<http://www.cloudera.com/documentation/enterprise/5-3-x/topics/release_notes.html>

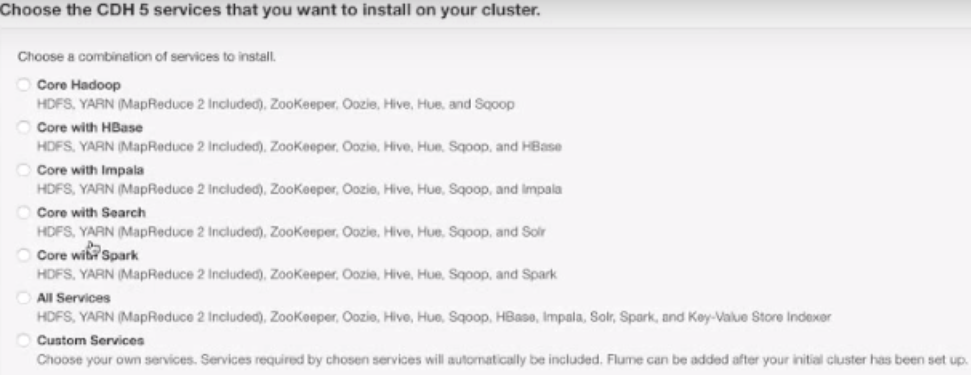
|  |  |  |
| --- | --- | --- |
| **S.no** | **Topic** | **Desc** |
|  | **Web console** |  |
|  | CM Path | <http://archive.cloudera.com/cm5/> |
|  | CDH Path | <https://archive.cloudera.com/cdh5/>   1. Open repo-as-tarball. To get the supporting version details |
|  | Cloudera community - for help | <https://community.cloudera.com/> |
|  | **Cloudera document for CM Installation** | <http://www.cloudera.com/documentation/enterprise/latest/topics/cm_ig_install_path_b.html> |
|  | SSH Issue | Amazon Web Services - Setup SSH Tunneling and Foxyproxy  <https://www.youtube.com/watch?v=HlfsKWODDwM&index=9&list=PLf0swTFhTI8rvUye27NdkTte3ldym2Nlu> |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **General Information** |  |
|  | **Version** | Version CDH 5.2.1 – Training (released on Dec-2014)  Version CDH 5.3.8 – Actual (released on Oct-2015)  Hadoop 2.5 (CDH 5.3.8 handles Hadoop 2.5) |
|  | **Order of add service** | Add service/Configure ECO System in this order   1. Zookeeper (HDFS has dependency on this for HA Setup) 2. HDFS 3. YARN (MR2 included) 4. Etc (explained below) |
| 1 | **NTP**  **<Clock offset>** | **Network Time Protocol (NTP)** is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.  In general, if multi server environment placed in different timezone then we need to set NTP on all the hosts. No need to customize NTP service, just install default NTP Service, it will automatically handle the time management |
| 1.1 | Check NTP is configured in your server  *<Start NTP and on>* | The below result shows ntp is available but it is off. So no need to install  >sudo su root  >service ntpd status  >service ntpd start  >chkconfig --list ntpd  >chkconfig ntpd on;  >chkconfig --list ntpd  ***<It is important that we need to do this in all the nodes in the cluster>*** |
| 2 | **Cloudera has 3 Components** | 1. Cloudera Distribution of Hadoop (CDH) 2. Cloudera Management Service 3. Cloudera Manager (CM) |
| 2.1 | Cloudera Distribution of Hadoop (CDH) | CDH consolidates all the Hadoop ecosystem tools as one package (or) parcel |
| 2.2 | Cloudera Management Service  *<Additional Services>* | Status Summary   1. Host Monitor 2. Event Server 3. Reports Manager 4. Alert Publisher 5. Service Monitor |
| 3 | Architecture | Master/Slave  Master – Cloudera Server will be installed  Slave – Cloudera Agent will be installed |
| 4 | Log path | URL Connectivity related issue  /var/log/cloudera-scm-agent  /var/log/cloudera-scm-server  DataNode related issue  /var/log/hadoo-hdfs |
| 5 | Cloudera Download, install and configure | 1. Cloudera Download the package (it has all eco systems) 2. Cloudera Install also a package (it has all eco systems) 3. Cloudera Configure. We can configure eco system based on our needs |
| 6 | CDH supported installation | 1. Core Hadoop: Zookeeper, HDFS, Yarn (MRV2), Oozie, Sqoop, Hive, Hue 2. Core with Hbase 3. Core with Impala 4. Core with Solr 5. Core with Spark 6. All Services 7. Custom Service |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Cloudera Manager pre-request** | **In addition to the following steps, also Refer** “**Cloudera document for CM Installation” – Path given above** |
| 1 | Password less login across the nodes | Make sure to generate ssh-keygen on master and move to all slave nodes. Test all the slaves are connected from master without password |
| 2 | Telnet | #Install telnet on all the nodes  > sudo yum -y install telnet |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Cloudera Manager (CM) Download**  ***<only on masternode>*** | Cloudera Manager is the industry's trusted tool for managing Hadoop in production |
| 1 | Linux Path | >ssh hduser@hdmaster |
| 2 | CM Path | <http://archive.cloudera.com/cm5/>   1. Go to Installer folder 2. Go to 5.3.8 folder (I’m installing version 5.3.8 ) 3. Right click and copy link on ‘cloudera-manager-installer.bin’ 4. >wget *<paste the above path>* |
| 3 | Chmod | By default the newly downloaded won’t have the execute permission. So chmod  >ls –ltr <after download>  >chmod +x cloudera-manager-installer.bin |
| 4 | Execute the CM Installer as sudo | It is mandatory to run the CM installation file as sudo from hduser  >sudo ./ cloudera-manager-installer.bin |
| 5 | telnet using hostname &  Default port: 7180  **<Pre-request>**  **Stop the firewall**  **Start the services**  Off the Firewall in **all** the nodes  stop and disable Firewall services.  Also…  *<Start cloudera-scm services>* | #stop and disable Firewalls and make sure the chkconfig is off. Start cloudera-scm-server and chkconfig is on for the same  >sudo su root  >**service iptables status;**  **>service iptables stop;**  **>service ip6tables status;**  **>service ip6tables stop;**  **> service cloudera-scm-server status**  **> service cloudera-scm-server start**  >service cloudera-scm-agent status  >service cloudera-scm-agent start  >service cloudera-scm-server-db status  >service cloudera-scm-server-db start  **>service mysqld status**  **>service mysqld start**  >service ipchains status;  >service ipchains stop;  **>chkconfig --list iptables**  **>chkconfig iptables off**  **> chkconfig --list ip6tables**  **>chkconfig ip6tables off**  **> chkconfig --list cloudera-scm-server**  **>chkconfig cloudera-scm-server on**  > chkconfig --list cloudera-scm-agent  >chkconfig cloudera-scm-agent on  > chkconfig --list cloudera-scm-server-db  >chkconfig cloudera-scm-server-db on  > chkconfig --list mysqld  >chkconfig mysqld on  > chkconfig --list ipchains  > chkconfig ipchains off |
