

Apache Knox Authentication

Date published: 2021-02-29

Date modified: 2021-03-91



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Apache Knox Overview.....	4
Securing Access to Hadoop Cluster: Apache Knox.....	4
Apache Knox Gateway Overview.....	4
Knox Supported Services Matrix.....	5
Knox Topology Management in Cloudera Manager.....	6
Considerations for Knox.....	8
 Proxy Cloudera Manager through Apache Knox.....	 8
 Installing Apache Knox.....	 8
Apache Knox Install Role Parameters.....	10
 Management of Knox shared providers in Cloudera Manager.....	 11
Configure Apache Knox authentication for PAM.....	12
Configure Apache Knox authentication for AD/LDAP.....	13
Add a new shared provider configuration.....	15
TLS Mutual Authentication.....	16
Management of existing Apache Knox shared providers.....	17
Add a new provider in an existing provider configuration.....	17
Modify a provider in an existing provider configuration.....	19
Disable a provider in an existing provider configuration.....	20
Saving aliases.....	22
Configuring Kerberos authentication in Apache Knox shared providers.....	24
 Management of services for Apache Knox via Cloudera Manager.....	 26
Enable proxy for a known service in Apache Knox.....	26
Disable proxy for a known service in Apache Knox.....	28
Add custom service to existing descriptor in Apache Knox Proxy.....	29
Add a custom descriptor to Apache Knox.....	31
 Management of Service Parameters for Apache Knox via Cloudera Manager.....	 32
Add custom service parameter to descriptor.....	32
Modify custom service parameter in descriptor.....	34
Remove custom service parameter from descriptor.....	36

Apache Knox Overview

Securing Access to Hadoop Cluster: Apache Knox

The Apache Knox Gateway (“Knox”) is a system to extend the reach of Apache™ Hadoop® services to users outside of a Hadoop cluster without reducing Hadoop Security. Knox also simplifies Hadoop security for users who access the cluster data and execute jobs. The Knox Gateway is designed as a reverse proxy.

Establishing user identity with strong authentication is the basis for secure access in Hadoop. Users need to reliably identify themselves and then have that identity propagated throughout the Hadoop cluster to access cluster resources.

Layers of Defense for a CDP Private Cloud Base Cluster

- Authentication: Kerberos

Cloudera uses Kerberos for authentication. Kerberos is an industry standard used to authenticate users and resources within a Hadoop cluster. CDP also includes Cloudera Manager, which simplifies Kerberos setup, configuration, and maintenance.

- Perimeter Level Security: Apache Knox

Apache Knox Gateway is used to help ensure perimeter security for Cloudera customers. With Knox, enterprises can confidently extend the Hadoop REST API to new users without Kerberos complexities, while also maintaining compliance with enterprise security policies. Knox provides a central gateway for Hadoop REST APIs that have varying degrees of authorization, authentication, SSL, and SSO capabilities to enable a single access point for Hadoop.

Cloudera recommends that you leverage the default PAM Authentication Provider for the benefits in performance and ease of administration rather than direct LDAP. For more details, see *Considerations for Knox*.

- Authorization: Ranger

OS Security: Data Encryption and HDFS

Related Information

[Considerations for Knox](#)

Apache Knox Gateway Overview

A conceptual overview of the Apache Knox Gateway, a reverse proxy.

Overview

Knox integrates with Identity Management and SSO systems used in enterprises and allows identity from these systems be used for access to Hadoop clusters.

Knox Gateway provides security for multiple Hadoop clusters, with these advantages:

- Simplifies access: Extends Hadoop’s REST/HTTP services by encapsulating Kerberos to within the Cluster.
- Enhances security: Exposes Hadoop’s REST/HTTP services without revealing network details, providing SSL out of the box.
- Centralized control: Enforces REST API security centrally, routing requests to multiple Hadoop clusters.
- Enterprise integration: Supports LDAP, Active Directory, SSO, SAML and other authentication systems.

Typical Security Flow: Firewall, Routed Through Knox Gateway

Knox can be used with both unsecured Hadoop clusters, and Kerberos secured clusters. In an enterprise solution that employs Kerberos secured clusters, the Apache Knox Gateway provides an enterprise security solution that:

- Integrates well with enterprise identity management solutions
- Protects the details of the Hadoop cluster deployment (hosts and ports are hidden from end users)
- Simplifies the number of services with which a client needs to interact

Knox Gateway Deployment Architecture

Users who access Hadoop externally do so either through Knox, via the Apache REST API, or through the Hadoop CLI tools.

Knox Supported Services Matrix

A support matrix showing which services Apache Knox supports for Proxy and SSO, for both Kerberized and Non-Kerberized clusters.

Table 1: Knox Supported Components

Component	UI Proxy (with SSO)	API Proxy
Atlas API	#	#
Atlas UI	#	#
Beacon		
Cloudera Manager API	#	#
Cloudera Manager UI	#	
Data Analytics Studio (DAS)	#	
Druid		
Falcon		
Flink		
HBase REST API(aka WebHBase & Stargate)		#
HBase UI	#	
HDFS UI	#	
HiveServer2 HTTP JDBC API (HS2 via HTTP)		#
HiveServer2 LLAP JDBC API		
HiveServer2 LLAP UI		
HiveServer2 UI		
Hue	#	
Impala HTTP JDBC API		#
Impala UI	#	
JobHistory UI	#	
JobTracker		#
Kudu UI	#	
Livy API + UI	#	#
LogSearch		

Component	UI Proxy (with SSO)	API Proxy
NameNode	#	#
NiFi	#	#
NiFi Registry	#	#
Oozie API	#	#
Oozie UI	#	
Phoenix (aka Avatica)		#
Profiler	#	
Ranger API	#	#
Ranger UI	#	
ResourceManager API	#	#
Schema Registry API + UI	#	#
Streams Messaging Manager (SMM) API	#	#
Streams Messaging Manager (SMM) UI	#	
Solr	#	#
Spark3History UI	#	
SparkHistory UI	#	
Storm		
Storm LogViewer		
Superset		
WebHCat		
WebHDFS		#
YARN UI	#	
YARN UI V2	#	
Zeppelin UI	#	
Zeppelin WS	#	

**Note:**

APIs, UIs, and SSO in the Apache Knox project that are not listed above are considered Community Features.

Community Features are developed and tested by the Apache Knox community but are not officially supported by Cloudera. These features are excluded for a variety of reasons, including insufficient reliability or incomplete test case coverage, declaration of non-production readiness by the community at large, and feature deviation from Cloudera best practices. Do not use these features in your production environments.

Knox Topology Management in Cloudera Manager

In CDP Private Cloud, you can manage Apache Knox topologies via Cloudera Manager using `cdp-proxy` and `cdp-proxy-api`.

Shared providers

The Cloudera Manager configurations where the `cdp-proxy` and `cdp-proxy-api` topologies can be managed are:

- Knox Simplified Topology Management - `cdp-proxy`
- Knox Simplified Topology Management - `cdp-proxy-api`

- The SSO authentication provider is used by the UIs using the Knox SSO capabilities, such as the Admin and Home Page UIs.
- The API authentication provider is used by predefined topologies, such as admin, metadata or cdp-proxy-api.
- You can add or modify new or existing shared provider configurations.
- You can save aliases using a new Knox Gateway command.

Services

You can enable or disable known or custom services in Knox proxy via Cloudera Manager.

There are two kinds of services in cdp-proxy:

- Known: officially-supported Knox services. Cloudera Manager provides and manages all the required service definition files.
- Custom: unofficial, tech preview, or community feature Knox services. You must supply the service definition files (service.xml and rewrite.xml) that exist in the KNOX_DATA_DIR/services folder. These are not recommended for production environments, and not supported by Cloudera.



Important:

These topologies will be deployed by Cloudera Manager only if Knox's service auto-discovery feature is turned on using the Enable/Disable Service Auto-Discovery checkbox on Cloudera Manager UI:



Important: Adding a custom service will only work if you provide the service definition files (service.xml and rewrite.xml) in the KNOX_DATA_DIR/services folder.

Service parameters

You can add, modify, or remove custom service parameters in Knox proxy via Cloudera Manager.

Considerations for Knox

Learn about the considerations before you get started with Knox.

Default PAM settings for Knox

Secure clusters require local OS accounts. Local OS accounts are most often achieved by using something like SSSD or Centrify which localizes user accounts from user stores or directories including LDAP/AD and so on.

This means that you should be able to use PAM and the local OS accounts straight away as long as they are on the same host as Knox. There are various ways to make local OS accounts available. SSSD or Centrify are technical solutions that make it seem like there are local OS accounts even though they are only in LDAP/AD. You can use real local OS accounts as well.

Proxy Cloudera Manager through Apache Knox

In order to have Cloudera Manager proxied through Knox, there are some steps you must complete.

Procedure

1. Set the value for `frontend_url`: Cloudera Manager Administration Settings Cloudera Manager Frontend URL :
 - Non-HA value: `https://$Knox_host:$knox_port`
 - HA value: `https://$Knox_loadbalancer_host:$Knox_loadbalancer_port`
2. Set allowed groups, hosts, and users for Knox Proxy: Cloudera Manager Administration Settings External Authentication :
 - Allowed Groups for Knox Proxy: *
 - Allowed Hosts for Knox Proxy: *
 - Allowed Users for Knox Proxy: *
3. Enable Kerberos/SPNEGO authentication for the Admin Console and API: Cloudera Manager Administration Settings External Authentication Enable SPNEGO/Kerberos Authentication for the Admin Console and API: : `true`
4. From Cloudera Manager Administration Settings External Authentication , set Knox Proxy Principal: `knox`.

What to do next

External authentication must be set up correctly. Cloudera Manager must be configured to use LDAP, following the standard procedure for setting up LDAP. This LDAP server should be the same LDAP that populates local users on Knox hosts (if using PAM authentication with Knox), or the same LDAP that Knox is configured to use (if using LDAP authentication with Knox).

Installing Apache Knox

This document provides instructions on how to install Apache Knox using the installation process.

