Cloudera Runtime 7.1.1

Apache Knox Authentication

Date published: 2020-04-28
Date modified: 2020-10-26

https://docs.cloudera.com/

Legal Notice

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

Using the Apache Knox Gateway UI..8

Proxy Cloudera Manager through Apache Knox................................ 10

Installing Apache Knox...10

Apache Knox Install Role Parameters.. 12

Managing Knox shared providers in Cloudera Manager.................... 13

Configure Apache Knox authentication for PAM... 14

Configure Apache Knox authentication for AD/LDAP......................................15

Managing existing Apache Knox shared providers.. 17

Add a new shared provider configuration...18

Add a new provider in an existing provider configuration.................................19

Modify a provider in an existing provider configuration.................................. 21

Disable a provider in an existing provider configuration.................................. 22

Saving aliases...24

Configure Kerberos authentication in Apache Knox shared providers............ 26

Managing services for Apache Knox via Cloudera Manager........................... 28

Enable proxy for a known service in Apache Knox... 29

Disable proxy for a known service in Apache Knox.. 30

Add a custom service to Apache Knox Proxy..31

Add a custom topology in the deployed Apache Knox Gateway......................33

Managing Service Parameters for Apache Knox via Cloudera Manager......... 35

Add a custom service parameter to a known service.. 35

Modify a custom service parameter..37

Remove a custom service parameter...39

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 Datacenter 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.
- Authorization: Ranger

  OS Security: Data Encryption and HDFS

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 Gateways 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		
NameNode	#	#
NiFi	#	#
NiFi Registry	#	#
Oozie API	#	#
Oozie UI	#	
Phoenix (aka Avatica)		#

Component	UI Proxy (with SSO)	API Proxy
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.

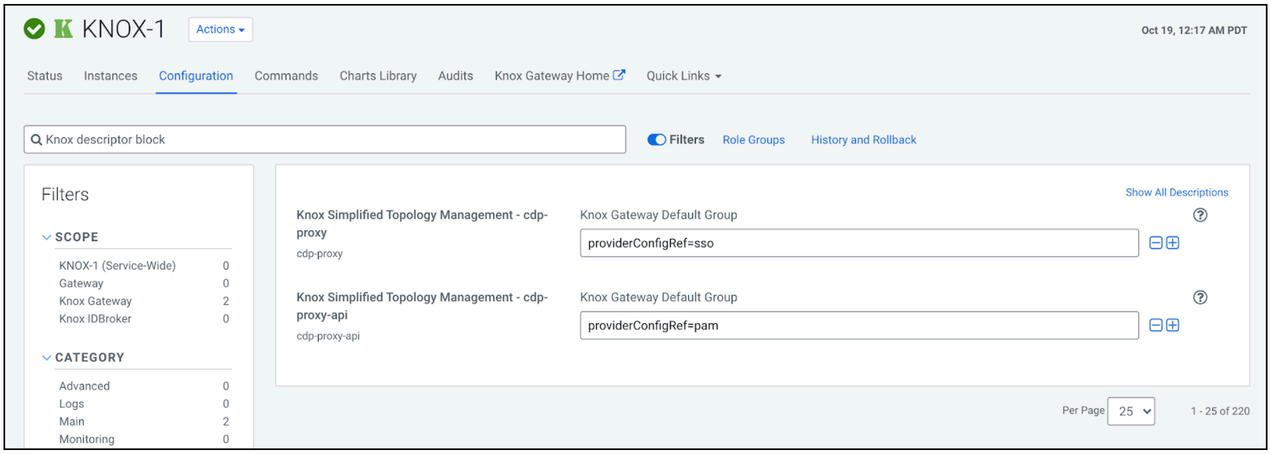
Knox Topology Management in Cloudera Manager

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



Knox Simplified Topology Management - cdp-proxy
cdp-proxy
Knox Gateway Default Group
providerConfigRef=sso

Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
Knox Gateway Default Group
providerConfigRef=pam

- 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

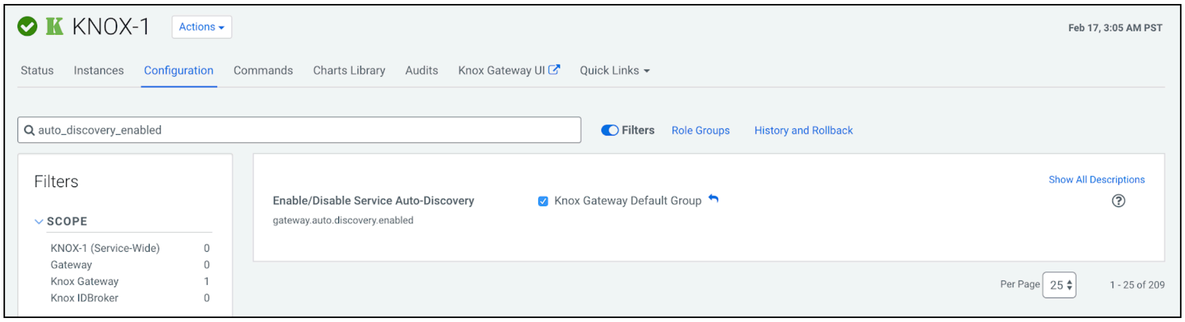
You can enable or disable known or custom services in Knox proxy via Cloudera Manager.

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) 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:



Enable/Disable Service Auto-Discovery
gateway.auto.discovery.enabled
☑ Knox Gateway Default Group

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.

Using the Apache Knox Gateway UI

Knox Proxy can be configured via the Knox Gateway 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.

Before you begin

When logging into the Gateway UI, Knox is expecting a user that can log into the operating system.

About this task

Cloudera Manager creates the majority of the topologies you need. You can use the Knox Gateway UI to create additional topologies or modify existing ones.

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). You can also manually set up Knox Proxy by manually configuring individual topology files.

Before you begin

- Cloudera Manager must be installed.

Procedure

1. Navigate from Cloudera Manager to the Knox Gateway UI: Cloudera Manager Clusters Knox Knox Gateway Home General Proxy Information Admin UI URL .
   The Knox Gateway UI opens, e.g. https://dw-weekly.field.Cloudera.com:8443/gateway/manager/admin-ui.
2. Login to the Gateway UI.

3. Create a Provider Configuration:
   a) From the Gateway UI homepage, click Provider Configurations + .

      The **Create a New Provider Configuration** wizard opens.
   b) Name the provider configuration: for example, CDP_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 Cloudera.
      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.
   a) Create a new descriptor. From the Gateway 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.Cloudera.com:8080
Cluster	dwweekly
Username	admin
Password alias	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 Gateway UI homepage Topologies <topology name, e.g. devcluster> .

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 Cloudera Data Platform Data Center installation process.

About this task

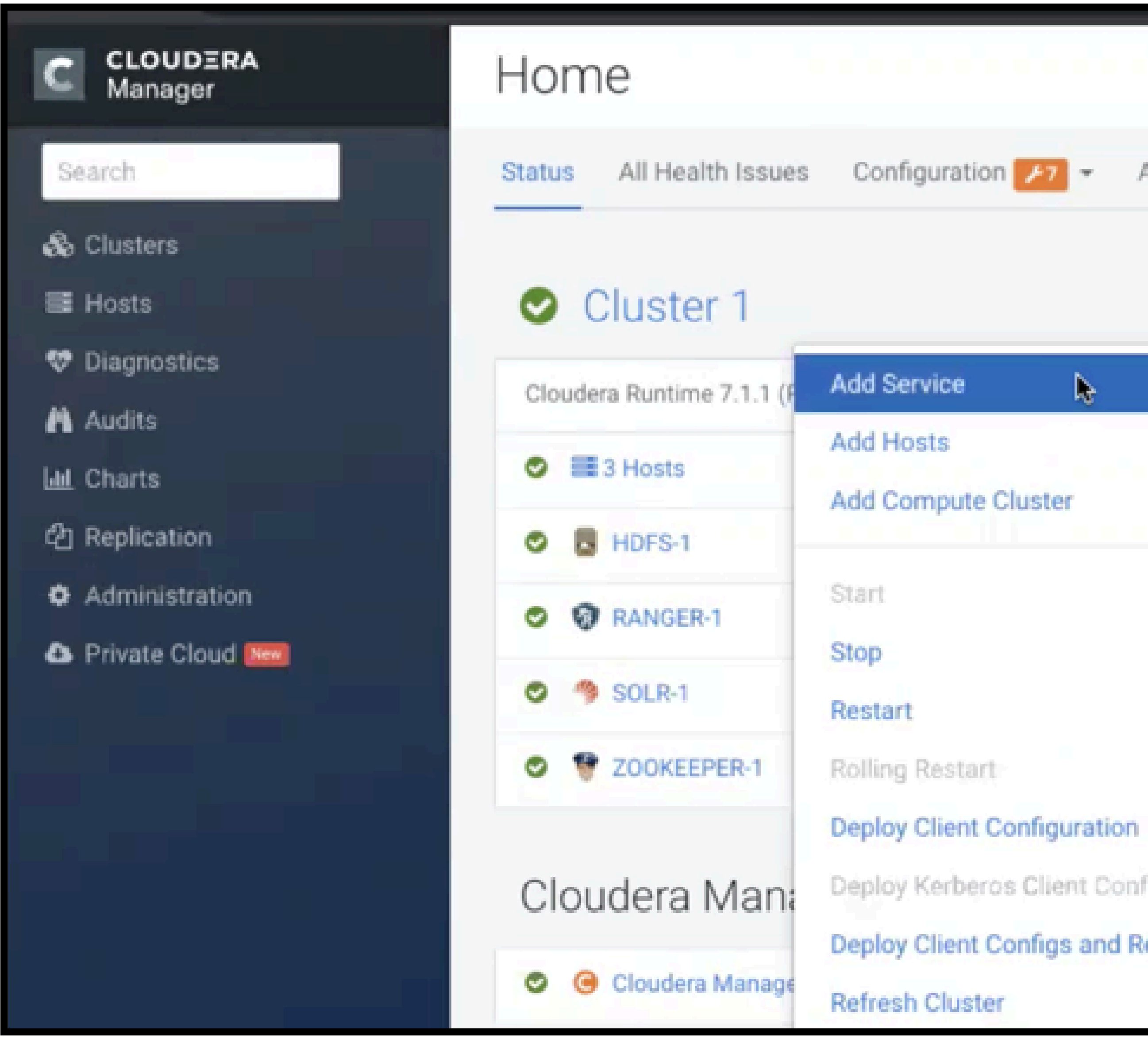
Apache Knox is an application gateway for interacting with the REST APIs and UIs. The Knox Gateway provides a single access point for all REST and HTTP interactions in your Cloudera Data Platform cluster.

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
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
kerberos.auth.enabled*	Yes	Boolean	false
ranger_knox_plugin_hdfs_audit_directory	No	Text	${ranger_base_audit_url}/knox
autorestart_on_stop	No	Boolean	false
knox_pam_realm_service	No	Text	login
save_alias_command_input_password	No	Text	-

Knox Gateway role parameters

Table 3: Required parameters for Knox Gateway role

Name	In Wizard	Type	Default Value
gateway_master_secret	Yes	Password	-
gateway_conf_dir	Yes	Path	/var/lib/knox/gateway/conf

Name	In Wizard	Type	Default Value
gateway_data_dir	Yes	Path	/var/lib/knox/gateway/data
gateway_port	No	Port	8443
gateway_path	No	Text	gateway
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.

Managing 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.

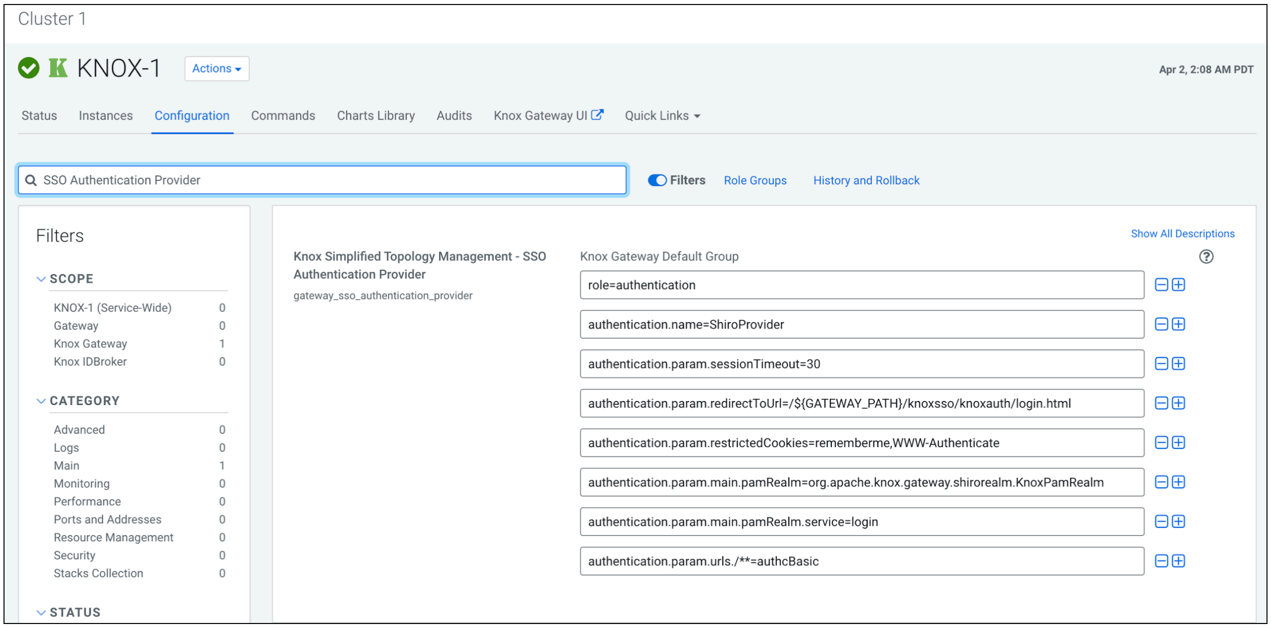
Configure Apache Knox authentication for PAM

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

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}/knoxsso/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
```



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.redirectToUrl=/${GATEWAY_PATH}/knoxsso/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 goes directly into 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.shirorealm.KnoxPamRealm
authentication.param.main.pamRealm.service=login
```