| 5.1 | Disable SELINUX (manual)  Firewall | >vi /etc/selinux/config  SELINUX=disabled  >sestatus -- check after reboot |
| 5.2 | telnet using hostname &  Default port: 7180 | # Get the hostname  >hostname –v  >hostname –f  >telnet <hostname> <default port#:7180>  Ex:>telnet 52.11.253.131 7180  Here 52.11.253.131 is the hostname and 7180 is port # |
| 5.3 | **How to handle telnet failure – part1** | Don’t do telnet or http immediately, the server has to do some configuration in background. So wait for few mins before try telnet and http |
| 5.4 | **How to handle telnet failure – part2** | If something went wrong with telnet then there should be an issue with firewall. So disable firewall using following commands and try telnet again  Off the Firewall in all the nodes  stop and disable Firewall services.  Also…  *<Start cloudera-scm services>*  <Details given below> |
| 5.5 | **How to handle telnet failure – part3** | If telnet still not working for 7180 then try with different port # like 22, etc  >telnet <hostname> <default port#:22>  Ex:>telnet 52.11.253.131 22  Then try back for 7180  Ex:>telnet 52.11.253.131 7180 |
| 6 | Login to Cloudera Manager (CM) in URL | Use http (not https). Default uid, pwd: admin/admin  <http://hostname:7180>  Ex: <http://52.11.253.131:7180> |
| 6.1 | How to handle http issue in linux | >sudo su root  >service httpd start  >chkconfig -- list httpd  >chkconfig httpd on  >service httpd restart  >exit -- from root |
|  | **Cloudera Manager configuration using URL** | <http://hostname:7180>  Configuration will get all the available eco systems in hadoop |
| 1 | Login | Login to URL using user/pwd: admin/admin |
| 2 | Version | Select the cloudera manager version you want to install |
| 3 | Master and Slave | It will ask all the master and slave node ips. Give all the required ips and make sure all the ips are one and connected from cloudera |
| 4 | Pre-request | All the nodes should up and running. Also make sure firewall is should be in off status |
| 4.1 | Off the Firewall in all the nodes  stop and disable Firewall services.  Also…  *<Start cloudera-scm services>* | #stop and disable Firewalls and make sure the chkconfig is off. Start cloudera-scm-server and chkconfig is on for the same |
| 5 | Provide Login info for installation | # We can’t give root password in prod. So give some admin user who can ssh to root or other nodes to install s/ws  Username:hduser  Password: hadoop1 |
| 5.1 | Validate the admin user | # Validate the admin user that you have given above, has all the required privileges like ‘wheel’, password less login, sudoers, etc  >id hduser; |
| 5.2 | Password less login | Login to given user and  >sudo ssh to all othernodes without password |
| 5.3 | id username | #To check the user is part of wheel  >id username  Ex: >id hduser |
| 5.4 | Sudoers chk | #Check the user can apply sudo command . Make sure %wheel is uncommented.  Sometimes sodoers might be big file, so you won’t have access.. so chk with SA s whether user has sudo access  >login to user  >sudo view /etc/sudoers |
| 5.5 | Click next to Start installation | #once all informations are verified, click continue button, so that CM will start to install software in all the nodes |
| 6 | **Log validation** | # Location for Log file for this installation  >sudo su root  >cd /var/log  >ls -ltr |
|  | **Cloudera Manager configuration using URL** | 1. Add Service: Add zookeeper first because HDFS has dependency on zookeeper to have High availability. Assign zookeeper to at least 3 to 4 master nodes as it is the coordinator 2. Add service: Add HDFS |
| 1 | Chose the eco systems to be installed | #default packages will be available like (or) choose custom   1. HDFS, MR, Hive, HUE, Zookeeper, Oozie, Hbase 2. HDFS, MR, Hive, HUE, Zookeeper, Oozie, Impala 3. Custom 4. Etc   #I’ve opted for custom and selected only HDFS |
| 2 | Role Assignment for HDFS | #Assign Role to each node. Keep only Management related roles in master node and move other roles to other nodes. Keep at least 3+ Master nodes, so that you can keep One dedicated master node for HDFS and another dedicated master node for HBase   1. Namenode – Non management, so move to hdslave1 2. Datanode – If you have 10 nodes, then keep 3 for master and keep datanode in remaining 7 nodes. Don’t disturb Master node for DataNode usage 3. Secondary Namenode– Non management, so move to hdslave2 4. Balancer – Non management, so move to hdslave1 5. HttpFS 6. NFS GateWay: configure XML files (core-site.xml, hdfs-site.xml, etc). details given below |
