Exp No: 1 **Date: 18.08.2022**

CREATION AND MANAGEMENT OF VIRTUAL MACHINE IN VIRTUALIZED ENVIRONMENT – VMWARE WORKSTATION

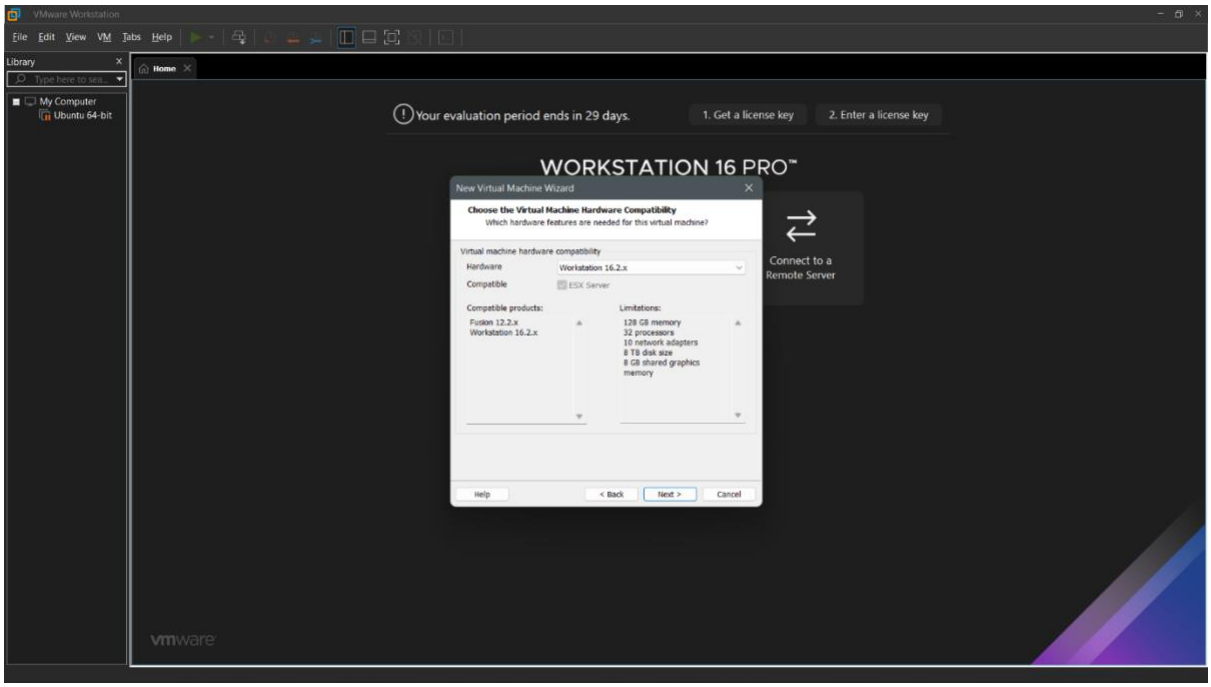
AIM:

To create and run a virtual machine in system using VmWare Workstation Pro.

PROCEDURE:

1. Launch a VM ware
2. Create new virtual machine
3. Customize the set-up
4. Set username and password
5. Browse for .iso file of an operating system
6. Configure the hardware capacity
7. Finish and power on the VM

OUTPUT:



RESULT:

The installation, configuration and running a python program in Ubuntu in VMware has been executed successfully.

Exp No: 2 **Date: 25.08.2022**

VIRTUALIZE A MACHINE AND CHECK HOW MANY MACHINES CAN BE UTILIZED AT A PARTICULAR TIME

AIM

To virtualize a machine and check how many machines can be utilized at a particular time.

PROCEDURE

1. Open the VMware.
2. Run the virtual machines installed in the VMware.
3. The virtual machines run parallelly in the VMware.
4. We can observe that many virtual machines can be run at a particular time.
5. The number depends on system configuration.

OUTPUT:

RESULT:

Thus, configuring and cloning and taking snapshots in VMware has been executed successfully.

Exp No: 3 **Date: 01.09.2022**

CREATE A VM CLONE AND ATTACH A VIRTUAL BLOCK TO THE CLONED VM

AIM:

To find procedure to attach a virtual block to virtual machine and check whether it holds data even after the release of the virtual machine.

a) Create a VM clone

b) Adding an additional block to Hard-Disk in Virtual Machine.

PROCEDURE:

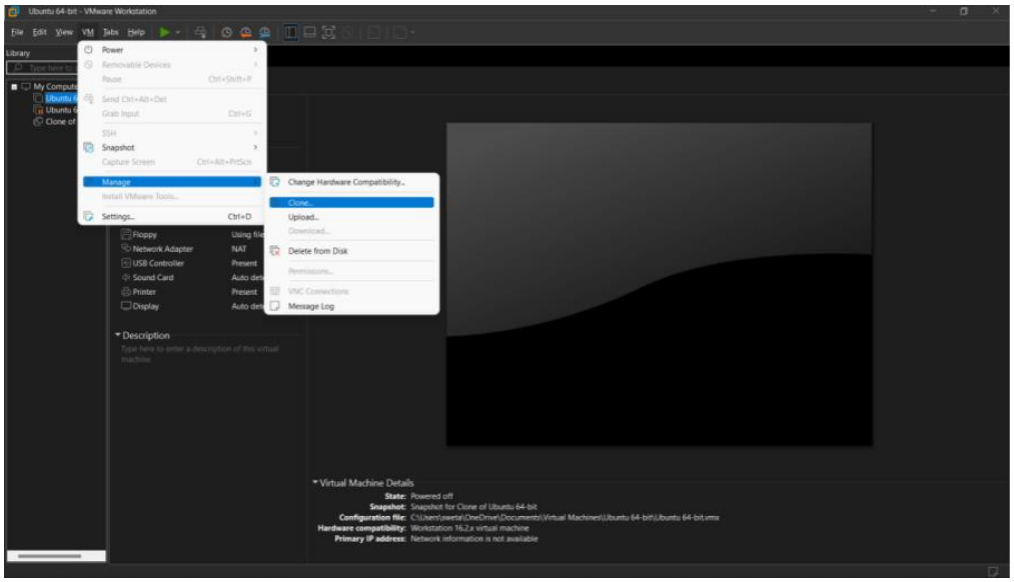
a) Create a VM clone

Click VM -> Manage -> Clone

b) Create a Virtual Block

1. Open the virtual machine settings editor (VM > Settings) and click Add
2. Click Hard Disk, then click Next.
3. Select Create a new virtual disk, then click Next.
4. Choose whether you want the virtual disk to be an IDE disk or SCSI disk.
5. Set the capacity for the new virtual disk.
6. Accept the default filename and location for the virtual disk file
7. The wizard creates the new virtual disk. It appears to your guest operating system as a new, blank hard disk. Use the guest operating system's tools to partition and format the new drive for use.

OUTPUT:



RESULT

Thus the procedure to create a VM clone and adding an additional block to Hard-Disk in virtual machine is completed successfully

EX. NO: 4 **DATE: 15.09.2022**

INSTALL, COMPILE AND RUN GCC COMPILER ON THE VIRTUAL MACHINE

AIM

To install and run a C Compiler in Ubuntu.

PROCEDURE

Step 1 – Open Console
Step 2 – Install gcc
a) gcc –v
b) sudo apt-get install gcc
c) sudo apt-get install build-essential C compiler will be installed
Step 3 - To Install Vim Editor
a) sudo apt-get update
b) sudo apt-get install vim
Step 4–Type the C program in terminal
hello.c

```
#include<stdio.h>
int main()
{
printf ("Hello");
return 0;
}
```

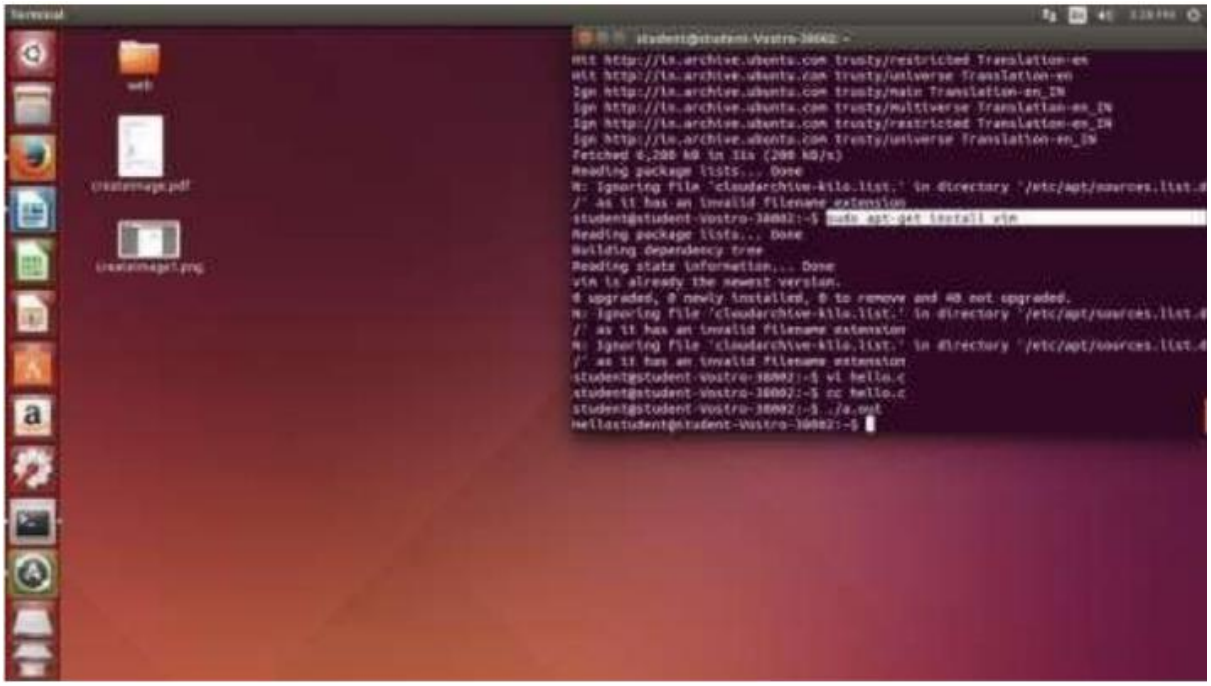
Step 5 – Compile and Execute C Program

```
cc hello.c
./a.out
```

OUTPUT


web
createimage.pdf
createimage1.png
student@student-Vostro-3800:~
[sudo] password for student:
Sorry, try again.
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
student@student-Vostro-3800:~$


