



FRANCIS XAVIERTM

ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

ACCREDITED BY NBA | ISO 9001:2015 Certified | DST-FIST Supported Institution
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956
Vannarpettai, Tirunelveli - 627003, Tamil Nadu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY RECORD

(Project Based Learning)

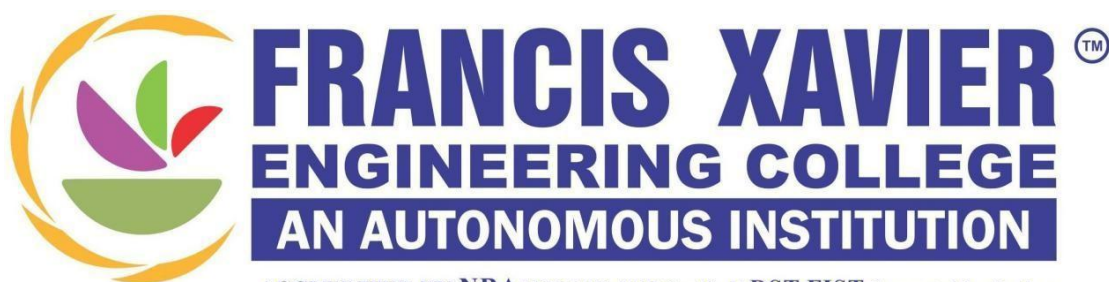
21CS7611 – Cloud Computing Laboratory

NAME :

REGISTER NO. :

ACADEMIC YEAR / SEM : 2024 - 2025 / VII

CLASS / BRANCH / SECTION : IV CSE A



ACCREDITED BY NBA | ISO 9001:2015 Certified | DST-FIST Supported Institution
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956
Vannarpettai, Tirunelveli - 627003, Tamil Nadu

BONAFIDE CERTIFICATE

Name	
Reg. No.	
Branch	
Semester	
Year/Section	
Course code/ Title	

Certified that this is a bonafide record of work done by the above student, in the “Cloud Computing” Laboratory during the academic year 2024-2025.

Course Instructor

HoD

Submitted for the End Semester Practical Examination on

Internal Examiner

External Examiner

INDEX

S.No.	DATE	TITLE OF THE EXPERIMENT	PAGE NO.	MARK	SIGN
1		Create virtual machine with different flavors of linux or windows OS on top of windows10 or11.	1		
2		Develop a procedure to transfer the files from one virtual machine to another virtual machine.	6		
3		Installation of Docker from Dockerhub and creating Containers using Dockers and uploading the containers in cloud.	8		
4		Build a Docker image and publish in cloud.	13		
5		Develop a procedure to create a Secure cloud.	17		
6		Develop a procedure to install storage controller and interact with it.	22		
7		Develop a procedure create a one node cluster.	27		
8		Write a word count program to demonstrate the use of Map and Reduce tasks.	33		

**EX NO : 1. CREATE VIRTUAL MACHINE WITH DIFFERENT FLAVORS OF LINUX
OR WINDOWS OS ON TOP OF WINDOWS10 OR 11.**

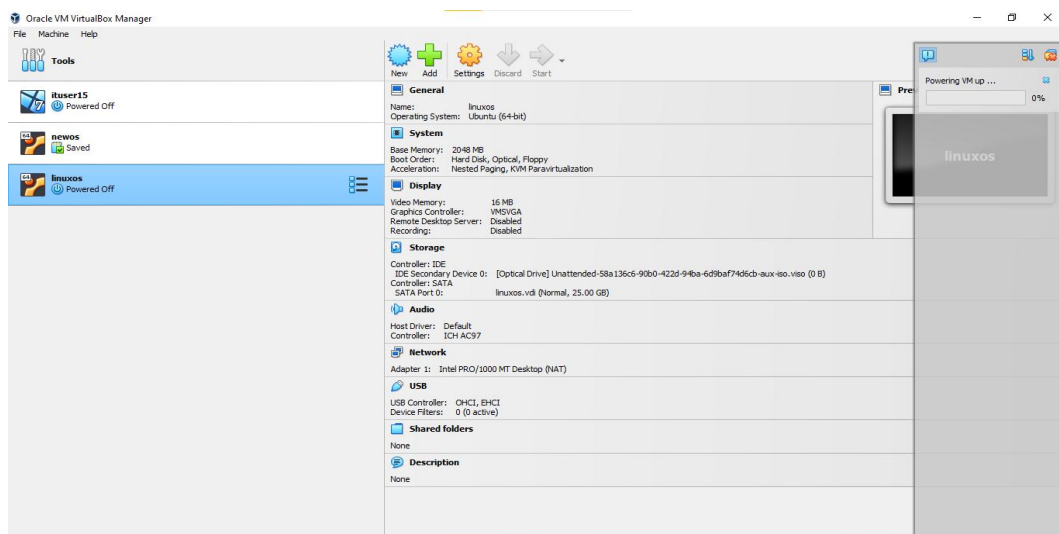
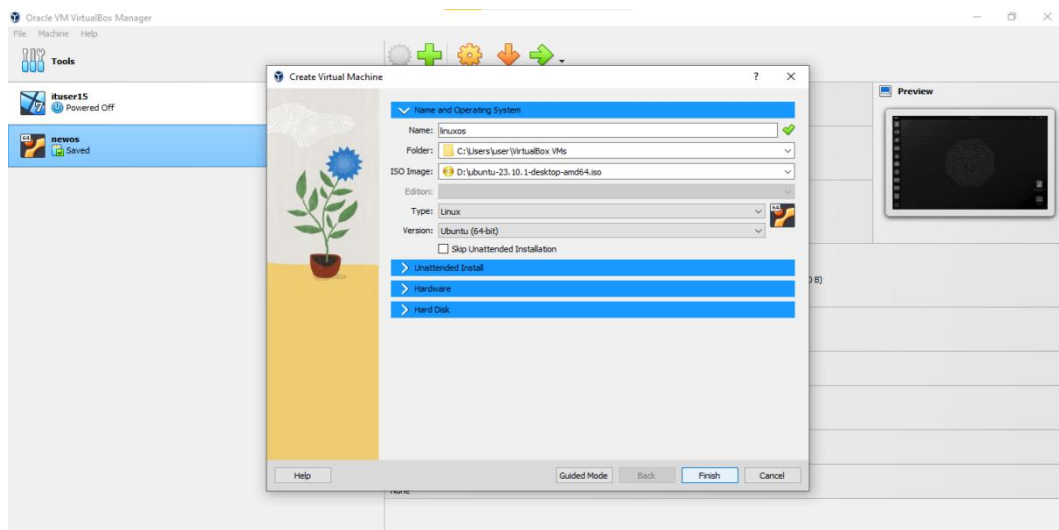
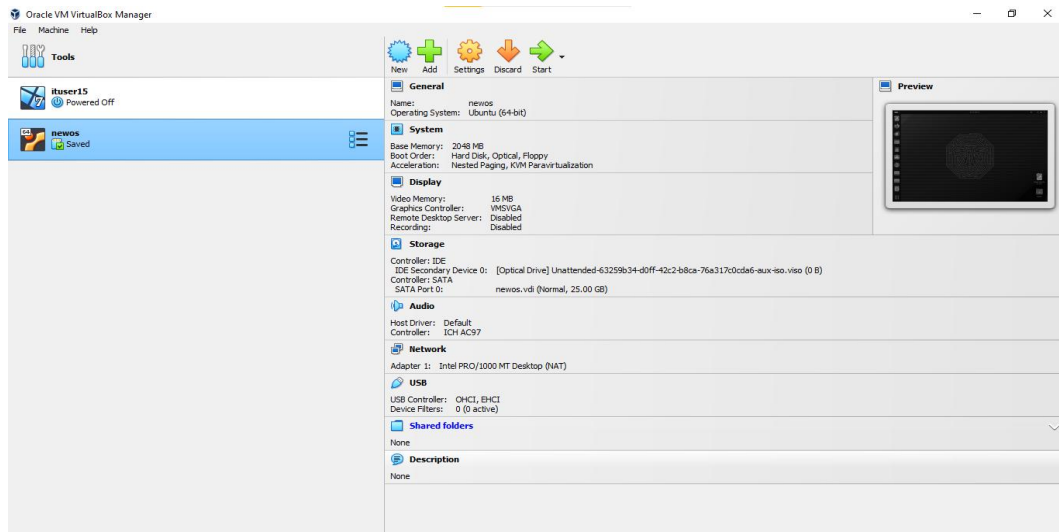
AIM:

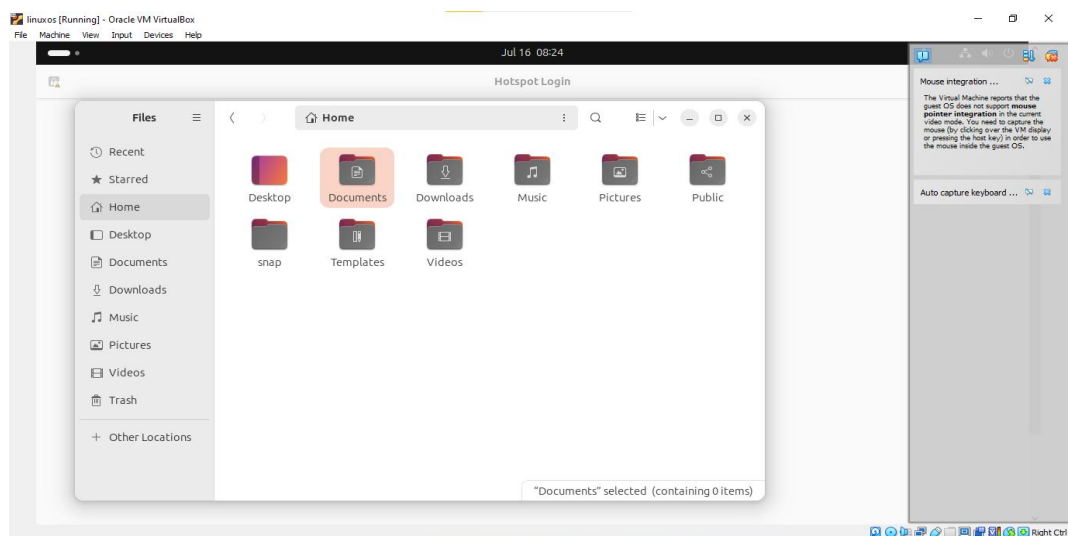
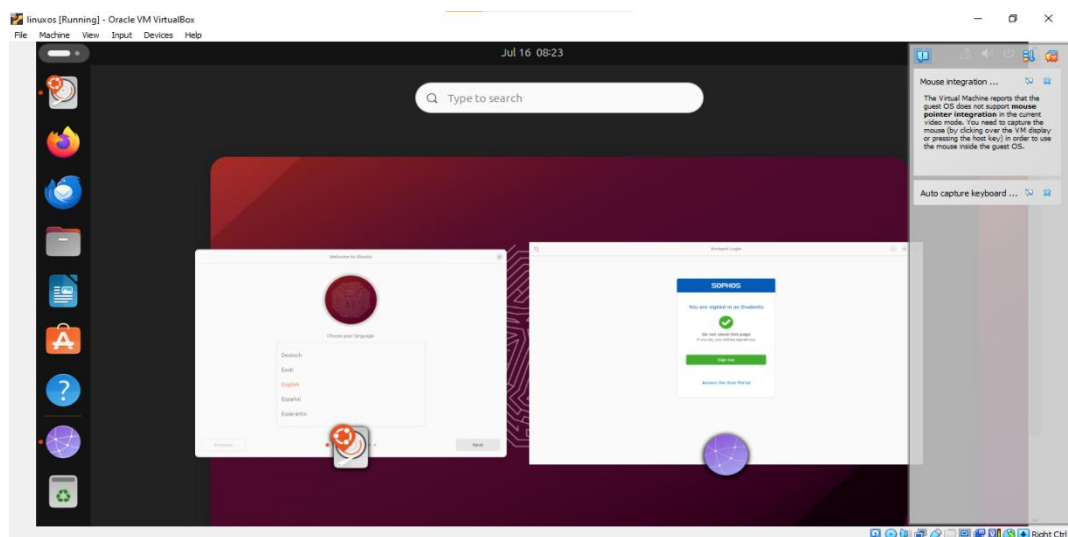
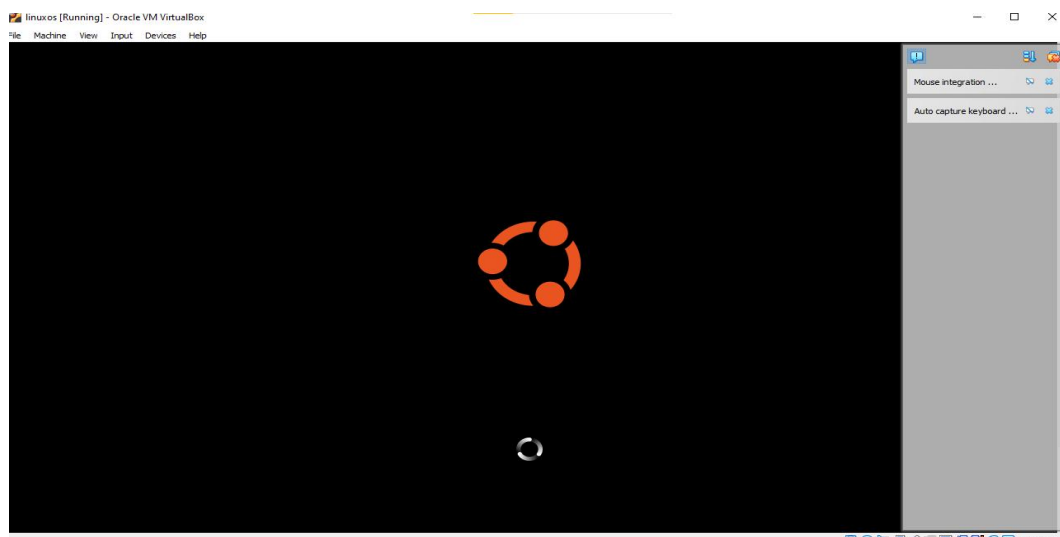
To Install & Create a virtual machine using VMware Software.

