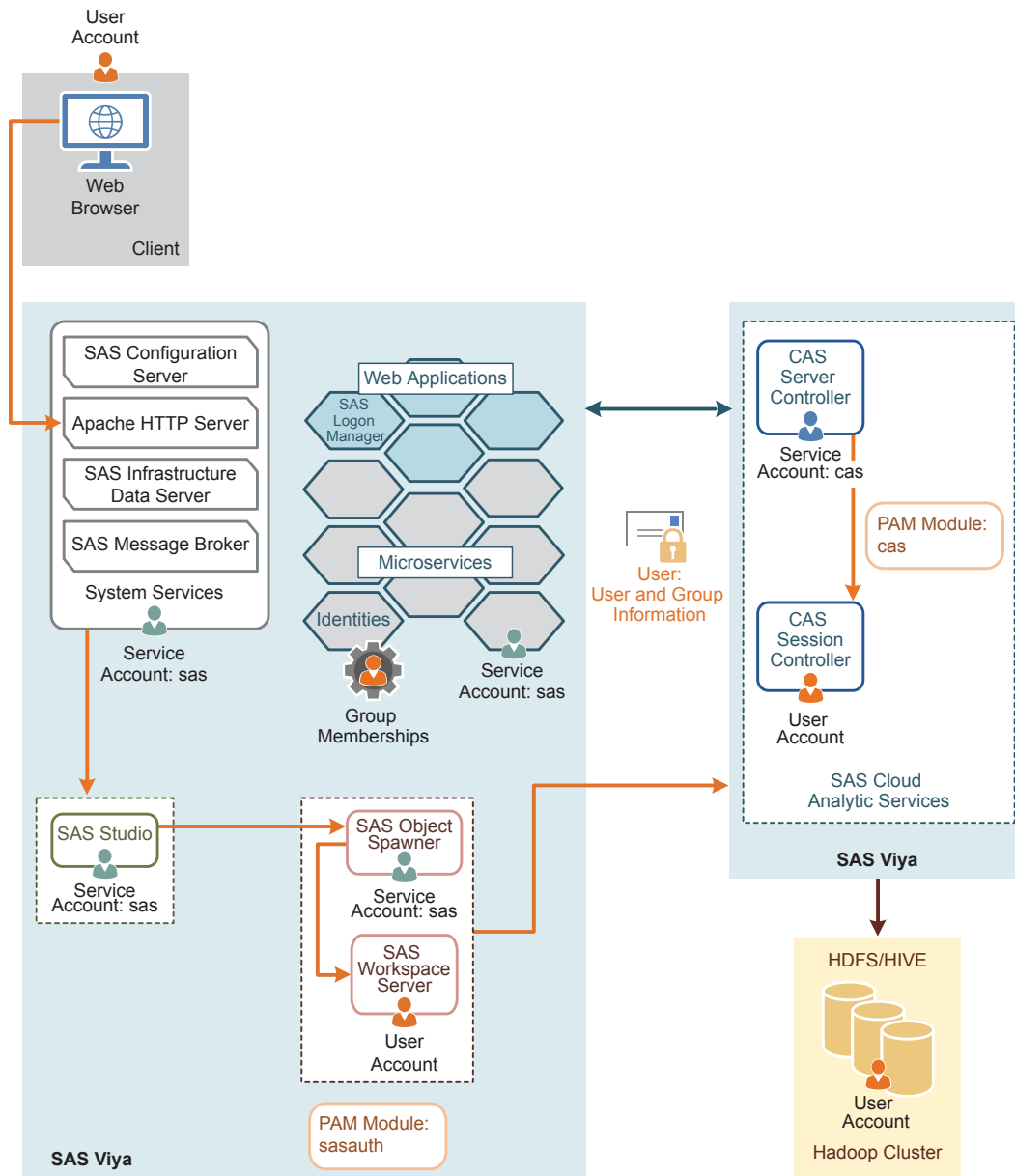
A credential must be stored for the end user. This can either be individually or from a group membership. A custom group must also be defined in SAS Environment Manager. The group can be named anything, but the ID must be CASHostAccountRequired. In addition, the operating system must generate the Kerberos credentials through PAM. SAS recommends that you use the System Services Security Daemon (SSSD).

Here is a list of the implications of outbound Kerberos with any other authentication mechanism used by SAS Logon Manager and a stored credential:

- Access to Secured Hadoop cluster is as the stored credentials. This could be an individual account or a shared account.
- Kerberos credentials might be automatically renewed by CAS. CAS attempts to renew the credentials. Alternatively, the operating system can provide options for renewal of credentials. For example, SSSD monitors and renews credentials before they expire.
- Authorizations set in SAS Environment Manager still apply to the end user.

Kerberos in SAS Viya Programming Interface with User Credentials

The following figure illustrates this scenario:



In this scenario, there is a full deployment and a user that provides his or her user ID and password to CAS. CAS uses its own pluggable authentication modules (PAM) configuration to validate the user's credentials and launch the CAS controller process running as the user. In addition, the CAS controller also uses the user ID and password to obtain an OAuth token from SAS Logon Manager. The OAuth token provides the user's group memberships from the Identities service. These memberships are essential in enforcing access control.

The PAM stack is configured to generate a Kerberos credentials cache during authentication. The resulting cache can be used to access Hadoop as the user.

Depending on the deployment options that you chose, users who access both the programming interface and the visual interface might have different access to Hadoop.

Kerberos in SAS 9.4 with User Credentials

In this scenario, end users provide their credentials to access SAS 9.4. SAS Workspace Server running SAS 9.4 is launched using a user ID and password, which are cached when SAS is launched.

This enables SAS Workspace Server to use these cached credentials when connecting to CAS. The user credentials can also be provided by other sources in a SAS 9.4 environment, such as SAS Metadata Server or an authinfo file in the user's home directory, because the process on the CAS controller is the same.

The user ID and password are validated through the PAM stack on the CAS controller and are used to generate an internal OAuth token from SAS Logon Manager. The PAM stack is responsible for initializing the Kerberos credentials for users. These Kerberos credentials are placed into a Kerberos Ticket cache, which makes them available to the CAS session for the connection to the secured Hadoop environment. The different sessions within SAS 9.4, SAS Viya, and the secured Hadoop environment run as the user.

Kerberos in SAS 9.4 with Delegation

In this scenario, SAS 9.4 is configured for Kerberos authentication. SAS Workspace Server running SAS 9.4 is launched using Kerberos credentials and the service principal for SAS Object Spawner running SAS 9.4 must be trusted for delegation. A Kerberos credential for the end user is available to SAS Workspace Server, which can be used to request a service ticket for the connection to CAS. CAS is provided with a Kerberos keytab and principal that it can use to validate this service ticket. Validating the service ticket authenticates the SAS 9.4 end user to CAS. The principal for CAS must also be trusted for delegation. CAS session must have access to the Kerberos credentials of the SAS 9.4 end user.

The Kerberos credentials that are made available to CAS are used to make a Kerberized connection to SAS Logon Manager running SAS Viya to obtain the SAS Viya internal OAuth token. Therefore, SAS Logon Manager running SAS Viya must be configured to accept Kerberos connections. For information about the configuration property that must be configured, see [“sas.logon.kerberos” in SAS Viya Administration: Configuration Properties](#). In addition, the Kerberos credentials for the SAS 9.4 end user are used to connect to the secure Hadoop environment.

Since all the principals are trusted for delegation, the SAS 9.4 end user can be authenticated using Kerberos with each component in the SAS Viya and SAS 9.4 integrated environment. Through the use of Kerberos authentication, the SAS 9.4 end user is authenticated in to CAS and out to the secure Hadoop environment.

Kerberos in SAS 9.4 with One-Time Password

In this scenario, the SAS 9.4 session can be a SAS Stored Process Server, SAS Pooled Workspace Server, or SAS Workspace Server using server launch credentials. The SAS 9.4 session is not running as the end user and does not have access to the end-user credentials. You can still connect to CAS and to the secured Hadoop environment by configuring one-time passwords generated by SAS Metadata Server running on SAS 9.4. SAS Metadata Server running on SAS 9.4 must be aware of CAS. This is done by creating a CAS server definition in SAS Metadata Server, using the AUTHDOMAIN= argument. For more information, see [CAS Statement: Summary of Optional Arguments](#).

The SAS Viya environment must be able to validate the one-time password that is used to connect to CAS. When CAS receives the one-time password during the connection, it is sent to SAS Logon Manager running on SAS Viya for validation and to obtain a SAS Viya internal OAuth token. SAS Logon Manager running on SAS Viya must be configured to enable this validation. For information about the configuration property that must be configured, see [“sas.logon.sas9” in SAS Viya Administration: Configuration Properties](#). SAS Logon Manager running on SAS Viya then passes the one-time password to SAS Web Infrastructure Platform running on SAS 9.4 to validate the password.

After the one-time password is validated, a SAS Viya internal OAuth token is generated and passed back to CAS.

CAS does not have access to the end-user credentials. Therefore, the session that is created is run using the account that is used to launch the controller process. By default, this account is cas. Since the end-user credentials are not available, the Kerberos credentials that are initialized for the session are from the Kerberos keytab provided to CAS. The connection to the secured Hadoop environment is made using those Kerberos credentials of the principal assigned to CAS.

Fallback Authentication

Fallback authentication is available when SAS Logon Manager is configured for Kerberos authentication. Fallback authentication is enabled by default. Once Kerberos authentication is configured, no additional configuration is required to use fallback authentication.

Fallback occurs when the browser is unable to perform Kerberos authentication. At this point, other authentication mechanisms (such as LDAP, SAML, PAM, or OAuth and OIDC) that are supported by SAS Logon Manager can be attempted. Multiple authentication methods can be used concurrently, in the same environment.

SAS Viya supports a custom fallback authentication security module. When the browser is not configured to perform Kerberos authentication, it falls back to the standard sign-in page and any other authentication mechanisms that are configured.

Note: Google Chrome attempts NTLM authentication before falling back to another authentication mechanism. A window might be displayed for the user to enter credentials. If this happens, they can cancel out of the dialog box to initiate fallback to the sign-in page.

If the user accesses the `/SASLogon/login` page in their browser (instead of being redirected to it), they will always get the sign-in page. The only way to initiate Kerberos authentication is to be redirected from another application to SAS Logon Manager. Therefore, if a user does not want to use Kerberos (for example, because their browser does not support it), they can bookmark the sign-in page to always fall back without the additional prompt.

With some versions of SAS Logon Manager, fallback authentication might fail because of a default content security policy setting. This setting was introduced in a 2022 SAS Viya hot fix. The updated content security policy can prevent the fallback script from running as expected and requires some configuration in SAS Environment Manager. For more information, see [“Enable Fallback Authentication” on page 55](#).

See Also

- [“Configure Kerberos \(Linux Full Deployment\)” on page 4](#)
- [“Authentication for Visual Interfaces” on page 59](#)

Kerberos Constrained Delegation

SAS Viya 3.5 introduces support for Kerberos constrained delegation. Kerberos constrained delegation can be used to authenticate in the following ways:

- to both CAS and SAS Compute Server from the SAS Viya 3.5 web applications

- from CAS to authenticate to data sources or from SAS Compute Server to CAS or data sources
- through a SAS 9.4 environment to CAS and onto your data source

Constrained delegation differs from unconstrained delegation in the following ways:

- The service does not require the user to forward either the Ticket-Granting Ticket (TGT) or the proxy ticket.
- The user does not need to authenticate using Kerberos, and the user does not need to have a TGT or a proxy service ticket.
- Windows Local Group policy can be used to limit the services that can be delegated.
- The client has no control over whether a service can delegate on behalf of the user. The client does not request delegation, nor does it pass a forwardable TGT to the service.
- The client cannot detect that delegation is, or has been, performed.

Kerberos constrained delegation, or Service for User (S4U), is a Microsoft extension to the Kerberos protocol. S4U provides two extensions to the Kerberos protocol. Together, these extensions allow a service to obtain a Kerberos service ticket on behalf of a user.

- The Service for User to Proxy (S4U2proxy) extension allows a service to obtain a service ticket on behalf of a user to a different service.
- The Service for User to Self (S4U2self) extension allows a service to obtain a Kerberos service ticket to itself.

The following table compares the two types of Kerberos constrained delegation:

Table 17 *Kerberos Constrained Delegation*

Types of Kerberos Constrained Delegation	Constraints	Kerberos Realm
traditional	The constraints are defined against the front-end service and control what back-end services the front-end service can delegate credentials to.	Both the front-end and back-end services must be in the same Kerberos realm.
resource-based	The constraint is configured on the back-end service. This enables the resource owner of the back-end service to configure what front-end services can delegate credentials to the back-end service.	The front-end services and back-end service do not have to be in the same Kerberos realm.