Terminal
web
createimage.pdf
createimage1.png
student@student-Vostro-3800:~
student@student-Vostro-3800:~$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
dpkg-dev fakeroot g++ g++-4.8 libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libfakeroot
libstdc++-4.8-dev
Suggested packages:
debian-keyring g++-multilib g++-4.8-multilib gcc-4.8-doc libstdc++6-4.8-dbg
libstdc++-4.8-doc
The following NEW packages will be installed:
build-essential dpkg-dev fakeroot g++ g++-4.8 libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libfakeroot
libstdc++-4.8-dev
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 20.1 MB of archives.
After this operation, 42.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y


web
createimage.pdf
createimage1.png
student@student-Vostro-3800:~
Hit http://in.archive.ubuntu.com trusty/restricted Translation-en
Hit http://in.archive.ubuntu.com trusty/universe Translation-en
Ign http://in.archive.ubuntu.com trusty/main Translation-en_IN
Ign http://in.archive.ubuntu.com trusty/multiverse Translation-en_IN
Ign http://in.archive.ubuntu.com trusty/restricted Translation-en_IN
Ign http://in.archive.ubuntu.com trusty/universe Translation-en_IN
Fetched 0,288 kB in 31s (288 kB/s)
Reading package lists... Done
W: Ignoring file 'cloudarchive-kilo.list.' in directory '/etc/apt/sources.list.d
/' as it has an invalid filename extension
student@student-Vostro-3800:~$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
vim is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
W: Ignoring file 'cloudarchive-kilo.list.' in directory '/etc/apt/sources.list.d
/' as it has an invalid filename extension
W: Ignoring file 'cloudarchive-kilo.list.' in directory '/etc/apt/sources.list.d
/' as it has an invalid filename extension
student@student-Vostro-3800:~$ vi hello.c
student@student-Vostro-3800:~$ cc hello.c
student@student-Vostro-3800:~$ ./a.out
hellostudent@student-Vostro-3800:~$


web
createimage.pdf
createimage1.png
student@student-Vostro-3800:~
Hit http://in.archive.ubuntu.com trusty/restricted Translation-en
Hit http://in.archive.ubuntu.com trusty/universe Translation-en
Ign http://in.archive.ubuntu.com trusty/main Translation-en_IN
Ign http://in.archive.ubuntu.com trusty/multiverse Translation-en_IN
Ign http://in.archive.ubuntu.com trusty/restricted Translation-en_IN
Ign http://in.archive.ubuntu.com trusty/universe Translation-en_IN
Fetched 0,288 kB in 31s (288 kB/s)
Reading package lists... Done
W: Ignoring file 'cloudarchive-kilo.list.' in directory '/etc/apt/sources.list.d
/' as it has an invalid filename extension
student@student-Vostro-3800:~$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
vim is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
W: Ignoring file 'cloudarchive-kilo.list.' in directory '/etc/apt/sources.list.d
/' as it has an invalid filename extension
W: Ignoring file 'cloudarchive-kilo.list.' in directory '/etc/apt/sources.list.d
/' as it has an invalid filename extension
student@student-Vostro-3800:~$ vi hello.c
student@student-Vostro-3800:~$ cc hello.c
student@student-Vostro-3800:~$ ./a.out
hellostudent@student-Vostro-3800:~$

EX. NO: 5 **DATE: 22.09.2022**

DEVELOP A SIMPLE EMAIL AUTOMATION SERVICE USING SALESFORCE

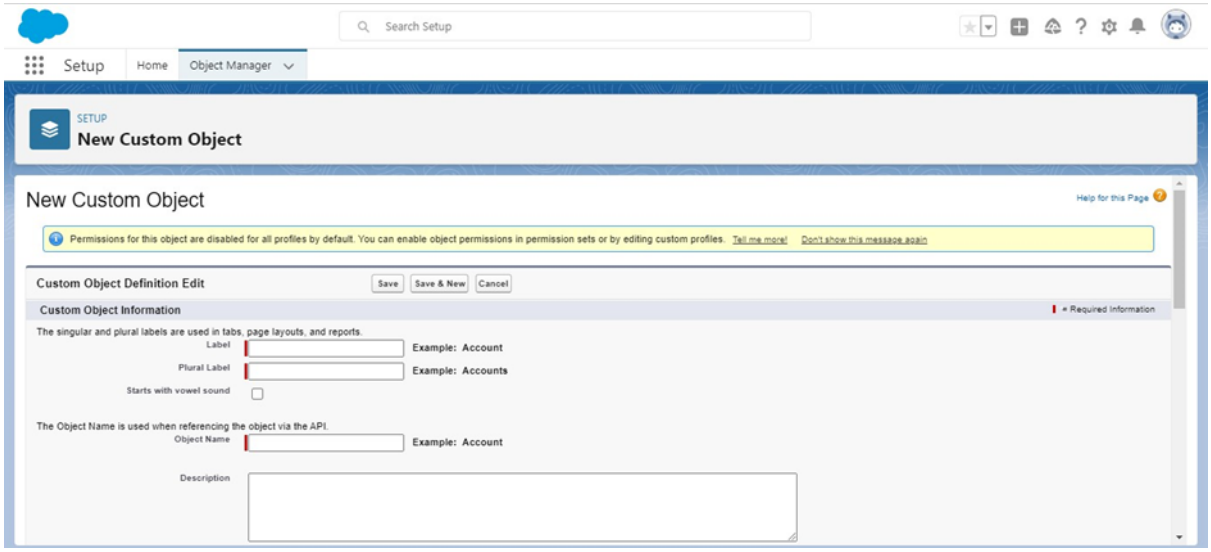
AIM:

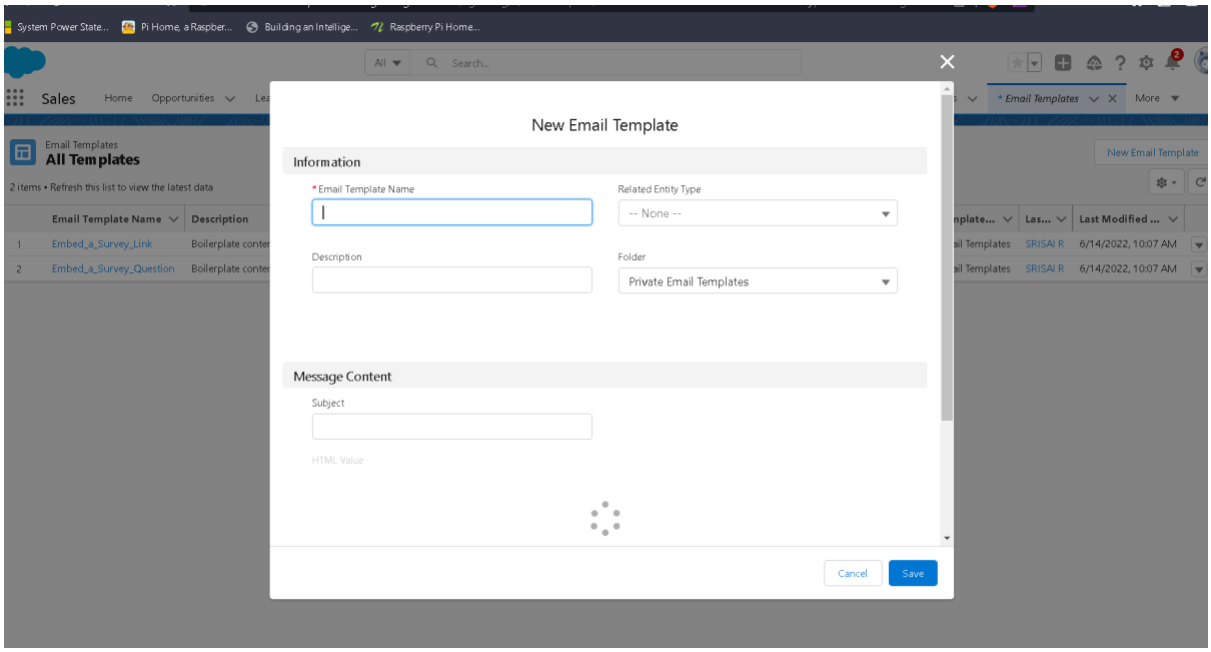
Develop a simple email automation service using Salesforce

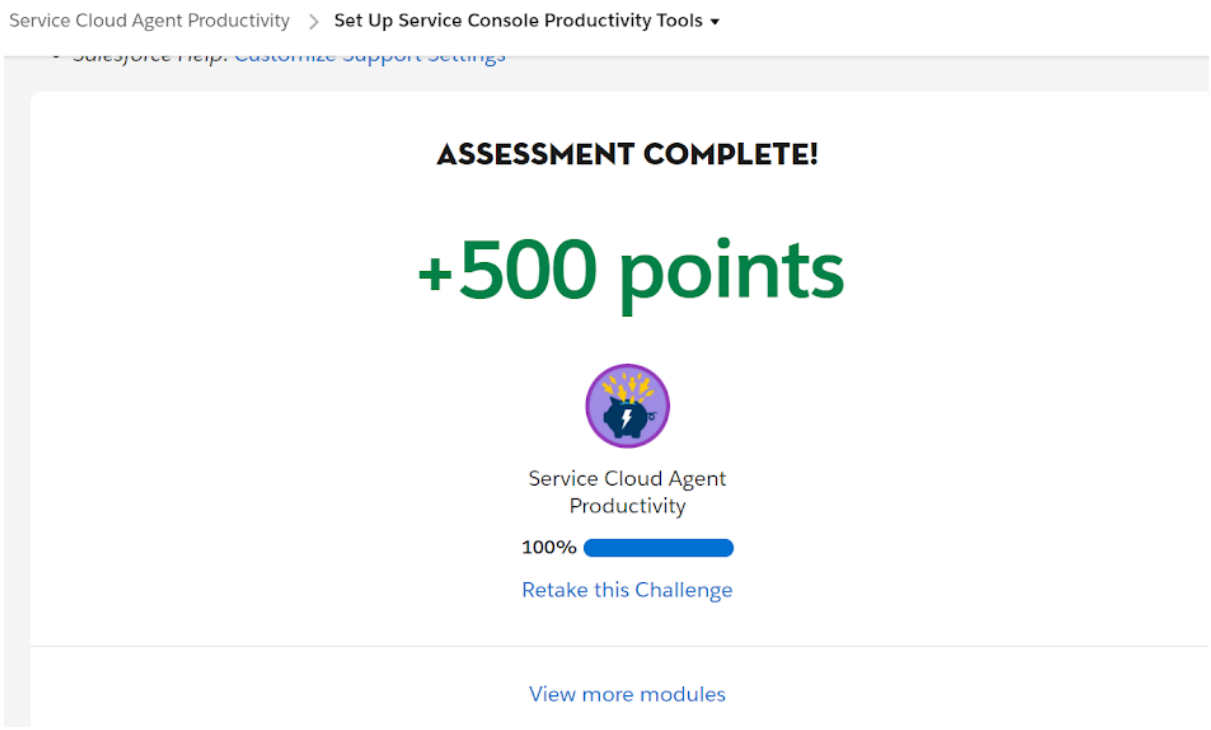
PROCEDURE:

1) Open Salesforce and login to the account.
2) Open the Service cloud agent productivity module
3) Do the setup service console productivity tools
a. Give users access to quick text and macros.
b. Customize your app for quick text, macros, and history.
c. Set up mass quick actions.
d. Customize an email action.
e. Enable email notifications for case owners
4) Open the Create Macros and Quick Text to Reduce Clicks
a. Create macros.
b. Create quick text.
5) Open the module Use All the Service Console Productivity Tools Together
a. Use split view.
b. Run a macro.
c. Use and find the keyboard shortcuts.
d. Insert quick text.
e. Perform mass quick actions.
f. Use the History utility.

OUTPUT:


Search Setup
Setup
Home
Object Manager
SETUP
New Custom Object
New Custom Object
Help for this Page
Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. Tell me more! Don't show this message again
Custom Object Definition Edit
Save
Save & New
Cancel
Custom Object Information
= Required Information
The singular and plural labels are used in tabs, page layouts, and reports.
Label
Example: Account
Plural Label
Example: Accounts
Starts with vowel sound
The Object Name is used when referencing the object via the API.
Object Name
Example: Account
Description


Sales
Home
Opportunities
Email Templates
All Templates
2 items • Refresh this list to view the latest data
Email Template Name
Description
New Email Template
Information
* Email Template Name
Related Entity Type
-- None --
Description
Folder
Private Email Templates
Message Content
Subject
HTML Value
Cancel
Save



RESULT:

Thus an instance and custom object has been created in salesforce successfully.