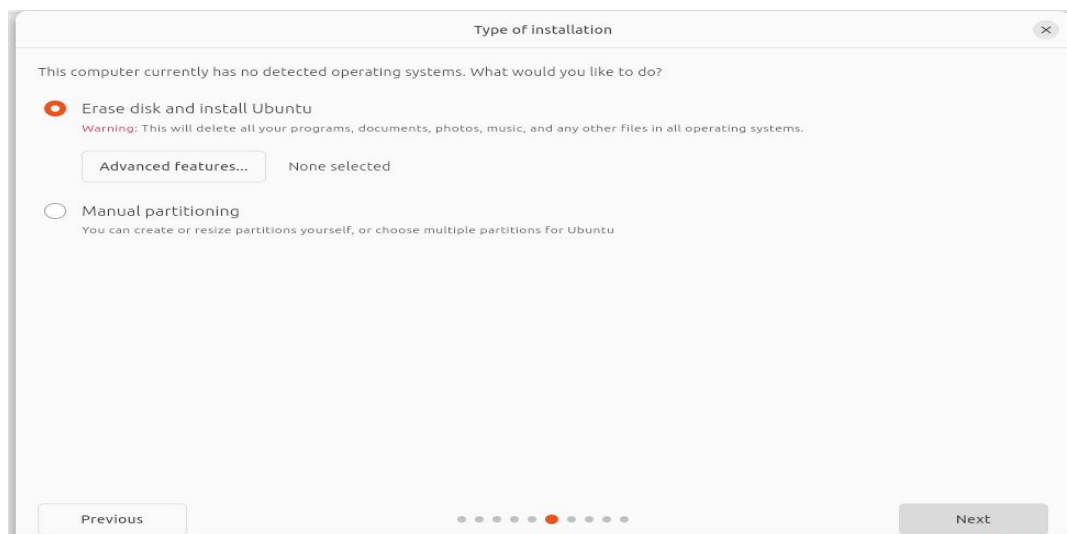
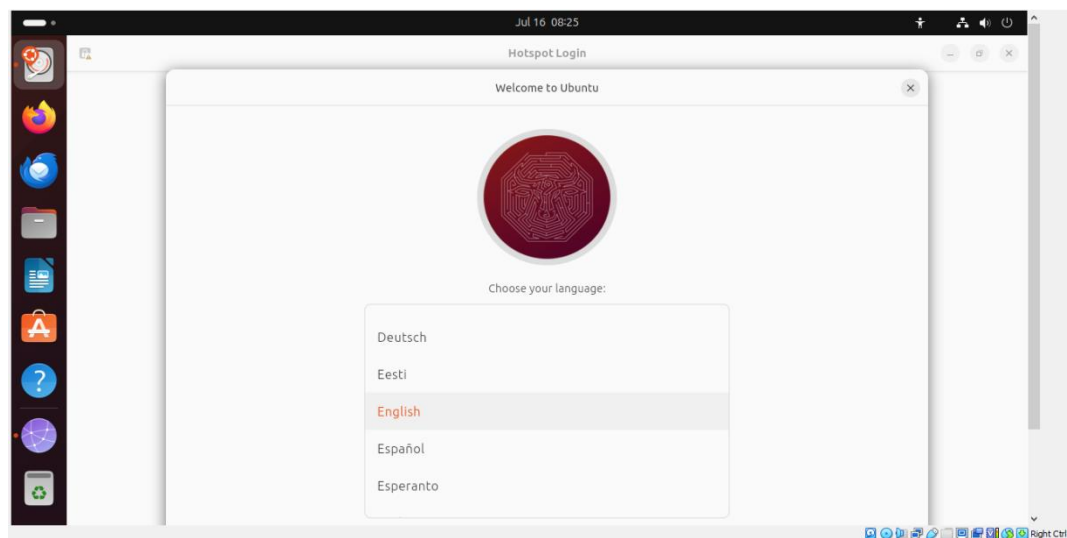
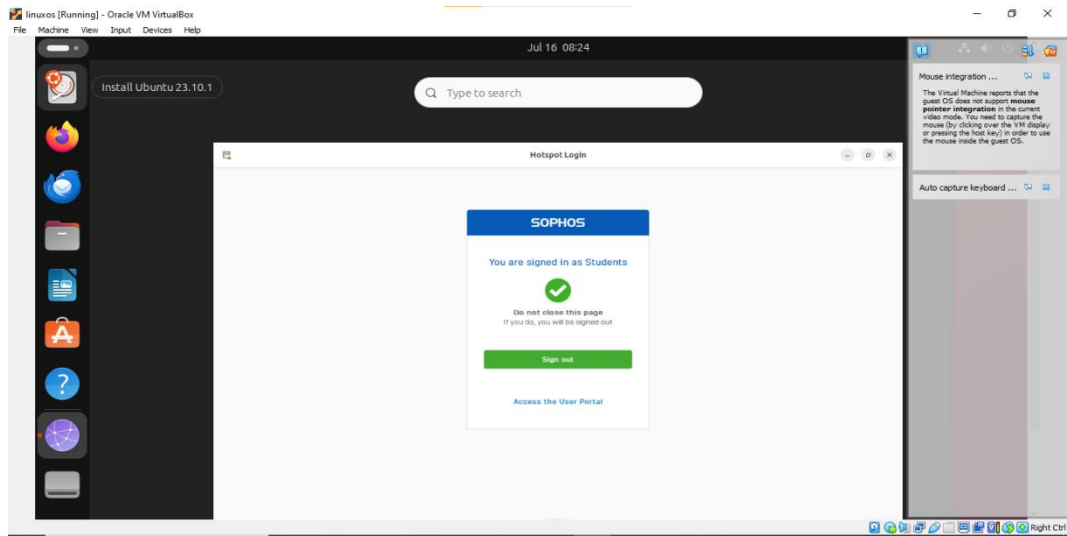
PROCEDURE:

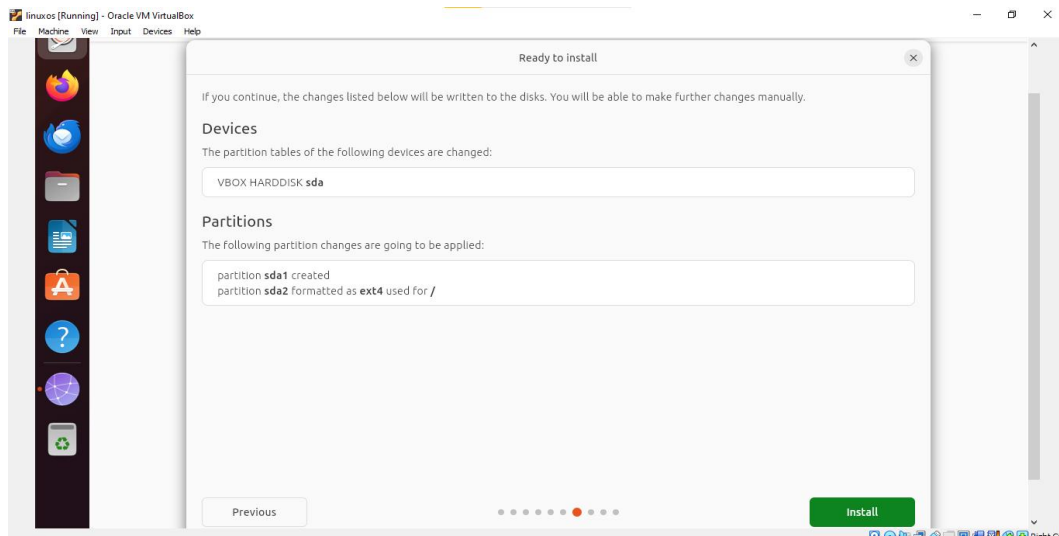
1. Create a new virtual machine.
2. Customize the virtual machine by specifying the hardware configuration.
3. Allocate memory for virtual machine
4. Select the ISO image of the OS to be installed and proceed.

OUTPUT:









Result:

Thus the virtual machine is created using VMware and windows XP OS is installed in the virtual machine.

