EMC ISILON HDP 2.3 AND AMBARI 2.1 INSTALLATION GUIDE

Version : 1.0

Table of Contents

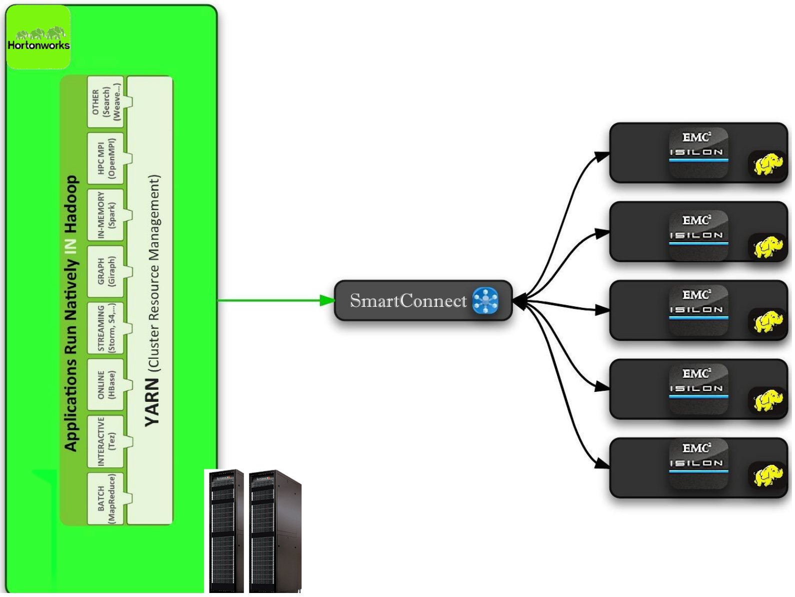
Table of Contents

Overview

This document gives an overview of HDP Installation on Isilon.

Architecture

In installing Hadoop with Isilon, the key difference is that, **each** Isilon Node contains a Hadoop Compatible **NameNode** and **DataNode**.



.

The compute and the storage are on separate set of node unlike a common of Hadoop Architecture.
HDP will connect to the EMC Isilon cluster using EMC SmartConnect. SmartConnect allows for transparent failover in case an Isilon Node Failure. The transactions will seamlessly failover to the new nodes. Isilon has various storage architectures which allows fast recovery of the failed nodes and file level data protection settings.

Certification

EMC Isilon has 2 steps with their release process

- Step 1 – Compliance Compatible
- Step 2 – Certified with HWX certification suite

EMC will release its support for Hortonworks HDP, stating it is compliance compatible.
There is a few months lag to get the complete in depth certification with Hortonworks Certification suite.

The below is compliance compatible.

Isilon Release	Additional Isilon Patch	Ambari Support	HDP Support
7.2.0.3	Patch 159065	2.1.0*	HDP 2.3

* - Only 2.1.0 (support for 2.1.1 is planned in 7.2.0.4 and 7.2.1.* in October/Nov, 2015 timeframe).

Ranger – Ranger plugin's except HDFS plugin works. This has been tested internally in HWX labs and is being tested. HDFS plugin will be worked in the future releases. Please contact Isilon product team for more details.

Note – WebHDFS port for Isilon is on on port 8082 and not 50070

High Level Steps

Please read the steps carefully as there are special deviations that is applied to attach the Isilon nodes to the HDP Compute cluster.

There are main steps

- Isilon Zones and IP Pool Configuration
- Ambari Install and Configuration
- Post Ambari Install Validation

Some key Isilon concepts

- Isilon is a cluster of hardware nodes, each node has its own CPU, Memory and Storage. Each node has 2 network interfaces – Backend and Frontend.
    - The backend is connected over Infiniband Network interface
    - The front end is 10 GB Ethernet interface
- The Isilon cluster run on OneFS, which is based on FreeBSD(Unix)
- Isilon provides Data Protection via mirroring or Reed Solomon FEC.
- Isilon has its own Data Management Console

Access Zones

Access zones, provides a method to logically partition cluster access and allocate resources to self-contained units, thereby providing a shared tenant environment.

In other words, it allows Isilon OneFs to segment the cluster configuration and separate the data into multiple self-contained units with their own sets of authentication providers; user mapping rules, and SMB shares.

A Hadoop/HDP Cluster will connect to a single Isilon zone. This is a one to one mapping.

This is part of Isilon Administration. Please work with your isilon administrator to create the needed isilon zone.

Useful video on zone:
https://www.youtube.com/watch?v=hF3W8o-n-Oo

By default, OneFS includes a single access zone called System**. You should not use the System zone for your cluster creation**.

Prepare Isilon zone

Before you create a zone, ensure that you are on 7.2.0.3 and installed the patch 159065.

(Note: both Hortonworks and Isilon team has access to download the patch from support.hortonworks.com)

Follow the following steps are needed:

1. Create a Isilon zone
2. Attach a pool of ip addresses to the zone
3. Assign a working directory to the zone
4. Create the needed users

Create a Zone

- Decide on a Zone Name. Ensure that the new zone that you want to create does not exist.