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

Configure Apache Knox authentication for AD/LDAP

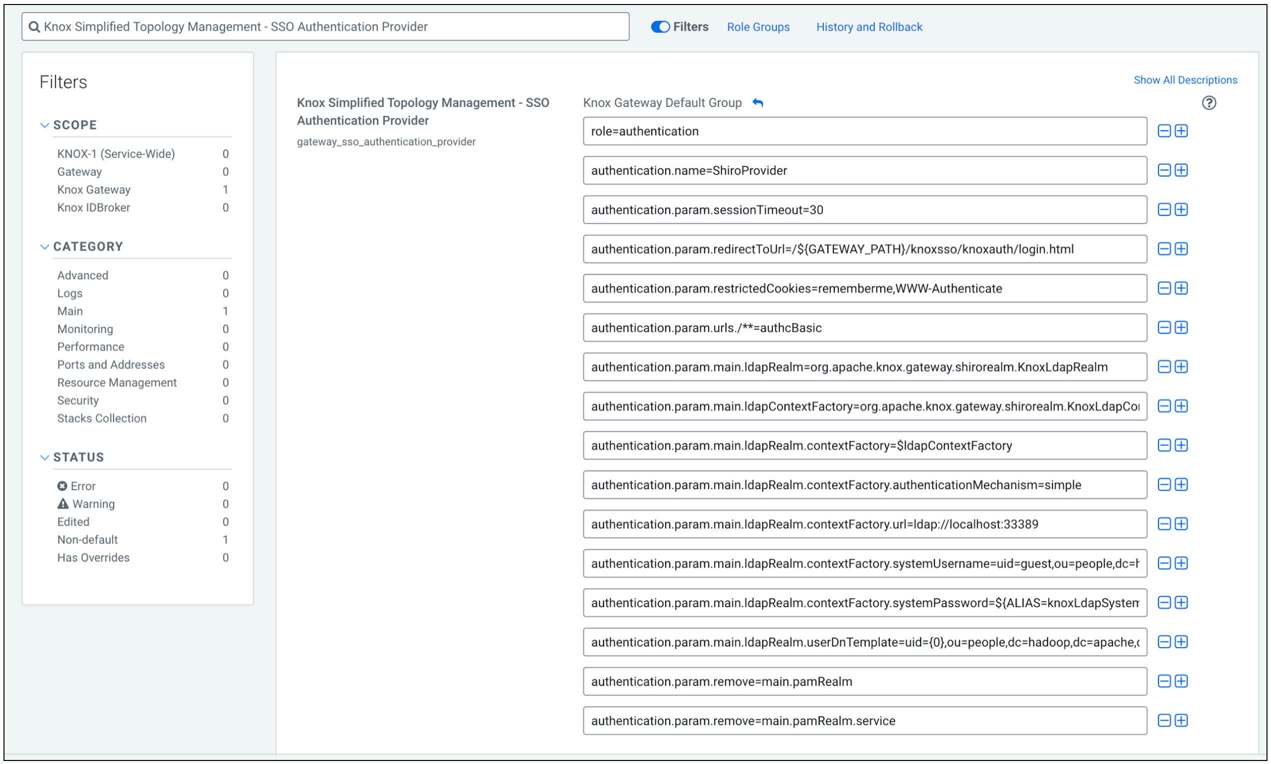
Knox authentication configurations for LDAP and AD in Cloudera Manager.

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=/${GATEWAY_PATH}/knoxsso/knoxauth/login.html
authentication.param.restrictedCookies=rememberme,WWW-Authenticate
authentication.param.urls./**=authcBasic
authentication.param.main.ldapRealm=org.apache.knox.gateway.shirorealm.KnoxLdapRealm
authentication.param.main.ldapContextFactory=org.apache.knox.gateway.shirorealm.KnoxLdapContextFactory
authentication.param.main.ldapRealm.contextFactory=$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=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
```

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.



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.redirectToUrl=/${GATEWAY_PATH}/knoxsso/knoxauth/login.html
authentication.param.restrictedCookies=rememberme,WWW-Authenticate
authentication.param.urls./**=authcBasic
authentication.param.main.ldapRealm=org.apache.knox.gateway.shirorealm.KnoxLdapRealm
authentication.param.main.ldapContextFactory=org.apache.knox.gateway.shirorealm.KnoxLdapCo
authentication.param.main.ldapRealm.contextFactory=$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=h
authentication.param.main.ldapRealm.contextFactory.systemPassword=${ALIAS=knoxLdapSystem
authentication.param.main.ldapRealm.userDnTemplate=uid={0},ou=people,dc=hadoop,dc=apache,
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/ad
min/api/v1/providerconfig/knoxsso'
...
  }, {
    "role" : "authentication",
    "name" : "ShiroProvider",
    "enabled" : true,
    "params" : {
      "main.ldapContextFactory" : "org.apache.knox.gateway.shirorealm.Knox
LdapContextFactory",
      "main.ldapRealm" : "org.apache.hadoop.gateway.shirorealm.KnoxLdapReal
m",
      "main.ldapRealm.contextFactory" : "$ldapContextFactory",
      "main.ldapRealm.contextFactory.authenticationMechanism" : "simple",
      "main.ldapRealm.contextFactory.systemPassword" : "${ALIAS=knoxLdapSys
temPassword}",
      "main.ldapRealm.contextFactory.systemUsername" : "uid=guest,ou=peop
le,dc=hadoop,dc=apache,dc=org",
      "main.ldapRealm.contextFactory.url" : "ldap://localhost:33389",
      "main.ldapRealm.userDnTemplate" : "uid={0},ou=people,dc=hadoop,dc=ap
ache,dc=org",
      "redirectToUrl" : "/${GATEWAY_PATH}/knoxsso/knoxauth/login.html",
      "restrictedCookies" : "rememberme,WWW-Authenticate",
      "sessionTimeout" : "30",
      "urls./**" : "authcBasic"
    }
```

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.shirorealm.Kno
xLdapRealm
authentication.param.main.ldapContextFactory=org.apache.knox.gateway.shir
orealm.KnoxLdapContextFactory
authentication.param.main.ldapRealm.contextFactory=$ldapContextFactory
authentication.param.main.ldapRealm.contextFactory.authenticationMechanism=
simple
authentication.param.main.ldapRealm.contextFactory.url=ldap://localhost:3
3389
authentication.param.main.ldapRealm.contextFactory.systemUsername=uid=guest
,ou=people,dc=hadoop,dc=apache,dc=org
authentication.param.main.ldapRealm.contextFactory.systemPassword=${ALIAS=
knoxLdapSystemPassword}
authentication.param.main.ldapRealm.userDnTemplate=uid={0},ou=people,dc=hado
op,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.

Managing 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 fo 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 c onf/cdp-resources.xml by implementing the following language elements:

- The key of a new entry should be like this: providerConfigs: providerConfig_1 [,providerConfig_2,..,provider Config_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 shared provider configuration

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

About this task

It is possible that you add a brand new shared provider configuration. In this example you will see how to create test Providers with the following providers set:

- authentication: ShiroProvider (LDAP) or PAM
- identity-assertion: Default
- authorization: Ranger (XASecurePDPKnox)

This particular authorization provider is set as follows (in its JSON descriptor):

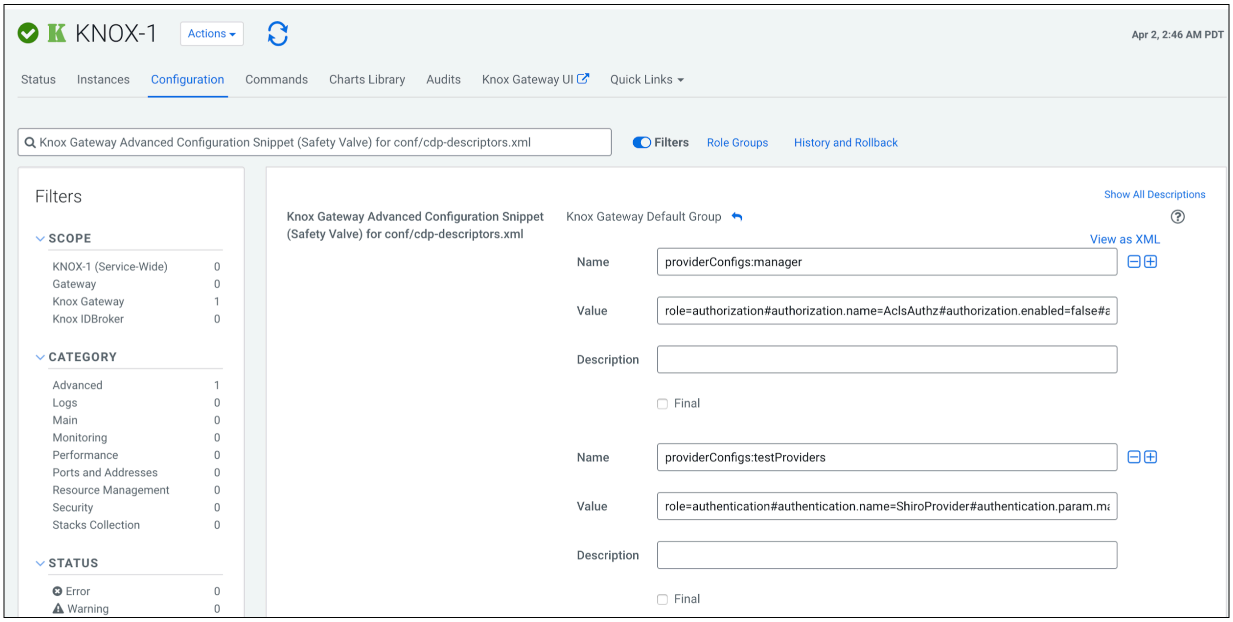
```
{
         "role": "authorization",
         "name": "AclsAuthz",
         "enabled": "true",
         "params": {
            "knox.acl.mode": "OR",
            "knox.acl": "KNOX_ADMIN_USERS;KNOX_ADMIN_GROUPS;*"
         }
      }
```

Procedure

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

   - name = providerConfigs:testProviders
   - value = role=authentication#authentication.name=ShiroProvider#authentication.param.main.pamRealm=org .apache.knox.gateway.shirorealm.KnoxPamRealm#authentication.param.main.pamRealm.service=login#ro

le=identity-assertion#identity-assertion.name=Default#role=authorization#authorization.name=XASecure
PDPKnox



Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml

Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml — Knox Gateway Default Group

Name: providerConfigs:manager

Value: role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#a

Description:

□ Final

Name: providerConfigs:testProviders

Value: role=authentication#authentication.name=ShiroProvider#authentication.param.ma

Description:

□ Final

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/testProviders'
{
  "providers" : [ {
    "role" : "authentication",
    "name" : "ShiroProvider",
    "enabled" : true,
    "params" : {
      "main.pamRealm" : "org.apache.knox.gateway.shirorealm.KnoxPamRealm",
      "main.pamRealm.service" : "login"
    }
  }, {
    "role" : "identity-assertion",
    "name" : "Default",
    "enabled" : true,
    "params" : { }
  }, {
    "role" : "authorization",
    "name" : "XASecurePDPKnox",
    "enabled" : true,
    "params" : { }
  } ]
}
```

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

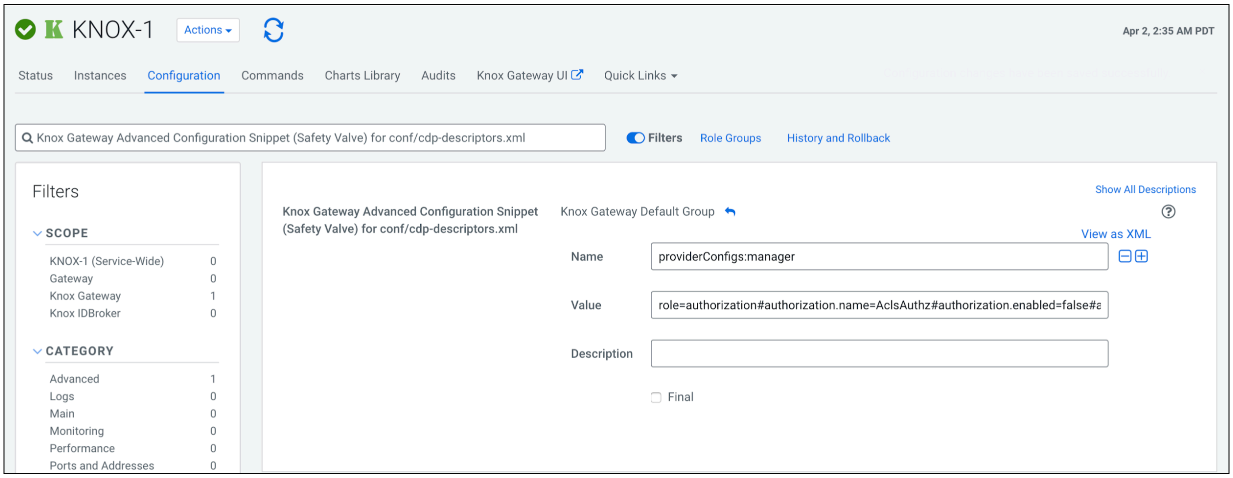
In this example you will see how to add a new HA provider (this time only the ATLAS service will be configured for high availability) in the manager shared provider configuration . This particular authorization provider is set as follows (in its JSON descriptor):

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

Procedure

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=myTestUser;KNOX_ADMIN_GROUPS;*#authorization.param.knox.acl.mode=OR#role=ha#ha.name=HaProvider#ha.param.ATLAS=enabled=true;maxFailoverAttempts=3;failoverSleep=1000;maxRetr

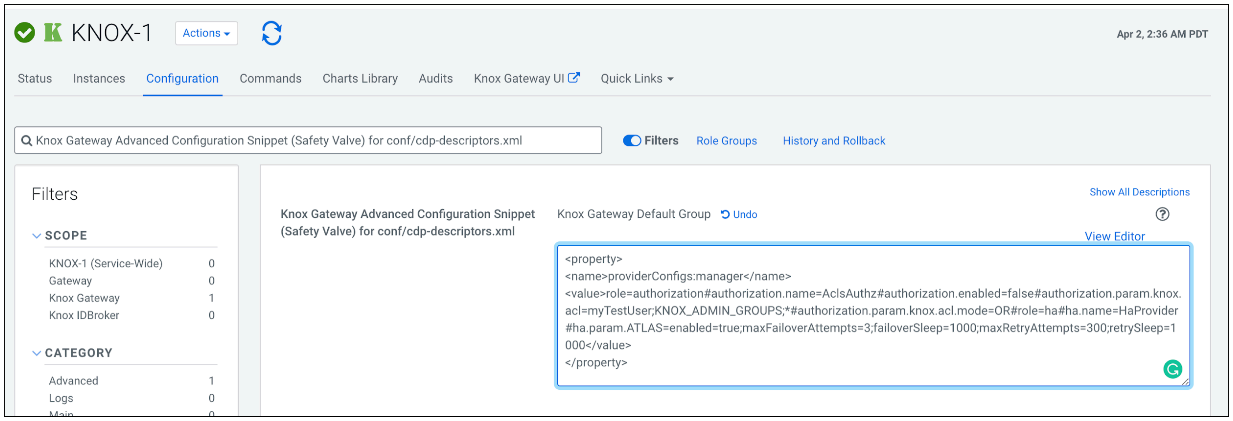


Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml — Knox Gateway Default Group

Name: providerConfigs:manager

Value: role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#a

Description:



Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml — Knox Gateway Default Group