EX. NO: 6 **DATE: 26.09.2022**

LAUNCH A CLOUD INSTANCE USING IBM CLOUD

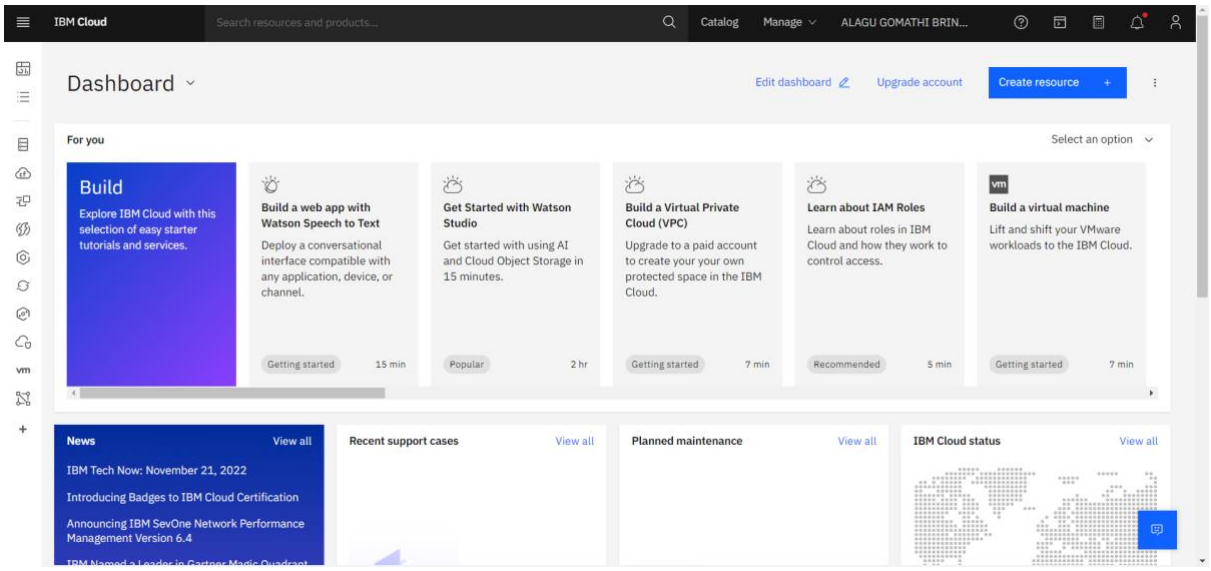
AIM:

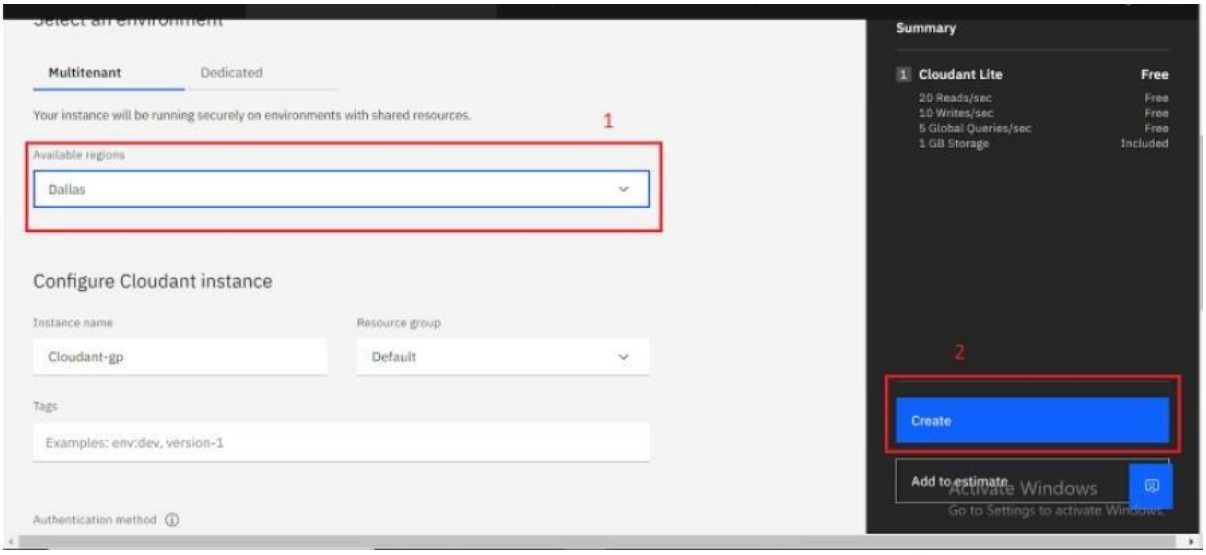
Develop a simple application to understand the concept of PAAS using IBM CLOUD

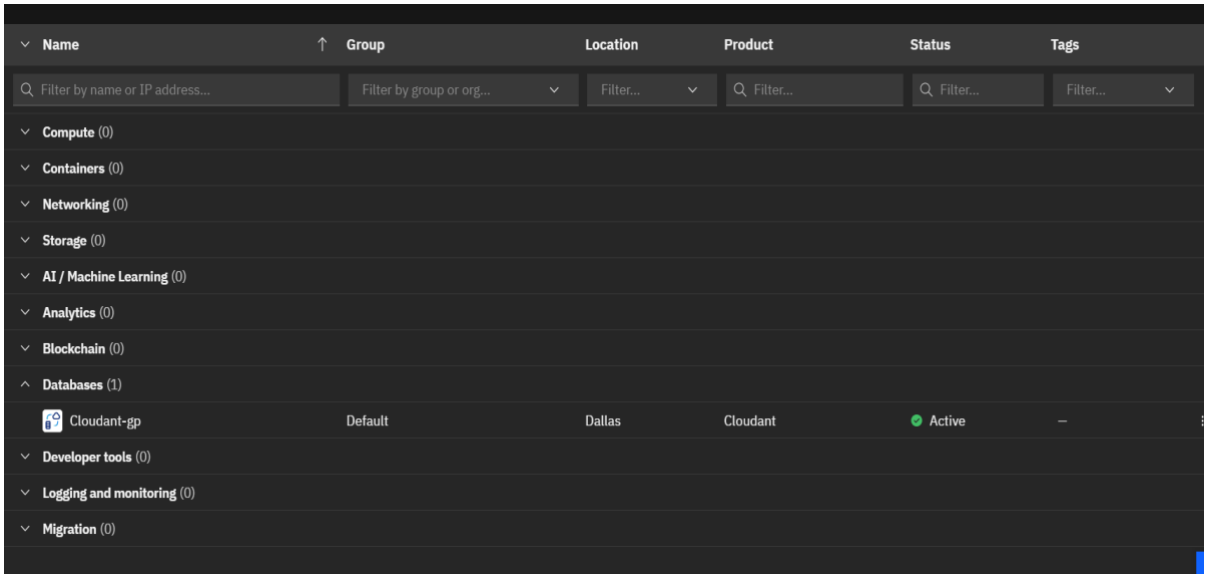
PROCEDURE:

1. Register to IBM Cloud
2. Sign in with your credentials
3. Log in to your IBM Cloud account, and click on Catalog
4. Type Cloudant in the Search bar and click to open it.
5. Select an offering and an environment
6. Select region as Dallas & Type an instance name then click on create service.
7. After you click create the system displays a message to say that the instance is being provisioned, which returns you to the Resource list. From the Resource list, you see that thestatus for your instance is, Provision in progress.
8. When the status changes to Active, click the instance.

OUTPUT:


IBM Cloud
Search resources and products...
Catalog
Manage
ALAGU GOMATHI BRIN...
Dashboard
Edit dashboard
Upgrade account
Create resource
For you
Select an option
Build
Explore IBM Cloud with this selection of easy starter tutorials and services.
Build a web app with Watson Speech to Text
Deploy a conversational interface compatible with any application, device, or channel.
Getting started
15 min
Get Started with Watson Studio
Get started with using AI and Cloud Object Storage in 15 minutes.
Popular
2 hr
Build a Virtual Private Cloud (VPC)
Upgrade to a paid account to create your your own protected space in the IBM Cloud.
Getting started
7 min
Learn about IAM Roles
Learn about roles in IBM Cloud and how they work to control access.
Recommended
5 min
Build a virtual machine
Lift and shift your VMware workloads to the IBM Cloud.
Getting started
7 min
News
View all
IBM Tech Now: November 21, 2022
Introducing Badges to IBM Cloud Certification
Announcing IBM SevOne Network Performance Management Version 6.4
Recent support cases
View all
Planned maintenance
View all
IBM Cloud status
View all


Multitenant
Dedicated
Your instance will be running securely on environments with shared resources.
1
Available regions
Dallas
Configure Cloudant instance
Instance name
Cloudant-gp
Resource group
Default
Tags
Examples: env:dev, version-1
Authentication method
Summary
1 Cloudant Lite
Free
20 Reads/sec
Free
10 Writes/sec
Free
5 Global Queries/sec
Free
1 GB Storage
Included
2
Create
Add to estimate
Activate Windows
Go to Settings to activate Windows.



RESULT

Thus, the launching of a cloud instance using a public IaaS cloud service like the IBM cloud has been done successfully.

EX. NO: 7 **DATE: 06.10.2022**

PUBLIC CLOUD SERVICE

AIM:

Develop a simple application to understand the concept of PAAS using Microsoft Azure

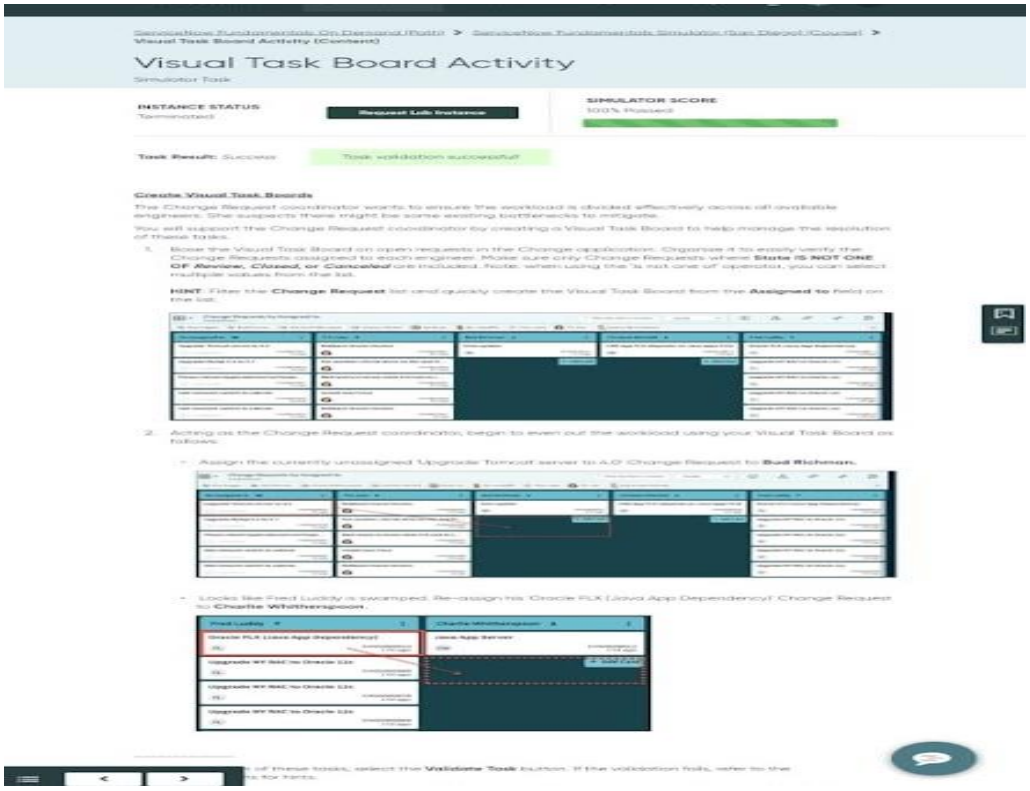
PROCEDURE:

1. Sign up in the ServiceNow portal

2. Enroll for course : ServiceNow Fundamentals On Demand .

3. Select visual task board activity under ServiceNow fundamentals simulator (San Diego)

OUTPUT:



RESULT:

Thus an application instance has been created in Microsoft Azure successfully.

