**Cloudera CASK CDAP Offline Installation Guide**

Contents

[Installing CDAP via Cloudera Manager 2](#_Toc482710344)

[CDAP Roles and Dependencies 2](#_Toc482710345)

[Perform Preliminary Configuration 3](#_Toc482710346)

[HDFS Permissions 4](#_Toc482710347)

[Download the Required CSD File and Configure Cloudera 5](#_Toc482710348)

[Installing the Parcel 5](#_Toc482710349)

[Starting the CDAP Service 7](#_Toc482710350)

[CASK Marketplace Offline Repository 14](#_Toc482710351)

[Download Packages 14](#_Toc482710352)

[Install Apache Maven 14](#_Toc482710353)

[Make Package Metadata 14](#_Toc482710354)

[Create a Web Server 15](#_Toc482710355)

[Update the CDAP Market URL within the CDAP Configuration Settings GUI as shown below: 15](#_Toc482710356)

[SSL Integration 15](#_Toc482710357)

[Create the self-signed certificate and key file for User Interface (UI) access 15](#_Toc482710358)

[Create the Java Keystore 15](#_Toc482710359)

[Active Directory Authentication 19](#_Toc482710360)

[Authentication Configuration Settings 21](#_Toc482710361)

[Proxy Settings for Authentication 22](#_Toc482710362)

[Nginx Inbound Configuration 22](#_Toc482710363)

[HAProxy Outbound Configuration 23](#_Toc482710364)

# Introduction

Installing CDAP within our Cloudera environment on RADON consists of a series of tasks. Because the Cloudera Hadoop Ecosystem resides within an isolated sandbox, there are a series of additional steps required to make the configuration work.

The steps below are detailed in this document.

1. Understanding the CDAP roles and dependencies within Hadoop.
2. Installation of CDAP within Cloudera
3. Building our own offline CDAP Marketplace
4. Configuring SSL for User Access to CDAP User Interface (UI) and communication
5. Configuring Active Directory Authentication for Access to CDAP UI
6. Setting up an Inbound Proxy For access to Sandbox from external network
7. Setting up an Outbound Proxy for Active Directory communication from within sandbox.

# Installing CDAP via Cloudera Manager

## CDAP Roles and Dependencies

* As CDAP depends on HDFS, YARN, HBase, ZooKeeper, and (optionally) Hive and Spark, it must be installed on cluster host(s) with full client configurations for these dependent services.
* The CDAP Master Service role (or CDAP Master) must be co-located on a cluster host with an HDFS Gateway, a YARN Gateway, an HBase Gateway, and—optionally—Hive or Spark Gateways.
* Note that these Gateways are redundant if you are co-locating the CDAP Master role on a cluster host (or hosts, in the case of a deployment with high availability) with actual services, such as the HDFS Namenode, the YARN resource manager, or the HBase Master.
* Note that the CDAP Gateway/Router Service is not a [Cloudera Manager Gateway Role](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cm_mc_managing_roles.html) but is instead another name for the CDAP Router Service.
* CDAP also provides its own Gateway role that can be used to install CDAP client configurations on other hosts of the cluster.
* All services run as the 'cdap' user installed by the parcel.

## Perform Preliminary Configuration

All of these changes can be made from within the Cloudera Manager configuration settings pages.

1. ZooKeeper's **maxClientCnxns** must be raised from its default. We suggest setting it to zero (0: unlimited connections). As each YARN container launched by CDAP makes a connection to ZooKeeper, the number of connections required is a function of usage.
2. Ensure that YARN has sufficient memory capacity by lowering the default minimum container size (controlled by the property **yarn.scheduler.minimum-allocation-mb**). Lack of YARN memory capacity is the leading cause of apparent failures that we see reported. We recommend starting with these settings:
   * **yarn.nodemanager.delete.debug-delay-sec**: 43200
   * **yarn.scheduler.minimum-allocation-mb**: 512 mb

Please ensure your yarn.nodemanager.resource.cpu-vcores and yarn.nodemanager.resource.memory-mb settings are set sufficiently to run CDAP, as described in the [CDAP Memory and Core Requirements](http://docs.cask.co/cdap/3.5.3/en/admin-manual/system-requirements.html#admin-manual-memory-core-requirements).

1. Add additional entries to the YARN Application Classpath for Spark jobs.

If you plan on running Spark programs from CDAP, CDAP requires that additional entries be added to the YARN application classpath, as the Spark installed on Cloudera Manager clusters is a "Hadoop-less" build and does not include Hadoop jars required by Spark.

To resolve this, go to the CM page for your cluster, click on the YARN service, click on the configuration tab, and then enter **mapreduce.application.classpath** in the search box. You will see entries similar to these:

$HADOOP\_MAPRED\_HOME/\*

$HADOOP\_MAPRED\_HOME/lib/\*

$MR2\_CLASSPATH

Copy all the entries to the **yarn.application.classpath** configuration for YARN on your Cluster. The **yarn.application.classpath** setting can be found by searching as mentioned above.

Add the entries required by scrolling to the last entry in the classpath form, clicking the "+" button to add a new text box entry field at the end. Once you have added all the entries from the **mapreduce.application.classpath** to the **yarn.application.classpath**, click on Save.

You can make these changes [using Cloudera Manager](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cm_mc_mod_configs.html). Please restart the stale services upon seeing a prompt to do so after making the above changes.

### HDFS Permissions

Ensure YARN is configured properly to run MapReduce programs. Often, this includes ensuring that the HDFS /user/yarn directory exists with proper permissions:

**#** su hdfs

**$** hdfs dfs -mkdir -p /user/yarn && hadoop fs -chown yarn /user/yarn && hadoop fs -chgrp yarn /user/yarn

hdfs dfs -mkdir -p /user/cdap && hadoop fs -chown cdap /user/cdap && hadoop fs -chgrp cdap /user/cdap

## Download the Required CSD File and Configure Cloudera

For the offline installation via Cloudera you will need the Custom Service Descriptor (CSD) file.