- For the purpose of example we will call the zone "**zonehdp**". You can name it to your organization's liking. Replace it with the version name that you want to assign.

```
1. hwxisi1-1# isi zone zones list
```

- **/ifs** is the default share across the nodes. Create a new directory for your zone under a directory "isitest".
  Isitest is just another hierarchy for the documentation purpose.

```
   hwxisi1-1# mkdir -p /ifs/isitest/zonehdp
```

- Create the zone

```
2. hwxisi1-1# isi zone zones create --name zonehdp --path
   /ifs/isitest/zonehdp
```

Attach a pool of ip addresses to the zone

- Associate an IP address pool with the zone. In this step you are creating the pool. Get the pool from your Isilon Admin. In this step replace the pool name, ip address range and zonename to an appropriate value.

```
     hwxisi1-1# isi networks create pool --name
     subnet0:poolhdp --ranges 172.18.150.110-172.18.150.119
     --access-zone zonehdp --ifaces=1:10gige-1+2:10gige-
     1+3:10gige-1+4:10gige-1+5:10gige-1+6:10gige-
     1+7:10gige-1+8:10gige-1
```

Assign a working directory to the zone

- Create the HDFS root directory. This is usually called *hadoop* and must be within the access zone directory.
- Set the HDFS root directory for the access zone
- Create an indicator file so that we can easily determine when we are looking your Isilon cluster via HDFS.

```
hwxisi1-1# mkdir -p /ifs/isitest/zonehdp/hadoop

hwxisi1-1# isi zone zones modify zonehdp --hdfs-root-
directory /ifs/isitest/zonehdp/hadoop;

hwxisi1-1# touch
/ifs/isitest/zonehdp/hadoop/THIS_IS_ISILON_isitest_zonehdp
```

- Check the hdfs thread settings and Block Size. If it is not set, set it using the isilon documentation in the appendix. . This is a one time activity

```
hwxisi1-1# isi hdfs settings view
   Default Block Size: 128M
Default Checksum Type: none
     Server Log Level: notice
       Server Threads: 256
```

```
hwxisi1-1# isi hdfs settings modify --server-threads 256
hwxisi1-1# isi hdfs settings modify --default-block-size 128M
```

Create the users and directories

- The scripts can be downloaded from (Claudio's github url. EMC Engineering officially supports this.)
  - https://github.com/claudiofahey/isilon-hadoop-tools/tree/master/onefs
- Extract the Isilon Hadoop Tools to your Isilon cluster. This can be placed in any directory under /ifs. It is recommended to use /ifs/*isitest*/scripts.
- Execute the script.

```
hwxisi1-1# bash /ifs/isitest/scripts/isilon-hadoop-
tools/onefs/isilon_create_users.sh --dist hwx --startgid
501 --startuid 501 --zone zonehdp

hwxisi1-1# bash /ifs/isitest/scripts/isilon-hadoop-
tools/onefs/isilon_create_directories.sh --dist hwx --
fixperm --zone zonehdp
```

Bug 30896 – Work Around

Map the *hdfs* user to the Isilon superuser. This will allow the *hdfs* user to chown (change ownership of) all files

```
hwxisi1-1# isi zone zones modify --user-mapping-rules="hdfs=>root" --
zone zonehdp
```

Permissions to root directory

Get the ZoneID from the following

```
isi zone zones view zonehdp
```

Replace the zoneid in the following command and execute it.

```
isi_run -z <zoneid>  "chown -R hdfs
/ifs/isitest/zonehdp/hadoop"
```

Restart Services

The command below will restart the HDFS service on Isilon to ensure that any cached user mapping rules are flushed. This will temporarily interrupt any HDFS connections coming from other Hadoop clusters

```
hwxisi1-1# isi services isi_hdfs_d disable ; isi services
isi_hdfs_d enable
```

Now you have completed the step in Isilon. We will now move to installing Hortonworks HDP 2.2. on the compute nodes. The Insatalltion will be performed using Apache Ambari. 1.7

Install Ambari Server

Ambari Server makes installation, configuration, management and monitoring of hadoop cluster simpler. Isilon zones have of Ambari Agent running on the Isilon Cluster.

Ambari server will be used to deploy HDP 2.3 to setup the hadoop cluster with HDP. Please follow the Hortonworks Installation Document for ensuring the pre-requisites for environment match

http://docs.hortonworks.com/HDPDocuments/Ambari-2.1.0.0/bk_Installing_HDP_AMB/content/_download_the_ambari_repo.html

The below steps are for CentOs 6 environment. Follow the steps from the Ambari Installation guide.

1. Complete the environment pre-requisites mentioned in the install guide.

2. Install the Ambari Server packages.

```
[root@hadoopmanager-server-0 ~]# wget -nv http://public-repo-
1.hortonworks.com/ambari/centos6/2.x/updates/2.1.0/ambari.repo -O
/etc/yum.repos.d/ambari.repo

[root@hadoopmanager-server-0 ~]# yum install ambari-server
```

3. Setup Ambari Server.

```
[root@hadoopmanager-server-0 ~]# ambari-server setup
```

4. Accept all defaults and complete the setup process.

5. Start the server.

```
[root@hadoopmanager-server-0 ~]# ambari-server start
```

6. Browse to `http://<ambari-host>:8080/`.
7. Login using the following account:

   Username: admin

   Password: admin

Deploy a Hortonworks Hadoop Cluster with Isilon for HDFS

You will deploy Hortonworks HDP Hadoop using the standard process defined by Hortonworks. Ambari Server allows for the immediate usage of an Isilon cluster for all HDFS services (NameNode and DataNode), no reconfiguration will be necessary once the HDP install is completed.

1. Configure the Ambari Agent on Isilon.

```
isiloncluster1-1# isi zone zones modify zonehdp --hdfs-ambari-namenode \ <smartconnectip/ip from ip pool>

 isiloncluster1-1# isi zone zones modify zonehdp --hdfs-ambari-server <hostname/ip of the ambari server>
```

2. Login to Ambari Server.
3. **Welcome:** Specify the name of your cluster *mycluster1*.
4. **Select Stack:** Select the HDP 2.2 stack.

