

Setup HDP 2.x using Ambari 2.x on CentOS 6.x

WIP

Hortonworks very recently released the Latest Version for HDP (V2.3) . Here are some of the new features :

1. Smart Configuration - provides a much simpler way to configure the most critical and frequently used parameters.
2. YARN Capacity Scheduler
3. Customizable Dashboards.
4. SQL Query Support , Visual "Explain" Plan and allowing to extend debugging experience when using the Tez execution Engine.
5. HDP 2.3 also provides a Pig Latin Editor that brings a modern browser-based IDE experience to Apache Pig.
6. There is also a File Browser for HDFS and an entirely new user experience for Apache Falcon with a web-forms approach to rapidly develop feeds and processes. The new Falcon UI also allows you to search and browse processes that have executed, to visualize lineage and to setup mirroring jobs to replicate files and databases between clusters or to cloud storage such as Microsoft Azure Storage.
7. Solr on YARN Technical Preview.
8. HA Availability on Apache Storm , Ranger and Falcon.
9. Data encryption in HDFS .
10. HDP 2.3 Ships with Ranger 0.5.0 which extends support to YARN, Solr, Kafka .
11. Solr will be used to do indexing audit data and serve real-time query results.
12. Introduction of Apache Atlas and Apache Mahout to HDP platform .
13. Hortonworks SmartSense targetted for better customer support.
14. Ambari Server extending support to MS SQL Server (Tech Preview)

For more info on the new features , go through this : <http://hortonworks.com/hdp/whats-new/>

This article will help you setup HDP 2.2 - Ambari 2.0 (Production Stable) / HDP 2.3 - Ambari 2.1.1 (Latest)

HDP 2.3 currently does not support Ubuntu Operating System .

This document is part of HortonWorks JumpStart Documentation.

This documentation is for reference purpose only. All commands will not apply to your use-case. Also, please do not copy and paste the commands in your terminal or shell. I would encourage you to type out all the commands pertaining to your use –case.

This documentation is divided into the following two parts

- Prerequisites Step - Set of steps performed to make the machine(s) ready for HDP installation.
- Installation Step - This step includes HDP Installation using Ambari and Sanity Checks

Prerequisites Step

1. On all nodes , observe if the correct OS (Version) has been installed.

```
[ ]# cat /etc/issue
CentOS release 6.5 (Final)
Kernel \r on an \m
```

2. On all nodes, check Passwordless SSH . For details refer : [Passwordless SSH](#)
3. On all nodes, make sure you performed checks for the steps 1-6 documented in [Common Prerequisites for installing Distributed Systems](#)
4. On all nodes, check Mount Points on all nodes. Execute this command to get info about disk drive and the mount points they are configured on

```
# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvde   202:64   0   8G  0 disk
xvde1  202:65   0   8G  0 part /
xvdf   202:80   0  60G  0 disk /mydrive
```

Contact admin, if you don't have access to creating mount points or mounting disks

5. On all nodes, perform the following steps to disable transparent huge pages . If there are square brackets around "never", this means that transparent huge pages have been disabled.

Open "/etc/grub.conf" file and add "transparent_hugepage=never" line to the kernel boot line and reboot your system. Sample file is as follows :

```
title CentOS (2.6.32-431.29.2.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-431.29.2.el6.x86_64 ro root=LABEL=_ 
console=ttyS0,115200 crashkernel=no
transparent_hugepage=never
initrd /boot/initramfs-2.6.32-431.29.2.el6.x86_64.img
```

Alternatively, open the "/etc/rc.local" file , add the following and reboot your server.

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

Validate it by typing the following command . If there are square brackets around "never", this means that transparent huge pages have been disabled.

```
# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
```

Other possible methods to check if transparent huge pages have been disabled are as follows . Type out the following commands and if the output is 0 , then transparent huge pages are disabled.

```
[ ]# grep -i HugePages_Total /proc/meminfo
HugePages_Total:      0
[ ]# cat /proc/sys/vm/nr_hugepages
0
```

6. On all nodes, install Java . For more details refer : [Installing Java on Linux \(rpm based\)](#)

For HDP V 2.3 , if you are installing Oracle JDK V 1.7 , then JDK 1.7_67 is the minimum which you should install .
For HDP V 2.3 , if you are installing Oracle JDK V 1.8 , then JDK 1.8_40 is the minimum which you should install .