EXNO :2 . FIND A PROCEDURE TO TRANSFER THE FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUALMACHINE

AIM:

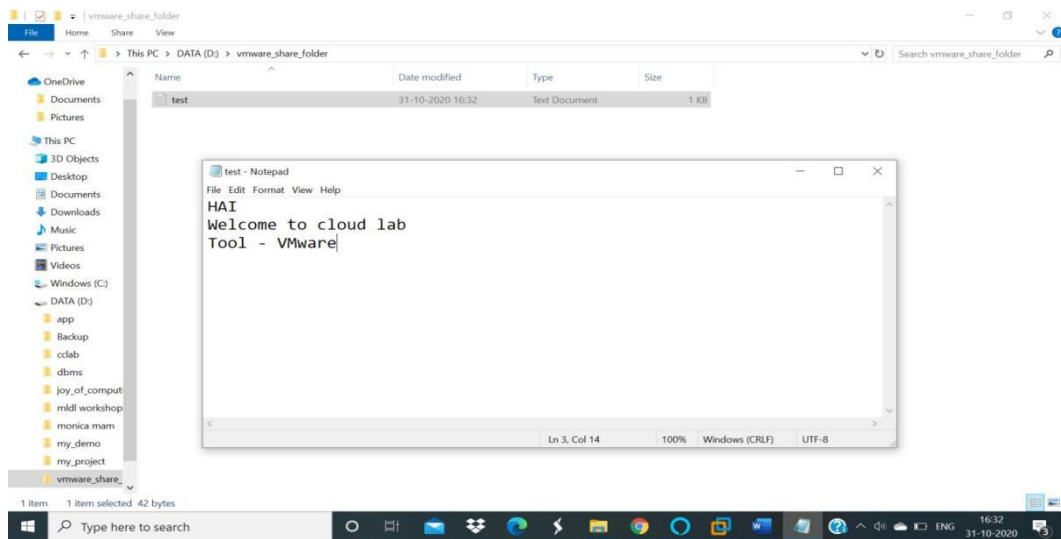
To share files between physical machine and virtual machine.

PROCEDURE:

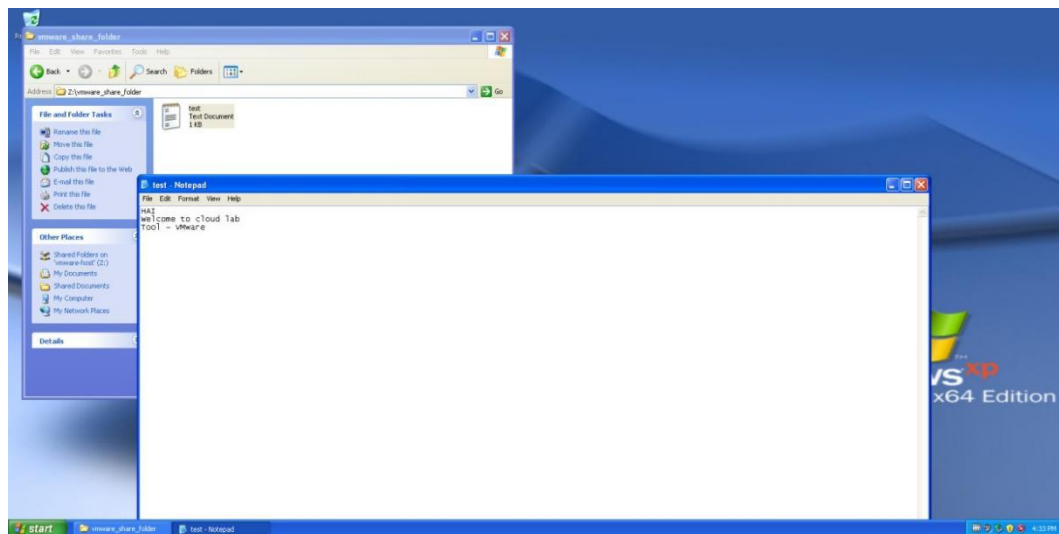
1. Open a virtual machine.
2. Select a file which is to be shared in the machine.
3. Select a already existing file for sharing or create a new file for sharing.
4. Open VMware workstation and go to VM tab and click install VMware tools.
5. Open a virtual machine to which the file has to shared and power on.
6. Goto settings of the virtual machine.
7. In VM Settings go to options.
8. Select the “shared folder” option
9. Enable the shared folder option by selecting “always enable” , check “Map as a networkdrive in windows path” and click add and click next.
10. Select the path of the shared folder from the host and press ok and then next.
11. Specify the attributes of shared folder, click finish and then ok.
12. GotoMyComputer and click refresh to see the shared folder.
13. Check for shareability.
14. Goto Host Machine open the file and check for updation.
15. Make some changes in the file in host machine and check for the updation in the virtual machine.

OUTPUT:

In Host Machine:



In Virtual Machine:



Result:

Thus the file is shared between physical and virtual machine.

EX NO : 3. INSTALLATION OF DOCKER FROM DOCKERHUB AND CREATING CONTAINERS USING DOCKERS AND UPLOADING THE CONTAINERS IN CLOUD.

AIM:

Installation of Docker from Dockerhub and creating Containers using Dockers and uploading the containers in cloud.

PROCEDURE:

1. Installing Docker

```
``bash
# Update package index
sudo apt-get update

# Install required packages
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

# Add Docker's official GPG key
curl -fsSL
https://download.docker.com/linux/ubuntu/gpg |
sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

# Set up stable repository
echo \
    "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee
/etc/sources.list.d/docker.list > /dev/null

# Install Docker Engine
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli
containerd.io

# Verify installation
sudo docker run hello-world
``
```

2. Create DockerHub Account

- Visit hub.docker.com
- Sign up for a new account
- Verify email address

3. Login to DockerHub via CLI

```
docker login
# Enter username and password when prompted
```

4. Create Sample Application

```
# Create project directory
mkdir docker-webapp
cd docker-webapp

# Create a simple web application
cat > index.html << EOF
<!DOCTYPE html>
<html>
<head>
<title>Docker Demo</title>
</head>
<body>
<h1>Welcome to Docker Container Demo</h1>
<p>This is a sample web page running in a Docker
container.</p>
</body>
</html>
EOF
```

5. Create Dockerfile

```
``dockerfile
FROM nginx:alpine
COPY index.html /usr/share/nginx/html/
EXPOSE 80
``
```

6. Build and Run Container

```
``bash
# Build image
docker build -t mywebapp:v1 .

# Run container
docker run -d -p 8080:80 mywebapp:v1

# List running containers
docker ps
``
```

7. Tag and Push to DockerHub

```
```bash
Tag image
docker tag mywebapp:v1
yourusername/mywebapp:v1

Push to DockerHub
docker push yourusername/mywebapp:v1
```
```

OUTPUT:

```
~# sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

```
teco
→ ~ docker login -u tericcabrel
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
→ ~
```

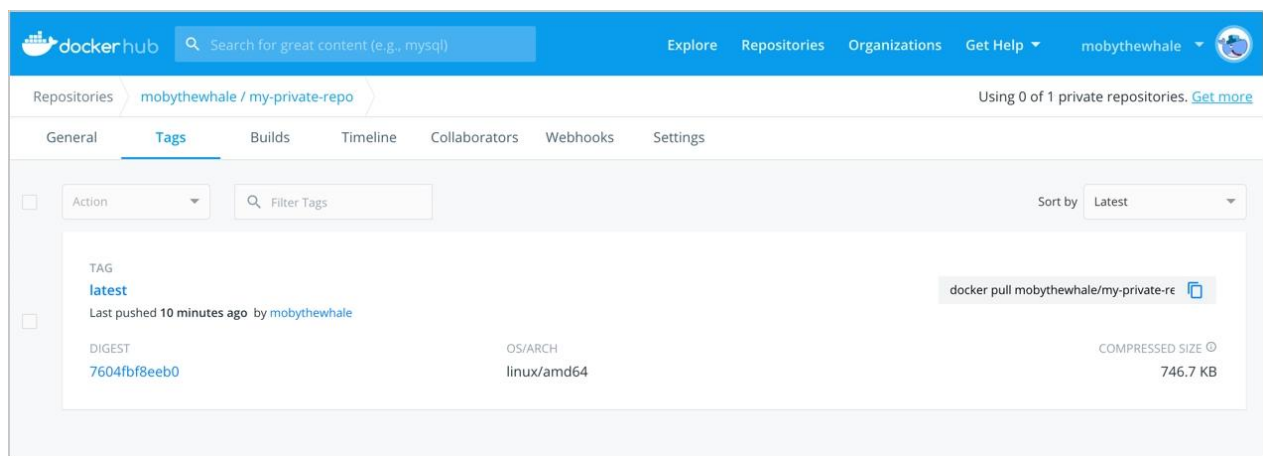
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER ...
[+] Building 42.3s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 130B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:14-alpine3.16
=> [1/4] FROM docker.io/library/node:14-alpine3.16@sha256:9b1499b199458ded11c14b1e0174396bccf98f126cc1e
=> => resolve docker.io/library/node:14-alpine3.16@sha256:9b1499b199458ded11c14b1e0174396bccf98f126cc1e
=> => sha256:243413fb782f6a2fbc5a3acdd609c2317455de043e3478f371657fd42dc3a402 1.16kB / 1.16kB
=> => sha256:11c67d6660258a58a84d81485401a15749af5c36cc16123ba0c6aa4f84cac204 6.46kB / 6.46kB
```

```
tim@nova:/home/./local/tmp/ % docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
946501d04a32       mysql              "/entrypoint.sh mysql" 9 hours ago
Up 9 hours          3306/tcp           phpmyadmin-mysql

tim@nova:/home/./local/tmp/ % docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
mysql               latest             6762f304c834       2 weeks ago
283.5 MB
corbinu/docker-phpmyadmin latest           9fe36d18c039       11 weeks ago
417.8 MB
ubuntu             14.04             d2a0ecffe6fa       11 weeks ago
188.4 MB
ubuntu             latest            d2a0ecffe6fa       11 weeks ago
188.4 MB
```

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\xampp\htdocs\PetStore> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
laibarazikhan/laravel_petstore   latest             6d84e647f921       9 days ago         835MB
sail-8.2/app          latest             6d84e647f921       9 days ago         835MB
mysql                 latest             412b8cc72e4a       2 weeks ago        531MB
redis                alpine             1c1b270ed420       3 weeks ago        29.9MB
mysql/mysql-server    8.0                1d9c2219ff69       3 months ago       496MB
PS C:\xampp\htdocs\PetStore> docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
PS C:\xampp\htdocs\PetStore>
```



RESULT:

Docker from Dockerhub is Installed and Containers using Dockers and uploading the containers in cloud is created.

EX NO : 4. BUILD A DOCKER IMAGE AND PUBLISH IN CLOUD.

AIM:

To build a Docker image and publish in cloud.

PROCEDURE:

1. Prerequisites Setup

- Install Docker Desktop on the local machine
- Create an account on a cloud platform (e.g., Docker Hub)
- Login to Docker Hub using CLI:

```
docker login
```

2. Create Application Files

- Create a new directory for the project:

```
mkdir docker-demo  
cd docker-demo
```

- Create a simple application (e.g., Python web app)

```
```python  
app.py
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
 return "Hello from Docker!"

if __name__ == '__main__':
 app.run(host='0.0.0.0', port=5000)
```
```

- Create requirements file:

```
echo "flask" > requirements.txt
```


3. Create Dockerfile

```
``dockerfile
FROM python:3.8-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY app.py .
EXPOSE 5000
CMD ["python", "app.py"]
``
```

4. Build Docker Image