Install Option:

Ambari Agent is already installed with Isilon OneFS. There are 2 ways of doing the following step. You can install the Ambari Agent on the compute nodes, then you do not need to go back register the Isilon host separately.

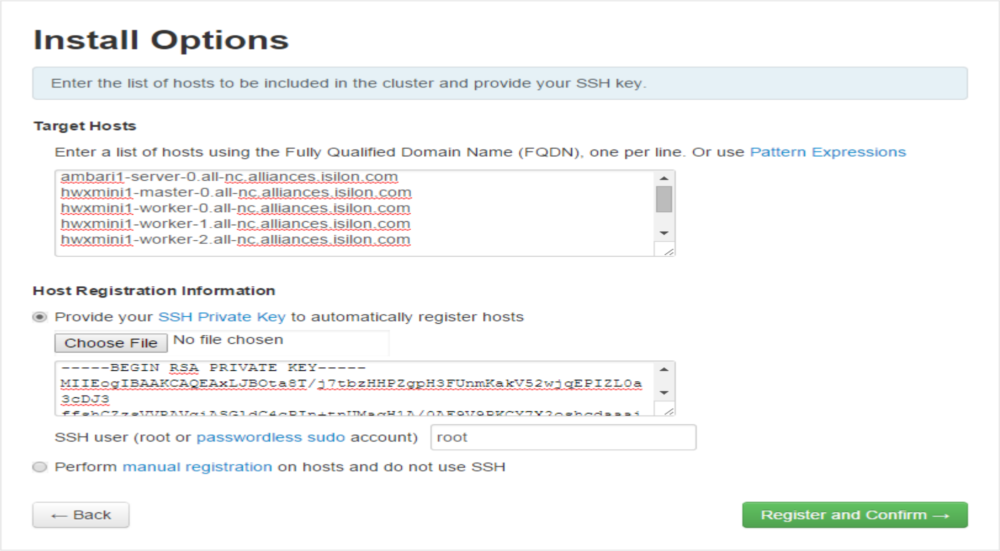
In the below steps you are installing the agent using Ambari UI wizard, and that is the reason you are going back to register the Agent.

Note

You will register your hosts with Ambari in two steps. First you will deploy the Ambari agent to your Linux hosts that will run HDP. Then you will go *back* one step and add Isilon.

1. Specify your Linux hosts for the compute nodes that will run HDP master components and slave components for your HDP cluster installation in the Target Hosts text box.

   Put in the ssh key



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

```
ambari1-server-0.all-nc.alliances.isilon.com
hwxmini1-master-0.all-nc.alliances.isilon.com
hwxmini1-worker-0.all-nc.alliances.isilon.com
hwxmini1-worker-1.all-nc.alliances.isilon.com
hwxmini1-worker-2.all-nc.alliances.isilon.com
```

