BDA/CC Lab Internal-1

1. Installation, Configuration, and Running of Hadoop and HDFS.

Open Ubuntu Terminal and enter the following commands for Hadoop Installation, configuration and running HDFS files.

1. ***Install java jdk 8***
   sudo apt install openjdk-8-jdk

2. ***sudo nano .bashrc***
   ➔ open .bashrc file and paste these commands
   export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
   export PATH=$PATH:/usr/lib/jvm/java-8-openjdk-amd64/bin
   export HADOOP_HOME=~/hadoop-3.2.4/
   export PATH=$PATH:$HADOOP_HOME/bin
   export PATH=$PATH:$HADOOP_HOME/sbin
   export HADOOP_MAPRED_HOME=$HADOOP_HOME
   export YARN_HOME=$HADOOP_HOME
   export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
   export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
   export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
   export HADOOP_STREAMING=$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.2.4.jar
   export HADOOP_LOG_DIR=$HADOOP_HOME/logs
   export PDSH_RCMD_TYPE=ssh

3. ***sudo apt-get install ssh***

4. ***now go to*** **https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.2.4/hadoop-3.2.4.tar.gz** ***download the tar file***

5. ***Extract the tar file***
   tar -zxvf ~/Downloads/hadoop-3.2.4.tar.gz

6. ***Change directory to hadoop***
   cd hadoop-3.2.4/etc/Hadoop

7. ***set path for JAVA_HOME***
   sudo nano hadoop-env.sh
   JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

8. ***sudo nano core-site.xml***

```
<configuration>
 <property>
 <name>fs.defaultFS</name>
 <value>hdfs://localhost:9000</value>  </property>
 <property>
<name>hadoop.proxyuser.dataflair.groups</name> <value>*</value>
 </property>
 <property>
<name>hadoop.proxyuser.dataflair.hosts</name> <value>*</value>
 </property>
 <property>
<name>hadoop.proxyuser.server.hosts</name> <value>*</value>
 </property>
 <property>
<name>hadoop.proxyuser.server.groups</name> <value>*</value>
 </property>
</configuration>
```

9. ***sudo nano hdfs-site.xml***

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

10. ***sudo nano mapred-site.xml***

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

<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_
HOME/share/hadoop/mapreduce/lib/*</value>
 </property>
</configuration>
```

11. ***sudo nano yarn-site.xml***

```
<configuration>
 <property>
 <name>yarn.nodemanager.aux-services</name>
 <value>mapreduce_shuffle</value>
 </property>
 <property>
 <name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP
_CONF_DIR,CLASSPATH_PREP
END_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
 </property>
</configuration>
```

12. localhost commands

- ➔ ssh localhost
- ➔ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
- ➔ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
- ➔ chmod 0600 ~/.ssh/authorized_keys
- ➔ hadoop-3.2.4/bin/hdfs namenode -format

13. format the file system

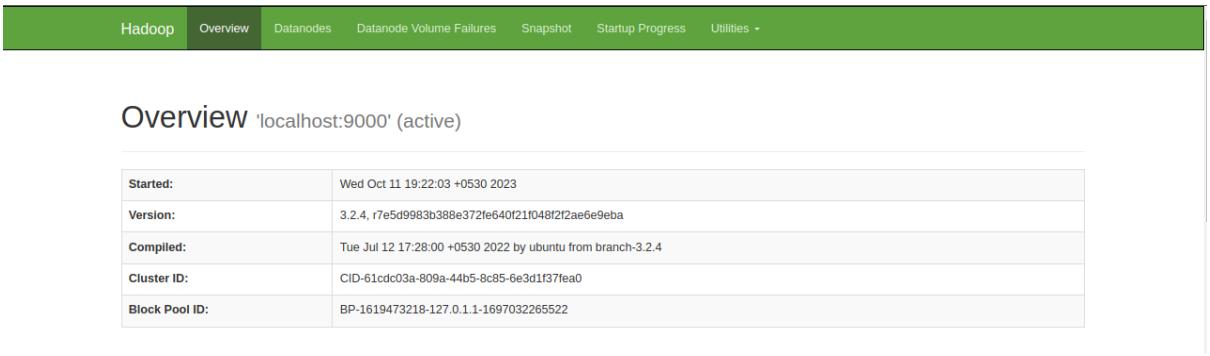
export PDSH_RCMD_TYPE=ssh

14. To start

start-all.sh