7. On all nodes, check if it has internet access

```
[root@solrmaster /]# ping www.google.com
PING www.google.com (74.125.28.99) 56(84) bytes of data.
64 bytes from pc-in-f99.1e100.net (74.125.28.99): icmp_seq=1 ttl=40 time=13.6 ms
64 bytes from pc-in-f99.1e100.net (74.125.28.99): icmp_seq=2 ttl=40 time=13.6 ms
64 bytes from pc-in-f99.1e100.net (74.125.28.99): icmp_seq=3 ttl=40 time=13.6 ms
64 bytes from pc-in-f99.1e100.net (74.125.28.99): icmp_seq=4 ttl=40 time=13.6 ms
64 bytes from pc-in-f99.1e100.net (74.125.28.99): icmp_seq=5 ttl=40 time=13.6 ms
```

8. If the client is going for a central server (mysql assumed in this document), check if the service is running with this command .

```
[]# service mysqld status
mysqld (pid 1227) is running...
```

If you want to Install MySQL follow these [Install MySQL](#), [Configure MySQL](#)

Installation Step

After the Pre-Requirements checks are done , proceed with the Ambari installation with the following steps:

1. Install the mysql-connector-java on all the nodes of your cluster . Run this command on all nodes. This will install in /usr/share/java .

```
yum install mysql-connector-java
```

2. Please observe the below figure as it gives an idea which version of HDP is compatible with which version of Ambari . Accordingly choose your ambari repo and HDP Repo. The Repo needs to be created on all nodes of your cluster.

Ambari*	HDP 2.3	HDP 2.2	HDP 2.1 (deprecated)	HDP 2.0 (deprecated)
2.1	x	x	x	x
2.0		x	x	x
1.7		x	x	x
1.6.1			x	x
1.6.0			x	x
1.5.1			x	x
1.5.0				x

List of Ambari Repos :

```

Ambari 2.1.1 Repo :
wget -nv
http://public-repo-1.hortonworks.com/ambari/centos6/2.x/updates/2.1.1/ambari.repo
-O /etc/yum.repos.d/ambari.repo
Ambari 2.0 Repo :
wget -nv
http://public-repo-1.hortonworks.com/ambari/centos6/2.x/updates/2.0.0/ambari.repo
-O /etc/yum.repos.d/ambari.repo
Ambari 1.7 Repo :
wget
http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.7.0/ambari.repo
-O /etc/yum.repos.d/ambari.repo
Ambari 1.6.1 Repo :
wget
http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.6.1/ambari.repo
-O /etc/yum.repos.d/ambari.repo
Ambari 1.6.0 Repo :
http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.6.0/ambari.repo
-O /etc/yum.repos.d/ambari.repo
Ambari 1.5.1 Repo :
wget
http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.5.1/ambari.repo
-O /etc/yum.repos.d/ambari.repo
Ambari 1.5.0 Repo :
wget
http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.5.0/ambari.repo
-O /etc/yum.repos.d/ambari.repo

```

Once you choose the Ambari Repo , and reach the stage where you are in the install wizard setup , choose your HDP Repo keeping in mind as to which version is compatible with your Ambari Version. (Refer above diagram). For e.g. : If you are choosing Ambari 1.5.0 , then you should not choose HDP 2.3 as they are not compatible with each other.

Also , in the install wizard , you can change the link related to your repo file to indicate another version of Ambari . For e.g. (as it stands now) , the latest version of HDP 2.2 is 2.2.8 , but if I want 2.2.6 , I can change that link and instead of 2.2.8 , substitute it with 2.2.6 and the installation will go through.

Hence, to conclude, **you can choose your Ambari Version and the compatible HDP Version and in that HDP Version specify the exact minor version release to get all possible combinations of installing Ambari with HDP .**

After you download the repo file for Ambari,it will put it in the /etc/yum.repos.d/ location. You could also "cd" to the directory to verify for the same.

If the client is going ahead with default databases which comes along with all the services, please skip Step 3.

3. If the client has setup MySQL Server as the central database, please follow this step. This step talks about setting up users for these services – hive , oozie, ambari. Open mysql with this command:

```
mysql -u root -p <you press enter>
```

Now enter the root pwd which opens the mysql shell. Execute these commands for each service as follows :

Hive:

```
CREATE USER 'hive'@'localhost' IDENTIFIED BY 'hive';
GRANT ALL PRIVILEGES ON *.* TO 'hive'@'localhost' WITH GRANT OPTION;
CREATE USER 'hive'@'%' IDENTIFIED BY 'hive';
GRANT ALL PRIVILEGES ON *.* TO 'hive'@'%' WITH GRANT OPTION;
CREATE USER 'hive'@'<FQDN of the HiveMetaStoreHost>' IDENTIFIED BY 'hive';
GRANT ALL PRIVILEGES ON *.* TO 'hive'@'<* FQDN of the HiveMetaStoreHost >' WITH
GRANT OPTION;
```

Ambari :

```
CREATE USER 'ambari'@'localhost' IDENTIFIED BY 'ambari';
GRANT ALL PRIVILEGES ON *.* TO 'ambari'@'localhost' WITH GRANT OPTION;
CREATE USER 'ambari'@'%' IDENTIFIED BY 'ambari';
GRANT ALL PRIVILEGES ON *.* TO 'ambari'@'%' WITH GRANT OPTION;
CREATE USER 'ambari'@'<FQDN of the AmbariServerHost>' IDENTIFIED BY 'ambari';
GRANT ALL PRIVILEGES ON *.* TO 'ambari'@'<* FQDN of the AmbariServerHost >' WITH
GRANT OPTION;
```