| 2.1 | NFS GateWay  XML Configuration file path.  Different path followed by Cloudera and default Hadoop apache | **#NFS GateWay**  **Cloudera path1:**  /var/run/cloudera-scm-agent/process  **Cloudera path2:** Under process (above path) you can find folders like below based on your configuration. It has XML files  Hdfs-NAMENODE  Hdfs-DATANODE  Hdfs-SECONDARYNAMENODE  **Hadoop apache default path for XML:**  /etc/Hadoop/conf |
| 3 | Role Assignment for Cloudera Management Service | #Assign ‘Cloudera Management service’ Role to each node.   1. Host Monitor 2. Event Server 3. Reports Manager 4. Alert Publisher 5. Service Monitor |
| 4 | Cluster Setup  *<use custom DB>* | Database setup. Cloudera Manager needs some DB to store the report information. So it provide embedded (inbuild) --or—we can choose custom DB if we have already  Choose:   1. Custome DB : MySQL, PostgreSQL, Oracle -- Mysql selected 2. Use Embedded DB: |
| 4.1 | Database Host name and Port # | It is very Important to mention DB Hostname **and Port #. The port # will vary based on DB you select.**  Ex: Google : MySQL Port Number, It will show 3306. There are more than one port number for Oracle, choose 1521 out of it. So select the DB first and identify corresponding Port |
| 4.2 | Validate the DB Host Nameand Port # | >telnet hostname port#  Ex:  >telnet 52.11.253.131 3306 |
| 4.3 | Database and user in DB | # Cloudera uses default DB and user called ‘rman’  # Make sure rman DB and user are available in the corresponding DB  # Make sure all the grants are provided to user  >create database rman;  >create user ‘rman’ identified by ‘hadoop1’;  >grant all on rman.\* to rman  Syntax: > grant all on db.\* to user; |
| 4.4 | Password | # provide DB password. Either mysql, oracle, PostgreSQL – password  #Note: don’t create new password, use the existing DB password  # Choose ‘show password’ to check  #i’m giving hadoop1 as my MySQL login password is haddop1 |
| 4.5 | Test connection | Test the DB connect, only then it will allow to continue |
| 5 | **Cluster Review for default setup** | 1. HDFS Block Size: Default: 128 MB 2. DataNode Failed Volumes Tolerated: Default : 0 3. DataNode Data Directory: Default: /dfs/dn 4. NameNode Data Directory: Default: /dfs/nn 5. HDFS Checkpoint directory: Default: /dfs/nn 6. Alerts: Mail server hostname : Default : localhost 7. Alerts: Mail server username & pwd: empty 8. Alerts: Mail message receiptnets: provide mail group 9. Custom Alert scripts: if any 10. Host Monitor Storage Directory: Default :/var/lib/cloudera-host-monitor 11. Service Monitor Storage Directory: Default: /var/lib/cloudera-service-monitor |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Cloudera Manager (CM) UnInstallation** |  |
| 1 | Linux Path | >ssh hduser@hdmaster  Once the installation is done, then the uninstall file will be automatically loaded in the following path  /usr/share/cmf/uninstall-cloudera-manager.sh |
| 2 | Command to uninstall | >sudo ./usr/share/cmf/uninstall-cloudera-manager.sh  >sudo yum remove ‘cloudera-manager-\*’  >sudo yum clean all |
| 3 | Add/Remove services in CM | # Any time we can add/remove eco system from Cloudera Manager  Note:  We can’t directly remove zookeeper. Because Hbase, hive and MapReduce will sit on top of zookeeper, so remove them first before remove Zookeeper |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Cloudera Manager Troubleshoot the issue (Known issue)** |  |
| 1 | Connection refused for CM URL – Part 1 | #Connection refused to Cloudera Manager  <https://forums.aws.amazon.com/thread.jspa?messageID=704830>   1. Make sure telnet works. >telnet hostname 7180 2. >service cloudera-scm-server status 3. If cloudera-scm-server is running then try to access the web server locally “wget localhost:7180” or "curl localhost:7180". If that works you've got a firewall problem. 4. > service cloudera-scm-server start – if not running already 5. Check the log file 6. /var/log/cloudera-scm-server/cloudera-scm-server.log 7. If no log file then, try : cloudera-scm-server.out |
| 1.1 | Connection refused for CM URL – Part 2  *<stop firewall and start cloudera-scm services>* | >service cloudera-scm-server status  >service cloudera-scm-agent status  >service cloudera-scm-server-db status  >chkconfig --list cloudera-scm-server  >chkconfig --list cloudera-scm-agent  >chkconfig --list cloudera-scm-server-db |
| 1.2 | Connection refused for CM URL – Part 3 | #Still getting issue after cloudera-scm-server start, then check the log in the below path  <https://forums.aws.amazon.com/thread.jspa?messageID=704830>  >sudo ls -l /var/log/cloudera-scm-server  > cd /var/log/cloudera-scm-server  >cat cloudera-scm-server.out  #Below is one of the issue that I’ve faced  Error creating bean with name 'com.cloudera.server.cmf.TrialState' |