For information about configuring on Linux, see [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 12 and on Windows see [“Configure Kerberos Constrained Delegation in Active Directory”](#) on page 24.

OAuth and OIDC Authentication (Linux Full Deployment)

Overview of OAuth and OIDC

Open Authorization (OAuth) is a token-based authorization standard on the internet. OAuth 2.0 acts as an intermediary on behalf of the user, giving the third-party service an access token that authorizes specific account information. OIDC is an extension to OAuth 2.0, which provides authentication support.

Key Terms

Table 18 Term Definitions

Term	Definition
Access token	Specifies identifying information for a user, including the user's credentials, groups, and privileges.
OpenID Connect	An authentication layer built on top of OAuth 2.0.
Flow	The process for obtaining an OAuth token.

How It Works in SAS Viya

An OAuth 2.0 and OIDC provider can be internal to your organization, such as Microsoft Entra ID, or it can be an external provider, such as Google Authenticator or Facebook. When the OAuth option is configured, this does not completely replace the default LDAP provider. Instead, when users access SAS Logon Manager, they are presented with the standard login form and an additional link below the form. Clicking the link initiates the OIDC authentication process. If the user is already authenticated to the OIDC provider, clicking the link provides single sign-on to SAS Viya. The user identity and group membership information is verified in your IAM system.

OAuth can provide single sign-on from the OAuth provider. For example, when users sign in to their Google account, they can access the visual interfaces of SAS Viya without being prompted for additional credentials.

Fallback authentication with OAuth and OIDC is available when SAS Logon Manager is configured for Kerberos authentication. If Kerberos authentication fails, OAuth and OIDC authentication can be attempted. For more information, see [“Fallback Authentication” on page 72](#).

Simplified configuration of IdP discovery is also available for logins. The email address of the end user is used for redirection to the OIDC IdP. For more information, see [“Configure IdP Discovery for OIDC” on page 33](#) and [“IdP Discovery for OIDC and SAML” on page 81](#).

See Also

- [“Configure SAS Viya with Information about the OIDC IdP” on page 29](#)

- [“Configure Single Sign-On with Automatic Redirect” on page 40](#)

SAML Authentication (Linux Full Deployment)

Note: Security Assertion Markup Language (SAML) is currently not supported for SAS Visual Analytics Apps (formerly SAS Mobile BI) and SAS Add-in for Microsoft Office connections.

Overview of SAML

The SAML standard defines a framework for exchanging security information about users between an IdP and service provider. This security information is packaged in the form of portable XML assertions that applications working across security domain boundaries can trust. SAML allows for single sign-on to web browser applications.

Key Terms

Table 19 *Term Definitions*

Term	Definition
Federation	Allows multiple identity management systems to work together and establish trust.
Assertion	A package of information, in the form of an XML document, that is created and sent during a federated access request.
Claims	Information that a federation member is asserting to be true.
Identity provider	A federation member that authenticates users and keeps track of their information. Creates assertions for the users, and sends them to service providers.
Service provider	A federation member that consumes assertions to make access control decisions for its applications.
Metadata	An XML document that is produced by a SAML provider to describe its service endpoint URLs, x.509 certificate, and other information in a standard way for consumption by partners in the federation.
Relying party	A server providing access to secure software.

How It Works in SAS Viya

SAML supports configuring SAS Logon Manager to be integrated with an external SAML IdP. This IdP can be internal or external to the customer's environment. If it is internal, a tool similar to Oracle Access Manager can be used. If it is external, something like Salesforce can be used.

SAML does not completely replace the default LDAP provider. End-users who are accessing SAS Logon Manager can select either SAML authentication or the default LDAP provider. The user identity and group membership information is looked up in LDAP. This option also provides single sign-on with the third-party SAML provider.

When a user attempts to access a service URL, the service provider initiates the exchange with an authentication request. The service provider is SAS Logon Manager. The IdP sends a response that contains the assertion. The SAML protocol defines the structure and content of these request and response messages. When the user logs on to a service or system, the service provider trusts the IdP to validate the credentials, instead of providing credentials to the service provider. Therefore, users do not have to provide their credentials directly to any component but the IdP.

The following functionality is available:

- Fallback authentication when SAS Logon Manager is configured for Kerberos authentication. If Kerberos authentication fails, SAML authentication can be attempted. For more information, see [“Fallback Authentication” on page 72](#).
- Simplified configuration of IdP discovery for logins. In previous releases, custom code and JavaScript were needed to enable this feature. Now, the email address of the end user is used for redirection to the SAML IdP. For more information, see [“Configure IdP Discovery for SAML” on page 39](#) and [“IdP Discovery for OIDC and SAML” on page 81](#).
- IdP-initiated sign-on. This enables you to redirect into the SAS Viya environment and specify where you want the browser to go. The *RelayState* parameter is designed to be a state that the service provider can pass to the IdP with the authentication request and get back in the response. The parameter can also be specified by the IdP to indicate where you want the browser to go after redirecting to SAS Viya. Here is an example:
 - /SASVisualAnalytics/
 - /SASDrive/
 - /SASEnvironmentManager/

For more information, see [“IdP Initiated Logon for SAML” on page 81](#).

See Also

- [“Configure SAML \(Linux Full Deployment\)” on page 33](#)
- [“Configure Single Sign-On with Automatic Redirect” on page 40](#)

Single Sign-On with Automatic Redirect

When single sign-on (SSO) with automatic redirect is used with SAML or OIDC, the configuration relies on part of the OIDC specification. Since SAS Logon Manager is an OIDC provider to the rest of the SAS web applications, this functions for both SAML and OIDC third-party authentication.

Because the end user will not interact with SAS Logon Manager, they will not see the standard sign-in page. As a result, users will not be able to log in with LDAP credentials or the sasboot account. The documented procedure includes options to enable certain clients to retain access to the sign-in page. As an alternative, users can directly access the `/SASLogon/login` URL to access the sign-in page, but they will need to enter the URL for the application that they want to access after logging in.

The SSO configuration uses the `login_hint` option. The `login_hint` option is part of the OIDC specification and supported by SAS Logon Manager. This query string option is passed in the authorize request, which is `/SASLogon/oauth/authorize`. SAS Logon Manager expects to receive an email domain in the hint, and this value is compared against a list of email domains configured for each SAML or OIDC IdP. If a match is found, the user is redirected automatically to that provider, bypassing the sign-in page.

See Also

[“Configure Single Sign-On with Automatic Redirect” on page 40](#)

SAS 9.4 Authentication

Overview of SAS 9.4 Authentication

This option enables integration between SAS Viya and an existing SAS 9.4 environment. The authentication to the SAS Viya visual interfaces is performed by SAS Logon Manager in SAS 9.4. None of the authentication occurs with SAS Logon Manager in SAS Viya. Any authentication mechanism supported by SAS 9.4 is supported by this configuration. For more information about the supported authentication mechanisms, see *SAS Intelligence Platform: Security Administration Guide*.

Note: All versions of SAS 9.4 support this configuration. The SAS 9.4 deployment does not have to be running the latest maintenance release.

How It Works in SAS Viya

Here is a sample scenario:

- 1 The client's web browser connects to SAS Logon Manager in SAS Viya.
 - a If the request to SAS Logon Manager in SAS Viya does not have an existing session, the SAS Logon Manager in SAS Viya displays the sign-in page, which contains a link to perform SAS 9.4 authentication and the page to perform LDAP authentication.
 - b If the end user selects the link, SAS Logon Manager in SAS Viya constructs an authentication request and redirects the client's web browser to the SAS 9.4 middle tier.
- 2 The client authenticates to SAS 9.4, receives a service ticket, and is redirected to SAS Logon Manager on SAS Viya.
- 3 The client's web browser connects to SAS Logon Manager on SAS Viya, including the SAS 9.4 service ticket in the request.

- 4 SAS Logon Manager on SAS Viya connects to the SAS 9.4 middle tier to validate the service ticket and the end user.
- 5 SAS Logon Manager on SAS Viya connects to the identities service to get the group information for the validated end user.
- 6 The identities service either looks up the validated end user in its cache or connects to Active Directory using the LDAP service account to update the cache.

The SAS 9.4 authentication configuration affects only the SAS Viya visual interfaces that are using SAS Logon Manager. An LDAP provider is still required by the identities service. For authentication to SAS Logon Manager in SAS Viya that does not involve a web browser, the credentials are first passed to SAS Logon Manager in SAS 9.4. If they fail, the credentials are tried against LDAP. Therefore, authentication with the administration command-line interface (CLI) and SAS Visual Analytics Apps is still performed using SAS Logon Manager in SAS Viya first. SAS Studio 5.2 (Basic) is not affected by this configuration.

If you want to configure TLS for either the SAS 9.4 or SAS Viya deployment, the Apache HTTP server certificate must be trusted. Import the certificate that is used by one deployment into the SAS certificate framework of the other deployment. For more information, see [“Configure SAS 9.4 Clients to Work with SAS Viya” in *Encryption in SAS Viya: Data in Motion*](#).

Compatibility of User Names

The identities service must be able to take the authenticated user name from SAS 9.4 and search for it in the SAS Viya LDAP provider. You can sign on to SAS 9.4 using an internal account (which includes the @saspw suffix), but such accounts cannot exist in the LDAP provider. Therefore, these internal accounts do not work with SAS Viya.

In addition, you can sign in to SAS 9.4 with an account that does not exist in any LDAP provider, such as a Google account. This does not work with SAS Viya unless the Google account corresponds to the accountId property that is used by the identities service. For more information about the accountId property, see [“sas.identities.providers.ldap.group \(Field Mappings\)” in *SAS Viya Administration: Configuration Properties*](#).

Finally, domain-qualified user names cannot be used with SAS Viya. Even if the SAS 9.4 environment passed the domain qualified user name, the domain is stripped.

Single Sign-On and Single Sign-Out

Single sign-on and single sign-out are supported between SAS Viya and SAS 9.4. During single sign-on, a user with an active SAS 9.4 session can access SAS Viya applications without being required to sign on to SAS Viya.

Single sign-out is initiated from SAS Viya. If a user has two browser tabs open, one with a SAS Viya web application and the other with a SAS 9.4 web application, selecting the sign-out option in SAS Viya also signs the user out of SAS 9.4. However, the reverse is not true. If the user signs out from the SAS 9.4 web application, he or she is not signed out from the SAS Viya web application.

PAM Authentication (Linux)

Overview of PAM

PAM enables you to determine how applications use authentication to verify the identity of a user. It is an industry-standard technology that extends UNIX host authentication to recognize additional authentication providers. PAM uses *modules* or libraries to access multiple authentication methodologies. SAS Viya supports host authentication.

Account modules are required when SAS Logon Manager is configured for PAM. This ensures that the user is not authenticated with an expired password. Some authentication providers allow a user to use an expired password and address this in the account modules.

How It Works in SAS Viya

Default PAM configuration files, *SAS-Viya-configuration-directory/etc/pam.d/service*, are installed as a part of the SAS Viya deployment process.

Note: For the CAS server, *service* is *cas*. For SAS Studio, *service* is *sasauth*.

In order for *sasauth* to perform authentication, entries must be made in the PAM configuration files that are provided by SAS. These entries describe the authentication services that are used when *sasauth* performs an authentication. This includes the account and auth modules. The session and password modules are not supported.

TIP In a multi-machine deployment, configure PAM on the host with SAS Object Spawner and the host with CAS controller.

Fallback authentication is available when SAS Logon Manager is configured for Kerberos authentication. If Kerberos authentication fails, PAM authentication can be attempted. For more information, see [“Fallback Authentication” on page 72](#).

Authinfo File

Authentication is used to control access to the CAS server and its resources. Your identity must be successfully authenticated before your session is created. When password information is not available, an attempt is made to find an authinfo file (.authinfo is the default file name on Linux). The authinfo file provides a user name and password to CAS for host authentication. It is an alternative to including passwords in programs.

You can also force the use of the authinfo file by specifying `authinfo=` in the CAS statement. An alternative method is to use the `CAS_AUTH_METHOD` environment variable.

The authinfo file is required when you are using the command line to submit commands for the following tasks:

- Run programs in batch mode. The `USER=` option in the CAS statement or SAS system option `CASUSER=` can be specified.
- Perform limited server administration using the **casadmin** command.