Host Registration Information

● Provide your SSH Private Key to automatically register hosts

No file chosen

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAxLJBOta8T/j7tbzHHPZgpH3FUnmKakV52wjqEPIZL0a
3cDJ3
```

SSH user (root or passwordless sudo account) root

○ Perform manual registration on hosts and do not use SSH

   Click the **Next** button to deploy the Ambari Agent to your Linux hosts and register them.

2. Once the Ambari Agent has been deployed and registered on your Linux hosts, click the **Back** button.

   Now you will add the **SmartConnect** address of the Isilon cluster (mycluster1-hdfs.lab.example.com) to your list of target hosts.

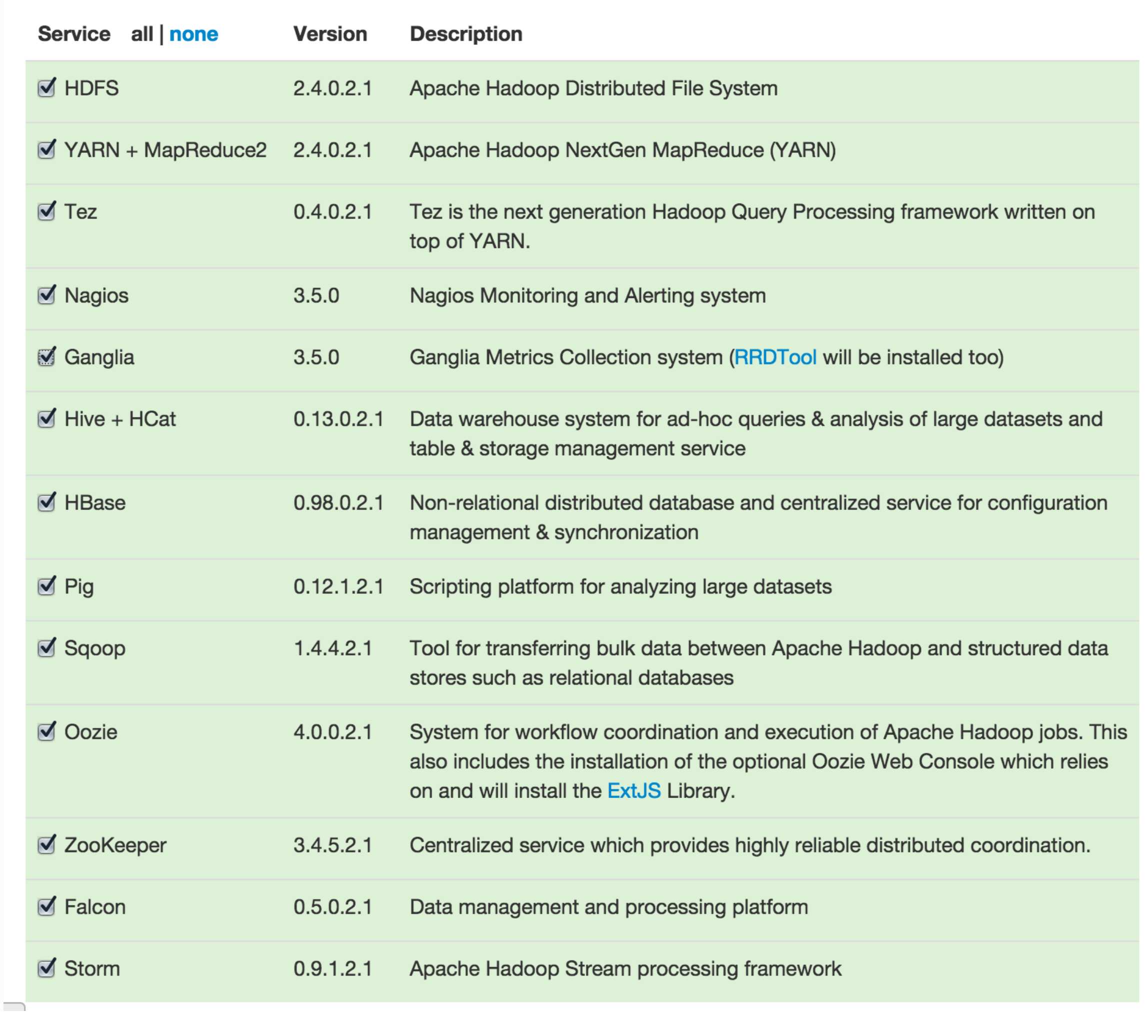
Check the box to "Perform manual registration on hosts and do not use SSH."

Click the **Next** button. You should see that Ambari agents on all hosts, including your Linux hosts and Isilon, become registered.

If SmartConnect is not available pick one IP Address from the IP Address pool.

5. **Choose Services:**

Select all the services.



Service all | none	Version	Description
☑ HDFS	2.4.0.2.1	Apache Hadoop Distributed File System
☑ YARN + MapReduce2	2.4.0.2.1	Apache Hadoop NextGen MapReduce (YARN)
☑ Tez	0.4.0.2.1	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
☑ Nagios	3.5.0	Nagios Monitoring and Alerting system
☑ Ganglia	3.5.0	Ganglia Metrics Collection system (RRDTool will be installed too)
☑ Hive + HCat	0.13.0.2.1	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
☑ HBase	0.98.0.2.1	Non-relational distributed database and centralized service for configuration management & synchronization
☑ Pig	0.12.1.2.1	Scripting platform for analyzing large datasets
☑ Sqoop	1.4.4.2.1	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
☑ Oozie	4.0.0.2.1	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 ExtJS Library.
☑ ZooKeeper	3.4.5.2.1	Centralized service which provides highly reliable distributed coordination.
☑ Falcon	0.5.0.2.1	Data management and processing platform
☑ Storm	0.9.1.2.1	Apache Hadoop Stream processing framework

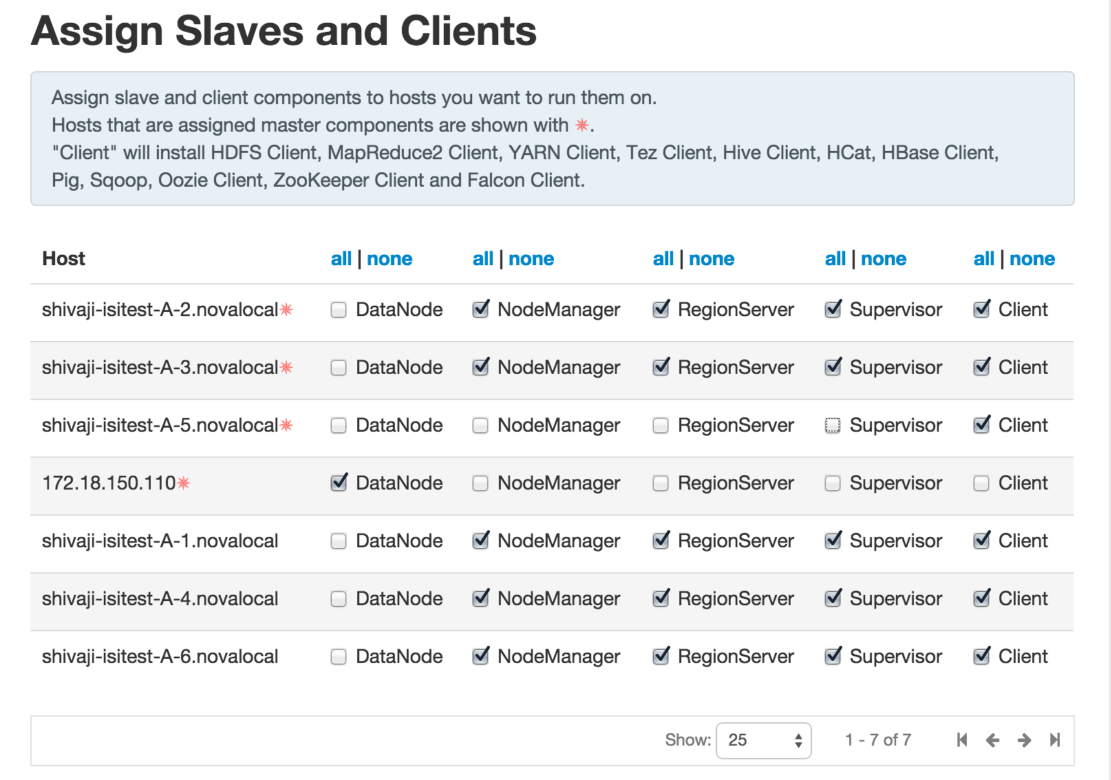
6. **Assign Masters:**

Assign NameNode and SNameNode components to the Isilon SmartConnect address.

ZooKeeper should be installed on mycluster1-master-0 and any two workers.

All other master components can be assigned to the master Node or Compute Nodes.

7. **Assign Slaves and Clients:**



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, Hive Client, HCat, HBase Client, Pig, Sqoop, Oozie Client, ZooKeeper Client and Falcon Client.

Host	all | none	all | none	all | none	all | none	all | none
shivaji-isitest-A-2.novalocal*	□ DataNode	☑ NodeManager	☑ RegionServer	☑ Supervisor	☑ Client
shivaji-isitest-A-3.novalocal*	□ DataNode	☑ NodeManager	☑ RegionServer	☑ Supervisor	☑ Client
shivaji-isitest-A-5.novalocal*	□ DataNode	□ NodeManager	□ RegionServer	□ Supervisor	☑ Client
172.18.150.110*	☑ DataNode	□ NodeManager	□ RegionServer	□ Supervisor	□ Client
shivaji-isitest-A-1.novalocal	□ DataNode	☑ NodeManager	☑ RegionServer	☑ Supervisor	☑ Client
shivaji-isitest-A-4.novalocal	□ DataNode	☑ NodeManager	☑ RegionServer	☑ Supervisor	☑ Client
shivaji-isitest-A-6.novalocal	□ DataNode	☑ NodeManager	☑ RegionServer	☑ Supervisor	☑ Client

Show: 25 1 - 7 of 7

Assign Data Node to the SmartConnect Isilon Node.

The rest to the compute Nodes.

8. **Customize Services:**

- Change the Webhdfs port from 50070 to 8082.
- Assign passwords to Hive, Oozie, and any other selected services that require them.
- Check that all local data directories are within /data/1, /data/2, etc. The following settings should be checked.

    1. YARN Node Manager log-dirs
    2. YARN Node Manager local-dirs
    3. HBase local directory
    4. ZooKeeper directory
    5. Oozie Data Dir
    6. Storm storm.local.dir

  In YARN, set yarn.timeline-service.store-class to
  `org.apache.hadoop.yarn.server.timeline.LeveldbTimelineStore`.

9. **Review:** Carefully review your configuration and then click Deploy.

10. After a successful installation, Ambari will start and test all of the selected services. Sometime it may fail for the first time around. You may need to retry couple of times. Review the Install, Start and Test page for any warnings or errors. It is recommended to correct any warnings or errors before continuing.

Adding a Hadoop User

You must add a user account for each Linux user that will submit MapReduce jobs. The procedure below can be used to add a user named hduser1.

Warning

The steps below will create local user and group accounts on your Isilon cluster. If you are using a directory service such as Active Directory, and you want these users and groups to be defined in your directory service, then DO NOT run these steps. Instead, refer to the OneFS documentation and EMC Isilon Best Practices for Hadoop Data Storage.

1. Add user to Isilon.

