Installing Apache Knox

Date of Publish: 2019-08-26

https://docs.hortonworks.com

Contents

Installing Knox Using Ambari Overview

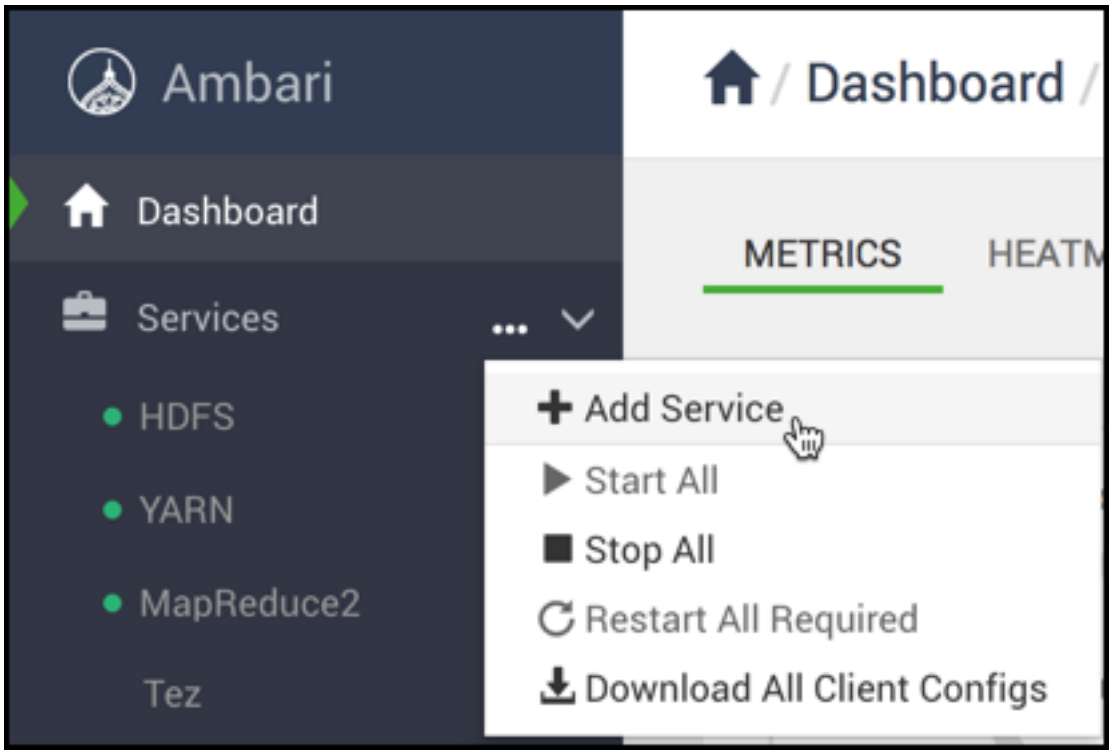
Apache Knox can be installed either manually using the Hortonworks Data Platform (HDP) or the Ambari User Interface (UI). The Knox service option will be made available through the Add Service wizard after the HDP cluster is installed using the installation wizard.

Install Knox

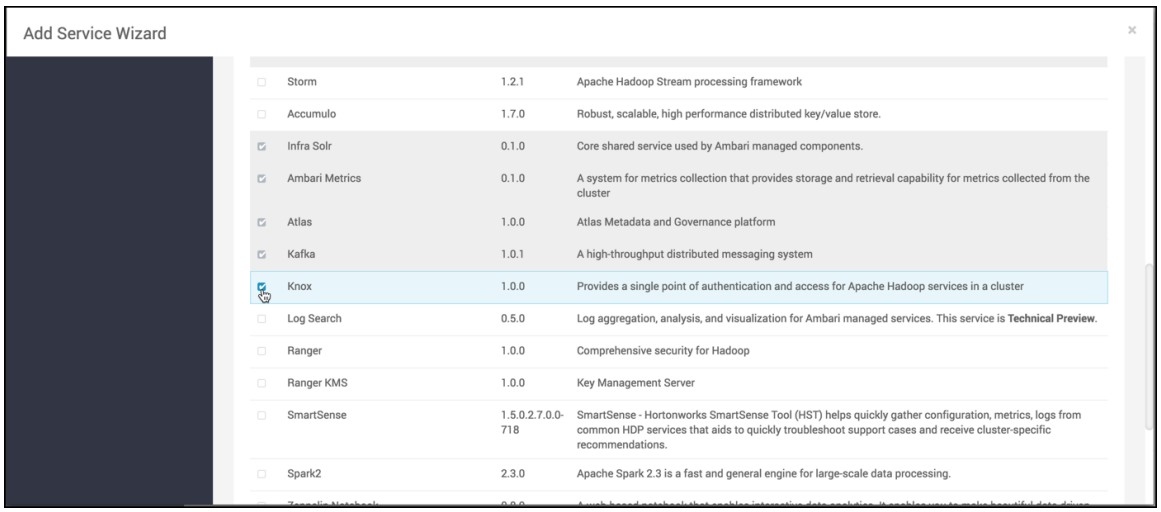
How to install Apache Knox using Ambari.