Oozie :

```
CREATE USER 'oozie'@'localhost' IDENTIFIED BY 'oozie';
GRANT ALL PRIVILEGES ON *.* TO 'oozie'@'localhost' WITH GRANT OPTION;
CREATE USER 'oozie'@'%' IDENTIFIED BY 'oozie';
GRANT ALL PRIVILEGES ON *.* TO 'oozie'@'%' WITH GRANT OPTION;
CREATE USER 'oozie'@'<FQDN of the OozieServerHost>' IDENTIFIED BY 'oozie';
GRANT ALL PRIVILEGES ON *.* TO 'oozie'@'<* FQDN of the OozieServerHost >' WITH
GRANT OPTION;
```

4. Next Step is the Ambari server Installation on the host which will have the Ambari Server. Run the following command :

```
yum install ambari-server
```

5. Install Ambari-Agents on all the hosts of the cluster. Run the following command .

```
yum install ambari-agent
```

6. On the host which has ambari-server installed, Run the Setup with the following command .

```
ambari-server setup -j <Path to your JDK>
```

Note: the `-j` option is used to specify the path to your installed JDK. If you do not specify this, it will go ahead with the setup and download a separate Oracle JDK and will not use the Java that we installed during the Pre-Reqs Step (Step1). You can make your own choices during the setup and hence some or even all will not co-relate with this. Also note, if the client has gone ahead with the default databases which come along with every service, you should say "N" , to the question , when it asks to "Enter Advanced Database Configuration".

7. After the setup , if the client has chosen MySql Server as the central database, goto your Mysql Server and login as ambari user with the following command.

```
mysql -u ambari -p <Press Enter and enter your ambari user's password>
```

This will open the mysql prompt. Run the next command .

```
create database ambari;
use ambari;
source /var/lib/ambari-server/resources/Ambari-DDL-MySQL-CREATE.sql;
```

These steps are important as when you start ambari , it will try to connect with the database "ambari" , and if it doesn't find it , it will throw an error saying it couldn't connect with the database.

8. Lets start the ambari-server with this command .

```
sudo service ambari-server start
```

Check the Log files to see if it has thrown any errors with this command.

```
cat /var/log/ambari-server/ambari-server.log
```

Open up a duplicate session of host having ambari server and run this command .

```
tail -f /var/log/ambari-server/ambari-server.log
```

This will tail the log of the server and if any changes happen , we can immediately check this tailing log.

9. On all nodes, run the following process.

- a. Run this command :

```
vi /etc/ambari-agent/conf/ambari-agent.ini
```

This will open the .ini for that ambari-agent . Edit the following entry .

```
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441
```

By default, it will be set to localhost , change it to your ambari-server FQDN.

- b. Start the ambari-agent with this command

```
sudo service ambari-agent start
```

Now notice the window which has the tailing log, as soon as you hit the command to start the ambari-agent, the log will note the change . It will print something like " Registration of host....." . In this way, we get to know that the agent has successfully registered with the server.

10. After all agents have successfully registered with the server, we can go to the ambari-server Web Wizard for Setup of Cluster. Go to your browser and type in the Address Bar :

<AmbariServerHostName : 8080>

This will open a Login Screen. Enter username and Password as "admin".

11. Next, you will be prompted with the following Screen. Click "Launch Install Wizard"

Welcome to Apache Ambari

Provision a cluster, manage who can access the cluster, and customize views for Ambari users.

Create a Cluster

Use the Install Wizard to select services and configure your cluster



[Launch Install Wizard](#)

[Manage Users + Groups](#)

[Deploy Views](#)

12. Next , you will be prompted with next screen asking you enter Name of the Cluster. Click on "Next"

Get Started

This wizard will walk you through the cluster installation process. First, start by naming your new cluster.

Name your cluster [Learn more](#)

YourClusterName

[Next →](#)

13. Next , select "HDP 2.2" or "HDP 2.3" as your version as indicated in the screenshot. Click on "Next"

Select Stack

Please select the service stack that you want to use to install your Hadoop cluster.

Stacks

- HDP 2.2
- HDP 2.1
- HDP 2.0

[Advanced Repository Options](#)

[← Back](#)

[Next →](#)

14. Register your hosts indicated in next screenshot. Please enter your FQDNs of all the members of your cluster. Then, click on "Perform Manual Registration on hosts and do not use SSH". We have taken this option as we have manually registered all the hosts (started all the ambari-agents manually.) Click on "Register and Confirm".

Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use [Pattern Expressions](#)

```
sfslave3.cloudwick.com
sfslave1.cloudwick.com
sfslave2.cloudwick.com
sfmaster.cloudwick.com
```

Host Registration Information

Provide your [SSH Private Key](#) to automatically register hosts

[Choose File](#) [No file chosen]

```
ssh private key
```

SSH User Account

root

Perform [manual registration](#) on hosts and do not use SSH

[← Back](#)

[Register and Confirm →](#)

15. Next screen will show ambari registering all the hosts supplied in the previous screenshot . we get the next screenshot as below.

Confirm Hosts

Registering your hosts.

Please confirm the host list and remove any hosts that you do not want to include in the cluster.