- Run commands in line mode.
- Sign on to SAS/CONNECT and specify the casuser in the RSUBMIT block of code. This action is performed when the casuser is different from the SAS Viya user or when the user is the same for both SAS Viya and CASUSER, but the password is different.

Note: SAS Studio 5.2 (Basic) uses the user credentials that were used to launch the workspace server to authenticate your connection to CAS. SAS Studio 5.2 (Basic) does not use the authinfo file for authentication unless it is forced to.

Typically, the authinfo file is stored in the `$HOME` directory.

The authinfo file format is based on the .netrc file specification. The .netrc file format is an older format. You can see the file specification at [Netrc Format](#). In addition to the standard .netrc file standards, the authinfo specification allows for putting commands in the file as well as using quoted strings for passwords. The quoted strings allow for spaces within passwords.

If the authinfo file contains values that match the host, port, or user name. The information contained in the authinfo file is used to connect to CAS.

The following system options and environment variables can be used to override the authinfo file. These options point to authinfo files that are located in a different directory or are named differently.

Here are the ways that the AUTHINFO system option, the environment variable, and the statement option can be used to override the authinfo file:

- Environment variable AUTHINFO takes precedence over the authinfo file.
- SAS system option AUTHINFO= (alias CASAUTHINFO=) overrides the AUTHINFO environment variable as well as the authinfo file.
- AUTHINFO= option in the CAS statement overrides the AUTHINFO= system option, the AUTHINFO environment variable, and the authinfo file.

For more information, see the following documents:

- [AUTHINFO= System Option](#)
- [CAS Statement](#)
- [CAS_AUTH_METHOD environment variable on page 111](#)
- [USER=user-ID argument](#)
- [Batch Mode in UNIX Environments](#)

Multi-Factor Authentication

Multi-Factor authentication (MFA) is a security system that requires more than one method of authentication from independent categories of credentials to verify the user's identity for a sign-on or other transaction.

MFA combines two or more of the following independent credentials:

- what the user knows – their password
- what the user has – a security token
- who the user is – biometric verification

The goal of MFA is to create a layered security defense, making it more difficult for an unauthorized person to access a target such as a physical location, computing device, network, or database.

Typical MFA scenarios include the following:

- swiping a card and entering a PIN
- logging on to a website and being requested to enter an additional one-time password (OTP) that the website's authentication server sends to the requester's phone or email address
- downloading a virtual private network (VPN) client with a valid digital certificate and logging on to the VPN before being granted access to a network
- swiping a card, scanning a fingerprint, and answering a security question
- attaching a universal serial bus (USB) hardware token to a desktop that generates a one-time passcode and using the one-time passcode to sign on to a VPN client

See Also

[“Configure Operating-System Authentication with PAM \(Linux\)” on page 42.](#)

Additional Authentication Topics

IdP Discovery for OIDC and SAML

By default, when one or more external IdPs are configured, they are listed on the sign-in page under **Or sign in with**. Each IdP is listed as a hyperlink that can be configured to display your choice of text. You can choose not to display the IdPs and instead configure IdP discovery. When configured, IdP discovery uses the domain of the user's email address to automatically select which IdP to use.

When IdP discovery is configured, the sign-in screen displays a **User ID or Email** field. If a user enters an email address, the domain from the email address is compared to the list of email domains that are configured for each IdP. If a match is found, the user is automatically redirected to that IdP, in the same way they would be if they clicked on the standard sign-in page. If no match is found, the standard **Sign In** page is displayed with the **User ID** and **Password** fields.

IdP Initiated Logon for SAML

You can update the SAML IdP to redirect to the SAS Viya website that you want the user to go to, using the *RelayState* parameter.

IdP discovery streamlines logging in when multiple IdPs are configured. Instead of a sign-in page with the standard user name and password fields followed by a list of SAML providers, users are prompted to enter their user name or email address and click **Next**. If the `emailDomain` option is configured for an external IdP, the user is automatically redirected to the provider. Otherwise, the user is prompted for a password. The `emailDomain` option is configured for SAML in [Step 1c on page 36](#).

In the IdP-initiated flow, use the *RelayState* parameter to specify the relative URL of a SAS web application to redirect to post-authentication. You must also ensure that the correct links are available for redirection to the SAS Viya environment.

Note: The relative URL requires a trailing slash (for example, /SASDrive/).

SAS/CONNECT Authentication

As an administrator, you might want to enable SAS Viya to accept connections for existing SAS 9 environments. SAS/CONNECT enables that connection, and passes credentials that can be used in the SAS Viya environment.

With SAS Viya, your credentials are used to authenticate to CAS when you are using SAS/CONNECT. When additional SAS/CONNECT servers are spawned, SAS/CONNECT forwards your credentials to the spawned SAS/CONNECT server session.

Here are the ways that SAS/CONNECT and CAS authenticate your user credentials:

- The spawner passes the SIGNON credentials to the SAS/CONNECT server where the credentials can be used to connect to CAS in the following situations:
 - when the user is using any environment that is not a SAS Viya environment
 - when the user is connecting to SAS Viya via the SAS/CONNECT spawner
- When the user is in the SAS Viya environment using SAS Studio and starting SAS/CONNECT server sessions (using SASCMD SIGNON or the CONNECT Spawner), the CAS credentials (if they exist) are passed to the SAS/CONNECT server in SAS Viya.
- When running SAS Viya in batch or line mode, the authinfo file is used to authenticate to CAS. If you specified the USER= option in the CAS statement, CASUSER= system option, or if you specified the CAS_AUTH_METHOD environment variable, authinfo file authentication is used.

For more information, see the following documents:

- [USER=user-ID](#)
- [CAS AUTH_METHOD environment variable on page 111](#)
- [“Operate \(Linux\)” in SAS Viya Administration: Programming Run-Time Servers](#)

Single Sign-On (Full Deployment)

Single sign-on (SSO) is an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. For example, SSO can enable a user to access SAS servers that run on different platforms without interactively providing the user's ID and password for each platform. SSO can also enable someone who is using one application to launch other applications based on the authentication that was performed when the user initially logged on.

SAS Logon Manager is the central point for handling changes to authentication mechanisms, such as the addition of third-party SSO products. SAS Viya supports Kerberos, SAML, and OIDC. For information about configuring these products, see:

- Kerberos (on [Linux](#) or on [Windows](#))
- SAML
- OIDC

Dual Authentication

Linux

In a dual authentication environment on Linux, users are validated against the LDAP server or equivalent IAM system and the host authentication mechanism. The following conditions exist:

- If PAM is configured to use local accounts and those users also sign on to the visual interfaces, those local accounts must match the accounts in the IdP that is used for SAS Logon Manager.
- If PAM is configured to use an LDAP server, SAS Logon Manager should be configured to use the same LDAP server.
- When directly connecting to the CAS server using SAS Studio or a batch job, the user ID and password that are supplied are authenticated against both the LDAP server and PAM.

Windows

In a dual authentication environment on Windows, users are validated against the LDAP server and the host authentication mechanism. The following conditions exist:

- The LDAP server should be configured to use the same Active Directory server that the Windows host is using.
- When directly connecting to the CAS server using SAS Studio or a batch job, the user ID and password that are supplied are authenticated against the LDAP server and host authenticated.

Authentication: Guest Access (Linux)

About Guest Access



Note: Guest access is currently not supported in a Windows deployment.

Guest access is an optional feature that provides anonymous Read-Only access to a subset of resources and functionality in participating applications. Guest access is supported for viewing reports in SAS Visual Analytics and SAS Visual Analytics App (previously called SAS Mobile BI).

For information about multi-tenancy, see [“Enable Guest Access” in SAS Viya Administration: Multi-tenancy](#).

Enable Guest Access

Note: In a multi-tenancy environment, the following steps must be repeated for each tenant that supports guest access.

- 1 Set the `sas.logon.provider.guest` configuration property, using SAS Environment Manager:
 - a In the applications menu () , select **Administration** ⇌ **Manage Environment**. In the navigation bar, select .
 - b Create a new configuration instance for **sas.logon.provider.guest**, ensuring that you enable the guest access option. For more information, see [“Create Configuration Instances” in SAS Viya Administration: Configuration Properties](#).
- 2 Add rules that provide the necessary access to functionality:
 - a From the SAS Viya machine where the command line interfaces are installed, create a default profile, if you have not already created one, and sign on. For more information, see [“Command-Line Interface: Preliminary Instructions” in SAS Viya Administration: Using the Command-Line Interfaces](#).
 - b Modify the authorization rules.

- For a new SAS Viya 3.5 installation, run the following command:

```
sas-admin authorization facilitate-guest
```

- For an upgrade from SAS Viya 3.4 to SAS Viya 3.5 in which guest access was not previously configured, run the following command:

```
sas-admin authorization facilitate-guest
```

- For an upgrade from SAS Viya 3.4 to SAS Viya 3.5 in which guest access was previously configured, complete the following steps:

- Run the `facilitate-guest` command.

```
sas-admin authorization facilitate-guest
```

Output similar to the following is displayed:

```
The jsonPatch was not valid.
```

```
Http Status: 400
```

```
ErrorCode: 1177
```

```
Detailed Messages:
```

```
correlator: e607fd5d-c4c8-4548-ad2d-b9e608ccf41a
```

```
traceId: 49f41d99e62595f2
```

```
path: /authorization/rules
```

```
FieldError: Rule [id=<defined_id>, type=GRANT, permissions=[READ], principal=null, principalType=guest, containerUri=null, objectUri=/identities/users/@currentUser, mediaType=null, condition=null, filter=null, reason=null, description=Guest Access: XXX, isEnabled=true, matchParams=false, isShare=false]:Provided authorization rule is a duplicate of this rule.
```

- Remove the rule ID that is specified in the output of the previous step:

```
sas-admin authorization remove-rule --id=<defined_id>
```

- Run the *facilitate-guest* command again. If an error message is displayed stating “Provided authorization rule is a duplicate of this rule”, repeat the previous step to remove the rule ID.

Repeat this step until the *facilitate-guest* command runs successfully.

- For an upgrade from a release prior to SAS Viya 3.4 to SAS Viya 3.5, run the following command:

```
sas-admin authorization facilitate-guest
```

- 3 If you have a preexisting source file, modify the direct access controls for the predefined caslibs on the server. Use the controls that are defined in the specified source file. The following command must be executed by a user who is a member of the Superuser role.

```
sas-admin cas facilitate-guest --source-file path-to-controls-file --server CAS-server-name --superuser
```

- 4 If you do not have a preexisting source file, you can generate one that contains the default access controls, make modifications to it, and use it as the source file. The following command must be executed by a user who is a member of the Superuser role.

```
1 sas-admin cas generate-guest-controls --output-location /path/
2 sas-admin cas facilitate-guest --source-file path-to-controls-file --server serverA --superuser
```

- 1 Generate a source file from the default access controls. The generated source file is named **facilitate-guest-controls.txt**.

Make the desired modifications to access controls to the source file that you just generated.

- 2 Modify the direct access controls using the source file that you just modified.

- 5 Add access controls that provide Read access to caslibs that should be accessible to guest users:

- a From the SAS Viya machine, if you have not already signed in to SAS Viya, sign on using the default profile that was created in the previous step.

- b Run the following commands as a user who is a member of the Superuser role:

```
sas-admin cas caslibs add-control --server server-name --caslib caslib-name --grant readInfo --guest --superuser
```


```
sas-admin cas caslibs add-control --server server-name --caslib caslib-name --grant select --guest --superuser
```

```
sas-admin cas caslibs add-control --server server-name --caslib caslib-name --grant limitedPromote --guest --superuser
```

- 6 Use SAS Environment Manager to grant Read access to folders and reports that should be accessible to guest users:

- a From the **Content** page, identify the folder to which you want to grant Read access to guest users.

- b Right-click and select **Edit authorization**.

- c Click  and select **Add Guest**. Grant Read and Read (convey) access. For more information, see “[General Authorization: How To \(Authorization Window\)](#)” in *SAS Viya Administration: General Authorization*.

- d Click **Save**.

Note: From the **Content** ⇒ **Users** ⇒ **guest** page, you can move folders and objects into the **My Folder** folder for the guest user. You can also create and add folder and report shortcuts into the **My Favorites** and **My Folder** folders. For more information, see “[Folders: How To](#)” in *SAS Viya Administration: Folders*.

Connect as Guest Users

Once guest access is enabled, guest users can view reports using SAS Visual Analytics and SAS Visual Analytics App. SAS Visual Analytics displays a guest sign-in button. SAS Visual Analytics App displays a guest sign-in button when a mobile connection is established.