```
<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=1
000</value>
</property>
```

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;m
axRetryAttempts=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

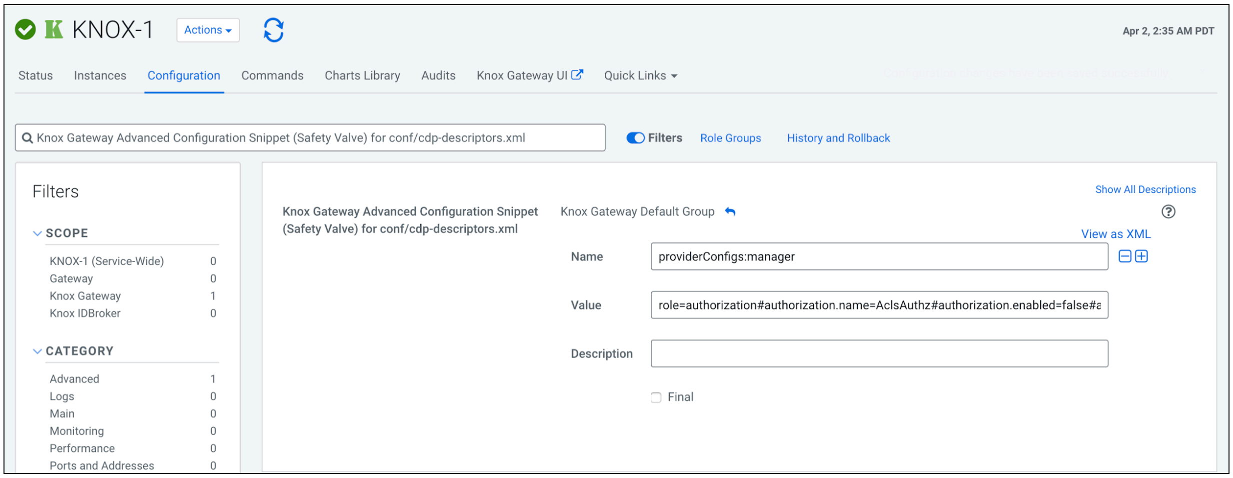
In this example you will see how to modify the authorization provider in the manager shared provider configuration. This particular authorization provider is set as follows (in its JSON descriptor):

```
{
         "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-resources.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

    Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml

Filters	
SCOPE	
KNOX-1 (Service-Wide)	0
Gateway	0
Knox Gateway	1
Knox IDBroker	0
CATEGORY	
Advanced	1
Logs	0
Main	0
Monitoring	0
Performance	0
Ports and Addresses	0

    Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-descriptors.xml — Knox Gateway Default Group

    Name: providerConfigs:manager

    Value: role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#a

    Description:

    □ Final

    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;m
axRetryAttempts=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.

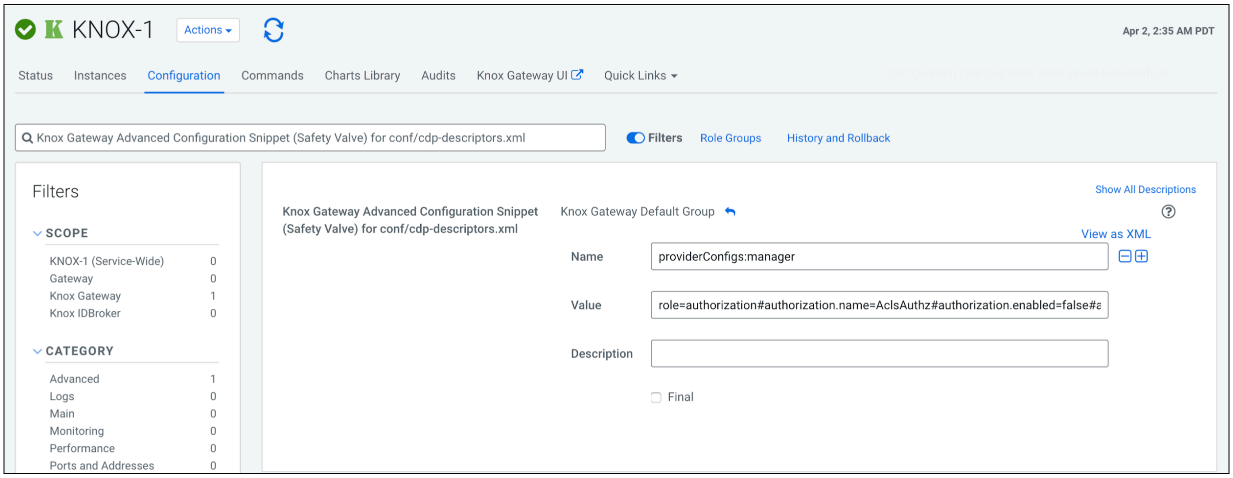
About this task

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-descriptors.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



Name: providerConfigs:manager

Value: role=authorization#authorization.name=AclsAuthz#authorization.enabled=false#a

Description:

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;m
axRetryAttempts=300;retrySleep=1000"
    }
  } ]
}
```

What to do next

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

Saving aliases

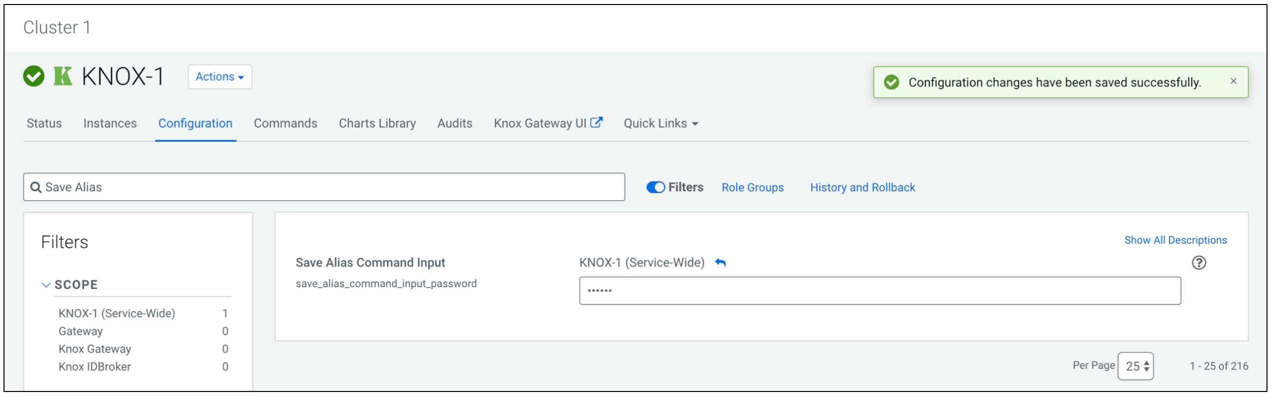
There is a new command available for the Knox Gateway role which allows end-users to save an alias=password pair to an arbitrary number of topologies on each host where an instance of the Knox Gateway is installed without the need of running the Knox CLI tool manually.

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 Actions ▾

Configuration changes have been saved successfully. ×

Status | Instances | Configuration | Commands | Charts Library | Audits | Knox Gateway UI | Quick Links ▾

Save Alias

Filters | Role Groups | History and Rollback

Filters

SCOPE

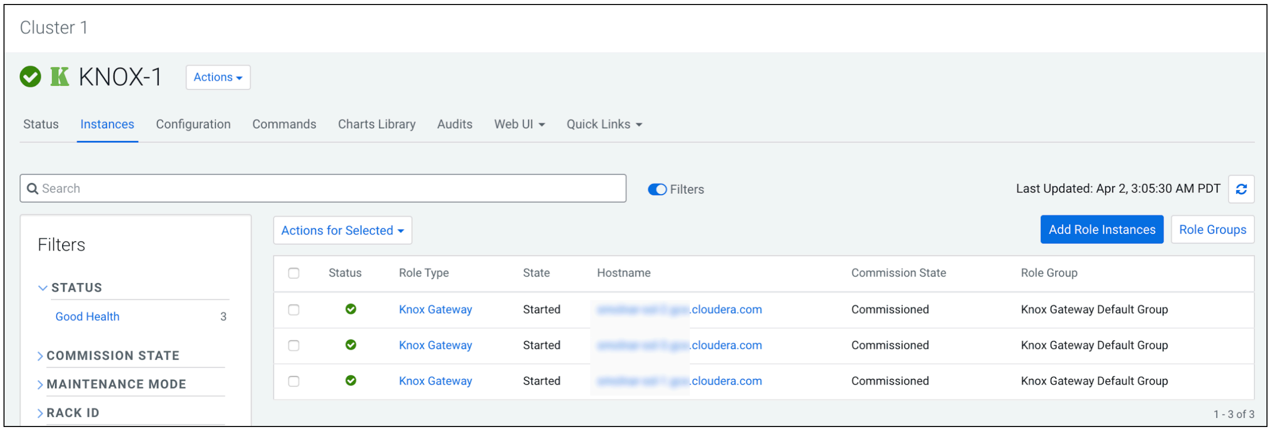
KNOX-1 (Service-Wide)	1
Gateway	0
Knox Gateway	0
Knox IDBroker	0

Show All Descriptions

Save Alias Command Input
save_alias_command_input_password

KNOX-1 (Service-Wide)
••••••

Per Page 25 | 1 - 25 of 216



Cluster 1

KNOX-1 Actions ▾

Status | Instances | Configuration | Commands | Charts Library | Audits | Web UI ▾ | Quick Links ▾

Search

Filters

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

Actions for Selected ▾

Add Role Instances | Role Groups

Filters

STATUS

Good Health 3

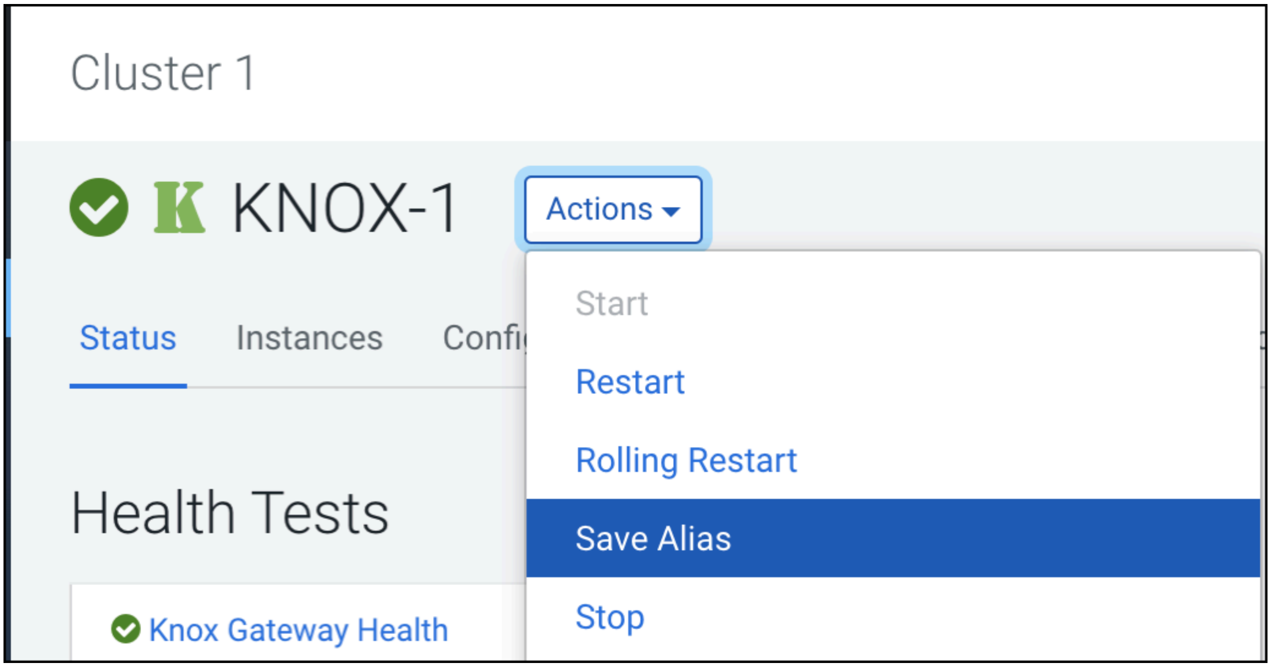
COMMISSION STATE

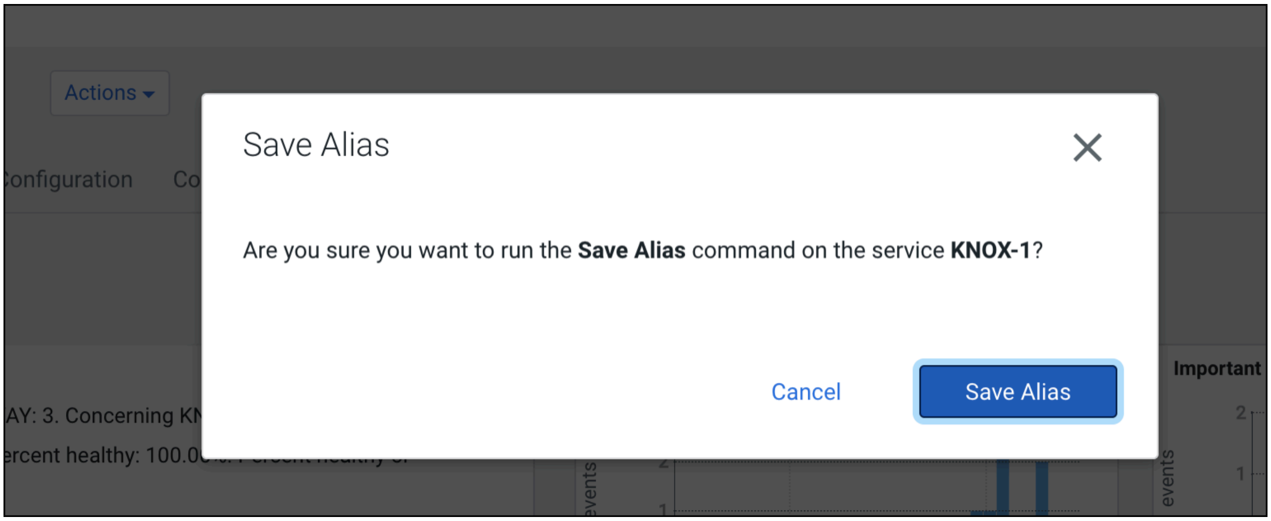
MAINTENANCE MODE

RACK ID

□	Status	Role Type	State	Hostname	Commission State	Role Group
□	✓	Knox Gateway	Started	[redacted].cloudera.com	Commissioned	Knox Gateway Default Group
□	✓	Knox Gateway	Started	[redacted].cloudera.com	Commissioned	Knox Gateway Default Group
□	✓	Knox Gateway	Started	[redacted].cloudera.com	Commissioned	Knox Gateway Default Group

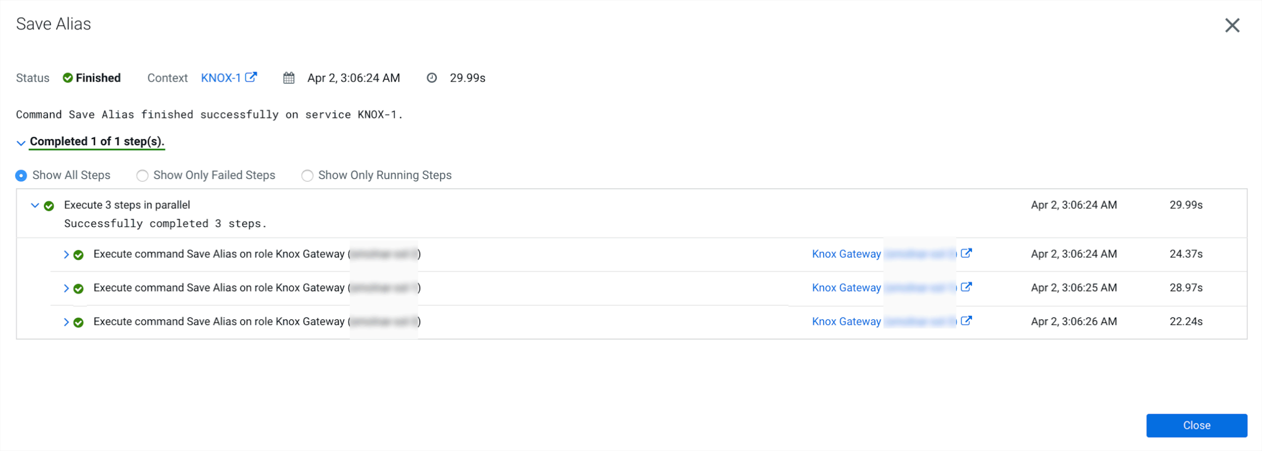
1 - 3 of 3





Save Alias

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


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

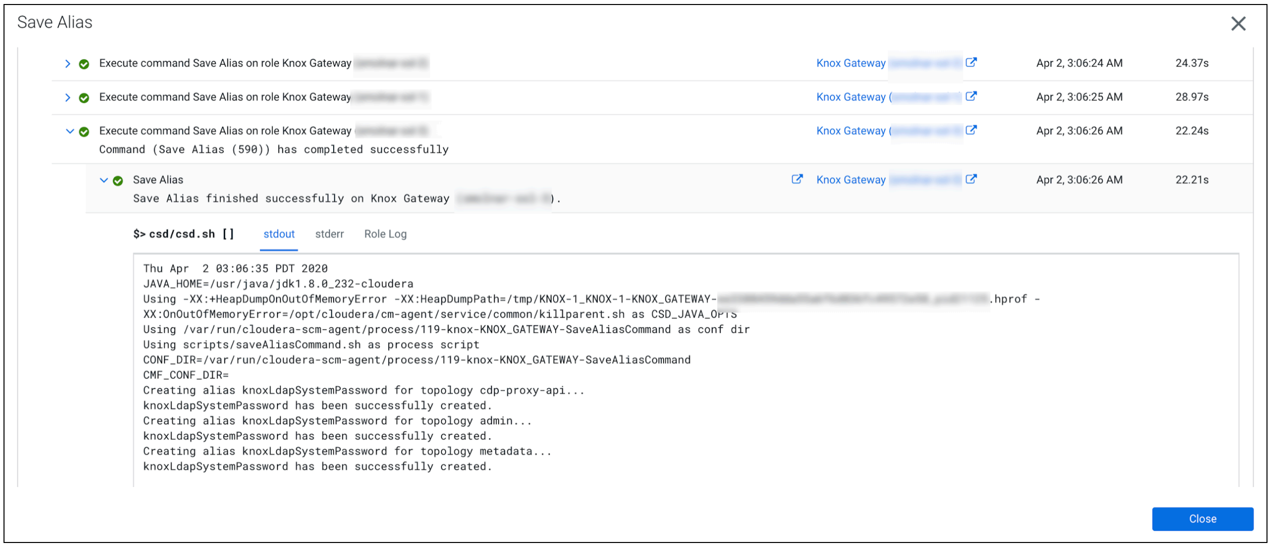
✓ Execute 3 steps in parallel Apr 2, 3:06:24 AM 29.99s