		Show: All (4) Installing (0) Registering (0) Success (4) Fail (0)	
Host	Progress	Status	Action
sfslave3.cloudwick.com	<div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div>	Success	
sfslave1.cloudwick.com	<div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div>	Success	
sfslave2.cloudwick.com	<div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div>	Success	
sfmaster.cloudwick.com	<div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div>	Success	

Show: 25 ▾ 1 - 4 of 4

All host checks passed on 4 registered hosts. [Click here to see the check results.](#)

[← Back](#)

[Next →](#)

If you have successfully completed all the Pre-Requirements from Step 1, you will get no warning and will say "All Host checks passed on X registered hosts" as in the above screenshot. Click on "Next".

16. This screen will prompt you for the services that need to be installed. For this demonstration, I have selected all services. Please see below screenshot for clarity.

Choose Services

Choose which services you want to install on your cluster.		
Service	Version	Description
<input checked="" type="checkbox"/> HDFS	2.6.0.2.2	Apache Hadoop Distributed File System
<input checked="" type="checkbox"/> YARN + MapReduce2	2.6.0.2.2	Apache Hadoop NextGen MapReduce (YARN)
<input checked="" type="checkbox"/> Tez	0.5.2.2.2	Tez is the next generation Hadoop Query Processing framework written on top of YARN
<input checked="" type="checkbox"/> Hive	0.14.0.2.2	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
<input checked="" type="checkbox"/> HBase	0.98.4.2.2	Non-relational distributed database and centralized service for configuration management & synchronization
<input checked="" type="checkbox"/> Pig	0.14.0.2.2	Scripting platform for analyzing large datasets
<input checked="" type="checkbox"/> Sqoop	1.4.5.2.2	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
<input checked="" type="checkbox"/> Oozie	4.1.0.2.2	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExJS Library .
<input checked="" type="checkbox"/> ZooKeeper	3.4.6.2.2	Centralized service which provides highly reliable distributed coordination
<input checked="" type="checkbox"/> Falcon	0.6.0.2.2	Data management and processing platform
<input checked="" type="checkbox"/> Storm	0.9.3.2.2	Apache Hadoop Stream processing framework

17. The next screen tells you to assign Masters for the services you selected from the previous screen . A good Division of the services would be to fairly divide all the Masters among all the servers in your cluster. In the given snapshot, I have included all the Hive Master Services in [sfmaster.cloudwick.com](#) , YARN services in [sfslave1.cloudwick.com](#),Storm Services in [sfslave2.cloudwick.com](#). Your configuration might be different but it is a good practice to have a good division of services. Click on "Next"

Assign master components to hosts you want to run them on.
* HiveServer2 and WebHCat Server will be hosted on the same host.

NameNode:	sfmaster.cloudwick.com (3.7)	sfmaster.cloudwick.com (3.7 GB, 1 cores)
SNameNode:	sfslave1.cloudwick.com (3.7)	NameNode, Hive Metastore
History Server:	sfslave1.cloudwick.com (3.7)	WebHCat Server, HiveServer2
ResourceManager:	sfslave1.cloudwick.com (3.7)	Oozie Server, Spark History Server
App Timeline Server:	sfslave1.cloudwick.com (3.7)	
Hive Metastore:	sfmaster.cloudwick.com (3.7)	
WebHCat Server:	sfmaster.cloudwick.com	
HiveServer2:	sfmaster.cloudwick.com (3.7)	
HBase Master:	sfslave2.cloudwick.com (3.7)	HBase Master, ZooKeeper Server
Oozie Server:	sfmaster.cloudwick.com (3.7)	Nimbus, DRPC Server, Storm UI Server
ZooKeeper Server:	sfslave2.cloudwick.com (3.7)	Kafka Broker
ZooKeeper Server:	sfslave1.cloudwick.com (3.7)	

18. In the Next Screen, we will assign slaves (Datanodes, NodeManagers, Client, RegionServer) to the server. For this demonstration , I have chosen this configuration as highlighted in the Screenshot. Click on "Next".

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with *.
"Client" will install HDFS Client, MapReduce2 Client, YARN Client, Tez Client, HCat Client, Hive Client, HBase Client, Pig, Sqoop, Oozie Client, ZooKeeper Client, Falcon Client, Slider and Spark Client.

Host	all none	all none	all none	all none	all none	all none	all nc
sfslave3.cloudwick.com*	<input type="checkbox"/>	<input type="checkbox"/> DataNode	<input type="checkbox"/> NodeManager	<input type="checkbox"/> RegionServer	<input type="checkbox"/> Supervisor	<input type="checkbox"/> Flume	<input type="checkbox"/> Client
sfslave1.cloudwick.com*	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> DataNode	<input checked="" type="checkbox"/> NodeManager	<input checked="" type="checkbox"/> RegionServer	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Flume	<input checked="" type="checkbox"/> Client
sfslave2.cloudwick.com*	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> DataNode	<input checked="" type="checkbox"/> NodeManager	<input checked="" type="checkbox"/> RegionServer	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Flume	<input checked="" type="checkbox"/> Client
sfmaster.cloudwick.com*	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> DataNode	<input checked="" type="checkbox"/> NodeManager	<input checked="" type="checkbox"/> RegionServer	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Flume	<input checked="" type="checkbox"/> Client

Show: 25 1 - 4 of 4

← Back Next →

19. The Next Screen will take you to the Properties page for each service . First one is the HDFS Service. Make sure your Data Directories are configured. If you a create a folder where the external disk/EBS Volume have been mounted on , make sure they are getting utilized by providing the path of your folder in Namenode Directory, Secondary Namenode Directory , DataNode Directory and so on. The next set of screenshots (Old Ambari) highlight the above fact.

Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

HDFS	MapReduce2	YARN	Tez	Hive 1	HBase	Pig	Sqoop	Oozie 1	ZooKeeper	Falcon
Storm	Flume	Knox 1	Slider	Spark	Kafka	Ambari Metrics	Misc			
Group	HDFS Default (4)	Manage Config Groups	Filter...							
▼ NameNode										
NameNode hosts	sfmaster.cloudwick.com									
NameNode directories	<input type="text" value="/mydrivemount/hadoop/hdfs/namenode"/>									
NameNode Java heap size	<input type="text" value="1024"/> MB									

Secondary NameNode

SNameNode host	sfslave1.cloudwick.com
SecondaryNameNode Checkpoint directories	/mydrivemount/hadoop/hdfs/namesesecondary

DataNode

DataNode hosts	sfslave3.cloudwick.com and 3 others
DataNode directories	/mydrivemount/hadoop/hdfs/data
DataNode maximum Java heap size	1024 MB
DataNode volumes failure toleration	0
File that stores mount point for each data dir	/etc/hadoop/conf/dfs_data_dir_mount.hist

Node Manager

yarn.nodemanager.resource.memory-mb	2048
yarn.nodemanager.vmem-pmem-ratio	2.1
yarn.nodemanager.log-dirs	/mydrivemount/hadoop/yarn/log
yarn.nodemanager.local-dirs	/mydrivemount/hadoop/yarn/local
yarn.nodemanager.remote-app-log-dir	/app-logs
yarn.nodemanager.remote-app-log-dir-suffix	logs
yarn.nodemanager.aux-services	mapreduce_shuffle

New Ambari :

HDFS **MapReduce2** **YARN** **Tez** **Hive 1** **HBase** **Pig** **Sqoop** **Oozie 1** **ZooKeeper** **Flume** **Kafka** **Misc**

Group **Hive Default (3)** **Manage Config Groups** **Filter...**

ACID Transactions **Interactive Query** **Security**

ACID Transactions **Default query queues** **Choose Authorization**

Run Compactor **Start Tez session at Initialization** **Run as end user instead of Hive user**

Number of threads used by Compactor **Session per queue** **HiveServer2 Authentication**

Customize Services

CLUSTER INSTALL WIZARD

Get Started **Select Stack** **Install Options** **Confirm Hosts** **Choose Services** **Assign Masters** **Assign Slaves and Clients** **Customize Services**

Review **Install, Start and Test** **Summary**

Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

HDFS **MapReduce2** **YARN** **Tez** **Hive 1** **HBase** **Pig** **Sqoop** **Oozie 1** **ZooKeeper** **Flume** **Kafka** **Misc**

Group **MapReduce2 Default (3)** **Manage Config Groups** **Filter...**

MapReduce

MapReduce Framework

Map Memory **Reduce Memory** **Sort Allocation Memory**

20. Next , goto the Hive tab and click on "Existing MySQL Database" and enter "hive" in the password fields . Enter "hive" again to confirm . Click on Test Connection. It should show the green tick. Similarly , go to the oozie tab, click on "Existing MySQL Database" and enter "oozie" as the password and enter the same to confirm . Then , in the Database URL for oozie , edit it and append "?createDatabaseIfNotExist=true" like below. Click on Test Connection . The next couple of screenshots prove this point.

```
jdbc:mysql://<database_host>/oozie??createDatabaseIfNotExist=true
```

21. Moving on , we will need to configure Flume . Goto the Flume Tab , and enter your configuration in the Textbox provided. Here is a snapshot of a sample Configuration.

```

# Flume agent config
## This configuration applies to all Flume agents
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source, this is just basic description of source ,
# You can configure your own configuration .
a1.sources.r1.type = netcat
a1.sources.r1.bind = sfmaster.cloudwick.com
a1.sources.r1.port = 3334

# Describe the sink
a1.sinks.k1.type = hdfs
a1.sinks.k1.hdfs.path = hdfs://sfmaster.cloudwick.com:8020/tmp/
a1.sinks.k1.hdfs.fileType = DataStream

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1

```

content

Flume agent configurations. Specifying the configuration here applies to all hosts in this configuration-group. Please use host specific configuration-groups to provide different configurations on different hosts. For configuring multiple agents on the same host, provide the combined agent configurations here. Each agent should have a different name.

22. Goto the Knox Tab , and set your Knox Gateway password . For this demonstration, I have kept the password as "knox" . Enter the same to confirm . Attached is Screenshot.