About this task

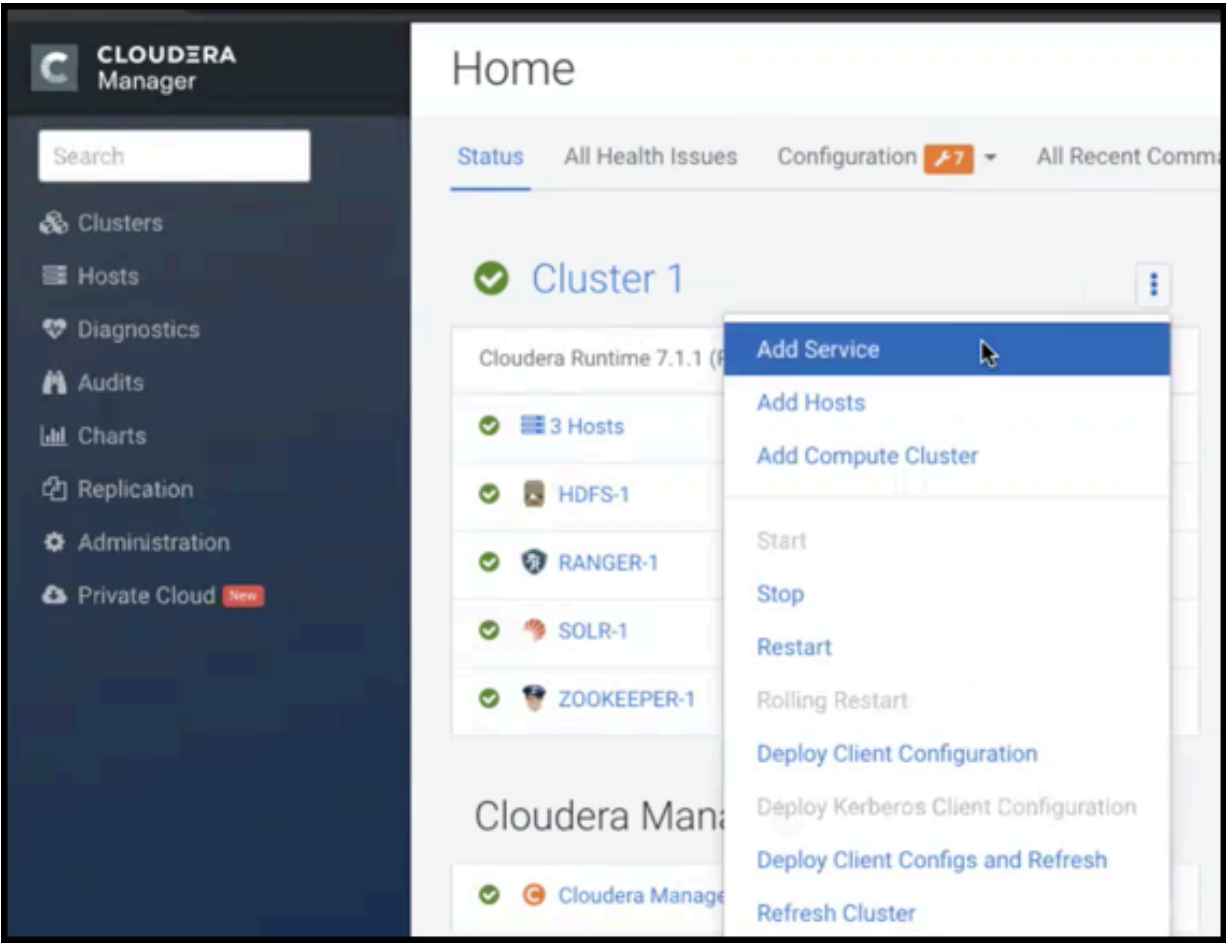
Apache Knox is an application gateway for interacting with the REST APIs and UIs. The Knox Gateway provides a single access point for all REST and HTTP interactions in your Cloudera Data Platform cluster.

Before you begin

When installing Knox, you must have Kerberos enabled on your cluster.

Procedure

- 1. From your Cloudera Manager homepage, go to Status tab \$Cluster Name ... Add Service



- 2. From the list of services, select Knox and click Continue.
- 3. On the **Select Dependencies** page, choose the dependencies you want Knox to set up:

HDFS, Ranger, Solr, Zookeeper	For users that require Apache Ranger for authorization. HDFS with Ranger. HDFS depends on Zookeeper, and Ranger depends on Solr.
HDFS, Zookeeper	HDFS depends on Zookeeper.
No optional dependencies	For users that do not wish to have Knox integrate with HDFS or Ranger.

- 4. On the **Assign Roles** page, select role assignments for your dependencies and click Continue:

Knox service roles	Description	Required?
Knox Gateway	If Knox is installed, at least one instance of this role should be installed. This role represents the Knox Gateway which provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters.	Required

Knox service roles	Description	Required?
KnoxIDBroker*	It is strongly recommended that this role is installed on its own dedicated host. As its name suggests this role will allow you to take advantage of Knox's Identity Broker capabilities, an identity federation solution that exchanges cluster authentication for temporary cloud credentials.*	Optional*
Gateway	This role comes with the CSD framework. The gateway structure is used to describe the client configuration of the service on each host where the gateway role is installed.	Optional

* Note: KnoxIDBroker appears in the Assign Roles page, but it is not currently supported in CDP Private Cloud.

5. On the **Review Changes** page, most of the default values are acceptable, but you must Enable Kerberos Authentication and supply the Knox Master Secret. There are additional parameters you can specify or change, listed in "Knox Install Role Parameters".
 - a) Click Enable Kerberos Authentication

Kerberos is required where Knox is enabled.
 - b) Supply the Knox Master Secret, e.g. `knoxsecret`.
 - c) Click Continue.
6. The **Command Details** page shows the status of your operation. After completion, your system admin can view logs for your installation under `stdout`.

Apache Knox Install Role Parameters

Reference information on all the parameters available for Knox service roles.

Service-level parameters

Table 2: Required service-level parameters

Name	In Wizard	Type	Default Value
<code>kerberos.auth.enabled*</code>	Yes	Boolean	false
<code>ranger_knox_plugin_hdfs_audit_directory</code>	No	Text	<code>\${ranger_base_audit_url}/knox</code>
<code>autorestart_on_stop</code>	No	Boolean	false
<code>knox_pam_realm_service</code>	No	Text	login
<code>save_alias_command_input_password</code>	No	Text	-

Knox Gateway role parameters

Table 3: Required parameters for Knox Gateway role

Name	In Wizard	Type	Default Value
<code>gateway_master_secret</code>	Yes	Password	-
<code>gateway_conf_dir</code>	Yes	Path	<code>/var/lib/knox/gateway/conf</code>
<code>gateway_data_dir</code>	Yes	Path	<code>/var/lib/knox/gateway/data</code>
<code>gateway_port</code>	No	Port	8443
<code>gateway_path</code>	No	Text	gateway

Name	In Wizard	Type	Default Value
gateway_heap_size	No	Memory	1 GB (min = 256 MB; soft min = 512 MB)
gateway_ranger_knox_plugin_conf_path	No	Path	/var/lib/knox/ranger-knox-plugin
gateway_ranger_knox_plugin_policy_cache_directory	No	Path	/var/lib/ranger/knox/gateway/policy-cache
gateway_ranger_knox_plugin_hdfs_audit_spool_directory	No	Path	/var/log/knox/gateway/audit/hdfs/spool
gateway_ranger_knox_plugin_solr_audit_spool_directory	No	Path	/var/log/knox/gateway/audit/solr/spool

Table 4: Optional parameters for Knox Gateway role

Name	Type	Default Value
gateway_default_topology_name	Text	cdp-proxy
gateway_auto_discovery_enabled	Boolean	true
gateway_cluster_configuration_monitor_interval	Time	60 seconds (minimum = 30 seconds)
gateway_auto_discovery_advanced_configuration_monitor_interval	Time	10 seconds (minimum = 5 seconds)
gateway_cloudera_manager_descriptors_monitor_interval	Time	10 seconds (minimum = 5 seconds)
gateway_auto_discovery_cdp_proxy_enabled_*	Boolean	true
gateway_auto_discovery_cdp_proxy_api_enabled_*	Boolean	true
gateway_descriptor_cdp_proxy	Text Array	Contains the required properties of cdp-proxy topology
gateway_descriptor_cdp_proxy_api	Text Array	Contains the required properties of cdp-proxy-api topology
gateway_sso_authentication_provider	Text Array	Contains the required properties of the authentication provider used by the UIs using the Knox SSO capabilities (Admin UI and Home Page). Defaults to PAM authentication.
gateway_api_authentication_provider	Text Array	Contains the required properties of the authentication provider used by pre-defined topologies such as admin, metadata or cdp-proxy-api. Defaults to PAM authentication.

Management of Knox shared providers in Cloudera Manager

Information on CDP Private Cloud topology management for Knox from within Cloudera Manager.

- Modifying the SSO authentication provider used by the UIs using the Knox SSO capabilities, such as the Admin and Home Page UIs.
- Modifying the API authentication provider used by predefined topologies, such as admin, metadata or cdp-proxy-api.
- Adding/modifying new/existing shared provider configurations.
- Saving aliases using a new Knox Gateway command.

Configure Apache Knox authentication for PAM

Knox authentication configurations for PAM in Cloudera Manager. PAM is the default SSO authentication provider in CDP Private Cloud.



