

Part 1: Launch AWS instances

Youtube: <https://www.youtube.com/watch?v=UeDSotLuWr0>

Step 1:

Login to aws console

Step 2:

Choose ubuntu machines

Step 3:

Make open security

Step 4:

Download the private key file and store it at a safe place as this is the only chance you would be able to download it. Amazon doesn't store it anywhere.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#retrieving-the-public-key-windows>

Step 5:

Launch 2 nodes and modify the names as datanode and namenode

Step 6:

Go to the aws console and make a note of public DNS for each node

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
<input checked="" type="checkbox"/>	Namenode	i-0e50906679f1d868c	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-188-99-110.us-west-2.compute.amazonaws.com
<input type="checkbox"/>	datanode1	i-0eb87b78a4b255d...	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-34-217-61-626.us-west-2.compute.amazonaws.com

Instance: **i-0e50906679f1d868c (Namenode)** Public DNS: **ec2-54-188-99-110.us-west-2.compute.amazonaws.com**

Description	Status Checks	Monitoring	Tags
Instance ID	i-0e50906679f1d868c	Public DNS (IPv4)	ec2-54-188-99-110.us-west-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	54.188.99.110
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-25-13.us-west-2.compute.internal
Availability zone	us-west-2b	Private IPs	172.31.25.13

For example

Namenode: ec2-54-188-99-999.us-west-2.compute.amazonaws.com

Datanode: ec2-34-217-61-926.us-west-2.compute.amazonaws.com

Part 2: Connect to AWS instances through Putty

Youtube: <https://www.youtube.com/watch?v=ic7Cl-kwpRA>

<https://www.youtube.com/watch?v=171nS9OxLzE>

Step 7:

Generate private key using Puttygen tool which can be understood by Putty

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

Step 8:

Open putty and start a connection with each node:

Use generated new ppk file in SSH->auth

Ip address is public DNS or public ip address of instance

Part 3: Install Java

Step 9: All nodes

Update repository

sudo apt-get update

Step 10: All nodes

Install Java

sudo apt-get -y install openjdk-8-jdk-headless

Part 4: Download Hadoop

Step 11: All nodes

sudo wget http://archive.apache.org/dist/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz

Step 12: All nodes

Extract files

tar xvzf hadoop-2.7.3.tar.gz

mv hadoop-2.7.3 hadoop

Part 5: Modify .bashrc file

Step 13: All nodes

Add following variables in ~/.bashrc file

```
#===== HADOOP VAR =====#
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_INSTALL=/home/ubuntu/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib/native"
```

Step 14: All nodes

Apply these changes

```
source ~/.bashrc
```

Part 6: Create HDFS storage dir

Step 15: All nodes

```
sudo mkdir -p /usr/local/hadoop/hdfs/data
```

```
sudo chown -R ubuntu:ubuntu /usr/local/hadoop/hdfs/data
```

Part 7: Modify configuration files

Step 16: All nodes

Go to the configuration directory

```
cd ~/hadoop/etc/hadoop
```

Step 17: All nodes

Set JAVA_HOME in hadoop-env.sh file

```
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Step 18: All nodes

Modify core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value><DNS of namenode>:9000</value>
  </property>
</configuration>
```

Step 19: Namenode

Modify hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///usr/local/hadoop/hdfs/data</value>
  </property>
</configuration>
```

Step 20: datanode

Modify hdfs-site.xml

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///usr/local/hadoop/hdfs/data</value>
</property>
</configuration>
```

Step 21: Namenode

Prepare mapred-site.xml

```
cp mapred-site.xml.template mapred-site.xml
```

Modify mapred-site.xml

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

Note: No need of mentioning job tracker address as its no longer used in MR2

https://www.cloudera.com/documentation/enterprise/5-4-x/topics/cdh_ig_mapreduce_to_yarn_migration.html

Step 22: Namenode

yarn-site.xml

```
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

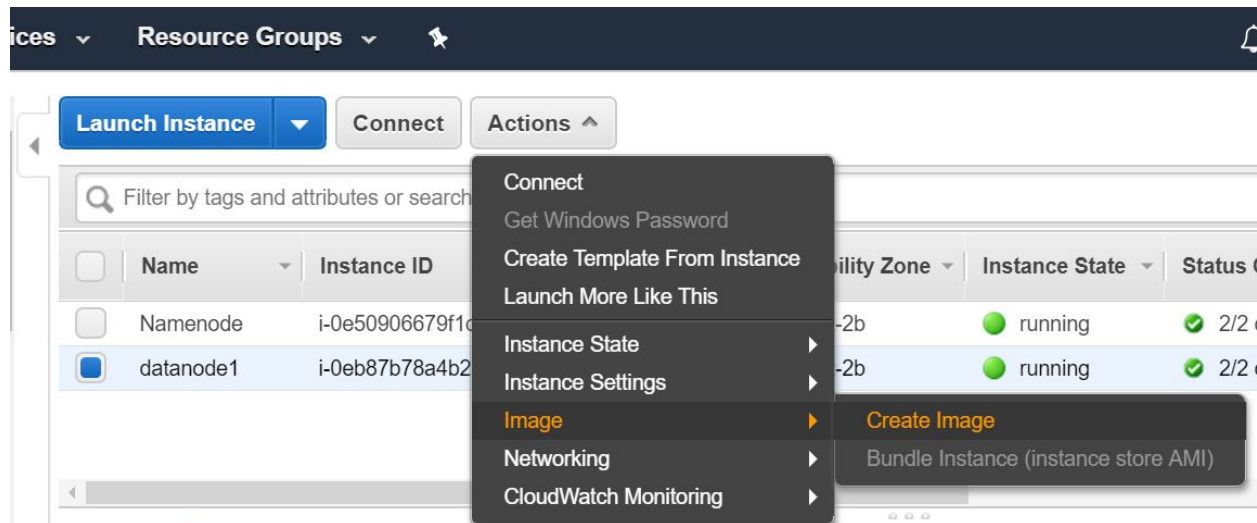
```

