Hadoop Basics with IBM BigInsights

Unit 4: Hadoop Administration

An IBM Proof of Technology

Catalog Number

Contents

Lab 3 Hadoop Administration

IBM BigInsights 4.0 Enterprise Edition enables firms to store, process, and analyze large volumes of various types of data using a wide array of machines working together as a cluster. In this exercise, you'll learn some essential Hadoop administration tasks from expanding a cluster to ingesting large amounts of data into the Hadoop Distributed File System (HDFS).

After completing this hands-on lab, you'll be able to:

- Manage a cluster running BigInsights to add or remove nodes as necessary
- Cover essential Hadoop administration tasks such as expanding disk space and how to start and stop services

Allow 60 minutes to 90 minutes to complete this lab.

This version of the lab was designed using IBM BigInsights and was tested on BigInsights 4.0. Throughout this lab you will be using the following account login information:

	Username	**Password**
rvm	root	password
Ambari login	admin	admin

3.1 Managing a Hadoop Cluster by Adding/Removing Nodes

In this section you will learn how to:

- Add and remove nodes through Ambari Web Console

Since you are using a VM image with a preconfigured single node configuration **you will not execute the adding of a second node**, but merely *explore how* you would do so on some future occasion in a real cluster environment.

Accordingly the procedures adding and removing nodes for the BigInsights QSE 4.0 have not been tested fully for this lab exercise environment.

The steps listed with underscore (e.g., __3) are intended for you to take action. Other paragraphs are for your information only.

3.1.1 Prepare your environment

So far you have been working with just a single node cluster. To add a second node to the cluster, you would need to have a second VM image. For clarification purposes, the existing image will be referred to as the master image.

That second copy of your image would need to be stored in a different directory. Boot it and go through the same process of accepting the licenses that you did for the Master image. Specify the same password for the child image as you did for the Master image.

The users on all the cluster nodes need to have the same logins and id numbers. Thus the child image would need the same username.

For a node to be added to a BigInsights cluster, BigInsights cannot be installed. Thus BigInsights would need to be uninstalled on the child image

Child image

The hostname and IP address on the child image have to be different from the hostname and IP address on the master image (since it is a direct copy it would initially have the same hostname and IP address as the master.) You also need to update the /etc/hosts file so the child image can communicate with the master image.

Master image

On the master image, **/etc/hosts** will also need to be edited, by adding the hostname and IP address for the child image exactly the same as for the child image.

Hostname and IP address resolution must be the same from all members of a cluster, no matter what the size.

More details can be found online at IBM's Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSPT3X_4.0.0/com.ibm.swg.im.infosphere.biginsights.install.doc/doc/install_prepare.html?lang=en

3.1.2 Setting up ssh

One of the key parts of managing a Hadoop cluster is being able to scale the cluster with ease, adding and removing nodes as needed. Adding a node can be done through a range of methods, of which we will cover adding from a Ambari Web Console, and from a terminal. Each of these methods can achieve the same results.

Again, we are not executing these actions. The description here shows the method of working with the process of setting up a second (or later) node in a cluster.

Before proceeding with adding a node, you should first verify that you can access the node you are trying to add. This can be done by simply “sshing” (pronounced as ess-ess-aitch-ing) the given node(s) as follows.

On the *master* image, **t**ype the following ssh command to make sure that you have connectivity between the master and the child images: **`ssh root@rvm2.svl.ibm.com`**

When doing ssh on a new IP you will typically get an authenticity message:

```
The authenticity of host 'rvm2.svl.ibm.com (192.168.70.201)' can't be established.
RSA key fingerprint is 29:2f:72:9f:f4:97:16:89:cf:d9:cc:09:d3:16:d9:bf.
Are you sure you want to continue connecting (yes/no)?
```

Go ahead and type **yes**, you will then get a warning:

```
Warning: Permanently added 'bivm2.ibm.com,192.168.70.201' (RSA) to the list of known
hosts.
```

Enter the password for root on the child image.

Enter **`exit`** to leave the ssh connection.

3.1.3 Working with the Ambari Web Console

One of the great features of IBM BigInsights 4.1 is the Ambari Web Console. The web console a user-friendly way for performing the tasks associated with Hadoop and other service administration.

All of the following steps are done on the *Master node*. The BigInsights services must be started.