Note: The Knox user needs permission on `/etc/shadow` for PAM authentication. Allow the Knox user to read the `/etc/shadow` file:

```
groupadd shadow
usermod -a -G shadow Knox
chgrp shadow /etc/shadow
chmod g+r /etc/shadow
```

SSO authentication for PAM

In CDP Private Cloud, Cloudera Manager added a new Knox configuration, called Knox Simplified Topology Management - SSO Authentication Provider, with the following initial configuration:

```
role=authentication
authentication.name=ShiroProvider
authentication.param.sessionTimeout=30
authentication.param.redirectToUrl=${GATEWAY_PATH}/knoxssso/knoxauth/login.html
authentication.param.restrictedCookies=rememberme,WWW-Authenticate
authentication.param.urls./*=authcBasic
authentication.param.main.pamRealm=org.apache.knox.gateway.shirorealm.KnoxPamRealm
authentication.param.main.pamRealm.service=login
```

The screenshot shows the Cloudera Manager interface for Cluster 1. The top navigation bar includes 'Status', 'Instances', 'Configuration' (selected), 'Commands', 'Charts Library', 'Audits', 'Knox Gateway UI', and 'Quick Links'. A search bar contains 'SSO Authentication Provider'. The left sidebar shows filters for SCOPE (KNOX-1, Gateway, Knox Gateway, Knox IDBroker) and CATEGORY (Advanced, Logs, Main, Monitoring, Performance, Ports and Addresses, Resource Management, Security, Stacks Collection). The main content area displays the configuration for 'Knox Simplified Topology Management - SSO Authentication Provider' (gateway_sso_authentication_provider). The configuration is organized into two columns: 'Knox Gateway Default Group' and 'Knox Gateway Default Group'. The 'Knox Gateway Default Group' column contains the following configuration items:

- role=authentication
- authentication.name=ShiroProvider
- authentication.param.sessionTimeout=30
- authentication.param.redirectToUrl=\${GATEWAY_PATH}/knoxssso/knoxauth/login.html
- authentication.param.restrictedCookies=rememberme,WWW-Authenticate
- authentication.param.main.pamRealm=org.apache.knox.gateway.shirorealm.KnoxPamRealm
- authentication.param.main.pamRealm.service=login
- authentication.param.urls./*=authcBasic

Every change here is applied to the KnoxSSO topology that affects manager, homepage and cdp-proxy topologies as they are using the federation provider.

API authentication for PAM

A new Knox configuration has been added for CDP Private Cloud, called Knox Simplified Topology Management - API Authentication Provider, with the following initial configuration:

```
role=authentication
```

```
authentication.name=ShiroProvider
authentication.param.sessionTimeout=30
authentication.param.urls./*=authcBasic
authentication.param.main.pamRealm=org.apache.knox.gateway.shiorealm.Knox
PamRealm
authentication.param.main.pamRealm.service=login
```

Every change here is applied to the admin, metadata, and cdp-proxy-api topologies.

Configure Apache Knox authentication for AD/LDAP

Knox authentication configurations for LDAP and AD in Cloudera Manager.

SSO authentication for AD/LDAP

In the following sample you will see how to change the PAM authentication (which comes default with Knox) to LDAP authentication. It is as simple as removing the default PAM related configuration in ShiroProvider and add LDAP related properties (e.g. with demo LDAP server configuration):

```
role=authentication
authentication.name=ShiroProvider
authentication.param.sessionTimeout=30
authentication.param.redirectToUrl=/${GATEWAY_PATH}/knoxssso/knoxauth/login.
html
authentication.param.restrictedCookies=rememberme,WWW-Authenticate
authentication.param.urls./*=authcBasic
authentication.param.main.ldapRealm=org.apache.knox.gateway.shiorealm.Knox
LdapRealm
authentication.param.main.ldapContextFactory=org.apache.knox.gateway.shiro
realm.KnoxLdapContextFactory
authentication.param.main.ldapRealm.contextFactory=$ldapContextFactory
authentication.param.main.ldapRealm.contextFactory.authenticationMechanism=s
imple
authentication.param.main.ldapRealm.contextFactory.url=ldap://localhost:33
389
authentication.param.main.ldapRealm.contextFactory.systemUsername=uid=guest,
ou=people,dc=hadoop,dc=apache,dc=org
authentication.param.main.ldapRealm.contextFactory.systemPassword=${ALIAS=k
noxLdapSystemPassword}
authentication.param.main.ldapRealm.userDnTemplate=uid={0},ou=people,dc=h
adoop,dc=apache,dc=org
authentication.param.remove=main.pamRealm
authentication.param.remove=main.pamRealm.service
```

After you finished editing the properties you have to save the configuration changes. This will make the Refresh Needed stale configuration indicator appear. Once the cluster refresh finishes, all topologies that are configured to use Knox SSO will be authenticated by the configured LDAP server.

Filters

SCOPE

KNOX-1 (Service-Wide)	0
Gateway	0
Knox Gateway	1
Knox IDBroker	0

CATEGORY

Advanced	0
Logs	0
Main	1
Monitoring	0
Performance	0
Ports and Addresses	0
Resource Management	0
Security	0
Stacks Collection	0

STATUS

Error	0
Warning	0
Edited	0
Non-default	1
Has Overrides	0

Knox Simplified Topology Management - SSO Authentication Provider
gateway_sso_authentication_provider

Knox Gateway Default Group

role=authentication

authentication.name=ShiroProvider

authentication.param.sessionTimeout=30

authentication.param.redirectUri=/\${GATEWAY_PATH}/knoxssso/knoxauth/login.html

authentication.param.restrictedCookies=rememberme,WWW-Authenticate

authentication.param.urls./**=authcBasic

authentication.param.main.LdapRealm=org.apache.knox.gateway.shirorealm.KnoxLdapRealm

authentication.param.main.LdapContextFactory=org.apache.knox.gateway.shirorealm.KnoxLdapContextFactory

authentication.param.main.LdapRealm.contextFactory=\$ldapContextFactory

authentication.param.main.LdapRealm.contextFactory.authenticationMechanism=simple

authentication.param.main.LdapRealm.contextFactory.url=ldap://localhost:33389

authentication.param.main.LdapRealm.contextFactory.systemUsername=uid=guest,ou=people,dc=org

authentication.param.main.LdapRealm.contextFactory.systemPassword=\${ALIAS=knoxLdapSystem}

authentication.param.main.LdapRealm.userDnTemplate=uid={0},ou=people,dc=hadoop,dc=apache,dc=org

authentication.param.remove=main.pamRealm

authentication.param.remove=main.pamRealm.service



Note:

As you can see we used a Knox alias when we declared the system password instead of writing the plain text password there. To make it easier for the end-users a new Knox Gateway command was created that allows them to save aliases on all hosts where a Knox Gateway is running. See [Saving aliases](#).

To verify:

```
$ curl -ku knoxui:knoxui 'https://johndoe-1.abc.cloudera.com:8443/gateway/admin/api/v1/providerconfig/knoxssso'

{
  "role" : "authentication",
  "name" : "ShiroProvider",
  "enabled" : true,
  "params" : {
    "main.LdapContextFactory" : "org.apache.knox.gateway.shirorealm.KnoxLdapContextFactory",
    "main.LdapRealm" : "org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm",
    "main.LdapRealm.contextFactory" : "$ldapContextFactory",
    "main.LdapRealm.contextFactory.authenticationMechanism" : "simple",
    "main.LdapRealm.contextFactory.systemPassword" : "${ALIAS=knoxLdapSystem}",
    "main.LdapRealm.contextFactory.systemUsername" : "uid=guest,ou=people,dc=hadoop,dc=apache,dc=org",
    "main.LdapRealm.contextFactory.url" : "ldap://localhost:33389",
    "main.LdapRealm.userDnTemplate" : "uid={0},ou=people,dc=hadoop,dc=apache,dc=org",
    "redirectToUrl" : "/${GATEWAY_PATH}/knoxssso/knoxauth/login.html",
    "restrictedCookies" : "rememberme,WWW-Authenticate",
    "sessionTimeout" : "30",
    "urls./**" : "authcBasic"
  }
}
```



Note: Any change in SSO authentication configuration alters the Knox SSO topology. This affects the manager, homepage, and cdp-proxy topologies because the SSO cookie federation provider is used.

API authentication for AD/LDAP

In the following sample you will see how to change the PAM authentication (which comes default with Knox) to LDAP authentication:

```
role=authentication
authentication.name=ShiroProvider
authentication.param.sessionTimeout=30
authentication.param.urls./*=authcBasic
authentication.param.main.ldapRealm=org.apache.knox.gateway.shirolealm.KnoxLdapRealm
authentication.param.main.ldapContextFactory=org.apache.knox.gateway.shirolealm.KnoxLdapContextFactory
authentication.param.main.ldapRealm.contextFactory=$ldapContextFactory
authentication.param.main.ldapRealm.contextFactory.authenticationMechanism=simple
authentication.param.main.ldapRealm.contextFactory.url=ldap://localhost:33389
authentication.param.main.ldapRealm.contextFactory.systemUsername=uid=guest,ou=people,dc=hadoop,dc=apache,dc=org
authentication.param.main.ldapRealm.contextFactory.systemPassword=${ALIAS=knoxLdapSystemPassword}
authentication.param.main.ldapRealm.userDnTemplate=uid={0},ou=people,dc=hadoop,dc=apache,dc=org
authentication.param.remove=main.pamRealm
authentication.param.remove=main.pamRealm.service
```

Every change here goes directly into admin, metadata, and cdp-proxy-api topologies.

Add a new shared provider configuration

Provider configurations are definitions of authentication and authorization controls for services proxied by Knox, which may be referenced by one or more descriptors.

Procedure

1. Define the providers:

- From Cloudera Manager Knox Configuration, add a new entry in Knox Gateway Advanced Configuration Snippet (Safety Valve) for `conf/cdp-resources.xml_role_safety_valve`.
- Name the provider configuration, and specify the desired providers and their corresponding configuration attributes.

Provider configuration entries are named as `providerConfigs:TOPOLOGY_NAME` (E.G., `providerConfigs:myTopology`).

- For each provider, the role is declared (E.G., `role=authentication`, `role=authorization`), and subsequently configured by defining properties of that role (E.G., `authentication.name=ShiroProvider`, `authentication.param.sessionTimeout=30`).

Example (LDAP authentication and Ranger authorization)

- Name=`providerConfigs:ldap-ranger-provider`
- Value=

```
role=authentication#
authentication.name=ShiroProvider#
authentication.param.sessionTimeout=30#
```

```

authentication.param.main.ldapRealm=org.apache.hadoop.gateway.shi
rorealm.KnoxLdapRealm#
authentication.param.main.ldapContextFactory=org.apache.knox.gate
way.shi
rorealm.KnoxLdapContextFactory#
authentication.param.main.ldapRealm.contextFactory=$ldapContext
Factory#
authentication.param.main.ldapRealm.contextFactory.authentication
Mechanism=simple#
authentication.param.main.ldapRealm.contextFactory.url=ldap://ld
ap-host:33389#
authentication.param.main.ldapRealm.contextFactory.systemUserna
me=uid=
guest,ou=people,dc=hadoop,dc=apache,dc=org#
authentication.param.main.ldapRealm.userDnTemplate=uid={0},ou=
people,dc=
hadoop,dc=apache,dc=org#
authentication.param.urls./**=authcBasic#
role=authorization#
authorization.name=XASecurePDPKnox

```

2. Save your changes.
3. Refresh your cluster: the Refresh needed stale configuration indicator appears; click it and wait until the refresh process completes.
4. Validate:

Using the Knox Admin UI (https://KNOX_GATEWAY_HOST:PORT/GATEWAY_PATH/gateway/manager/admin-ui/), navigate to the Provider Configurations, and verify that your provider configuration was generated with the providers and parameters you specified.