```
isiloncluster1-1# isi auth groups create hduser1 --zone
zone1 \ --provider local isiloncluster1-1

# isi auth users create hduser1 --primary-group hduser1 \
--zone zone1 --provider local \ --home-directory
/ifs/isiloncluster1/zone1/hadoop/user/hduser1
```

2. Add user to Hadoop nodes. Usually, this only needs to be performed on the master-0 node.

```
[root@mycluster1-master-0 ~]# adduser hduser1
```

3. Create the user's home directory on HDFS. In the below command you sudo as **hdfs** and then executing the "**hdfs**" command.

```
[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -
mkdir -p /user/hduser1

[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -
chown hduser1:hduser1 \ /user/hduser1

[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -
chmod 755 /user/hduser1
```

Validation

Ambari Service Check

Ambari has built-in functional tests for each component. These are executed automatically when you install your cluster with Ambari. To execute them after installation, select the service in Ambari, click the *Service Actions* button, and select *Run Service Check*.

Functional Tests

The tests below should be performed to ensure a proper installation. Perform the tests in the order shown.

You must create the Hadoop user *hduser1* before proceeding.

HDFS

```
[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -ls /
Found 5 items -rw-r--r--   1 root  hadoop              0
2014-08-05 05:59 /THIS_IS_ISILON drwxr-xr-x   - hbase
hbase             148 2014-08-05 06:06 /hbase drwxrwxr-
x   - solr  solr               0 2014-08-05 06:07 /solr
drwxrwxrwt   - hdfs  supergroup       107 2014-08-05 06:07
/tmp drwxr-xr-x   - hdfs  supergroup       184 2014-08-05
06:07 /user

[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -put -f
/etc/hosts /tmp

[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -cat
/tmp/hosts 127.0.0.1 localhost

[root@mycluster1-master-0 ~]# sudo -u hdfs hdfs dfs -rm -
skipTrash /tmp/hosts

[root@mycluster1-master-0 ~]# su - hduser1

[hduser1@mycluster1-master-0 ~]$ hdfs dfs -ls / Found 5
items -rw-r--r--   1 root  hadoop              0 2014-08-05
05:59 /THIS_IS_ISILON drwxr-xr-x   - hbase
hbase             148 2014-08-05 06:28 /hbase drwxrwxr-
x   - solr  solr               0 2014-08-05 06:07 /solr
drwxrwxrwt   - hdfs  supergroup       107 2014-08-05 06:07
/tmp drwxr-xr-x   - hdfs  supergroup       209 2014-08-05
06:39 /user

[hduser1@mycluster1-master-0 ~]$ hdfs dfs -ls ...
```

YARN / MapReduce

```
[hduser1@mycluster1-master-0 ~]$ hadoop jar \
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar \
pi 10 1000 ... Estimated value of Pi is
3.14000000000000000000

[hduser1@mycluster1-master-0 ~]$ hadoop fs -mkdir in
```

You can put any file into the *in* directory. It will be used the datasource for subsequent tests.