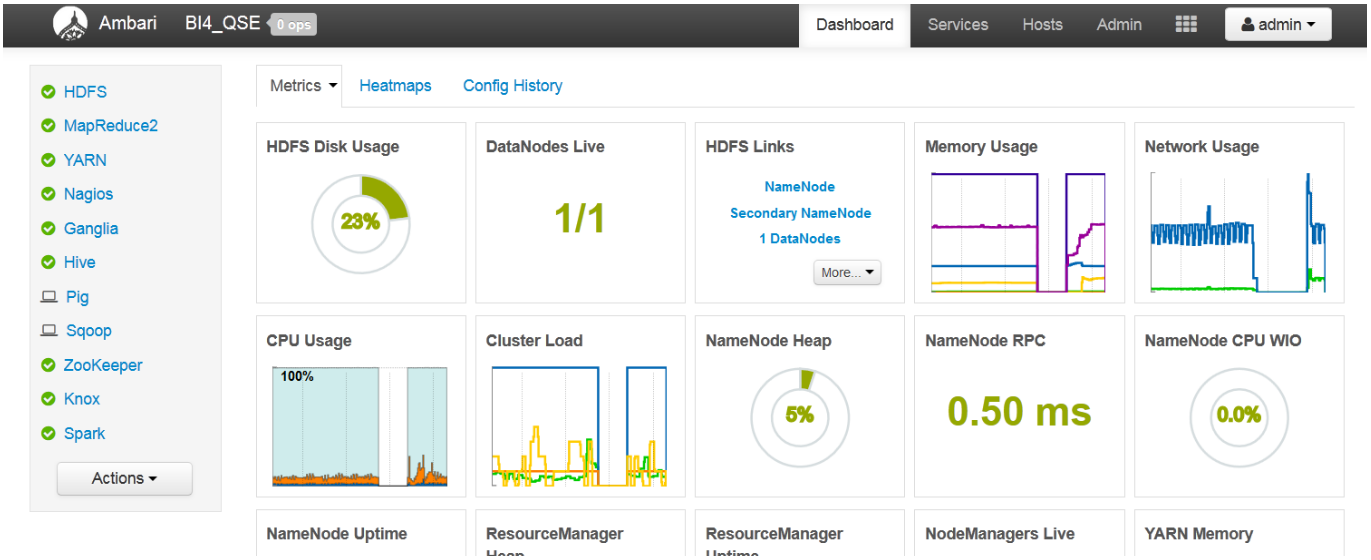
__1. Start the Ambari Web Console on your local machine:

Start the Firefox browser

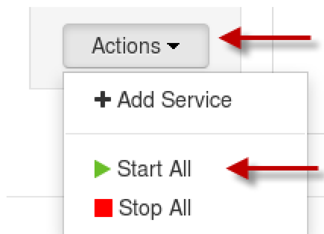
Use the URL: **rvm.svl.ibm.com:8080**

Sign in to Ambari (**admin** / **admin**)

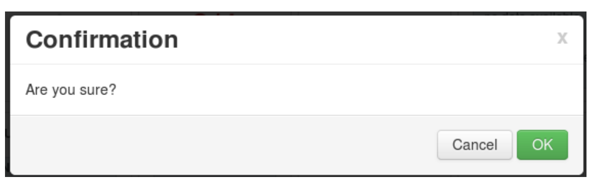
__2. When you are signed in, you will see the following Dashboard page. If the service on the left-hand side shows a *red-triangle with exclamation point*, the particular service is not currently running; if the individual service has a green-circle with check mark, the particular service is running.



To start all services, click **Actions** at the bottom of the left-hand side, and then **Start All**:

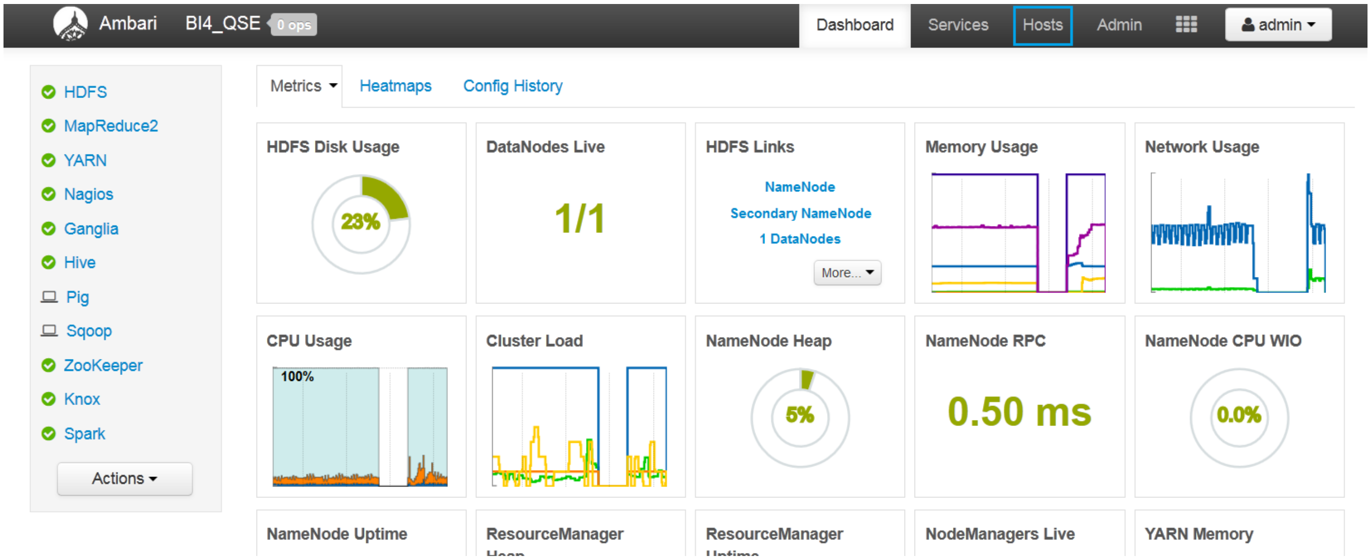


Confirm with **OK**:

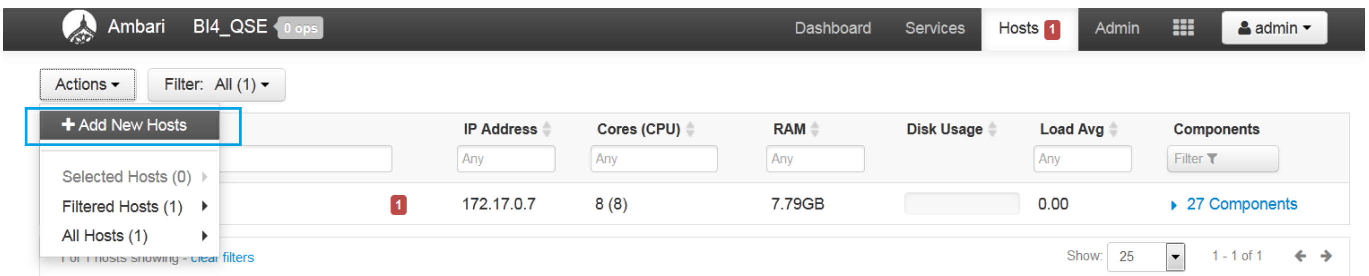


Once all components have started successfully as shown on the Ambari Web Console, you can proceed with the following actions using the Ambari Web Console.