TLS Mutual Authentication

Mutual authentication with TLS provides the Knox gateway with the means to establish a strong trust relationship with another party. This is especially useful when applications that act on behalf of end-users send requests to Knox.

While this feature does establish an authenticated trust relationship with the client application, it does not determine the end-user identity through this authentication. It will continue to look for credentials or tokens that represent the end-user within the request and authenticate or federate the identity accordingly.

To enable TLS Mutual Authentication, set the following in CM Knox Configuration Knox Service (or Gateway) Advanced Configuration Snippet (Safety Valve) for `conf/gateway-site.xml`:

```
gateway.client.auth.needed = true
```

The truststore path for client authentication can be set in Cloudera Manager Knox Configuration Knox Service (or Gateway) Advanced Configuration Snippet (Safety Valve) for `conf/gateway-site.xml`

```
gateway.truststore.path
```

This parameter can point to the standard truststore (typically `truststore.jks`) on the host if it contains all of the necessary `TrustedCertEntry`'s for client authentication. Even if the JKS is password protected, Knox can still get `TrustedCertEntry` content from it, so no password is necessary.

If `gateway.client.auth.needed = true` and `gateway.truststore.path` is unset, then it will look at this default location `/var/lib/knox/gateway/data/security/keystores/gateway.jks` for truststore AND keystore entries, which is an atypical configuration for JKS usage in our stack and not recommended.

These two parameters are distinct, but can point to the same truststore: `gateway.truststore.path` is for client authentication in the context of TLS Mutual Authentication, and `gateway.httpclient.truststore.path` is for when the Knox Gateway is an HTTP Client to other TLS servers.

Management of existing Apache Knox shared providers

You can add, modify, or disable an existing shared provider configuration in Apache Knox via Cloudera Manager.

The following default shared provider configurations are deployed in CDP Private Cloud with Knox:

Table 5: Default shared provider configurations

Configuration	Used by these topologies
admin	admin
homepage	homepage
knoxsso	homepage cdp-proxy manager
manager	manager
metadata	metadata
pam	cdp-proxy-api
sso	cdp-proxy



Note: pam and sso are available only if service auto-discovery is enabled for Knox Gateway role.

The following changes are allowed in any of these shared providers:

- Disable a particular provider
- Modify a particular provider
- Add a new provider

All of these actions can be done via editing the Knox Gateway Advanced Configuration Snippet (Safety Valve) for `conf/cdp-descriptors.xml` by implementing the following language elements:

- The key of a new entry should be like this: `providerConfigs: providerConfig_1 [,providerConfig_2,...,providerConfig_3]`
- The value should contain the following name/value pairs separated by a hash (#) character:

```
role=webappsec|authentication|federation|identity-assertion|authorization|
hostmap|ha
$role.name=ROLE_NAME (e.g. ShiroProvider)
$role.enabled=true|false (optional; defaults to 'true')
$role.param.param_1=value_1 (parameters are optional too)
...
$role.param_N.param1=value_N
```

Add a new provider in an existing provider configuration

An example of how to add a new provider to the authorization provider in the manager shared provider configuration.

About this task

In this example you will see how to add a new HA provider (this time only the ATLAS service will be configured for high availability) in the manager shared provider configuration. This particular authorization provider is set as follows (in its JSON descriptor):

```
{
  "role": "authorization",
```

```

    "name": "AclsAuthz",
    "enabled": "true",
    "params": {
      "knox.acl.mode": "OR",
      "knox.acl": "KNOX_ADMIN_USERS;KNOX_ADMIN_GROUPS;*"
    }
  }
}

```

Procedure

- From Cloudera Manager Knox Configuration, add the following entry in the Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml:
 - name = providerConfigs:manager
 - value = role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#authorization.param.knox.acl=myTestUser;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR#role=ha#ha.name=HaProvider#ha.param.ATLAS=enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRet

The screenshot shows the Cloudera Manager interface for Knox Gateway configuration. The left sidebar contains a 'Filters' section with 'SCOPE' and 'CATEGORY' filters. The main area displays the 'Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml'. The 'Name' field is set to 'providerConfigs:manager' and the 'Value' field contains the configuration snippet: 'role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#authorization.param.knox.acl=myTestUser;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR#role=ha#ha.name=HaProvider#ha.param.ATLAS=enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRet'. The 'Description' field is empty. There is a 'Final' checkbox and a 'View as XML' link.

This screenshot shows the same configuration page as the previous one, but with the 'View Editor' link clicked. The 'Value' field now displays the configuration snippet in XML format:


```

<property>
  <name>providerConfigs:manager</name>
  <value>role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#authorization.param.knox.acl=myTestUser;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR#role=ha#ha.name=HaProvider#ha.param.ATLAS=enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000</value>
</property>
    
```

 The 'Final' checkbox is still present, and there is a 'View Editor' link.

- Save your changes.
- Refresh the cluster.
- Validate:

```

$ curl -ku knoxui:knoxui 'https://johndoe-1.abc.cloudera.com:8443/gateway/admin/api/v1/providerconfig/manager'
{
  "providers" : [
    ...
  ], {
    "role" : "authorization",

```

```

    "name" : "AclsAuthz",
    "enabled" : false,
    "params" : {
      "knox.acl" : "myTestUser;KNOX_ADMIN_GROUPS;*",
      "knox.acl.mode" : "OR"
    }
  }, {
    "role" : "ha",
    "name" : "HaProvider",
    "enabled" : true,
    "params" : {
      "ATLAS" : "enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000"
    }
  }
]
}

```

Modify a provider in an existing provider configuration

An example of how to modify the authorization provider in the manager shared provider configuration.

About this task

In this example you will see how to modify the authorization provider in the manager shared provider configuration. This particular authorization provider is set as follows (in its JSON descriptor):

```

{
  "role": "authorization",
  "name": "AclsAuthz",
  "enabled": "true",
  "params": {
    "knox.acl.mode": "OR",
    "knox.acl": "KNOX_ADMIN_USERS;KNOX_ADMIN_GROUPS;*"
  }
}

```

Procedure

1. From Cloudera Manager Knox Configuration, add the following entry in the Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml:
 - name = providerConfigs:manager
 - value = role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#authorization.param.knox.acl=myTestUser;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR

The screenshot shows the Cloudera Manager interface for editing a Knox Gateway Advanced Configuration Snippet. The snippet is named 'providerConfigs:manager' and its value is 'role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#authorization.param.knox.acl=myTestUser;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR'. The interface includes a search bar, filters on the left, and a form to edit the snippet.

With this change you are authorizing a user called myTestUser to login and execute administrative actions on the Knox Admin UI.

2. Save your changes.
3. Refresh the cluster.
4. Validate:

```
$ curl -ku KnoxUI:knoxui 'https://johndoe-1.abc.cloudera.com:8443/gateway/admin/api/v1/providerconfig/manager'
{
  "providers" : [
    ...
  ], {
    "role" : "authorization",
    "name" : "AclsAuthz",
    "enabled" : false,
    "params" : {
      "knox.acl" : "myTestUser;KNOX_ADMIN_GROUPS;*",
      "knox.acl.mode" : "OR"
    }
  }, {
    "role" : "ha",
    "name" : "HaProvider",
    "enabled" : true,
    "params" : {
      "ATLAS" : "enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000"
    }
  }
]
```

Disable a provider in an existing provider configuration

An example of how to disable the authorization provider in the manager shared provider configuration.

About this task

In this example you will see how to disable the authorization provider in the manager shared provider configuration. This particular authorization provider is set as follows (in its JSON descriptor):

```
{
  "role": "authorization",
  "name": "AclsAuthz",
  "enabled": "true",
  "params": {
    "knox.acl.mode": "OR",
    "knox.acl": "KNOX_ADMIN_USERS;KNOX_ADMIN_GROUPS;*"
  }
}
```

Procedure

1. From Cloudera Manager Knox Configuration, add the following entry in the Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-resources.xml:

- name = providerConfigs:manager
- value = role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#authorization.param.knox.acl=KNOX_ADMIN_USERS;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR

2. Save your changes.
3. Refresh the cluster.
4. Validate:

```
$ curl -ku knoxui:knoxui 'https://johndoe-1.abc.cloudera.com:8443/gateway/admin/api/v1/providerconfig/manager'
{
  "providers" : [
    ...
  ], {
    "role" : "authorization",
    "name" : "AclsAuthz",
    "enabled" : false,
    "params" : {
      "knox.acl" : "myTestUser;KNOX_ADMIN_GROUPS;*",
      "knox.acl.mode" : "OR"
    }
  }, {
    "role" : "ha",
    "name" : "HaProvider",
```

```

    "enabled" : true,
    "params" : {
      "ATLAS" : "enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000"
    }
  }
}

```

What to do next

The only change is that the enabled flag was changed to false.

Saving aliases

There is a new command available for the Knox Gateway role which allows end-users to save an alias=password pair to an arbitrary number of topologies on each host where an instance of the Knox Gateway is installed without the need of running the Knox CLI tool manually.

A new password-type input field is added, called `save_alias_command_input_password`. The format of an entry in this input field should be: `topology_name_1[:topology_name_2[:...:topology_name_N]].alias_name=password`

Example: `cdp-proxy-api:admin:metadata.knoxLdapSystemPassword=guest-password`.

After the end-user entered a meaningful and valid value and saved the configuration changes he/she can run the command from Knox's action list: Actions/Save Alias.



Tip: If you need to add a Gateway level alias, please use `__gateway` as topology name. For instance: `__gateway.knoxLdapSystemPassword=admin-password`.

Cluster 1

KNOX-1

Configuration

Save Alias

Filters

SCOPE

- KNOX-1 (Service-Wide) 1
- Gateway 0
- Knox Gateway 0
- Knox IDBroker 0

Save Alias Command Input

save_alias_command_input_password

KNOX-1 (Service-Wide)

.....