```
docker build -t yourusername/docker-demo:v1 .
```

5. Test Locally

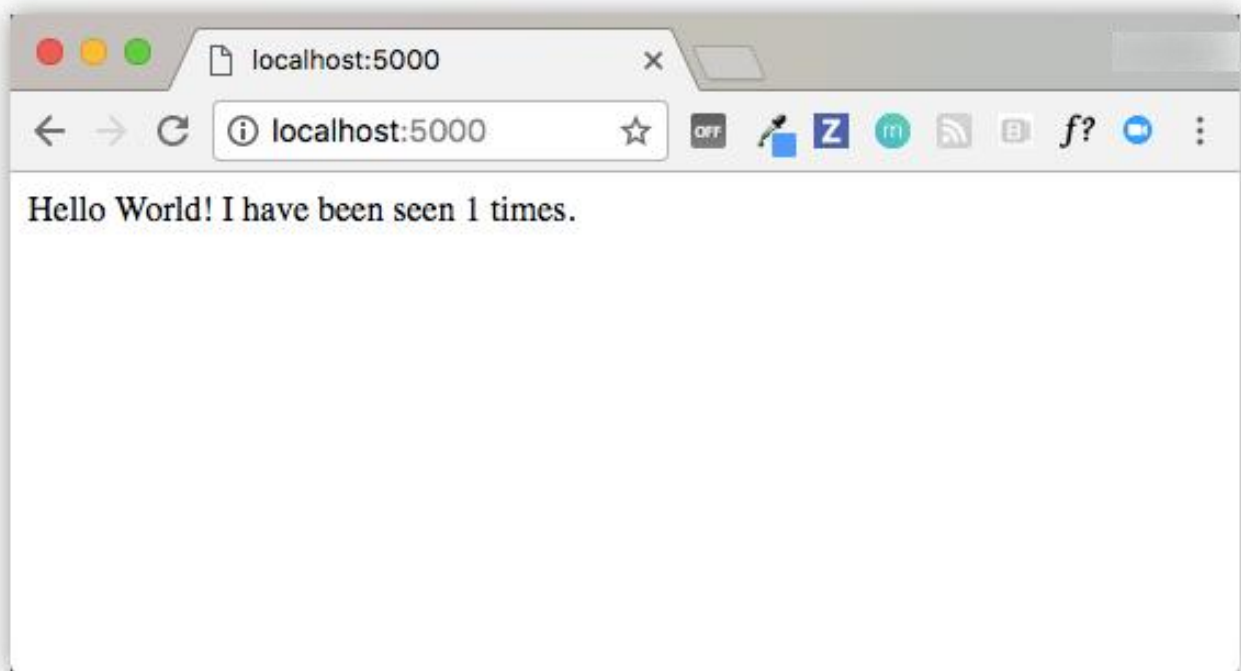
```
docker run -p 5000:5000 yourusername/docker-demo:v1
```

6. Push to Cloud Registry

```
docker push yourusername/docker-demo:v1
```

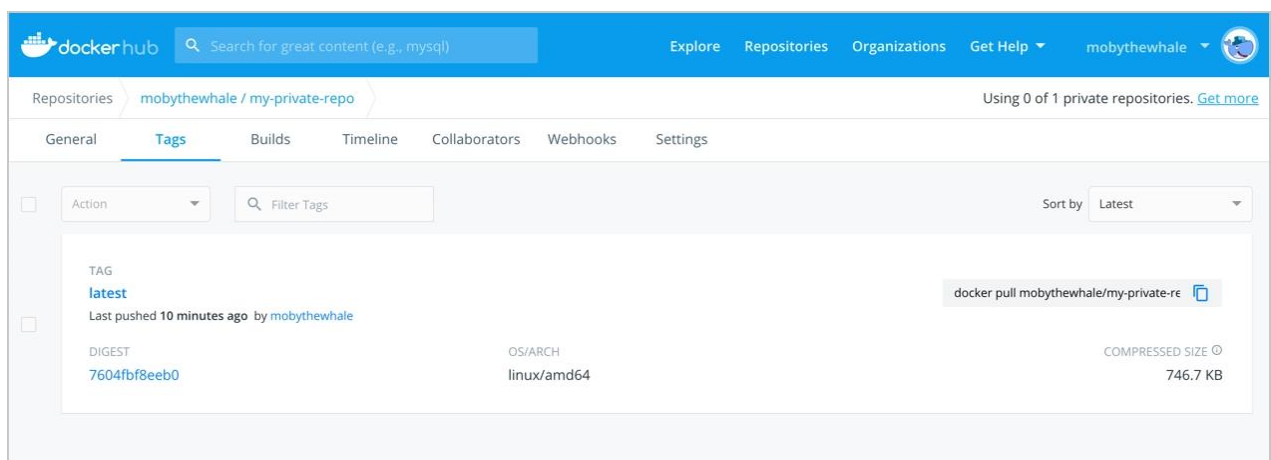
OUTPUT:

```
hemantasundaray@: ~/Desktop/docker $ docker build . -t docker-demo:v1
[+] Building 7.1s (5/5) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 153B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/alpine:latest 6.9s
=> CACHED [1/1] FROM docker.io/library/alpine:latest@sha256:02bb6f428431fbc2809c5d1b41eab5a68350 0.0s
=> => resolve docker.io/library/alpine:latest@sha256:02bb6f428431fbc2809c5d1b41eab5a68350194fb50 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:2910eca4ae53906f69b40b88ee2cae079bce6cbedd3deb608cedb4bc755e2e6d 0.0s
=> => naming to docker.io/library/docker-demo:v1 0.0s
```



```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL powershell + - [ ] [ ] [ ] [ ] [ ]
PS C:\xampp\htdocs\PetStore> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
laibarazikhan/laravel_petstore   latest             6d84e647f921       9 days ago         835MB
sail-8.2/app          latest             6d84e647f921       9 days ago         835MB
mysql                 latest             412b8cc72e4a       2 weeks ago        531MB
redis                 alpine            1c1b270ed420       3 weeks ago        29.9MB
mysql/mysql-server     8.0               1d9c2219ff69       3 months ago       496MB
PS C:\xampp\htdocs\PetStore> docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
PS C:\xampp\htdocs\PetStore> 
```



RESULT:

Thus a Docker image is built and published in cloud.

EX NO :5. DEVELOP A PROCEDURE TO CREATE A SECURE CLOUD.

AIM:

To install a open nebula and to create a virtual machine.

PROCEDURE:

1. Open-nebula offers a simple but feature-rich and flexible solution to build and manage enterprise clouds and virtualized data centers.
2. OpenNebula is designed to be simple.
3. Simple to install, update and operate by the admins, and simple to use by end users.

OUTPUT:

```
grep -E 'svm|vmx' /proc/cpuinfo
```

Package Layout

opennebula-common: Provides the user and common files

libopennebula-ruby: All ruby libraries **opennebula-node:** Prepares a node as an opennebula-node

opennebula-sunstone:

OpenNebula Sunstone Web Interface

opennebula-tools: Command Line interface

opennebula-gate: Gate server that enables communication between VMs and OpenNebula

opennebula-flow: Manages services and elasticity

opennebula: OpenNebula Daemon

Step 1. Installation in the Frontend

1. Add the repository

```
sudo wget -q -O- http://downloads.opennebula.org/repo/Ubuntu/repo.key | apt-key add -  
sudo echo "deb http://downloads.opennebula.org/repo/Ubuntu/14.04 stable opennebula"
```

> /etc/apt/sources.list.d/opennebula.list

2. Install the required packages

```
sudo apt-get update  
sudo apt-get install opennebula opennebula-sunstone nfs-kernel-server
```

3. Configure and start the services

There are two main processes that must be started, the main OpenNebuladaemon: **oned**, and the graphical user interface: **sunstone**.

Sunstone listens only in the loopback interface by default for security reasons. To change it edit /etc/one/sunstone-server.conf and

change :host: 127.0.0.1 to

:host: 0.0.0.0. The command to restart the Sunstone:

4. Configure Network File Service (NFS) (This is not needed if both Frontend and Nodes are in the

```
sudo /etc/init.d/opennebula-sunstone restart
```

same machine)

```
/var/lib/one/ *(rw,sync,no_subtree_check,root_squash)
```

Export /var/lib/one/ from the frontend to the worker nodes. To do so add the following to the /etc/exports file in the frontend:

Refresh the NFS export by the following command

```
sudo service nfs-kernel-server restart
```

5. Configure SSH public key

```
sudo su - oneadmin  
cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

Add the following snippet to ~/.ssh/config so it doesn't prompt to add the keys to the

```
$ cat << EOT > ~/.ssh/config  
Host *  
    StrictHostKeyChecking no  
    UserKnownHostsFile /dev/null
```

known_hosts file:.

Step 2. Installation in the Nodes

1. Install the required packages

```
sudo apt-get install opennebula-node nfs-common bridge-utils
```

2. Configure the network

In DHCP, edit /etc/network/interfaces

```
auto lo iface lo inet  
loopbackauto br0  
iface br0 inet dhcp bridge_ports eth0 bridge_fd 9  
bridge_hello 2  
bridge_maxage 12 bridge_stpoff
```

Restart the network

```
sudo /etc/init.d/networking restart
```

3. Configure NFS (This is not needed if both Frontend and Nodes are in the same machine) edit the file /etc/fstab as

```
<Frontend IP>:/var/lib/one/ /var/lib/one/ nfs soft,intr,rsize=8192,wsz=8192,noauto  
Mount the NFS  
sudo mount /var/lib/one
```

4. Configure Qemu oneadmin user must be able to manage libvirt as root

```
# cat << EOT >
```

```
/etc/libvirt/qemu.conf user = "oneadmin"  
group = "oneadmin" dynamic_ownership = 0 EOT
```

Restart libvirt

```
sudo service libvirt-bin restart
```

Step 3: Start the sunstone from the web browser

```
http://frontend:9869
```

oneadmin password is available in the file `~/.one/one_auth`

1. From the command line, to login as oneadmin in the Frontend , execute

```
sudo su - oneadmin
```

2. Adding Host

To start running VMs, you should first register a worker node for OpenNebula

onehost create localhost -i kvm -v kvm -n dummy

Run **onehost list** - Try these command until it's on. If there is any failure, look at `/var/log/one/oned.log`

3. Adding Virtual resources

To create networks, create a network template file "mynetwork.one" and

```
NAME = "private"
TYPE = FIXED
BRIDGE = br0
LEASES = [ IP=192.168.0.100 ]
LEASES = [ IP=192.168.0.101 ]
LEASES = [ IP=192.168.0.102 ]
```

edit the file as follows

Create resources in opennebula

```
$ onevnet create mynetwork.one
$ oneimage create --name "CentOS-6.5_x86_64" \
  --path "http://appliances.c12g.com/CentOS-6.5/centos6.5.qcow2.gz" \
  --driver qcow2 \
  --datastore default
$ onetemplate create --name "CentOS-6.5" --cpu 1 --vcpu 1 --memory 512 \
  --arch x86_64 --disk "CentOS-6.5_x86_64" --nic "private" --vnc \
  --ssh
```

Monitor the resources are created by running the command **oneimage list**

In order to dynamically add ssh keys to Virtual Machines we must add

```
$ EDITOR = vi oneuser update oneadmin
```

our ssh key to the user template, by editing the user

```
SSH_PUBLIC_KEY="ssh-dss AAAAB3NzaC1kc3MAAACBANBWTQmm4Gt..."
```

template: Add a new line to the template (cat ~/.ssh/id_dsa.pub)

4. To run a virtual machine

\$ onetemplate instantiate "Centos-6.5" --name "My Scratch VM"

To test, execute `onevm list`, VM going from Pending to Prolog to Running. If it fails, check the log as

`/var/log/one/<VM_ID>/vm.log`

Result:

Thus open nebula is installed and created a virtual machine successfully.

EX NO :6 . DEVELOP A PROCEDURE TO INSTALL STORAGE CONTROLLER AND INTERACT WITH IT.