HDFS MapReduce2 YARN Tez Hive HBase Pig Sqoop Oozie ZooKeeper Falcon Storm

Flume Knox Slider Spark Kafka Ambari Metrics Misc

Group: Knox Default (4) Manage Config Groups Filter...

Knox Gateway

Knox Gateway host: sfslave3.cloudwick.com

Knox Master Secret: ****

Advanced gateway-log4

23. There are no changes to be done in the Configurations for the rest of the Services such as Slider, Spark, Kafka , Ambari Metrics and Misc. Click on "Next", which will take you to the "Review" Screen . Here's a snapshot for the same. Click on "Deploy". This will display a pop-up summarizing the number of tasks which will be deployed.

Review

Please review the configuration before installation

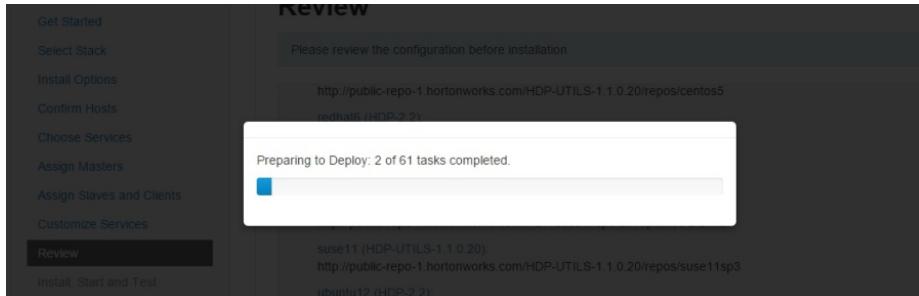
http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/centos5
 redhat6 (HDP-2.2):
 http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.2.4.2
 redhat6 (HDP-1.1.0.20):
 http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/centos6
 suse11 (HDP-2.2)
 http://public-repo-1.hortonworks.com/HDP/suse11sp3/2.x/updates/2.2.4.2
 suse11 (HDP-1.1.0.20):
 http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/suse11sp3
 ubuntu12 (HDP-2.2):
 http://public-repo-1.hortonworks.com/HDP/ubuntu12/2.x/updates/2.2.4.2
 ubuntu12 (HDP-UTILS-1.1.0.20):
 http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/ubuntu12

Services:

HDFS
 DataNode : 3 hosts
 NameNode : sfmaster.cloudwick.com
 SNameNode : sfslave1.cloudwick.com

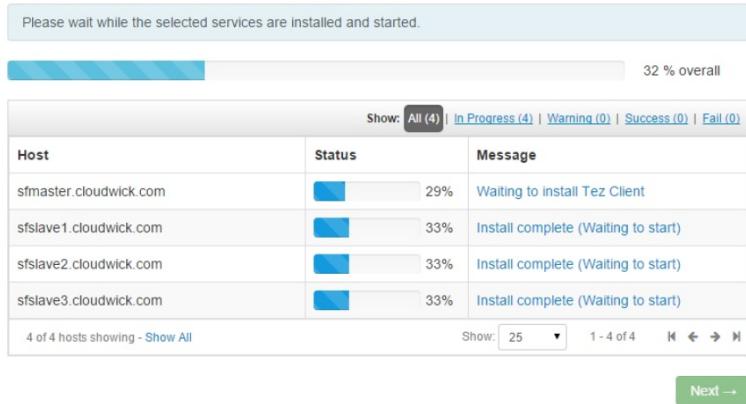
YARN + MapReduce2
 App Timeline Server : sfslave1.cloudwick.com

← Back Print Deploy →

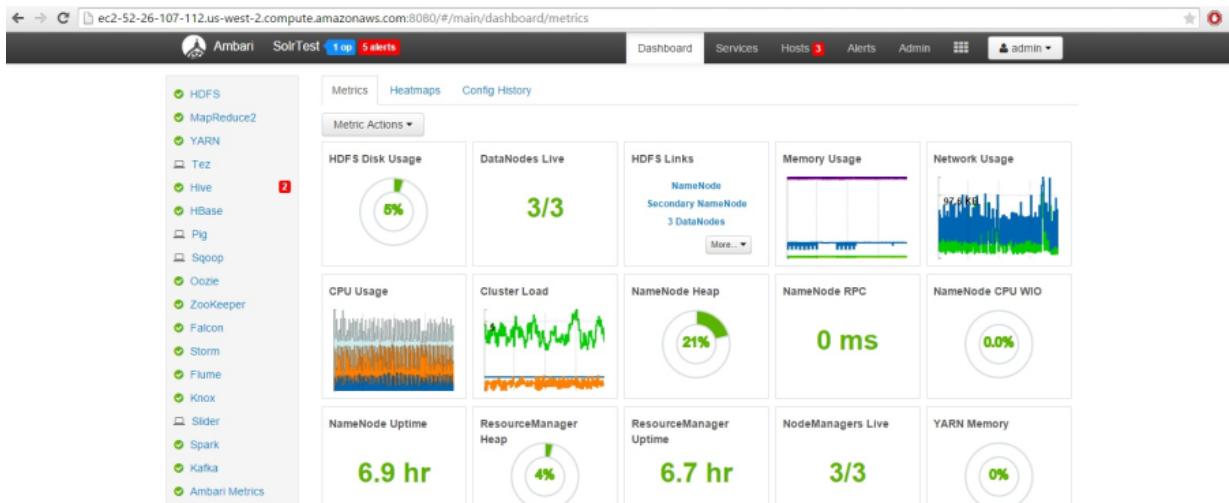


24. Following which, will show a screen showing the installation process in parallel, for all the nodes. The Installation will take a while to finish .(It took 30 – 40 min for this demonstration)

Install, Start and Test



25. After the installation is complete, the next screen shows a Dashboard with all the chosen services installed , with green checks indicating normal function . See below screenshot.