Procedure

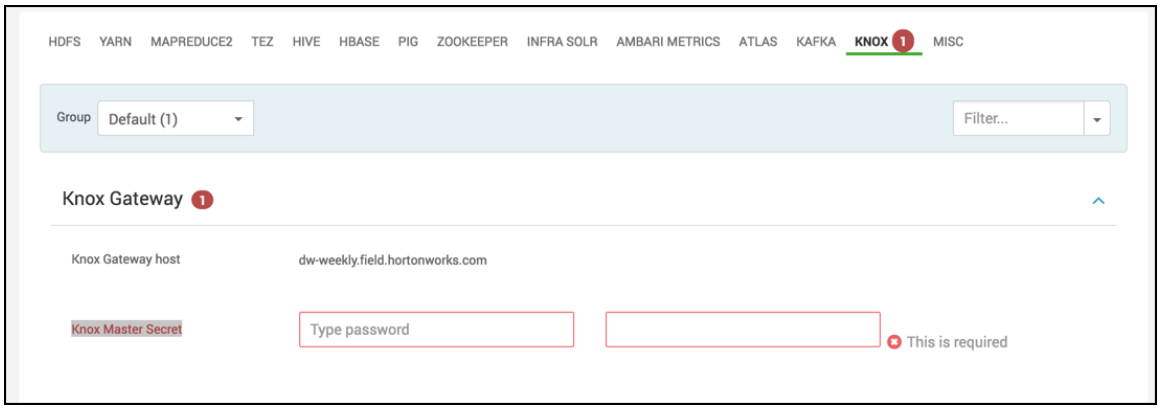
1. Log into your Ambari cluster with your designated user credentials.
   The main Ambari Dashboard page will be displayed.
2. In the left navigation menu, click Actions, then select Add Service.



3. On the Choose Services page, select Knox, then click Next.



4. You are prompted to Assign Masters. Make a note of the Knox Gateway host for use in subsequent installation steps. Click **Next**.
5. On the **Knox** tab, enter the **Knox Master Secret**, e.g. Password1!, and click **Next**.



6. Click **Deploy**.
7. After the install succeeds, click through the rest of the screens.

Related Tasks

Set Up Knox Proxy

Enable Knox SSO Using the Ambari CLI

Configure Knox SSO for HDFS, Oozie, MapReduce2, Zeppelin, or YARN

Set Up Knox Proxy

As of HDP 3.0, Knox Proxy is configured via the Knox Admin UI. To set up proxy, you will first define the provider configurations and descriptors, and the topologies will be automatically generated based on those settings.

About this task

The same topologies that were manageable in Ambari previously, still are. Within the Knox Admin UI, the topologies that are managed by Ambari should be read-only. Within an Ambari managed cluster, the Knox Admin UI is to be used for creating additional topologies. When a Knox instance is not managed by Ambari, all topology management will be done via the Knox Admin UI.

The following steps show the basic workflow for how to set up Knox Proxy. It involves defining provider configurations and descriptors, which are used to generate your topologies, which can define proxy (among other

things). For examples of how to set up proxy for a specific service, see “Configuring Proxy with Apache Knox”. It is recommended that you use the dynamic topology file generation in the Knox Admin UI; these steps utilize that workflow. You can also manually set up Knox Proxy by manually configuring individual topology files.

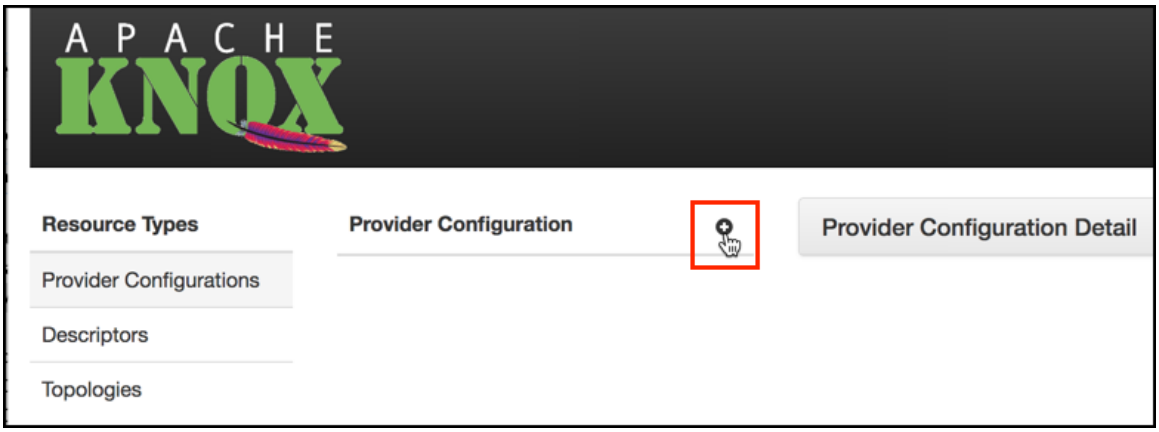
Before you begin

- Ambari is installed.
- The Demo LDAP server is running: **Ambari** > **Knox** > **Actions** > **Start Demo LDAP**.
- If you are proxying to services outside of the Knox host domain or redirecting to services for SSO that are in another domain, your whitelist is explicitly configured to accommodate that: **Ambari** > **Knox** > **Configs** > **Advanced knoxsso-topology**, e.g.

```
<param>
    <name>knoxsso.redirect.whitelist.regex</name>
    <value>^https?:\/\/(.*\.field\.hortonworks\.com|localhost|
127\.0\.0\.1|0:0:0:0:0:0:0:1|::1):[0-9].*$</value
</param>
```

