

SAS® Viya® 3.4 Administration: Authentication

Authentication: Overview	2
Authentication Options	2
Authentication: How To	2
Authentication Mechanisms	2
Session Management Using SAS Environment Manager	24
Additional Authentication Topics	25
Authentication: Concepts	27
Authentication Architecture	27
Authentication and SAS Viya Services	27
In-bound and Out-bound Authentication	28
Authentication Options	28
Authentication Mechanisms	33
Additional Authentication Topics	46
Authentication: Guest Access (Linux)	47
About Guest Access	47
Enable Guest Access	48
Connect as Guest Users	49
Generate Custom Links to Reports	50
Disable Guest Access	50
Authentication: OpenID Connect Scenario (Linux Full Deployment)	52
Configure OpenID Connect Provider Properties for IBM Security Access Manager	52
Configure OpenID Connect Provider in IBM Security Access Manager	53
OpenID Connect and IBM Security Access Manager	55
Authentication: Reference	56
CAS Environment Variables for Clients	56
CAS Environment Variables for Administrators	56
Authentication: Troubleshooting	57

Authentication: Overview

Authentication is the process of verifying the identity of a user that is attempting to log on to or access software. In SAS Viya, authentication options vary, based on which interface and operating system are being used in your environment:

Table 1 *Authentication Options*

Type of Deployment	Operating System	Authentication Mechanism
full deployment	Linux	<p>The pluggable authentication module (PAM) validates the user's credentials when accessing SAS Studio 4.x and CAS Server Monitor.</p> <p>Batch jobs submit credentials that require validation.</p> <p>Users can be authenticated through SAS Logon Manager, using an LDAP provider, Kerberos, Security Assertion Markup Language (SAML), PAM, or OAuth and OpenID Connect.</p>
	Windows	<p>Host authentication validates the user's credentials when accessing SAS Studio 4.x 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 include direct LDAP authentication (which is referred to as LDAP in this documentation), host authentication, Kerberos, Security Assertion Markup Language (SAML), and OAuth 2.0 with OpenID Connect. Pluggable authentication modules (PAM) extend UNIX host authentication.

The following sections are listed alphabetically. Configure the authentication mechanism that is appropriate for your environment. For more information, see ["Authentication Mechanisms" on page 33](#).

Note: On Windows deployments, Windows host authentication validates the user's credentials when accessing SAS Studio 4.x and CAS Server Monitor, and for batch jobs. For SAS Viya visual interfaces and configuration of the middle-tier environment, Kerberos is the only supported authentication mechanism.

Configure Kerberos (Linux Full Deployment)

To configure Kerberos on Linux, you must do the following:

- Perform prerequisites to verify that certain conditions are met.
- Set up SAS Logon Manager, SAS Cloud Analytic Services, and SAS Launcher Server.
- Configure your web browser for Kerberos.

Verify Kerberos Prerequisites

Before configuring Kerberos, make sure that the following exists:

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

- 1 Three 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 Servers for Which Kerberos Can Be Configured on Linux

Server	Service Class
SAS Logon Manager	HTTP
SAS Cloud Analytic Services	sascas
SAS Launcher Server	sas-launcher

- 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. This service account must be trusted for unconstrained delegation, allowing delegation to all services.

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`

- 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 (arcfour-hmac)
```

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 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

- 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 LDAP provider. It must match the user ID that you use to log on to your system. Also, 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 SPN is mapped to the user principal name.

```
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 24](#).
 - 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 Configuration Fields and Values

Field	Value	Description
debug	On	Specifies whether to write debug messages in the log.
disableDelegationWarning	Off	Specifies whether to display a warning message to users when Kerberos credential delegation is not properly configured. Note: This option is available starting in the May 2019 release of SAS Viya 3.4.
holdOnToGSSContext:	On	This option is required to enable Kerberos delegation from SAS Logon Manager.
keyTabLocation	<i>file:///path-to-http-keytab-file</i> Note: You must use forward slashes (for example, <i>file:///c:/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, <i>HTTP/fully.qualified.hostname@REALM</i>).	On Linux, issue the <code>ktab -l -k FILE:path-to-http-keytab-file.keytab</code> command.
spn	<i>service-principal-name</i>	Specifies the service principal name, if it differs from the principal name in the keytab.
stripRealmForGss	On	Specifies whether to remove the realm from the user principal name.
impersonate	Off	Specifies whether to impersonate the user credentials using the Microsoft S4USelf extension to Kerberos for outgoing connections.

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 **spring** instance, click .
- c In the Edit spring Configuration window, add **kerberos** to the **profiles.active** field.

The following value should be specified for the **profiles.active** field:

```
ldap,postgresql,kerberos
```

d 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 Kerberos for SAS Cloud Analytic Services

1 Create a keytab file for CAS to use.

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 reside in the `/etc/sascas.keytab` file and be readable only by CAS. If you save the file in a different directory or use a different filename, set the `KRB5_KTNAME` environment variable (for example, `env.KRB5_KTNAME = 'fully-qualified-filename'`) to the fully qualified filename. For more information, see “[CAS Environment Variables](#)” in *SAS Viya Administration: SAS Cloud Analytic Services*.

2 Verify that the 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, CAS uses the following Kerberos principal name: `sascas/fully-qualified-DNSname`. CAS 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 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 24.

- 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 6.7:

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

- 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 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 /opt/sas/viya/config/consul.conf
```

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

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

Note: The previous command must be on one line. It is shown on more than one line for display purposes only.

- 6 Restart SAS Launcher Server.


- For Red Hat Enterprise Linux 6.7:

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

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

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

- 7 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 24](#).
- 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**.

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 Microsoft Edge and Google Chrome to Use Kerberos

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 check boxes for the following items are selected:
 - **Include all local (Intranet) sites not listed in other zones**
 - **Include all sites that bypass the proxy server**
 - b Click **Advanced** and add your domain name to the **Websites** list to ensure that Internet Explorer recognizes any site with your domain name as the intranet.
 - c 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 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 which servers Microsoft Edge can delegate to. 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 Kerberos (Windows Full Deployment)

To configure Kerberos on Windows, you must do the following:

- Perform prerequisites to verify that certain conditions are met.
- Set up SAS Logon Manager.
- Configure your web browser for Kerberos.

Verify Kerberos Prerequisites

Before configuring Kerberos, make sure that the following exists:

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

- 1 Two 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 4](#) is mapped to the service accounts from [Step 2](#).

Table 4 Servers for Which Kerberos Can Be Configured on Windows

Server	Service Class
SAS Logon Manager	HTTP
SAS Cloud Analytic Services	sascas

- 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 4](#).

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.

Note: For CAS, the SPN must be registered on the service account that is running the server. This service account must be trusted for unconstrained delegation, allowing delegation to all services. If an SPN is not found, then contact your information technology support group for assistance with registering the machine.

- 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 will be 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 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. The server on which it is running must be marked in Active Directory as trusted for delegation to any service. If the machine 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 both 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

- 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 LDAP provider. It must match the user ID that you use to log on to your system. Also, 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 SPN is mapped to the user principal name.