Per Page 25 1 - 25 of 216

Cluster 1

KNOX-1

Instances

Search

Filters

STATUS

- Good Health 3

COMMISSION STATE

MAINTENANCE MODE

RACK ID

Actions for Selected

Status	Role Type	State	Hostname	Commission State	Role Group
Good Health	Knox Gateway	Started	cloudera.com	Commissioned	Knox Gateway Default Group
Good Health	Knox Gateway	Started	cloudera.com	Commissioned	Knox Gateway Default Group
Good Health	Knox Gateway	Started	cloudera.com	Commissioned	Knox Gateway Default Group

Last Updated: Apr 2, 3:05:30 AM PDT

1 - 3 of 3

Cluster 1

✓ K

KNOX-1

Status

Instances

Config

Health Tests

✓ Knox Gateway Health

Actions

Start

Restart

Rolling Restart

Save Alias

Stop

Actions

Save Alias

Are you sure you want to run the **Save Alias** command on the service **KNOX-1**?

Cancel

Save Alias

Save Alias

Status ✓ Finished Context [KNOX-1](#) Apr 2, 3:06:24 AM 29.99s

Command Save Alias finished successfully on service KNOX-1.

Completed 1 of 1 step(s).

Show All Steps

Show Only Failed Steps

Show Only Running Steps

Execute 3 steps in parallel

Successfully completed 3 steps.

Execute command Save Alias on role Knox Gateway

Knox Gateway

Apr 2, 3:06:24 AM

24.37s

Execute command Save Alias on role Knox Gateway

Knox Gateway

Apr 2, 3:06:25 AM

28.97s

Execute command Save Alias on role Knox Gateway

Knox Gateway

Apr 2, 3:06:26 AM

22.24s

Close

23

Save Alias

Execute command Save Alias on role Knox Gateway

Knox Gateway

Apr 2, 3:06:24 AM

24.37s

Execute command Save Alias on role Knox Gateway

Knox Gateway

Apr 2, 3:06:25 AM

28.97s

Execute command Save Alias on role Knox Gateway

Knox Gateway

Apr 2, 3:06:26 AM

22.24s

Command (Save Alias (590)) has completed successfully

Save Alias

Save Alias finished successfully on Knox Gateway

Knox Gateway

Apr 2, 3:06:26 AM

22.21s

\$> csd/csd.sh []

stdout stderr Role Log

```

Thu Apr  2 03:06:35 PDT 2020
JAVA_HOME=/usr/java/jdk1.8.0_232-cloudera
Using -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp/KNOX-1-KNOX_GATEWAY-...hprof -
XX:OnOutOfMemoryError=/opt/cloudera/cm-agent/service/common/killparent.sh as CSD_JAVA_OPTS
Using /var/run/cloudera-scm-agent/process/119-knox-KNOX_GATEWAY-SaveAliasCommand as conf dir
Using scripts/saveAliasCommand.sh as process script
CONF_DIR=/var/run/cloudera-scm-agent/process/119-knox-KNOX_GATEWAY-SaveAliasCommand
CMF_CONF_DIR=
Creating alias knoxLdapSystemPassword for topology cdp-proxy-api...
knoxLdapSystemPassword has been successfully created.
Creating alias knoxLdapSystemPassword for topology admin...
knoxLdapSystemPassword has been successfully created.
Creating alias knoxLdapSystemPassword for topology metadata...
knoxLdapSystemPassword has been successfully created.

```

Close

Configuring Kerberos authentication in Apache Knox shared providers

An example of how to add the kerberos-auth configuration provider from Cloudera Manager.

Procedure

- From Cloudera Manager Knox Configuration, add the following entry in the Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-resources.xml:

- Name = providerConfigs:kerberos-providers
- Value =

```

role=authentication#
authentication.name=HadoopAuth#
authentication.param.sessionTimeout=30#
authentication.param.config.prefix=hadoop.auth.config#
authentication.param.hadoop.auth.config.type=kerberos#
authentication.param.hadoop.auth.config.signature.secret=${ALIAS=AUTH_CONFIG_SIGNATURE_SECRET}
authentication.param.hadoop.auth.config.token.validity=1800#
authentication.param.hadoop.auth.config.cookie.path=/#
authentication.param.hadoop.auth.config.simple.anonymous.allowed=false#
authentication.param.hadoop.auth.config.kerberos.principal=AUTH_CONFIG_KERBEROS_PRINCIPAL#
authentication.param.hadoop.auth.config.kerberos.keytab=AUTH_CONFIG_KERBEROS_KEYTAB#
authentication.param.hadoop.auth.config.kerberos.name.rules=DEFAULT

```



Important: Paste the value as a single line, without line-breaks.

- Add a safety valve name/value pair in Cloudera Manager Knox Configuration, in Knox Gateway Environment Advanced Configuration Snippet (Safety Valve):

```
Name = IDBROKER_KERBEROS_DT_PROXYUSER_BLOCK
```



```
Value = "proxyuser_block": "none"
```

Knox Gateway Environment Advanced Configuration Snippet (Safety Valve)		Knox Gateway Default Group	View as Text
KNOX_GATEWAY_role_env_safety_valve	Key	IDBROKER_KERBEROS_DT_PROXYUSER_BLOCK	
	Value	"proxyuser_block": "none"	

3. Save your changes.
4. Refresh the cluster.
5. Validate with a curl command: `curl -k https://host-10-00-100-100:8443/gateway/admin/api/v1/providerconfig/kerberos-providers`

```
# curl -k https://host-10-00-100-100:8443/gateway/admin/api/v1/providerconfig/kerberos-providers
{
  "providers" : [ {
    "role" : "authentication",
    "name" : "HadoopAuth",
    "enabled" : true,
    "params" : {
      "config.prefix" : "hadoop.auth.config",
      "hadoop.auth.config.cookie.path" : "/",
      "hadoop.auth.config.kerberos.keytab" : "/var/run/cloudera-scm-agent/process/81-knox-KNOX_GATEWAY/knox.keytab",
      "hadoop.auth.config.kerberos.name.rules" : "DEFAULT",
      "hadoop.auth.config.kerberos.principal" : "HTTP/host-10-00-100-100.coe.cloudera.com@CLOUDERA.COM",
      "hadoop.auth.config.signature.secret" : "${ALIAS=AUTH_CONFIG_SIGNATURE_SECRET}",
      "hadoop.auth.config.simple.anonymous.allowed" : "false",
      "hadoop.auth.config.token.validity" : "1800",
      "hadoop.auth.config.type" : "kerberos",
      "proxyuser_block" : "none"
    }
  }, {
    "role" : "identity-assertion",
    "name" : "HadoopGroupProvider",
    "enabled" : true,
    "params" : {
      "CENTRAL_GROUP_CONFIG_PREFIX" : "gateway.group.config."
    }
  }, {
    "role" : "authorization",
    "name" : "XASecurePDPKnox",
    "enabled" : true,
    "params" : { }
  }, {
    "role" : "ha",
    "name" : "HaProvider",
    "enabled" : true,
    "params" : {
      "HBASE" : "maxFailoverAttempts=3;failoverSleep=1000;enabled=true",
      "HIVE" : "maxFailoverAttempts=3;failoverSleep=1000;enabled=true;zookeeperEnsemble=maxFailoverAttempts=3;failoverSleep=1000;enabled=true;zookeeperEnsemble=gb120175161.systems.uk.company:2181,gb120175162.systems.uk.company:2181,gb120175163.systems.uk.company:2181;zookeeperNamespace=hiveserver2",
      "OOZIE" : "maxFailoverAttempts=3;failoverSleep=1000;enabled=true",
      "WEBHCAT" : "maxFailoverAttempts=3;failoverSleep=1000;enabled=true",
```

```

"WEBHDFS" : "maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000;enabled=true"
}
} 1,
"readOnly" : true
}

```

Related Information

[Saving aliases](#)

Management of services for Apache Knox via Cloudera Manager

You can enable or disable known or custom services in Knox proxy via Cloudera Manager.

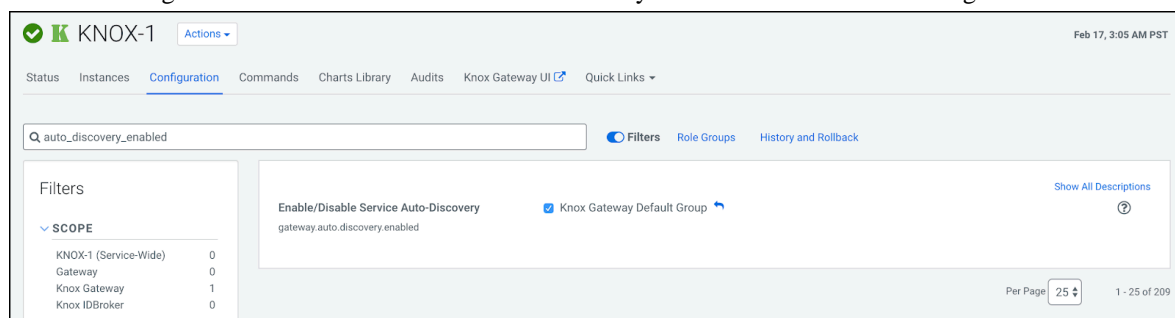
There are two kinds of services in cdp-proxy:

- **Known:** officially-supported Knox services. Cloudera Manager provides and manages all the required service definition files.
- **Custom:** unofficial, tech preview, or community feature Knox services. You must supply the service definition files (service.xml and rewrite.xml) exist in the KNOX_DATA_DIR/services folder. These are not recommended for production environments, and not supported by Cloudera.



Important:

These topologies will be deployed by Cloudera Manager only if Knox's service auto-discovery feature is turned on using the Enable/Disable Service Auto-Discovery checkbox on Cloudera Manager UI:



For a comprehensive list of known services that can be enabled, see “Knox Supported Services Matrix”.

Related Information

[Knox Supported Services Matrix](#)

Enable proxy for a known service in Apache Knox

How to enable auto-discovery for a known service in Knox proxy via Cloudera Manager.

About this task

“Known” services are officially-supported Knox services (like Apache Atlas, Ranger, Solr, etc.) Cloudera Manager provides and manages all the required service definition files.

For the purposes of this example, we add ATLAS and ATLAS UI to cdp-proxy. You can add more services; for a comprehensive list of knoxn services that can be enabled, see “Knox Supported Services Matrix”.