Procedure

1. Navigate from Ambari to the Knox Admin UI: **Ambari** > **Knox** > **Quick Links** > **Knox Admin UI**.
   The Knox Admin UI opens, e.g. https://dw-weekly.field.hortonworks.com:8443/gateway/manager/admin-ui.
2. Login to the Admin UI.
   If you have not yet changed the credentials, the default credentials are admin/admin-password.
3. Create a Provider Configuration:
   a) From the Admin UI homepage, click **Provider Configurations** > +.



   The **Create a New Provider Configuration** wizard opens.
   b) Name the provider configuration: for example, hdp_ui_provider.
   c) Add an Authentication provider:
      1. Click **Add Provider**.
      2. Select **Authentication** and click **Next**.
      3. Choose your Authentication Provider Type: **LDAP, PAM, Kerberos, SSO (HeaderPreAuth), SSO Cookie (SSOCookieProvider), JSON Web Tokens (JWT), CAS, OAuth, SAML, OpenID Connect, Anonymous**.

         Note: OAuth, OpenID Connect, and CAS are community supported, they are not officially supported by Hortonworks.
      4. Complete the required fields and click **OK**.
   d) Add an Authorization provider:
      1. Click **Add Provider**.

2. Select **Authorization** and click **Next**.
3. Click **Access Control Lists**.
4. Fill out the required fields and click **OK**.

e) Add an Identity Assertion provider:

1. Click **Add Provider**.
2. Select **Identity Assertion** and click **Next**.
3. Choose a Identity Assertion Provider Type: **Default, Concatenation, SwitchCase, Regular Expression, Hadoop Group Lookup (LDAP)**.

   Recommended: Default.
4. Fill out the required fields and click **OK**.

f) Add an HA provider:

1. Click **Add Provider**.
2. Select **HA** and click **Next**.
3. Select **Add Service** and click **Next**.
4. Fill out the required fields and click **OK**.

4. Define Descriptors for the topology to auto-discover services from Ambari.

a) Create a new descriptor. From the Admin UI homepage, click **Descriptors** > **+**.

b) Name the descriptor.

c) Beside the Provider Configuration field, click the **edit** button and select the Provider Configuration you created before.

d) Add Services (e.g., JOBTRACKER, HIVE, HDFSUI, STORM) by clicking the checkbox beside the service.

If the service you are looking for is not listed, you can add it later by editing the configuration (the plus icon next to services will present a text box.)

e) Add Discovery details:

Field	Example value
Address	http://dw-weekly.field.hortonworks.com:8080
Cluster	dwweekly
Username	admin
Password alias	ambari-discovery-password

f) Click **OK**.

What to do next

Verify the topology was generated correctly. You can review the XML topology file for accuracy from **Admin UI homepage** > **Topologies** > **<topology name, e.g. devcluster>**.

Related Tasks

Install Knox

Enable Knox SSO Using the Ambari CLI

Configure Knox SSO for HDFS, Oozie, MapReduce2, Zeppelin, or YARN

Example: Configure Knox Gateway for YARN UI

This example shows you how to set up a new custom Knox topology for YARN UI and installing services for YARN.

Setting up Topology File

1. Login to Ambari and access Knox service page.

   Knox Admin UI link could be found on the right pane of the Ambari's Knox page.

   Once this link is clicked, user will be asked to provide a username and password. This will be based on the ldap configured for the manager.

2. Accessing Knox admin UI page for topology creation

   Once admin lands in to the Knox admin UI, there are fundamentally three steps more to create a topology of desired use case.

   a. Create a custom provider configuration
   b. Define Descriptors for the topology to auto-discover services from Ambari
   c. Save and verify the topology which is created

   Next steps will cover topology creation in detail.
3. Creating a custom Provider Configuration

   Admin can click on the “Provider Configurations” in left panel to list all available providers. Click on the “+” button on the right side to create a new provider.