```
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 24](#).
 - 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 5 Configuration Fields and Values

Field	Value	Description
debug	On	Specifies whether to write debug messages in the log.
disableDelegationWarning	Off	Specifies whether to display a warning message to users when Kerberos credential delegation is not properly configured. Note: This option is available starting in the May 2019 release of SAS Viya 3.4.
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 (for example, file:///c:/path-to-http-keytab-file).	Specifies the Uniform Resource Identifier (URI).


Field	Value	Description
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.exe -l -k FILE:path-to-http-keytab-file.keytab</code> command from the directory where Java is installed on your machine.
spn	<i>service-principal-name</i>	Specifies the service principal name, if it differs from the principal name in the keytab.
stripRealmForGss	On	Specifies whether to remove the realm from the user principal name.
impersonate	Off	Specifies whether to impersonate the user credentials using the Microsoft S4USelf extension to Kerberos for outgoing connections.

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 **spring** instance, click .

c In the Edit spring Configuration window, add **kerberos** to the **profiles.active** field.

The following value should be specified for the **profiles.active** field:

```
ldap,postgresql,kerberos
```

d Click **Save**.

6 Restart the SAS Logon Manager service. 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.

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.

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
```

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] <non_delegated_user> local MAIN NoUser [tkidentgss.c:741] - User
<non_delegated_user>@<domain_name> successfully authenticated using the OAuth authentication provider.
```

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.

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 Microsoft Edge and Google Chrome to Use Kerberos

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 check boxes for the following items are selected:
 - **Include all local (Intranet) sites not listed in other zones**
 - **Include all sites that bypass the proxy server**
 - b Click **Advanced** and add your domain name to the **Websites** list to ensure that Internet Explorer recognizes any site with your domain name as the intranet.
 - c 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 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 which servers Microsoft Edge can delegate to. 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 OpenID Connect (Linux Full Deployment)

Configure OpenID Connect

OpenID Connect uses a reverse proxy server as the single sign-on entry point for initial user authentication. To configure the OpenID Connect, complete the following:

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 24](#).
- 2 In the **Definitions** list, select **sas.logon.oauth.providers.external_oauth**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New **sas.logon.oauth.providers.external_oauth** 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 6 OAuth Configuration Fields, Values, and Descriptions

Configuration Field	Value	Description
<code>addShadowUserOnLogin</code>	On Note: This option should always be On .	Specifies that a local shadow user should be added once authentication is successful.
<code>attributeMapping.user_name</code>	By default, the value is <code>user_name</code> .	Specifies the attribute from the provider, which contains the user name. The value specified is used by the <code>scopes</code> option.
<code>authUrl</code>	No default value	Specifies the URL to the authorization endpoint of the third-party.
<code>emailDomain</code>	<code>domain_name1, domain_name2, domain_name3</code>	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 and is optional.

Configuration Field	Value	Description
issuer	<code>https://hostname/auth/realms/<i>realm_name</i></code>	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.
linkText	The default value is “Use your corporate credentials”.	Specifies the text that should be displayed on the sign-in page.
relyingPartyId	<code>account-name-OAuth</code>	Specifies the client ID that is registered with the provider.
relyingPartySecret	No default value	Specifies the secret that is registered with the provider for the client ID.
scopes	The list should contain openid .	This option depends on what is defined for the <code>attributeMapping.userName</code> . The scope tells the provider what fields to get back from the provider. Depending on the provider, they might need to include a scope to get back the user name field.
showLinkText	On	Specifies that the link text should be shown on the sign-in page.
tokenUrl	No default value	The URL to obtain the token key endpoint. Specify either this value or the <code>tokenKey</code> , but not both.
type	By default, the value is oidc1.0 .	Specifies the protocol type. 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 <i>openid</i> and <code>respose_type=token</code> . For those providers, use type oauth2.0 .
tokenKey	No default value	Specifies the HMAC key or RSA public key that is used to sign tokens.
tokenKeyUrl	No default value	Specifies the URL to obtain the token key.

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
```

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.

See Also

[“Authentication: OpenID Connect Scenario \(Linux Full Deployment\)” on page 52](#)

Configure Identity Provider Discovery for OpenID Connect

Starting in the May 2019 release of SAS Viya 3.4, you can complete the following steps to enable identity provider discovery:

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 24](#).
- 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: Specify the *emailDomain* option that is defined in the **sas.logon.oauth.providers.external_oauth** definition.

- 5 Click **Save**.

Configure PAM (Linux)

Default Pluggable authentication module (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 SAML (Linux Full Deployment)

Overview

Before configuring Security Assertion Markup Language (SAML), 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 the Security Assertion Markup Language (SAML) typically follows this pattern:

- 1 [“Configure SAS Viya as a SAML Service Provider” on page 19](#)
- 2 [“Configure the SAML Identity Provider – Relying Party Configuration” on page 20](#)
- 3 [“Configure SAS Viya with Information about the SAML Identity Provider” on page 21](#)

Note: By default, SAS Viya allows only same-origin requests. Authentication requests from the SAML identity provider might be seen as cross-origin. Therefore, the origin of the SAML provider might need to be added. For details, see [“Configure Cross-Origin Resource Sharing” on page 25](#).

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 24](#).
- 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 7 SAML Configuration Fields and Descriptions

Field	Description
entityBaseURL	The external URL for the SAS Logon web application in SAS Viya (for example, https://hostname.example.com/SASLogon).
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.
serviceProviderCertificate	Paste a copy of the PEM-encoded (base64) certificate, which is used by the service provider.
serviceProviderKey	Paste a copy of the PEM-encoded (base64) key, which is used by the service provider.
serviceProviderKeyPassword	Provide the password for the service provider, or leave blank if there is no password.

Field	Description
setProxyParams	Important: This field should not be modified. The value should remain false .
signMetaData	Specifies whether the local service provider should sign the metadata.
signRequest	Specifies whether the local service provider should sign the SAML requests.
wantAssertionSigned	Specifies whether the assertions should be signed.
signatureAlgorithm	Specifies the algorithm for SAML signatures. Acceptable values are SHA1, SHA256, and SHA512. The default value is SHA256.
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.
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.
maxAuthenticationAge	Specifies the maximum time (in seconds) between users initial authentication with the identity provider (IdP) and processing of an authentication statement. The default value is 864000.

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
```

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 Identity Provider – 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 identity provider 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 identity provider.

Configure SAS Viya with Information about the SAML Identity Provider

1 Complete the following steps in SAS Environment Manager:

- a In the **Definitions** list, select **sas.logon.saml.providers.external_saml**.
- b In the top right corner of the window, click **New Configuration**.
- c In the New sas.logon.saml.providers.external_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 8 SAML External Provider Configuration Fields and Descriptions

Field	Description
addShadowUserOnLogin	Add a local shadow user upon successful authentication. If set to false, users must already exist in the database to log on.
assertionConsumerIndex	The index of the assertion consumer service to use from identity provider metadata. The value must be a positive integer. The default value is 0.
idpMetadata	The metadata XML content. This can be useful if manual changes need to be made to the identity provider metadata. Note: If you use the URL to the IDP's SAML Metadata, it must always be available or users will receive an error when they attempt to sign on. To avoid this, you may download the XML and paste it into the idpMetadata field.
linkText	The hyperlink to display on the sign-in page.
metadataTrustCheck	Specify whether to trust the identity provider certificate.
nameID	The field is populated with a default value. Verify with your information technology support group that the value is correct.
showSamlLoginLink	Determines whether a link should be displayed on the logon page for this identity provider.
skipSslValidation	Specifies whether to skip the TLS validation of the certificate.
emailDomain	Specifies a comma-separated list of email domains for users that can sign on with the SAML provider. It is used with IdP discovery and is optional.
authnContext	The comma-separated list of authentication contexts that are included in SAML requests to the IdP.