See Also

- [SAS Report Viewer 8.3 Documentation](#)
- [SAS Visual Analytics: Viewing Reports](#)
- [SAS Visual Analytics App Documentation](#)

Generate Custom Links to Reports

You can create a custom web link for guest users, allowing them to access a specific report. If guest access is enabled, the custom link is configured to bypass the sign-in page and automatically connect the user as guest. If guest access is disabled, a sign-in page is displayed, where users can choose to connect as a guest or sign-in with their credentials.


Generate Custom Links to Reports Using SAS Report Viewer

- 1 From SAS Report Viewer, open the report to which you want to generate a link.
- 2 Click  and then select **Share report** ⇒ **Link**.
- 3 In the Generate Link window, customize the link, if necessary, in the **Link** field.
- 4 Click **Copy Link**. You can paste the link and distribute to guest users.

See Also

[SAS Report Viewer 8.3 Documentation](#)




Generate Custom Links to Reports Using SAS Visual Analytics

- 1 From SAS Visual Analytics, open the report to which you want to generate a link.
- 2 Click  and then select **Copy Link**.
- 3 In the Copy Link window, customize the link, if necessary, using the Options selections.
- 4 Click **Copy Link**. You can paste the link and distribute to guest users.

See Also

[SAS Visual Analytics: Viewing Reports](#)

Disable Guest Access

- 1 Set the `sas.logon.provider.guest` configuration property, using SAS Environment Manager:
 - a In the applications menu () , select **Administration** ⇨ **Manage Environment**. In the navigation bar, select .
 - b From the **Definitions** view, select **sas.logon.provider.guest**.
 - c Click . In the Edit `sas.logon.provider.guest` Configuration window, select the option to disable guest access.

.....
Note: The `sas.logon.provider.guest` option is tenant-specific and must be disabled for each tenant.

- d Click **Save**.
- 2 (Optional) Remove the rules that provide the necessary access to functionality:
 - a From the SAS Viya machine, navigate to the `SAS-Viya-installation-directory/home/bin` directory.
 - b At the command prompt, create a default profile and sign on by entering the following commands:

```
sas-admin profile init
sas-admin auth login
```

- c Modify the authorization rules by running the following command:

```
sas-admin authorization disable-guest-access
```

.....
Note: This command removes the rules that were automatically loaded by the `facilitate-guest` command. If you manually created any custom rules, using either SAS Environment Manager

or the command-line interface, you must manually remove those rules. A list of the remaining guest rules can be viewed on the SAS Environment Manager **Rules** page.

- 3 (Optional) Run the following commands as a user who is a member of the Superuser role to remove CAS Access grants:

```
sas-admin cas sessions create --server server-name --name clisession --superuser
```

```
sas-admin cas caslibs remove-control --server server-name --caslib VAModels
--grant readInfo --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib VAModels
--grant select --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib VAModels
--grant limitedPromote --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib ReferenceData
--grant readInfo --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib ReferenceData
--grant select --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib ReferenceData
--grant limitedPromote --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib AppData
--grant readInfo --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib AppData
--grant select --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib AppData
--grant limitedPromote --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib Formats
--grant readInfo --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib Formats
--grant select --guest --session-id session-id
```

```
sas-admin cas caslibs remove-control --server server-name --caslib Formats
--grant limitedPromote --guest --session-id session-id
```

```
sas-admin cas sessions delete --server server-name --session-id session-id
```

Note: These commands remove the grants that were automatically defined by the `facilitate-guest` command. If you manually created any custom grants, using either SAS Environment Manager or the command-line interface, you must manually remove those grants.

You can also remove guest access with the `cas remove-guest-controls` command. However, this command removes only the default set of direct access controls and not any other guest access controls that you might have applied. To view which direct access controls are removed, see the

`facilitate-guest-controls.txt` file that is generated by the `cas generate-guest-controls` command.

Scenario: OIDC with Microsoft Entra ID (Linux Full Deployment)

The following sections describe how to configure Microsoft Entra ID (formerly Microsoft Azure Active Directory or Azure AD) as an OIDC provider. In this scenario, SAS Viya uses Microsoft Entra ID as the single sign-on entry point for initial user authentication with OIDC as the protocol. In addition, you must configure LDAP or SCIM, which SAS Viya requires for loading user identities from Microsoft Entra ID in Azure. LDAP is enabled by default. For more information, see [“How to Configure SCIM”](#) in *SAS Viya Administration: Identity Management*.

Configure Microsoft Entra ID for OIDC

- 1 Log on to the Azure portal.
- 2 From the Home page, click **Entra ID**.
- 3 In the left pane, click **App registrations** ⇒ **New Registration**.
- 4 Create the new registration.
 - a Provide a name for the SAS Viya deployment.
 - b At the bottom of the screen, specify the redirect URI for SASLogon (for example: `https://hostname.example.com/SASLogon/login/callback/azure`).

.....

Note: The name, *azure*, is configured in [Step 4 on page 90](#) using SAS Environment Manager. It is specified as the name field for the `sas.logon.oauth.providers` definition.

.....

- c Leave the default values for the other options and click **Register**.

Azure displays an overview of the application registration. You can reach this page again by clicking **Overview** in the left pane.

.....

Note: Save the Application (client) ID. This value is specified as the *relyingPartyId* in SAS Environment Manager.

.....

- 5 Click **Certificates and secrets** ⇒ **New client secret**.
- 6 Under **Expires**, select an expiration option. Microsoft recommends short-lived secrets. Click **Add**.

.....

Note: Save the value that is displayed. You will not be able to retrieve this value once you leave this page. If you lose it, you must create a new secret.

When the client secret expires, create a new one in the Azure Portal. Be sure to update the the new client secret value in the **relyingPartySecret** field for the `sas.logon.oauth.providers` configuration. This field is described below, in [“Configure OIDC Provider Properties for Microsoft Entra ID” on page 90](#).

- 7 Add the optional claim.
 - a Click **Token Configuration** ⇒ **Add Optional Claim**.
 - b Select **Token type ID** and then select from the list of claims that should appear in the ID token sent to SAS Viya. The claim that you want to use as a user name in SAS Viya must be included. In most cases, you should use `email` or `upn`. You can add more claims than you need and later select which ones to use.
 - c Click **Add**.

A list of all of the claims to be included in the ID token is displayed.
- 8 Set the API permissions.
 - a Click **API Permissions**.
 - b The permissions should be added automatically. If no permissions are listed, click **Add a permission** and follow the prompts to add permissions. The permission that you select determines what claims are sent to SAS Viya and can be used by SAS Viya for the `username`. The Microsoft Graph **User.Read** or **email** permissions provide the option to choose `email`. The Microsoft Graph **profile** permission provides the option to choose `UPN`.
 - c The administrator must grant the permission consent.
- 9 Several endpoints from Entra ID are needed to configure SAS Viya. From the App registrations Overview page, click **Endpoints**. Leave this window open so that you can access the endpoints information during the following steps.

Configure OIDC Provider Properties for Microsoft Entra ID

- 1 Log on to SAS Environment Manager and navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Configuration Instances” in SAS Viya Administration: Configuration Properties](#).
- 2 In the **Definitions** list, select **sas.logon.oauth.providers**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New `sas.logon.oauth.providers` Configuration window, enter the values from the following table:

Table 20 Configuration Fields and Descriptions

Configuration Field	Description
attributeMappings.user_name	Defines the attribute, returned by the OIDC provider, that should be mapped to the <i>username</i> attribute in SAS Viya. The attribute must match what is configured in LDAP or the SCIM provisioning for the <i>userName</i> mapping. The attribute must be available from the ID token claims in Step 7 on page 90 (for example, email or UPN). This field is required.
authUrl	From the Entra ID Endpoints page, use the OAuth 2.0 authorization endpoint (v2) value. This field is required.
discoveryUrl	The URL used to discover the provider and obtain information needed to interact with it. From the Entra ID Endpoints, use the OpenID Connect metadata document URL. This field is optional. If a value is not specified, you must enter values for the issuer and tokenKeyUrl fields.
emailDomain	A comma-delimited list of email domains that are valid with this OIDC provider (for example, <i>mycompany.com</i>). This field is optional.
issuer	From the Entra ID Endpoints page, find the URL for the OIDC metadata document. Copy this URL. Open a browser tab and navigate to this URL. In the JSON, locate the issuer field, and use the value that is supplied. This field is optional if you specify a value for the discoveryUrl field.
linkText	Enter label text for the login link that will be displayed on the sign-in page. By default, the value is Use your corporate credentials . This field is optional.
name	Corresponds to the <code>redirect_uri</code> that is provided in the App registration that is created in the Azure portal. This field is required. IMPORTANT Do not include a period or other special characters in the value specified for the name.
relyingPartyId	Enter the Application (client) ID from the App registrations Overview page that you saved during Step 4c on page 89 . This field is required.
relyingPartySecret	A shared secret between the SAS Viya environment and the OIDC provider. Enter the client secret that you

Configuration Field	Description
	created during Step 5 on page 89 . This field is optional.
responseType	Enter the expected response type, <code>code</code> . This field is required.
scope	The comma-delimited field of scopes for the authorization request. The list should contain <code>openid</code> . If you are using a UPN for the username claim, include <code>profile</code> . This field is required. Note: The list of scopes must not contain white space.
tokenKeyUrl	The public key used to sign tokens either directly entered into SAS Environment Manager or provided at the given URL. From the browser where you loaded the OIDC metadata document, locate the <code>jwtks_uri</code> field. This field is optional if you specify a value for <code>discoveryUrl</code> . Note: While logging on, if you receive an error message indicating that the signing key cannot be found or that the token signature could not be verified in order to validate the ID token, add <code>?appid=Application-client-ID</code> to the URL (for example, <code>https://login.microsoftonline.com/2b34b84d-489f-4f71-ba55-0225369e1c9b/discovery/v2.0/keys?appid=Application-client-ID</code>). If a value for the <code>discoveryUrl</code> field is not specified, the <code>tokenKeyUrl</code> option must be specified.
tokenUrl	The endpoint on the OIDC provider where tokens are issued. From the Entra ID Endpoints page, use the <code>OAuth 2.0 token endpoint (v2)</code> value. This field is required.

- 5 Leave the default values for the remaining fields and click **Save**.

A link to log on with Azure should now be displayed on the SAS Logon Manager page.

If you see an error that refers to a violation of a "Content Security Policy directive," your program requires a change to the content security policy setting for SAS Logon Manager. Follow the steps that are described in ["Update the Content Security Policy" on page 48](#).

Scenario: SAML with Microsoft Entra ID (Linux Full Deployment)

The following tasks provide details about how to configure Microsoft Entra ID (formerly Microsoft Azure Active Directory or Azure AD) as a Security Assertion Markup Language (SAML) provider. In this scenario, SAS Viya uses Microsoft Entra ID as the single sign-on entry point for initial user authentication with SAML as the protocol.

Note: You must configure Transport Layer Security (TLS) with a trusted CA in order to complete this configuration.

Configure Microsoft Entra ID for SAML

Note: To perform this task, you must have administrator privileges in the Azure portal.

Prior to completing this task, you must configure an Entra ID (Active Directory) in Azure.

- 1 Log on to the Azure portal.
- 2 From the Home page, click **Entra ID**.
- 3 Add a non-gallery application to **Enterprise Applications** in the Azure portal for SAS Viya. For details about how to administer Microsoft Entra ID, see [Microsoft Entra ID documentation](#).

Note: The name of the application should represent the SAS Viya deployment.

Configure SAML Provider Properties for Microsoft Entra ID

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 Configure the **sas.logon.saml** definition.
 - a In the **Definitions** list, select **sas.logon.saml**.

Note: If you change any of the `sas.logon.saml` properties, the new metadata must be provided to the Relying Party in the federated service. If it is not, the SAML connections might fail.

- b In the top right corner of the window, click **New Configuration**.
- c In the New `sas.logon.saml` Configuration window, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:

Table 21 *SAML Configuration Fields and Descriptions*

Configuration Field	Description
<code>entityBaseURL</code>	The external URL for the SAS Logon web application in SAS Viya (for example, <code>https://hostname.example.com/SASLogon</code>). This field is required.
<code>entityID</code>	The unique ID that represents the service provider that is included in protocol messages between relying parties. Change from the default value that is pre-populated. This field is required.
<code>maxAuthenticationAge</code>	Specifies the maximum time (in seconds) between users' initial authentication with the IdP and processing of an authentication statement. The default value is 864000. This field is optional.
<code>serviceProviderCertificate</code>	Paste a copy of the PEM-encoded (base64) certificate, which is used by the service provider. This field is required.
<code>serviceProviderKey</code>	Paste a copy of the PEM-encoded (base64) key, which is used by the service provider. This field is required.
<code>serviceProviderKeyPassword</code>	Provide the password for the service provider, or leave blank if there is no password. This field is optional.
<code>setProxyParams</code>	IMPORTANT This field should not be modified. The value should remain Off .
<code>signatureAlgorithm</code>	Specifies the algorithm for SAML signatures. Acceptable values are SHA1, SHA256, and SHA512. The default value is SHA256. This field is optional.