   Admin can select all the providers which are needed for defining “hdp_ui_provider”

   - Authentication (Anonymous)
   - Authorization (AclsAuthz/Access Control Lists)
   - HAProvider (Default)
   - Identity-assertion (Default)

   These 4 providers could be added by selecting each and giving values from the auto populated options. Detailed steps are given below.

   a. Add Authentication>Anonymous.
   b. Add Authorization>Access Control Lists.
   c. Add HAProvider>Default.
   d. Add Identity-Assertion>Default.
   e. Save the provider by clicking on save button at right bottom.
4. Defining Descriptors for topology: Click on “+” button near to Descriptor to define a new custom descriptor.

   a. Add all details for a descriptor:

      - Define a name for the descriptor
      - Select YARNUI from the below list
      - Configure Ambari address in “Discovery - Address”
      - Configure Ambari cluster name in “Discovery - Cluster”
      - Provide Ambari user name in “Discovery - Username”
      - “Discovery Password Alias” could be left as it is as below manual step to be ran on knox machine to avoid configuring password.

   b. Creating password alias, e.g.,

      ```
      [root@ctr-e138-1518143905142-240189-01-046340 services]# /usr/$REPO/
      $VERSION/knox/bin/knoxcli.sh create-alias ambari.discovery.password
      Enter password:
      Enter password again:
      ambari.discovery.password has been successfully created.
      ```

   c. Select provider configuration as “hdp_ui_provider”.
   d. Press “Ok” to save the details.
   e. Select “hdp_ui” descriptor to add “YARNUIV2” service.

      Admin can add custom services which are see on the right pane under “Descriptor Detail”.

      Not all services listed are officially supported. See “Knox- Supported Services” for details on which services are supported.
5. Verify topology:

   Topologies>Select one topology: This is read-only pane where all configuration which are done for “hdp_ui” could be verified.

Changing QuickLinks for YARN UIs

Admin need to paste below quicklink.json file in Ambari server machine to ensure that YARN UIs quick links are accessible only via proxy.

1. Quick Link template

```
{
  "name": "default",
  "description": "default quick links configuration",
  "configuration": {
    "protocol":
    {
      "type": "HTTPS_ONLY"
    },
    "links": [
      {
        "name": "resourcemanager_ui",
        "label": "ResourceManager UI",
        "requires_user_name": "false",
        "component_name": "KNOX_GATEWAY",
        "url": "%@://%@:%@/gateway/hdp_ui/yarnuiv2/",
        "port": {
          "https_property": "gateway.port",
          "https_default_port": "8443",
          "regex": "^(\\d+)$",
          "site": "gateway-site"
        }
      },
      {
        "name": "resourcemanager_logs",
        "label": "ResourceManager logs",
        "requires_user_name": "false",
        "component_name": "KNOX_GATEWAY",
        "url": "%@://%@:%@/gateway/hdp_ui/yarn/logs",
        "port": {
          "https_property": "gateway.port",
          "https_default_port": "8443",
          "regex": "^(\\d+)$",
          "site": "gateway-site"
        }
      },
      {
        "name": "resourcemanager_jmx",
        "label":"ResourceManager JMX",
        "requires_user_name": "false",
        "component_name": "KNOX_GATEWAY",
        "url":"%@://%@:%@/gateway/hdp_ui/yarn/jmx",
        "port": {
          "https_property": "gateway.port",
          "https_default_port": "8443",
          "regex": "^(\\d+)$",
          "site": "gateway-site"
        }
      },
      {
        "name": "thread_stacks",
        "label":"Thread Stacks",
        "requires_user_name": "false",
        "component_name": "KNOX_GATEWAY",
        "url":"%@://%@:%@/gateway/hdp_ui/yarn/stacks",
        "port": {
          "https_property": "gateway.port",
          "https_default_port": "8443",
```

```
          "regex": "^(\\d+)$",
          "site": "gateway-site"
        }
      }
    ]
  }
}
```

2. Place quicklinks.json in Ambari: In ambari-server host, at following path, place the quicklink file:

```
/var/lib/ambari-server/resources/stacks/$REPO/$VERSION/services/YARN/
quicklinks/quicklinks.json
```

   Please ensure that existing quicklinks.json in replaced with the attached json file from this document.
3. Restart Ambari: ambari-server restart
4. Verify QuickLinks.

   Post these steps, YARN Quick links will be accessible only via knox proxy.

Example: Configure Knox Gateway for LDAP

This example shows you how to set up the Knox Gateway with ShiroProvider, which involves configuring a provider for LDAP.

Context

LDAP authentication is configured by adding a "ShiroProvider" authentication provider to the cluster's topology file. When enabled, the Knox Gateway uses Apache Shiro (org.apache.shiro.realm.ldap.JndiLdapRealm) to authenticate users against the configured LDAP store.

Setting up Topology File

1. Login to Ambari and access Knox service page.

   Knox Admin UI link could be found on the right pane of the Ambari's Knox page.

   Once this link is clicked, user will be asked to provide a username and password. This will be based on the ldap configured for the manager.
2. Accessing Knox admin UI page for topology creation

   Once admin lands in to the Knox admin UI, there are fundamentally three steps more to create a topology of desired use case.

   a. Create a custom provider configuration
   b. Define Descriptors for the topology to auto-discover services from Ambari
   c. Save and verify the topology which is created

   Next steps will cover topology creation in detail.
3. Creating a custom Provider Configuration

   Admin can click on the "Provider Configurations" in left panel to list all available providers. Click on the "+" button on the right side to create a new provider.