- d Click **Save**.

2 Edit the *SAS-Viya-configuration-directory/etc/sysconfig/sas-javaesntl/sas-java-services* file, and add the following line where the truststore options are set:

```
[[ -f $truststore ]] && export
java_global_option_truststore_password="-Djavax.net.ssl.trustStorePassword=changeit"
```

3 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.

Configure Identity Provider Discovery for SAML

Starting in the May 2019 release of SAS Viya 3.4, you can complete the following steps to enable identity provider discovery:

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 24](#).
- 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: Specify the *emailDomain* option that is defined in the **sas.logon.oauth.providers.external_oauth** definition.
- 5 Click **Save**.

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 24](#).
- 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 9 SAS 9.4 Configuration Fields and Descriptions

Field	Description
autoLink	Specifies whether to automatically open the link to SAS 9.4 when the logon page is displayed. Note: If the <i>autoLink</i> property is enabled, then the SAS Logon Manager in SAS Viya form 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.
enabled	Specifies whether to enable sign-ins using SAS 9.4 credentials.
linkText	Specifies the hyperlink to display on the sign-in page. Note: By default, the end user is presented with a link at the bottom of the standard SAS Logon Manager in SAS Viya form. The text of the link is controlled by the <i>linkText</i> property. This default behavior means that end users can choose to either use SAS 9.4 to authenticate or use the LDAP provider.
sas9LogonUrl	Specifies the URL of the SAS Logon Manager in SAS 9.4 (for example, https://SAS9_hostname/SASLogon).
showLinkText	Specifies whether to display the link text on the sign-in page.
single.signOn.enabled	Specifies whether to redirect to SAS 9.4 for single sign-on.
single.signOut.enabled	Specifies whether the local sign-out should also sign the user out of SAS 9.4.
viyaLogonUrl	Specifies the URL of the SAS Logon Manager in SAS Viya (for example, https://SASViya_hostname/SASLogon).

- 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
```

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


- 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 box.

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, specify `session.timeout`.
 - d In the **Value** field, specify 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 Logins

As a SAS administrator, you can disable logons 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 24](#).
- 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` is not supported.

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
```

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.

Obtain an Access Token Using Password Credentials

You can use the following commands to register a client ID and secret. You can also use the commands to obtain a token that can be used to call a SAS Viya API and to access SAS Viya credentials from SAS 9.4.

1 Register a new client ID and secret by completing the following steps:

Note: You must register a client ID once.

- a Obtain a token to register a new client ID and secret. For more information, see [“Obtain an OAuth Access Token to Register a New Client ID” in *Encryption in SAS Viya: Data in Motion*](#).

- b Use the token to register the new client ID and secret by running the following curl command:

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.

```
curl -X POST http://localhost/SASLogon/oauth/clients -H "Content-Type: application/json"
-H "Authorization: Bearer token-from-previous-step"
-d '{
    "client_id": "client-id",
    "client_secret": "client-secret",
    "scope": ["openid", "*"],
    "resource_ids": "none",
    "authorities": ["uaa.none"],
    "authorized_grant_types": ["password"]
}'
```

Note: The value for the **scope** parameter can be a list of scopes and groups that you **might** request when obtaining a token. You might also specify the wildcard "*" to request all scopes always. Ensure that you specify the list correctly. SAS Viya treats group memberships as scopes. Therefore, the list of scopes is the list of group memberships that you might request when obtaining a token. The "openid" is a special scope that represents authentication only and should always be included.

2 A token can be used until it expires. By default, this is 12 hours. To acquire a token, run the following curl command:

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

Note: The values for *client-id* and *client-secret* should be the same as the values that were specified in [Step 1b](#).

- 3 Retrieve the access token information from the results of the curl command in [Step 2](#). This access token is used to perform the following tasks:
 - a Call a SAS Viya API by passing the HTTP Authorization header as a Bearer token: `Authorization: Bearer access_token`.
 - b Assign the access token to the `SAS_VIYA_TOKEN` environment variable. Setting this environment variable enables you to access SAS Viya credentials from SAS 9.4. For example, when the `AUTHDOMAIN=` option is set on a `CAS` or `LIBNAME` statement, an attempt is first made to retrieve credentials from the SAS Viya Credentials service before searching the metadata. For more information, see [SAS_VIYA_TOKEN Environment Variable](#).

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](#).

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. By default, authentication is performed via a Lightweight Directory Access Protocol (LDAP) provider. Authentication support is also available for Kerberos, OAuth 2.0 with OpenID Connect, 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 and 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).

Authentication and SAS Viya Services

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

Table 10 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 LDAP provider.
Authorization service	Provides authorization information to other services.

Service Name	Description
Launcher Service	Provides the connection and authentication to the SAS Launcher Server. Resolves the credentials that are used when authenticating to the SAS Launcher Server.
SAS Cloud Analytic Services	Authenticates end users launching CAS sessions by way of the SAS Cloud Analytic Services controller.
SAS 9.4	Supports several mechanisms for coupling authentication with SAS 9.4.
SAS Studio 4.4	Leverages the SAS Object Spawner to authenticate users accessing SAS Studio 4.4.

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.

Starting in the May 2019 release of SAS Viya 3.4, 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 4](#).