```
Successfully completed 3 steps.
```

✓ Execute command Save Alias on role Knox Gateway ([redacted]) Knox Gateway [redacted] Apr 2, 3:06:24 AM 24.37s

✓ Execute command Save Alias on role Knox Gateway ([redacted]) Knox Gateway [redacted] Apr 2, 3:06:25 AM 28.97s

✓ Execute command Save Alias on role Knox Gateway ([redacted]) Knox Gateway [redacted] Apr 2, 3:06:26 AM 22.24s



Execute command Save Alias on role Knox Gateway [redacted] Knox Gateway [redacted] Apr 2, 3:06:24 AM 24.37s

Execute command Save Alias on role Knox Gateway [redacted] Knox Gateway ([redacted] Apr 2, 3:06:25 AM 28.97s

Execute command Save Alias on role Knox Gateway [redacted] Knox Gateway ([redacted] Apr 2, 3:06:26 AM 22.24s

```
Command (Save Alias (590)) has completed successfully
```

Save Alias Knox Gateway [redacted] Apr 2, 3:06:26 AM 22.21s

```
Save Alias finished successfully on Knox Gateway [redacted]).
```

$> 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-1-KNOX_GATEWAY-[redacted].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.
```

Configure Kerberos authentication in Apache Knox shared providers

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

Procedure

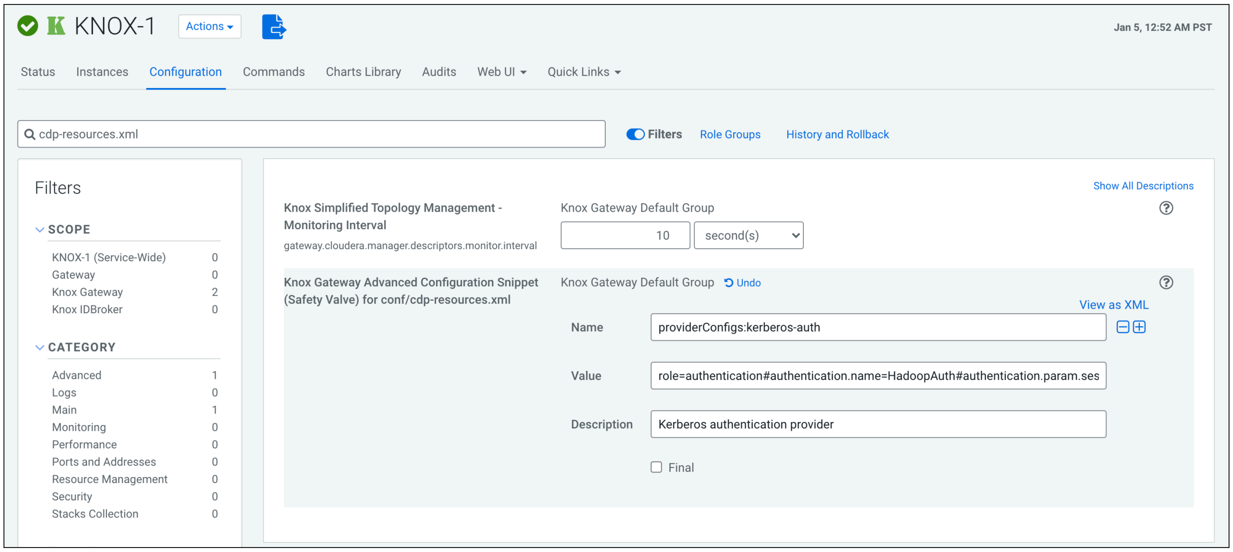
1. From Cloudera Manager Knox Configuration , add the following entry in the Knox Gateway Advanced   Co nfiguration Snippet (Safety Valve) for   conf/cdp-resources.xml:

   - name = providerConfigs:kerberos-auth
   - 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_KER
   BEROS_KEYTAB#
   ```

```
authentication.param.hadoop.auth.config.kerberos.name.rules=DEFAULT
```

Note: Note: Paste the value = code as a single line, for e.g. role=authentication#authentication.name=HadoopAuth#authentication.param.sessionTimeout=30[...]#authentication.param.hadoop.auth.config.hadoop.proxyuser.impala.groups=*



Knox Simplified Topology Management - Monitoring Interval
gateway.cloudera.manager.descriptors.monitor.interval
Knox Gateway Default Group: 10 second(s)

Knox Gateway Advanced Configuration Snippet (Safety Valve) for conf/cdp-resources.xml
Knox Gateway Default Group

Name: providerConfigs:kerberos-auth

Value: role=authentication#authentication.name=HadoopAuth#authentication.param.ses

Description: Kerberos authentication provider

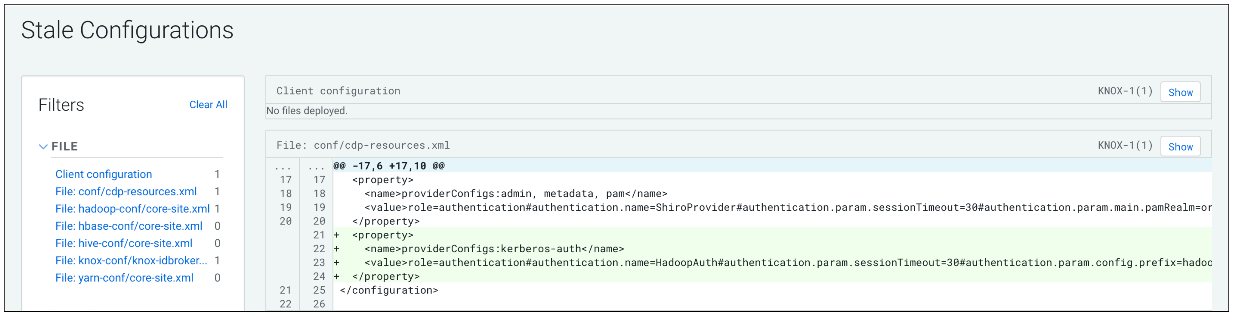
☐ Final

Where:

- AUTH_CONFIG_KERBEROS_PRINCIPAL is the actual SPNEGO principal generated for the given host (see Administration -> Security -> Kerberos Principals / HTTP).
- AUTH_CONFIG_KERBEROS_KEYTAB is the Cloudera Manager-generated keytab file of the current Knox process. It is located in Cloudera Manager's CONF_DIR which can be found on the Processes tab of the Knox Gateway instance.
- ALIAS=AUTH_CONFIG_SIGNATURE_SECRET must not be stored as a plain text password, so use Knox's alias service. This means that whatever topology will reference that shared provider configuration, the AUTH_CONFIG_SIGNATURE_SECRET must be created for it (see "Saving aliases" for details.

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/kerberos-auth
{
  "providers" : [ {
    "role" : "authentication",
    "name" : "HadoopAuth",
    "enabled" : true,
    "params" : {
      "config.prefix" : "hadoop.auth.config",
      "hadoop.auth.config.cookie.path" : "/",
      "hadoop.auth.config.hadoop.proxyuser.hive.groups" : "*",
      "hadoop.auth.config.hadoop.proxyuser.hive.hosts" : "*",
      "hadoop.auth.config.hadoop.proxyuser.httpfs.groups" : "*",
      "hadoop.auth.config.hadoop.proxyuser.httpfs.hosts" : "*",
      "hadoop.auth.config.hadoop.proxyuser.hue.groups" : "*",
      "hadoop.auth.config.hadoop.proxyuser.hue.hosts" : "*",
      "hadoop.auth.config.hadoop.proxyuser.impala.groups" : "*",
      "hadoop.auth.config.hadoop.proxyuser.impala.hosts" : "*",
      "hadoop.auth.config.hadoop.proxyuser.livy.groups" : "*",
      "hadoop.auth.config.hadoop.proxyuser.livy.hosts" : "*",
      "hadoop.auth.config.hadoop.proxyuser.oozie.groups" : "*",
      "hadoop.auth.config.hadoop.proxyuser.oozie.hosts" : "*",
      "hadoop.auth.config.kerberos.keytab" : "/var/run/cloudera-scm-agent/
process/163-knox-IDBROKER/knox.keytab",
      "hadoop.auth.config.kerberos.name.rules" : "DEFAULT",
      "hadoop.auth.config.kerberos.principal" : "HTTP/sampleHost@ABC.CLOUD
ERA.COM",
      "hadoop.auth.config.signature.secret" : "${ALIAS=AUTH_CONFIG_SIGNATU
RE_SECRET}",
      "hadoop.auth.config.simple.anonymous.allowed" : "false",
      "hadoop.auth.config.token.validity" : "1800",
      "hadoop.auth.config.type" : "kerberos",
      "sessionTimeout" : "30"
    }
  } ],
  "readOnly" : true
}
```



Stale Configurations

Client configuration	1
File: conf/cdp-resources.xml	1
File: hadoop-conf/core-site.xml	1
File: hbase-conf/core-site.xml	0
File: hive-conf/core-site.xml	0
File: knox-conf/knox-idbroker...	1
File: yarn-conf/core-site.xml	0

Client configuration KNOX-1(1)

No files deployed.