   Admin can select all the providers which are needed for defining "hdp_ui_provider"

   - Authentication (LDAP)
   - Authorization (AclsAuthz/Access Control Lists)
   - HAProvider (Default)
   - Identity-assertion (Default)

   These 4 providers could be added by selecting each and giving values from the auto populated options. Detailed steps are given below.

a. Add Authentication>LDAP.
b. Add Authorization>Access Control Lists.
c. Add HAProvider>Default.
d. Add Identity-Assertion>Default.
e. Save the provider by clicking on save button at right bottom.

4. Defining Descriptors for topology: Click on "+" button near to Descriptor to define a new custom descriptor.

   a. Add all details for a descriptor:

   - Define a name for the descriptor
   - Select $Services from the below list
   - Configure Ambari address in "Discovery - Address"
   - Configure Ambari cluster name in "Discovery - Cluster"
   - Provide Ambari user name in "Discovery - Username"
   - "Discovery Password Alias" could be left as it is as below manual step to be ran on knox machine to avoid configuring password.

   b. Creating password alias, e.g.,

   ```
   [root@ctr-e138-1518143905142-240189-01-046340 services]# /usr/$REPO/
   $VERSION/knox/bin/knoxcli.sh create-alias ambari.discovery.password
   Enter password:
   Enter password again:
   ambari.discovery.password has been successfully created.
   ```

   c. Select provider configuration as "hdp_ui_provider".
   d. Press "Ok" to save the details.
   e. Select "hdp_ui" descriptor to add "$SERVICES".

   Admin can add custom services which are see on the right pane under "Descriptor Detail".

   Not all services listed are officially supported. See "Knox- Supported Services" for details on which services are supported.

5. Verify topology:

   Topologies>Select one topology: This is read-only pane where all configuration which are done for "hdp_ui" could be verified.

Changing QuickLinks for $SERVICE UIs

Admin need to paste below quicklink.json file in Ambari server machine to ensure that $SERVICE UIs quick links are accessible only via proxy.

1. Quick Link template

```
{
  "name": "default",
  "description": "default quick links configuration",
  "configuration": {
    "protocol":
    {
      "type": "HTTPS_ONLY"
    },
    "links": [
      {
        "name": "resourcemanager_ui",
        "label": "ResourceManager UI",
        "requires_user_name": "false",
        "component_name": "KNOX_GATEWAY",
        "url": "%@://%@:%@/gateway/hdp_ui/$SERVICE/",
        "port": {
```

```
            "https_property": "gateway.port",
            "https_default_port": "8443",
            "regex": "^(\\d+)$",
            "site": "gateway-site"
          }
        },
        {
          "name": "resourcemanager_logs",
          "label": "ResourceManager logs",
          "requires_user_name": "false",
          "component_name": "KNOX_GATEWAY",
          "url": "%@://%@:%@/gateway/hdp_ui/$service/logs",
          "port": {
            "https_property": "gateway.port",
            "https_default_port": "8443",
            "regex": "^(\\d+)$",
            "site": "gateway-site"
          }
        },
        {
          "name": "resourcemanager_jmx",
          "label":"ResourceManager JMX",
          "requires_user_name": "false",
          "component_name": "KNOX_GATEWAY",
          "url":"%@://%@:%@/gateway/hdp_ui/$service/jmx",
          "port": {
            "https_property": "gateway.port",
            "https_default_port": "8443",
            "regex": "^(\\d+)$",
            "site": "gateway-site"
          }
        },
        {
          "name": "thread_stacks",
          "label":"Thread Stacks",
          "requires_user_name": "false",
          "component_name": "KNOX_GATEWAY",
          "url":"%@://%@:%@/gateway/hdp_ui/$service/stacks",
          "port": {
            "https_property": "gateway.port",
            "https_default_port": "8443",
            "regex": "^(\\d+)$",
            "site": "gateway-site"
          }
        }
      ]
    }
}
```

2. Place quicklinks.json in Ambari: In ambari-server host, at following path, place the quicklink file:

```
/var/lib/ambari-server/resources/stacks/$REPO/$VERSION/services/$SERVICE/
quicklinks/quicklinks.json
```

   Please ensure that existing quicklinks.json in replaced with the attached json file from this document.
3. Restart Ambari: ambari-server restart
4. Verify QuickLinks.

   Post these steps, $SERVICE Quick links will be accessible only via knox proxy.

Enable Knox SSO Using the Ambari CLI

To enable Knox SSO (Single Sign-on) for your cluster, you must begin with the Ambari CLI wizard. It prompts you for SSO settings and propagates them across your cluster. The wizard then configures SSO for Atlas, Ambari, and Ranger UIs.

About this task

When you enable SSO, unauthenticated users who try to access a service (e.g., Ambari, Atlas, etc), are redirected to the Knox SSO login page for authentication. This makes signing into services faster and easier, with fewer credentials to remember.

Before you begin

The Ambari Server must be running and you must be logged in as root.