Configuration Field	Description
signMetaData	Specifies whether the local service provider should sign the metadata. This field is required.
signRequest	Specifies whether the local service provider should sign the SAML requests. This field is required.
socket.connectionManagerTimeout	Specifies the amount of time (in milliseconds) before the connection pooling times out for HTTP requests for SAML metadata. The default value is 10000. This field is optional.
socket.soTimeout	Specifies the amount of time (in milliseconds) before the read times out for HTTP requests for SAML metadata. The default value is 10000. This field is optional.
wantAssertionSigned	Specifies whether the assertions should be signed. This field is required.

- d Click **Save**.
- e Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

- 3 Configure the **sas.logon.saml.providers** definition.
 - a In the **Definitions** list, select **sas.logon.saml.providers**.
 - b In the top right corner of the window, click **New Configuration**.
 - c In the New sas.logon.saml.providers Configuration window, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:

Table 22 SAML External Provider Configuration Fields and Descriptions

Configuration Field	Description
addShadowUserOnLogin	<p>Add a local shadow user upon successful authentication. The default value is set to On. This field is optional.</p> <p>Note: This option should always be set to On.</p>
assertionConsumerIndex	<p>The index of the assertion consumer service to use from identity provider metadata. The default value is 0. This field is optional.</p>
authnContext	<p>The comma-separated list of authentication contexts that are included in SAML requests to the IdP. This field is optional.</p>
emailDomain	<p>Specifies a comma-separated list of email domains for users that can sign on with the SAML provider. It is used with IdP discovery. This field is optional.</p>
idpMetadata	<p>The IdP metadata, or the URL to the metadata. The former can be useful if manual changes need to be made to the identity provider metadata. In the Azure portal, this value is specified as App Federation Metadata Url. This field is required.</p> <p>To find the Federation Metadata for Azure, in the Azure portal, navigate to the Enterprise Application that you created in Step 3 on page 93. Click Single Sign-On on the left pane. Click Edit for the SAML Signing Certificate. If a certificate does not exist, create and activate it. The URL is shown under App Federation Metadata Url. You can also download the XML from the Federation Metadata XML link and paste the XML content in this field.</p>
linkText	<p>Enter label text for the login link that will be displayed on the login page (for example, Entra ID using SAML). This field is optional.</p>
metadataTrustCheck	<p>Specify whether to check the Azure identity provider certificate. Set to On. This field is required.</p>
name	<p>Specifies a unique name for this provider (for example, azure).</p>

Configuration Field	Description
	IMPORTANT Do not include periods, spaces, or other special characters in the value specified for the name.
nameID	The default value is unspecified in Azure and can usually be left as the default. This field is required.
showSamlLoginLink	Determines whether a link should be displayed on the sign-in page for this identity provider. Set to On . This field is required.
skipSslValidation	Specifies whether to skip the TLS validation of SAML protocol messages. Set to Off . This field is optional.

d Click **Save**.

- Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

- If you see an error that refers to a violation of a "Content Security Policy directive," your program requires a change to the content security policy setting for SAS Logon Manager. Follow the steps that are described in "[Update the Content Security Policy for SAML](#)" on page 40.

Configure the Enterprise Application in Microsoft Entra ID

- Download the SAS metadata XML file from <https://hostname.example.com/SASLogon/saml/metadata>.
- Open the metadata file.
You will use the information from the file to populate the fields in the **Basic SAML Configuration** in the Azure portal.
- In the Azure portal, navigate to the Enterprise Application that you created during [Step 3 on page 93](#).

- 4 Under the Enterprise Application, click **Set up Single Sign-On with SAML**.
- 5 In the **Basic SAML Configuration** section, enter values for the fields. The following table provides guidance on what information needs to be provided:

Table 23 Basic SAML Configuration Fields and Descriptions

Field	Description
Identifier (Entity ID)	Specify the entityID value from Table 21 on page 94 .
Reply URL (Assertion Consumer Service URL)	Specify the <i>Assertion Consumer Service URL</i> from the SAS metadata XML file downloaded during Step 1 on page 97 .
Sign on URL	Leave the field blank. This setting instructs Azure to always use the Assertion Consumer Service URL.
Relay State	Specify a relative URI to any SAS application (for example, <i>/SASDrive/</i>).
Logout Url	This field is optional. If you want to have single logoff, find the <i>Logout URL</i> from the SAS metadata XML file that you downloaded during Step 1 on page 97 .

- 6 In the **User Attributes and Claims** section, for the **Unique User Identifier**, specify the attribute that you want to use as the unique *username* in SAS Viya.

Configure Cross-Origin Settings for SAML

The SAML protocol uses the `POST` binding to send a SAML response from the identity provider (IdP) back to the service provider (SP). The web browser treats this as a cross-origin request because it is initiated from the IdP. Security settings on the server must specifically instruct the web browser to send cookies to the SP. If the SP receives the SAML response without the session cookie, it cannot link it to the original SAML request and fails with an error.

To fix the error, complete the following steps:

- 1 In SAS Environment Manager, edit the CORS configuration instance. For more information, see [“Edit Configuration Instances” in SAS Viya Administration: Configuration Properties](#).
- 2 Select **sas.common.web.security.cors**.
- 3 Create or edit the **sas.common.web.security.cors** configuration.

a Select **SAS Logon Manager** from the list of services.

b Set **allowedOrigins** to the specific Origin header value from the SAML IdP.

You can determine this value by turning on developer tools in the browser and looking at the HTTP request headers on the first request back to SAS Viya after authenticating with the IdP. For Azure, this is usually `https://login.microsoftonline.com`.

Note: Do not use wildcards.

c Click **Save**.

4 Create or edit the **sas.common.web.security.cookies** configuration.

a Select **SAS Logon Manager** from the list of services.

b Set **sameSite** to **None**.

c Click **Save**.

d Restart the SAS Logon Manager Service.

For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

On Windows, in Windows Services Manager, right-click the **SAS Logon Manager Service** and select **Restart**.

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

Authentication: OIDC with ISAM Scenario (Linux Full Deployment)

In the following tasks, OIDC uses IBM Security Access Manager (ISAM) WebSEAL reverse proxy server as the single sign-on entry point for initial user authentication. Other providers can be used, but configuration instructions are not provided here.

To configure the OAuth and OIDC, complete the following sections:

Configure OIDC Provider Properties for ISAM

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- 2 In the **Definitions** list, select **sas.logon.oauth.providers**.

- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New sas.logon.oauth.providers Configuration window, enter values for the required fields, based on your environment. [Table 24 on page 100](#) provides guidance about the information needed for the listed fields.

Table 24 *OIDC Configuration Fields and Descriptions*

Configuration Field	Description
addShadowUserOnLogin	A local shadow user should be added once authentication is successful. This field is required. Note: This option should always be set to On .
attributeMapping.user_name	The attribute claim to use as the user name. For ISAM, use sub . This field is required.
authUrl	The URL to the authorization endpoint (for example, <code>https://hostname.example.com/isam/oidc/endpoint/amapp-runtime-ISAMOP/authorize</code>). This field is required.
discoveryUrl	Specifies the URL that is used to discover the provider and obtain information that is needed to interact with it. This field is optional. If a value is not specified, you must enter values for the issuer field and either the tokenKeyUrl or tokenKey field.
emailDomain	Specifies a comma separated list of email domains for users that can sign in with the OIDC provider. It is used with identity provider discovery. This field is optional.
issuer	The principal that issued the token, specified as a case-sensitive string or URI. This is your WebSEAL instance (for example, the reverse proxy entry point, <code>https://oidcidp.example.com</code>). This field is optional if you specify a value for the discoveryUrl field.
linkText	The text that should be displayed on the sign-in page for the provider (for example, OpenID Connect Login Using ISAM Reverse Proxy [WebSEAL]). This field is optional.
name	Corresponds to the <code>redirect_uri</code> that is provided in the App registration that is created in the ISAM portal. This field is required. IMPORTANT Do not include a period or other special characters in the value specified for the name.
relyingPartyId	The client ID that is registered with the provider. This field is required.

Configuration Field	Description
relyingPartySecret	The secret that is registered with the provider for the client ID. This field is optional.
scopes	The comma-delimited list of scopes for the authorization request. The list should contain <code>openid</code> . This field is required. Note: SAS Viya does not process any additional scopes that are returned in the token.
showLinkText	The link text should show on the sign-in page. This field is required.
tokenKey	Specifies the HMAC key or RSA public key that is used to sign ID tokens. This field is optional if you specify a value for <code>discoveryUrl</code> . Note: If a value for the <code>discoveryUrl</code> field is not specified, either the <code>tokenKey</code> or <code>tokenKeyUrl</code> field must be specified.
tokenKeyUrl	Specifies the URL to obtain the signing key. This field is optional if you specify a value for <code>discoveryUrl</code> . Note: If a value for the <code>discoveryUrl</code> field is not specified, either the <code>tokenKey</code> or <code>tokenKeyUrl</code> field must be specified.
tokenUrl	The URL to the tokens from the provider. This field is required.
type	The protocol type. By default, the value is <code>oidc1.0</code> . This field is required. Note: SAS Viya requires an <code>id_token</code> in the authorization response from the provider. However, some providers return an <code>id_token</code> when the scope in the authorization request is <code>openid</code> and <code>respose_type=token</code> . For those providers, use type <code>oauth2.0</code> .

5 Click **Save**.

6 Restart the SAS Logon Manager Service.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

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

Configure OIDC Provider in ISAM

For basic steps to configure OIDC in ISAM 9.0.3.1, search for the Access Manager Federation Cookbook 9.0.0.0- 9.0.3.0 at [IBM Security Community](#).

To configure OIDC provider, complete the following steps:

- 1 In the ISAM 9.0.3.x admin console, create the WebSEAL reverse proxy instance as a single sign-on entry point.
- 2 Configure an OIDC provider and its partner.

An OIDC provider on ISAM is a federation. First create a federation that represents the OIDC provider. Then, create a partner that represents the SAS Viya application under it.

- 3 Create a federation for OIDC provider. [Table 25 on page 102](#) displays the values that you should provide while creating the new federation.

Table 25 Create New Federation Values

Field Name	Value
Federation Name	ISAMOP
Protocol for this federation	OpenID Connect
Role	OpenID Connect provider
Issuer Identifier	www.oidcidp.example.com Note: This is your WebSEAL instance.
Signature Algorithm	HS256
Grants	Authorization Code
Identity Mapping	Do not perform identity mapping . The same user name exists both in ISAM LDAP and SAS Viya LDAP.

- 4 Create an OIDC provider partner for SAS Viya (SASLogon). [Table 26 on page 103](#) displays the values that you should provide while creating the new partner.