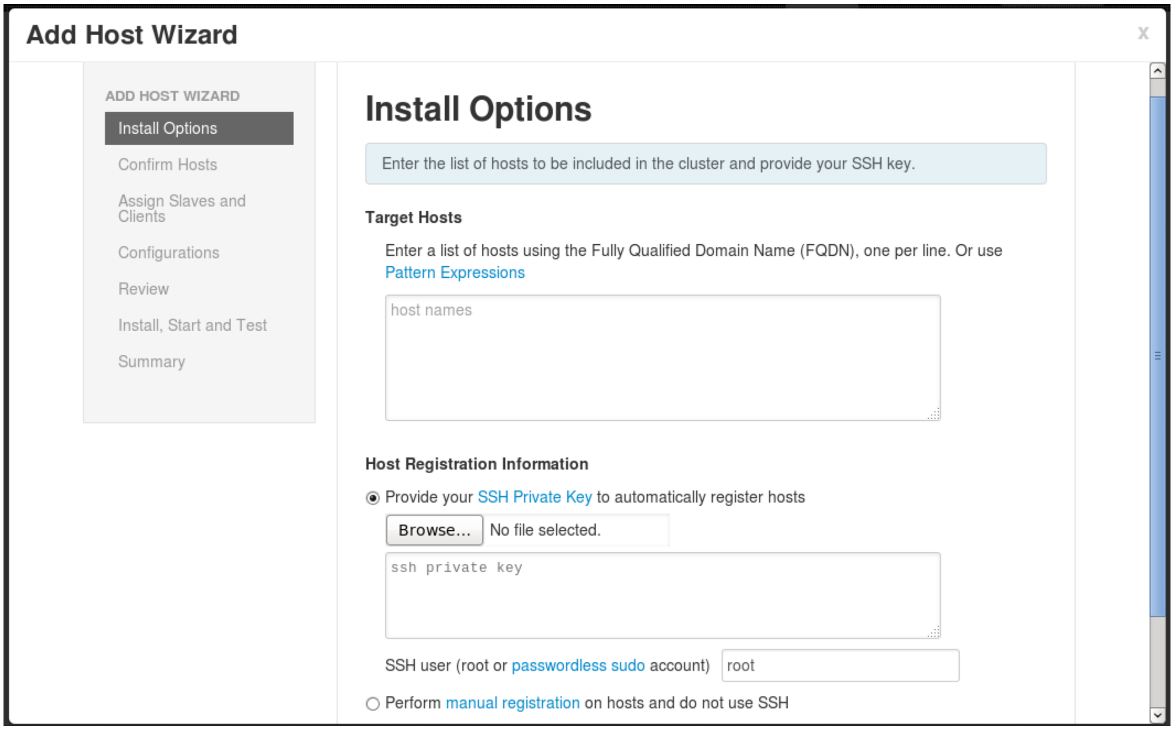
__3. You will now see on the Ambari Dashboard. To work with Adding / Deleting Hosts from the Ambari Web Console, select the **Hosts** tab.



__4. On the Hosts tab, select **Add New Hosts** from the **Actions** drop down on the left.



__5. This provides you with the places where you fill in the information for adding Hosts to your cluster:



This is a multi-step process:

- Install options
- Confirm Hosts
- Assign Slaves and Clients
- Configurations
- Review
- Install, Start, and Test
- Summary

Since we will not be actually performing this work, because we do not have another host to add to our cluster for this lab, review the meaning of these individual steps at:

https://ambari.apache.org/1.2.3/installing-hadoop-using-ambari/content/ambari-chap3-1.html

The remaining parts of this Hands-on Lab can be adequately done on a single-node cluster and thus the following steps should be done in full.

3.2 Managing a Hadoop Cluster

In this section you will learn how to:

- Check the health of the cluster and individual nodes within that cluster
- Perform checks on the disk and storage of HDFS

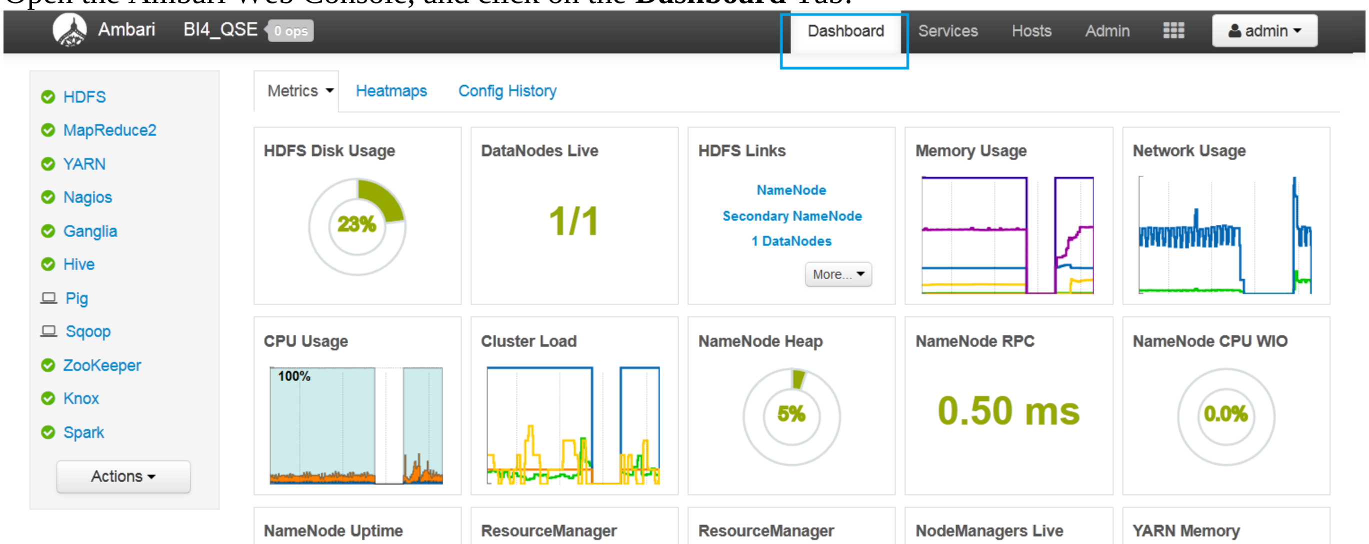
Typical Hadoop clusters rely on being able to use multiple inexpensive, commodity computers as nodes working together as a Hadoop cluster. Because of this, and the way in which hardware and hard disk drives operate from a mechanical point, the hardware is bound to fail over the years. Hadoop handles this efficiently by replicating the data across the various nodes (3-node replication by default).