Out-bound Authentication

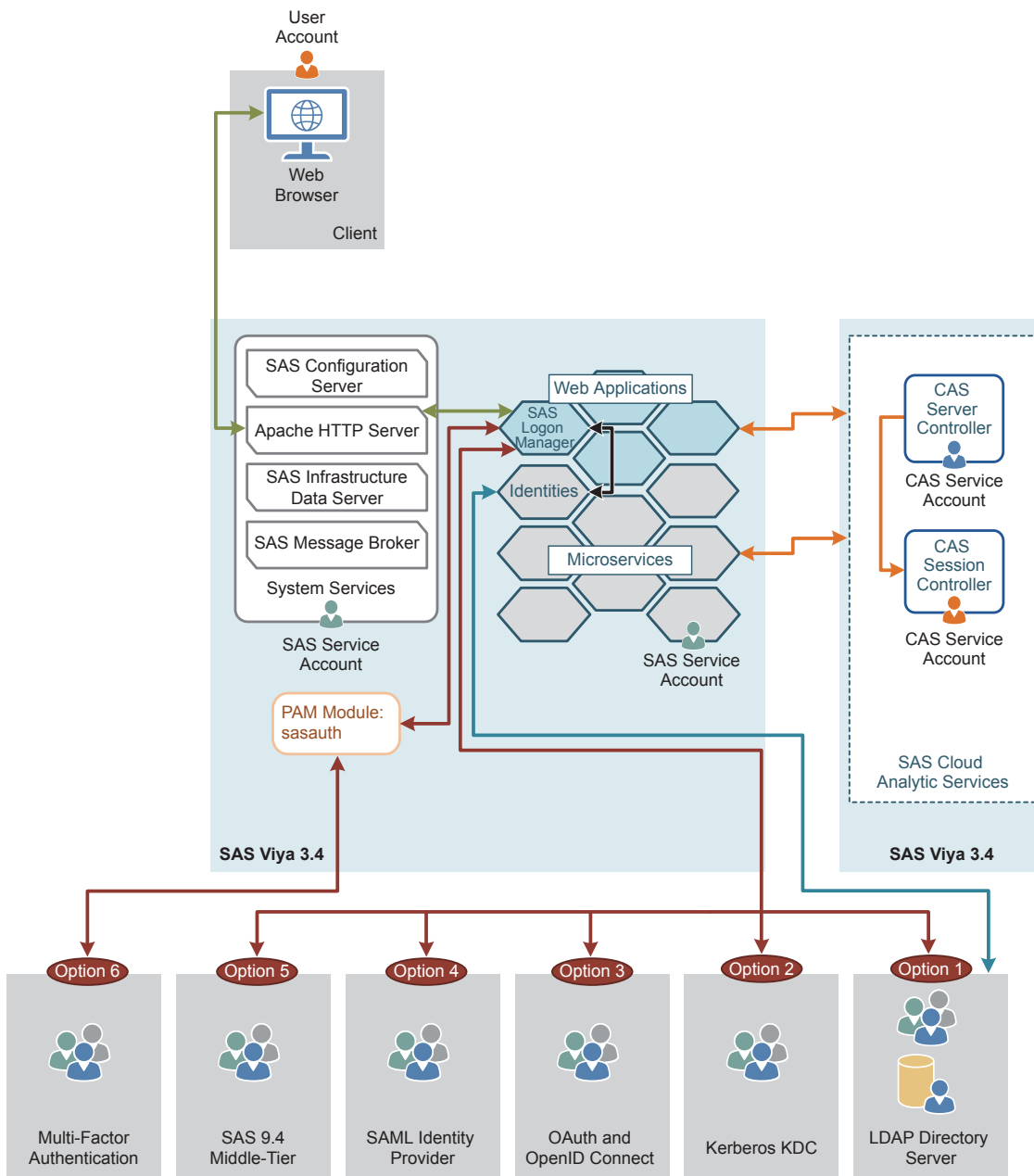
Out-bound authentication is the authentication of the SAS process to a downstream process. Out-bound authentication occurs after the end user is initially authenticated to SAS Logon Manager. Out-bound authentication occurs to SAS Cloud Analytic Services, SAS Compute Server (through SAS Launcher Service), 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 logon form and submits the entered credentials to LDAP. The identity service verifies users in LDAP. For more information, see [“LDAP Authentication \(Full Deployment\)”](#) on page 33.
- The second option is Kerberos. In this configuration, SAS Logon Manager uses SPNEGO to authenticate users against the Kerberos Key Distribution Center (KDC). The identity service verifies users in LDAP. For more information, see [“Kerberos Authentication \(Full Deployment\)”](#) on page 33.

Note: SAS Cloud Analytic Services 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 OpenID Connect. In this configuration, SAS Logon Manager uses OAuth 2.0 and OpenID Connect to authenticate users. The identity service verifies users in LDAP. For more information, see [“OAuth and OpenID Connect Authentication \(Linux Full Deployment\)”](#) on page 40.

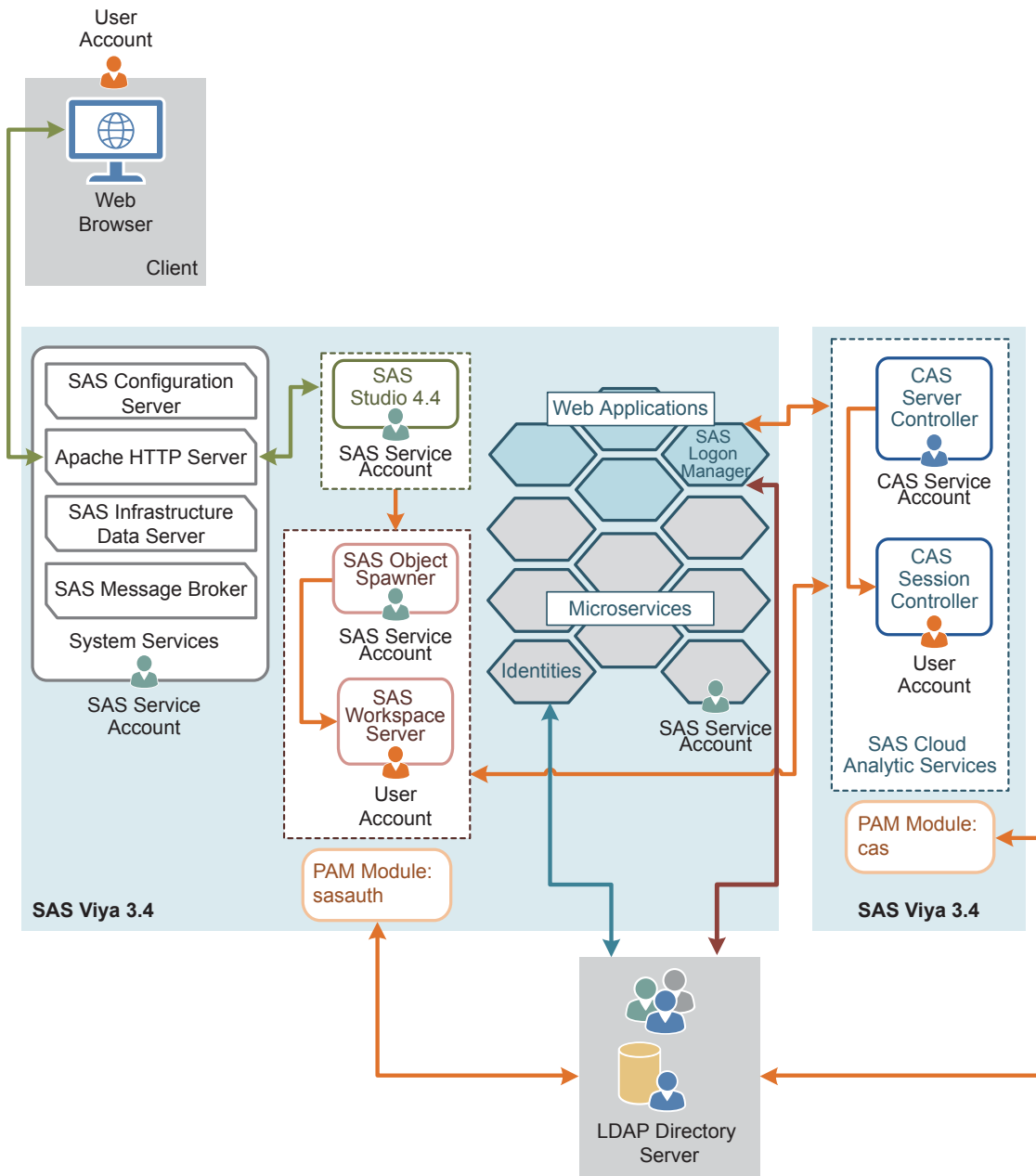
- The fourth option is Security Assertion Markup Language (SAML). In this configuration, SAS Logon Manager uses a SAML provider to authenticate users. The identity service verifies users in LDAP. For more information, see [“SAML Authentication \(Linux Full Deployment\)” on page 41](#).
- The fifth option is SAS 9.4. In this configuration, SAS Logon Manager supports single sign-on and single sign-off with SAS 9.4. The identity service verifies users in LDAP. For more information, see [“SAS 9.4 Authentication” on page 42](#).
- 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 identity service verifies users in LDAP. For more information, see [“PAM Authentication \(Linux\)” on page 44](#).