```
veeranna@veeranna-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as veeranna in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [veeranna-VirtualBox]
Starting resourcemanager
Starting nodemanagers
```

https://localhost:9870



15. To stop

Stop-all.sh

2. Implement the following file management tasks in Hadoop: Adding files and directories, retrieving files and Deleting files.

1. ***Create a Directory***
   hdfs dfs -mkdir -p tdata

2. ***Insert a file into the directory***
   hdfs dfs -put /home/veeranna/Downloads/input.txt tdata/

3. ***Copy the file from hadoop to local directory***
   hdfs dfs -get tdata/input.txt /home/veeranna/

4. ***Create empty file in hdfs***
   hdfs dfs -touchz tdata/test.txt

5. ***Read the content from the file***
   hdfs dfs -cat tdata/test.txt

6. ***Copy From Local and copy To Local***
   hdfs dfs -copyFromLocal /home/veeranna/demo.txt tdata/
   hdfs dfs -copyToLocal tdata/test.txt test.txt.hdfs

7. ***To set replication factor***
   hdfs dfs -setrep -w 5 tdata/test.txt

   Output → Replication 5 set: tdata/test.txt
   Waiting for tdata/test.txt ... done

8. ***To get replication factor***
   hdfs dfs -stat "%r" tdata/test.txt

   Output → 5

9. ***List of files of directory***
   hdfs dfs -ls

   Output → Found 1 items
   drwxr-xr-x - veeranna supergroup 0 2023-09-03 11:34 tdata

10. ***Copy the file content from one location to other***
    hdfs dfs -cp tdata/input.txt test

11. ***Move file from one place to another***
    hdfs dfs -mv tdata/demo.txt test

12. ***To delete a directory***
    hadoop fs -rm -r /user/veeranna/test

    Output → Deleted /user/veeranna/test

3. Create Virtual machines using Open-source software: VM Ware/ Oracle Virtual Box.

To create a virtual machine using Oracle VirtualBox, follow these steps:

1. ***Download and Install VirtualBox***

➔ Visit the Oracle VirtualBox website (https://www.virtualbox.org/) and download the latest version of VirtualBox for your operating system.

➔ Run the installer and follow the on-screen instructions to install VirtualBox.

2. ***Download an Operating System ISO***

➔ Obtain the ISO image of the operating system you want to install on the virtual machine. You can download Linux distributions or other OS ISOs from their respective official websites.

3. ***Open VirtualBox and create a new virtual machine***

➔ Launch Oracle VirtualBox after installation.

➔ Click on the "New" button in the VirtualBox Manager window to start creating a new virtual machine.

4. ***Name and Operating System***

➔ In the "Name and Operating System" window:

➔ Enter a name for your virtual machine.

➔ Select the type of operating system you are installing (e.g., Linux, Windows, macOS).

5. ***Memory (RAM) Allocation***

➔ Allocate memory (RAM) to your virtual machine. Choose an amount that suits your requirements but doesn't exceed the available physical RAM on your host system.

6. ***Hard Disk Creation and file type***

➔ Choose the option to "Create a virtual hard disk now" and click "Create."

➔ Select the file type for the virtual hard disk (usually VDI or VMDK) and click "Next."

7. ***File Location and Size***

➔ Specify the location where you want to store the virtual hard disk file and set the size of the disk. Ensure you allocate enough space for your OS and applications.

8. ***Create Virtual Machine***

➔ Review your settings in the summary window and click "Create" to create the virtual machine.

9. ***Attach ISO File***

➔ In the VirtualBox Manager, select your virtual machine.

➔ Click on "Settings" and go to the "Storage" section, Browse and select the OS ISO file you downloaded.

10. ***Start the Virtual Machine and Enjoy using it***

➔ click the "Start" button to power it on. Follow the on-screen instructions to install the operating system on your virtual machine.

➔ Once the OS is installed and configured, you can use your virtual machine just like a physical computer.

4. Use Amazon EC2 to create a Virtual machine.

Here are the steps to create and use an EC2 instance:

1. ***Sign in to the AWS Management Console***

➔ Go to the AWS Management Console (https://aws.amazon.com/).
➔ Sign in with your AWS account credentials.

2. ***Open the EC2 Dashboard and Launch EC2 Instance***

➔ Once signed in, select "Services" at the top left corner of the console.
➔ Under "Compute," select "EC2" to open the EC2 Dashboard.
➔ In the EC2 Dashboard, click the "Launch Instance" button to start the instance creation process.

3. ***Choose an Amazon Machine Image (AMI)***

➔ Select an AMI based on your requirements (e.g., Ubuntu, Windows Server).
➔ Choose the appropriate AMI for your use case, and click "Select."

4. ***Choose an Instance Type***

➔ Select the instance type that suits your workload. Instance types vary in terms of CPU, memory, and other resources.
➔ Click "Next: Configure Instance Details" when ready.
➔ Modify settings as needed and click "Next: Add Storage."

5. ***Add Storage***

➔ Specify the storage (EBS volumes) for your EC2 instance.
➔ Configure the size and type of the root volume and Click "Next: Add Tags" when done.

6. ***Configure Security Group***

➔ Create a new security group or select an existing one.
➔ Configure inbound and outbound rules to control traffic to and from your instance.

7. ***Review and launch***

➔ Review all the settings you've configured for your EC2 instance. Click “Launch”

8. ***create a Key Pair***

➔ If you haven't created an EC2 key pair before, you'll be prompted to create one.
➔ Download the private key (.pem) file and store it in a secure location. You'll need this key to access your instance securely.

9. ***Launch the Instance and Connect to EC2 Instance***

➔ After creating or selecting a key pair, click the "Launch Instances" button.
➔ Once the instance is running, you can connect to it using SSH (for Linux instances) or RDP (for Windows instances) with the private key.

10. ***Start Using Your EC2 Instance***

➔ You can now use your EC2 instance for various tasks, such as hosting a website, running applications, or performing data analysis.

5. Use Amazon S3 to create bucket and upload objects.

To use Amazon Simple Storage Service (Amazon S3) to create a bucket and upload objects, follow these steps:

1. ***Sign in to AWS and open S3 Dashboard***

➔ Sign in to the AWS Management Console using your AWS account credentials.
➔ From the AWS Management Console, navigate to the S3 dashboard.

2. ***Create a Bucket***

➔ Enter a globally unique name for your bucket (S3 bucket names must be unique across all AWS accounts).
➔ Choose the AWS region where you want to create the bucket.
➔ Configure optional settings like versioning, logging, and tags.
➔ Click "Create" to create the bucket.

3. ***Upload Objects***

➔ In the bucket you just created, click the "Upload" button to upload objects.
➔ Click "Add files" or "Add folder" to select the files or folders you want to upload.
➔ Configure settings such as permissions and metadata for the uploaded objects.
➔ Click "Upload" to start the upload process.

4. ***Manage Objects***

➔ After uploading objects, you can manage them within the S3 bucket.
➔ You can set permissions, configure lifecycle policies, and organize objects into folders (known as "prefixes" in S3).

5. ***Access Objects***

➔ To access objects in your S3 bucket, click on the object's name in the S3 dashboard.
➔ You'll see a URL that you can use to access the object via a web browser or programmatically through APIs.

6. ***Set Bucket Policies***

➔ If you want to control access to your bucket and objects further, you can configure bucket policies.
➔ Bucket policies allow you to define fine-grained permissions for different users or applications.

7. ***Configure Cross-Region Replication***

➔ If you need to replicate your data to another AWS region for redundancy or compliance purposes, you can set up cross-region replication.

8. ***Monitor and Manage***

➔ Regularly monitor your S3 bucket for usage, billing, and access patterns.
➔ Use AWS CloudWatch and other AWS services for monitoring and alerting.