| 1.3 | Connection refused for CM URL – Part 4  *<Some armature commands>* | [root@52 cloudera-scm-server]# ping localhost  [root@52 cloudera-scm-server]# telnet localhost 22  [root@52 cloudera-scm-server]# hostname  [root@52 cloudera-scm-server]# telnet 52.11.253.131 22  [root@52 cloudera-scm-server]# telnet 52.11.253.131 7180 |
| 2 | CM – Health Issues – Part 1  <Clock offset issue> | **Network Time Protocol (NTP)** is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.  In general, if multi server environment placed in different timezone then we need to set NTP on all the hosts. No need to customize NTP service, just install default NTP Service, it will automatically handle the time management. |
| 2.1 | CM – Health Issues – Part 2  <Clock offset issue>  <To be done on all the nodes> | # How to fix clock offset issue?  Install NTP Service, it will automatically resolve the issue. Some times even it has been already installed, it may be off. So make sure the installation and turn on.  #Steps to Install NTP Service:   1. >service ntpd status 2. If service not available then start it using below command 3. >service ntpd start 4. >chkconfig --list ntpd 5. If ntpd is off then on it using below command 6. >chkconfig ntpd on |
| 3 | CM – Health issues  *< Namenode connectivity Bad issue>* | 1. Make sure the active name node is connected. Check the log. 2. Make sure the password less login works fine… if not, it may create issue 3. If no jobs are running, restart the HDFS service once. Sometime it will automatically resolve the issue 4. >Telnet slavehost 7180… if connection refused then it might be a firewall issue between the hosts |
| 4 | CM – Health Issues  *< DataNode is not connected to one or more of its NameNode(s)>* | There are multiple reasons   1. Datanode where we want to store data should be given read and write access to other users. so change the mode of the directory as follows: >sudo chmod 755 /dfs/dn 2. sdf |
| 5 | **HDFS Canary**  *<Canary test failed>* | Canary Tests are minimal tests to quickly and automatically verify that everything you depend on is ready. You run Canary tests before other time-consuming tests, and before wasting time investigating in your code when the other tests |
| 6 | MapReduce and other Jobs are running in local mode not in cluster mode issue | Assign the **Gateway roles** to all the nodes in MapReduce, so that the job will run in cluster mode.  CM Home -> MapReduce -> Instance -> Add roles (button) – Go to Gateway and select all nodes  # Things to be done after add new services  If you run MapReduce job immediately after add MapReduce service then job will run in local mode instead of cluster mode. |
| 7 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Add/Remove/ Enable Services , Roles** | # The below diagram is not correct for Pig as pig is available in Slave but not in master |
|  | **Dependency to add service** | **Zookeeper:**   1. Zookeeper – No dependency 2. Post-request: HDFS, Oozie, HBase   **HDFS:**   1. Pre-request: Zookeeper – 2. HDFS 3. Post-request: Yarn/MapReduce   **YARN/MRV2:**   1. Pre-request: HDFS, Zookeeper 2. Yarn/MRV2 3. Post-request: Hive   **Hive:**   1. Pre-request: HDFS, Yarn/MRV2, zookeeper (to interact HDFS with Hive) 2. Pre-request: DB for metastore (MySQL, Oracle) 3. Hive   **HBase:**   1. Pre-request: Zookeeper, HDFS 2. HBase   **Oozie:**   1. Pre-request: HDFS, YARN, Zookeeper (works with tools like Hbase) 2. Oozie 3. Post-request: Hue   **Hue:**   1. Pre-request: ZooKeeper, HDFS, YARN (MRV2), Hive, Oozie 2. Hue   **Sqoop: ?**  **Flume:**   1. Pre-request: HDFS, *<need to check anything else required>* 2. Flume   **Tez:**   1. Pre-request: Only Hortonworks supports Tez. Cloudera will not support it |
|  | Dependency on Master Node | A sample Cluster to understand the dependency   1. GW – Gateway (Both in Master & Slave) 2. CM Services – Cloudera Manager Service 3. CM\_HM - Host Monitor 4. CM\_ES - Event Server 5. CM\_RM - Reports Manager 6. CM\_AP - Alert Publisher 7. CM\_SM - Service Monitor 8. NN – Namenode (Only in Master) 9. SNN – Secondary Namenode (Only in Master) 10. HA – High availability (Only in Master) 11. DN – DataNode (Only in Slave) 12. MySQL 13. Yarn – Resource Manager (RM), Job History Server(JHS) ?, NodeManager (NM) 14. Hive (Only In Master) 15. Pig (Client Only eco system) 16. Sqoop (Client only ecosystem) – Prerequest MySQL JDBC/Oracle JDBC connector setup and link JDBC connector to Sqoop library. 17. HBase (Both in master (called as HMaster) & Slave (called as HRegionServer). 3 Masters are highly desired, Configure HMaster on server that are running Zookeeper, HRegionServer on all slave that are running HDFS and YARN) 18. Oozie 19. Flume: *<I’ve installed ‘flume agent’ in slaves, need to verify whether it required to be installed in master too, if so how many master>* 20. ZK – Zookeeper (Only in Master. 3 to 5 master nodes, keep it in dedicated server for better performance) 21. Hue (Only in Master, Only one Node – where you don’t hv nn,snn,ha) 22. Spark: History server (HS): Need to select one history server. *<Hope it should be only one master, To be verified>* 23. Client machine: sqoop – <Can we install it in Master?> -- To be verified 24. Master 1: GW, CM\_HM, CM\_ES, CM\_RM, CM\_AP, CM\_SM 25. Master 2: GW, NN, ZK 26. Master 3: GW, SNN, ZK 27. Master 4: GW, HA, ZK 28. Master 5: GW, Hue, RM, 29. Master 6: GW, MySQL, Hive (hive metastore is on MySQL), Oozie (Oozie repository is on MySQL), 30. Master 7: GW, 31. Slave 8: GW, DN, NM 32. Slave 9: GW, DN, NM 33. Slave 10: GW, DN, NM 34. Slave 11: GW, DN, NM 35. Slave 12: GW, DN, NM 36. Slave 13: GW, DN, NM 37. Slave 14: GW, DN, NM 38. Slave 15: GW, DN, NM 39. Slave 16: GW, DN, NM 40. Slave 17: GW, DN, NM 41. Slave 18: GW, DN, NM 42. Slave 19: GW, DN, NM 43. Slave 20: GW, DN, NM |