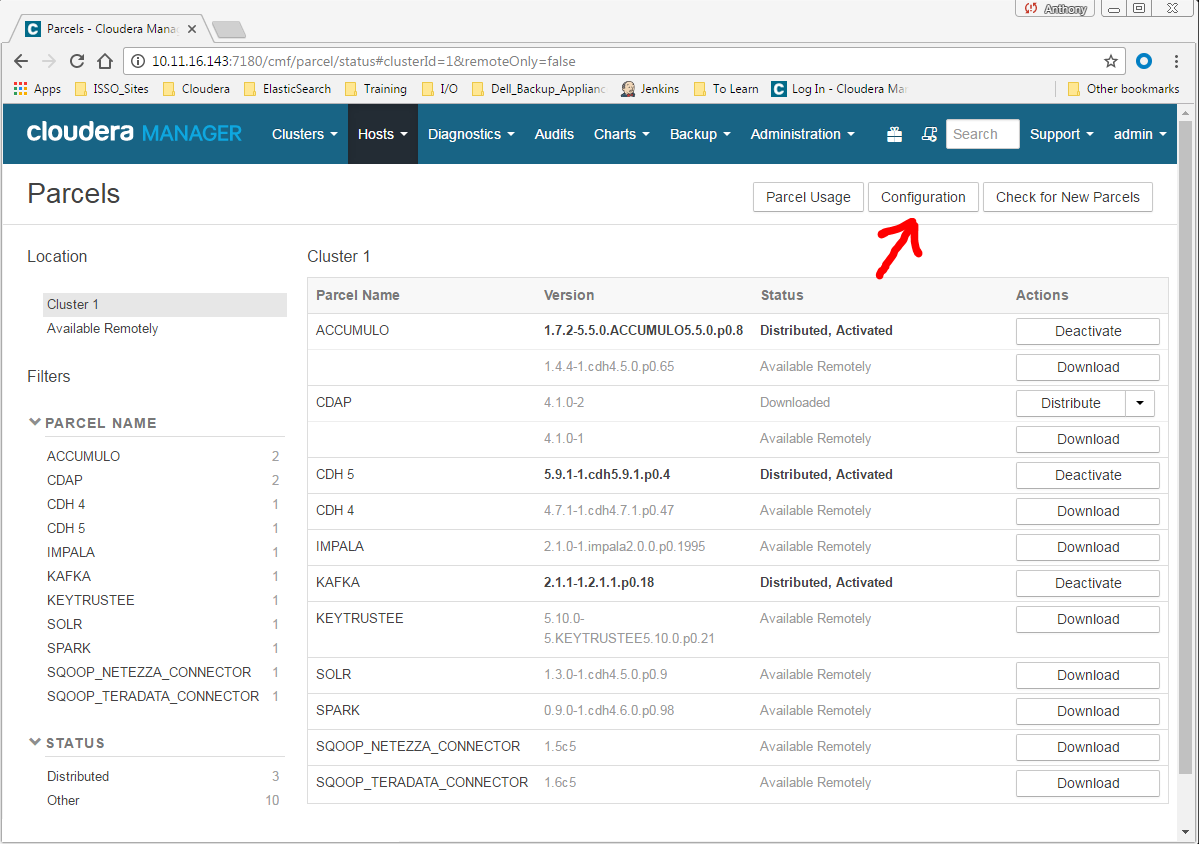
1. Point your browser or use wget to go to <http://spacewalk/cobbler/repo_mirror/cloudera_parcels/cdap/CDAP-4.1.2.jar> and grab the CSD File.
2. Place this file in the **/opt/cloudera/csd** directory on your Cloudera Management Server
3. Set the file ownership to **cloudera-scm:cloudera-scm** with permission **644**.
4. Restart the Cloudera Manager Server: **service cloudera-scm-server restart**

## Installing the Parcel

Once you have the CSD in place, the parcel repo should be defined inside your Cloudera Management Server, however, this location is incorrect. You must change it to point to your local CDAP repo.

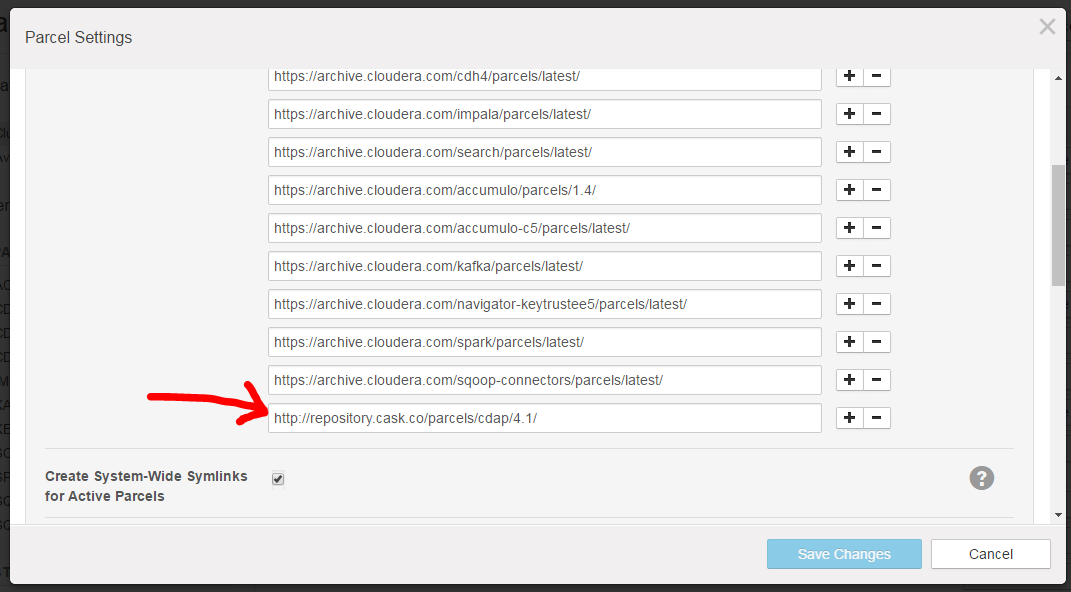
1. Click “Parcel Icon”  in the main navigation bar.
2. Click in the top navigation bar or click **Hosts** and click the **Parcels** tab.
   * 1. Click the **Configuration** button.

NOTES: ***Assuming centos6 the path is:***[***http://repository.cask.co/parcels/cdap/4.1/CDAP-4.1.0-2-el6.parcel***](http://repository.cask.co/parcels/cdap/4.1/CDAP-4.1.0-2-el6.parcel), ***if it’s another OS the parcel portion of the path can be derived from***[***http://repository.cask.co/parcels/cdap/4.1/manifest.json***](http://repository.cask.co/parcels/cdap/4.1/manifest.json)



* + **Menu**
    1. Select **Administration** > **Settings**.
    2. Select **Category** > **Parcels** .