Table 26 Create New Partner Values

Field Name	Value
Name	ISAM-to-SASViya
Enabled	Yes
Connection Template	OIDC
Client ID	isamClientID
Client Secret	isamClientSecret
Client Display Name	SAS Viya Client
Response Types	code, id-token token, and token
Allow Refresh Token Grant	Enabled
Redirect URIs	https://sas-viya-host/SASLogon/login/callback/external_oauth
Scope	openid

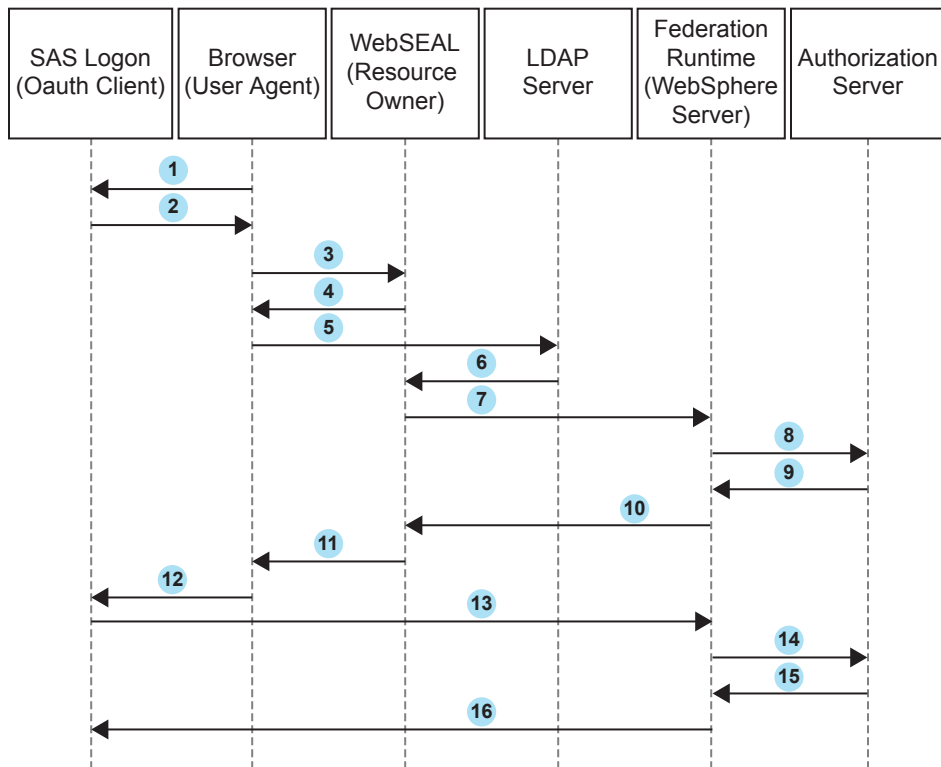
- 5 Test your configuration by accessing SAS Environment Manager. The text that you specified in the *linkText* field in [Step 4 on page 100](#) should be displayed

If you see an error that refers to a violation of a "Content Security Policy directive," your program requires a change to the content security policy setting for SAS Logon Manager. Follow the steps that are described in ["Update the Content Security Policy" on page 48](#).

OIDC and ISAM

The following diagram depicts the IBM Security Access Manager reverse proxy components and process flow.

Figure 1 IBM Security Access Manager Components and Flow



In this figure, the numbered arrows correspond to the following activities:

- 1 A client browser (user agent) accesses SAS Logon Manager (OAuth client)
- 2 SAS Logon Manager redirects the client browser to the SAS Logon Manager sign-in page. The end user clicks **OpenID Connect logon using ISAM Reverse Proxy (WebSEAL)**.
- 3 The client browser sends an authentication request to WebSEAL (resource owner).
- 4 WebSEAL redirects the client browser to the IBM Security Access Manager (ISAM) sign-in page. The end user provides authentication information.
- 5 The client browser sends the authentication information to the LDAP server.
- 6 The LDAP server authenticates the user with IBM Security Access Manager.
- 7 WebSEAL sends an authorization request to the ISAM federation run time (WebSphere Application Server).
- 8 The ISAM federation run time sends the authorization request to the authorization server.
- 9 The authorization server sends an authorization code to the ISAM federation run time.
- 10 The ISAM federation run time sends the authorization code to WebSEAL.
- 11 WebSEAL sends the authorization code to the client browser.
- 12 The client browser sends the authorization code to SAS Logon Manager.
- 13 SAS Logon Manager sends a request to the ISAM federation run time to convert the authorization code to an access token.

- 14 The ISAM federation run time sends the request to the authorization server.
- 15 The authorization server sends the access token to the ISAM federation run time.
- 16 The ISAM federation run time sends the access token to SAS Logon Manager.

Authentication: OIDC with Okta Scenario (Linux Full Deployment)

The following tasks provide one way to configure Okta as an OpenID Connect (OIDC) provider. In this scenario, OIDC uses Okta as the single sign-on entry point for initial user authentication. Other providers can also be used, but configuration instructions are not provided here.

In addition, you must configure LDAP or SCIM, which SAS Viya requires for loading users from Azure. For more information, see [“How to Configure SCIM” in SAS Viya Administration: Identity Management](#).

Create the Web Application Using the Okta Admin Console

- 1 Log on to Okta Developer Console and click **Admin**.

.....

Note: At a minimum, the *App Admin* permission is required.

.....

- 2 In the Okta Dashboard, navigate to **Applications** ⇒ **Add Application**.
 - a In the Create New Application window, select **Web**, and then click **Next**.
 - b Under Application Settings, enter values for the fields, based on your environment. [Table 27 on page 105](#) provides guidance about the information needed for the listed fields.

Table 27 *Web Application Settings and Suggested Values*

Setting	Suggested Value
Name	Provide a name that identifies your application with Okta.
Base URLs	Leave this field blank.
Login redirect URIs	Specifies the URI of the SAS Logon Manager OAuth configuration, where Okta redirects after

Setting	Suggested Value
	authentication. Use <code>https://hostname.example.com/SASLogon/login/callback/okta</code> . Note: The name, <i>okta</i> , is configured in Step 3d on page 107 using SAS Environment Manager. It is specified as the name field for the <code>sas.logon.oauth.providers</code> definition.
Logout redirect URIs	Specifies the URI of the SAS Logon Manager OAuth configuration. This value must match the Login redirect URI .
Group assignments	Select the users to whom you want to grant access.
Grant type allowed	Select Authorization Code .

- c Click **Done**.

In the Client Credentials section, note the **Client ID** and **Client secret** values.

- 3 In the Okta Dashboard, navigate to **API** ⇒ **Authorization Servers**.

Note: The following steps might differ, depending on your Okta configuration.

- a In the API window, select **default** ⇒ **Claims** and click **Add Claim**.
- b In the Add Claim window, enter values for the fields, based on your environment. [Table 28 on page 106](#) provides guidance about the information needed for the listed fields.

Table 28 Add Claim Settings and Suggested Values

Setting	Suggested Value
Name	Provide a name for your claim. For Okta, use sub_sas .
Include in token type	Select ID Token .
Value type	Select Expression .
Value	Specify the following sting: <code>String.substringBefore(user.email, "@")</code> .
Disable claim	Do not select this setting.
Include in	Select Any scope .

- c Click **Create**.

Configure OIDC Provider Properties for Okta

- 1 Display the configuration settings for your Okta authorization server.
 - a Log on to Okta Developer Console and click **Admin**.
 - b In the Okta Dashboard, navigate to **API** ⇒ **Authorization Servers**.
 - c In the API window, select **default**.
 - d Under Settings, note the value of the **Metadata URI** setting.
- 2 In a web browser, navigate to URL that was noted in [Step 1d on page 107](#) (for example, `https://hostname.example.com/oauth2/default/.well-known/oauth-authorization-server`).

The configuration settings for your authorization server are displayed. You will specify these settings in SAS Environment Manager.

- 3 Log on to SAS Environment Manager.

Note: In a multi-tenant environment, the following must be configured for the tenant.

- a From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 44](#).
- b In the **Definitions** list, select **sas.logon.oauth.providers**.
- c In the top right corner of the window, click **New Configuration**.
- d In the New sas.logon.oauth.providers Configuration window, enter values for the required fields, based on your environment. [Table 29 on page 107](#) provides guidance about the information needed for the listed fields. Some of your values might differ, depending on your Okta configuration.

Table 29 *OIDC Configuration Fields and Descriptions*

Configuration Field	Description
addShadowUserOnLogin	A local shadow user should be added once authentication is successful. This field is required. Note: This option should always be set to On .
attributeMapping.user_name	The attribute claim to use as the user name. For Okta, use sub_sas . This field is required. Note: The value sub_sas can be substituted with another claim that was created during the Okta configuration. This value must be the same as the Name specified in Step 3b on page 106 .

Configuration Field	Description
authUrl	The URL to the authorization endpoint (for example, <code>https://hostname.example.com/oauth2/default/v1/authorize</code>). This field is required.
clientAuthInBody	Specifies whether to include the client credentials in the request body. This field is optional. Note: This option should always be set to On .
discoveryUrl	Specifies the URL that is used to discover the provider and obtain information that is needed to interact with it. This field is optional. If a value is not specified, you must enter values for the issuer field and either the tokenKeyUrl or tokenKey field.
emailDomain	Specifies a comma-separated list of email domains for users that can sign in with the OIDC provider. It is used with identity provider discovery. This field is optional.
issuer	The principal that issued the token, specified as a case-sensitive string or URI. This is your Okta instance (for example, the entry point, <code>https://oidcidp.example.com</code>). To see the value for your server, see Step 2 on page 107 . This field is optional if you specify a value for the discoveryUrl field.
linkText	The text that should be displayed on the sign-in page for the provider (for example, OpenID Connect Login Using Okta). This field is optional.
name	Corresponds to the <code>redirect_uri</code> that is provided in the App registration that is created in the Okta portal. This field is required. IMPORTANT Do not include a period or other special characters in the value specified for the name.
relyingPartyId	The client ID that is registered with the provider. This value was noted in Step 2c on page 106 . This field is required.
relyingPartySecret	The client secret that is registered with the provider for the client ID. This value was noted in Step 2c on page 106 . This field is optional.
scopes	The comma-delimited list of scopes for the authorization request. The list should contain openid . This field is required.

Configuration Field	Description
	Note: SAS Viya does not process any additional scopes that are returned in the token.
showLinkText	The link text should show on the sign-in page. This field is required.
tokenKey	Specifies the HMAC key or RSA public key that is used to sign ID tokens. This field is optional if you specify a value for discoveryUrl . Note: If a value for the discoveryUrl field is not specified, either the tokenKey or tokenKeyUrl field must be specified.
tokenKeyUrl	Specifies the URL to obtain the signing key. This field is optional if you specify a value for discoveryUrl . Note: If a value for the discoveryUrl field is not specified, either the tokenKey or tokenKeyUrl field must be specified.
tokenUrl	The URL to obtain tokens from the provider. This field is required.
type	The protocol type. By default, the value is oidc1.0 . This field is required. Note: SAS Viya requires an <code>id_token</code> in the authorization response from the provider. However, some providers return an <code>id_token</code> when the scope in the authorization request is <code>openid</code> and <code>response_type=token</code> . For those providers, use type oauth2.0 .

e Click **Save**.

4 Restart the SAS Logon Manager Service.

- For Red Hat Enterprise Linux 6.7:

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

- For Red Hat Enterprise Linux 7.x or later and SUSE Linux:

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

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

5 Test your configuration by accessing SAS Environment Manager. The text that you specified in the `linkText` field in [Step 3d on page 107](#) should be displayed.

- 6 If you see an error that refers to a violation of a "Content Security Policy directive," your program requires a change to the content security policy setting for SAS Logon Manager. Follow the steps that are described in ["Update the Content Security Policy"](#) on page 48.

Authentication: Passwords

Update Account Passwords on Windows

SAS Cloud Analytic Services

If you change the SAS Cloud Analytic Services (CAS) service account password on the Windows domain, you must complete the following:

- 1 If you use a single account for the HTTP service and CAS service accounts, re-create the keytab file.
- 2 For any Windows service that is running as the CAS user, update the password.
 - a In Windows Services Manager, right-click the service name and select **Properties**.
 - b In the Properties window, select the **Log On** tab.
 - c Update the password and click **OK**.
- 3 Run the encryptCasUser.bat command to update the deployment scripts. For more information, see ["Specify Credentials for the cas User Account"](#) in *SAS Viya for Windows: Deployment Guide*.

SAS Infrastructure Data Server

If you change the PostgreSQL user account password on Windows, you must complete the following:

- 1 For any Windows service that is running as the PostgreSQL user, update the password.
 - a In Windows Services Manager, right-click the service name and select **Properties**.
 - b In the Properties window, select the **Log On** tab.
 - c Update the password and click **OK**.
- 2 Regenerate the credentials file each time the password changes. For more information, see ["Specify Credentials for the postgres User Account"](#) in *SAS Viya for Windows: Deployment Guide*.

Authentication: Reference

CAS Environment Variables for Clients

The environment variables that are described in this section are set on the client. These variables affect how clients authenticate with the CAS server.

CAS_AUTH_METHOD=authinfo | kerberos

specifies the authentication method that CAS clients use.

Valid in operating system command line

Category Security

Linux specifics Environment variables on Linux are case-sensitive.

See [Authinfo File Authentication](#)

Examples In these examples, the CAS client is forced to authenticate using the credentials in the authinfo file (Kerberos authentication is not attempted). Here are two examples of specifying the command for Linux.

```
export CAS_AUTH_METHOD=authinfo
```

```
set CAS_AUTH_METHOD=kerberos
```

CASSPN=HTTP@FQDN

specifies the service principal name (SPN) that the CAS clients use to connect to the CAS server in a Kerberos environment. When connecting to a SAS Viya 3.5 CAS server, the SPN should be specified as `sascas@FQDN`. When connecting to a SAS Viya 4 CAS server, the SPN should be specified as `HTTP@FQDN`, where `FQDN` is the fully qualified domain name of the primary node of the Kubernetes cluster.

