Install CDH Using Cloudera Manager

System Pre-configuration Checks

Using the steps below, verify that all instances are ready. You must modify them when necessary, which includes installing missing packages and changing kernel tunables or other system settings. You will have to make the following updates on all of your nodes.

1. Update `yum`
2. Change the run level to multi-user text mode
3. Disable SE Linux
4. Disable firewall
5. Check `vm.swappiness` and update permanently as necessary.
    - Set the value to `1`
6. Disable transparent hugepage support permanently
7. Check to see that `nscd` service is running
8. Check to see that `ntp` service is running
    - Disable `chrony` as necessary
9. Disable IPV6
10. During the installation process, Cloudera Manager Server will need to remotely access each of the remaining nodes. In order to facilitate this, you may either set up an admin user and password to be used by Cloudera Manager Server or setup a private/public key access. Whichever method you choose, make sure you test access with `ssh` before proceeding.
11. Show that forward and reverse host lookups are correctly resolved
    - In this lab, we will use `/etc/hosts` Files setting to accomplish this
    - Add the necessary information to the `/etc/hosts` files
    - Check to make sure that File lookup has priority
    - Use `getent` to make sure you are getting proper host name and ip address
12. Change the hostname of each of the nodes to match the FQDN that you entered in the /etc/hosts file.
    - Reboot each of the nodes

Path B install using CM 5.15.x

The full rundown is here. You will have to modify your package repo to get the right release. The default repo download always points to the latest version.

Use the documentation to complete the following objectives:

- Install JDK on all the nodes.
- Install a supported JDBC mysql connector or mariadb connector on all nodes
- On the host that you will install CM:
    - Configure the repository for CM 5.15.2
    - Install CM
    - Install and enable Maria DB (or a DB of your choice)
        - Don't forget to secure your DB installation
        - **You can refer to instructions below for more details on installing Maria DB**
    - Install the mysql connector or mariadb connector
    - Create the necessary users and databases
        - Grant them the necessary rights
    - Setup the CM database
- Start the CM server and prepare to install the cluster through the CM GUI installation process
- Do not continue until you can browse your CM instance at port 7180

Install and Configure MySQL (Maria DB)

Choose one of these plans to follow:

- You can use the steps documented here for MariaDB or here for MySQL.
- The steps below are MySQL-specific.
    - If you are using RHEL/CentOS 7.x, **use MariaDB**.

MySQL installation - Plan Two Detail

1. Download and implement the official MySQL repo
    - Enable the repo to install MySQL 5.5
    - Install the `mysql` package on all nodes
    - Install `mysql-server` on the server and replica nodes
    - Download and copy the JDBC connector to all nodes.
2. You should not need to build a `/etc/my.cnf` file to start your MySQL server
    - You will have to modify it to support replication. Check MySQL documentation.
3. Start the `mysqld` service.
4. Use `/usr/bin/mysql_secure_installation` to:
   a. Set password protection for the server
   b. Revoke permissions for anonymous users

c. Permit remote privileged login
d. Remove test databases
e. Refresh privileges in memory
f. Refreshes the `mysqld` service