|  | **Stale configuration**  *<mandatory after configuration change for existing service>* | # Any new configuration changes (for already existing service) should end with stale configuration to deploy it in required nodes. Not required for newly added services. But if you make any changes to the existing service then it should end with stale configuration  The Stale Configurations page provides differential views of changes made in a cluster. For any configuration change, the page contains entries of all affected attributes. For example, you made a cange to hdfs-site.xml then Stale configuration icon will pop-up near HDFS (in Cloudera Manager) to re-deploy the changes made. |
|  |  |  |
| 1 | **Remove Role from a Node** | # How to remove an existing role from a node?  # How to transfer a role from current node a to different node b?  Ex: YARN  **Step 1:** CM Host -> Yarn -> Instances -> Select a role to be removed/modified (ex: Job history on node a) -> Action -> stop it -> Action -> Delete (Note: some roles can’t be deleted, so just stop)  **Step 2:** CM Host -> Yarn -> Instances -> Select a role to be added/modified (ex: Job history on node b) -> Action -> Add role (button)  **Step3:** Stale configuration |
| 2 | **Add service Zookeeper** | # Zookeeper is the prerequest for HDFS and YARN  It is a co-ordination service (so configure in at least 3 master nodes) |
|  | Cloudera Navigation | CM Home -> Add service -> Zookeeper -> provide at least 3 master hosts -> default path for data directory and logs /var/lib/zookeeper/ -> |
| 3 | **Add Service HDFS** | CM Home -> Add Service -> HDFS -> .. .TBD |
| 4 | **Enable High Availability** | Pre-request for HA is Zookeeper for co-ordination |
| 4.1 | Enable High Availability  Cloudera Navigation | CM Home -> HDFS -> Instance -> Enable High Availability (Button) -> Enter Service ID (nameservice1) -> Assign Roles (provide **Namenode host** and **Journal node hosts**) -> Cloudera will automatically initialize zookeepr for co-ordination -> next  **Namenode host**(two nodes): Provide host for active and passive name nodes.  Ex: if you already have Namenode in master1, then select master1 again (as active) and one additional host Master 2 (as passive). The default path for all the name node is /dfs/nn/ in the corresponding hosts  **JournalNode hosts**(three nodes)**:** Provide Master 3, Master 4 and Master 5. So that journal node will maintain additional copies of editlogs and FSImage in Master 3 and 4 and 5. It is mandatory to mention the Journal node path. Ex: /dfs/jn/  **Failover Controller:** Will automatically configured on Active and Passive nodes  **Expected issue:** If namenode directory is not empty then format will fail, we can ignore this issue during HA setup |
| 4.2 | Enable High Availability  Impact on XML file after HA configuration | **#** Impact on XML file after HA configuration  **Core-site.xml (very less difference)**  **Before HA:**  <configuration> <property><name>fs.defaultFS</name>  <value>hdfs://masternode-where namenode sits:port#</value>  </property></configuration>  **After HA:**  <configuration> <property><name>fs.defaultFS</name>  <value>hdfs://nameservice1</value>  </property></configuration>  **Hdfs-site.xmls (has many changes like new zookper nodes and nameservice details**  **After HA:**  <name><dfs.ha.namenodes.nameservice1</name>  <value>active node, passive node </value>  <name><dfs.ha.namenode.rpc-address.nameservice1.activenode</name>  <value>active node</value>  <name><dfs.ha.namenode.rpc-address.nameservice1.passivenode</name>  <value>passive node</value>  Etc.. |
| 4.3 | Enable High Availability  # Impact on Web UI After HA configuration | # Impact on Web UI After HA configuration  **Before HA:**  CM Home -> HDFS -> We can see name node UI link  **After HA:**  CM Home -> HDFS -> we can see two name node UI Link (standby and active) |
| 5 | **Add MapReduce V1**  <Just for reference> | # MRV1 classic version just for reference  CM Home -> Add Service -> MapReduce 1-> select HDFS (prerequest) -> (Automatically Tasktracker assigned to Slave Node where Data Node installed but JobTracker assigned to Master Node where Namenode ‘NOT’ installed) -> This is default setting, we can change if we want -> Job Tracker local data directory (/mapred/jt) by default -> Task Tracker local data directory (/mapred/local) by default |