Note : A few pointers about the dashboard :

- When you bring up the cluster, sometimes not all the services will be up and running, which will indicated in Red. To diagnose the problem , click on the service which will take you to a page which will show all the allied services of the main service. For e.g. , If HDFS is shown in red, click on HDFS , it will take you to a page which will show all allied services like Namenode, Secondary Namenode and DataNodes Running or not running. You could go to the service which is down and Start Up . This will show a dialog box indicating the background process of starting the chosen service.
- Some Processes have certain dependencies. Hence, if a certain service is down , it may be due to a dependent service which is down . For e.g. : If the NameNode is down , the DataNodes will also go down , in turn the HBase Master will also go down and hence the Zookeeper will go down , which in turn will bring down the Kafka Brokers. So,

identifying the key process is very important in these circumstances.

- c. Once you have started a certain service , it will take up sometime to bring up the service. This is due to the architecture of Ambari . Ambari Agents poll Ambari Server every 30 seconds , hence although the service might have been locally started on the slave nodes in a couple of seconds , the ambari agents will send that report to the ambari server in the next 30 second cycle , hence, it will take a while to show up on your dashboard.

26. Now that your cluster is up and Running , let us perform some sanity checks . They are elaborated in the following points .
27. HDFS Sanity Check – Click on the HDFS Service and then click on "Quick Links", this pops-out some options. Click on "Namenode UI". This will open a new tab on your browser and show the Namenode Web UI. Navigate to the bottom (right –hand side) and click on "Legacy UI". This will open another WebUI . Click on "Live Nodes". It will take you to a page which show the current health of your Datanodes. Check if the configured capacity matches to your chosen/ desired space, as shown in this table. If you find any difference , it means that you have not configured or missed some mount point. A Sample Screenshot :

NameNode 'solrmaster.cloudwick.com:8020'

Node	Transferring Address	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks	Block Pool Used (GB)	Block Pool Used (%)	Failed Volumes	Version
solrmaster	10.0.242.27.50010	0	In Service	58.06	0.40	2.69	54.97	0.68		94.69	369	0.40	0.68	0	2.6.0.2.2.4.2.2
solrslave1	10.0.237.227.50010	2	In Service	58.06	0.40	2.46	55.20	0.68		95.08	362	0.40	0.68	0	2.6.0.2.2.4.2.2
solrslave2	10.0.180.245.50010	1	In Service	58.06	0.40	2.70	54.96	0.68		94.67	362	0.40	0.68	0	2.6.0.2.2.4.2.2

Overview 'solrmaster.cloudwick.com:8020' (active)

Started:	Wed Jun 03 08:34:19 UTC 2015
Version:	2.6.0.2.4.2.2.224563eb448969d07902aed959ac13c652b2b272
Compiled:	2015-05-31T19:49Z by jenkins (no branch)
Cluster ID:	CID-7278344-17a6-41ff-9992-25320c4bcb38
Block Pool ID:	BP-212532269-0.0.242.27.1432847281961

Summary

Memory in HDFS
Secondary in HDFS
Secondary in HDFS
10270.768 MB available (27.3% of 37.3 GB)
Used Memory used 226.44 MB of 3000.00 MB Memory. Max-Heap Memory 61000.41 MB.
Max-Heap Memory used 40.41 MB of 132.21 MB Current Non-Heap Memory. Max-Non-Heap Memory 353.40 MB.

28. Similarly for other services , Click on the Service in the Ambari Dashboard , and then click on "Quick Links" and to their Respective UI. Here are a few more examples. Checking Yarn Services: Check the Resource Manager Web UI, Job History UI . Screenshots are as follows :



All Applications

Logged in as: dr.who

Cluster Metrics																
Cluster	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
Show 20+ entries																
	ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Uptime	History				
	application_1432847970095_0006	solr	com.lucidworks.hadoop.ingest.IngestJob	MAPREDUCE	default	Thu May 28 17:42:03	Thu May 28 17:43:47	FINISHED	SUCCEEDED			History				
	application_1432847970095_0005	hdfs	com.lucidworks.hadoop.ingest.IngestJob	MAPREDUCE	default	Thu May 28 17:42:03	Thu May 28 17:43:47	FINISHED	SUCCEEDED			History				
	application_1432847970095_0004	hdfs	com.lucidworks.hadoop.ingest.IngestJob	MAPREDUCE	default	Thu May 28 17:42:03	Thu May 28 17:43:47	FINISHED	FAILED			History				
	application_1432847970095_0003	solr	com.lucidworks.hadoop.ingest.IngestJob	MAPREDUCE	default	Thu May 28 17:42:03	Thu May 28 17:43:47	FINISHED	FAILED			History				
	application_1432847970095_0002	hdfs	QuasiMonteCarlo	MAPREDUCE	default	Thu May 28 17:42:03	Thu May 28 17:43:47	FINISHED	SUCCEEDED			History				
	application_1432847970095_0001	hdfs	QuasiMonteCarlo	MAPREDUCE	default	Thu May 28 17:42:03	Thu May 28 17:43:47	FINISHED	SUCCEEDED			History				