AIM:

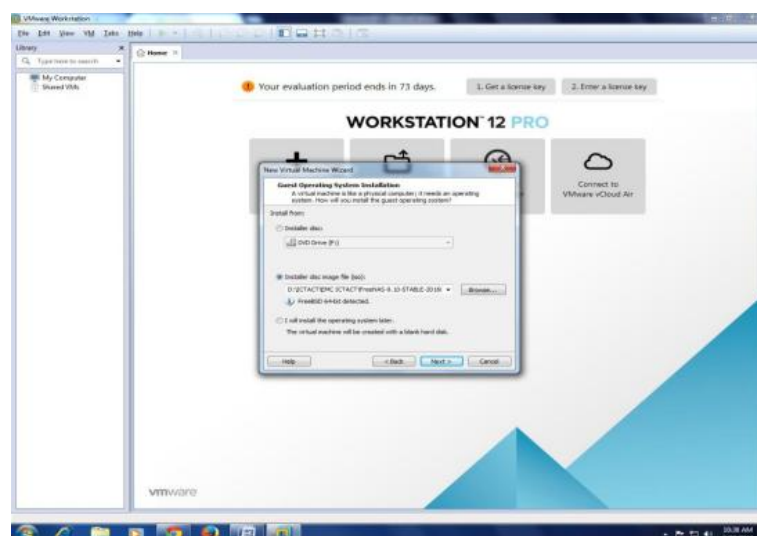
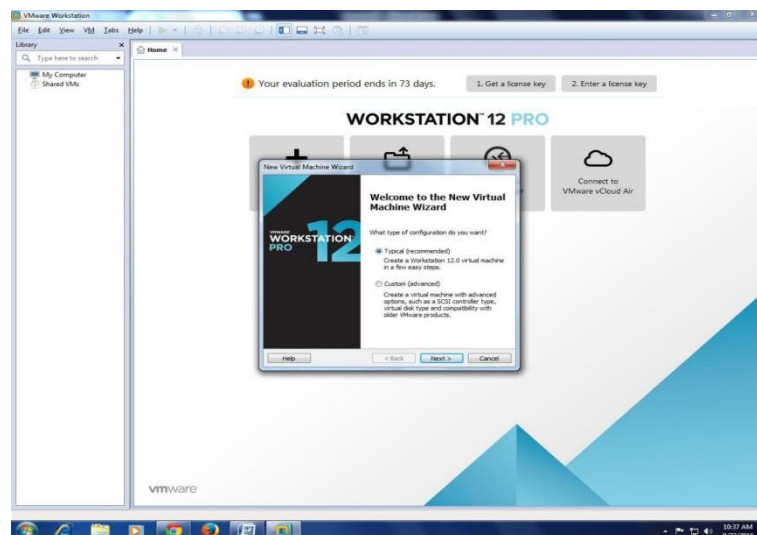
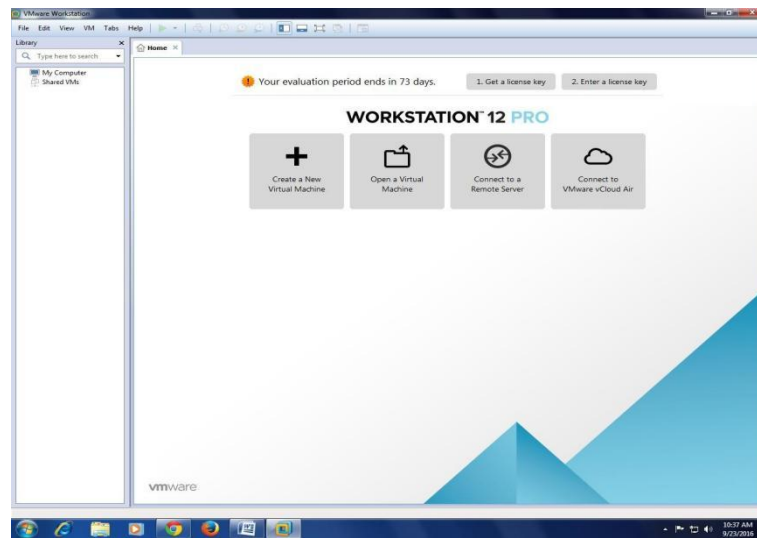
To install a Network Storage Controller.

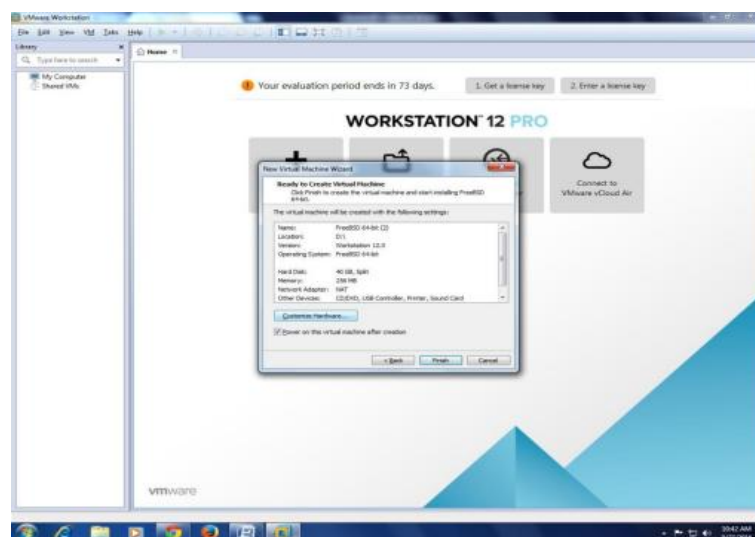
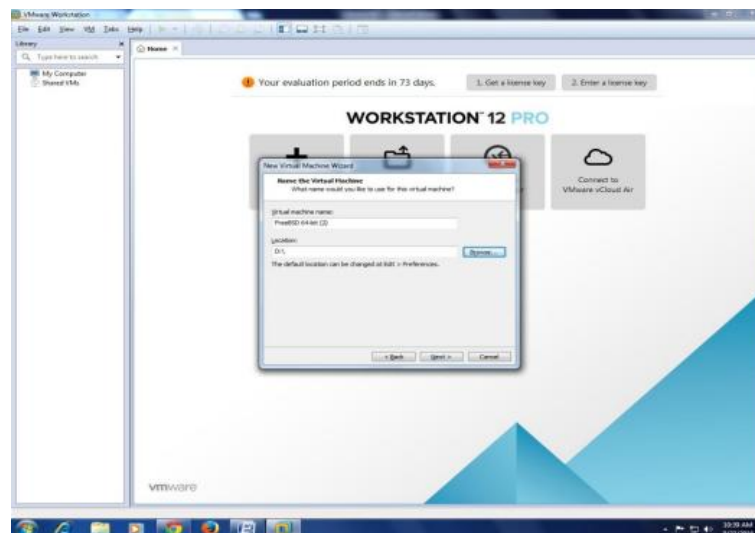
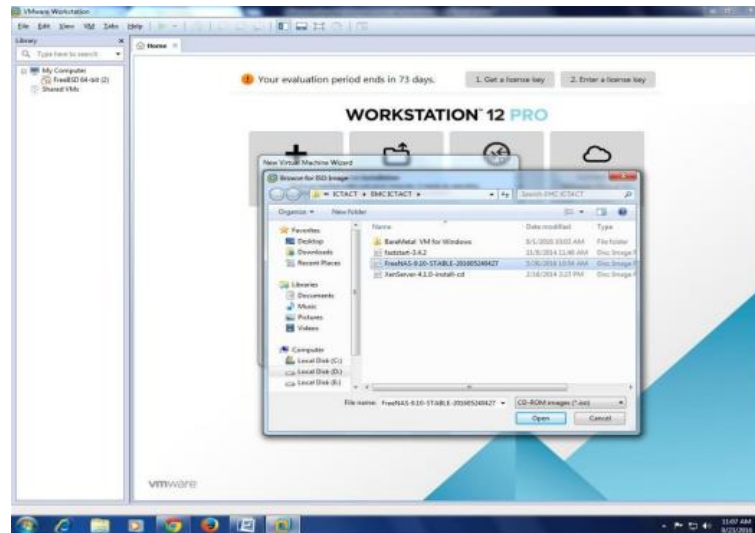
PROCEDURE:

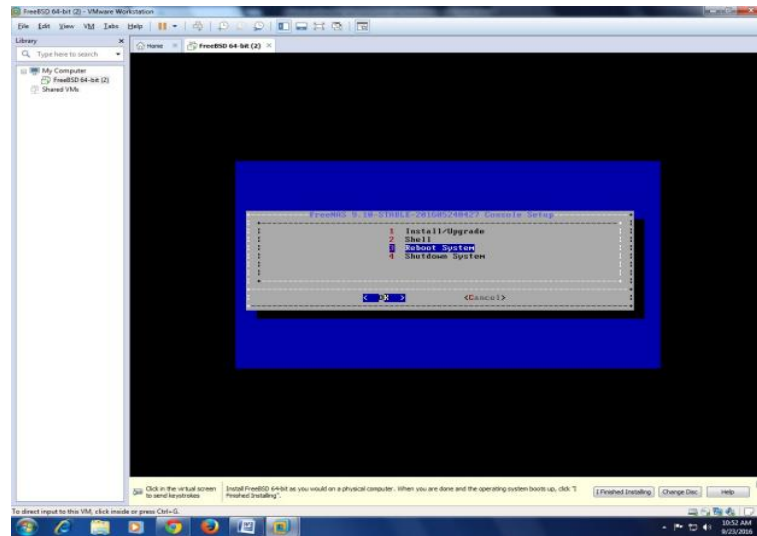
1. Create a virtual machine.
2. Configure the virtual machine.
3. Select the ISO image of the OS to be installed and proceed.

OUTPUT:

Open the Virtual Machine Software







RESULT :

Thus the storage controller installation was done successfully

EX NO :7. DEVELOP A PROCEDURE CREATE A ONE NODE CLUSTER.

AIM:

To mount the one node Hadoop cluster using FUSE and access files on HDFS in the same way as we do on Linux operating systems.

PROCEDURE:

FUSE (Filesystem in Userspace) enables you to write a normal user application as a bridge for a traditional filesystem interface.

The hadoop-hdfs-fuse package enables you to use your HDFS cluster as if it were a traditional filesystem on Linux. It is assumed that you have a working HDFS cluster and know the hostname and port that your NameNode exposes.

To install fuse-dfs on Ubuntu systems:

```
hdpuser@jiju-PC:~$ wget http://archive.cloudera.com/cdh5/one-click-  
install/trusty/amd64/cdh5-repository_1.0_all.deb
```

```
--2016-07-24 09:10:33-- http://archive.cloudera.com/cdh5/one-click-install/trusty/amd64/cdh5-  
repository_1.0_all.deb
```

```
Resolving archive.cloudera.com (archive.cloudera.com)... 151.101.8.167
```

```
Connecting to archive.cloudera.com (archive.cloudera.com)|151.101.8.167|:80... connected. HTTP  
request sent, awaiting response... 200 OK
```

```
Length: 3508 (3.4K) [application/x-debian-package] Saving to: 'cdh5-repository_1.0_all.deb'
```

```
100%[=====>] 3,508 --K/s in 0.09s
```

```
2016-07-24 09:10:34 (37.4 KB/s) - 'cdh5-repository_1.0_all.deb' saved [3508/3508] hdpuser@jiju-  
PC:~$ sudo dpkg -i cdh5-repository_1.0_all.deb
```

```
Selecting previously unselected package cdh5-repository.
```

```
(Reading database ... 170607 files and directories currently installed.) Preparing to unpack cdh5-  
repository_1.0_all.deb ...
```

```
Unpacking cdh5-repository (1.0) ... Setting up cdh5-repository (1.0) ...
```

```
gpg: keyring `/etc/apt/secring.gpg' created
```

```
gpg: keyring `/etc/apt/trusted.gpg.d/cloudera-cdh5.gpg' created
```

```
gpg: key 02A818DD: public key "Cloudera Apt Repository" imported gpg: Total number processed:  
1
```

```
gpg: imported: 1
```

```
hdpuser@jiju-PC:~$ sudo apt-get update
```

```
hdpuser@jiju-PC:~$ sudo apt-get install hadoop-hdfs-fuse
```

Reading package lists... Done Building dependency tree Reading state information... Done

The following extra packages will be installed:

avro-libs bigtop-jsvc bigtop-utils curl hadoop hadoop-0.20-mapreduce hadoop-client hadoop-hdfs
hadoop-mapreduce hadoop-yarn libcurl3 libhdfs0 parquet parquet-format zookeeper

The following NEW packages will be installed:

avro-libs bigtop-jsvc bigtop-utils curl hadoop hadoop-0.20-mapreduce hadoop-client hadoop-hdfs
hadoop-hdfs-fuse hadoop-mapreduce hadoop-yarn libhdfs0 parquet parquet-format zookeeper

The following packages will be upgraded:

libcurl3

1 upgraded, 15 newly installed, 0 to remove and 702 not upgraded. Need to get 222 MB of archives.