1. In the **Remote Parcel Repository URLs** list, click  to open an additional row.
2. Enter the path to the repository: **http://spacewalk/cobbler/repo\_mirror/cloudera\_packages/cdap/**

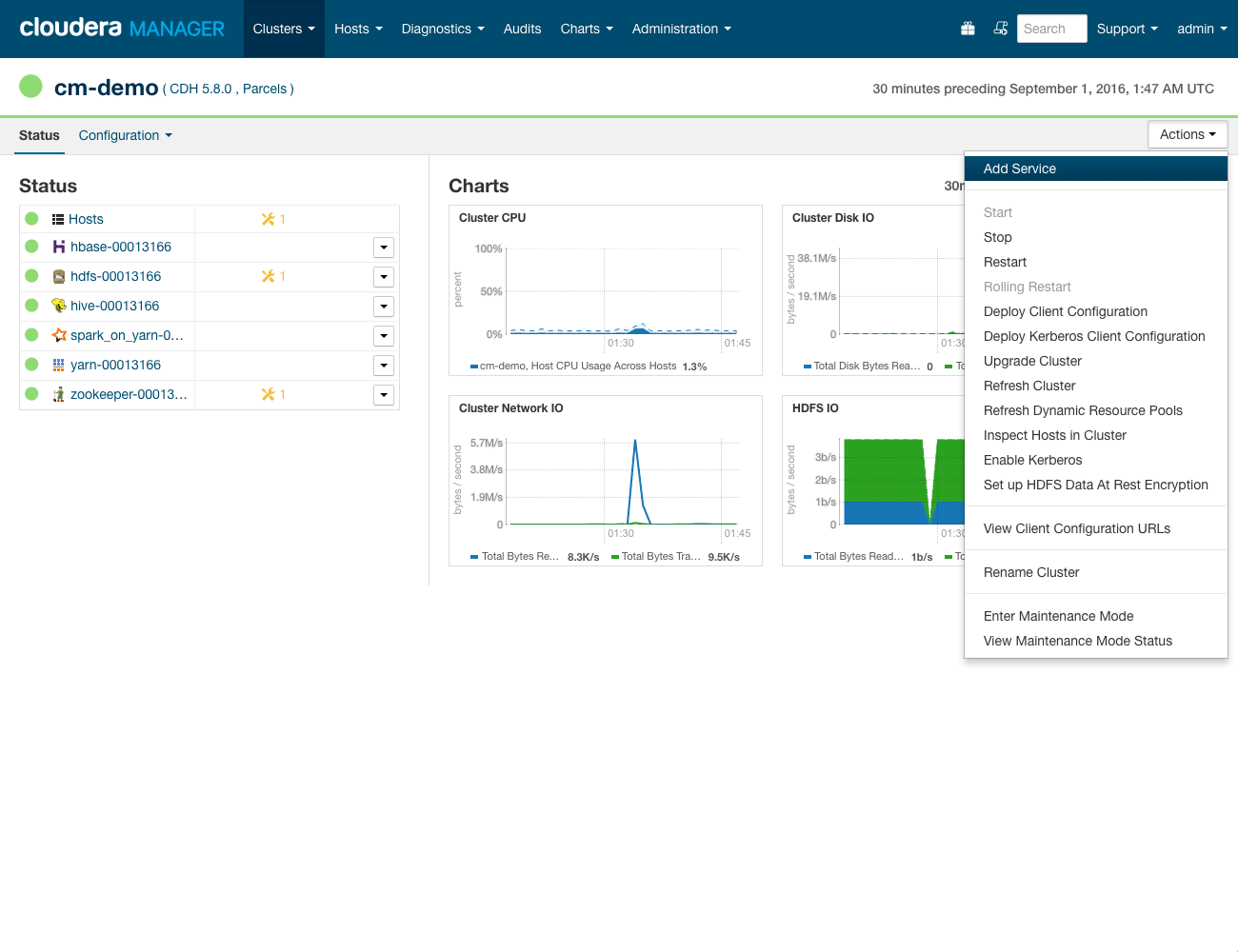


1. Click **Save Changes** to commit the changes.
2. The external parcel should appear in the set of parcels available for download.
3. Next, click **Download**, then **Distribute**, and **Activate** the parcel.

## Starting the CDAP Service

Once the parcel is installed, you need to start the service.

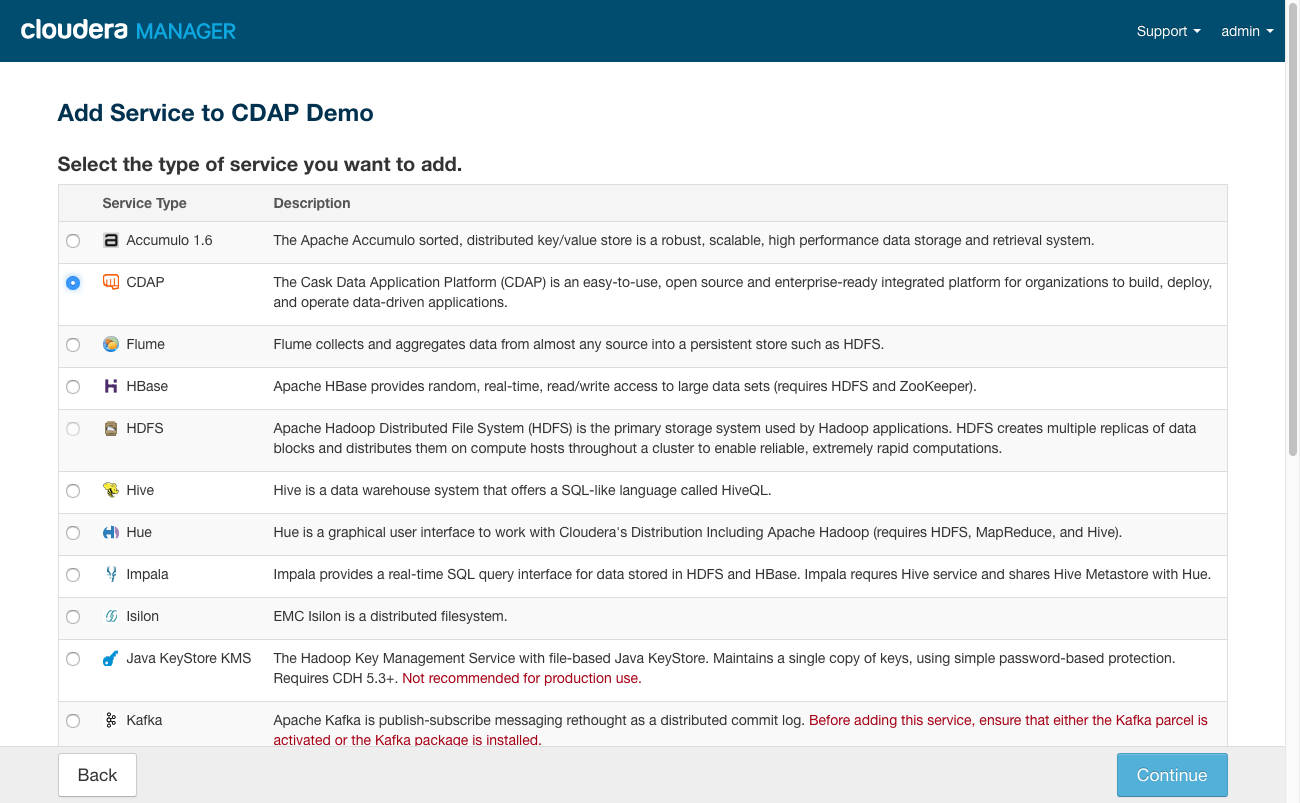
1. Start from the Cloudera Manager Admin Console's Home page, selecting Add Service from the menu for your cluster:

[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-01.png)

**Cloudera Manager:** Starting the Add Service Wizard.

**Add Service Wizard: Selecting CDAP**

Use the Add Service Wizard and select CDAP.