Procedure

1. From the command line, begin the SSO setup wizard: ambari-server setup-sso.
2. When prompted, enter your Ambari Admin credentials.
3. Depending on your configuration, choose a path:
   - If SSO is not configured, it prompts Do you want to configure SSO authentication.
     - Enter y to continue through the wizard.
     - Enter n to exit the wizard.
   - If SSO is already configured, it prompts Do you want to disable SSO authentication.
     - Enter y to disable SSO for Ambari and the services (if services were being managed). Then it exits the wizard.
     - Enter n to continue through the wizard.
4. Enter the provider URL using the format: https://<hostname>:8443/gateway/knoxsso/api/v1/websso.
   https://dw-weekly.field.hortonworks.com:8443/gateway/knoxsso/api/v1/websso
5. Populate the Public Certificate PEM:
   a) Export the Knox certificate: ./knoxcli.sh export-cert --type PEM

```
[root@dw-weekly bin]# ./knoxcli.sh export-cert --type PEM
Certificate gateway-identity has been successfully exported to: /usr/
$REPO/$VERSION/knox/data/security/keystores/gateway-identity.pem
```

   b) Copy the contents of the file, excluding the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----.

```
[root@dw-weekly bin]# ./knoxcli.sh export-cert --type PEM
Certificate gateway-identity has been successfully exported to: /usr/
$REPO/$VERSION/knox/data/security/keystores/gateway-identity.pem
[root@dw-weekly bin]# vi /usr/$REPO/$VERSION/knox/data/security/
keystores/gateway-identity.pem

-----BEGIN CERTIFICATE-----
MIIoqToofo6gfwIffgIIfJG6+oql7YUwGQYJKoqIhvoNfQZFfQfwGTZLMfkGf1UZfhMoVVMxGTfL
fgNVffgTfFRlo3QxGTfLfgNVffoTfFRlo3QxGqfNfgNVffoTfkhhqG9voGZNMfsGf1UZoxMZVGVq
GGZoMoYGf1UZfxMfqHotG2Vlf2x5LmqpqWxkLmhvonRvfnGvomtqLmNvfTfZFw0xOGf2MTUxNjU0
MjffFw0xOTf2MTUxNjU0MjffMHUxoqfJfgNVffYTflVTMQ0wowYGVQQIZwRUqXN0MQ0wowYGVQQH
ZwRUqXN0MQ8wGQYGVQQKZwqIYWRvf3fxGTfLfgNVffsTfFRlo3QxKGfmfgNVffMTH2R3LXGlqWts
ZS5mfWVsqo5of3J0f253f3Jroy5jf20wgq8wGQYJKoqIhvoNfQZffQfGgY0fMIGJfoGffMjs9Q6M
f4f4Ussf/Yffpfr7k3Gx8v0/
Vlum6OL3Mr0vYQFtNSvGMZTZ25QQ8YHOvGf4frqi9lqwj6qwZYWf
RQUTIxuiOGPiMhK70onmLflmqpoGYmSJ3/shfOUoyN7+JiImYYn/
rJvt4Yt362gGvJynfsZGGKko
```

```
johF4v0FLoqGfgMfffZwGQYJKoqIhvoNfQZFfQfGgYZfZomm8ZTJJufW4vfp8O51Qx7J4ioY6G69
qgf76j4Oh8fqGqRVfoKYvrIZuJsZKHpIPGhtnVtqHG8YYf6vffSXoMmGpp5qfvZLfqnR1HNl6oZq
qf7J9qn9MPZqlrf5/kOGY85w0UUkVqotRLjsK/niHhojGKffJrok7hMUo7TYwfQ
-----END CERTIFICATE-----
```

c) When prompted Public Certificate PEM (empty line to finish input), paste the contents of the cert.pem file.

```
MIIoqToofo6gfwIffgIIfJG6+oql7YUwGQYJKoqIhvoNfQZFfQfwGTZLMfkGf1UZfhMoVVMxGTfL
fgNVffgTfFRlo3QxGTfLfgNVffoTfFRlo3QxGqfNfgNVffoTfkhhqG9voGZNMfsGf1UZoxMZVGVq
GGZoMoYGf1UZfxMfqHotG2Vlf2x5LmqpqWxkLmhvonRvfnGvomtqLmNvfTfZFw0xOGf2MTUxNjU0
MjffFw0xOTf2MTUxNjU0MjffMHUxoqfJfgNVffYTflVTMQ0wowYGVQQIZwRUqXN0MQ0wowYGVQQH
ZwRUqXN0MQ8wGQYGVQQKZwqIYWRvf3fxGTfLfgNVffsTfFRlo3QxKGfmfgNVffMTH2R3LXGlqWts
ZS5mfWVsqo5of3J0f253f3Jroy5jf20wgq8wGQYJKoqIhvoNfQZffQfGgY0fMIGJfoGffMjs9Q6M
f4f4Ussf/Yffpfr7k3Gx8v0/
Vlum6OL3Mr0vYQFtNSvGMZTZ25QQ8YHOvGf4frqi9lqwj6qwZYWf
RQUTIxuiOGPiMhK70onmLflmqpoGYmSJ3/shfOUoyN7+JiImYYn/
rJvt4Yt362gGvJynfsZGGKko
johF4v0FLoqGfgMfffZwGQYJKoqIhvoNfQZFfQfGgYZfZomm8ZTJJufW4vfp8O51Qx7J4ioY6G69
qgf76j4Oh8fqGqRVfoKYvrIZuJsZKHpIPGhtnVtqHG8YYf6vffSXoMmGpp5qfvZLfqnR1HNl6oZq
qf7J9qn9MPZqlrf5/kOGY85w0UUkVqotRLjsK/niHhojGKffJrok7hMUo7TYwfQ
```