Exp No: 8 **Date: 13.10.2022**

MODEL CLOUD ENVIRONMENT USING CLOUD SIM

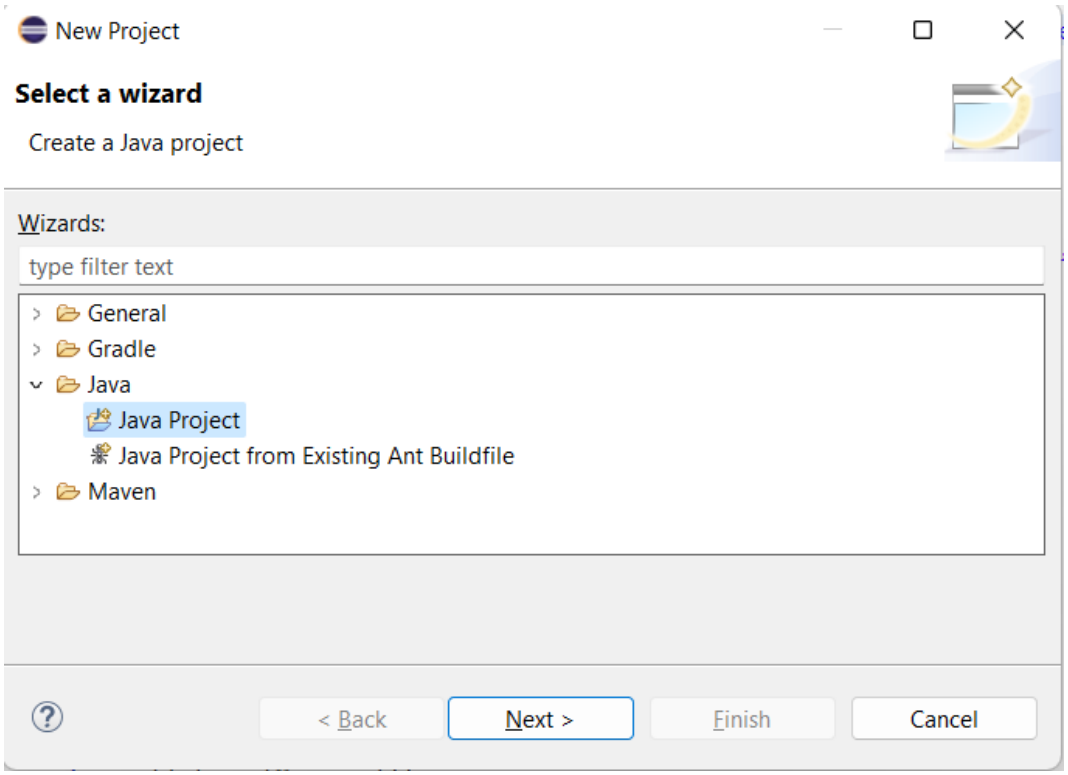
AIM:

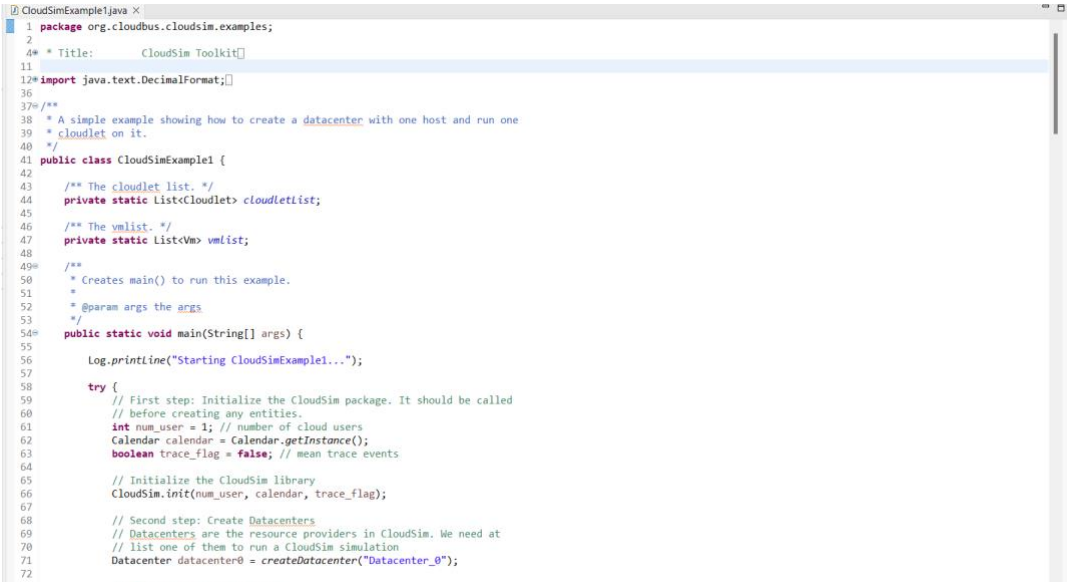
Installation of CloudSim into Eclipse to model a cloud computing environment and analyze the VM provisioning using CloudSim.

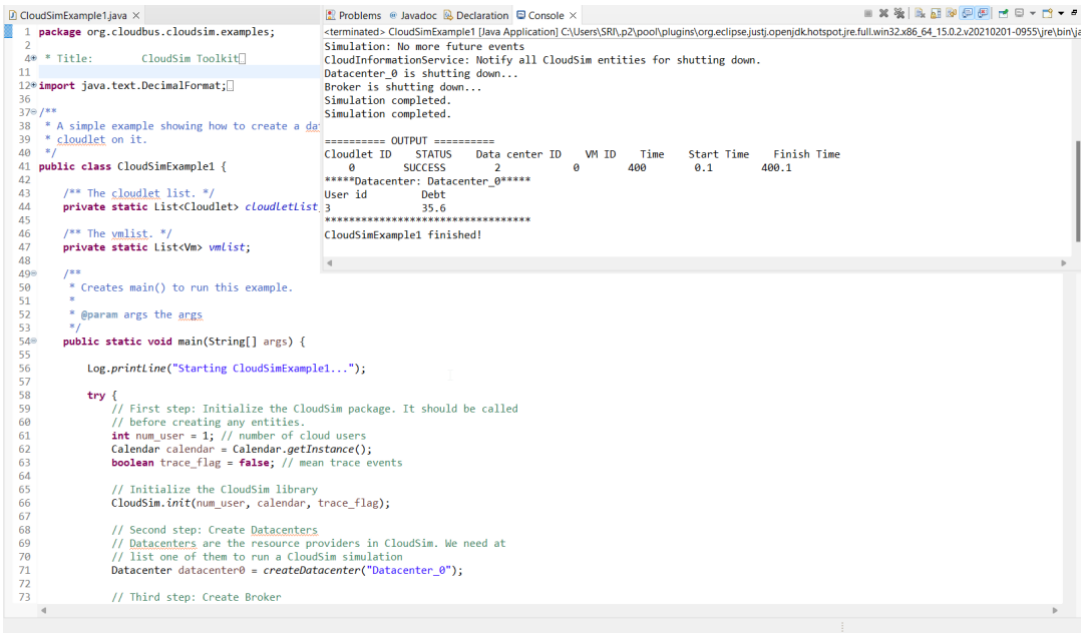
PROCEDURE:

1) Open up Eclipse and go to Menu Section, then click File, keep on clicking New and finally select java project.
2) A JRE environment is required and hence make sure to have jdk installed and finally give a name to the project.
3) CloudSim package is required to be downloaded and extract the CloudSim zip file to set the directory for the project.
4) Common Math from apache is a jar file which is also required to run the math functions in the CloudSim file.
5) Now all the files are installed in the Eclipse environment.
6) Run a program in Cloudsim environment to check if the simulation is complete.

OUTPUT:


New Project
Select a wizard
Create a Java project
Wizards:
type filter text
General
Gradle
Java
Java Project
Java Project from Existing Ant Buildfile
Maven
< Back
Next >
Finish
Cancel


CloudSimExample1.java
package org.cloudbus.cloudsim.examples;
* Title: CloudSim Toolkit
import java.text.DecimalFormat;
/**
* A simple example showing how to create a datacenter with one host and run one
* cloudlet on it.
*/
public class CloudSimExample1 {
/** The cloudlet list. */
private static List<Cloudlet> cloudletList;
/** The vmlist. */
private static List<Vm> vmlist;
/**
* Creates main() to run this example.
*
* @param args the args
*/
public static void main(String[] args) {
Log.printLine("Starting CloudSimExample1...");
try {
// First step: Initialize the CloudSim package. It should be called
// before creating any entities.
int num_user = 1; // number of cloud users
Calendar calendar = Calendar.getInstance();
boolean trace_flag = false; // mean trace events
// Initialize the CloudSim library
CloudSim.init(num_user, calendar, trace_flag);
// Second step: Create Datacenters
// Datacenters are the resource providers in CloudSim. We need at
// list one of them to run a CloudSim simulation
Datacenter datacenter0 = createDatacenter("Datacenter_0");
// Third step: Create Broker




```java
package org.cloudbus.cloudsim.examples;

/*
 * Title:        CloudSim Toolkit
 */
import java.text.DecimalFormat;

/**
 * A simple example showing how to create a da
 * cloudlet on it.
 */
public class CloudSimExample1 {

    /** The cloudlet list. */
    private static List<Cloudlet> cloudletList

    /** The vmlist. */
    private static List<Vm> vmlist;

    /**
     * Creates main() to run this example.
     *
     * @param args the args
     */
    public static void main(String[] args) {

        Log.printLine("Starting CloudSimExample1...");

        try {
            // First step: Initialize the CloudSim package. It should be called
            // before creating any entities.
            int num_user = 1; // number of cloud users
            Calendar calendar = Calendar.getInstance();
            boolean trace_flag = false; // mean trace events

            // Initialize the CloudSim library
            CloudSim.init(num_user, calendar, trace_flag);

            // Second step: Create Datacenters
            // Datacenters are the resource providers in CloudSim. We need at
            // list one of them to run a CloudSim simulation
            Datacenter datacenter0 = createDatacenter("Datacenter_0");

            // Third step: Create Broker
```

RESULT:

Thus, the cloud environment was modeled using the cloud sim tool and eclipse.