```
[hduser1@mycluster1-master-0 ~]$ hadoop fs -put -f
/etc/hosts in

[hduser1@mycluster1-master-0 ~]$ hadoop fs -ls in ...
[hduser1@mycluster1-master-0 ~]$ hadoop fs -rm -r out

[hduser1@mycluster1-master-0 ~]$ hadoop jar \
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar \
wordcount in out ... [hduser1@mycluster1-master-0 ~]$
hadoop fs -ls out Found 4 items -rw-r--r--   1 hduser1
hduser1          0 2014-08-05 06:44 out/_SUCCESS -rw-r--r--
   1 hduser1 hduser1         24 2014-08-05 06:44 out/part-
r-00000 -rw-r--r--   1 hduser1 hduser1          0 2014-08-
05 06:44 out/part-r-00001 -rw-r--r--   1 hduser1
hduser1          0 2014-08-05 06:44 out/part-r-00002

[hduser1@mycluster1-master-0 ~]$ hadoop fs -cat out/part*
localhost     1 127.0.0.1     1
```

Browse to the YARN Resource Manager GUI `http://mycluster1-master-0.lab.example.com:8088/`.

Browse to the MapReduce History Server GUI `http://mycluster1-master-0.lab.example.com:19888/`. In particular, confirm that you can view the complete logs for task attempts.

Hive

```
[hduser1@mycluster1-master-0 ~]$ hadoop fs -mkdir -p
sample_data/tab1

[hduser1@mycluster1-master-0 ~]$ cat - > tab1.csv
1,true,123.123,2012-10-24 08:55:00 2,false,1243.5,2012-10-
25 13:40:00 3,false,24453.325,2008-08-22 09:33:21.123
4,false,243423.325,2007-05-12 22:32:21.33454
5,true,243.325,1953-04-22 09:11:33

[hduser1@mycluster1-master-0 ~]$ hadoop fs -put -f tab1.csv
sample_data/tab1

[hduser1@mycluster1-master-0 ~]$ hive

hive> DROP TABLE IF EXISTS tab1; CREATE EXTERNAL TABLE tab1
(    id INT,    col_1 BOOLEAN,    col_2 DOUBLE,    col_3
TIMESTAMP ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hduser1/sample_data/tab1';  DROP TABLE IF
EXISTS tab2;  CREATE TABLE tab2 (    id INT,    col_1
BOOLEAN,    col_2 DOUBLE,    month INT,    day INT ) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',';  INSERT
OVERWRITE TABLE tab2 SELECT id, col_1, col_2, MONTH(col_3),
DAYOFMONTH(col_3) FROM tab1 WHERE YEAR(col_3) = 2012;

OK Time taken: 28.256 seconds

hive> show tables;

OK tab1 tab2 Time taken: 0.889 seconds, Fetched: 2 row(s)

hive> select * from tab1;

OK

1      true   123.123       2012-10-24 08:55:00
2      false  1243.5        2012-10-25 13:40:00
3      false  24453.325     2008-08-22 09:33:21.123
4      false  243423.325    2007-05-12 22:32:21.33454
5      true   243.325       1953-04-22 09:11:33 Time taken:
1.083 seconds, Fetched: 5 row(s)
```

```
hive> select * from tab2; OK
1      true   123.123       10     24 2      false  1243.5
10     25 Time taken: 0.094 seconds, Fetched: 2 row(s)

hive> select * from tab1 where id=1; OK
1      true   123.123       2012-10-24 08:55:00 Time taken:
15.083 seconds, Fetched: 1 row(s)  hive> select * from tab2
where id=1;

OK 1      true   123.123       10     24 Time taken: 13.094
seconds, Fetched: 1 row(s)

hive> exit;
```

Pig

```
[hduser1@mycluster1-master-0 ~]$ pig

grunt> a = load 'in';

grunt> dump a; ... Success! ...

grunt> quit;
```

HBase

```
[hduser1@mycluster1-master-0 ~]$ hbase shell

hbase(main):001:0> create 'test', 'cf' 0 row(s) in 3.3680
seconds => Hbase::Table - test

hbase(main):002:0> list 'test' TABLE test 1 row(s) in
0.0210 seconds => ["test"]

hbase(main):003:0> put 'test', 'row1', 'cf:a', 'value1' 0
row(s) in 0.1320 seconds

hbase(main):004:0> put 'test', 'row2', 'cf:b', 'value2' 0
row(s) in 0.0120 seconds

hbase(main):005:0> scan 'test'
ROW                         COLUMN+CELL  row1
          column=cf:a,timestamp=1407542488028,value=value1
 row2                        column=cf:b,timestamp=14075424
99562,value=value2 2 row(s) in 0.0510 seconds