When setting this value from a SAS 9.4 program, use `CASSPN=HTTP/FQDN`, and enclose the value of `CASSPN` in quotation marks.

If you are using SAS Scripting Wrapper for Analytics Transfer (SAS SWAT), this variable is also used to connect the python-swat client to the CAS server running in SAS Viya using Kerberos. On the python-swat client, when connecting to a SAS Viya 3.5 CAS server, set the `CASSPN` environment variable to the string `sascas@service-principal-name`, where `service-principal-name` is the Kerberos service principal name. Make this modification before invoking SAS SWAT.

Valid in operating system command line

Category Security

Linux specifics Environment variables on Linux are case-sensitive.

Examples In this example, the python-swat client will use the SPN string "sascas@myhost.example.com" to connect to a SAS Viya 3.5 CAS server when using Kerberos.

```
export CASSPN=sascas@myspn.example.com
```

CAS Environment Variables for Administrators

The environment variables in this section affect authentication with the CAS server.

env.CASSTRIPOAUTH=1

strips the `@domain` that is a part of the user name when authenticating a user name and password with SASLogon.

This environment variable helps in resolving a mismatch between host authentication and LDAP authentication when host authentication requires "user@domain" format user names, and LDAP authentication requires "user" format user names without the @domain part with SASLogon.

Valid in `cas_usermods.settings`

Category Environment

Default Does not strip the domain name that is specified after the @ character in the user name.

Example Here is an example where the user name is `user01@company.com`. It is authenticated as:

```
'user01'
```

CASUSERIGNORECASE='ON'

when in effect (specified using any value), causes the CAS server to ignore the letter casing for user names during authentication, group lookup, and process launch. Always specify `CASUSERLOWERCASE` whenever specifying `CASUSERIGNORECASE`, unless instructed otherwise by SAS Technical Support.

The typical scenario for declaring `CASUSERIGNORECASE` is when users run their CAS sessions under their own host account and the user authentication system is configured to be case-insensitive and contains uppercase or mixed case user names. For more information, see ["The CASHostAccountRequired Custom Group" in SAS Viya Administration: Identity Management](#).

Valid in [cas_usermods.settings file](#)

Category Administration

Default off

Restrictions Applies to Linux only.

`CASUSERIGNORECASE` is case sensitive.

Requirement	Use with CASUSERLOWERCASE.
Note	To turn off CASUSERIGNORECASE, remove its definition.
Example	In this example, CASUSERIGNORECASE is in effect: <code>export CASUSERIGNORECASE='on'</code>

CASUSERLOWERCASE='ON'

when in effect (specified using any value), causes the CAS server to convert user names to lowercase during group lookup. CASUSERLOWERCASE is typically used in conjunction with CASUSERIGNORECASE.

The typical scenario for declaring CASUSERLOWERCASE is when users run their CAS sessions under their own host account and the user authentication system is configured to be case-insensitive and contains uppercase or mixed case user names. For more information, see [“The CASHostAccountRequired Custom Group” in SAS Viya Administration: Identity Management.](#)

Valid in	cas_usermods.settings file
Category	Administration
Default	off
Restrictions	Applies to Linux only. CASUSERLOWERCASE is case sensitive.
Requirement	Use with CASUSERIGNORECASE.
Note	To turn off CASUSERLOWERCASE, remove its definition.
Example	In this example, CASUSERLOWERCASE is in effect: <code>export CASUSERLOWERCASE='on'</code>

CASUSERFORCELOWER='ON'

when in effect (specified using any value), causes the CAS server to convert user names to lowercase. Using CASUSERFORCELOWER eliminates the need for either CASUSERIGNORECASE or CASUSERLOWERCASE.

The typical scenario for declaring CASUSERFORCELOWER is when users run their CAS sessions under their own host account and the user authentication system is configured to be case-insensitive and contains uppercase or mixed case user names. For more information, see [“The CASHostAccountRequired Custom Group” in SAS Viya Administration: Identity Management.](#)

Valid in	cas_usermods.settings file
Category	Administration
Default	Off
Restriction	CASUSERFORCELOWER is case sensitive.
Note	To turn off CASUSERFORCELOWER, remove its definition.
Example	In this example, CASUSERFORCELOWER is in effect: <code>export CASUSERFORCELOWER='on'</code>

env.CASSPN=HTTP@FQDN

specifies the service principal name (SPN) that the CAS clients use to connect to the CAS server in a Kerberos environment. When connecting to a SAS Viya 3.5 CAS server, the SPN should be specified as `sascas@FQDN`. When connecting to a SAS Viya 4 CAS server, the SPN should be specified as `HTTP@FQDN`, where *FQDN* is the fully qualified domain name of the primary node of the Kubernetes cluster.

Category Security

See Also

[“Where Do I Set CAS Environment Variables?” in SAS Viya Administration: SAS Cloud Analytic Services](#)

Authentication: External Languages Package

Overview

Because the functionality of the EXTLANG package might conflict with your site's security policies, the package includes mechanisms that enable administrators of SAS Cloud Analytic Services (CAS) to control users access to installed external-language interpreters and usable storage areas.

Administrators can do the following:

- Specify *allowlisted* directories, which are directories to which users can write or that contain source code that users are allowed to load.
- Control whether users can specify paths to the executables for interpreting and compiling external-language programs.
- Control whether users can insert external-language source code directly into their SAS programs.
- Control whether users can specify environment variables for their external-language programs.

By taking advantage of the hierarchical nature of XML, users can specify attributes as concisely as possible on a global, per-group, per-language, or per-group-per-language basis. The following sections describe the configuration file that is used for access control. The path to this XML-formatted file is specified using the CAS environment variable, `SAS_EXTLANG_SETTINGS`. For information about where to set environment variables for your particular CAS deployment, see [“CAS Environment Variables” in SAS Viya Administration: SAS Cloud Analytic Services](#).

External Languages Access Control Configuration

By default, the EXTLANG package is disabled, all text attributes are set to an empty string, and all Boolean values are set to 'BLOCK'. The following XML schema can be configured in the configuration file. The EXTLANG tag is required, but the other tags are optional.

EXTLANG

Contains attributes that control the default settings for nested tags. Users can specify the following attributes in the EXTLANG tag:

version=level

Specifies the XML document's version number. This attribute exists to accommodate future changes to the XML definition.

mode='ALLOW' | 'ANARCHY' | 'BLOCK'

Specifies the access mode for the EXTLANG package. Possible values for this attribute are:

'ALLOW'

Allows use of the EXTLANG package and initializes all Boolean properties to 'BLOCK'. (The properties can be explicitly overridden).

'ANARCHY'

Allows use of the EXTLANG package and initializes all Boolean properties to 'ALLOW'. (the properties can be explicitly overridden).

'BLOCK'

Does not allow use of the EXTLANG package.

allowAllUsers='ALLOW' | 'BLOCK'

Controls which users can run external-language programs. Possible values for this attribute are:

'ALLOW'

Allows all users to use the EXTLANG package. As the administrator, you can still specify GROUP blocks to specify group-specific overrides. For users who are not specified in a GROUP block, the attributes from the DEFAULT tag are applied.

'BLOCK'

Allows only users specified in a GROUP block to use the EXTLANG package.

The following tags are all optional:

DEFAULT

Enables users to specify global default settings. Any attributes that are specified in the DEFAULT block override the values that are initialized according to the mode= attribute that is specified in the EXTLANG start tag. These settings cascade down to all LANGUAGE and GROUP blocks that are nested in the EXTLANG block. The following attributes can be specified in the DEFAULT tag:

scratchDisk=location

Specifies a location on the file system in which to store temporary files that are used to enable support for external languages. This location must have enough space to store shared variables (that is, variables that are transferred between the SAS and external-language environments). Environment variables are expanded. By default, the system-defined default temporary directory is used.

diskAllowlist=paths

Specifies one or more file system paths that users can use for the following:

- Loading source code from files.
- Specifying as their *scratchDisk*.
- Specifying as options specific to certain objects. For example, the 'TEMPDIR' and 'EXECPATH' options of the PYTHON2, PYTHON3, and R objects of the EXTLANG package

Users can specify multiple paths by separating them with the platform-specific path separator, which is ":" on Linux systems. Environment variables are expanded. The escape character is the "\". A literal "\" can be inserted using "\\". Attempting to push a file that is not inside an allowlisted directory or its subdirectories will result in an error. The diskAllowList= attribute does not affect the ability to insert external-language source code in other ways. For example, users can still insert code inline. Attempting to specify a path for the scratchDisk= attribute that is not inside this directory will result in an error.

userSetScratchDisk='ALLOW' | 'BLOCK'

Controls whether users are allowed to specify the path to the location that is specified in the scratchDisk= attribute. By default, userSetScratchDisk='BLOCK'.

'ALLOW'

Allows users to specify the scratch disk.

'BLOCK'

Prevents users from specifying the scratch disk. A run-time error will occur if a user program attempts to set the scratch disk.

userSetEnv='ALLOW' | 'BLOCK' | 'UNDERSCORE'

Controls whether users are allowed to specify environment variables to be passed to the external-language program via the AddEnvVariable method. By default, userSetEnv='BLOCK'. The following values can be specified:

'ALLOW'

Allows users to specify environment variables whose names consist of a string of ASCII characters.

'BLOCK'

Does not allow users to specify environment variables.

'UNDERSCORE'

Allows users to specify environment variables whose name consists of an underscore followed by a string of ASCII characters.

userSetInterpreter='ALLOW' | 'BLOCK'

Controls whether users can specify the external-language executable's path. By default, userSetInterpreter='BLOCK'.

'ALLOW'

Allows users to specify the path to the external-language interpreter.

'BLOCK'

Does not allow users to specify the path to the -language interpreter. An error will occur if a user attempts to specify the path.

LANGUAGE

Enables users to override default settings on a per-language basis. Users can nest one or more LANGUAGE tags directly in the DEFAULT block or within a GROUP block. Any attribute specified

in a LANGUAGE tag becomes the default value for that language (that is, those attributes cascade down for all GROUP blocks that do not specify them for that language). If a LANGUAGE tag is nested within a GROUP block, the specified attributes apply only to users within that group.

name='PYTHON2' | 'PYTHON3' | 'R'

Specifies the name of the language being configured. Users must specify this attribute.

Users can also specify the following optional attributes in the LANGUAGE tag. Unspecified attributes inherit the corresponding attribute value from the DEFAULT block, which inherits default values according to the mode attribute of the EXTLANG start tag:

interpreter=*path*

Specifies the default path to the external-language executables. This path will be used by all workers in the CAS cluster. Attempting to use objects of the EXTLANG package will result in an error if the interpreter and userSetInterpreter attributes are not specified.

userInlineCode='ALLOW' | 'BLOCK'

Controls whether users can add external-language code from within their SAS programs.

'ALLOW'

Allows users to insert inline code.

'BLOCK'

Does not allow users to insert inline code. By default, userInlineCode='BLOCK'.

userSetEnv='ALLOW' | 'BLOCK' | 'UNDERSCORE'

Controls whether users can specify environment variables. Acceptable values are:

'ALLOW'

Allows users to specify environment variables whose names consist of a string of ASCII characters.

'BLOCK'

Does not allow users to specify environment variables. By default, userSetEnv='BLOCK'.

'UNDERSCORE'

Allows users to specify environment variables whose name consists of an underscore followed by a string of ASCII characters.

userSetInterpreter='ALLOW' | 'BLOCK'

Controls whether users can specify the path to the language executable.

'ALLOW'

Allows users to set the interpreter executable.

'BLOCK'

Does not allow users to set the interpreter executable. An error will occur if a user attempts to specify the path to the interpreter.

ENVIRONMENT

Enables users to set environment variables. Users can nest one or more ENVIRONMENT tags within a LANGUAGE tag. Environment variables are set in the external-language interpreter's running environment. In order for environment variables to be passed, the value of the userSetEnv attribute of the enclosing LANGUAGE block must be 'ALLOW' or 'UNDERSCORE'.

Note: A run-time error will occur under these conditions:

- if userSetEnv='BLOCK' and a user attempts to set an environment variable

- if `userSetEnv='UNDERSCORE'` and a user attempts to set an environment variable whose name does not begin with an underscore

Administrators can specify variables that begin with underscore regardless.