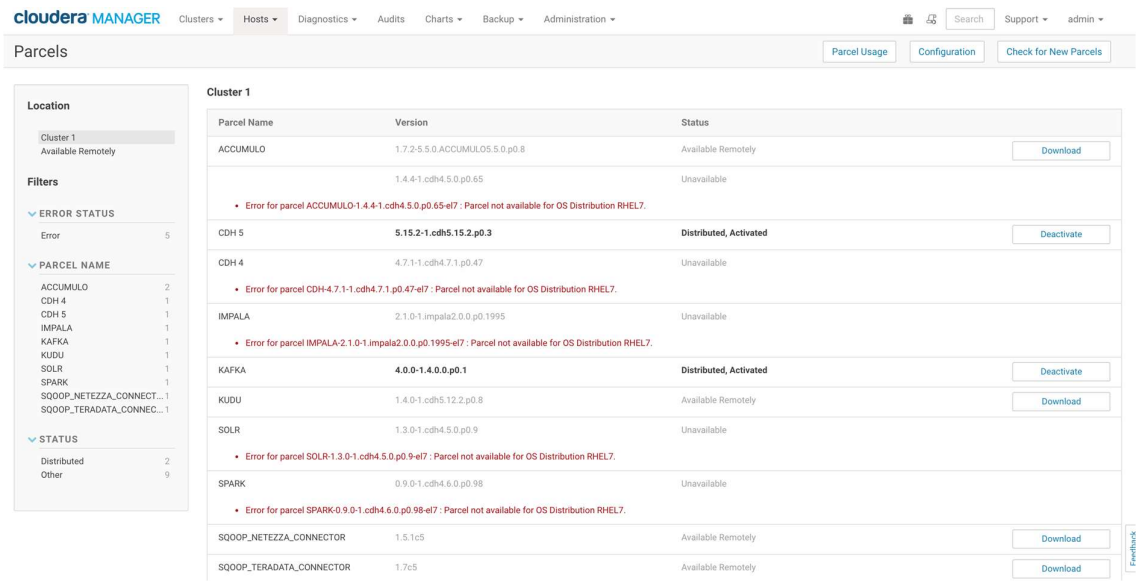
Install a cluster and deploy CDH

Go to the CM GUI installer and install CDH 5.15.2

1. Specify hosts for your CDH cluster installation
2. You may choose to use a common user / password or key based depending on how you have set it up earlier
3. Do not use Single User Mode. Do not. Don't do it.
4. Ignore any steps in the CM wizard that are marked `(Optional)`
5. Install the Data Hub Edition
6. Install CDH using parcels
7. Allow CM to install CM Agent on each of the nodes
8. Allow CM to download, distribute and activate CDH on each of the nodes
9. Choose the type of installation you want
    a. We will require Flume, Hive, Impala and Spark later on
10. Assign roles to each of the hosts: Please refer here for hints.

Install Sqoop, Spark and Kafka

Installing Kafka requires some special attention. You will need to download, distribute and activate the Kafka package before you can add the Kafka service.



Install sqoop from the cluster -> Add service menu.

Install spark1 from the cluster -> Add service menu. Do not choose the standalone. Please select the cluster version with Yarn.

Install spark2 as an additional service. Follow the instructions here:
https://www.cloudera.com/documentation/spark2/latest/topics/spark2.html

Let's test out our cluster

1. Create user "training" in linux and in hdfs.
    a. Give this user sudo capabilities.
        i. DO NOT USE visudo. *Hint: group Wheel*
    b. Create home directory for user "training" in HDFS and set appropriate ownership and access rights
    c. Now, try ssh'ing into the hosts as user "training"

2. In MySQL create the sample tables that will be used for the rest of the test
    a. In MySQL, create a database and name it "test"
    b. Create 2 tables in the test databases: authors and posts.
        i. You will use the authors.sql and posts.sql script files that will be provided for you to generate the necessary tables
    c. Create and grant user "training" with password "training" full access to the test database. (It is ok if you give training full access to the entire MySQL database)

3. Extract tables authors and posts from the database and create Hive tables.
    a. Use Sqoop to import the data from authors and posts
    b. For both tables, you will import the data in tab delimited text format
    c. The imported data should be saved in training's HDFS home directory
        i. Create authors and posts directories in your HDFS home directory
        ii. Save the imported data in each
    d. In Hive, create 2 tables: authors and posts. They will contain the data that you imported from Sqoop in above step.
    e. You are free to use whatever database in Hive.
    f. Create authors as an external table.
    g. Create posts as a managed table.

4. Create and run a Hive/Impala query. From the query, generate the results dataset that you will use in the next step to export in MySQL.
    a. Create a query that counts the number of posts each author has created.
        i. The id column in authors matches the author_id key in posts.
    b. The output of the query should provide the following information:

Source	Output Column Name
Id from authors	Id
first_name from authors	fname
last_name from authors	Lname
Aggregated count of number of posts	num_posts

   a. The output of the query should be saved in your HDFS home directory.
      i. Save it under “results” directory

5. Export the data from above query to MySQL
   a. Create a MySQL table and name it “results”
      i. Make sure it has the necessary columns of matching type as the results of your query from above
   b. The table should be created under the database “test”
   c. Finally, export into MySQL the results of your query.

Setup your cluster to begin your analysis

- Log into Hue as user “centos” with password “centos”
  - This will create a HDFS directory and give administrative rights for the user
  - Go to the administration and create user “training”
    - Give full rights to this user
- From the bit.ly folder (http://bit.ly/SKT_COMB_LAB) , download the yelp dataset.