Exp No: 9 **Date: 20.10.2022**

IMPLEMENT ROUND ROBIN TASK SCHEDULING IN BOTH TIMESHARED AND SPACE SHARED CPU ASSIGNMENT

AIM:

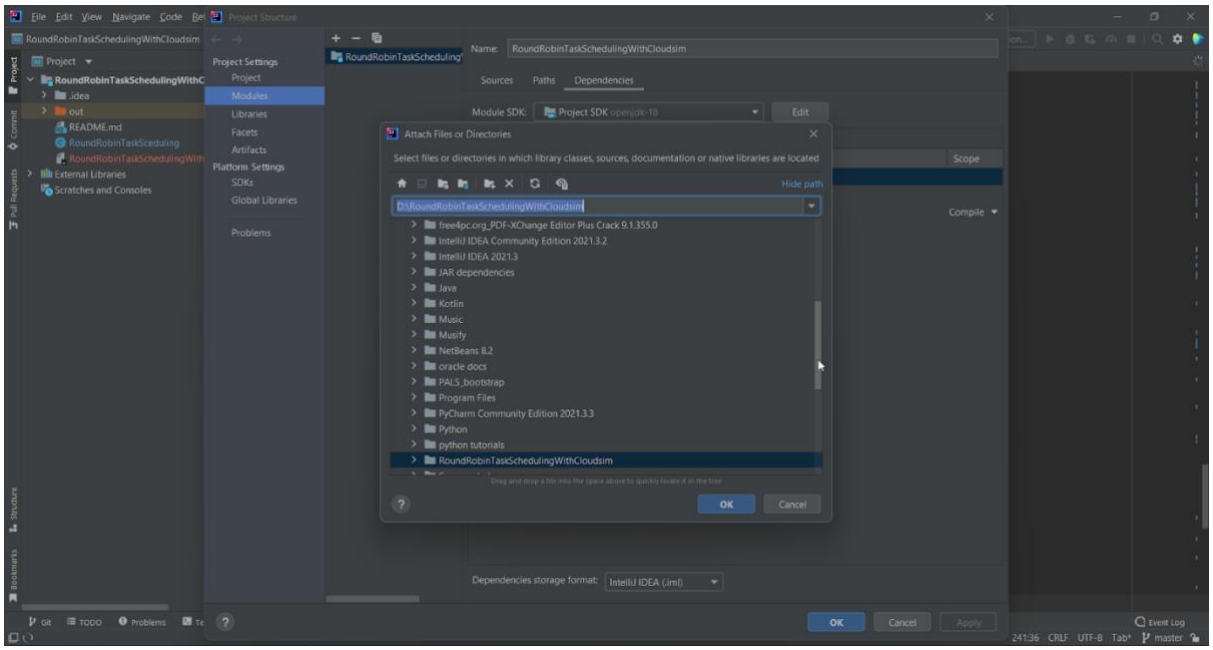
Implement RoundRobin task scheduling in both TimeShared and SpaceShared CPU assignments.

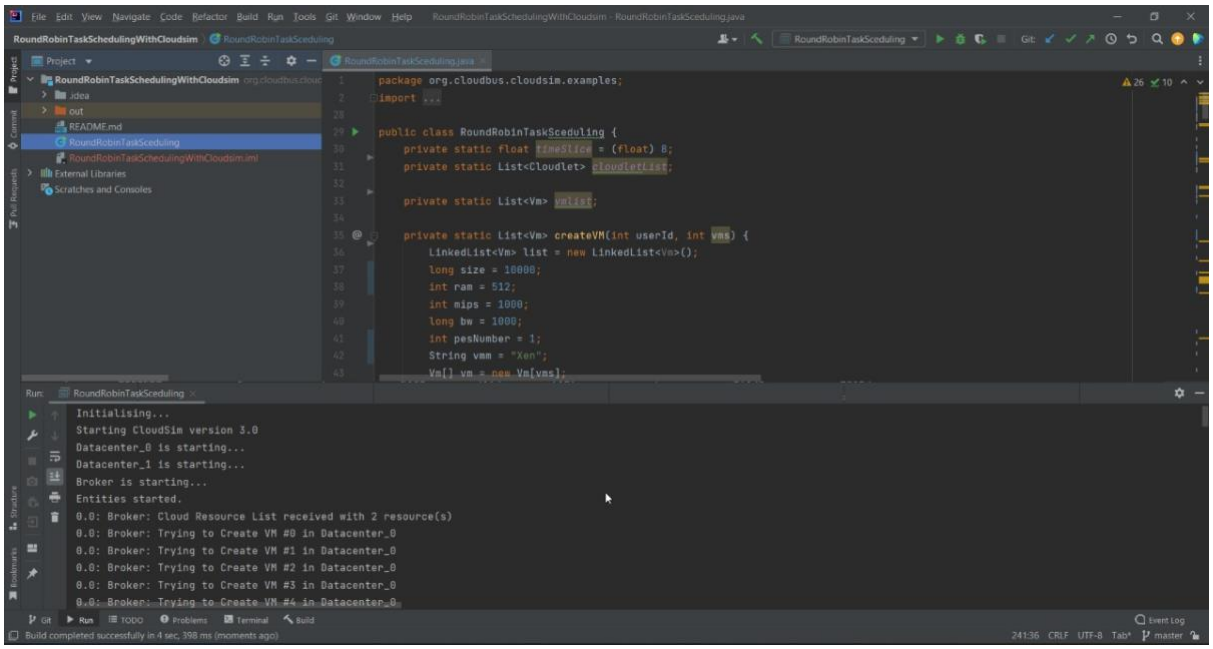
PROCEDURE:

1) Initialize the CloudSim library and datacenters.
2) Create 2 VM's and the second VM will have twice the priority of VM1 and so will receive twice CPU time. Add the VMs to the vmList.
3) Create two Cloudlets. Where,

Cloudlet properties
intid = 0;
longlength = 40000;
longfileSize = 300;
longoutputSize = 300;

4) Add the cloudlets to a list and submit cloudlet list to the broker.
5) Bind the cloudlets to the vms. This way, the broker will submit the bound cloudlets only to the specific VM.
6) Create a datacenter. The steps needed to create a PowerDatacenter,
(i) We need to create a list to store our machine
(ii) Machine contains one or more PEs or CPUs/Cores.
(iii) Create PEs and add these into a list.
(iv) Create Hosts with its id and list of PEs and add them to the list of machines
(v) create another machine in the Data center.
(vi) Create a DatacenterCharacteristics object and create a PowerDatacenter object.
7) Print the Cloudlet objects and observe the result





RESULT:

Thus, RoundRobin task scheduling in both TimeShared and SpaceShared CPU assignment is implemented successfully.

Exp No: 10 **Date: 27.10.2022**

SET UP A SINGLE HADOOP CLUSTER AND SHOW THE PROCESS USING WEB UI

AIM:

To set-up one node Hadoop cluster.

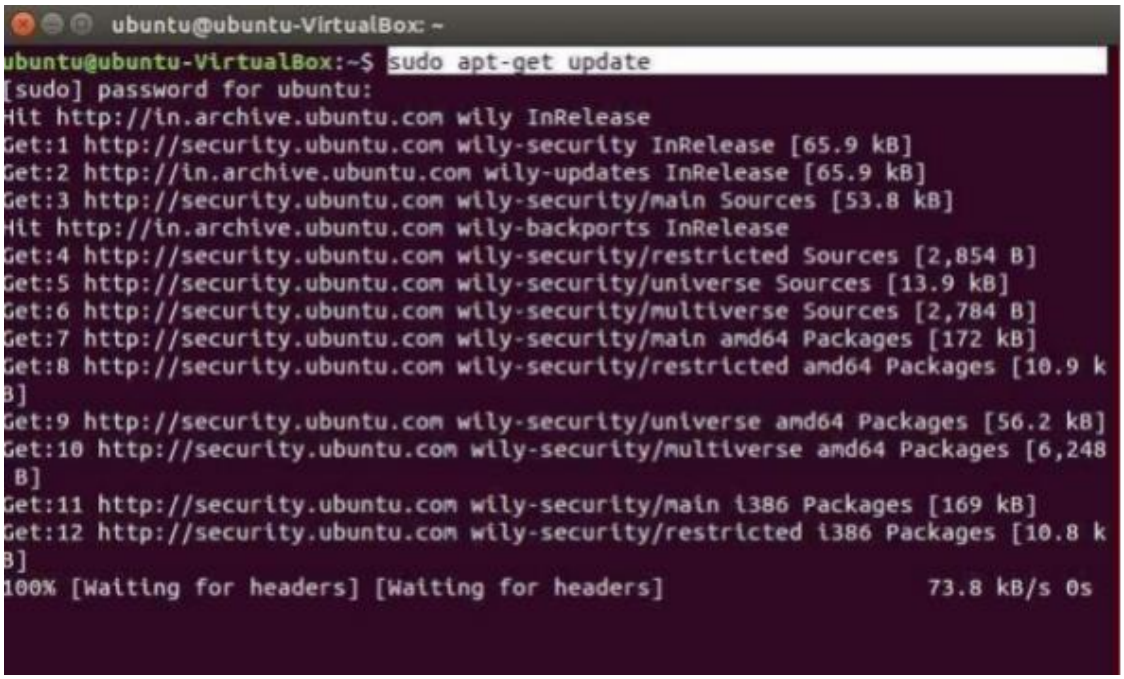
PROCEDURE:

1. System Update
2. Install Java
3. Add a dedicated Hadoop user
4. Install SSH and setup SSH certificates
5. Check if SSH works
6. Install Hadoop
7. Modify Hadoop config files
8. Format Hadoop filesystem
9. Start Hadoop
10. Check Hadoop through web UI
11. Stop Hadoop

PROCEDURE

Step 1 – System Update

$ sudo apt-get update



Step 2 – Install Java and Set JAVA_HOME

//This first thing to do is to setup the webupd8 ppa on your system. Run the following command and proceed.

$ sudo apt-add-repository ppa:webupd8team/java

$ sudo apt-get update

//After setting up the ppa repository, update the package cache as well.

//Install the Java 8 installer

$ sudo apt-get install oracle-java8-installer

// After the installation is finished, Oracle Java is setup. Run the java command again to check the version and vendor.

```
buntu@ubuntu-VirtualBox:~$ sudo apt-get install oracle-java8-installer
eading package lists... Done
uilding dependency tree
eading state information... Done
he following packages were automatically installed and are no longer required:
 libntdb1 python-ntdb
se 'apt-get autoremove' to remove them.
he following extra packages will be installed:
 gsfonts-x11 java-common
uggested packages:
 default-jre equivs binfmt-support visualvm ttf-baekmuk ttf-unfonts
 ttf-unfonts-core ttf-kochi-gothic ttf-sazanami-gothic ttf-kochi-mincho
 ttf-sazanami-mincho ttf-arphic-uming
he following NEW packages will be installed:
 gsfonts-x11 java-common oracle-java8-installer
 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
eed to get 163 kB of archives.
fter this operation, 511 kB of additional disk space will be used.
 you want to continue? [Y/n] y
et:1 http://ppa.launchpad.net/webupd8team/java/ubuntu/ wily/main oracle-java8-i
staller all 8u101+8u101arm-1~webupd8~2 [23.6 kB]
```

Step 3 – Add a dedicated Hadoop user

$ sudo addgroup hadoop

```
ubuntu@ubuntu-VirtualBox:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
```

$ sudo adduser --ingroup hadoop hduser

```
ubuntu@ubuntu-VirtualBox:~$ sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

// Add hduser to sudo user group

$ sudo adduser hduser sudo

```
ubuntu@ubuntu-VirtualBox:~$ sudo adduser hduser sudo
Adding user `hduser' to group `sudo' ...
Adding user hduser to group sudo
Done.
ubuntu@ubuntu-VirtualBox:~$
```

Step 4 – Install SSH and Create Certificates

$ sudo apt-get install ssh