3.2.1 Visual health check of a cluster using the Ambari Web Console

Servers, machines, and disk drives are all prone to a physical failure over time. When running a large cluster with dozens of nodes, it is crucial to over time maintain a constant health check of hardware and take appropriate actions when necessary. BigInsights v4.1 allows for a quick and simple way to perform these types of health checks on a cluster.

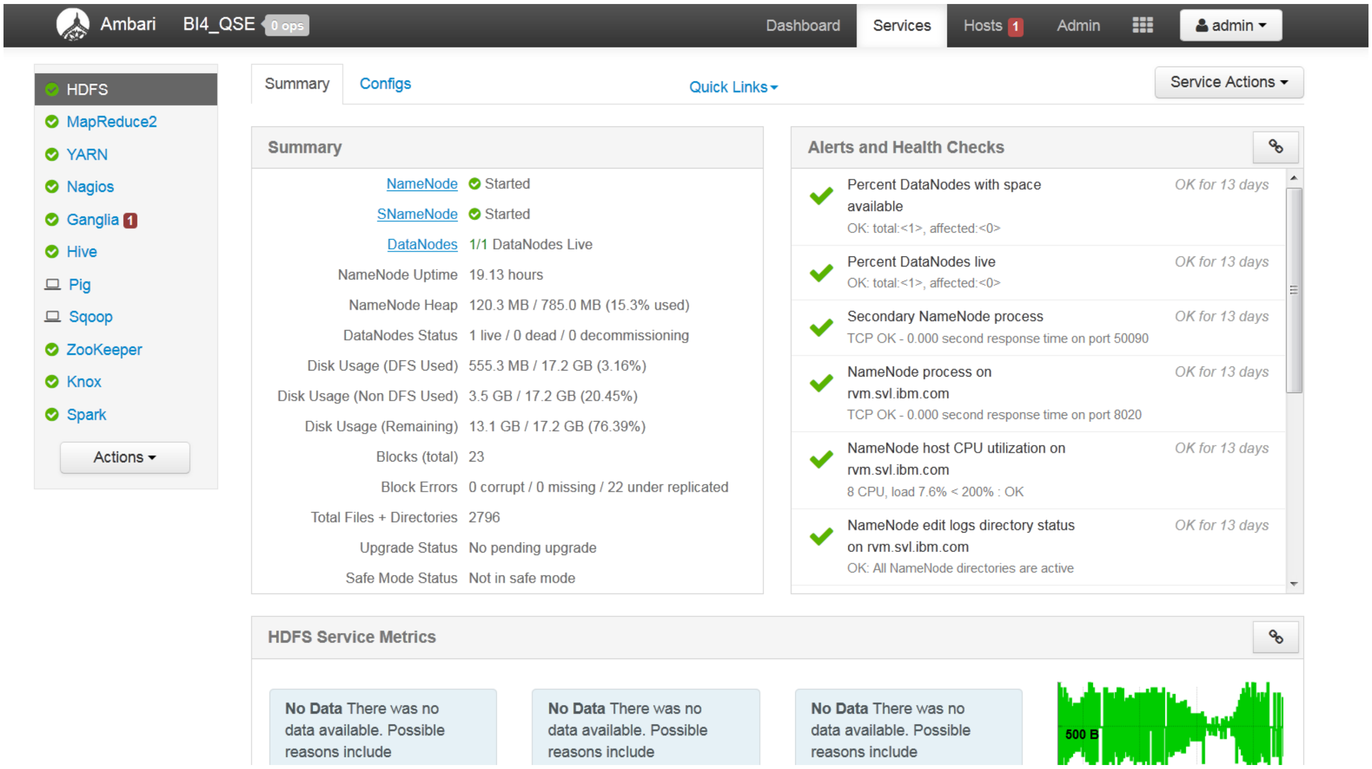
You can visually check the status of your cluster by following these simple steps that require you have a login on the Ambari Web Console:

__6. Open the Ambari Web Console, and click on the **Dashboard** Tab:



On the left-side check the list of services. If any service shows a red-triangle with an exclamation point, that service is not running.

__7. On the left-hand panel, click on **HDFS**. For a healthy disk system, you should see:



3.2.2 DFS disk check using a terminal window

There are various ways to monitoring the DFS Disk, and this should be done occasionally to avoid space issues which can arise if there is low disk storage remaining. One such issue can occur if the "hadoop healthcheck" or heartbeat as it is also referred to sees that a node has gone offline. If a node is offline for a certain period of time, the data that the offline node was storing will be replicated to other nodes (since there is a 3-node replication, the data is still available on the other 2 nodes). If there is limited disk space, this can quickly cause an issue.

Some of these commands require that you are logged in as an HDFS administrator, hdfs. You can login to hdfs by first going to root.

__8. In the bash window, enter the following commands:

```
su hdfs
```

__9. *While logged in as hdfs*, you can quickly access the dfs report by entering the following command:

```
hdfs dfsadmin –report
```

```
[hdfs@rvm /]$ hdfs dfsadmin -report
Configured Capacity: 18433347584 (17.17 GB)
Present Capacity: 14663487488 (13.66 GB)
DFS Remaining: 14081208320 (13.11 GB)
DFS Used: 582279168 (555.30 MB)
DFS Used%: 3.97%
Under replicated blocks: 22
Blocks with corrupt replicas: 0
Missing blocks: 0

-------------------------------------------------
Live datanodes (1):

Name: 172.17.0.7:50010 (rvm.svl.ibm.com)
Hostname: rvm.svl.ibm.com
Decommission Status : Normal
Configured Capacity: 18433347584 (17.17 GB)
DFS Used: 582279168 (555.30 MB)
Non DFS Used: 3769860096 (3.51 GB)
DFS Remaining: 14081208320 (13.11 GB)
DFS Used%: 3.16%
DFS Remaining%: 76.39%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 4
Last contact: Wed Jul 15 16:39:59 UTC 2015


[hdfs@rvm /]$ _
```