Note: If you use a Knox gateway to access cluster services, you must add the following properties to the Knox configuration:

- `ranger.plugin.hive.use.x-forwarded-for.ipaddress=<true>` (if you want to use the actual client IP address)
- `ranger.plugin.hive.trusted.proxy.ipaddress=<knox.IP.address>`

Procedure

1. From Cloudera Manager Knox Configuration, check the Gateway Auto Discovery (cdp-proxy) - \$Component boxes.

In this example, we enable:

- `gateway_auto_discovery_cdp_proxy_enabled_atlas`
- `gateway_auto_discovery_cdp_proxy_enabled_atlas_ui`

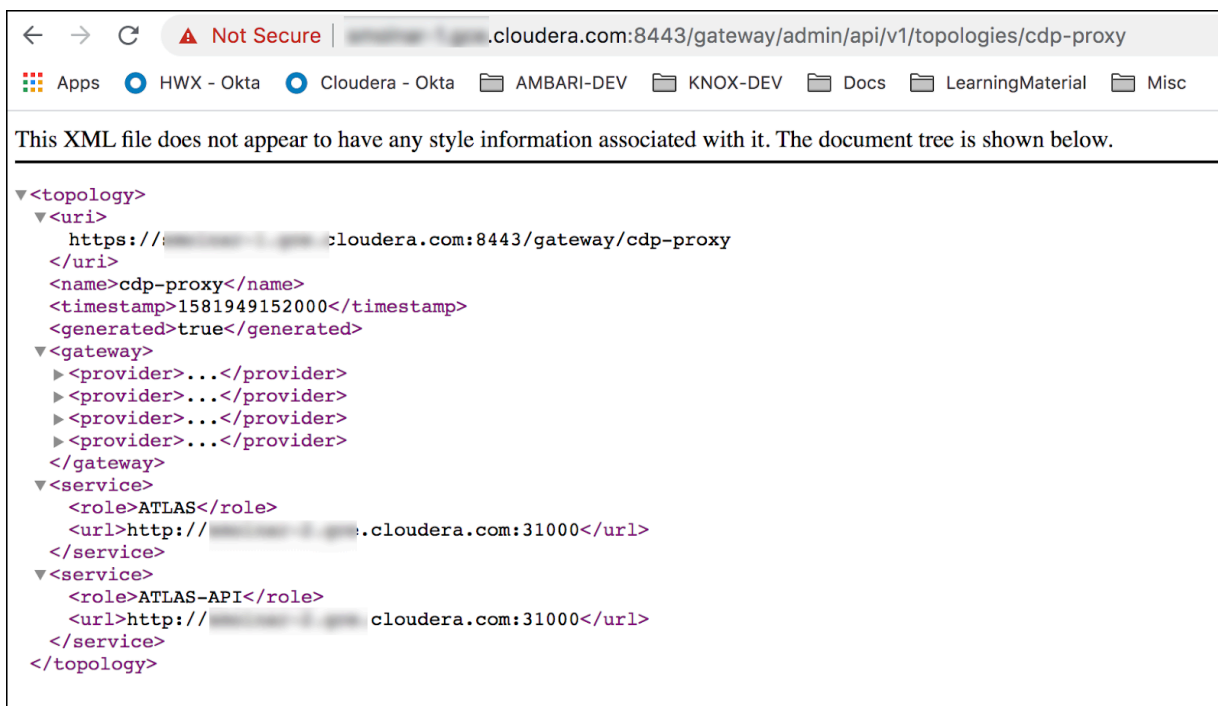
The screenshot shows the Cloudera Manager interface for KNOX-1. The 'Configuration' tab is selected, and a search filter 'auto_discovery_cdp_proxy_enabled_atlas' is applied. The 'Filters' section on the left shows the scope and category. The main content area displays two configuration items: 'Enable Auto Discovery (cdp-proxy) - Atlas API' and 'Enable Auto Discovery (cdp-proxy) - Atlas Web UI', both with checkboxes for 'Knox Gateway Default Group'.

2. Save your changes.
3. The 'Refresh needed' stale configuration indicator appears; click it and wait until the refresh process finishes.

The screenshot shows the 'Stale Configurations' page in Cloudera Manager. It displays a list of stale configurations for the file 'conf/auto-discovery-advanced-configuration-cdp-proxy.properties'. The configurations are listed with their file paths and the status 'KNOX-1(1)'. The configurations are:

- `gateway.auto.discovery.cdp-proxy.enabled.atlas=false`
- `gateway.auto.discovery.cdp-proxy.enabled.atlas-api=false`
- `gateway.auto.discovery.cdp-proxy.enabled.atlas=true`
- `gateway.auto.discovery.cdp-proxy.enabled.atlas-api=true`
- `gateway.auto.discovery.cdp-proxy.enabled.cm-api=false`
- `gateway.auto.discovery.cdp-proxy.enabled.cm-ui=false`
- `gateway.auto.discovery.cdp-proxy.enabled.hbaseui=false`
- `gateway.auto.discovery.cdp-proxy.enabled.hdfsui=false`

4. Validate that ATLAS in cdp-proxy was added by going to the following URL: `https://$KNOX_GATEWAY_HOST:$PORT/$GATEWAY_PATH/admin/api/v1/topologies/cdp-proxy`.



Related Information

[Add custom service parameter to descriptor](#)

[Knox Supported Services Matrix](#)

Disable proxy for a known service in Apache Knox

How to remove auto-discovery for a known service in Knox proxy via Cloudera Manager.

About this task

“Known” services are officially-supported Knox services (like Apache Atlas, Ranger, Solr, etc.) Cloudera Manager provides and manages all the required service definition files.

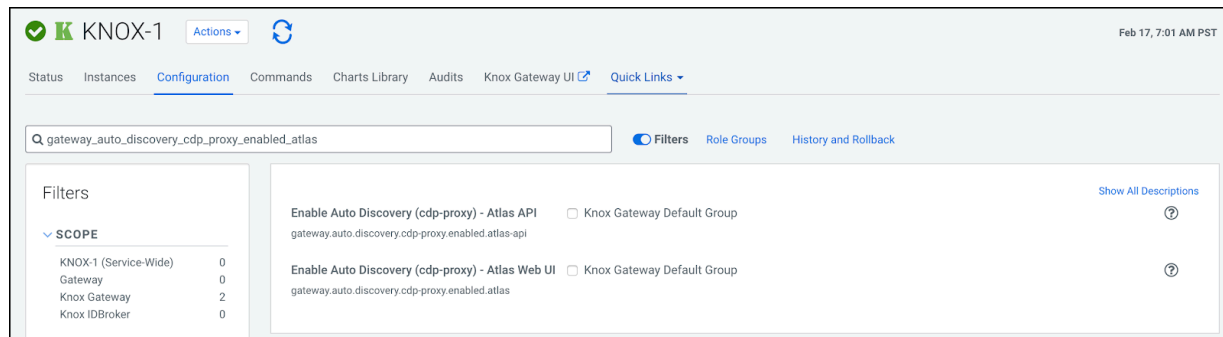
In this example, we are going to remove the previously added ATLAS and ATLAS-UI services from cdp-proxy. We disable the `gateway_auto_discovery_cdp_proxy_enabled_atlas` and `gateway_auto_discovery_cdp_proxy_enabled_atlas_ui` checkboxes on Knox’s Configuration page in CM, save the changes and refresh the cluster.

Procedure

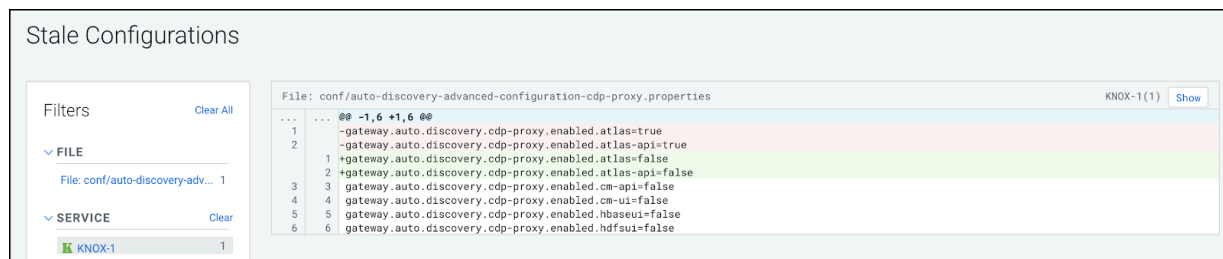
1. From Cloudera Manager Knox Configuration, uncheck the Gateway Auto Discovery (cdp-proxy) - \$Component boxes.

In this example, we disable:

- gateway_auto_discovery_cdp_proxy_enabled_atlas
- gateway_auto_discovery_cdp_proxy_enabled_atlas_ui



2. Save your changes.
3. The 'Refresh needed' stale configuration indicator appears; click it and wait until the refresh process finishes.



4. Validate that custom service got removed by going to the following URL: `http s://$KNOX_GATEWAY_HOST:$PORT/$GATEWAY_PATH/admin/api/v1/topologies/cdp-proxy`.



Add custom service to existing descriptor in Apache Knox Proxy

How to add a custom service to an existing descriptor in Knox proxy using Cloudera Manager.

About this task

“Custom” services are unofficial, tech preview, or community feature Knox services. You must supply the service definition files (service.xml and rewrite.xml) which exist in the KNOX_DATA_DIR/services folder. These are not recommended for production environments, and not supported by Cloudera.

In this example, a custom service (*MY_SERVICE*) is added in cdp-proxy with the following attributes:

- Version : the service’s version, for example, 1.0.0.
- URL: the service URL, for example, https://sampleHost:1234.
- Service parameter: a sample service parameter, for example, myValue.



Important: Adding a custom service only works if you provide the service definition files (service.xml and rewrite.xml) in the KNOX_DATA_DIR/services folder.

To achieve the goals you need to add three new entries with the above-listed parameters in Knox Simplified Topology Management - cdp-proxy. Then you save the changes, refresh the cluster and check if the newly added custom service is available in cdp-proxy.

Procedure

1. From Cloudera Manager Knox Configuration , add the three new entries with the above-listed parameters.

```
MY_SERVICE:version=1.0.0
MY_SERVICE:url=https://sampleHost:1234
MY_SERVICE:customServiceParameter=myValue
```

2. Save your changes.
3. The ‘Refresh needed’ stale configuration indicator appears; click it and wait until the refresh process completes.