hbase(main):006:0> get 'test', 'row1'
COLUMN                      CELL  cf:a
   timestamp=1407542488028,value=value1 1 row(s) in 0.0240
seconds hbase(main):007:0> quit
```

Use Case - Searching Wikipedia

One of the many unique features of Isilon is its multi-protocol support. This allows you, for instance, to write a file using SMB (Windows) or NFS (Linux/Unix) and then read it using HDFS to perform Hadoop analytics on it.

In this section, we exercise this capability to download the entire Wikipedia database (excluding media) using your favorite browser to Isilon. As soon as the download completes, we'll run a Hadoop grep to search the entire text of Wikipedia using our Hadoop cluster. As this search doesn't rely on a word index, your regular expression can be as complicated as you like.

1. First, let's connect your client (with your favorite web browser) to your Isilon cluster.
    1. If you are using a Windows host or other SMB client:
        1. Click Start -> Run.
        2. Enter: `\\<Isilon Host>\ifs`
        3. You may authenticate as *root* with your Isilon root password.
        4. Browse to *\ifs\isiloncluster1\zone1\hadoop\tmp*.
        5. Create a directory here called *wikidata*. This is where you will download the Wikipedia data to.
    2. If you are using a Linux host or other NFS client:
        1. Mount your NFS export.

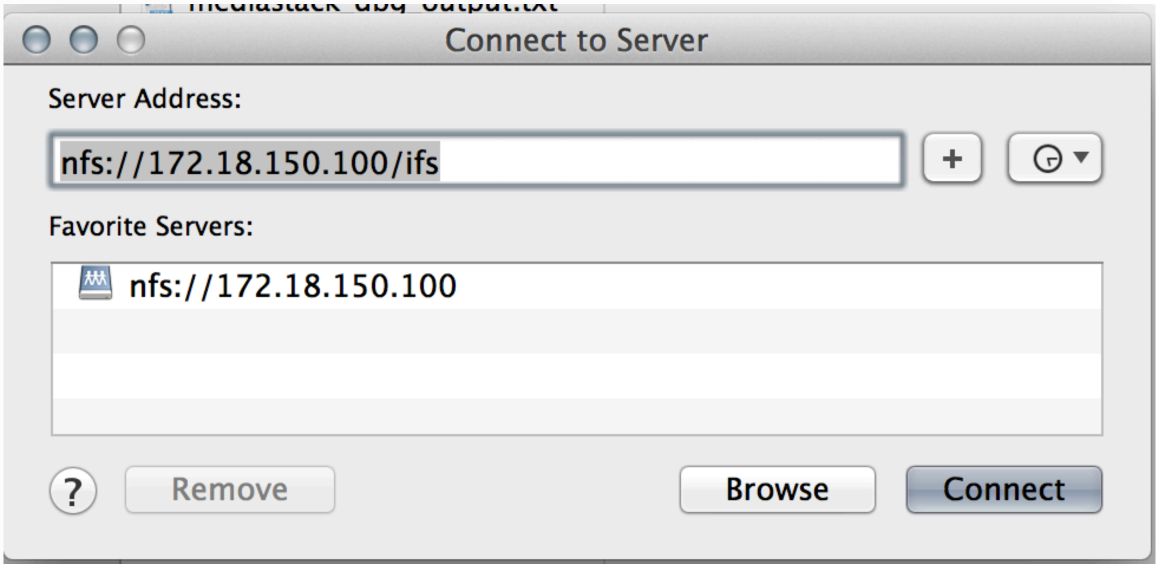
```
[root@workstation ~]$ mkdir /mnt/isiloncluster1
[root@workstation ~]$ echo \ subnet0-
pool0.isiloncluster1.lab.example.com:/ifs \
/mnt/isiloncluster1 nfs \
nolock,nfsvers=3,tcp,rw,hard,intr,timeo=600,retrans=
2,rsize=131072,wsize=524288 \ >> /etc/fstab

[root@workstation ~]$ mount -a
```

```
[root@workstation ~]$ mkdir -p \
/mnt/isiloncluser1/isiloncluster1/zone1/hadoop/tmp/w
ikidata
```

2. On Mac

   How to create an NFS Mount

   http://support.apple.com/kb/TA22243

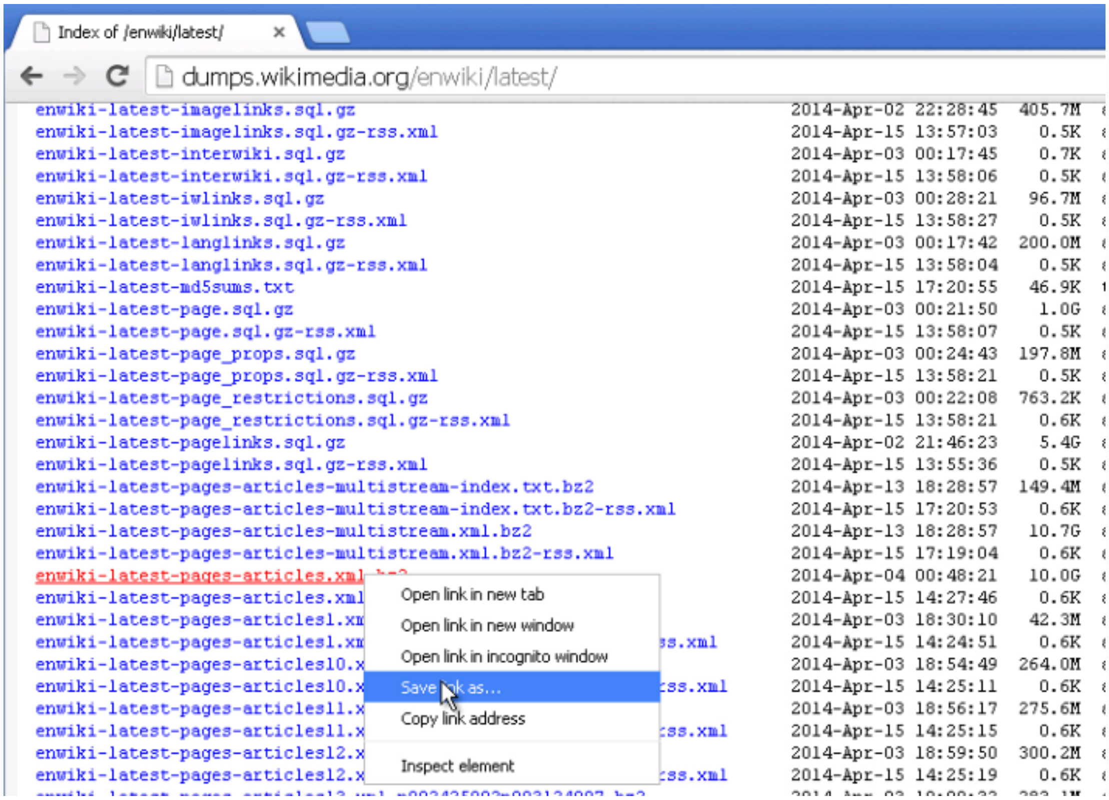


Server Address: nfs://172.18.150.100/ifs

Favorite Servers:

nfs://172.18.150.100

3. On our favorite web browser and go to http://dumps.wikimedia.org/enwiki/latest.