```
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libntdb1 python-ntdb
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libck-connector0 ncurses-term openssh-server openssh-sftp-server
  ssh-import-id
Suggested packages:
  rssh molly-guard monkeysphere
The following NEW packages will be installed:
  libck-connector0 ncurses-term openssh-server openssh-sftp-server ssh
  ssh-import-id
0 upgraded, 6 newly installed, 0 to remove and 8 not upgraded.
Need to get 661 kB of archives.
```

$ su hduser

```
ubuntu@ubuntu-VirtualBox:~$ su hduser
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

$ ssh-keygen -t rsa -P ""

```
hduser@ubuntu-VirtualBox:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:K/F5oNmAqhY02Axp0vzew4EnrN+UGgDTgxIiFPHpT7Q hduser@ubuntu-VirtualBox
The key's randomart image is:
+---[RSA 2048]----+
|=@o              |
|@.* .            |
|=* * o           |
|.o= *.+          |
|. .=.EooS        |
| ...= *B +       |
|  o. *+.= .      |
| o  o .. .       |
|o                |
+----[SHA256]-----+
```

// Set Environmental variables

$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

```
hduser@ubuntu-VirtualBox:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_k
eys
```

Step 6 – Install Hadoop

$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.8.4/hadoop-2.8.4.tar.gz

// Extract Hadoop-2.8.4

$ sudo tar xvzf hadoop-2.8.4.tar.gz

```
hduser@ubuntu-VirtualBox:~$ tar xvzf hadoop-2.7.2.tar.gz
```

// Create a folder ‘hadoop’ in /usr/local
$ sudo mkdir –p /usr/local/hadoop

```
hduser@ubuntu-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop
[sudo] password for hduser:
```

// Move the Hadoop folder to /usr/local/hadoop
$ sudo mv hadoop-2.8.4 /usr/local/hadoop

// Assigning read and write access to Hadoop folder
$ sudo chown –R hduser:hadoop /usr/local/hadoop

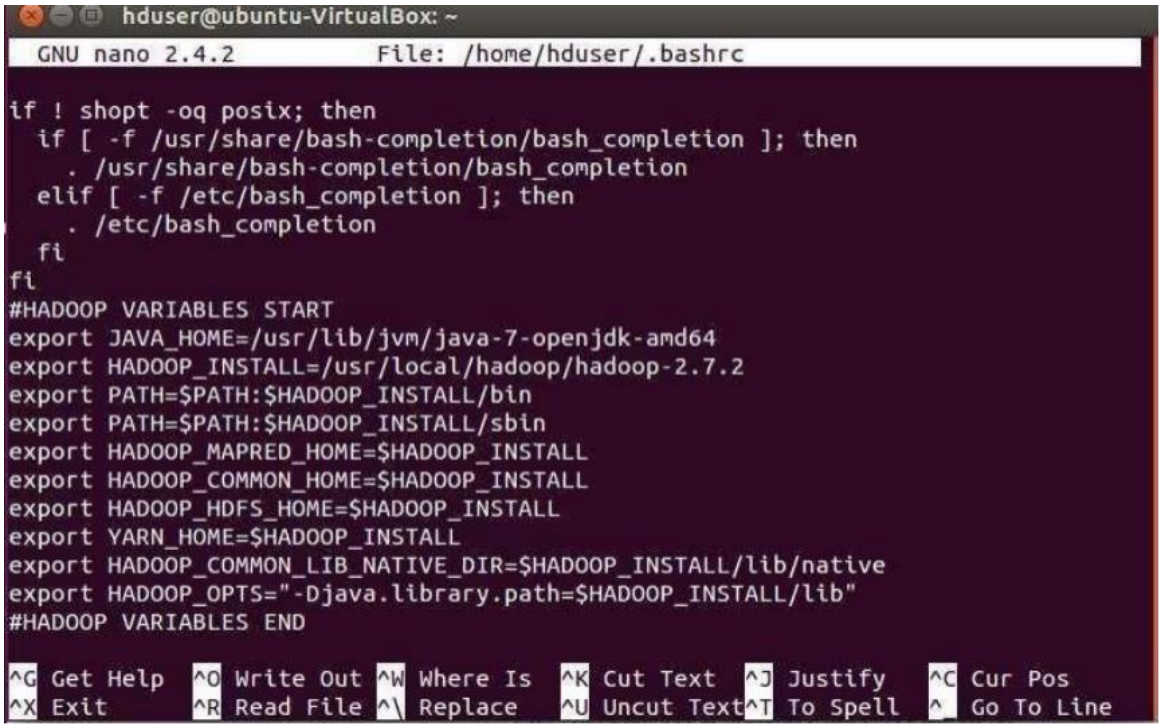
Step 7 - Modify Hadoop config files

//Hadoop Environmental variable setting – The following files will be modified

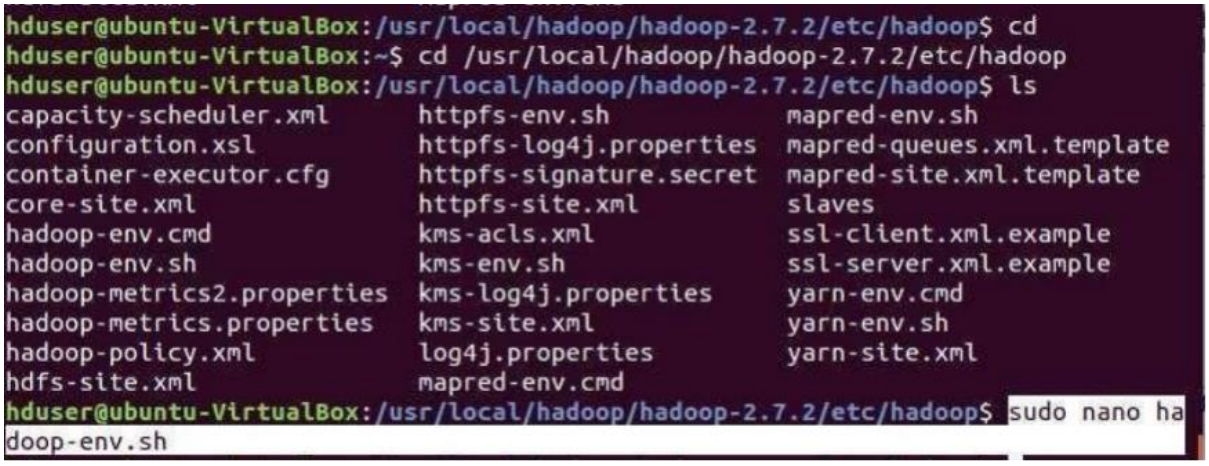
1. ~/.bashrc
2. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/hadoop-env.sh
3. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/core-site.xml
4. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/hdfs-site.xml
5. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/yarn-site.xml
6. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml.template

$ sudo nano ~/.bashrc
// Add the following lines at the end of the file

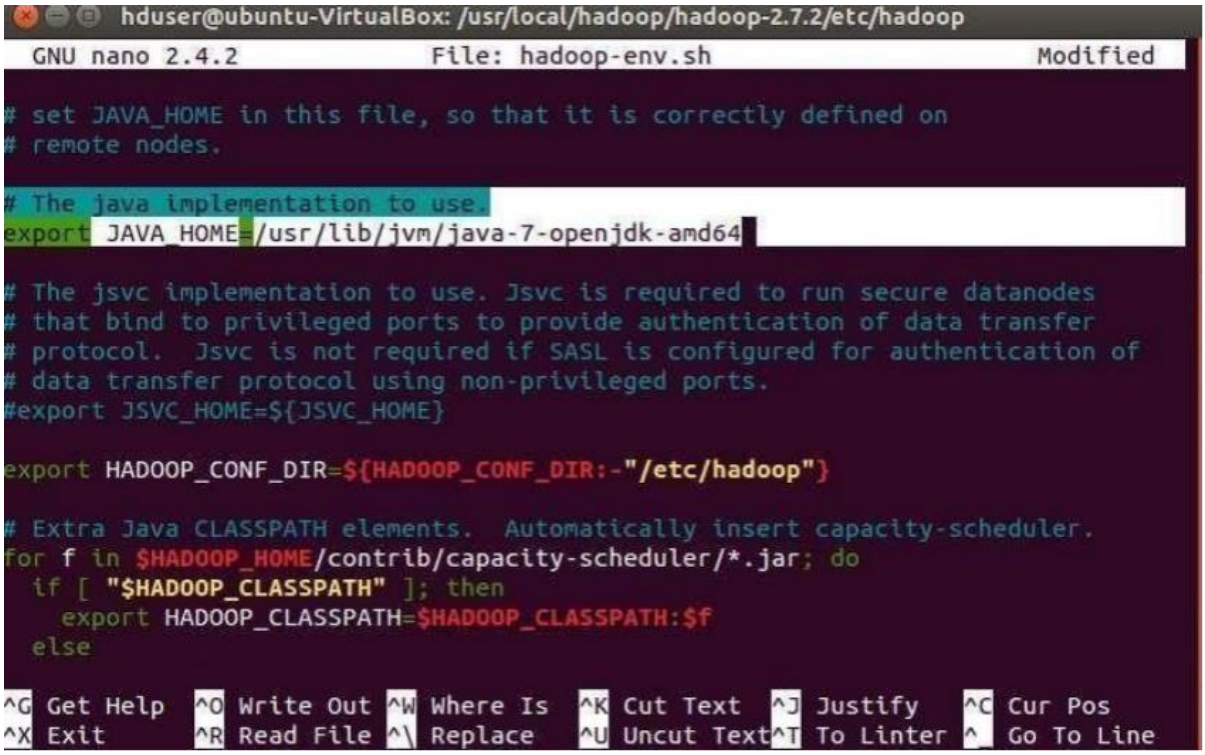
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop/hadoop-2.8.4 export
PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME export
HADOOP_COMMON_HOME=$HADOOP_HOME export
HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native export
HADOOP_OPTS="-D.java.library.path=$HADOOP_HOME/lib" export
PATH=$PATH:/usr/local/hadoop/hadoop-2.8.4/bin



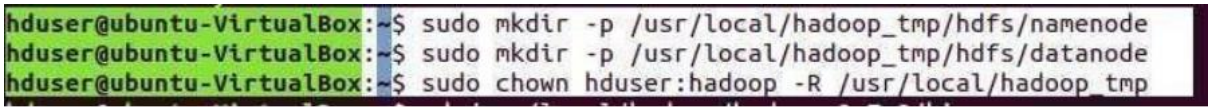
// Configure Hadoop Files
$ cd /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/
$ sudo nano hadoop-env.sh



// Add following line in hadoop-env.sh – Set JAVA variable in Hadoop
The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-8-oracle