After this operation, 267 MB of additional disk space will be used. Do you want to continue? [Y/n]
Y

Get:1 <http://in.archive.ubuntu.com/ubuntu/trusty-updates/main> libcurl3 amd64 7.35.0-1ubuntu2.7 [173 kB]

Get:2 <https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/trusty-cdh5/contrib> avro-libs all 1.7.6+cdh5.8.0+112-1.cdh5.8.0.p0.74~trusty-cdh5.8.0 [47.0 MB]

Get:3 <http://in.archive.ubuntu.com/ubuntu/trusty-updates/main> curl amd64 7.35.0-1ubuntu2.7 [123 kB]

Get:4 <https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/trusty-cdh5/contrib> parquet-format all 2.1.0+cdh5.8.0+12-1.cdh5.8.0.p0.70~trusty-cdh5.8.0 [479 kB]

Get:5 <https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/trusty-cdh5/contrib> parquet all 1.5.0+cdh5.8.0+174-1.cdh5.8.0.p0.71~trusty-cdh5.8.0 [27.1 MB]

Get:6 <https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/trusty-cdh5/contrib> hadoop all 2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0 [28.2 MB]

Get:7 <https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/trusty-cdh5/contrib> libhdfs0 amd64 2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0 [320 kB]

Get:8 <https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/trusty-cdh5/contrib> hadoop-hdfs-fuse amd64 2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0 [317 kB]

Fetches 222 MB in 3min 28s (1,064 kB/s)

(Reading database ... 170612 files and directories currently installed.) Preparing to unpack .../libcurl3_7.35.0-1ubuntu2.7_amd64.deb ...

Unpacking libcurl3:amd64 (7.35.0-1ubuntu2.7) over (7.35.0-1ubuntu2) ... Selecting previously unselected package curl.

Preparing to unpack .../curl_7.35.0-1ubuntu2.7_amd64.deb ... Unpacking curl (7.35.0-1ubuntu2.7) ...

Selecting previously unselected package avro-libs.

Preparing to unpack .../avro-libs_1.7.6+cdh5.8.0+112-1.cdh5.8.0.p0.74~trusty-cdh5.8.0_all.deb

...

Unpacking avro-libs (1.7.6+cdh5.8.0+112-1.cdh5.8.0.p0.74~trusty-cdh5.8.0) ... Selecting previously unselected package bigtop-utils.

Preparing to unpack .../bigtop-utils_0.7.0+cdh5.8.0+0-1.cdh5.8.0.p0.72~trusty-cdh5.8.0_all.deb

...

Unpacking bigtop-utils (0.7.0+cdh5.8.0+0-1.cdh5.8.0.p0.72~trusty-cdh5.8.0) ... Selecting previously unselected package bigtop-jsvc.

Preparing to unpack .../bigtop-jsvc_0.6.0+cdh5.8.0+847-1.cdh5.8.0.p0.74~trusty-cdh5.8.0_amd64.deb ...

Unpacking bigtop-jsvc (0.6.0+cdh5.8.0+847-1.cdh5.8.0.p0.74~trusty-cdh5.8.0) ... Selecting previously unselected package zookeeper.

Preparing to unpack .../zookeeper_3.4.5+cdh5.8.0+94-1.cdh5.8.0.p0.76~trusty-cdh5.8.0_all.deb

...

Unpacking zookeeper (3.4.5+cdh5.8.0+94-1.cdh5.8.0.p0.76~trusty-cdh5.8.0) ... Selecting previously unselected package parquet-format.

Preparing to unpack .../parquet-format_2.1.0+cdh5.8.0+12-1.cdh5.8.0.p0.70~trusty-cdh5.8.0_all.deb ...

Unpacking parquet-format (2.1.0+cdh5.8.0+12-1.cdh5.8.0.p0.70~trusty-cdh5.8.0) ... Selecting previously unselected package hadoop-yarn.

Preparing to unpack .../hadoop-yarn_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_all.deb ...

Unpacking hadoop-yarn (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Selecting previously unselected package hadoop-mapreduce.

Preparing to unpack .../hadoop-mapreduce_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_all.deb ...

Unpacking hadoop-mapreduce (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Selecting previously unselected package hadoop-hdfs.

Preparing to unpack .../hadoop-hdfs_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_all.deb ...

Unpacking hadoop-hdfs (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Selecting previously unselected package hadoop-0.20-mapreduce.

Preparing to unpack .../hadoop-0.20-mapreduce_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_amd64.deb ...

Unpacking hadoop-0.20-mapreduce (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ...

Selecting previously unselected package hadoop-client.

Preparing to unpack .../hadoop-client_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_all.deb ...

Unpacking hadoop-client (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Selecting previously unselected package parquet.

Preparing to unpack .../parquet_1.5.0+cdh5.8.0+174-1.cdh5.8.0.p0.71~trusty-cdh5.8.0_all.deb ...
Unpacking parquet (1.5.0+cdh5.8.0+174-1.cdh5.8.0.p0.71~trusty-cdh5.8.0) ...

Selecting previously unselected package hadoop.

Preparing to unpack .../hadoop_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_all.deb

...

Unpacking hadoop (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Selecting previously unselected package libhdfs0.

Preparing to unpack .../libhdfs0_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_amd64.deb ...

Unpacking libhdfs0 (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Selecting previously unselected package hadoop-hdfs-fuse.

Preparing to unpack .../hadoop-hdfs-fuse_2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0_amd64.deb ...

Unpacking hadoop-hdfs-fuse (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Processing triggers for man-db (2.6.7.1-1) ...

Setting up libcurl3:amd64 (7.35.0-1ubuntu2.7) ... Setting up curl (7.35.0-1ubuntu2.7) ...

Setting up avro-libs (1.7.6+cdh5.8.0+112-1.cdh5.8.0.p0.74~trusty-cdh5.8.0) ... Setting up bigtop-utils (0.7.0+cdh5.8.0+0-1.cdh5.8.0.p0.72~trusty-cdh5.8.0) ... Setting up bigtop-jsvc (0.6.0+cdh5.8.0+847-1.cdh5.8.0.p0.74~trusty-cdh5.8.0) ... Setting up zookeeper (3.4.5+cdh5.8.0+94-1.cdh5.8.0.p0.76~trusty-cdh5.8.0) ...

update-alternatives: using /etc/zookeeper/conf.dist to provide /etc/zookeeper/conf (zookeeper-conf) in auto mode

Setting up parquet-format (2.1.0+cdh5.8.0+12-1.cdh5.8.0.p0.70~trusty-cdh5.8.0) ... Setting up parquet (1.5.0+cdh5.8.0+174-1.cdh5.8.0.p0.71~trusty-cdh5.8.0) ...

Setting up hadoop (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ...

update-alternatives: using /etc/hadoop/conf.empty to provide /etc/hadoop/conf (hadoop-conf) in auto mode

Setting up hadoop-yarn (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Setting up libhdfs0 (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ...

Setting up hadoop-mapreduce (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Setting up hadoop-hdfs (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ...

Setting up hadoop-0.20-mapreduce (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Setting up hadoop-client (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ...

Setting up hadoop-hdfs-fuse (2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty-cdh5.8.0) ... Processing triggers for libc-bin (2.19-0ubuntu6) ...

hdpuser@jiju-PC:~\$ sudo mkdir -p /home/hdpuser/hdfs

[sudo] password for hdpuser:

hdpuser@jiju-PC:~\$ sudo hadoop-fuse-dfs dfs://localhost:54310 /home/hdpuser/hdfs/

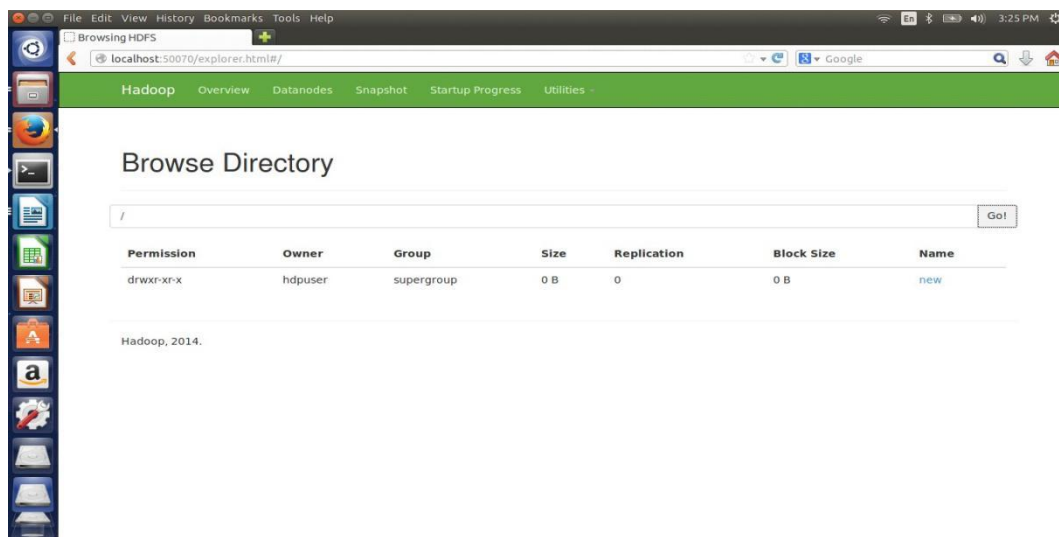
INFO /data/jenkins/workspace/generic-package-ubuntu64-14-04/CDH5.8.0-Packaging-Hadoop-2016-07-12_15-43-10/hadoop-2.6.0+cdh5.8.0+1601-1.cdh5.8.0.p0.93~trusty/hadoop-hdfs-project/hadoop-hdfs/src/main/native/fuse-dfs/fuse_options.c:164 Adding FUSE arg /home/hdpuser/hdfs/

```

hdpuser@jiju-PC:~$ ls /home/hdpuser/hdfs/
hdpuser@jiju-PC:~$ mkdir /home/hdpuser/hdfs/new
hdpuser@jiju-PC:~$ ls /home/hdpuser/hdfs/
new
hdpuser@jiju-PC:~$ mkdir /home/hdpuser/hdfs/example
hdpuser@jiju-PC:~$ ls -l /home/hdpuser/hdfs/total 8

drwxr-xr-x 2 hdpuser 99 4096 Jul 24 15:28 example
drwxr-xr-x 2 hdpuser 99 4096 Jul 24 15:19 new

```



To Unmont the file system

Using umount command the filesystem can be unmounted. `hdpuser@jiju-PC:~$ sudo umount /home/hdpuser/hdfs`

NOTE: You can now add a permanent HDFS mount which persists through reboots.