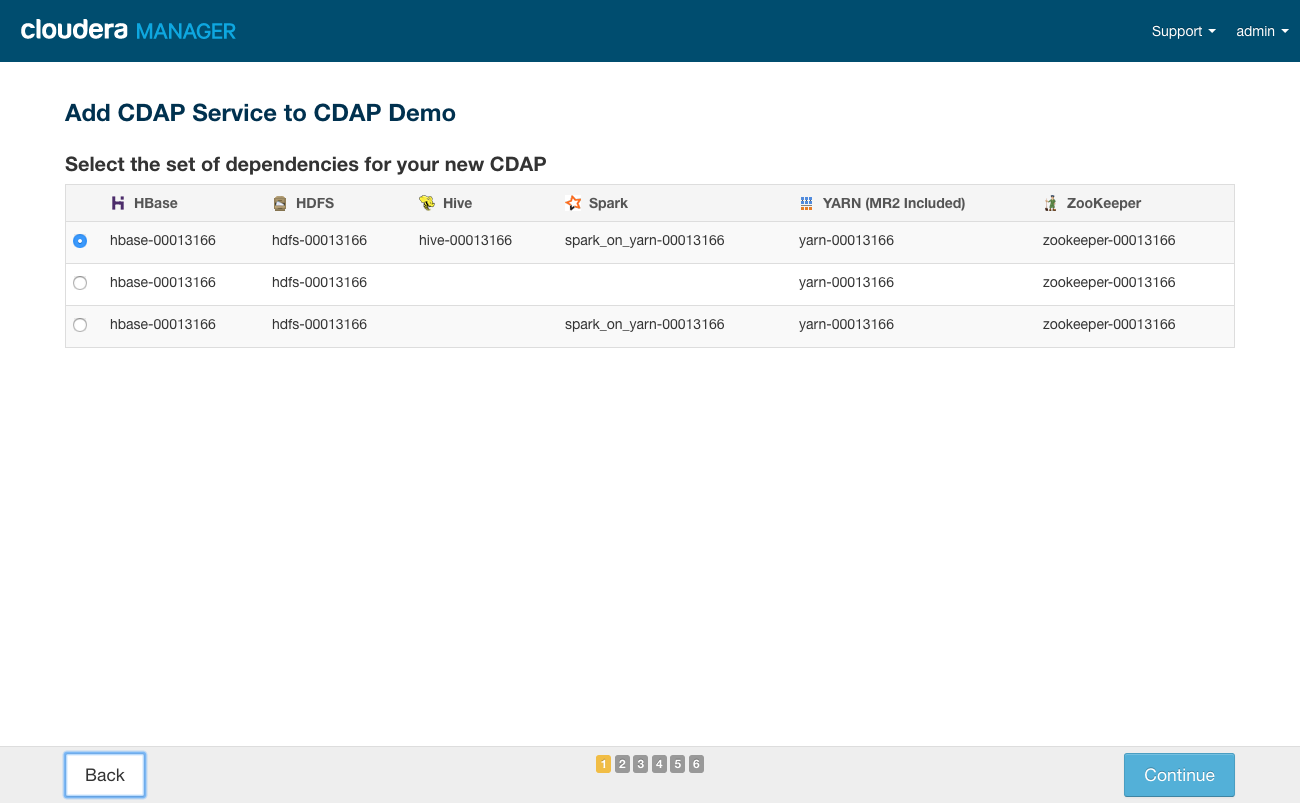
[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-02.png)

**Add Service Wizard:** Selecting CDAP as the service to be added.

**Add Service Wizard: Specifying Dependencies**

The **Hive dependency** is for the CDAP "Explore" component, which is enabled by default. Note that if you do not select Hive, you will need **to disable CDAP Explore** in a later page when you review these changes.

**DO NOT SELECT HIVE OPTION**

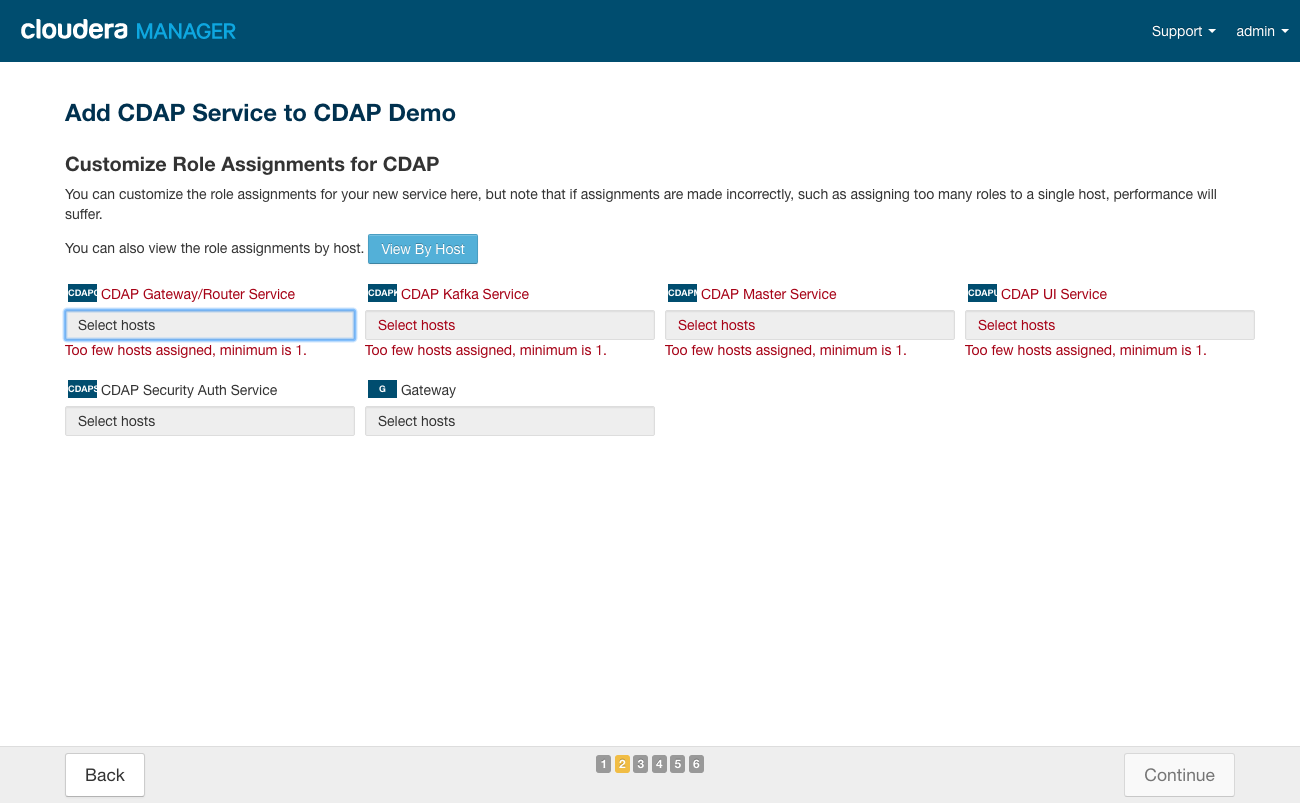
[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-03.png)