With all six options, the connection to SAS Cloud Analytic Services (CAS) environment 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 entered into SAS Studio via 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 be a local account on the host or, depending on the PAM configuration, an account in the LDAP provider. Once the user is authenticated, SAS Workspace Server is started. The PAM configuration file for SAS Studio is `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](#) or [AUTHINFO= CAS statement 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 `cas` and includes the password module.

The CAS controller uses the user ID and password to obtain an internal OAuth token from SAS Logon Manager. This requires the user ID and password to be valid in the LDAP provider that is configured for SAS Logon

Manager. Otherwise, CAS cannot obtain an OAuth token, and the session fails. Therefore, PAM for SAS Studio (sasauth), PAM for CAS (cas), and SAS Logon Manager should all use the same or equivalent LDAP providers. These three components should be sending the user ID and password that was entered into SAS Studio to the same provider. Otherwise, errors might be generated when trying to connect.

Programming Interfaces with Symmetric Multiprocessing CAS Server

In a symmetric multi-processing (SMP) environment, a CAS server consists of a controller and runs on a single machine. The following details the authentication process:

- 1 The end user connects to the SAS Studio 4.4 application and enters their user name and password in the logon form. SAS Studio is proxied by the Apache HTTP Server.
- 2 SAS Studio 4.4 passes the user name and password to SAS Object Spawner to start the SAS Workspace Server for the end user.
- 3 SAS Object Spawner uses the PAM configuration that is defined in `/etc/pam.d/sasauth` to validate the user name and password and launches SAS Workspace Server as the end user.
- 4 The end user enters code to start their 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 presenting the end user's user name and password.
- 6 SAS Logon Manager validates the user name and password against the defined LDAP Provider. SAS Logon also connects to the Identities service to obtain group information to include 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 sends the OAuth token back 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 one backup controller (optional). Each component runs on a separate machine. The authentication process for MPP SAS Cloud Analytic Services is essentially the same as for SMP CAS, with the following key differences:

- Initial communication between the CAS controller and CAS workers is via Secure Socket Shell (SSH).
- On-going 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*

Authentication Mechanisms

LDAP Authentication (Full Deployment)

Overview of LDAP

In SAS Viya, LDAP is used for 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. The Identities service always makes a direct connection to LDAP to obtain user and group information. By default, SAS Logon Manager authenticates users using a direct connection to the configured LDAP provider. To ensure that network connections are secure, the connection between the browser and the Apache HTTP Server can be secured with HTTPS. In addition, the connection between SAS Logon Manager and the LDAP provider can be secured with LDAPS.

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 11 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.

Term	Definition
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

In addition to using the LDAP provider to obtain user and group information, you can configure SAS Logon Manager for Kerberos authentication. This option replaces the option to use the default LDAP provider for authentication to SAS Logon Manager. Kerberos provides the user with single sign-on capabilities from the browser on their desktop. Single sign-on allows the user to access the SAS Viya visual interfaces without being prompted to enter their credentials.

For information, see [“Configure Kerberos \(Linux Full Deployment\)” on page 3](#).