6. When prompted Use SSO for Ambari [y/n] (n)?, enter Y to use or N to not use SSO for Ambari.

   Ambari does not need to be configured for SSO in order for the services to be configured for SSO (and vice-versa).

7. When prompted Manage SSO configurations for eligible services [y/n] (n)?, enter your selection.

   - y begins the service SSO setup wizard.
   - n exits the SSO setup wizard, saving your PEM setup and Ambari SSO selections.

   If you choose Y, the configurations for each eligible service are changed depending on the your selection when prompted.

   If you choose N, Ambari does not alter the existing configuration for any service. This is important if the cluster was set up using Blueprints and you do not want Ambari to change the SSO settings explicitly set.

8. If you chose y, you are prompted Use SSO for all services [y/n] (y)?.

   - y automatically sets up SSO for all available services.
   - n enters SSO set up for each individual service, allowing you to choose for which services you wish to enable SSO.

9. For the JWT Cookie name (), hadoop-jwt is the default.

10. Leave JWT audiences list empty.
    The prompt returns Ambari Server 'setup-sso' completed successfully.

11. Select **Ambari** > **Actions** > **Restart All Required** to restart all other services that require a restart.

Example

Example Knox SSO via ambari-server setup-sso

```
[root@dw-weekly ~]# $JAVA_HOME/bin/keytool -export -alias gateway-identity
 -rfc -file cert.pem -keystore /usr/$REPO/current/knox-server/data/security/
keystores/gateway.jks
[root@dw-weekly ~]# cd /usr/$REPO/current/knox-server/bin
[root@dw-weekly bin]# ./knoxcli.sh export-cert --type PEM
Certificate gateway-identity has been successfully exported to: /usr/$REPO/
$VERSION/knox/data/security/keystores/gateway-identity.pem
[root@dw-weekly bin]# vi /usr/$REPO/$VERSION/knox/data/security/keystores/
gateway-identity.pem
// <copy the certificate>
```

```
[root@dw-weekly ~]# ambari-server setup-sso
Using python  /usr/bin/python
```

```
Setting up SSO authentication properties...
Enter Ambari Admin login: admin
Enter Ambari Admin password:

SSO is currently not configured
Do you want to configure SSO authentication [y/n] (y)? y
Provider URL (https://knox.example.com:8443/gateway/knoxsso/api/v1/websso):
 https://dw-weekly.field.hortonworks.com:8443/gateway/knoxsso/api/v1/websso
Public Certificate PEM (empty line to finish input):
MIIoqToofo6gfwIffgIIfJG6+oql7YUwGQYJKoqIhvoNfQZFfQfwGTZLMfkGf1UZfhMoVVMxGTfL
fgNVffgTfFRlo3QxGTfLfgNVffoTfFRlo3QxGqfNfgNVffoTfkhhqG9voGZNMfsGf1UZoxMZVGVq
GGZoMoYGf1UZfxMfqHotG2Vlf2x5LmqpqWxkLmhvonRvfnGvomtqLmNvfTfZFw0xOGf2MTUxNjU0
MjffFw0xOTf2MTUxNjU0MjffMHUxoqfJfgNVffYTflVTMQ0wowYGVQQIZwRUqXN0MQ0wowYGVQQH
ZwRUqXN0MQ8wGQYGVQQKZwqIYWRvf3fxGTfLfgNVffsTfFRlo3QxKGfmfgNVffMTH2R3LXGlqWts
ZS5mfWVsqo5of3J0f253f3Jroy5jf20wgq8wGQYJKoqIhvoNfQZffQfGgY0fMIGJfoGffMjs9Q6M
f4f4Ussf/Yffpfr7k3Gx8v0/Vlum6OL3Mr0vYQFtNSvGMZTZ25QQ8YHOvGf4frqi9lqwj6qwZYWf
RQUTIxuiOGPiMhK70onmLflmqpoGYmSJ3/shfOUoyN7+JiImYYn/rJvt4Yt362gGvJynfsZGGKko
johF4v0FLoqGfgMfffZwGQYJKoqIhvoNfQZFfQfGgYZfZomm8ZTJJufW4vfp8O51Qx7J4ioY6G69
qgf76j4Oh8fqGqRVfoKYvrIZuJsZKHpIPGhtnVtqHG8YYf6vffSXoMmGpp5qfvZLfqnR1HNl6oZq
qf7J9qn9MPZqlrf5/kOGY85w0UUkVqotRLjsK/niHhojGKffJrok7hMUo7TYwfQ=

Use SSO for Ambari [y/n] (n)? y
Manage SSO configurations for eligible services [y/n] (n)? y
 Use SSO for all services [y/n] (n)? y
JWT Cookie name (hadoop-jwt): hadoop-jwt
JWT audiences list (comma-separated), empty for any ():
Ambari Server 'setup-sso' completed successfully.
```

```
[root@dw-weekly ~]# ambari-server restart
```

What to do next

You must next manually configure Knox SSO by using component configuration files. These steps are documented in “Set up Knox SSO via Component Config Files”.