3.3 Hadoop Administration

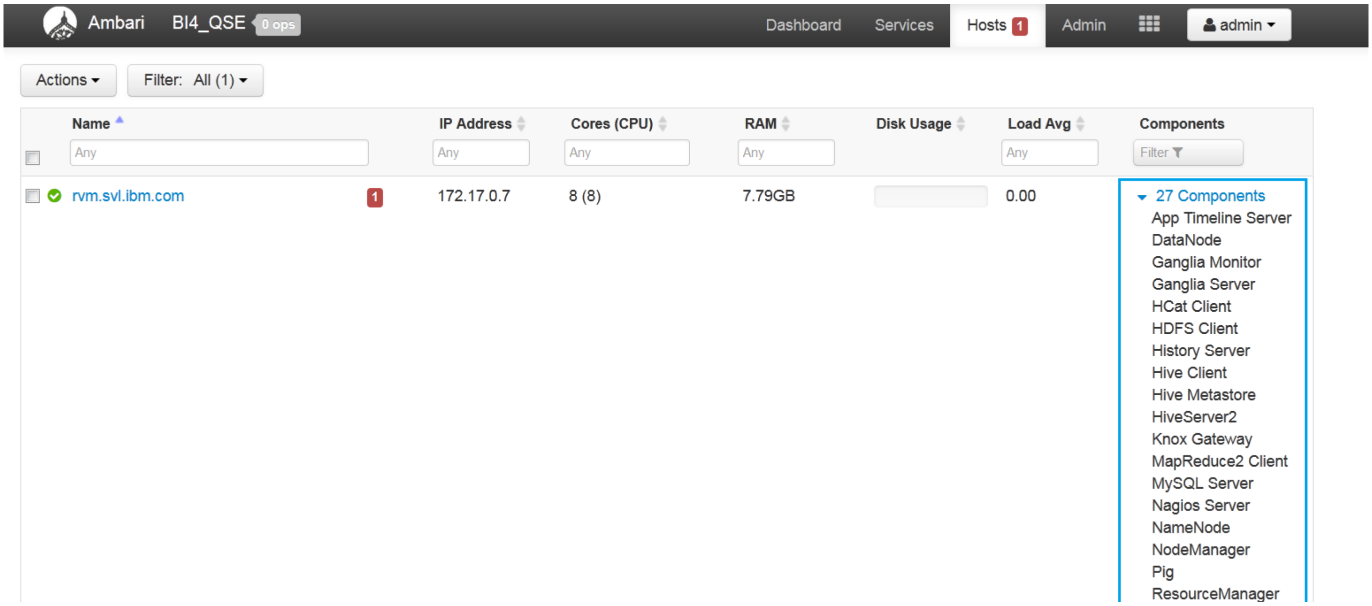
After completing this section, you’ll be able to:

- Start and stop individual services to best optimize the cluster performance
- Change default parameters within Hadoop such as the HDFS Block Size
- Manage service-specific slave nodes

3.3.1 Administering Specific Services

A single node can have a wide variety of services running at any given time, as seen in the screenshot below. Depending on your system and needs, it may not always be necessary to have all of the services running, as the more services running the more resources and computing power is being consumed by them.

In the Ambari Web Console, on the hosts tab, there are a list of hosts in the cluster. Here we have just one (rvm.svl.ibm.com), as listed in the first/left column. In the right-hand column here, you can see that 31 components are running. By clicking on 27 Components, you will get a list of the actual components (highlighted for this illustration as a vertical bar and arrow):



Stopping specific services can be done easily through the Ambari Web Console. We are not going to stop any services in this Hands-On Lab, but this is where you can find what is running.

3.3.2 Configuring Hadoop Default Settings

__10. The configuration files for Hadoop and the various services can be found in a number of configuration files that end with the name –**site.xml**. You can easily find these files by doing a search as **root**:

`exit` [if you were logged in as the hdfs user]

`find ./etc -name "*-site.xml"`

```
[root@rvm /]# find ./etc -name "*-site.xml"
./etc/hive/conf/hive-site.xml
./etc/hive/conf/mapred-site.xml
./etc/hive/conf.server/hive-site.xml
./etc/hive/conf.server/mapred-site.xml
./etc/hadoop/conf.empty/yarn-site.xml
./etc/hadoop/conf.empty/mapred-site.xml
./etc/hadoop/conf.empty/hdfs-site.xml
./etc/hadoop/conf.empty/core-site.xml
./etc/spark/conf/hive-site.xml
./etc/hive-webhcat/conf/webhcat-site.xml
./etc/knox/conf/gateway-site.xml
./etc/sqoop/conf/sqoop-site.xml
[root@rvm /]#
```

The principle ones, of interest here, are:

- core-site.xml — covers the Hadoop system
- hdfs-site.xml — covers HDFS specific configuration parameters
- mapred-site.xml — covers MapReduces specific configuration parameters
- yarn-site.xml — covers YARN specific configuration parameters

Ambari is used for Hadoop provisioning, management, and monitoring across the cluster. Thus, although we will look at and modify the configuration files locally here in this Hands-On Lab, Ambari or another mechanism is needed to synchronize these changes across the whole cluster.

3.3.3 Increasing Storage Block Size

There are certain attributes from Apache Hadoop which are imported, and some have been changed to improve performance. One such attribute is the default block size used for storing large files.