```
enwiki-latest-imagelinks.sql.gz                                          2014-Apr-02 22:28:45  405.7M
enwiki-latest-imagelinks.sql.gz-rss.xml                                  2014-Apr-15 13:57:03    0.5K
enwiki-latest-interwiki.sql.gz                                           2014-Apr-03 00:17:45    0.7K
enwiki-latest-interwiki.sql.gz-rss.xml                                   2014-Apr-15 13:58:06    0.5K
enwiki-latest-iwlinks.sql.gz                                             2014-Apr-03 00:28:21   96.7M
enwiki-latest-iwlinks.sql.gz-rss.xml                                     2014-Apr-15 13:58:27    0.5K
enwiki-latest-langlinks.sql.gz                                           2014-Apr-03 00:17:42  200.0M
enwiki-latest-langlinks.sql.gz-rss.xml                                   2014-Apr-15 13:58:04    0.5K
enwiki-latest-md5sums.txt                                                2014-Apr-15 17:20:55   46.9K
enwiki-latest-page.sql.gz                                                2014-Apr-03 00:21:50    1.0G
enwiki-latest-page.sql.gz-rss.xml                                        2014-Apr-15 13:58:07    0.5K
enwiki-latest-page_props.sql.gz                                          2014-Apr-03 00:24:43  197.8M
enwiki-latest-page_props.sql.gz-rss.xml                                  2014-Apr-15 13:58:21    0.5K
enwiki-latest-page_restrictions.sql.gz                                   2014-Apr-03 00:22:08  763.2K
enwiki-latest-page_restrictions.sql.gz-rss.xml                           2014-Apr-15 13:58:21    0.6K
enwiki-latest-pagelinks.sql.gz                                           2014-Apr-02 21:46:23    5.4G
enwiki-latest-pagelinks.sql.gz-rss.xml                                   2014-Apr-15 13:55:36    0.5K
enwiki-latest-pages-articles-multistream-index.txt.bz2                   2014-Apr-13 18:28:57  149.4M
enwiki-latest-pages-articles-multistream-index.txt.bz2-rss.xml           2014-Apr-15 17:20:53    0.6K
enwiki-latest-pages-articles-multistream.xml.bz2                         2014-Apr-13 18:28:57   10.7G
enwiki-latest-pages-articles-multistream.xml.bz2-rss.xml                 2014-Apr-15 17:19:04    0.6K
enwiki-latest-pages-articles.xml.bz2                                     2014-Apr-04 00:48:21   10.0G
enwiki-latest-pages-articles.xml                                         2014-Apr-15 14:27:46    0.6K
enwiki-latest-pages-articles1.xm                                         2014-Apr-03 18:30:10   42.3M
enwiki-latest-pages-articles1.xm                           ss.xml        2014-Apr-15 14:24:51    0.6K
enwiki-latest-pages-articles10.x                                         2014-Apr-03 18:54:49  264.0M
enwiki-latest-pages-articles10.x                           ss.xml        2014-Apr-15 14:25:11    0.6K
enwiki-latest-pages-articles11.x                                         2014-Apr-03 18:56:17  275.6M
enwiki-latest-pages-articles11.x                           ss.xml        2014-Apr-15 14:25:15    0.6K
enwiki-latest-pages-articles12.x                                         2014-Apr-03 18:59:50  300.2M
enwiki-latest-pages-articles12.x                           ss.xml        2014-Apr-15 14:25:19    0.6K
```

4. Locate the file *enwiki-latest-pages-articles.xml.bz2* and download it directly to the *wikidata* folder on Isilon. Your web browser will be writing this file to the Isilon file system using SMB or NFS.

Note

This file is approximately 10 GB in size and contains the entire text of the English version of Wikipedia. If this is too large, you may want to download one of the smaller files such as*enwiki-latest-all-titles.gz*.

5. Now let's run the Hadoop grep job. We'll search for all two-word phrases that begin with *EMC*.

```
[hduser1@mycluster1-master-0 ~]$ hadoop fs -ls
/tmp/wikidata
```

```
[hduser1@mycluster1-master-0 ~]$ hadoop fs -rm -r
/tmp/wikigrep

[hduser1@mycluster1-master-0 ~]$ hadoop jar \
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar \
grep /tmp/wikidata /tmp/wikigrep "EMC [^ ]*"
```

6. When the the job completes, use your favorite text file viewer to view the output file*/tmp/wikigrep/part-r-00000*. You may open the file in a text editor from the NFS Mount