File: conf/cdp-resources.xml KNOX-1(1)

```
@@ -17,6 +17,10 @@
   <property>
     <name>providerConfigs:admin, metadata, pam</name>
     <value>role=authentication#authentication.name=ShiroProvider#authentication.param.sessionTimeout=30#authentication.param.main.pamRealm=or
   </property>
+  <property>
+    <name>providerConfigs:kerberos-auth</name>
+    <value>role=authentication#authentication.name=HadoopAuth#authentication.param.sessionTimeout=30#authentication.param.config.prefix=hadoo
+  </property>
 </configuration>
```

Related Information

Saving aliases

Managing services for Apache Knox via Cloudera Manager

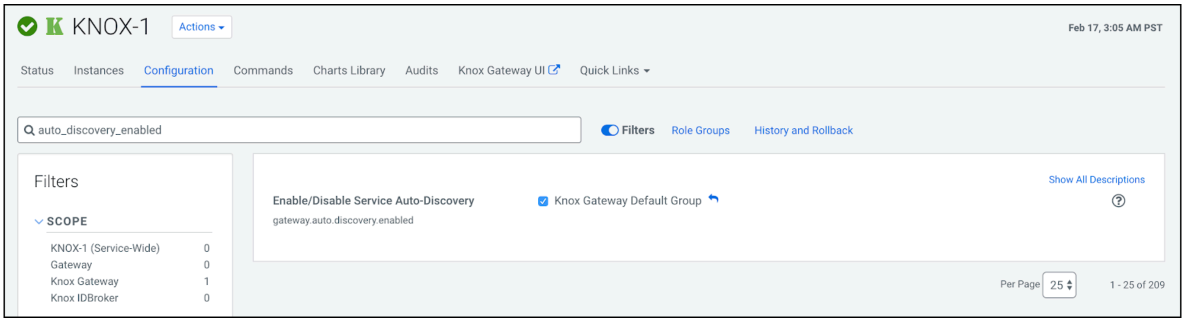
You can enable or disable known or custom services in Knox proxy via Cloudera Manager.

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:



Enable/Disable Service Auto-Discovery
gateway.auto.discovery.enabled
☑ Knox Gateway Default Group

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

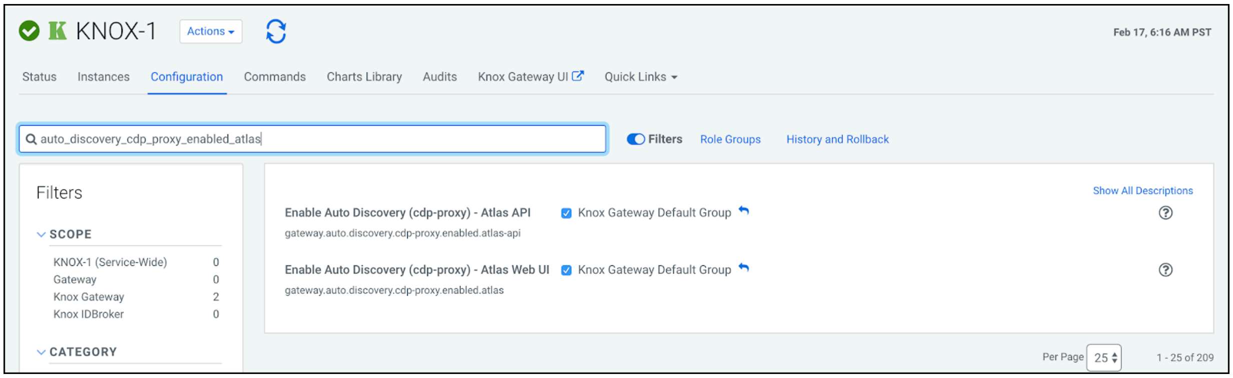
"Known" services are officially-supported Knox services (like Apache Atlas, Ranger, Solr, etc.) Cloudera Manager provides and manages all the required service definition files.

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".

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

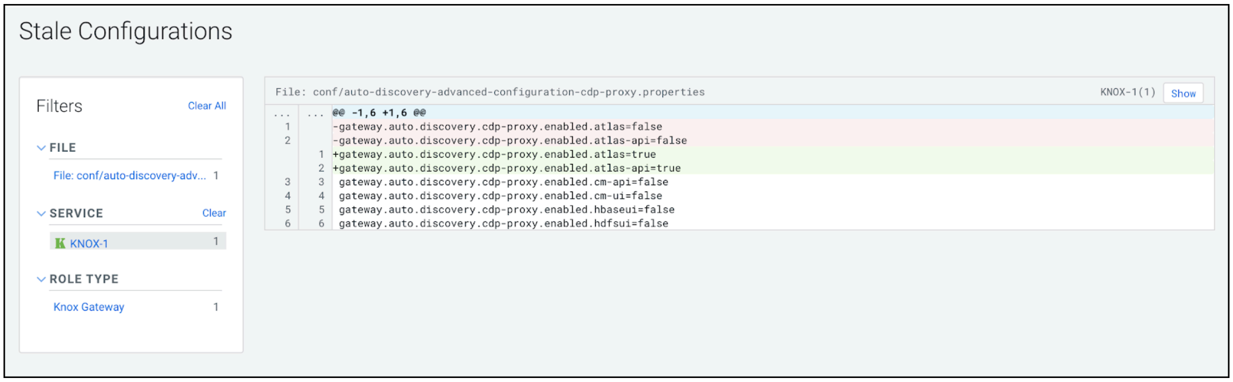


Enable Auto Discovery (cdp-proxy) - Atlas API
gateway.auto.discovery.cdp-proxy.enabled.atlas-api
☑ Knox Gateway Default Group

Enable Auto Discovery (cdp-proxy) - Atlas Web UI
gateway.auto.discovery.cdp-proxy.enabled.atlas
☑ Knox Gateway Default Group

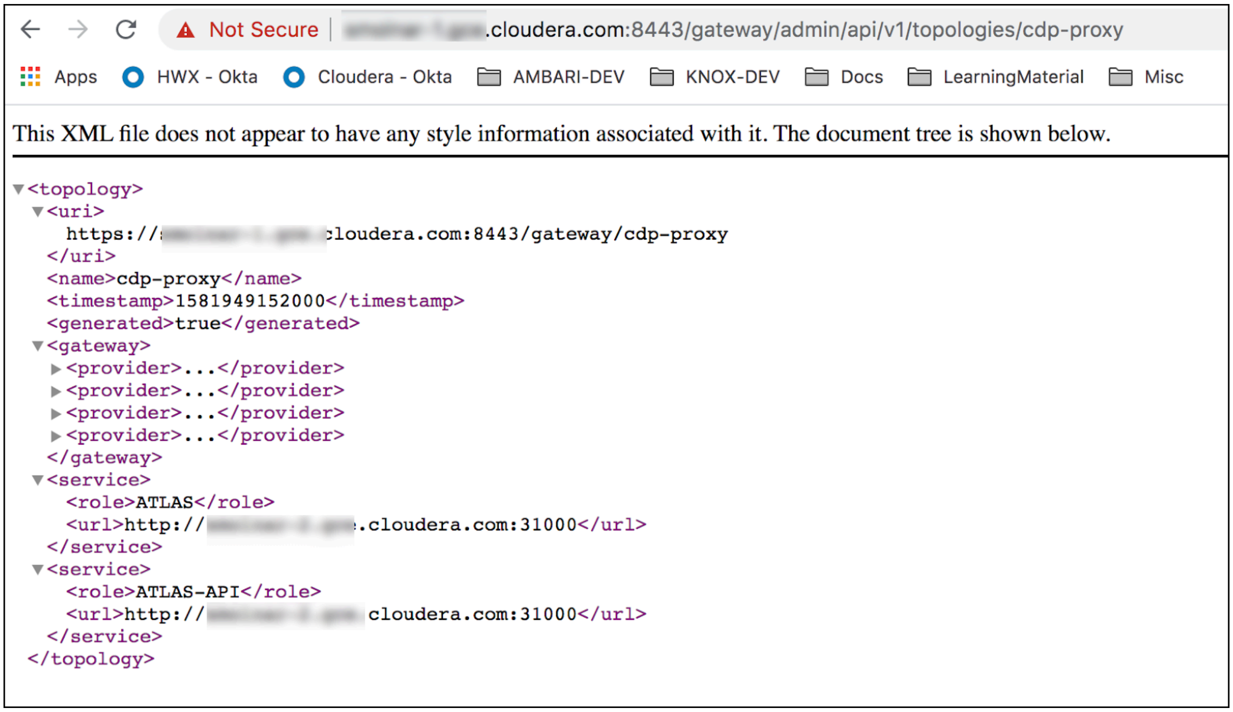
2. Save your changes.

3. The ‘Refresh needed’ stale configuration indicator appears; click it and wait until the refresh process finishes.



```
File: conf/auto-discovery-advanced-configuration-cdp-proxy.properties
...  ...  @@ -1,6 +1,6 @@
-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.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<topology>
  ▼<uri>
     https://[redacted]cloudera.com:8443/gateway/cdp-proxy
   </uri>
   <name>cdp-proxy</name>
   <timestamp>1581949152000</timestamp>
   <generated>true</generated>
  ▼<gateway>
    ▶<provider>...</provider>
    ▶<provider>...</provider>
    ▶<provider>...</provider>
    ▶<provider>...</provider>
   </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>
```

Related Information

Add a custom service parameter to a known service

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.

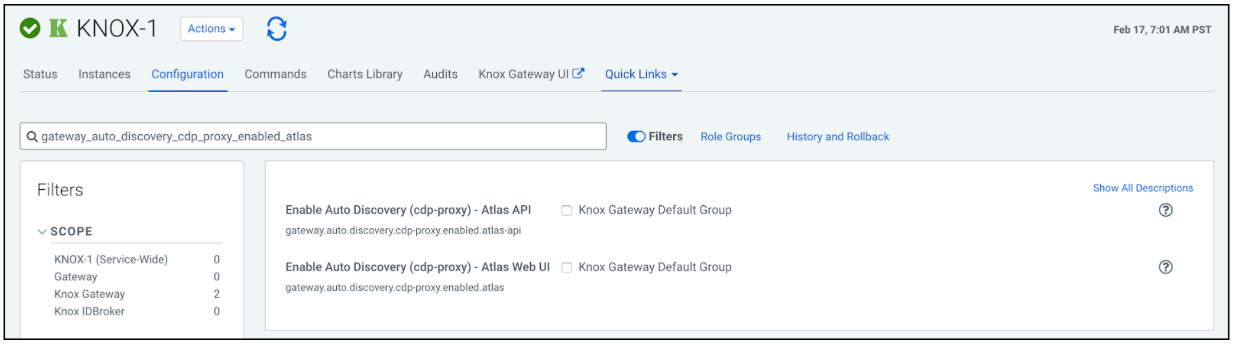
About this task

“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

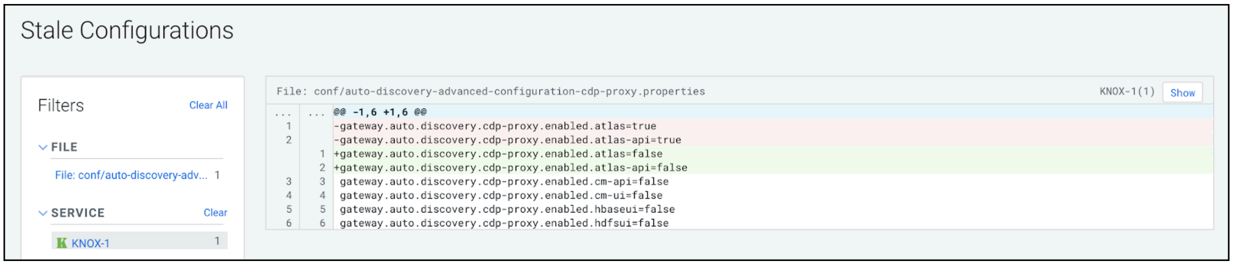
1. From Cloudera Manager Knox Configuration , uncheck the Gateway Auto Discovery (cdp-proxy) - $Component boxes.
   In this example, we disable:
   - gateway_auto_discovery_cdp_proxy_enabled_atlas
   - gateway_auto_discovery_cdp_proxy_enabled_atlas_ui



Enable Auto Discovery (cdp-proxy) - Atlas API ☐ Knox Gateway Default Group
gateway.auto.discovery.cdp-proxy.enabled.atlas-api

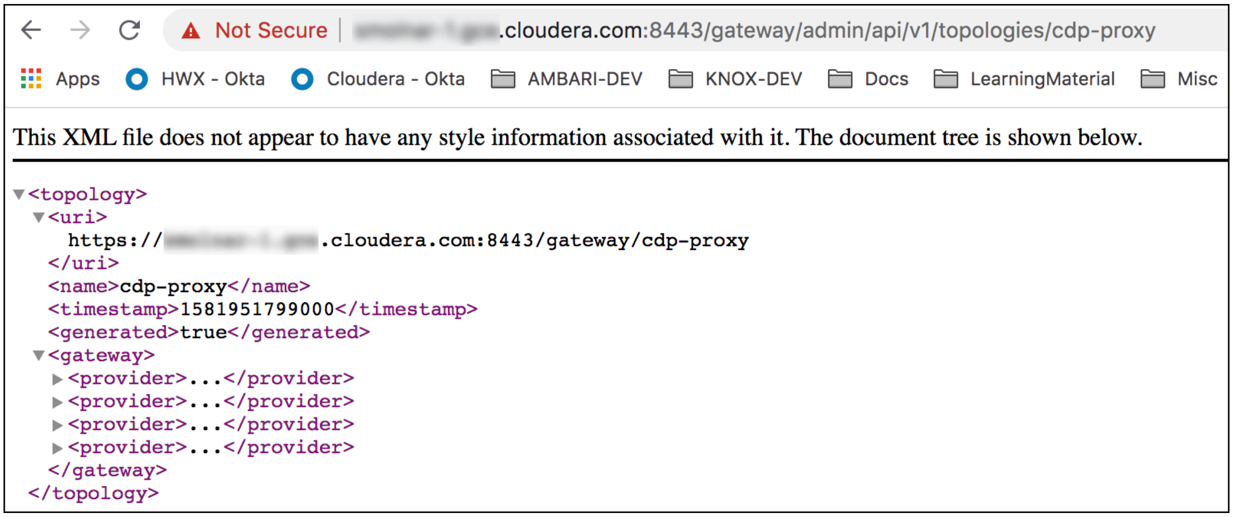
Enable Auto Discovery (cdp-proxy) - Atlas Web UI ☐ Knox Gateway Default Group
gateway.auto.discovery.cdp-proxy.enabled.atlas

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



```
File: conf/auto-discovery-advanced-configuration-cdp-proxy.properties
@@ -1,6 +1,6 @@
-gateway.auto.discovery.cdp-proxy.enabled.atlas=true
-gateway.auto.discovery.cdp-proxy.enabled.atlas-api=true
+gateway.auto.discovery.cdp-proxy.enabled.atlas=false
+gateway.auto.discovery.cdp-proxy.enabled.atlas-api=false
 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 custom service got removed by going to the following URL: http s://*$KNOX_GATEWAY_HOST*:*$PORT*/*$GATEWAY_PATH*/admin/api/v1/topologies/cdp-proxy.



[redacted].cloudera.com:8443/gateway/admin/api/v1/topologies/cdp-proxy

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<topology>
  ▼<uri>
     https://[redacted].cloudera.com:8443/gateway/cdp-proxy
   </uri>
   <name>cdp-proxy</name>
   <timestamp>1581951799000</timestamp>
   <generated>true</generated>
  ▼<gateway>
    ▶<provider>...</provider>
    ▶<provider>...</provider>
    ▶<provider>...</provider>
    ▶<provider>...</provider>
   </gateway>
</topology>
```

Add a custom service to Apache Knox Proxy

How to add a custom service in Knox proxy via 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) exist in the KNOX_DATA_DIR/services folder. These are not recommended for production environments, and not supported by Cloudera.