| 5.1 | Assign roles to MapReduce  <Set Gateway to MapReduce, otherwise jobs will run in local instead of cluster> | # Things to be done after add MapReduce services  If you run MapReduce job immediately after add MapReduce service then job will run in local mode instead of cluster mode.  CM Home -> MapReduce -> Instance -> Add roles (button) – Go to Gateway and select all nodes -> Stale configuration -> Deploy Client Configuration  So assign the *Gateway roles to all the nodes for MapReduce*, so that the job will run in cluster mode. |
| 6 | **Add Yarn (MR2 included)** | # Yarn (MR2 included)  CM Home -> **Add Service** -> **Yarn (MR2 included)-**> select HDFS & Zookeepr (prerequest) -> **Role Assignment** -> 1. Resource Manager (similar to job tracker) -> 2. Job history server (in MRV1 logs scattered to multiple locations, to make single server to store logs, this is required. Use same node used for resource manager). 3. Node manager (something similar to task tracker, automatically pick wherever data nodes are running) -> **Node Manager local directory** -> /yarn/nm by default -> finish |
| 6.1 | Add Yarn (MR2 included)  <Set Gateway to YARN, otherwise jobs will run in local instead of cluster> | CM Home -> Yarn -> Instance (Menu) -> Add roles (button) -> Gateway (select all nodes) -> Stale configuration -> Deploy Client Configuration |
| 7 | **Add service: Hive** | Hive is a data warehouse system that offers a SQL like called HiveQL.  # Configure Hive using cloudera manager   1. Configure Hive meta store 2. Configure hiveserver2 3. Configure Gateway   Pre-request : HDFS and Yarn, zookeepr will be required if you want to interact HDFS with Hive |
| 7.1 | Configuration | 1. Gateway: by default all the nodes 2. Hive metastore server: One master node. Choose the one where we have Cloudera Management services 3. Hiveserver2 is for Thrift Server. Use the same above master node. Unless hiveserver2 is running, developer will not be able to connect hive 4. WebHCatserver: no need to select any server |
| 7.2 | Database  <repository for metastore> | # By default metastore is the database  # By default postgresql. Select MySQL as we already have a custom DB  # Database hostname: select the same master node. It is important to change the port number for Mysql. 3306  # Database & user name: hive  # password:  Before run test connection, make sure the database and user ‘hive’ is available in mysql. If not, Create it using below commands |
| 7.3 | Create database & user in Mysql | >login to master host  >mysql –u root –p  >show databases;  >create database hive;  >create user hive identified by ‘pwd1’;  >grant all on hive.\* to hive;  Click on ‘test connection’ in cloudera |
| 7.4 | Hive ware house directory & port | # default directory for hive warehouse  /user/hive/warehouse  #default Hive metastore port  9083 |
| 7.5 | DB dictionary | # login to db  >mysql –u root –p  >use hive; -- To login to hive db  >show tables; -- To list all the tables |
| 7.6 | CM internal steps | 1. CM will create metastore database tables 2. CM will create hive user directory in HDFS. >haoop fs –ls /user/hive 3. CM will create hive data warehouse directory under. /user/hive/warehouse |
| 8 | **Add service: Oozie** | # Configure Oozie…watch the video again and update details clearly   1. Oozie Server: One master node |
| 8.1 | Oozie server Data dictionary | /var/lib/oozie/data |
| 8.2 | MySQL | 1. DB, user to be created 2. MySQL connector link to be created |
| 8.3 | DB parameter | 1. Add Oozie service first, stop it and go to db configuration and change required changes in DB parameter 2. EX: Change the DB name from derby to My SQL, change user, pwd, etc |
| 8.4 | Safety volve | To add new property |
| 9 | **Add service – Pig** |  |
| 9.1 | Pig configuration | There is no option to add service for Pig in Cloudera Manager. When we install HDFS and client setup. All the pig configurations and parameter setups are done automatically.  But manually check the pig is installed on your machine and properties are using proper NameNode and Resource Manager.  **After Pig Installation**   1. Cloudera Manager will not show Pig in service list . Because there is No Daemon process for pig in CM 2. It is just like a client based tool which will be deployed in slaves at the time of installation. So no need to ‘start’ or ‘stop’ the pig service |
| 9.2 | Check Pig availability | # To check pig is installed. It will enter into ‘grunt’ shell  >pig  Grunt> |
| 9.3 | Configuration path | # The default path  Cd /etc/pig/config |
| 9.4 | Check the pig installation path (for jar) | # It is not mandatory….. Just to find the location where pig software is installed  >sudo find / -name “\*pig\*.jar” |
| 9.5 | Properties | Cloudera Manager will not support to update pig properties which is highly unlikely. So go to  >cd /etc/pig/conf/pig.properites  >cd /var/run/cloudera-scm-agent/process |
| 9.6 | More details | Refer quick\_ref\_pig |