**Add Service Wizard, Page 1:** Setting the dependencies (in this case, including Hive).

**Add Service Wizard: Customize Role Assignments**

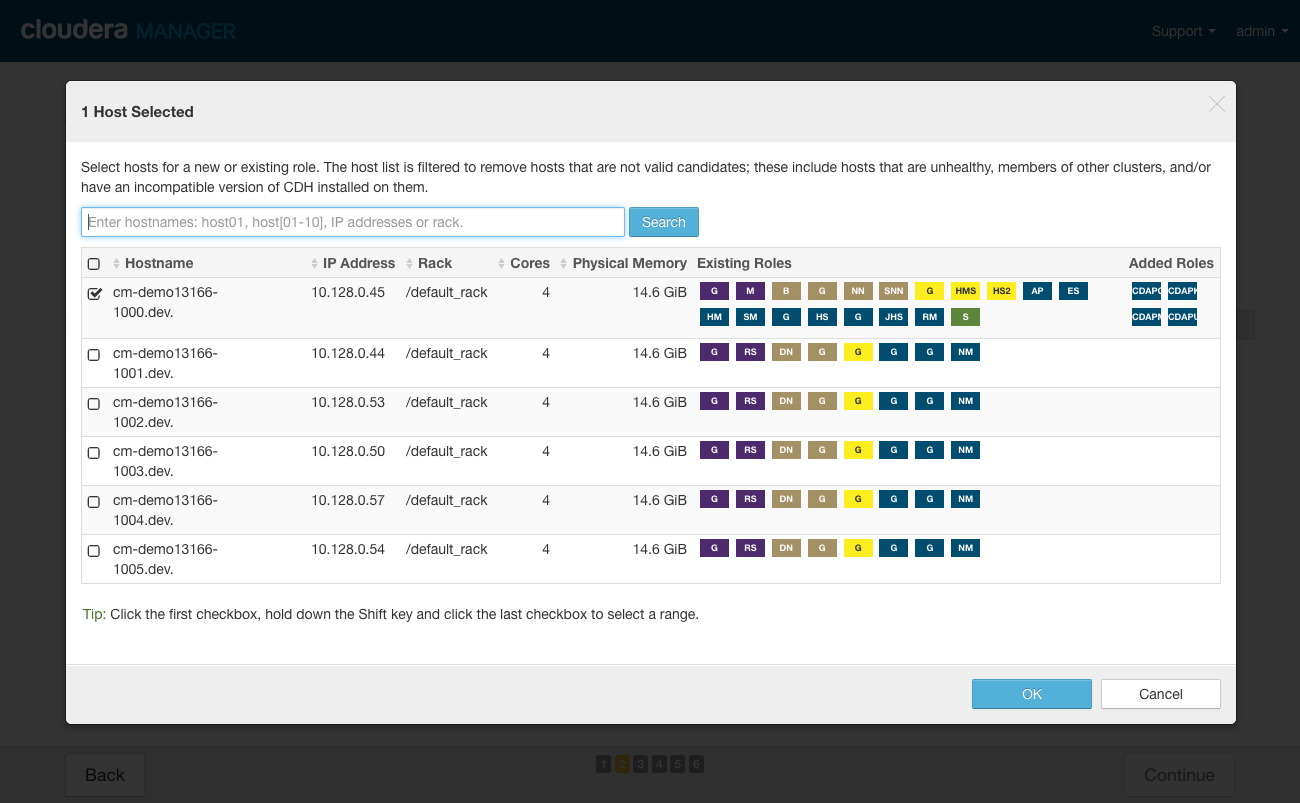
**Customize Role Assignments:** Ensure the CDAP Master role is assigned to hosts co-located with service or gateway roles for HBase, HDFS, YARN, and (optionally) Hive and Spark.

**NOTE: Put UI Service, Router/Gateway Service, and Gateway on the same host**

[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-04.png)

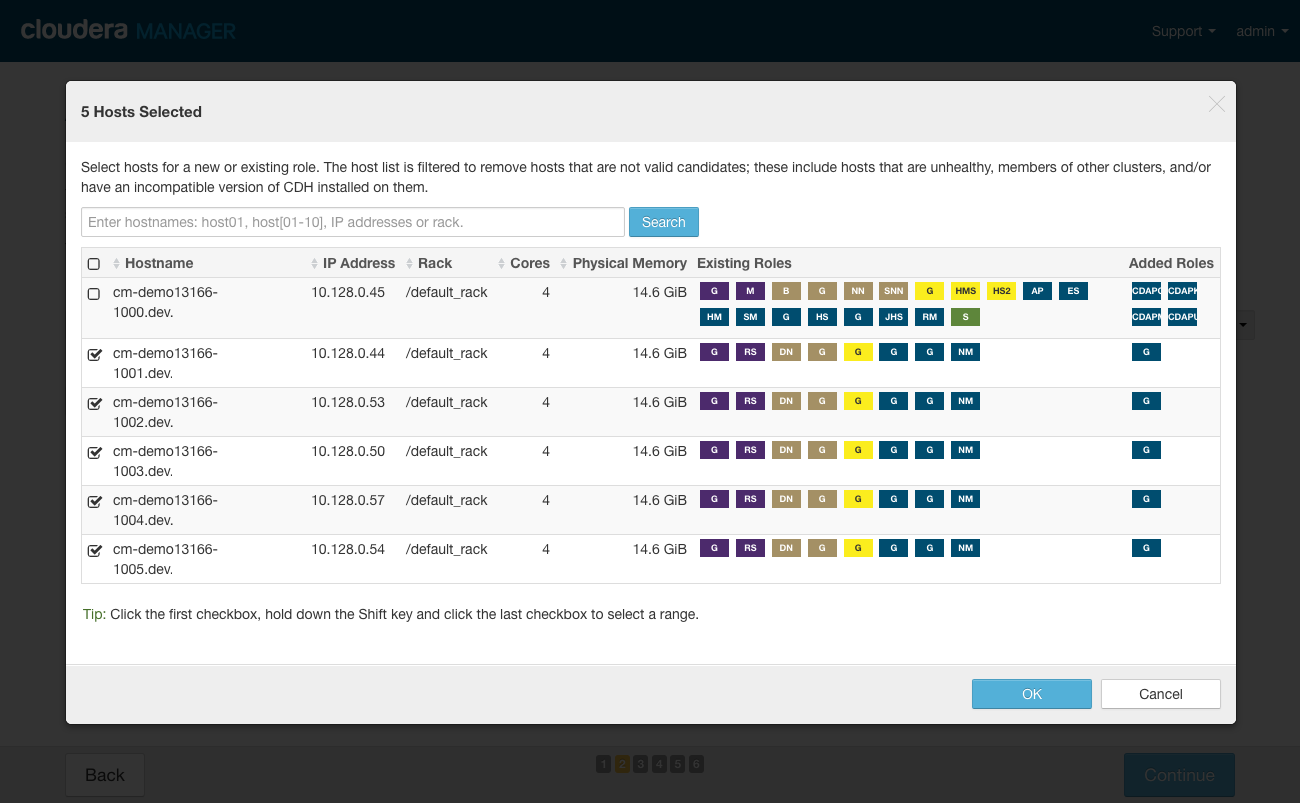
**Add Service Wizard, Page 2:** When customizing Role Assignments, the CDAP Security Auth Service can be added later, if required.