4. Validate that MY_SERVICE in cdp-proxy is added by navigating to the following URL: `http://$KNOX_GATEWAY_HOST:$PORT/$GATEWAY_PATH/admin/api/v1/topologies/cdp-proxy`.

Add a custom descriptor to Apache Knox

How to add a custom descriptor to Apache Knox using Cloudera Manager.

About this task

Custom descriptors can be deployed to Apache Knox using Cloudera Manager. These descriptors, combined with referenced provider configurations, are transformed into Knox topologies. Using Cloudera Manager means that these descriptors only ever need to be changed in one place to affect all Knox Gateway instances in the cluster.

Fundamentally, descriptors contain the declaration of services to proxy and a reference to provider configuration defining how authentication and authorization for those proxied services should be handled. A descriptor also may similarly declare Knox applications as topologies do.

Service declarations consist of at least the name of the service being proxied. They optionally include one or more endpoint URLs and one or more service-specific parameters.

Descriptors optionally include discovery information, allowing Knox to dynamically discover the endpoint URLs for the declared services.

Procedure

1. Define the descriptor contents:

- a) From Cloudera Manager Knox Configuration, add a new entry in Knox Gateway Advanced Configuration Snippet (Safety Valve) for `conf/cdp-resources.xml_role_safety_valve`.
- b) Name the topology, specify the providerConfigRef, and enumerate the services and associated service URLs. Optional service details include version (E.G., `HIVE:version=0.13.0`) and service parameters (E.G., `HIVE:httpclient.connectionTimeout=5m`)

Static URL Example (HIVE and WEBHDFS with PAM authentication)

- Name=my-custom-topology
- Value=

```
providerConfigRef=pam#
HIVE:url=https://hive-host-1:10001/cliservice#
WEBHDFS:url=https://hdfs-host-1:20470/webhdfs#
WEBHDFS:url=https://hdfs-host-2:20470/webhdfs
```

Discovery Example (HIVE and WEBHDFS with PAM authentication)



Note: If the CDP cluster is not enabled with Auto-TLS, then you must add the Cloudera Manager certificate to the Knox truststore and restart the Knox service.

- Name=my-discoverable-topology
- Value=

```
discoveryType=ClouderaManager#
discoveryAddress=https://cm-host:7183#
cluster=Cluster 1#
providerConfigRef=pam#
HIVE:#
WEBHDFS:
```

2. Save the changes.

3. Refresh the Knox instances' configuration: the Refresh needed stale configuration indicator appears; click it and wait until the refresh process completes.
4. Validate:
Using the Knox Admin UI (https://KNOX_GATEWAY_HOST:PORT/GATEWAY_PATH/gateway/manager/admin-ui/), navigate to the Topologies, and verify that your topology was generated with the services and URLs you specified.

Management of Service Parameters for Apache Knox via Cloudera Manager

You can add, modify, or remove custom service parameters in Knox proxy via Cloudera Manager.

Add custom service parameter to descriptor

How to add a custom service parameter to a descriptor using Cloudera Manager..

Before you begin

The descriptor you wish to add a custom service parameter to must be enabled. See “Add a known service to cdp-proxy”.

About this task

In this example, you are adding a custom service parameter with a custom value (myCustomServiceParameter=myValue) to ATLAS in cdp-proxy.



Note: If you use a Knox gateway to access cluster services, you must add the following properties to the Knox configuration:

- ranger.plugin.hive.use.x-forwarded-for.ipaddress=<true> (if you want to use the actual client IP address)
- ranger.plugin.hive.trusted.proxy.ipaddress=<knox.IP.address>

Procedure

1. From Cloudera Manager Knox Configuration, add a new line in the Knox Simplified Topology Management - cdp-proxy panel in the following format:
`$$SERVICE_NAME[:$PARAMETER_NAME=$PARAMETER_VALUE].`
 ATLAS:myCustomServiceParameter=myValue

The url and version parameter names are preserved keywords to set the given service's URL and version. Valid declarations:

```
ATLAS:url=http://localhost:123
ATLAS:version:3.0.0
```


Actions
Oct 19, 12:30 AM PDT

Status
Instances
Configuration
Commands
Charts Library
Audits
Knox Gateway Home
Quick Links

Filters
Role Groups
History and Rollback

Filters

SCOPE

KNOX-1 (Service-Wide) 0

Gateway 0

Knox Gateway 2

Knox IDBroker 0

CATEGORY

Advanced 0

Logs 0

Main 2

Monitoring 0

Performance 0

Knox Simplified Topology Management - cdp-proxy

cdp-proxy

providerConfigRef=sso

ATLAS.myCustomServiceParameter=myValue

Knox Gateway Default Group

providerConfigRef=sso

ATLAS.myCustomServiceParameter=myValue

Knox Simplified Topology Management - cdp-proxy-api

cdp-proxy-api

providerConfigRef=pam

Knox Gateway Default Group

providerConfigRef=pam

Show All Descriptions

Per Page 25

1 - 25 of 220

2. Save your changes.
3. The 'Refresh needed' stale configuration indicator appears; click it and wait until the refresh process completes.

4. Validate that ATLAS in cdp-proxy got updated with the new service parameter by navigating to the following URL: `https://$KNOX_GATEWAY_HOST:$PORT/$GATEWAY_PATH/admin/api/v1/topologies/cdp-proxy`.

The screenshot shows a web browser window with the address bar displaying `cloudera.com:8443/gateway/admin/api/v1/topologies/cdp-proxy`. The page content indicates that the XML file does not have any style information, and the document tree is shown below. The XML tree is expanded to show the following structure:

```
<?xml version="1.0"?>
<topology>
  <uri>https://[redacted].cloudera.com:8443/gateway/cdp-proxy</uri>
  <name>cdp-proxy</name>
  <timestamp>1603092694000</timestamp>
  <generated>true</generated>
  <gateway>
    ...
  </gateway>
  <service>
    <role>ATLAS</role>
    <param>
      <name>myCustomServiceParameter</name>
      <value>myValue</value>
    </param>
  </service>
  <service>
    ...
  </service>
  <service>
    ...
  </service>
  <service>
    ...
  </service>
  <service>
    ...
  </service>
  <service>
    ...
  </service>
</topology>
```

Modify custom service parameter in descriptor

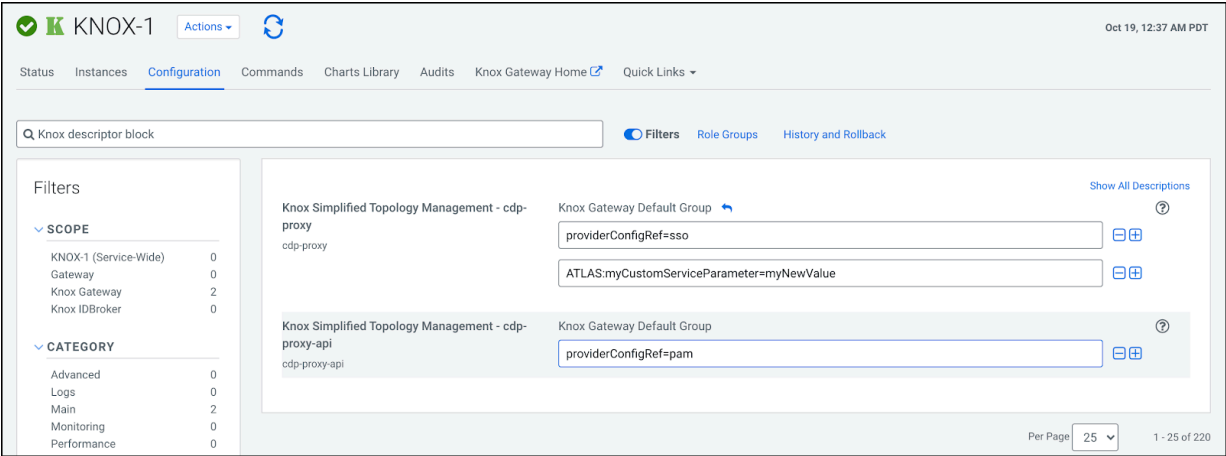
How to edit a custom service parameter in a descriptor using Cloudera Manager.

About this task

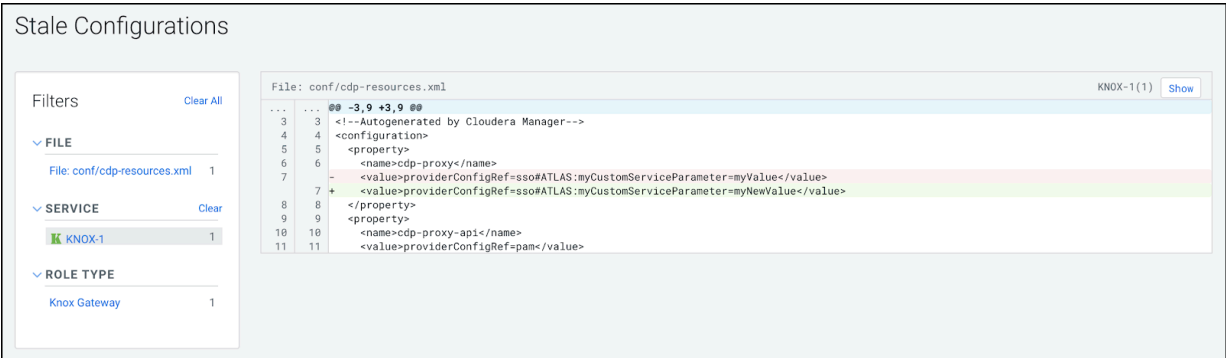
In this sample, you are going to update a previously entered service parameter - `myCustomServiceParameter=myValue` to `myNewValue`- for ATLAS in cdp-proxy. You change that entry, save the changes, and refresh the cluster.