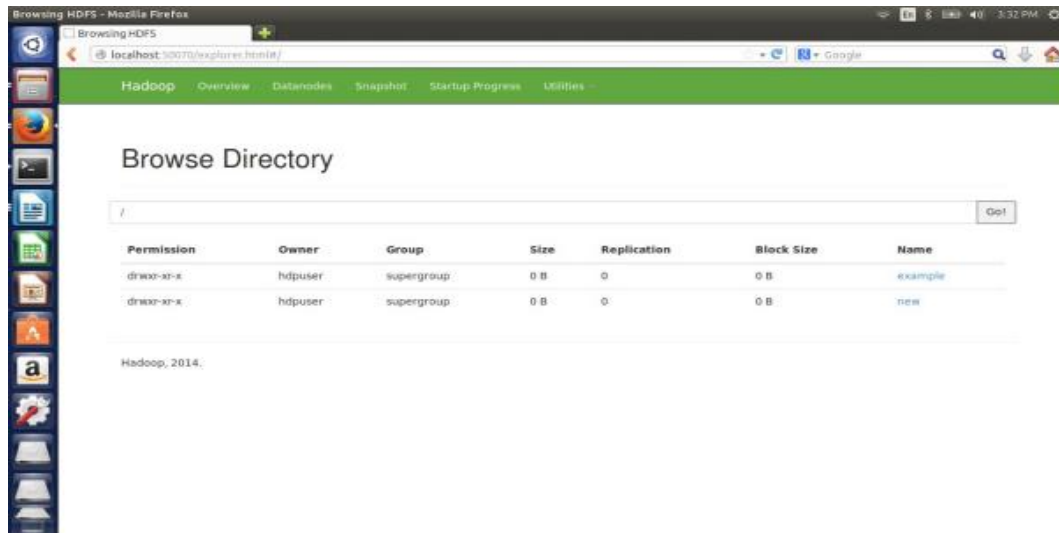
To add a system mount:

Open `/etc/fstab` and add lines to the bottom similar to these: `(sudo vi /etc/fstab)`

```

hadoop-fuse-dfs#dfs://<name_node_hostname>:<namenode_port><mount_point>      fuse
allow_other,usetrash,rw 2 0

```



For example:

Sudo hadoop-fuse-dfs#dfs://localhost:54310/home/hdpuser/hdfs

Test to make sure everything is working properly:

\$ mount <mount_point>

hdpuser@jiju-PC:~\$ **sudo mount /home/hdpuser/hdfs**

Result:

Thus fuse has been installed successfully.

EX NO :8 . WRITE A WORD COUNT PROGRAM TO DEMONSTRATE THE USE OF MAP AND REDUCE TASKS

AIM:

To write a java program to count number of words in a file using map reduce concept.

PROCEDURE:

1. Open eclipse
2. File> new> java project
3. Libraries>add external jars..
4. File system>usr> lib>hadoop> select all jar files
5. Click ok
6. Again add external jars.
7. File system >client>select all jar files >click ok
8. Click finish
9. Rightclickwordcount> new> class
10. Wordcount>src> default package > wordcount.java
11. Run the wordcount java program
12. Rightclickwordcount> export
13. Open command prompt

PROGRAM :

```
import java.io.IOException; import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;

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.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordCount {
public static class TokenizerMapper
```

```

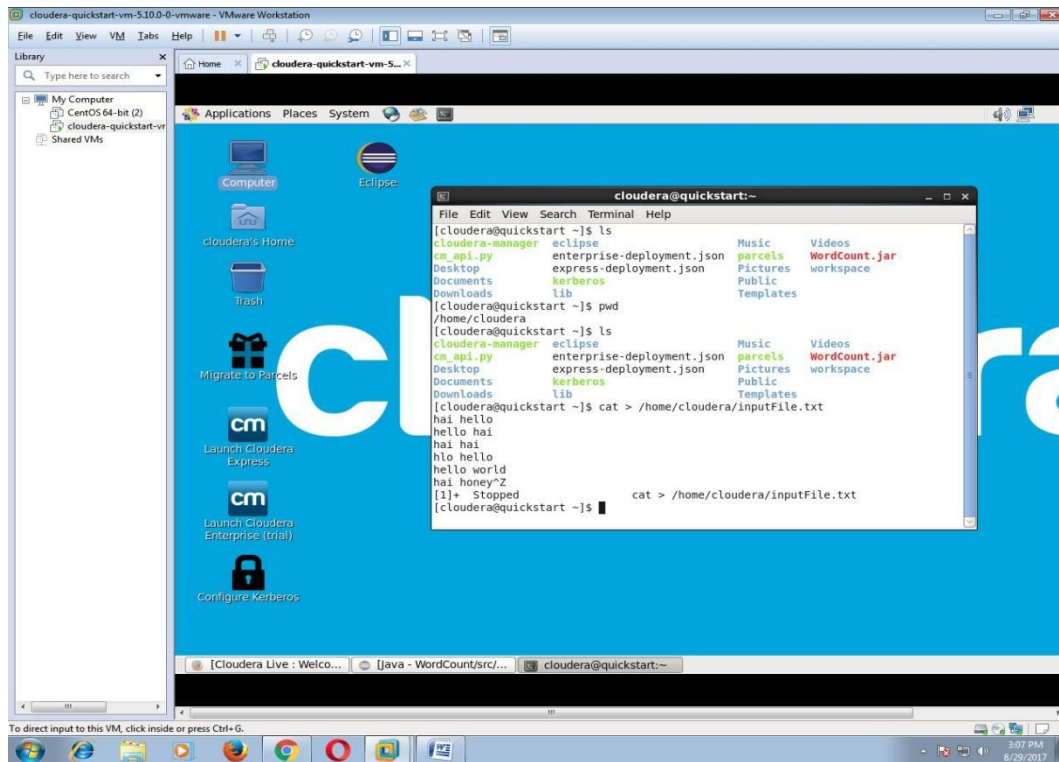
extends Mapper<Object, Text, Text, IntWritable>{
private final static IntWritable one = new IntWritable(1); private Text word = new Text();
public void map(Object key, Text value, Context context
) throws IOException, InterruptedException { StringTokenizer itr = new
StringTokenizer(value.toString());
while (itr.hasMoreTokens()) { word.set(itr.nextToken()); context.write(word, one);
}
}
}

public static class IntSumReducer
extends Reducer<Text,IntWritable,Text,IntWritable> { private IntWritable result = new
IntWritable();
public void reduce(Text key, Iterable<IntWritable> values,
Context context
) throws IOException, InterruptedException { int sum = 0;
for (IntWritable val : values) { sum += val.get();
}
result.set(sum); context.write(key, result);
}
}

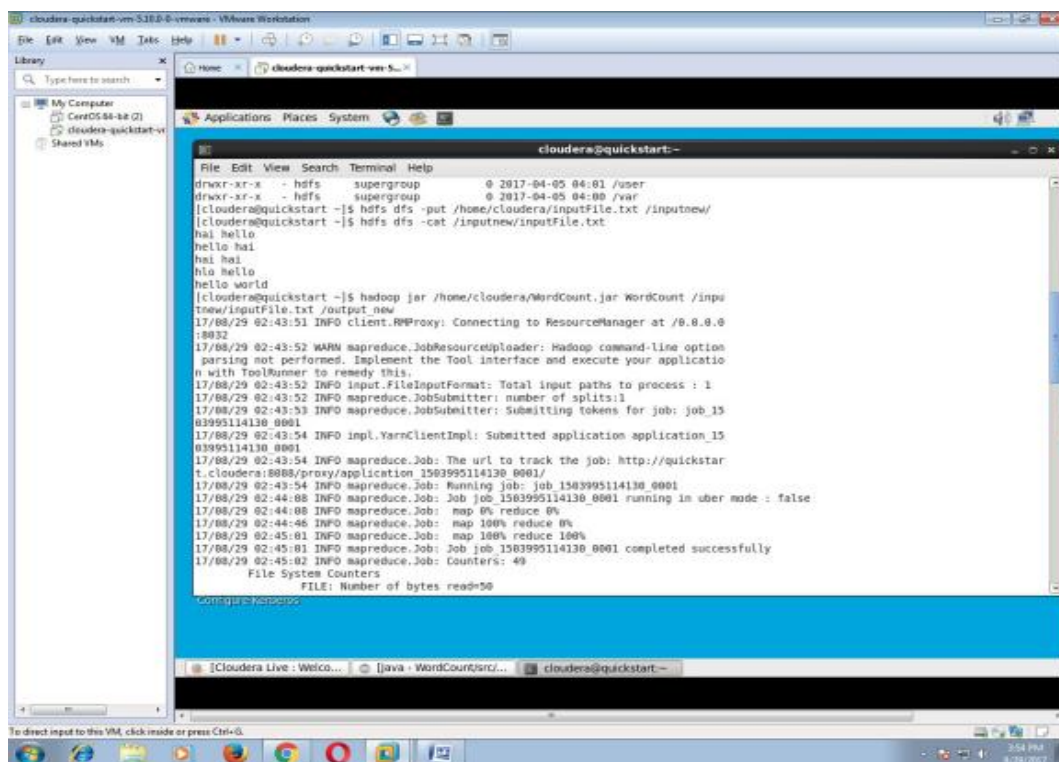
public static void main(String[] args) throws Exception
{ Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word count");
job.setJarByClass(WordCount.class);
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