**Add Service Wizard: Customize Role Assignments**

[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-04b.png)

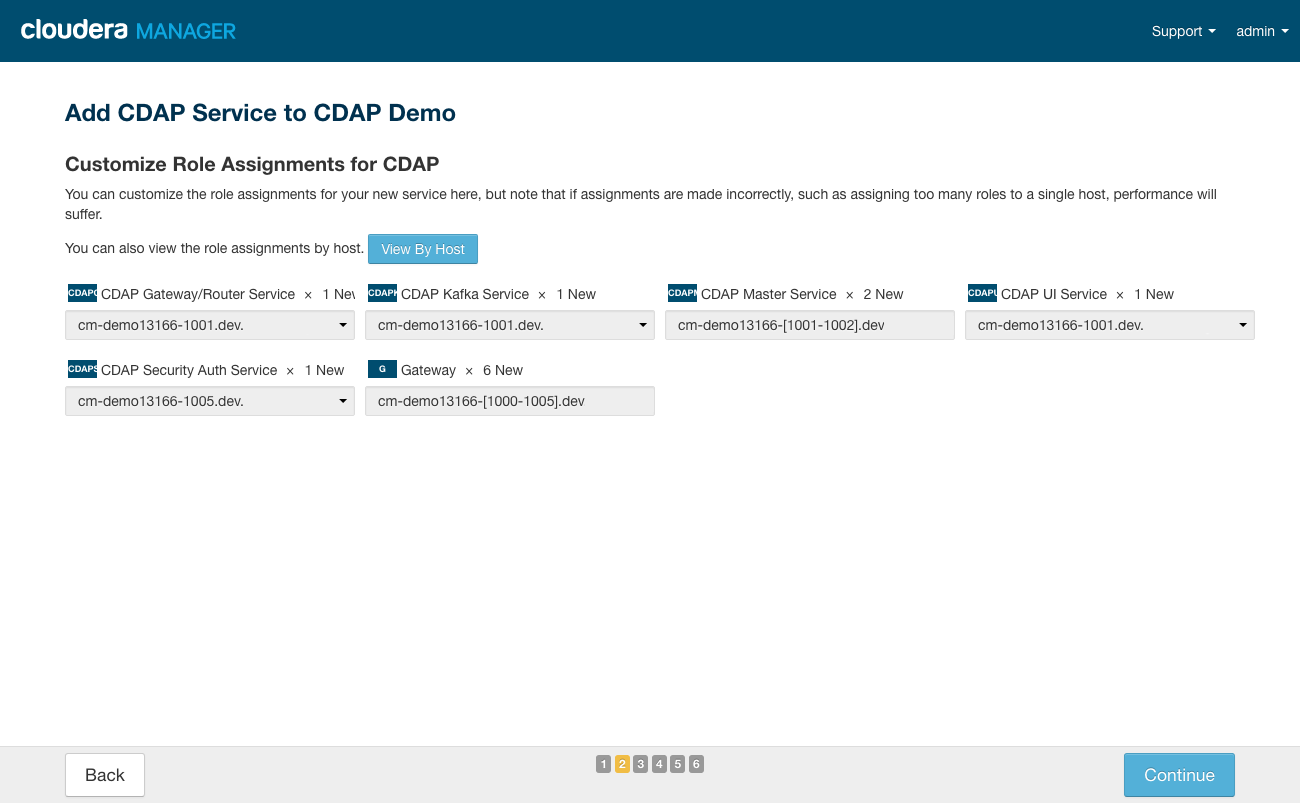
**Add Service Wizard, Page 2 (dialog):** Assigning the CDAP Master Role to a host with the HBase, HDFS, YARN, Hive, and Spark Gateway roles. It could also be on a host with running services instead.

**Add Service Wizard: Customize Role Assignments**

[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-04c.png)

**Add Service Wizard, Page 2 (dialog):** Completing assignments with the CDAP Gateway client added to other nodes of the cluster; it can be added to nodes with CDAP roles.

**Add Service Wizard: Customize Role Assignments**

[](http://docs.cask.co/cdap/3.5.3/en/admin-manual/_images/cloudera-csd-05.png)

**Add Service Wizard, Page 2:** Completed role assignments.

# CASK Marketplace Offline Repository

Because we are using CDAP on an isolated network, we must support the CDAP MarketPlace locally. To do this, we have to host a simple repository.

## Download Packages

Download the packages from the github repo or use the provided zip file.

Install Apache Maven (If not already installed)

* 1. Download package from: <http://www.namesdir.com/mirrors/apache/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz>
  2. tar xzvf apache-maven-3.5.0-bin.tar.gz
  3. Export Maven Path: **export PATH=/opt/apache-maven-3.5.0/bin:$PATH**

## Make Package Metadata (Only necessary if you don’t zip up the Agora directory)

Once you have the directory of plugins, you need to build the packages.json by doing the following:

1. cd into the cask-marketplace directory (from git clone or zip file)
2. cd packager
3. mvn clean package
4. cd ..
5. java –cp packager/target/\*:packager/target/lib/\* co.cask.marketplace.Tool build

## Create a Web Server

The CASK Marketplace requires only a simple http server to serve out the packages. Just link or place the cask marketplace directory in your web path. (Ex. /var/www/cobbler/repo\_mirror/cdap/market

Update the CDAP Market URL within the CDAP Configuration Settings GUI as shown below:

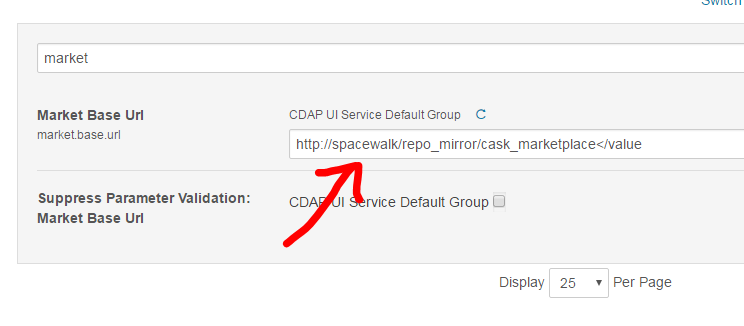


Figure - Market URL

You can just search for ‘market’ in the configuration setting search box.