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

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

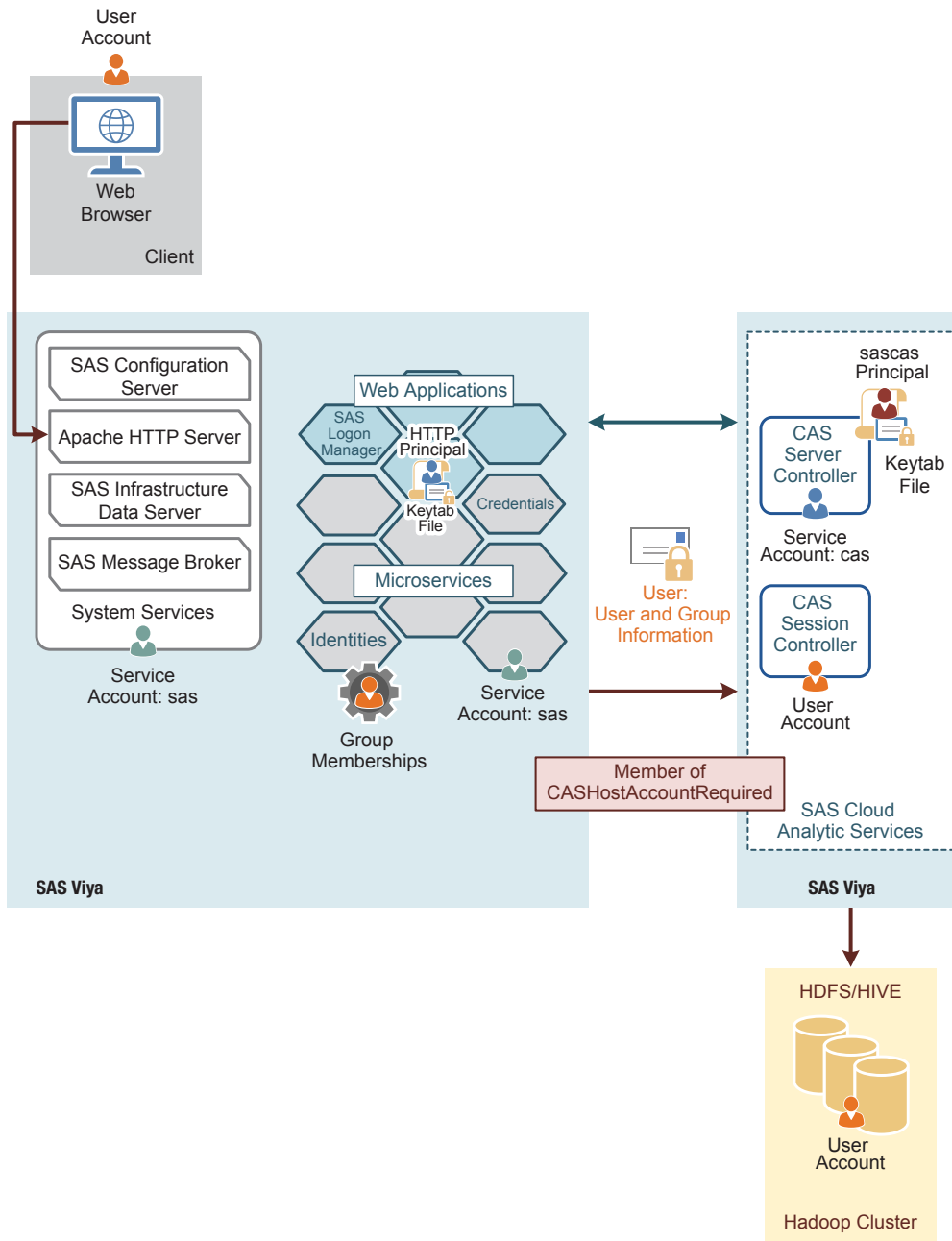
Table 12 *Kerberos Scenarios*

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 35 .	Kerberos (delegation)	End user	End user
SAS Viya visual interface See “Kerberos in SAS Viya Visual Interface (Outbound from CAS)” on page 37 .	Kerberos	Service account (cas)	Service account (sascas)

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 (Leveraging Stored Credentials)” on page 37.	Kerberos	End user	End user
SAS Viya programming interface See “Kerberos in SAS Viya Programming Interface with User Credentials” on page 37.	User ID and password	End user	End user
SAS 9.4 See “Kerberos in SAS 9.4 with User Credentials” on page 38.	User ID and password	End user	End user
SAS 9.4 See “Kerberos in SAS 9.4 with Delegation” on page 39.	Kerberos (delegation)	End user	End user
SAS 9.4 See “Kerberos in SAS 9.4 with One-Time Password” on page 39.	One-time password	Service account (cas)	Service account (sascas)

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 Kerberos delegation needs to take place. The user's delegated Kerberos ticket can then be used for user access to Kerberized services as himself or herself. For more information, see ["The CASHostAccountRequired Custom Group" in SAS Viya Administration: Identity Management](#).

Note: On Windows, user sessions are launched under the user identity. However, there are times when 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 SAS Cloud Analytic Services” on page 6](#)

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 OpenID Connect.

A registered principal for SAS Cloud Analytic Services (CAS) 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.

The following 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 OpenID Connect.

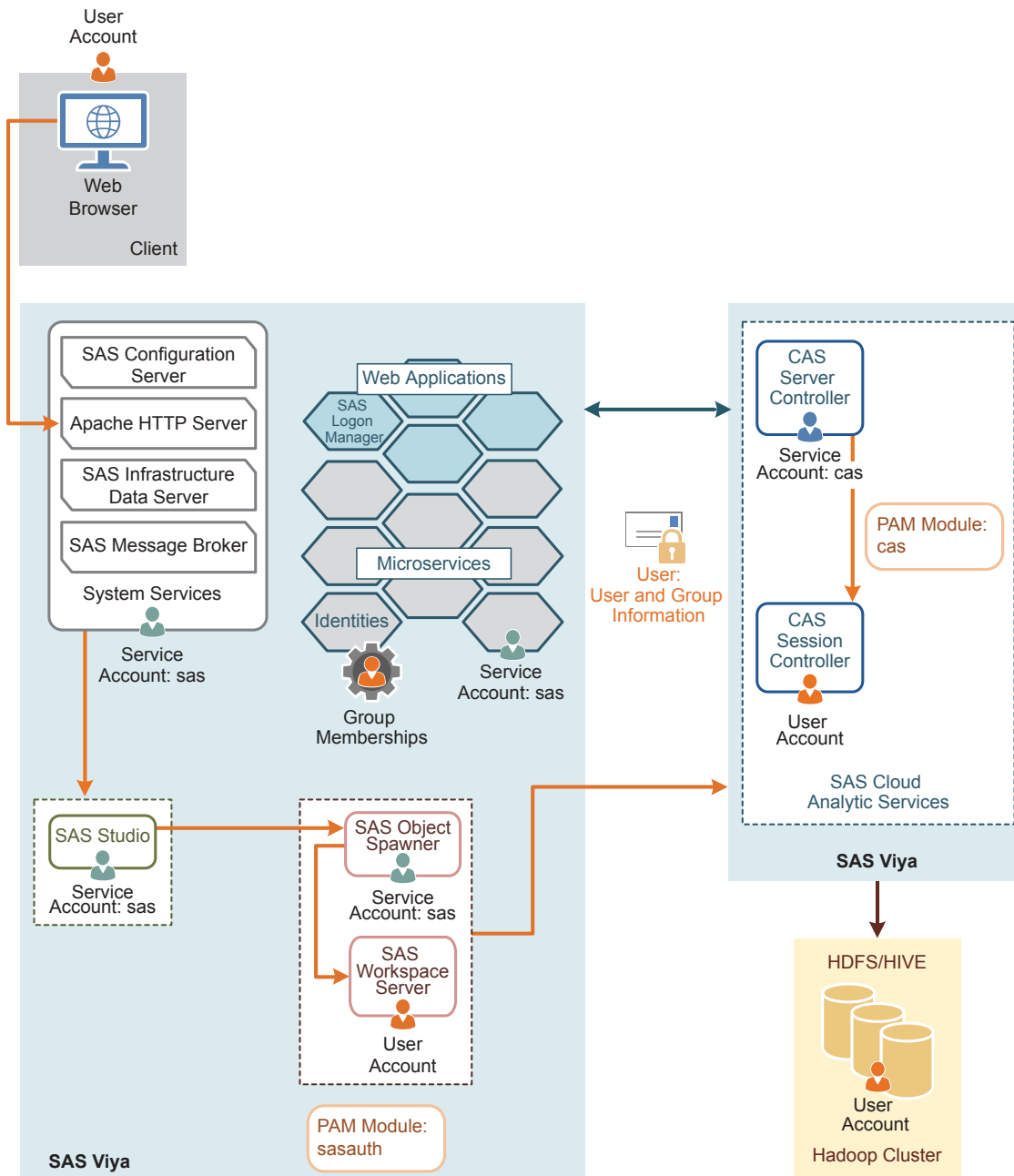
A credential must be stored for the end user. This can either be individually or from a group membership. A custom group must also be defined in SAS Environment Manager. The group can be named anything, but the ID must be `CASHostAccountRequired`. In addition, the operating system must generate the Kerberos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

The following 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 is used to generate an internal OAuth token from SAS Logon Manager running SAS Viya. The PAM stack is responsible for initializing the Kerberos credentials for the end user. 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 end 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 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

Starting in the May 2019 release of SAS Viya 3.4, 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. Also, if you apply the May 2019 upgrade to an existing SAS Viya 3.4 environment that is configured for Kerberos, you will automatically get fallback authentication support.

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 OpenID Connect) 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 logon form and any other authentication mechanisms that are configured.

Note: Google Chrome and Microsoft Internet Explorer attempt NTLM authentication before falling back to another authentication mechanism. A window might be displayed for the user to enter their credentials. If this happens, they can cancel out of the box to initiate fallback to the login form.