Consider the following short example. You have a 1GB file, on a 3-node replication cluster. With a block-size of 128MB, this file will be split into 24 blocks (8 blocks, each replicated 3 times), and then stored on the Hadoop cluster accordingly by the master node. Increasing and decreasing the block size can have very specific use-case implications; however, for the sake of this lab we will not cover those Hadoop specific questions, but rather how to change these default values.

Hadoop uses a standard block storage system to store the data across its data nodes. Since block size is slightly more of an advanced topic, we will not cover the specifics as to what and why the data is stored as blocks throughout the cluster.

The default block size value for IBM BigInsights 4.1 is currently set at 128MB (as opposed to the Hadoop default of 64MB in versions of Hadoop prior to Hadoop 2). If your specific use-case requires you to change this, it can be easily modified through Hadoop configuration files.

When making any Hadoop core changes, it is good practice (and a requirement for most) to restart any services that you changed. For the block size, remember to restart “HDFS” and “Console” services after you have made the changes.

__11. Move to the directory where Hadoop staging configuration files are stored. In this directory, you will see a file named “**hdfs-site.xml**”, one of the site-specific configuration files, which is on every host in your cluster. Edit the file with **vi**

```
cd /usr/iop/current/hadoop-client/conf
vi hdfs-site.xml
```

__12. Navigate to the property **dfs.blocksize**, and you will see the value is set to 128MB, the default block size for BigInsights. For the purpose of this lab, we will not change the value.

```
<!--Fri Jan  8 12:06:20 2016-->
    <configuration>

    <property>
      <name>dfs.block.access.token.enable</name>
      <value>true</value>
    </property>

    <property>
      <name>dfs.blockreport.initialDelay</name>
      <value>120</value>
    </property>

    <property>
      <name>dfs.blocksize</name>
      <value>134217728</value>
    </property>

    <property>
      <name>dfs.client.file-block-storage-locations.timeout.millis</name>
      <value>3000</value>
    </property>

    <property>
"hdfs-site.xml" [noeol] 330L, 7806C
```

3.3.4 Configuring the replication factor

__13. Navigate to the property dfs.replication. The current default replication factor will depend on the number of DataNodes that you have in your cluster and also the way that the cluster has been set up. If you only have one node, then realistically the value is 1. If you have two DataNodes, then it would make sense to have the value of 2. For three or more DataNodes, the standard / default value is 3. You can change the default value by setting an appropriate value as the default in the following lines to this file (**hdfs-site.xml**). The value will be the number of your choice. Use the **Page Up** and **Page Down** keys to navigate the file.

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>

<property>
  <name>dfs.replication.max</name>
  <value>50</value>
</property>

<property>
  <name>dfs.secondary.namenode.kerberos.https.principal</name>
  <value>HTTP/_HOST@EXAMPLE.COM</value>
</property>

<property>
  <name>dfs.secondary.namenode.kerberos.principal</name>
  <value>nn/_HOST@EXAMPLE.COM</value>
</property>
```

3.3.5 Limit DataNode disk usage

__14. Navigate to the property named *dfs.datanode.du.reserved*. This value represents reserved space in bytes per volume. HDFS will always leave this much space free for non-dfs use.

```
<property>
  <name>dfs.datanode.du.reserved</name>
  <value>1073741824</value>
</property>

<property>
  <name>dfs.datanode.failed.volumes.tolerated</name>
  <value>0</value>
  <final>true</final>
</property>

<property>
  <name>dfs.datanode.handler.count</name>
  <value>40</value>
</property>

<property>
  <name>dfs.datanode.hdfs-blocks-metadata.enabled</name>
  <value>true</value>
</property>

<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:50075</value>
```

For the purpose of this lab, we will not save this configuration change. This part of the Hands-On Lab is intended to just let you browse how to change some of the configuration values when you need it later on. However, in real-life, once you have made changes to this file, you would then synchronize this file across all appropriate nodes in the cluster.

NOTE: This configuration file is site-specific which means it only is affective for a node this file belongs to this cluster that can read the configuration files.

Configuration can also be done through the Ambari Web Console, but that is beyond the scope and time allotted for this Lab.

For more information on the configuration files:

- https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html
- Core: http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml
- HDFS: http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml
- MapRed: http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml
- YARN: http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-common/yarn-default.xml

3.4 Summary

Congratulations! You have now experience some common tasks of Hadoop administrations.

NOTES

NOTES