In this example, we are going to add a custom service (*MY_SERVICE*) in cdp-proxy with the following attributes:

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

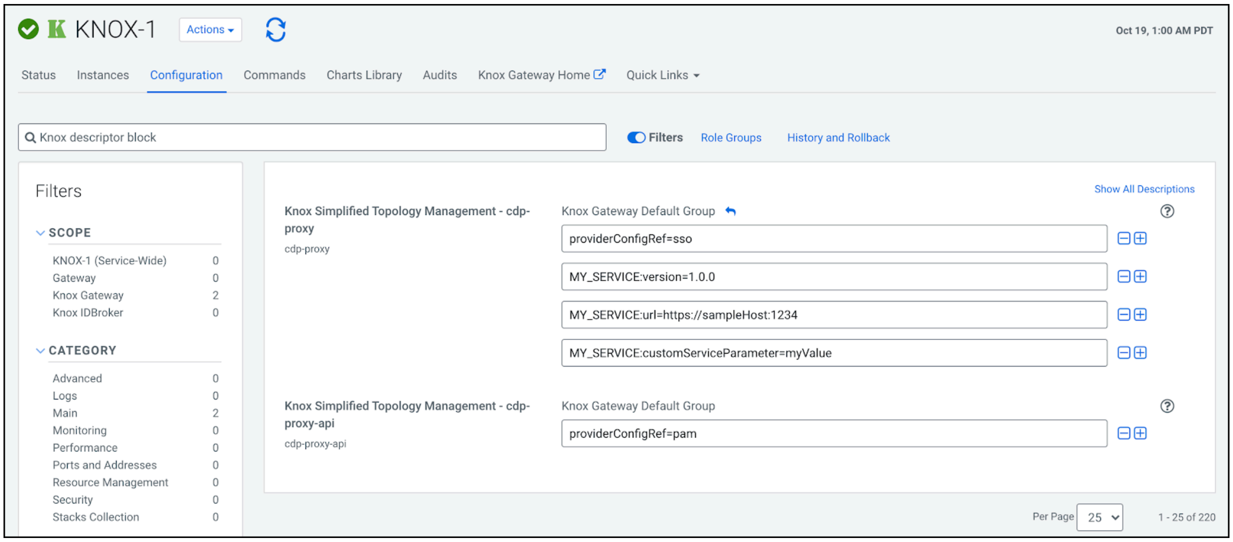
Important: Adding a custom service will only work if you provide the service definition files (service.xml and rewrite.xml) in the KNOX_DATA_DIR/services folder.

To achieve the goals we need to add 3 new entries with the above-listed parameters in Knox Simplified Topology Management - cdp-proxy. Then we 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 3 new entries with the above-listed parameters.

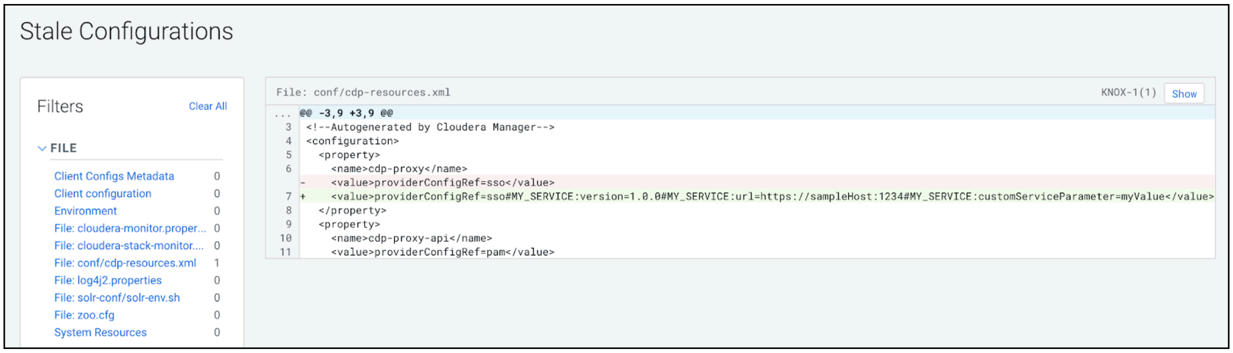
```
MY_SERVICE:version=1.0.0
MY_SERVICE:url=https://sampleHost:1234
MY_SERVICE:customServiceParameter=myValue
```



Knox Simplified Topology Management - cdp-proxy
cdp-proxy
Knox Gateway Default Group
providerConfigRef=sso
MY_SERVICE:version=1.0.0
MY_SERVICE:url=https://sampleHost:1234
MY_SERVICE:customServiceParameter=myValue

Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
Knox Gateway Default Group
providerConfigRef=pam

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


Stale Configurations
Filters
Clear All
FILE
Client Configs Metadata 0
Client configuration 0
Environment 0
File: cloudera-monitor.proper... 0
File: cloudera-stack-monitor.... 0
File: conf/cdp-resources.xml 1
File: log4j2.properties 0
File: solr-conf/solr-env.sh 0
File: zoo.cfg 0
System Resources 0
KNOX-1(1)
Show

File: conf/cdp-resources.xml

```
@@ -3,9 +3,9 @@
 <!--Autogenerated by Cloudera Manager-->
 <configuration>
   <property>
     <name>cdp-proxy</name>
-    <value>providerConfigRef=sso</value>
+    <value>providerConfigRef=sso#MY_SERVICE:version=1.0.0#MY_SERVICE:url=https://sampleHost:1234#MY_SERVICE:customServiceParameter=myValue</value>
   </property>
   <property>
     <name>cdp-proxy-api</name>
     <value>providerConfigRef=pam</value>
```

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

Add a custom topology in the deployed Apache Knox Gateway

How to add a custom service to a custom topology in Knox proxy via Cloudera Manager.

About this task

In this example, we are going to add a custom service (MY_SERVICE) in custom-topology with the following attributes:

- ProviderConfigRef: a string representing a reference of an existing share-provider. We will use the pre-configured pam provider.
- Version: the service's version, e.g. 1.0.0.
- URL: the service URL, e.g. https://sampleHost:1234.
- Service parameter: a sample service parameter, e.g. myValue.

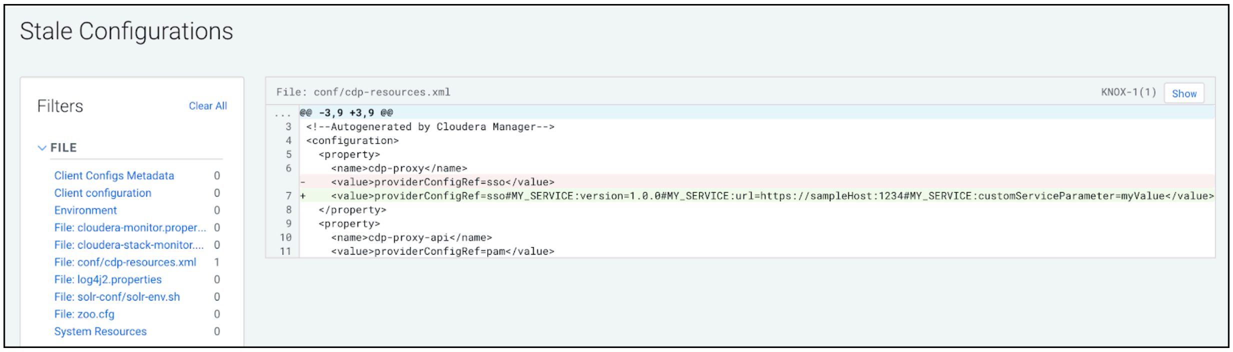
Procedure

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

```
Name = custom-topology
Value =providerConfigRef=pam#
MY_SERVICE:version=1.0.0#
MY_SERVICE:url=https://sampleHost:1234#
MY_SERVICE:myCustomServiceParameter=myValue
Description = This is a custom topology with one service called MY_SERVICE
```

2. Save your changes.

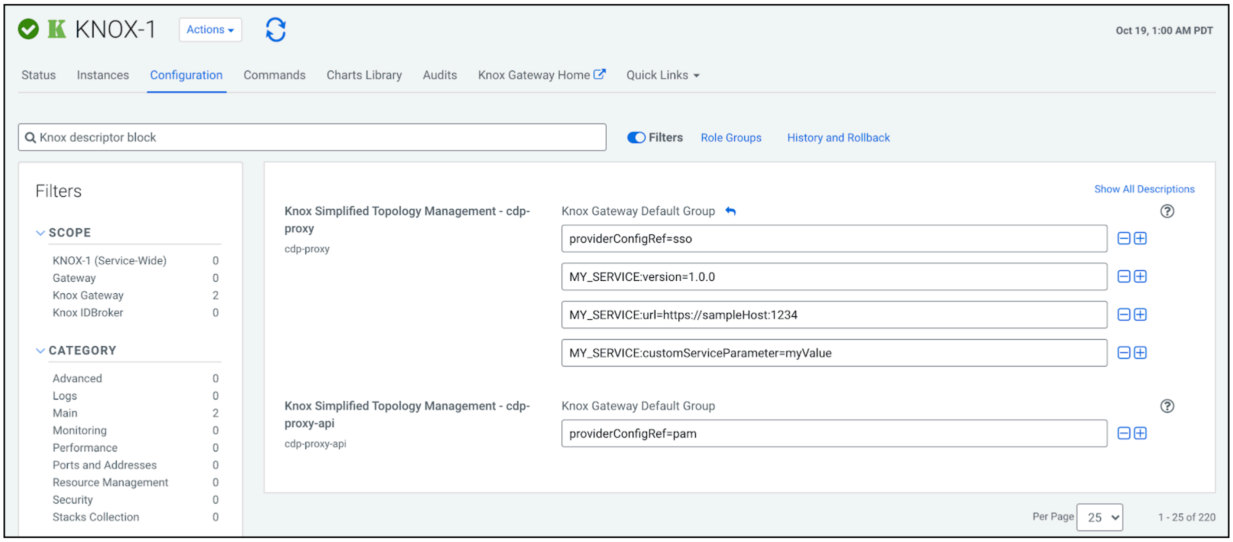
3. The ‘Refresh needed’ stale configuration indicator appears; click it and wait until the refresh process finishes.



FILE	
Client Configs Metadata	0
Client configuration	0
Environment	0
File: cloudera-monitor.proper...	0
File: cloudera-stack-monitor....	0
File: conf/cdp-resources.xml	1
File: log4j2.properties	0
File: solr-conf/solr-env.sh	0
File: zoo.cfg	0
System Resources	0

File: conf/cdp-resources.xml

```
... @@ -3,9 +3,9 @@
 <!--Autogenerated by Cloudera Manager-->
 <configuration>
   <property>
     <name>cdp-proxy</name>
-    <value>providerConfigRef=sso</value>
+    <value>providerConfigRef=sso#MY_SERVICE:version=1.0.0#MY_SERVICE:url=https://sampleHost:1234#MY_SERVICE:customServiceParameter=myValue</value>
   </property>
   <property>
     <name>cdp-proxy-api</name>
     <value>providerConfigRef=pam</value>
```



Knox descriptor block

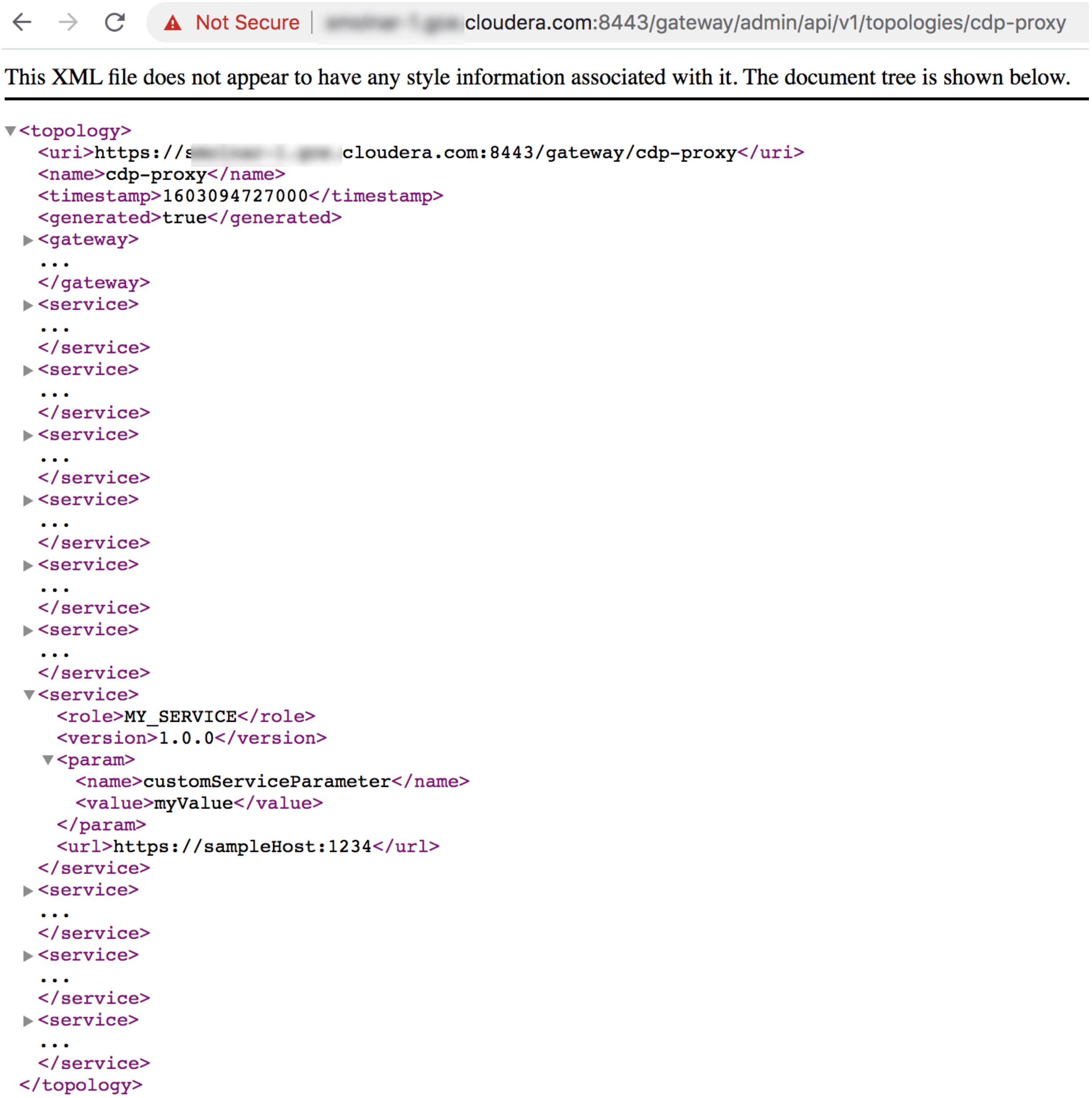
SCOPE	
KNOX-1 (Service-Wide)	0
Gateway	0
Knox Gateway	2
Knox IDBroker	0

CATEGORY	
Advanced	0
Logs	0
Main	2
Monitoring	0
Performance	0
Ports and Addresses	0
Resource Management	0
Security	0
Stacks Collection	0

Knox Simplified Topology Management - cdp-proxy
cdp-proxy
Knox Gateway Default Group
providerConfigRef=sso
MY_SERVICE:version=1.0.0
MY_SERVICE:url=https://sampleHost:1234
MY_SERVICE:customServiceParameter=myValue

Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
Knox Gateway Default Group
providerConfigRef=pam

4. Validate that the operation was successful by going to the following URL: https://*$KNOX_GATEWAY_HOST*:*$PORT*/*$GATEWAY_PATH*/admin/api/v1/topologies/cdp-proxy.