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

See Also

- [“Configure Kerberos \(Linux Full Deployment\)” on page 3](#)
- [“Authentication for Visual Interfaces” on page 28](#)

OAuth and OpenID Connect Authentication (Linux Full Deployment)

Overview of OAuth and OpenID Connect

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. OpenID Connect is an extension to OAuth 2.0, which provides authentication support.

Key Terms

Table 13 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 OpenID Connect provider can be internal to the customer's environment, or it can be an external provider, such as Google Authenticator or Facebook. When the OAuth 2.0 option is configured, this does not completely replace the default LDAP provider. Instead, when users access SAS Logon Manager, they are presented with a link to authenticate using OAuth 2.0 and the standard logon form using the LDAP provider. Users can select which to use. The user identity and group membership information is looked up in LDAP. OAuth 2.0 can provide single sign-on from the OAuth 2.0 provider. For example, when a user signs in to his or her Google account, the user can access the visual interfaces of SAS Viya without being prompted any further for credentials.

Starting in the May 2019 release of SAS Viya 3.4, fallback authentication is available when SAS Logon Manager is configured for Kerberos authentication. If Kerberos authentication fails, OAuth and OpenID Connect authentication can be attempted. For more information, see [“Fallback Authentication” on page 40](#).

Also, in the May 2019 release of SAS Viya 3.4 is the simplified configuration of identity provider discovery for logins. In prior releases, custom code and JavaScript were needed to configure this feature. Now, the email address of the end user is used for redirection to the OpenID Connect Identity Provider. For more information, see [“Configure Identity Provider Discovery for OpenID Connect” on page 18](#).

See Also

[“Configure OpenID Connect” on page 16](#)

SAML Authentication (Linux Full Deployment)

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

Overview of SAML

The SAML standard defines a framework for exchanging security information about users between an identity provider 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 14 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.

Term	Definition
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 identity provider. This identity provider 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.com can be used. SAML does not completely replace the default LDAP provider. End-users accessing SAS Logon Manager can choose 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 identity provider 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 identity provider to validate the credentials, instead of providing credentials to the service provider. Therefore, users do not have to provide their credentials directly to anyone but the identity provider.

Starting in the May 2019 release of SAS Viya 3.4, 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 40](#).
- Simplified configuration of identity provider discovery for logins. In prior releases, custom code and JavaScript were needed to configure this feature. Now, the email address of the end user is used for redirection to the SAML Identity Provider. For more information, see [“Configure Identity Provider Discovery for SAML” on page 22](#).
- Identity provider 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 identity provider with the authentication request and get back in the response. The parameter can also be specified by the identity provider 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 [“Identity Provider Initiated Logon for SAML ” on page 46](#).

See Also

[“Configure SAML \(Linux Full Deployment\)” on page 19](#).

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 the SAS Logon Manager in SAS 9.4. None of the authentication

occurs with the 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 logon form, which contains a link to perform SAS 9.4 authentication and the form to do 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 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 custom and LDAP 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 impacts only the SAS Viya 3.4 visual interfaces using the SAS Logon Manager. An LDAP provider is still required by the Identities service. For authentication to SAS Logon Manager in SAS Viya that is not through a browser, the credentials are first passed to the 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 App (previously called SAS Mobile BI) is still authenticated against the SAS Logon Manager in SAS Viya first. SAS Studio 4.4 is not impacted 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. You need to import the certificate of the 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 correctly search for it in the SAS Viya LDAP provider. You can log 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 accounts do not work with SAS Viya.

Also, 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 is 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 is 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

Pluggable authentication module (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.

Starting in the May 2019 release of SAS Viya 3.4, 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 SAS Cloud Analytic Services (CAS) server, *service* is *cas*. For SAS Studio, *service* is *sasauth*.

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.

Starting in the May 2019 release of SAS Viya 3.4, 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 40](#).

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. SAS Studio authenticates the connection to CAS by using your user credentials. When password information is not available, an attempt is made to find an authinfo file (.authinfo is the default filename 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 user credentials are used to authenticate your connection to CAS. SAS Studio does not use the authinfo file for authentication.

Typically, the authinfo file resides 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, 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 56](#)
- [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
- what 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 log on to a VPN client

See Also

[“Configure PAM \(Linux\)” on page 18.](#)

Additional Authentication Topics

Identity Provider Initiated Logon for SAML

Starting in the May 2019 release of SAS Viya 3.4, you can update the SAML identity provider 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 identity providers are configured. Instead of a logon form 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 identity provider, 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 21](#).

In the identity provider-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 56](#)

- [SAS/CONNECT 9.4 User's Guide](#)
- [“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 the following SSO products:

- [Kerberos on page 33](#)
- [“SAML Authentication \(Linux Full Deployment\)” on page 41](#)
- [OAuth 2.0 and OpenID Connect on page 40](#)

Dual Authentication

Linux

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

- If PAM is configured to use local accounts and those users also log on to the visual components, then those local accounts must match the LDAP server 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



Guest access is an optional feature that provides anonymous Read-Only access to a subset of resources and functionality in participating applications. Prior to the August 2019 release for SAS Viya 3.4, guest access is supported for viewing reports in SAS Report Viewer and SAS Visual Analytics App. Starting in the August 2019 release for SAS Viya 3.4, guest access is no longer supported for viewing reports in SAS Report Viewer.

Reports can be viewed only 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 [“Create at Least One Profile” in SAS Viya Administration: Using the Command-Line Interfaces](#).
 - b Modify the authorization rules.

- For a new SAS Viya 3.4 installation, run the following command:

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

- For an upgrade from SAS Viya 3.3 to SAS Viya 3.4 in which guest access was not previously configured, run the following command:

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

- For an upgrade from SAS Viya 3.3 to SAS Viya 3.4 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 3.3 to SAS Viya 3.4, run the following command:

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

- c Modify the direct access controls for the predefined caslibs on the server, using the controls that are defined in the specified source file, run the following command:

Note: 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
```

For more information about the controls file, see [“Enable Guest Access” in SAS Viya Administration: Using the Command-Line Interfaces](#).

- 3 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
```

- 4 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 [“Content Management: How To” in SAS Viya Administration: Content Management](#).

Connect as Guest Users

Prior to the August 2019 release for SAS Viya 3.4, once guest access is enabled, guest users can view reports using SAS Report Viewer and SAS Visual Analytics App. SAS Report Viewer displays a guest logon button. SAS Visual Analytics App displays a guest logon button when a mobile connection is established.

Starting in the August 2019 release for SAS Viya 3.4, 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 logon button. SAS Visual Analytics App displays a guest logon 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 logon page and automatically connect the user as guest. If guest access is disabled, a logon page is displayed, where users can choose to connect as a guest or log on 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.