JobHistory

Logged in as: dr.who

Retired Jobs													
Application	Show 20+ entries	Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
	Job ID	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed	
	job_1432847970095_0005	2015/05/29 00:42:28 UTC	2015/05/29 00:43:46 UTC	job_1432847970095_0005	com.lucidworks.hadoop.ingest.IngestJob	solr	default	SUCCEEDED	2	2	0	0	
	job_1432847970095_0004	2015/05/29 00:42:28 UTC	2015/05/29 00:43:46 UTC	job_1432847970095_0004	com.lucidworks.hadoop.ingest.IngestJob	hdfs	default	SUCCEEDED	2	2	0	0	
	job_1432847970095_0003	2015/05/29 00:42:28 UTC	2015/05/29 00:43:46 UTC	job_1432847970095_0003	com.lucidworks.hadoop.ingest.IngestJob	solr	default	FAILED	0	0	0	0	
	job_1432847970095_0002	2015/05/29 00:42:28 UTC	2015/05/29 00:43:46 UTC	job_1432847970095_0002	QuasiMonteCarlo	hdfs	default	FAILED	1	1	0	0	
	job_1432847970095_0001	2015/05/29 00:42:28 UTC	2015/05/29 00:43:46 UTC	job_1432847970095_0001	QuasiMonteCarlo	hdfs	default	SUCCEEDED	16	16	1	1	

29. Check Storm UI :

Storm UI

Cluster Summary

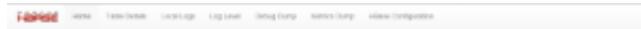
Version	Nimbus uptime	Supervisors	Used slots	Free slots	Total slots	Executors	Tasks
0.9.2.2.4.2-2	9h 14m 37s	3	0	6	6	0	0

Topology summary

Name	Id	Owner	Status	Uptime	Num workers	Num executors	Num tasks	Scheduler Info
Supervisor summary								
Id				Host		Uptime	Slots	Used slots
a3fe8feld1bd9-4676-80ca-1659bcfa654				solmaster.cloudwick.com		8h 7m 16s	2	0
c5fea880-5dbb-4d9d-a6cf-550973b3d68				solslave2.cloudwick.com		8h 4m 36s	2	0
c6uffa91-3d2f-4427-955b-28962892c5d8				solslave1.cloudwick.com		7h 44m 17s	2	0

Nimbus Configuration

30. Check HBase UI :



Master solmaster.cloudwick.com

Region Servers

Servername	Start time	Requests Per Second	Num Regions
solmaster.cloudwick.com (44333221434229)	Wed Jun 03 09:08:44 UTC 2015	0	1
solslave1.cloudwick.com (443332215480878)	Wed Jun 03 09:08:59 UTC 2015	0	0
solslave2.cloudwick.com (443332223327938)	Wed Jun 03 09:08:57 UTC 2015	0	1
Total		0	2

Backup Masters

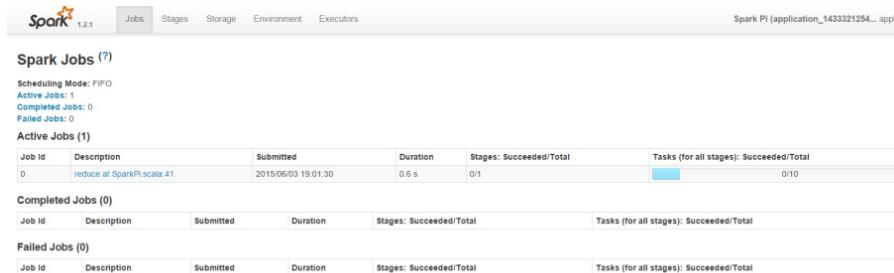
31. Check Oozie UI :



32. Check the Spark History Server UI. When you first check the UI, it will show the Spark UI , but will show a message something like :

No completed applications found!

Did you specify the correct logging directory? Please verify your setting of `spark.history.fs.logDirectory` and whether you have the permissions to access it. It is also possible that your application did not run to completion or did not stop the `SparkContext`. This is because we haven't run any spark applications yet. To do so, we can run a sample spark application. After you submit a spark application, refresh the page. You would see something like this :



33. If you see all the Web UIs until this point, your cluster is up and running !!

This completes your Ambari 2.0 / Ambari 2.1.1 Installation with HDP 2.2 / HDP 2.3 on CentOS