| 10 | **Add Service - Hue** | # Hue is a web interface. It consolidates all Hadoop eco system tools under one web interface |
| 10.1 | Steps | Add service -> Add all components you want to run using Hue (*dependencies*) -> Start the service |
| 10.2 | Dependencies  *< Add all components you want to run using Hue>* | # There are 5 mandatory dependencies, the remaining are optional (but won’t harm to add remaining services as part of Hue)  Mandatory dependency services:   1. ZooKeeper 2. HDFS 3. YARN (MRV2) 4. Hive 5. Oozie   Non-mandatory services:   1. Impala 2. Sqoop/Sqoop2 3. HBase |
| 10.3 | Node | # Choose one Master node where you don’t have NN, SNN, HA |
| 10.4 | WUI | CM Home -> Hue -> Hue Web UI (under quick links) -> First time login with uid & pwd as admin & admin  # It will automatically create a folder for user ‘admin’ in HDFS  >Hadoop fs –ls /user/admin |
| 10.5 | User & Group | #Note: Whenever you add a new user, it will create a directory in HDFS. Ex: You are adding a new user in Hue ‘kumar’. It will add a directory in HDFS /user/kumar  After login to Hue -> Admin (menu) -> Manage Users -> Add users (add new user details) -> known issue (can’t make home directory for user) -> Because admin doesn’t has access to create user under home directory (/user/kumar)  # Also Integrate with LDAP (Need to check LDAP details)  # Also create a group and let user part of the group  # Hue Web UI: Port:8888  # Hue 3.6.0 |
| 10.6 | Dependencies  *< Modify components you want to run using Hue>* | CM Home -> Hue -> Configuration -> disable/enable services |
| 11 | **Add Service: Sqoop** | # Pre-request to export/import data between MySQL and HDFS, setup MySQL JDBC connector and assign JDBC Connector to sqoop library. Refer quick ref sqoop for more details  CM Home -> Add service -> Sqoop -> Choose one client  # Sqoop does not have many parameter, because it uses HDFS and MapReduce and for log refer MapReduce log, there is no sqoop log |
|  |  |  |
|  |  |  |
|  |  |  |
| 11 | **Add Service – Tez** | Tez is supported by Only HortonWorks. Cloudera will not support. |
| 12 | **Add Service – Flume** | *<I’ve installed ‘flume agent’ in slaves, need to verify whether it required to be installed in master too, if so how many master>* |
| 13 | **Add Service – Spark** | Two Versions:   1. Spark: Apache Spark is an open source cluster computing system. This service runs Spark as an application on YARN 2. Spark (Standalone): Apache Spark is an open source cluster computing system. This is the standalone version of the service which does not use YARN for resource management. Cloudera recommends using Spark on YARN instead of this standalone version |
| 13.1 | **History server** | 1. History server (HS): Need to select one history server. *<Hope it should be only one master, To be verified>* 2. Gateway(G): Make sure gate way role assigned to all the hosts |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 15 | **ACL** | # ACL configuration setup  CM Home -> Yarn (MR2 included) -> Configuration -> Service wide -> Enable Resource manager ACL |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Remove hosts** | # Follow the below steps to remove hosts from the cluster  # Refer QuickRef\_Cloudera\_CDH\_CM (under remove hosts)   1. Go to host tab 2. Stop all services/components 3. Decommission/Delete the host from the cluster (cloudera uses decommission, hortonworks uses delete) 4. Delete the host from the cluster 5. Uninstall 6. Clouder: “cloudera-manager-agent” and “cloudera-manager-daemon” 7. Ambari: “ambary-agent” and all other softwares 8. Cleanup all the remnants 9. Steps will differ for master and slaves. Extremely carful while decommissioning the master 10. Run balancer carefully (as it can exhaust network) |
|  | **Remove hosts**  *<in detail>* | # Below steps from Hortonworks… almost similar for Cloudera too   1. Step1: Stop all services/component: Host menu -> select host (to be removed) -> Action button -> selected host (menu) -> hosts (menu) -> Turn on maintenance mode (so that it reduce the alert) -> Stop all components 2. Step2: Decommission/Delete the host from the cluster (cloudera uses decommission, hortonworks uses delete): Select host (to be remove) -> click on it -> Host action (button) -> Delete 3. Step 2: pre-request: Stop the agent 4. Step 2: Ex: >ssh –t –t [ec2-user@ip-10-0-0-179.ec2.internal](mailto:ec2-user@ip-10-0-0-179.ec2.internal) “sudo ambary-agent stop” 5. Step 3: Uninstall: 6. Step 3: Ex: >ssh –tt [ec2-user@ip-10-0-0-179.ec2.internal](mailto:ec2-user@ip-10-0-0-179.ec2.internal) “sudo yum –y remove ambary\*” 7. Step 3: Ex: >ssh –tt [ec2-user@ip-10-0-0-179.ec2.internal](mailto:ec2-user@ip-10-0-0-179.ec2.internal) “sudo yum –y remove hdp\*” 8. Step 4: Remove folder and cleanup all the remnants 9. Step 4: Ex: >ssh –tt [ec2-user@ip-10-0-0-179.ec2.internal](mailto:ec2-user@ip-10-0-0-179.ec2.internal) “sudo ls –ltr /etc” 10. Step 4: Ex: >ssh –tt [ec2-user@ip-10-0-0-179.ec2.internal](mailto:ec2-user@ip-10-0-0-179.ec2.internal) “sudo ls –ltr /hadoop” 11. Step 5: Run balancer 12. Note: Cloudera manager will let you stop all services on a click of button and remove the host 13. Note: Ambari do not let you stop all services on a click of button, instead it provide REST API to take care of those from command line |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Cloudera Configuration group** | # Role, XML Properties, Configurations are Grouped by Cloudera as follows  CM Home -> Eco system (zookeeper, hdfs, yarn, etc) -> Configuration -> Category |