Authentication: OpenID Connect Scenario (Linux Full Deployment)

In the following tasks, OpenID Connect 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 OpenID Connect, complete the following sections:

Configure OpenID Connect Provider Properties for IBM Security Access Manager

- 1 From SAS Environment Manager, navigate to the SAS Logon Manager configuration definitions. For more information, see [“Edit Authentication Configuration Instances” on page 24](#).
- 2 In the **Definitions** list, select **sas.logon.oauth.providers.external_oauth**.
- 3 In the top right corner of the window, click **New Configuration**.
- 4 In the New **sas.logon.oauth.providers.external_oauth** 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 15 OpenID Connect Configuration Fields and Descriptions

Configuration Field	Description
<code>addShadowUserOnLogin</code>	A local shadow user should be added once authentication is successful.
<code>attributeMapping.user_name</code>	The attribute claim to use as the user name. For ISAM, use sub .
<code>authUrl</code>	The URL to the authorization endpoint (for example, <code>https://hostname.example.com/isam/oidc/endpoint/amapp-runtime-ISAMOP/authorize</code>).
<code>emailDomain</code>	Specifies a comma separated list of email domains for users that can sign on with the OpenID Connect provider. It is used with identity provider discovery and is optional.
<code>issuer</code>	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>).

Configuration Field	Description
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]).
relyingPartyId	The client ID that is registered with the provider.
relyingPartySecret	The secret that is registered with the provider for the client ID.
scopes	The comma-delimited list of scopes for the authorization request. The list should contain openid . 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.
tokenUrl	The URL to the token endpoint.
type	The protocol type. By default, the value is oidc1.0 . Note: SAS Viya requires an id_token in the authorization response from the provider. However, some providers return an id_token when the scope in the authorization request is openid and response_type=token . For those providers, use type oauth2.0 .
tokenKey	Specifies the HMAC key or RSA public key that is used to sign tokens.
tokenKeyUrl	Specifies the URL to obtain the token key.

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
```

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 OpenID Connect Provider in IBM Security Access Manager

For basic steps to configure OpenID Connect in ISAM 9.0.3.1, see: [IBM SECURITY ACCESS MANAGER Federation Cookbook 9.0.0.0 – 9.0.3.0 Installation, SAML 2.0, OpenID Connect, and Secure Token Service](#). To configure OpenID Connect 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 OpenID Connect Provider and its partner.
An OpenID Connect Provider on ISAM is a federation. First create a federation that represents the OpenID Connect Provider. Then, create a partner that represents the SAS Viya application under it.
- 3 Create a federation for OpenID Connect Provider. The following table shows values that you should provide while creating the new federation.

Table 16 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 OpenID Connect Provider Partner for SAS Viya (SASLogon). The following table shows values that you should provide while creating the new partner.

Table 17 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

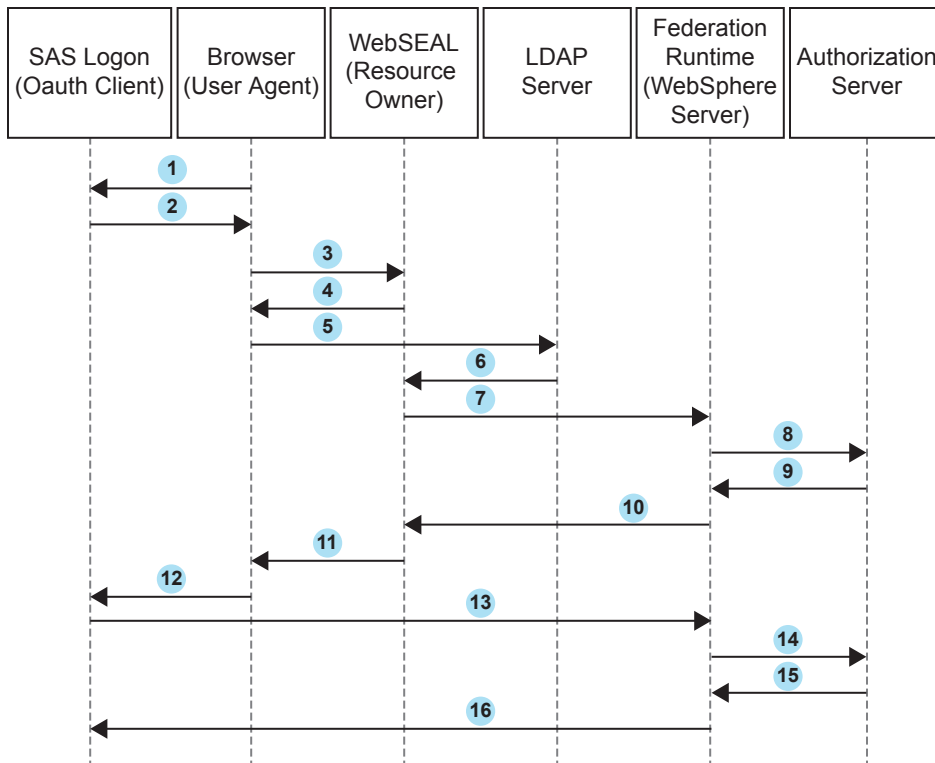
Field Name	Value
Scope	openid

5 Test your configuration.

OpenID Connect and IBM Security Access Manager

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 logon 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 their authentication information.
- 5 The client browser sends the authentication information to the Lightweight Directory Access Protocol (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: Reference

CAS Environment Variables for Clients

The environment variables in this section are set on the client and affect how the client authenticates 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
Operating environment	Environment variables on Linux are case-sensitive.
See	Authinfo File Authentication
Examples	<p>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.</p> <pre>export CAS_AUTH_METHOD=authinfo</pre> <pre>set CAS_AUTH_METHOD=kerberos</pre>

CAS Environment Variables for Administrators

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

env.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 `env.CASUSERLOWERCASE` whenever specifying `env.CASUSERIGNORECASE`, unless instructed otherwise by SAS Technical Support.

The typical scenario for declaring `env.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	casconfig_usermods.lua file
Category	Administration
Default	off
Restrictions	Applies to Linux only. <code>env.CASUSERIGNORECASE</code> is case sensitive.
Requirement	Use with <code>env.CASUSERLOWERCASE</code> .
Note	To turn off <code>env.CASUSERIGNORECASE</code> , remove its definition.
Example	In this example, <code>env.CASUSERIGNORECASE</code> is in effect: <code>env.CASUSERIGNORECASE= 'on '</code>

`env.CASUSERLOWERCASE='ON'`

when in effect (specified using any value), causes the CAS server to convert user names to lower letter casings during group lookup. `env.CASUSERLOWERCASE` is typically used in conjunction with `env.CASUSERIGNORECASE`.

The typical scenario for declaring `env.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	casconfig_usermods.lua file
Category	Administration
Default	off
Restrictions	Applies to Linux only. <code>env.CASUSERLOWERCASE</code> is case sensitive.
Requirement	Use with <code>env.CASUSERIGNORECASE</code> .
Note	To turn off <code>env.CASUSERLOWERCASE</code> , remove its definition.
Example	In this example, <code>env.CASUSERLOWERCASE</code> is in effect: <code>env.CASUSERLOWERCASE= 'on '</code>

Authentication: Troubleshooting

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 configuring Kerberos 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:

```
/opt/sas/viya/home/bin/sas-bootstrap-config --token-file  
$consul-token kv write --force config/SASLogon/spring/profiles.active ldap,postgresql
```

Note: The previous command must be on one line. It is shown on more than one line for display purposes only.

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 [“env.CASUSERIGNORECASE='ON'” on page 56](#).