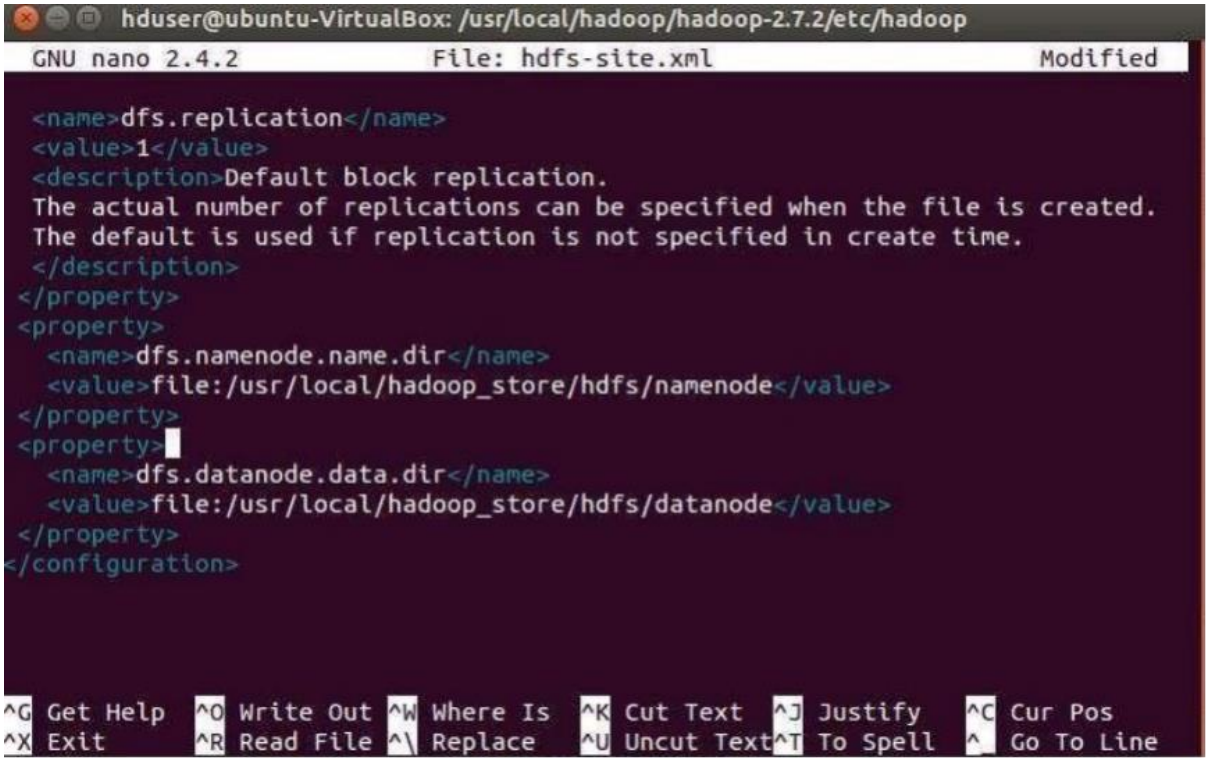


// Create datanode and namenode
$ sudo mkdir –p /usr/local/hadoop_tmp/hdfs/namenode
$ sudo mkdir –p /usr/local/hadoop_tmp/hdfs/datanode
// Changing ownership to hadoop_tmp
$ sudo chown –R hduser:hadoop /usr/local/hadoop_tmp



// Edit hdfs-site.xml
$ sudo nano hdfs-site.xml
// Add the following lines between <configuration> …… </configuration>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>

</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
</configuration>



// Edit core-site.xml

$ sudo nano core-site.xml
// Add the following lines between <configuration> …… </configuration>
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
// Edit yarn-site.xml
$ sudo nano yarn-site.xml
// Add the following lines between <configuration> …… </configuration>
<configuration>
<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.Shuffle-Handler</value>
</property>
</configuration>

// Edit mapred-site.xml

$ cp /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml

$ sudo nano mapred-site.xml

// Add the following lines between <configuration> …… </configuration>

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

Step 8 – Format Hadoop File System
$ cd /usr/local/hadoop/hadoop-2.8.4/bin
$ hadoop namenode -format

Step 9 - Start Hadoop

$ cd /usr/local/hadoop/hadoop-2.8.4/sbin

// Starting dfs services

$ start-dfs.sh

```
hduser@ubuntu-VirtualBox:/usr/local/hadoop/hadoop-2.7.2/sbin$ start-dfs.sh
16/07/15 22:55:47 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/hadoop-2.7.2/logs/had
oop-hduser-namenode-ubuntu-VirtualBox.out
localhost: starting datanode, logging to /usr/local/hadoop/hadoop-2.7.2/logs/had
oop-hduser-datanode-ubuntu-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:+j+WF1JPsOOvl5mgcc7v9A/rU8jVQEHE8WfLmt2aEo8.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts
.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/hadoop-2.7.2/l
ogs/hadoop-hduser-secondarynamenode-ubuntu-VirtualBox.out
```

// Starting mapreduce services
$ start-yarn.sh

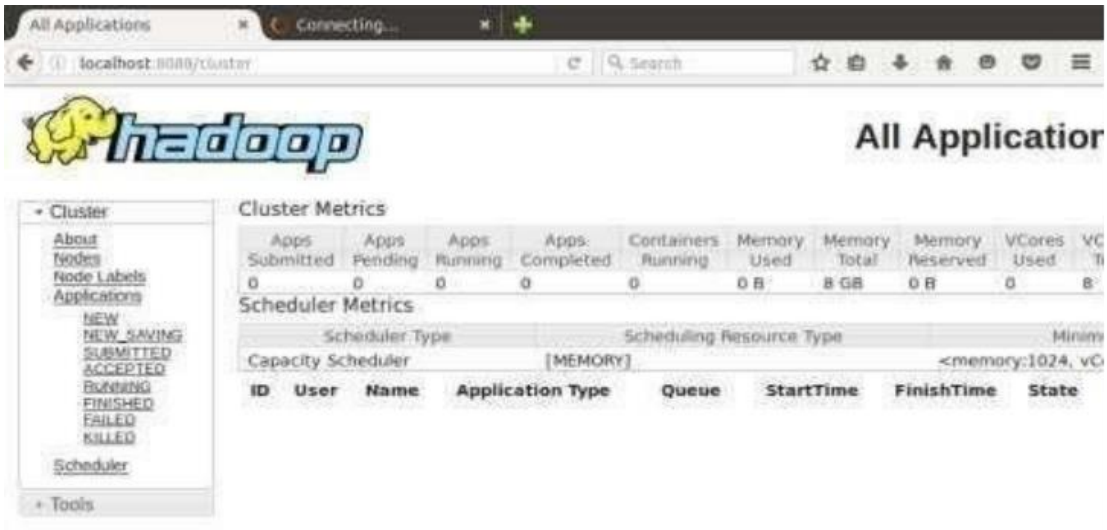
```
hduser@ubuntu-VirtualBox:/usr/local/hadoop/hadoop-2.7.2/sbin$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/hadoop-2.7.2/logs/yarn-hd
user-resourcemanager-ubuntu-VirtualBox.out
localhost: starting nodemanager, logging to /usr/local/hadoop/hadoop-2.7.2/logs/
yarn-hduser-nodemanager-ubuntu-VirtualBox.out
```

$ jps

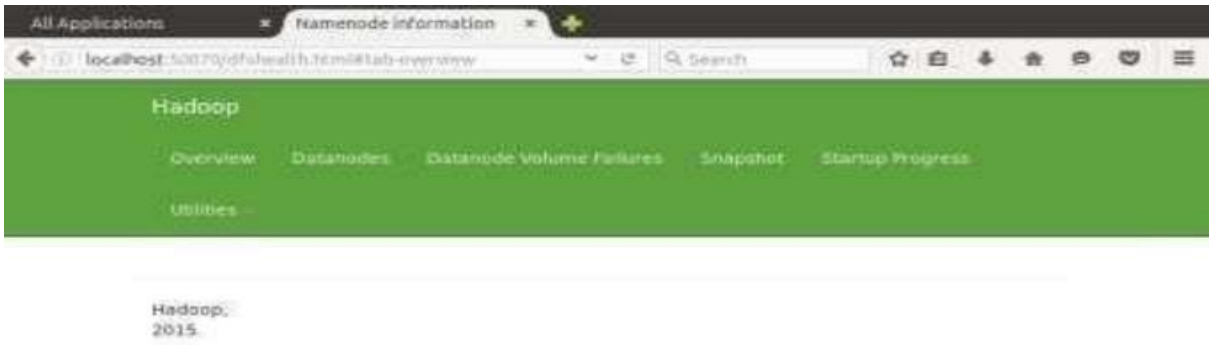
```
hduser@ubuntu-VirtualBox:/usr/local/hadoop/hadoop-2.7.2/sbin$ jps
12425 SecondaryNameNode
12609 ResourceManager
12733 NodeManager
13131 Jps
12205 DataNode
12080 NameNode
```

Step 10 - Check Hadoop through web UI

Go to browser type http://localhost:8088 – All Applications Hadoop Cluster



Go to browser type http://localhost:50070 – Hadoop Namenode



Step 11 - Stop Hadoop

$ stop-dfs.sh

$ stop-yarn.sh

RESULT

Thus, the procedure to install single-node Hadoop is executed successfully.

Exp No: 11 **Date: 03.11.2022**

DEMONSTRATE THE MAP REDUCE PROGRAMMING MODEL BY COUNTING THE NUMBER OF WORDS IN A FILE

AIM:

To demonstrate the MAP REDUCE programming model for counting the number of words in a file.

PROCEDURE

Step 1 - Open Terminal

$ su hduser

Password:

Step 2 - Start dfs and mapreduce services

$ cd /usr/local/hadoop/hadoop-2.7.2/sbin

$ start-dfs.sh

$ start-yarn.sh

$ jps

Step 3 - Check Hadoop through web UI // Go to browser type http://localhost:8088 – All Applications Hadoop Cluster // Go to browser type http://localhost:50070 – Hadoop Namenode

Step 4 – Open New Terminal

$ cd Desktop/

$ mkdir inputdata

$ cd inputdata/

$ echo “Hai, Hello, How are you? How is your health?” >> hello.txt

$ cat>> hello.txt

Step 5 – Go back to old Terminal

$ hadoop fs –copyFromLocal /home/hduser/Desktop/inputdata/hello.txt /folder/hduser // Check in hello.txt in Namenode using Web UI

Step 6 – Download and open eclipse by creating workspace

Create a new java project.

Step 7 – Add jar to the project

You need to remove dependencies by adding jar files in the hadoop source folder. Now Click on Project tab and go to Properties.Under Libraries tab, click Add External JARs and select all the jars in the folder (click on 1st jar, and Press Shift and Click on last jat to select all jars in between and click ok)

/usr/local/hadoop/hadoop-2.7.2/share/hadoop/commonand

/usr/local/hadoop/hadoop-2.7.2/share/hadoop/mapreduce folders.