Users must specify the following attributes in the ENVIRONMENT tag:

`name=variableName`

Specifies the environment variable name, which must be a string of ASCII characters

`value=variableValue`

Specifies the value of the variable, which must be a string of Unicode characters

GROUP

Specifies group-specific overrides. Every user must be specified in a GROUP block, unless `allowAllUsers='ALLOW'` in the EXTLANG tag. Users cannot belong to multiple groups. Any languages that do not contain a LANGUAGE block that is enclosed by the GROUP block inherit the default language attributes that are defined in the corresponding LANGUAGE block that is enclosed by the DEFAULT block. The default language attributes can themselves be implicitly defined according to the mode attribute of the EXTLANG start tag.

`name=groupName`

Specifies a string that identifies the group. If no users attribute is specified in the GROUP tag, the attributes that are defined in this GROUP block are applied to the operating system account whose user name is `groupName`. Users must specify the following attribute in the GROUP tag.

Users can also specify the following attributes of the GROUP start tag:

`users=list`

Specifies a comma-delimited list of user names to which the settings that are defined in this GROUP block will apply. Each user must belong to only one group. The escape character is `"\"`. A literal `"\"` can be inserted using `"\""`.

`scratchDisk=directory`

Specifies the temporary working directory for this group. By default, the location that is specified in the `scratchDisk=` attribute in the DEFAULT block is used.

`diskAllowList=paths`

Specifies a list of *allowlisted* paths. Users can specify multiple paths by using the operating system's native path separator (which is `:` on Linux). If this attribute is specified, source code that the users of this group push from a file must reside under a path in this list. The location that is specified in the `scratchDisk=` attribute must also reside under a path in this list.

`userSetScratchDisk='ALLOW' | 'BLOCK'`

Specifies whether the users in this group are allowed to specify the scratch disk.

'ALLOW'

Allows users in this group to set the `scratchDisk` location.

'BLOCK'

Does not allow users to set the `scratchDisk` location. An error will occur if a user attempts to specify the `scratchDisk` location.

Sample Access Control File for the EXTLANG Package

Content similar to the following can be defined in the access control configuration file:

```
<EXTLANG version="1.0" mode="ALLOW" allowAllUsers="BLOCK">
  <DEFAULT scratchDisk="/smalldisk/sas/scratch"
    diskAllowlist="/secure/sas/allowed_scripts:/allowlist"
    userSetScratchDisk="BLOCK"
    userSetEnv="BLOCK"
    userSetInterpreter="BLOCK">
    <LANGUAGE name="PYTHON2"
      userInlineCode="BLOCK"
      interpreter="/some/path/python2"
      userSetEnv="ALLOW"
      userSetInterpreter="BLOCK">
      <ENVIRONMENT name="PYTHONPATH" value="/some/path1:/some/path2" />
      <ENVIRONMENT name="LDLIBRARYPATH" value="/some/ldpath1:/some/ldpath2" />
    </LANGUAGE>
    <LANGUAGE name="PYTHON3"
      interpreter="/some/path/python3"
      userSetEnv="BLOCK"
      userSetInterpreter="BLOCK">
      <ENVIRONMENT name="PYTHONPATH" value="/some/path1:/some/path2" />
      <ENVIRONMENT name="LDLIBRARYPATH" value="/some/ldpath1:/some/ldpath2" />
    </LANGUAGE>
    <LANGUAGE name="R"
      interpreter="/some/path/Rscript"
      userSetEnv="BLOCK"
      userSetInterpreter="BLOCK">
      <ENVIRONMENT name="LDLIBRARYPATH" value="/some/ldpath1:/some/ldpath2" />
    </LANGUAGE>
  </DEFAULT>
  <GROUP name="DanDLyons"
    scratchDisk="$HOME/scratch" />
  <GROUP name="SassySean"
    userSetInterpreter="ALLOW"
    userSetEnv="UNDERSCORE">
    <LANGUAGE name="PYTHON3"
      interpreter="$HOME/anaconda/bin/python3.5">
      <ENVIRONMENT name="_ALGORITHM" value="BEST" />
    </LANGUAGE>
  </GROUP>
  <GROUP name="sasUsers"
    users="Sam, Ada, Sergey"
    scratchDisk="/pan1"
    diskAllowlist="/home/$USER:/authorized/path"
    userSetScratchDisk="BLOCK"
    userReadDisk="BLOCK">
    <LANGUAGE name="PYTHON2"
      userInlineCode="ALLOW"
```

```

        userSetEnv="BLOCK"
        userSetInterpreter="BLOCK" />
    </GROUP>
</EXTLANG>

```

The following provides details about the above sample code:

- In the EXTLANG tag:
 - Specifying mode='ALLOW' enables the EXTLANG package and initializes all Boolean attributes to 'BLOCK'.
 - Specifying allowAllUsers='BLOCK' prohibits users whose user name does not appear in a GROUP from using the package.
- In the DEFAULT tag, restrictive settings initialized in the EXTLANG tag are overridden. Five settings are overridden in this block:
 - The scratchDisk= *attribute* is set to /smalldisk/sas/scratch. All temporary files that the EXTLANG package creates will go here; it must be large enough to accommodate all data sets that every user must work with.
 - The diskAllowlist= *attribute* specifies that users can push code files that reside only in the /smalldisk/sas/scratch and /allowlist directories.
 - The userSetScratchDisk, userSetEnv, and userSetInterpreter attributes are set to 'BLOCK'; this is done for readability since they were all initialized by mode='ALLOW' in the enclosing EXTLANG tag.
- In the LANGUAGE tags, default language settings are entered.
 - For PYTHON2, the interpreter path is specified and users are not allowed to change this setting in their program (unless the setting is overridden in a GROUP block). Users are not able to set environment variables. Default values are then defined for 'PYTHONPATH' and 'LDLIBRARYPATH' environment variables.
 - The settings for PYTHON3 are the same as for PYTHON2, except the interpreter executable path is different.
 - A similar configuration is specified for R, but only the 'LDLIBRARYPATH' environment variable is set.
- GROUP tags are used to override default settings for users.
 - The first GROUP block does not have a users= *attribute*, so the settings that it overrides will apply to user DanDLyons. All external-language programs run by DanDLyons will be stored in the scratch subdirectory within DanDLyons' home directory.
 - The next GROUP block does not have a users= *attribute*, so the settings that it overrides apply only to user SassySean. This user is assigned a different default Python 3 interpreter, which SassySean can change programmatically because userSetInterpreter='ALLOW'. The default restriction on setting environment variables is also relaxed so that SassySean can set environment variables as long as they begin with an underscore.
 - The final GROUP tag specifies multiple users to which the specified attributes apply. These users' temporary data will be stored in /pan1 and they can push only scripts that are under their home directory. The userInlineCode= *attribute* is overridden to 'ALLOW', so these users can add code in their SAS program. The remainder of the settings are inconsequential because they match the defaults.

Authentication: Troubleshooting

Event ID 14 error event on Domain Controller

With a patch for Windows that Microsoft issued in November 2022, a new default encryption type, AES, is applied to session keys. This change, which addresses a privilege vulnerability in Authentication Negotiation, affects accounts that do not have a default encryption type set, or that have a default set to an encryption type that has been deprecated. If you are running SAS with Kerberos and Active Directory, you might be using RC4-HMAC as the encryption type for Active Directory, and you will experience issues as a result of this change. If your KDC or your Active Directory environment is affected, you might see a Microsoft-Windows-Kerberos-Key-Distribution-Center Event ID 14 error event in the System section of the Event Log on your Domain Controller. The event in the log will include the following text:

```
While processing an AS request for target service service, the account account name
did not have a suitable key for generating a Kerberos ticket (the missing key has an ID of 1).
The requested etypes : 18 3. The accounts available etypes : 23 18 17.
Changing or resetting the password of <account name> will generate a proper key.
```

This error is logged because the encryption that is available for older versions of Java is no longer sufficient after the settings in the Windows patch have been applied. If you have an older version or maintenance release of SAS®9, the accompanying Java Runtime Environment (such as the SAS Private JRE) is likely to hit this error. Therefore, this Microsoft patch provides a good reason to upgrade SAS or migrate to SAS Viya, which provides a newer version of Java with compliant encryption.

Resolution:

To resolve this error, you must configure unlimited-strength encryption for Active Directory. Microsoft has provided a fix that you can apply to your Domain Controllers at the following website: <https://learn.microsoft.com/en-us/windows/release-health/status-windows-11-22h2#2953msgdesc>.

You must also replace any keytab files that only contain the RC4-HMAC keys. Validate that the keytab files that are used with SAS have been modified to contain AES encryption keys by running the following command:

```
klist -ket /opt/sas/http.keytab
```

If the AES encryption keys are present, results that resemble the following are returned:

```
rw----- 1 sas sas 508 Nov 18 03:22 /opt/sas/http.keytab
Keytab name: FILE:/opt/sas/http.keytab
KVNO Timestamp                Principal
-----
 2 11/18/2019 03:20:50 HTTP/sasviya01.gellab.net@GELLAB.NET (arcfour-hmac)
 2 11/18/2019 03:20:50 HTTP/sasviya01.gellab.net@GELLAB.NET (aes128-cts-hmac-sha1-96)
 2 11/18/2019 03:20:50 HTTP/sasviya01.gellab.net@GELLAB.NET (aes256-cts-hmac-sha1-96)
```

One additional point to note is that this KDC change does not affect the contents of the client or server `krb5.conf` file because the KDC will not use RC4-HMAC encryption.

After applying the fix for the Domain Controllers and replacing the keytab files, you should validate that the SAS Private JRE that you are using has required Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction policy files. The steps to determine whether these files are present depend on the SAS platform. Consult the following table for the steps:

Table 30 Validate JRE Encryption Settings

SAS Platform	Validation Steps
SAS 9.4	<p>Run the following command:</p> <pre>ls -l /opt/sas/SASHome/ SASPrivateJavaRuntimeEnvironment/9.4/jre/lib/ security/policy/unlimited</pre> <p>Note: In some SAS Visual Analytics environments (for SAS 9.4) that use co-located Hadoop, a separate JRE deployment could be present. That JRE also requires validation. Here is an example of the location to check: <code>JAVA_HOME\jre\lib\security</code>.</p>
SAS Viya 3.x	<p>Run the following command:</p> <pre>ls -l /usr/lib/jvm/java-1.8.0-openjdk-build- id.x86_64/jre/lib/security/policy/unlimited/</pre>

For either of these commands, check the output for the following files:

- local_policy.jar
- US_export_policy.jar

Instructions for updating the SAS Private JRE are provided in the following SAS Note: <https://support.sas.com/kb/56/203.html>. If migrating to SAS Viya or updating the SAS Private JRE is not an option, you can also add policy files to the JRE.

After configuring Kerberos for SAS Logon Manager, you are unable to log on to a visual interface, such as SAS Environment Manager.

Resolution:

You must use a web browser on a different machine. Once Kerberos is enabled on Windows, a browser running on the same machine where the services are deployed cannot connect to SAS Viya visual interfaces.

The Kerberos authentication handshake fails and a session is not launched.

Resolution:

Users can store their credentials from the **My Credentials** page. Then, if the Kerberos handshake fails, authentication will fallback to the stored credentials in DefaultAuth. For more information, see [“Add New Credentials” in SAS Viya Administration: External Credentials](#).

After Kerberos is configured for SAS Logon Manager, no one is able to log on to SAS Environment Manager.

Resolution:

If the information that you specified while adding Kerberos to the active profile, `profiles.active`, is incorrect or missing, the only way to change the information is by using the SAS Bootstrap Config CLI.

Run the following command on a single line. Multiple lines are used for each command to improve readability.

```
/opt/sas/viya/home/bin/sas-bootstrap-config --token-file $consul-token kv write --force  
config/SASLogon/spring/profiles.active ldap,postgresql
```

For more information, see [“Use SAS Bootstrap Config CLI on Consul to Manage the KV Store and ACL Tokens”](#) in *Encryption in SAS Viya: Data in Motion*.

Linux group lookup fails when user names are uppercase or mixed case.**Resolution:**

This problem typically occurs when Active Directory is used as the back-end user store. Use the `CASUSERIGNORECASE` environment variable to force SAS Cloud Analytic Services (CAS) to ignore letter casing during authentication and session launch. In addition, use the `CASUSERLOWERCASE` environment variable to force CAS to use the lowercase version of the user’s name when doing the group lookup. For more information, see [“CASUSERIGNORECASE=’ON’”](#) on page 112.