| 1 | Zookeeper |  |
| 2 | HDFS |  |
| 3 | Yarn (Including MR2) | # Each group has sub-groups as well.   1. Service wide: 2. Gateway default group 3. Job History Server default group 4. **Node Manager Default group:** Logs, Monitoring, Performance, port & address, Resource management, security and stacks collections are sub-groups. 5. Resource Manager default group |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Cloudera Manager Logs** | Default |
|  | Zookeeper |  |
|  | HDFS |  |
|  | YARN (MRV2 include) | CM Home -> Yarn (MRV2 include) -> Configuration -> search for log.dir  **Default path:**  /var/log/Hadoop-mapreduce  /var/log/Hadoop-yarn  **Also logs are available in HDFS in MRV2. Where as in MRv1 logs won’t be available in hdfs:**  >Hadoop fs –ltr -R /tmp/logs/hduser | More |
|  |  |  |

**Quick Ref: Cloudera Distribution for Hadoop – CDH**

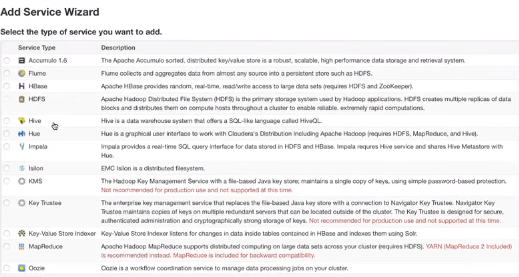
|  |  |  |
| --- | --- | --- |
|  | **CDH** | **CDH (Cloudera Distribution Hadoop) is open-source Apache Hadoop distribution provided by Cloudera Inc. CDH (Cloudera's Distribution Including Apache Hadoop) is the most complete, tested, and widely deployed distribution of Apache Hadoop.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Hortonworks Cluster setup for 6 servers - Ambari**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server 1 ip-172-31-13-154** | **Server 2 ip-172-31-29-153** | **Server3 ip-172-31-38-183** | **Server 4 ip-172-31-43-77** | **Server 5 ip-172-31-53-48** | **Server 6 (Master) ip-172-31-58-37** |
| Accumulo Client | Accumulo Client | Accumulo Client | Accumulo Client | Accumulo Client | Accumulo Client |
|  | Accumulo GC |  |  |  |  |
|  | Accumulo Master |  |  |  |  |
|  | Accumulo Monitor |  |  |  |  |
|  | Accumulo Tracer |  |  |  |  |
|  | Accumulo TServer |  |  |  |  |
| App Timeline Server |  |  |  |  |  |
|  | Atlas Metadata Server |  |  |  |  |
| DataNode | DataNode | DataNode | DataNode | DataNode | DataNode |
|  | DRPC Server |  |  |  |  |
| Falcon Client | Falcon Client | Falcon Client | Falcon Client | Falcon Client | Falcon Client |
|  | Falcon Server |  |  |  |  |
| Flume | Flume | Flume | Flume | Flume | Flume |
| HBase Client | HBase Client | HBase Client | HBase Client | HBase Client | HBase Client |
| HBase Master | HBase Master |  |  |  |  |
| RegionServer | RegionServer |  |  |  |  |
| HCat Client | HCat Client | HCat Client | HCat Client | HCat Client | HCat Client |
| HDFS Client | HDFS Client | HDFS Client | HDFS Client | HDFS Client | HDFS Client |
| History Server |  |  |  |  |  |
| Hive Client | Hive Client | Hive Client | Hive Client | Hive Client | Hive Client |
| Hive Metastore |  |  |  |  |  |
| HiveServer2 |  |  |  |  |  |
| Kafka Broker | Kafka Broker | Kafka Broker | Kafka Broker | Kafka Broker | Kafka Broker |
|  | Knox Gateway |  |  |  |  |
| Mahout | Mahout | Mahout | Mahout | Mahout | Mahout |
| MapReduce2 Client | MapReduce2 Client | MapReduce2 Client | MapReduce2 Client | MapReduce2 Client | MapReduce2 Client |
| Metrics Collector |  |  |  |  |  |
| Metrics Monitor | Metrics Monitor | Metrics Monitor | Metrics Monitor | Metrics Monitor | Metrics Monitor |
|  |  |  |  |  | NameNode |
| NFSGateway | NFSGateway | NFSGateway | NFSGateway | NFSGateway | NFSGateway |
|  | Nimbus |  |  |  |  |
| NodeManager | NodeManager | NodeManager | NodeManager | NodeManager | NodeManager |
| Oozie Server |  |  |  |  |  |
| Oozie Client | Oozie Client | Oozie Client | Oozie Client | Oozie Client | Oozie Client |
| Pig | Pig | Pig | Pig | Pig | Pig |
|  |  |  |  |  | ResourceManager |
|  | SNameNode |  |  |  |  |
| Slider | Slider | Slider | Slider | Slider | Slider |
| Spark Client | Spark Client | Spark Client | Spark Client | Spark Client | Spark Client |
|  | Spark History Server |  |  |  |  |
| Sqoop | Sqoop | Sqoop | Sqoop | Sqoop | Sqoop |
|  | Storm UI Server |  |  |  |  |
| Supervisor | Supervisor | Supervisor | Supervisor | Supervisor | Supervisor |
| Tez Client | Tez Client | Tez Client | Tez Client | Tez Client | Tez Client |
| WebHCat Server |  |  |  |  |  |
| YARN Client | YARN Client | YARN Client | YARN Client | YARN Client | YARN Client |
| ZooKeeper Client | ZooKeeper Client | ZooKeeper Client | ZooKeeper Client | ZooKeeper Client | ZooKeeper Client |
| ZooKeeper Server | ZooKeeper Server | ZooKeeper Server | ZooKeeper Server | ZooKeeper Server | ZooKeeper Server |

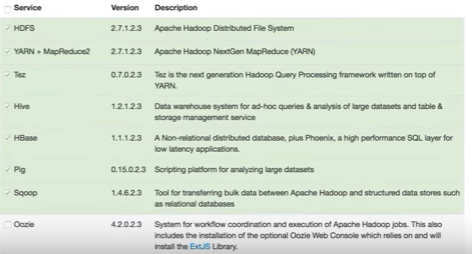


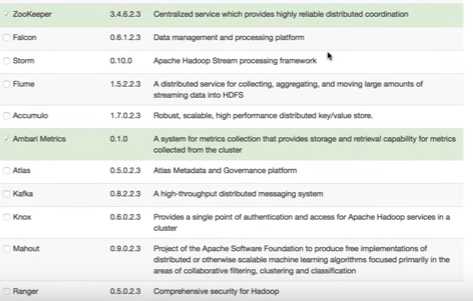
**CDH**





**Ambari/Hortonworks**

****

****