[redacted]cloudera.com:8443/gateway/admin/api/v1/topologies/cdp-proxy

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<topology>
   <uri>https://s[redacted]cloudera.com:8443/gateway/cdp-proxy</uri>
   <name>cdp-proxy</name>
   <timestamp>1603094727000</timestamp>
   <generated>true</generated>
 ▶<gateway>
   ...
   </gateway>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 ▼<service>
     <role>MY_SERVICE</role>
     <version>1.0.0</version>
   ▼<param>
       <name>customServiceParameter</name>
       <value>myValue</value>
     </param>
     <url>https://sampleHost:1234</url>
   </service>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 ▶<service>
   ...
   </service>
 </topology>
```

Managing Service Parameters for Apache Knox via Cloudera Manager

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

Add a custom service parameter to a known service

How to add a custom service parameter to a service via Cloudera Manager..

Before you begin

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

About this task

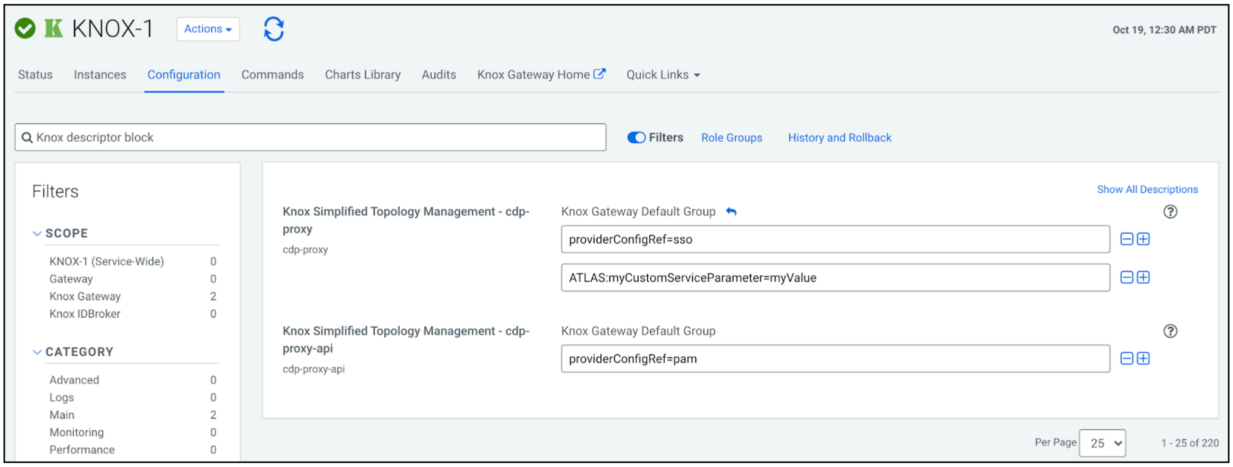
In this example, we are adding a custom service parameter with a custom value (myCustomServiceParameter=myValue) to ATLAS in cdp-proxy.

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:

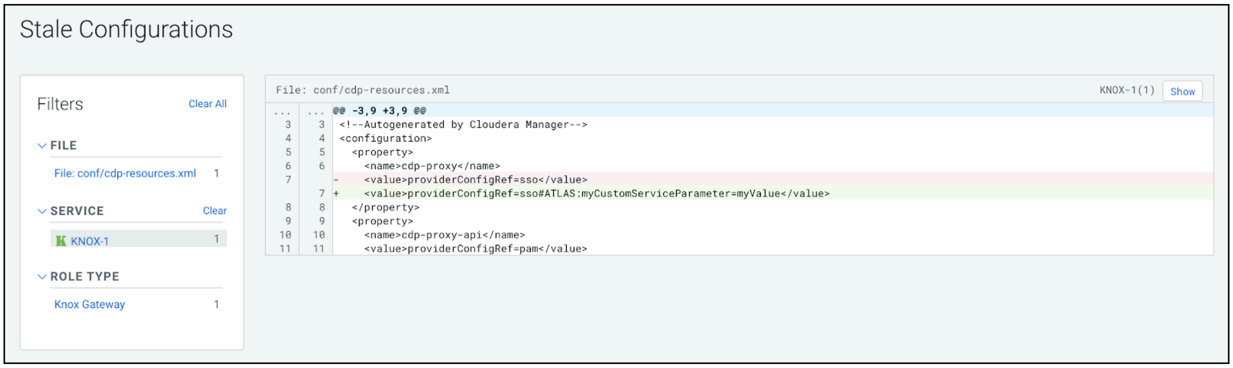
   ```
   HIVE:url=http://localhost:123
   HIVE:version:3.0.0
   HIVE:test.parameter.name=test.parameter.value
   ```

   Knox Simplified Topology Management - cdp-proxy
   cdp-proxy
   Knox Gateway Default Group
   providerConfigRef=sso
   ATLAS:myCustomServiceParameter=myValue

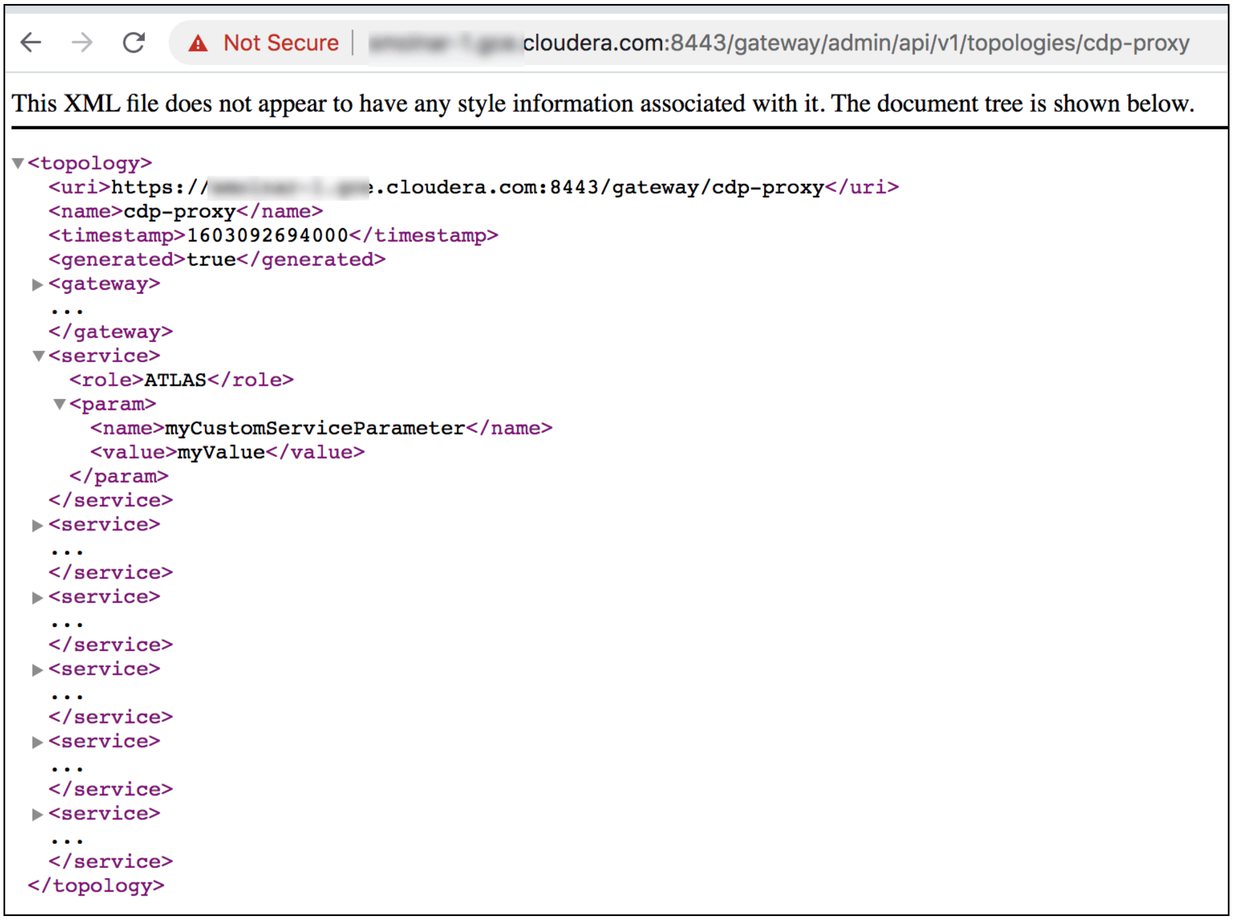
   Knox Simplified Topology Management - cdp-proxy-api
   cdp-proxy-api
   Knox Gateway Default Group
   providerConfigRef=pam

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

   ```
   File: conf/cdp-resources.xml
   @@ -3,9 +3,9 @@
    <!--Autogenerated by Cloudera Manager-->
    <configuration>
      <property>
        <name>cdp-proxy</name>
   -    <value>providerConfigRef=sso</value>
   +    <value>providerConfigRef=sso#ATLAS:myCustomServiceParameter=myValue</value>
      </property>
      <property>
        <name>cdp-proxy-api</name>
        <value>providerConfigRef=pam</value>
   ```

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



Not Secure
[redacted]cloudera.com:8443/gateway/admin/api/v1/topologies/cdp-proxy

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<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 a custom service parameter

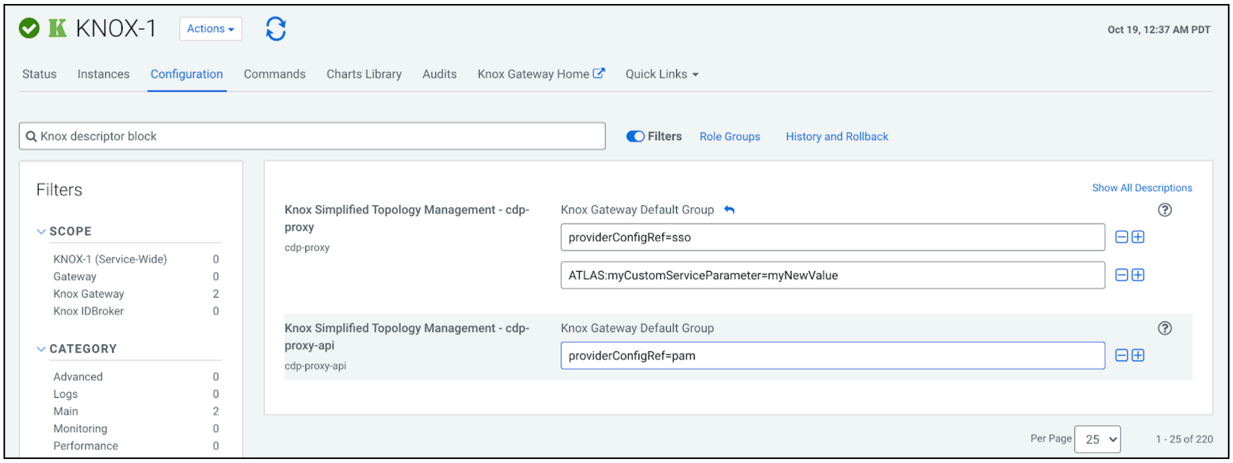
How to edit a custom service parameter in Knox via Cloudera Manager.

About this task

In this sample, we are going to update a previously entered service parameter - myCustomServiceParameter=my Value to myNewValue- for ATLAS in cdp-proxy. We change that entry, save our changes, and refresh our 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

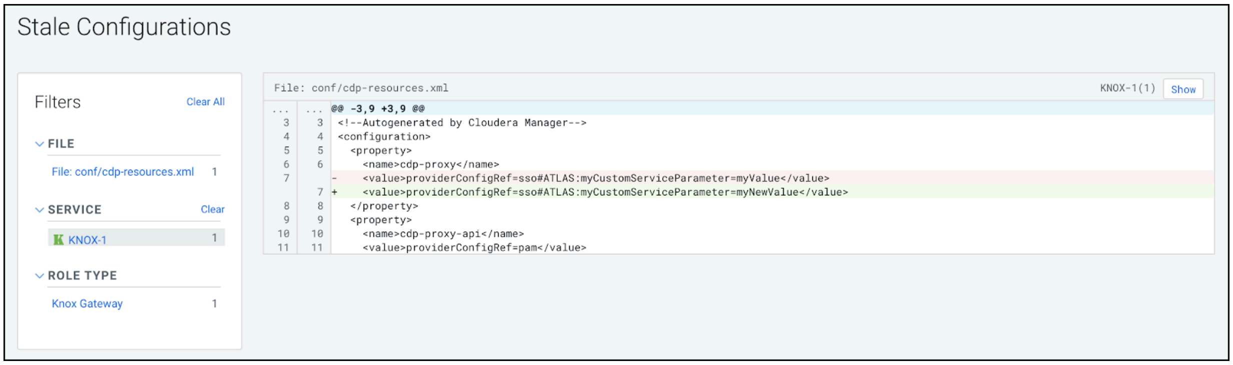


Knox descriptor block

Knox Simplified Topology Management - cdp-proxy
cdp-proxy
Knox Gateway Default Group
providerConfigRef=sso
ATLAS:myCustomServiceParameter=myNewValue

Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
Knox Gateway Default Group
providerConfigRef=pam

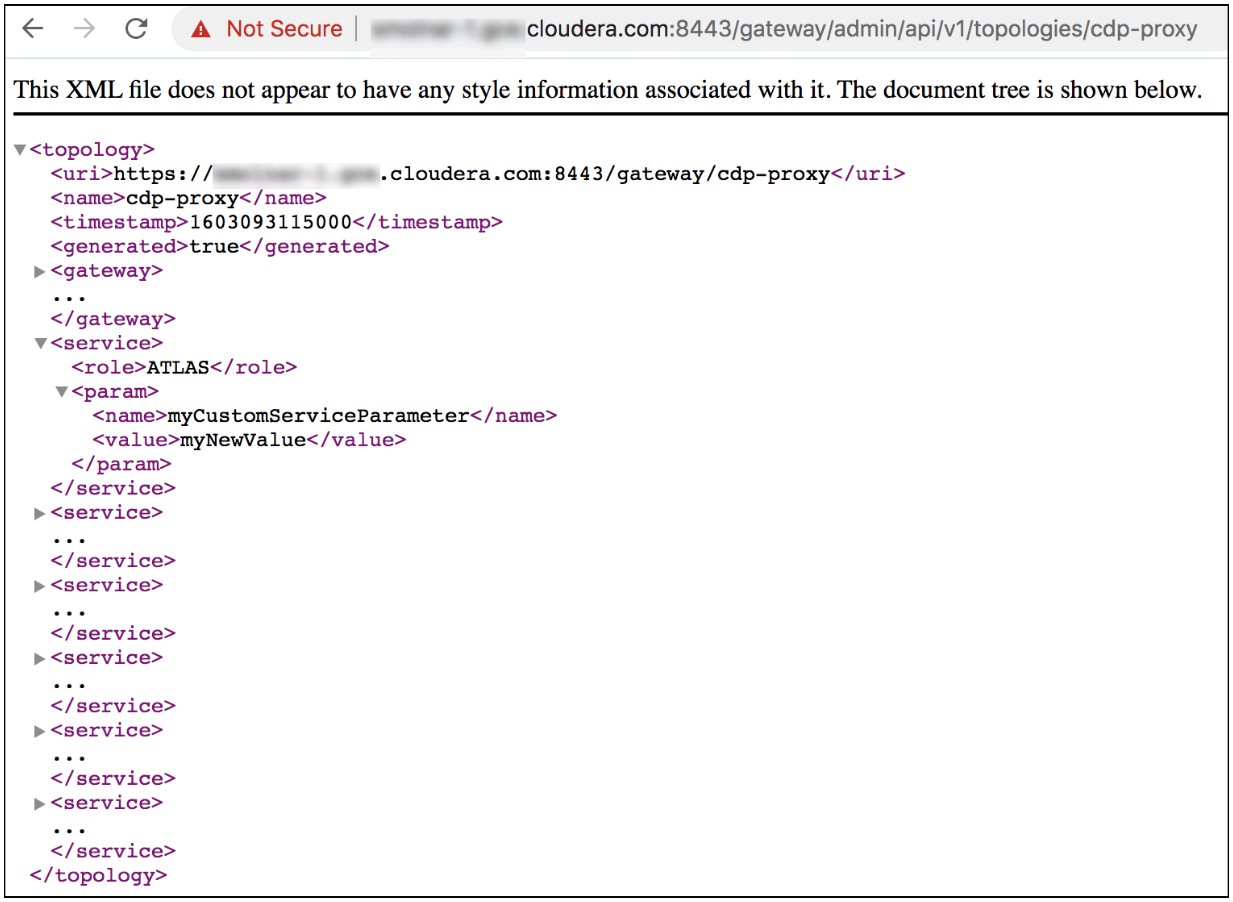
2. Save your changes.
3. The ‘Refresh needed’ stale configuration indicator appears; click it and wait until the refresh process finishes.


Stale Configurations

File: conf/cdp-resources.xml