Related Tasks

Install Knox

Set Up Knox Proxy

Configure Knox SSO for HDFS, Oozie, MapReduce2, Zeppelin, or YARN

Related Information

Configure Knox SSO for HDFS, Oozie, MapReduce2, Zeppelin, or YARN

Configure Knox SSO for HDFS, Oozie, MapReduce2, Zeppelin, or YARN

As of HDP-3.0.0, SSO is enabled using the ambari-server setup-sso wizard. SSO for Ambari, Atlas, and Ranger is automatically enabled by the wizard. To enable SSO for HDFS, Oozie, MapReduce2, Zeppelin, or YARN, you must manually change their configuration files. Users who try to access these components will be redirected to the Knox SSO login page for authentication.

Before you begin

You must be running Ambari 2.7.0.0 with HDP-3.0.0 or higher.

You must have already enabled SSO using ambari-server setup-sso.

Procedure

1. In Ambari, set the following properties for your components:

- HDFS: core-site.xml

```
"hadoop.http.authentication.type":
 "org.apache.hadoop.security.authentication.server.JWTRedirectAuthenticationHandler
"hadoop.http.authentication.public.key.pem": “$SSOPUBLICKEY"
"hadoop.http.authentication.authentication.provider.url":
 "$SSOPROVIDERURL"
```

- Oozie: oozie-site.xml

```
oozie.authentication.type=org.apache.hadoop.security.authentication.server.JWTRedir
oozie.authentication.authentication.provider.url=https://
$KNOX_HOST:8443/gateway/knoxsso/api/v1/websso
oozie.authentication.public.key.pem=$KNOX_PUBLIC_KEY
optional: oozie.authentication.expected.jwt.audiences=$AUDIENCES
 (default: EMPTY; which means ALL)
optional: oozie.authentication.jwt.cookie=$COOKIE-NAME (default: hadoop-
jwt)
```

- MapReduce2: core-site.xml

```
"hadoop.http.authentication.type":
 "org.apache.hadoop.security.authentication.server.JWTRedirectAuthenticationHandler
"hadoop.http.authentication.public.key.pem": “$SSOPUBLICKEY"
"hadoop.http.authentication.authentication.provider.url":
 "$SSOPROVIDERURL"
```

- Zeppelin: Advanced zeppelin-shiro-ini > shiro_ini_content

```
knoxJwtRealm = org.apache.zeppelin.realm.jwt.KnoxJwtRealm
knoxJwtRealm.providerUrl = $PROVIDERURL
knoxJwtRealm.login = gateway/knoxsso/knoxauth/login.html
knoxJwtRealm.publicKeyPath = $PATH_OF_KNOX-SSO.PEM
knoxJwtRealm.logoutAPI = false
knoxJwtRealm.logout = gateway/knoxssout/api/v1/webssout
knoxJwtRealm.cookieName = hadoop-jwt
knoxJwtRealm.redirectParam = originalUrl
knoxJwtRealm.groupPrincipalMapping = group.principal.mapping
knoxJwtRealm.principalMapping = principal.mapping
authc = org.apache.zeppelin.realm.jwt.KnoxAuthenticationFilter
```

- Zeppelin: Advanced spark2-env, for SPARK_HISTORY_OPTS

```
export SPARK_HISTORY_OPTS=’
-
Dspark.ui.filters=org.apache.hadoop.security.authentication.server.AuthenticationFi
-
Dspark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.params
 ="type=org.apache.hadoop.security.authentication.server.JWTRedirectAuthenticationH
kerberos.principal=$SPARK_HISTORY_KERBEROS_PRINCIPAL,
kerberos.keytab=$SPNEGO_KEYTAB,
authentication.provider.url=$PROVIDER_URL ,
public.key.pem=$PUBLIC_KEY”’
```

- YARN: core-site.xml

```
"hadoop.http.authentication.type":
 "org.apache.hadoop.security.authentication.server.JWTRedirectAuthenticationHandler
"hadoop.http.authentication.public.key.pem": “$SSOPUBLICKEY"
"hadoop.http.authentication.authentication.provider.url":
 "$SSOPROVIDERURL"
```

2. Click Save and confirm subsequent prompts.

3. Click **Ambari** > **Actions** > **Restart All Required** to restart all other services that require a restart.

Related Tasks

Install Knox

Set Up Knox Proxy

Enable Knox SSO Using the Ambari CLI

Related Information

Enable Knox SSO Using the Ambari CLI