<property>
  <name>yarn.resourcemanager.hostname</name>
  <value><DNS of namenode></value>
</property>
</configuration>

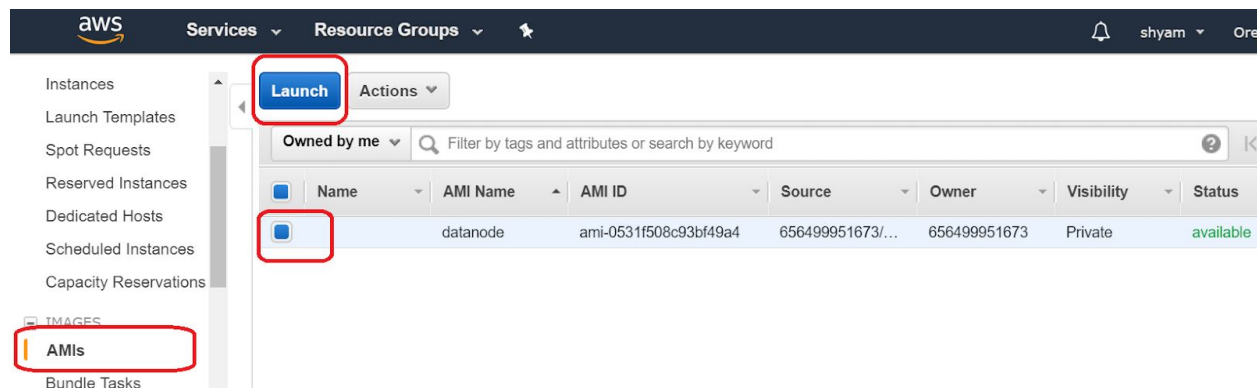
```

Part 8: create multiple instances of datanode

Create an Image of datanode



Launch multiple instances of this image from AMIs



Part 9: Configure SSH

Step 23: namenode

ssh-keygen

Step 24: namenode

Copy the content of file `/home/ubuntu/.ssh/id_rsa.pub` on clipboard

Step 25: all nodes (datanodes + namenode)

Paste that content in file `~/.ssh/authorized_keys`

Step 26: namenode

Configure node details in `~/.ssh/config` file.

Host namenode

HostName <DNS of namenode>

User ubuntu

IdentityFile ~/.ssh/id_rsa

Host datanode1

HostName <DNS of datanode1>

User ubuntu

IdentityFile ~/.ssh/id_rsa

Host datanode2

HostName <DNS of datanode2>

User ubuntu

IdentityFile ~/.ssh/id_rsa

Host datanode3

HostName <DNS of datanode3>

User ubuntu

IdentityFile ~/.ssh/id_rsa

Step 27: namenode

check SSH connectivity

`ssh namenode`

`ssh datanode1`

`ssh datanode2`

`ssh datanode3`

Part 10: set up masters and slaves

Step 28: Namenode

Modify files as following:

1. masters

<DNS of namenode>

2. slaves

<DNS of datanode1>

<DNS of datanode2>

<DNS of datanode3>

Part 11: Format namenode

Step 29: Namenode

hdfs namenode -format

Part 12: Start services

Step 30: Namenode

Start HDFS service

start-dfs.sh

Step 31: Namenode

Start YARN service

start-yarn.sh

=====

Notes

References:

<https://www.novixys.com/blog/setup-apache-hadoop-cluster-aws-ec2/>

<https://letsdobigdata.wordpress.com/2014/01/13/setting-up-hadoop-multi-node-cluster-on-amazon-ec2-part-1/>

Check public IPv4 on console:

```
curl v4.ident.me
```

Block location for a file in HDFS

```
hdfs fsck /myfile.txt -files -blocks -locations
```

Minimal Replicated Block:

<https://issues.apache.org/jira/browse/HDFS-8720>

Install Spark:

<https://data-flair.training/blogs/install-apache-spark-multi-node-cluster/>