# SSL Integration

We want to enable SSL for access to CDAP, however, since we are going through a proxy, there are a few additional steps.

## Create the self-signed certificate and key file for User Interface (UI) access

1. **cd to /etc/ssl**
2. **openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout cdap\_key.pem -out cdap\_cert.pem**

## Create the Java Keystore

The CDAP Router and Gateway wanted a JKS keystore. This requires a password to be set, so make sure to annotate the password for input into the CDAP settings field. If no other password is set (the <Enter> is pressed when prompted for a password), the same password will used for the JKS Keystore File Password and the JKS Keystore Key Password.To generate a self-signed keystore run:

1. **cd /etc/ssl**
2. **keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks -storepass password -validity 3650 -keysize 2048**

Configure CDAP SSL Settings

The easiest way to do this is to use the Cloudera Manager interface. The settings are shown below

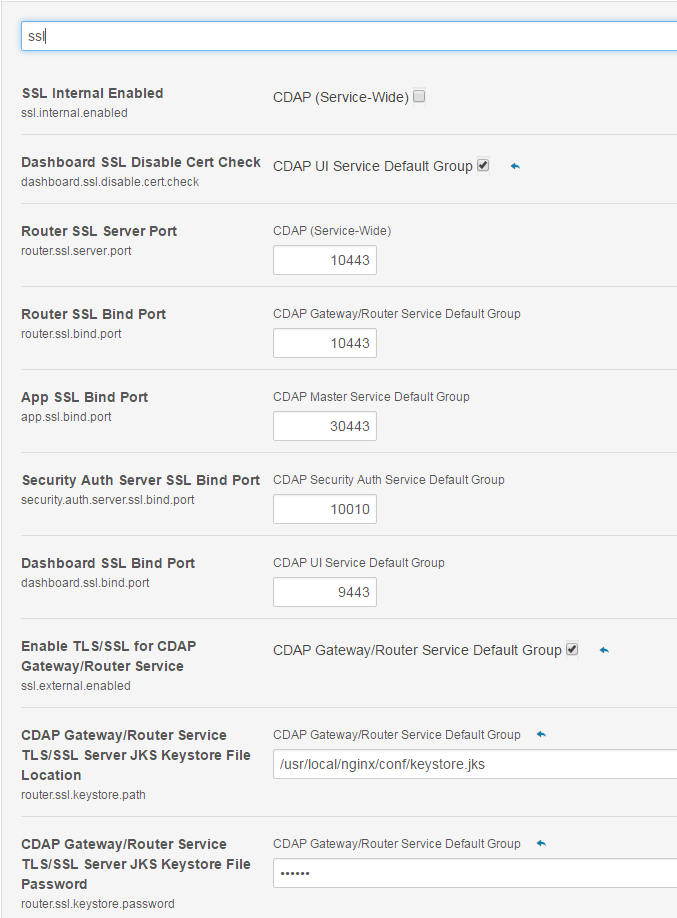


Figure - SSL Settings 1

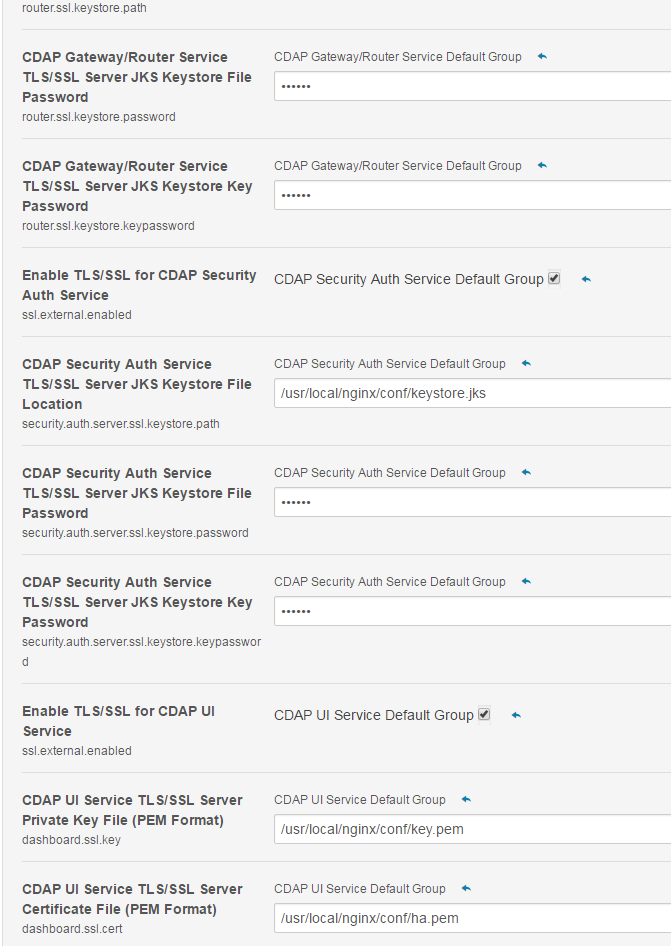


Figure - SSL Settings 2

# Active Directory Authentication

In order to provide authentication services for CDAP UI access, the service must be configured to authenticate against our local Active Directory.

Reference Document: <http://docs.cask.co/cdap/4.0.0/en/admin-manual/security/perimeter-security.html>

The easiest way to make the required changes is to go through Cloudera -🡪 CDAP -🡪 Configuration as shown below:

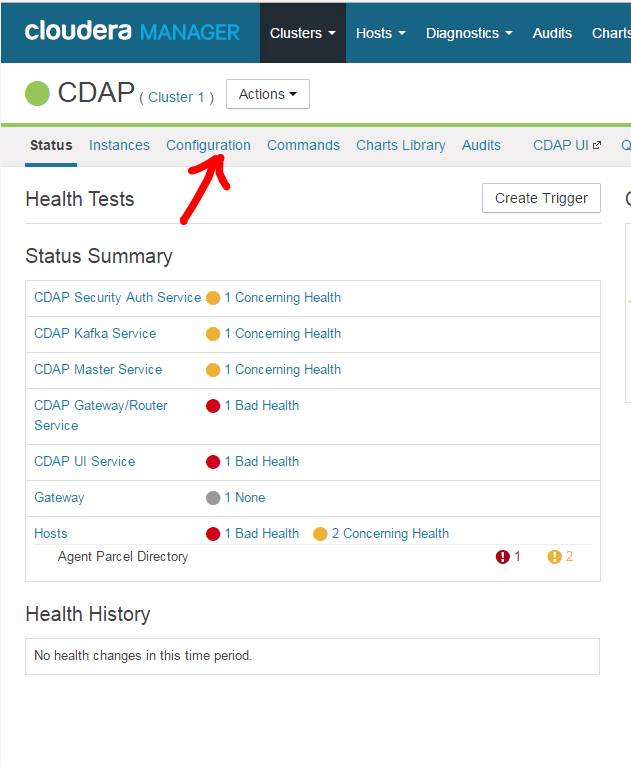


Figure - CDAP Configuration

The easiest way is to use the search box and search for authentication as shown below:

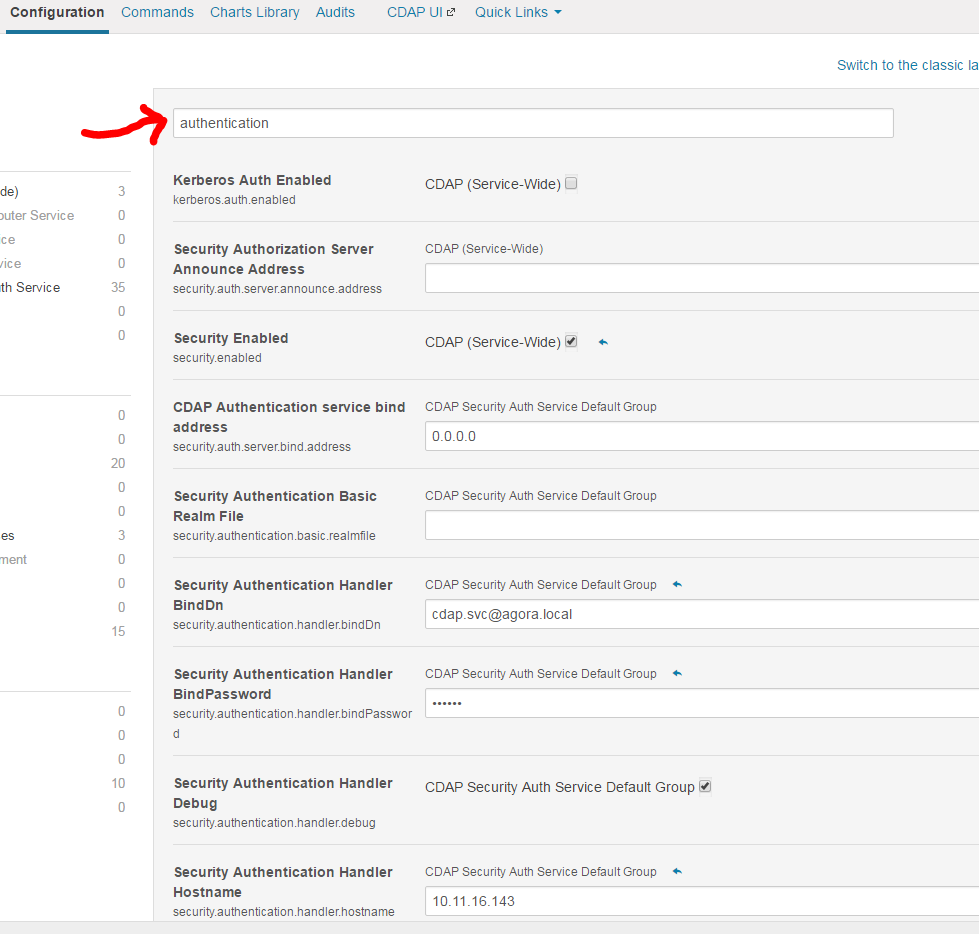


Figure - Search Authentication

The variables below need to be set

## Authentication Configuration Settings

**security.authentication.handlerClassName**: co.cask.cdap.security.server.LDAPAuthenticationHandler

**security.authentication.loginmodule.className:** co.cask.cdap.security.server.LDAPLoginModule

**security.authentication.handler.debug:** false

**security.authentication.handler.hostname:** [fwf-proxy.your.domain](http://ldap.example.com/) (this is the haproxy server which will forward as discussed below)

**security.authentication.handler.port:** 389

**security.authentication.handler.userBaseDn**: OU=Sites,OU=EXAMPLE,DC=example,DC=com

**security.authentication.handler.userRdnAttribute**: sAMAccountName

**security.authentication.handler.userIdAttribute**: sAMAccountName

**security.authentication.handler.userObjectClass**: user

**security.authentication.handler.bindDn:** cdap.svc@your.domain

Note: When creating this account in AD, the generic pass Password1234!@#$

**security.authentication.handler.bindPassword:** password

## Proxy Settings for Authentication

Because the Cloudera cluster sits in an isolated sandbox, a proxy server is required to communicate from routed subnets. We use nginx to proxy inbound traffic to the Cloudera cluster and HAProxy to proxy outbound Active Directory traffic to the domain controller. We will also use SSL which is discussed below, but referenced in the conf file below.

### Nginx Inbound Configuration

Create a conf file for cdap.conf under /etc/nginx/conf.d/cdap.conf with the following contents:

**server {**

**listen 443;**

**server\_name proxyserver.your.domain;**

**ssl on;**

**ssl\_certificate /etc/ssl/cdap\_cert.pem;**

**ssl\_certificate\_key /etc/ssl/cdap\_key.pem;**

**ssl\_session\_cache shared:SSL:10m;**

**location / {**

**proxy\_pass** [**https://192.168.16.143:9443**](https://192.168.16.143:9443)**;**

**proxy\_set\_header Host $host;**

**# re-write redirects to http as to https, example: /home**

**proxy\_redirect http:// https://;**

**}**

**}**

### HAProxy Outbound Configuration

The CDAP Authentication service will point all Active Directory requests to port 389 on the proxy server, which will then route the requests to the RADON/AGORA domain controller. Front end is the listening end.

Create or edit the /etc/haproxy/haproxy.cfg file as follows:

**global**

**log 127.0.0.1 local2**

**chroot /var/lib/haproxy**

**pidfile /var/run/haproxy.pid**

**maxconn 2048**

**user haproxy**

**group haproxy**

**daemon**

**tune.ssl.default-dh-param 2048**

**defaults**

**# mode http**

**# log global**

**# option httplog**

**# option dontlognull**

**# option http-server-close**

**# option forwardfor except 127.0.0.0/8**

**# option redispatch**

**# retries 3**

**timeout http-request 10s**

**timeout queue 1m**

**timeout connect 10s**

**timeout client 1m**

**timeout server 1m**

**timeout http-keep-alive 10s**

**timeout check 10s**

**maxconn 3000**

**###### FRONT END CONFIGS#####**

**frontend ldap**

**bind \*:389**

**default\_backend ldap-backend**

**####### BACK END CONFIGS######**

**backend ldap-backend**

**server dc1.you.domain 192.168.10.10:389**