Procedure

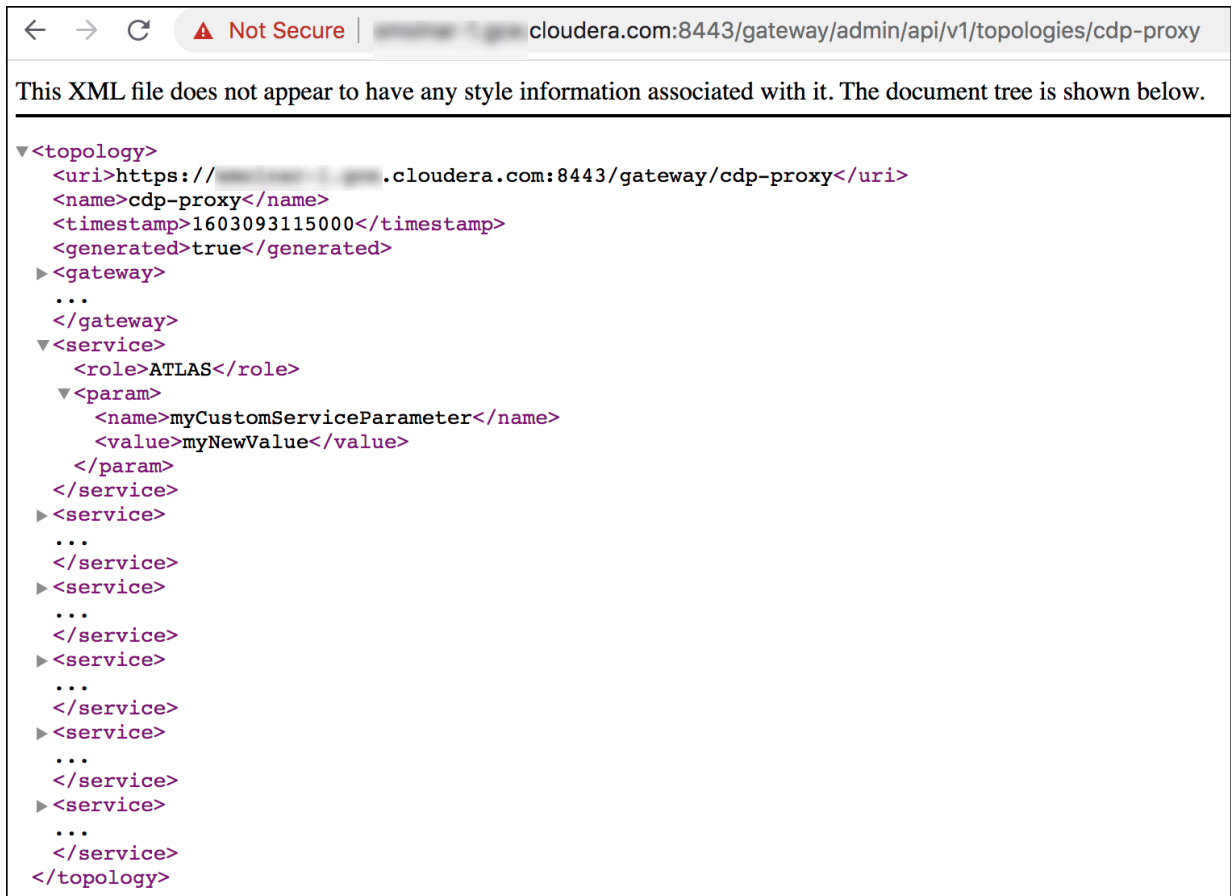
1. From Cloudera Manager Knox Configuration , change the service parameter in the Knox Simplified Topology Management - cdp-proxy panel.
Change ATLAS:myCustomServiceParameter=myValue to ATLAS:myCustomServiceParameter=myNewValue



2. Save your changes.
3. The 'Refresh needed' stale configuration indicator appears; click it and wait until the refresh process completes.



4. Validate that custom service parameter got updated with the changes by navigating to the following URL: `http s://$KNOX_GATEWAY_HOST:$PORT/$GATEWAY_PATH/admin/api/v1/topologies/cdp-proxy`.



Remove custom service parameter from descriptor

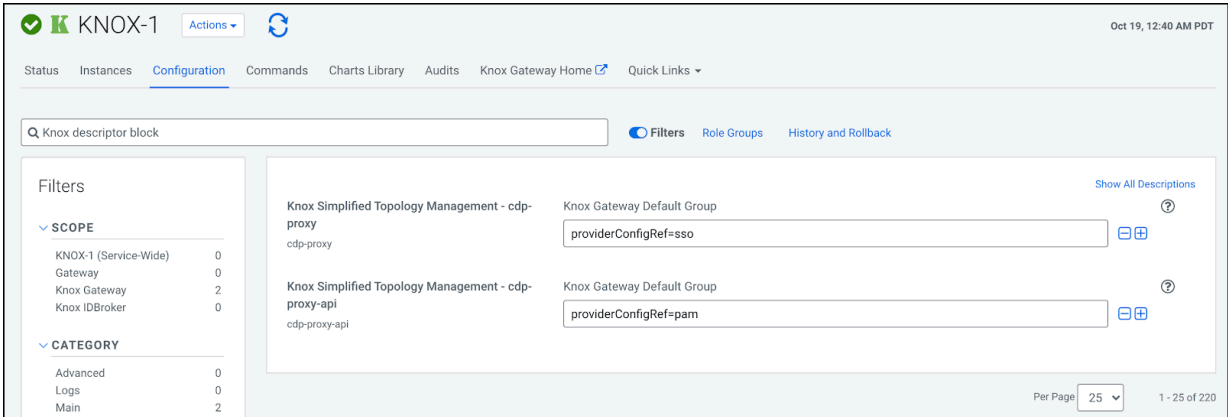
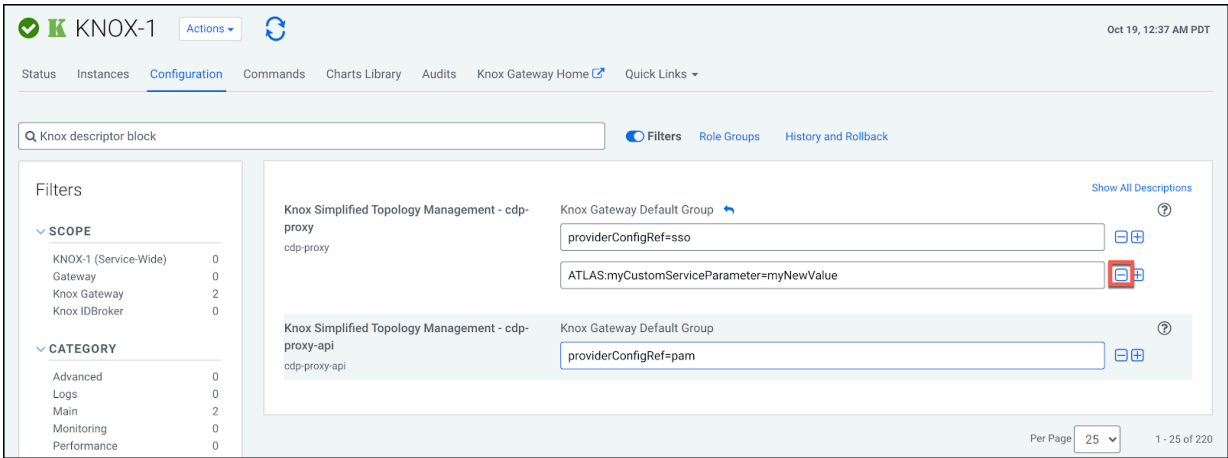
How to remove a custom service parameter from a descriptor using Cloudera Manager.

About this task

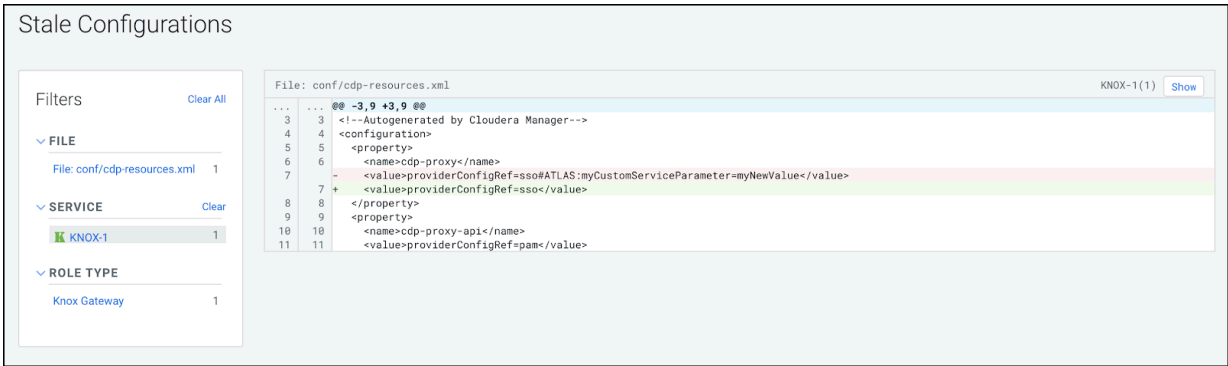
In this sample, you are going to remove a previously entered service parameter - `myCustomServiceParameter=myNewValue` - from ATLAS in `cdp-proxy`. You remove that entry, save the changes, and refresh the cluster.

Procedure

1. From Cloudera Manager Knox Configuration , remove the service parameter in the Knox Simplified Management - cdp-proxy panel.
- Click the minus (–) sign next to ATLAS:myCustomServiceParameter=myNewValue.



2. Save your changes.
3. The 'Refresh needed' stale configuration indicator appears; click it and wait until the refresh process completes.



4. Validate that custom service parameter got removed with the changes by navigating to the following URL: `https://$KNOX_GATEWAY_HOST:$PORT/$GATEWAY_PATH/admin/api/v1/topologies/cdp-proxy`.



The screenshot shows a web browser window with the address bar displaying `https://[redacted].gce.cloudera.com:8443/gateway/admin/api/v1/topologies/cdp-proxy`. The browser's developer tools are open, showing the XML document tree. The XML structure is as follows:

```
<?xml version="1.0"?>
<topology>
  <uri>https://[redacted].gce.cloudera.com:8443/gateway/cdp-proxy</uri>
  <name>cdp-proxy</name>
  <timestamp>1581950909000</timestamp>
  <generated>true</generated>
  <gateway>...</gateway>
  <service>
    <role>ATLAS</role>
    <url>http://[redacted].cloudera.com:31000</url>
  </service>
  <service>
    <role>ATLAS-API</role>
    <url>http://[redacted].cloudera.com:31000</url>
  </service>
</topology>
```