```
@@ -3,9 +3,9 @@
 <!--Autogenerated by Cloudera Manager-->
 <configuration>
   <property>
     <name>cdp-proxy</name>
-    <value>providerConfigRef=sso#ATLAS:myCustomServiceParameter=myValue</value>
+    <value>providerConfigRef=sso#ATLAS:myCustomServiceParameter=myNewValue</value>
   </property>
   <property>
     <name>cdp-proxy-api</name>
     <value>providerConfigRef=pam</value>
```

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



Not Secure
[redacted]cloudera.com:8443/gateway/admin/api/v1/topologies/cdp-proxy

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<topology>
   <uri>https://[redacted].cloudera.com:8443/gateway/cdp-proxy</uri>
   <name>cdp-proxy</name>
   <timestamp>1603093115000</timestamp>
   <generated>true</generated>
  ▶<gateway>
   ...
   </gateway>
  ▼<service>
     <role>ATLAS</role>
    ▼<param>
       <name>myCustomServiceParameter</name>
       <value>myNewValue</value>
     </param>
   </service>
  ▶<service>
   ...
   </service>
  ▶<service>
   ...
   </service>
  ▶<service>
   ...
   </service>
  ▶<service>
   ...
   </service>
  ▶<service>
   ...
   </service>
 </topology>
```

Remove a custom service parameter

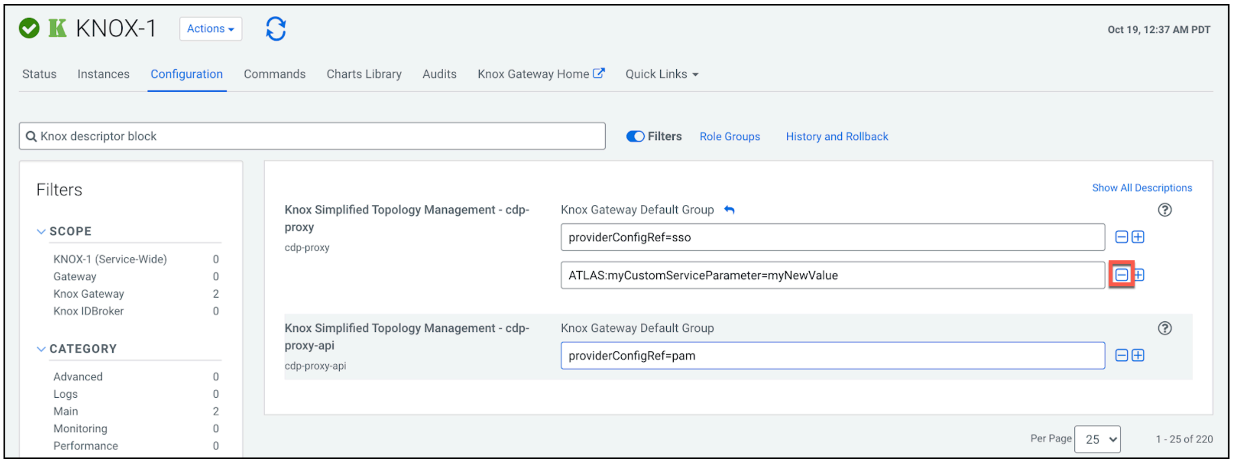
How to remove a custom service parameter in Knox via Cloudera Manager.

About this task

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

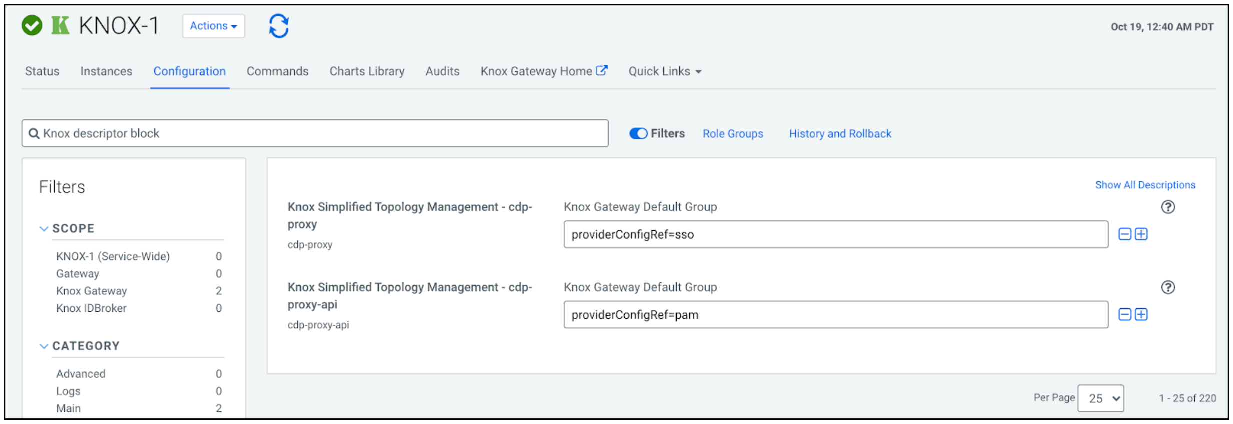
Procedure

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



Knox Simplified Topology Management - cdp-proxy
cdp-proxy
Knox Gateway Default Group
providerConfigRef=sso
ATLAS:myCustomServiceParameter=myNewValue

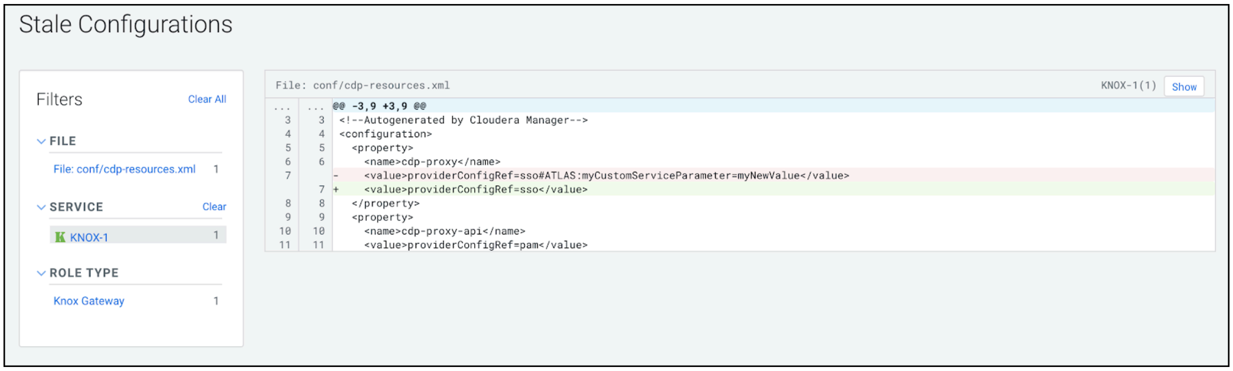
Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
Knox Gateway Default Group
providerConfigRef=pam



Knox Simplified Topology Management - cdp-proxy
cdp-proxy
Knox Gateway Default Group
providerConfigRef=sso

Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
Knox Gateway Default Group
providerConfigRef=pam

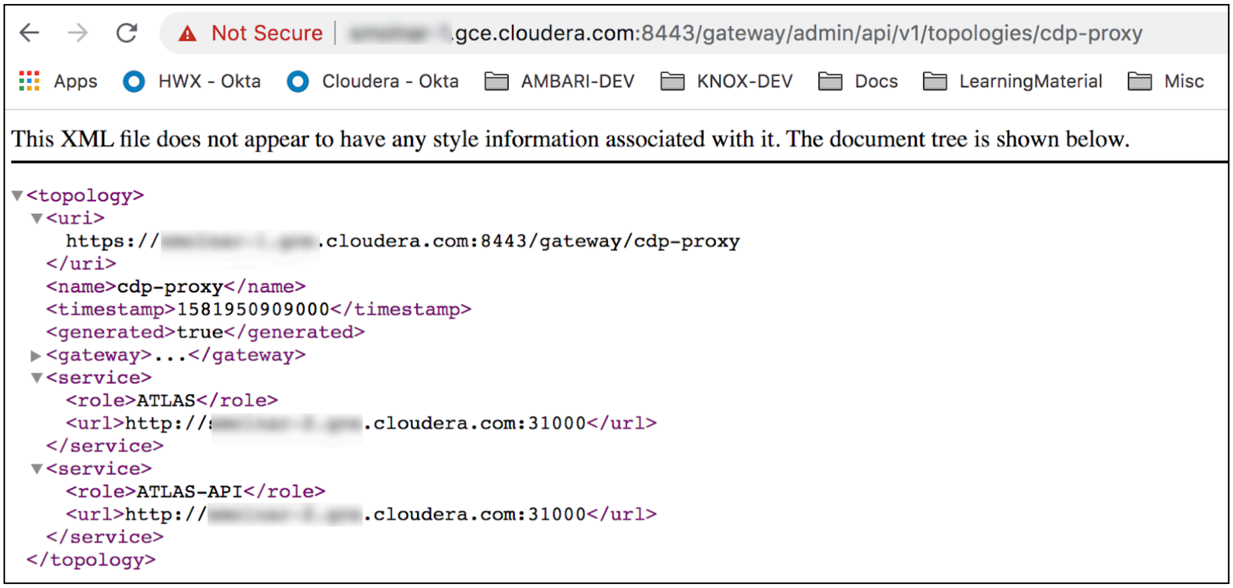
2. Save your changes.
3. The ‘Refresh needed’ stale configuration indicator appears; click it and wait until the refresh process finishes.



File: conf/cdp-resources.xml

```
@@ -3,9 +3,9 @@
 <!--Autogenerated by Cloudera Manager-->
 <configuration>
   <property>
     <name>cdp-proxy</name>
-    <value>providerConfigRef=sso#ATLAS:myCustomServiceParameter=myNewValue</value>
+    <value>providerConfigRef=sso</value>
   </property>
   <property>
     <name>cdp-proxy-api</name>
     <value>providerConfigRef=pam</value>
```

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



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<topology>
  ▼<uri>
     https://[redacted].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>
```