Step -8 – WordCount Program

Create 3 java files named

- WordCount.java
- WordCountMapper.java
- WordCountReducer.java

WordCount.java

```
import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapred.FileOutputFormat;

import org.apache.hadoop.mapred.JobClient; import

org.apache.hadoop.mapred.JobConf;
```

```
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
import org.apache.hadoop.io.Text;
public class WordCount extends Configured implements Tool {
@Override
        public int run(String[] arg0) throws Exception {
                // TODO Auto-generated method
                stub if(arg0.length<2)
                {
System.out.println("check the command line arguments");
                 }
                JobConf conf=new JobConf(WordCount.class);
                FileInputFormat.setInputPaths(conf, new Path(arg0[0]));
                        FileOutputFormat.setOutputPath(conf, new
Path(arg0[1])); conf.setMapperClass(WordMapper.class);
conf.setReducerClass(WordReducer.class);
                        conf.setOutputKeyClass(Text.class);
                        conf.setOutputValueClass(IntWritable.class);
                        conf.setOutputKeyClass(Text.class);
                        conf.setOutputValueClass(IntWritable.class);
                        JobClient.runJob(conf);
                        return 0;
        }
```

```
        public static void main(String args[]) throws Exception
        {
                int exitcode=ToolRunner.run(new WordCount(),
                args); System.exit(exitcode);
        }
}
```

WordCountMapper.java

```
  import java.io.IOException;
  import org.apache.hadoop.io.IntWritable;
  import org.apache.hadoop.io.LongWritable;
  import org.apache.hadoop.mapred.MapReduceBase;
  import org.apache.hadoop.mapred.OutputCollector;
  import org.apache.hadoop.mapred.Reporter;
  import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.Mapper;
  public class WordCountMapper extends MapReduceBase implements
  Mapper<LongWritable,Text,Text,IntWritable>
  {
          @Override
        public void map(LongWritable arg0, Text arg1, OutputCollector<Text,
  IntWritable> arg2, Reporter arg3)
                        throws IOException {
                // TODO Auto-generated method stub
```

```
                String s=arg1.toString();
                for(String word:s.split(" "))
                {
arg2.collect(new Text(word),new IntWritable(1));
                }
        }
}
```

WordCountReducer.java

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.io.Text;
public class WordCountReducer implements Reducer<Text,IntWritable,Text,IntWritable> {
@Override
public void configure(JobConf arg0) {
                // TODO Auto-generated method stub
        }
        @Override
        public void close() throws IOException {
                // TODO Auto-generated method stub
```

```
        }

        @Override

        public void reduce(Text arg0, Iterator<IntWritable> arg1,
OutputCollector<Text, IntWritable> arg2, Reporter arg3)

                throws IOException {

            // TODO Auto-generated method

            stub int count=0;

            while(arg1.hasNext())

            {

                IntWritable i=arg1.next();

                count+=i.get();

            }

            arg2.collect(arg0,new IntWritable(count));

            }

}
```

Step 9 - Creatr JAR file

Now Click on the Run tab and click Run-Configurations. Click on New Configuration button on the left top side and Apply after filling the following properties.

Step 10 - Export JAR file

Now click on File tab and select Export. under Java, select Runnable Jar.

In Launch Config – select the config fie you created in Step 9 (WordCountConfig).

➢ Select an export destination (let's say desktop.)

➢ Under Library handling, select Extract Required Libraries into generated JAR and click Finish. ➢ Right-Click the jar file, go to Properties and under Permissions tab, Check Allow executing file

as a program. and give Read and Write access to all the users

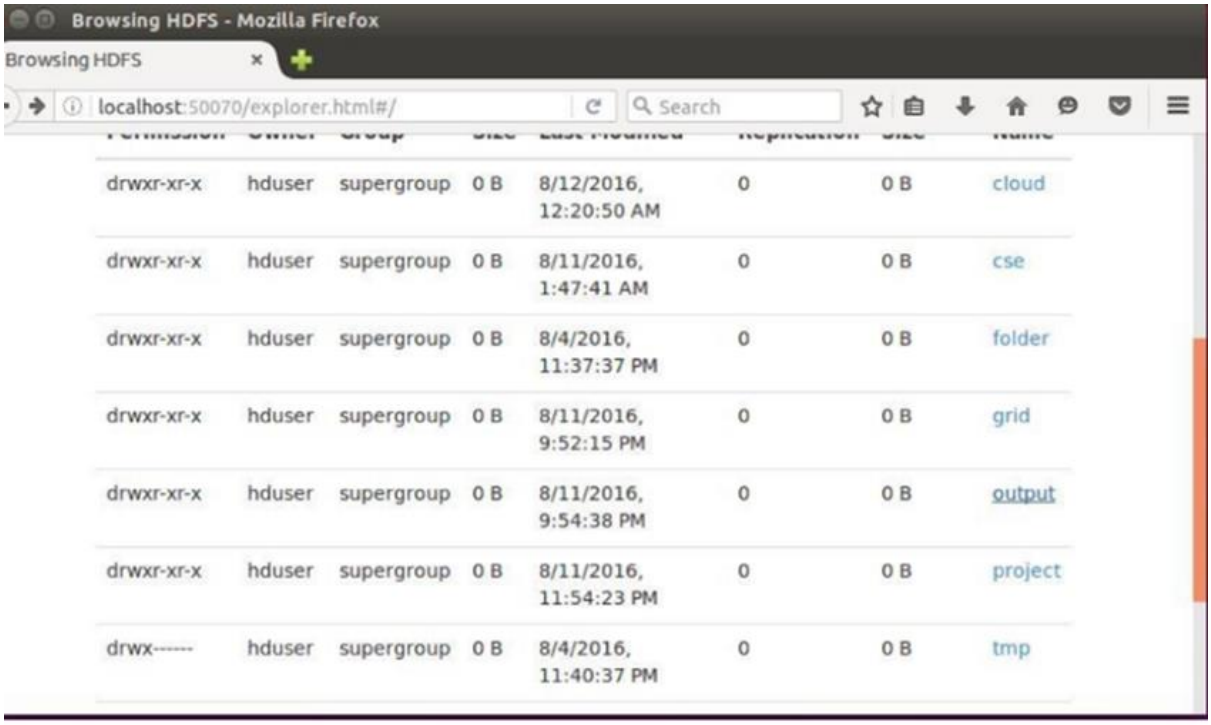
Step 11 – Go back to old Terminal for Execution of WordCount Program $hadoop jar wordcount.jar/usr/local/hadoop/input/usr/local/hadoop/output

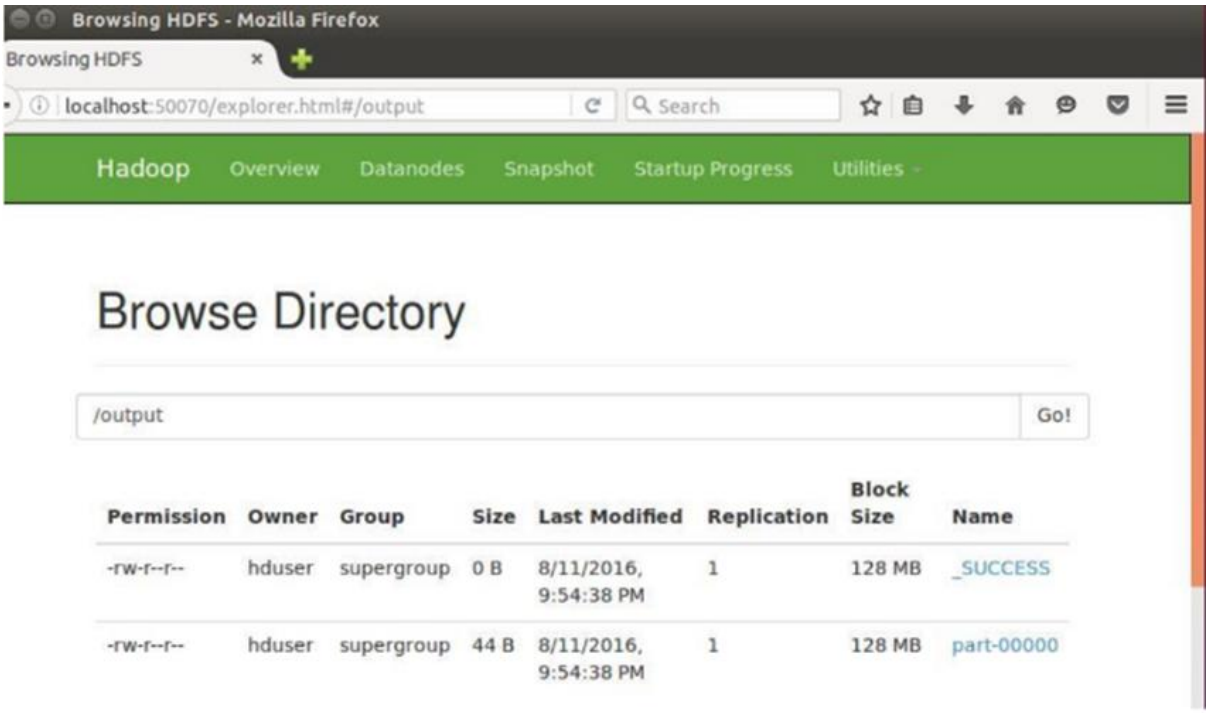
Step 12 – To view results in old Terminal
$hdfs dfs -cat /usr/local/hadoop/output/part-r-00000

Step 13 - To Remove folders created using hdfs

$ hdfs dfs -rm -R /usr/local/hadoop/output

OUTPUT





RESULT

Thus the map reduce programming model for counting the number of words in a file has been executed successfully.

Exp No: 12 **Date: 10.11.2022**

IMPLEMENT THE MAX TEMPERATURE MAPREDUCE PROGRAM TO IDENTIFY THE YEAR WISE MAXIMUM TEMPERATURE FROM SENSOR DATA

AIM

To implement the Max temperature MapReduce program to identify the year wise maximum temperature from the sensor data.

Description

Sensors senses weather data in big text format containing station ID, year, date, time, temperature, quality etc. from each sensor and store it in a single line. Suppose thousands of data sensors are there, then we have thousands of records with no particular order. We require only a year and maximum temperature of particular quality in that year.

For example:
Input string from sensor:
002902907099999*1902*010720004+64333+023450
FM-12+
000599999V0202501N027819999999N0000001N9-*00331*+
99999098351ADDGF102991999999999999999999

Here: 1902 is year
0033 is temperature
1 is measurement quality (Range between 0 or 1 or 4 or 5 or 9)

Here each mapper takes the input key as "byte offset of line" and value as "one weather sensor read i.e one line". and parse each line and produce an intermediate key "year" and intermediate value as "temperature of certain measurement qualities" for that year.

The combiner will form set values of temperature. Year and set of values of temperatures is given as input <key, value> to reducer and Reducer will produce year and maximum temperature for that year from the set of temperature values.

PROGRAM

*/
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;

```
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.LongWritable; import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper; import
org.apache.hadoop.mapreduce.Reducer;


//Mapper class


class MaxTemperatureMapper
extends Mapper<LongWritable, Text, Text, IntWritable> { private static final int MISSING

= 9999;

@Override
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {

String line = value.toString(); String year = line.substring(15, 19); int airTemperature;
if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs airTemperature =
Integer.parseInt(line.substring(88, 92));
} else {
airTemperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (airTemperature != MISSING && quality.matches("[01459]")) { context.write(new
Text(year), new IntWritable(airTemperature));
}
}
}
//Reducer class
class MaxTemperatureReducer
extends Reducer<Text, IntWritable, Text, IntWritable> {
@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException {
int maxValue = Integer.MIN_VALUE; for (IntWritable value : values) {
maxValue = Math.max(maxValue, value.get());
}
```

```
context.write(key, new IntWritable(maxValue));
}
}
//Driver Class
public class MaxTemperature {
public static void main(String[] args) throws Exception { if (args.length != 2) {
System.err.println("Usage: MaxTemperature <input path=""> <output path>"); System.exit(-
1);
}
Job job = Job.getInstance(new Configuration()); job.setJarByClass(MaxTemperature.class);
job.setJobName("Max temperature");
FileInputFormat.addInputPath(job, new Path(args[0])); FileOutputFormat.setOutputPath(job,
new Path(args[1]));
job.setMapperClass(MaxTemperatureMapper.class);
job.setReducerClass(MaxTemperatureReducer.class);
job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class);
job.submit();
}
}
```

OUTPUT:

Input for String :

0029029070999991902010720004+64333+023450FM-12+
000599999V0202501N027819999999N0000001N9-00331+
99999098351ADDGF102991999999999999999999

RESULT

Output Text contain year and maximum temperature in that year as 1902 33