```

OUTPUT:



```
cloudera@quickstart:~$ ls
cloudera-manager  eclipse
cm_api.py         enterprise-deployment.json  Music      Videos
Desktop          express-deployment.json   parcels    WordCount.jar
Documents        kerberos                 Pictures   workspace
Downloads        lib                      Public     Templates
[cloudera@quickstart ~]$ pwd
/home/cloudera
[cloudera@quickstart ~]$ ls
cloudera-manager  eclipse
cm_api.py         enterprise-deployment.json  Music      Videos
Desktop          express-deployment.json   parcels    WordCount.jar
Documents        kerberos                 Pictures   workspace
Downloads        lib                      Public     Templates
[cloudera@quickstart ~]$ cat > /home/cloudera/inputFile.txt
hai hello
hai hai
hello hai
hai hai
hlo hello
hello world
hai honeyZ
[!] Stopped
[cloudera@quickstart ~]$ cat > /home/cloudera/inputFile.txt
```



```
drwxr-xr-x - hdfs supergroup 0 2017-04-05 04:01 /user
drwxr-xr-x - hdfs supergroup 0 2017-04-05 04:00 /var
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/inputFile.txt /inputnew/
[cloudera@quickstart ~]$ hdfs dfs -cat /inputnew/inputFile.txt
hai hello
hai hai
hello hai
hai hai
hlo hello
hello world
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount /inputnew/inputFile.txt /output_new
17/08/29 02:43:51 INFO client.RMProxy: Connecting to ResourceManager at /6.8.8.0:8032
17/08/29 02:43:52 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
17/08/29 02:43:52 INFO Input.FileInputFormat: Total input paths to process : 1
17/08/29 02:43:52 INFO mapreduce.JobSubmitter: number of splits:1
17/08/29 02:43:53 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1503995114130_0001
17/08/29 02:43:54 INFO impl.YarnClientImpl: Submitted application application_1503995114130_0001
17/08/29 02:43:54 INFO mapreduce.Job: The url to track the job: http://quickstart1.cloudera:8088/proxy/application_1503995114130_0001/
17/08/29 02:43:54 INFO mapreduce.Job: Running job: job_1503995114130_0001
17/08/29 02:44:08 INFO mapreduce.Job: Job job_1503995114130_0001 running in user mode = false
17/08/29 02:44:08 INFO mapreduce.Job: map 0% reduce 0%
17/08/29 02:44:08 INFO mapreduce.Job: map 100% reduce 0%
17/08/29 02:45:01 INFO mapreduce.Job: map 100% reduce 100%
17/08/29 02:45:01 INFO mapreduce.Job: Job job_1503995114130_0001 completed successfully
17/08/29 02:45:02 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=50
```

```

cloudera-quickstart-vm-5182-0-vmware - VMware Workstation
File Edit View VM Tools Help
Library
Type here to search
My Computer
CereOS-64-bit (2)
cloudera-quickstart-vm-5182-0
Shared VMs

Applications Places System

cloudera@quickstart:~$
File Edit View Search Terminal Help
17/08/29 02:45:01 INFO mapreduce.Job: map 100% reduce 100%
17/08/29 02:45:01 INFO mapreduce.Job: Job job_1503995114130_0001 completed successfully
17/08/29 02:45:02 INFO mapreduce.Job: Counters: 49
File System Counters
  FILE: Number of bytes read=50
  FILE: Number of bytes written=242389
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=169
  HDFS: Number of bytes written=28
  HDFS: Number of read operations=6
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=34006
  Total time spent by all reduces in occupied slots (ms)=11374
  Total time spent by all map tasks (ms)=34006
  Total time spent by all reduce tasks (ms)=11374
  Total vcore-seconds taken by all map tasks=34006
  Total vcore-seconds taken by all reduce tasks=11374
  Total megabyte-seconds taken by all map tasks=35641344
  Total megabyte-seconds taken by all reduce tasks=11646978
Map-Reduce Framework
  Map input records=5
  Map output records=10
  Map output bytes=90
  Map output materialized bytes=50
  Input split bytes=119
  Combine input records=10
  Combine input records=10
  Combine output records=4
  Reduce input groups=4
  Reduce shuffle bytes=50
  Reduce input records=4
  Reduce output records=4
  Spilled Records=0
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=278
  CPU time spent (ms)=1950
  Physical memory (bytes) snapshot=328749056
  Virtual memory (bytes) snapshot=3007586304
  Total committed heap usage (bytes)=226627384
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=50
File Output Format Counters
  Bytes Written=28
  Longitude=NotSet

[Cloudera Live: Welco... [Java - WordCount/src/... cloudera@quickstart:~$

```

```

cloudera-quickstart-vm-5182-0-vmware - VMware Workstation
File Edit View VM Tools Help
Library
Type here to search
My Computer
CereOS-64-bit (2)
cloudera-quickstart-vm-5182-0
Shared VMs

Applications Places System

cloudera@quickstart:~$
File Edit View Search Terminal Help
Total megabyte-seconds taken by all reduce tasks=11646978
Map-Reduce Framework
  Map input records=5
  Map output records=10
  Map output bytes=90
  Map output materialized bytes=50
  Input split bytes=119
  Combine input records=10
  Combine output records=4
  Reduce input groups=4
  Reduce shuffle bytes=50
  Reduce input records=4
  Reduce output records=4
  Spilled Records=0
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=278
  CPU time spent (ms)=1950
  Physical memory (bytes) snapshot=328749056
  Virtual memory (bytes) snapshot=3007586304
  Total committed heap usage (bytes)=226627384
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=50
File Output Format Counters
  Bytes Written=28
  Longitude=NotSet

[Cloudera Live: Welco... [Java - WordCount/src/... cloudera@quickstart:~$

```


PROJECT REPORT

Employee Management System

Submitted by

AARON S (95072112002)

ABDUL AJIZ S.S (95072112003)

DEEPESH PATEL (95072112022)

GURU M (95072112032)

INDEX

| S.No | Date | List of Contents | Page | Marks | Sign |
|------|------|----------------------|------|-------|------|
| 1 | | Abstract | 40 | | |
| 2 | | Introduction | 41 | | |
| 3 | | System configuration | 42 | | |
| 4 | | Architecture Diagram | 43 | | |
| 5 | | Module description | 44 | | |
| 6 | | Sample code | 45 | | |
| 7 | | Output Screenshots | 48 | | |
| 8 | | Conclusion | 50 | | |

Abstract

The Employee Management System is a cloud-based application designed to streamline employee-related processes within organizations. Built with React.js for the frontend and AWS services for the backend, it leverages a serverless architecture to enhance scalability and performance. AWS SQS handles messaging, SES manages email notifications, and Lambda functions (written in Python) process backend logic, all orchestrated through API Gateway. AWS Amplify is used for hosting, ensuring secure, high-availability access. MongoDB serves as the database, storing employee records efficiently. This system supports essential functionalities such as employee record management, notification handling, and payroll processes, demonstrating effective use of modern cloud technologies to deliver reliable, scalable, and secure solutions.

Introduction

The Employee Management System is a cloud-based application developed to streamline and automate essential employee-related processes, enhancing efficiency and reducing manual workload within organizations. This project employs a serverless architecture, utilizing a combination of AWS services and modern web development technologies to ensure high scalability, performance, and security. The frontend is developed using React.js, providing a dynamic, user-friendly interface that allows users to manage employee information, notifications, and payroll processes effectively.

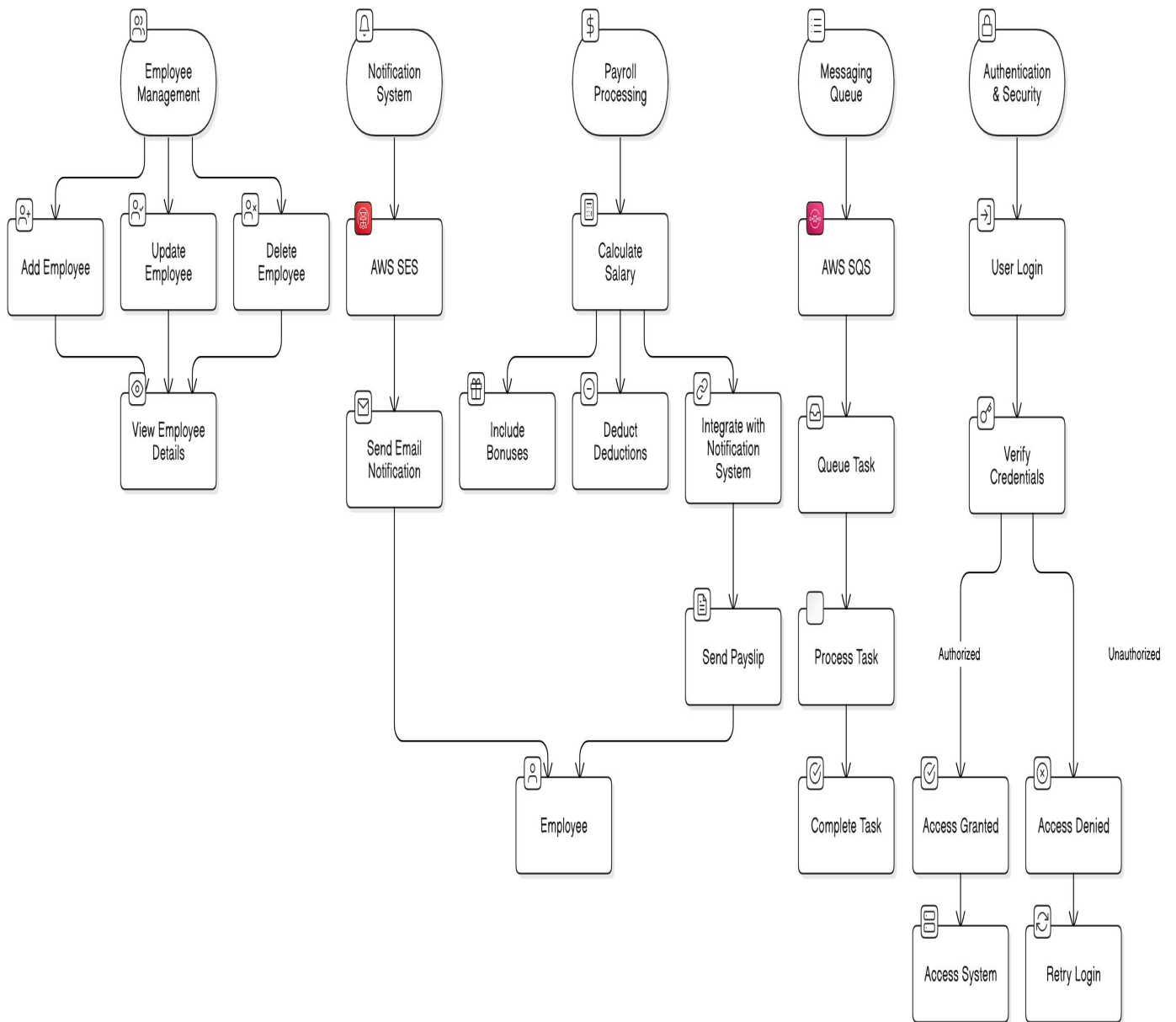
On the backend, the application integrates multiple AWS services: AWS Simple Queue Service (SQS) for efficient message queuing, Simple Email Service (SES) for automated email communication, and Lambda functions written in Python for executing backend logic, all orchestrated through AWS API Gateway to manage secure API requests. MongoDB is chosen as the database, offering a flexible and scalable structure to store employee records and other data.

For hosting and deployment, AWS Amplify is utilized, ensuring continuous integration and secure access, which contributes to the system's high availability and reliability. This Employee Management System not only demonstrates the benefits of serverless architecture—such as reduced operational overhead, automatic scaling, and cost efficiency—but also showcases the synergy of cloud services and modern front-end technologies to build robust, scalable solutions for real-world organizational needs.

Hardware-Software Requirements

- a) Operating System : Windows (Windows 7/ 8/ 10/ 11).
- b) Programming Tool : React JS, AWS Services, Mongo DB.
- c) IDE : VS Code
- d) Processor : intel i3 and higher versions.
- e) RAM : 8GB
- f) Storage : 200 GB Hard Disk

Architecture Diagram



Module Description

1. Employee Management

This module allows administrators to add, update, and delete employee records. It provides an organized view of employee details, improving data accessibility and enabling efficient employee information management.

2. Notification System

Utilizes AWS SES for automated email notifications to employees, such as for payslips or updates. It ensures timely and reliable communication within the organization through scheduled and triggered messages.

3. Payroll Processing

Handles employee salary calculations, including bonuses and deductions. This module integrates with the notification system to send payslips, simplifying payroll tasks and improving financial accuracy and transparency.

4. Messaging Queue

Uses AWS SQS to manage task queuing for asynchronous operations. It ensures efficient handling of background tasks like data processing, improving application performance by distributing workload effectively.

5. Authentication & Security

Implements secure user authentication and access control, ensuring only authorized personnel access the system. This module prioritizes data protection, enhancing overall security and compliance within the application.

Sample Code

```
import React, { useState, useEffect } from 'react';
import { BrowserRouter as Router, Routes, Route, Link, Navigate, useLocation } from 'react-router-dom';
import Home from './pages/Home';
import Login from './pages/Login';
import SendPayslip from './pages/Sendpayslip';
import Addemployee from './pages/Addemployee';
import './App.css';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faArrowLeft, faBars, faTimes } from '@fortawesome/free-solid-svg-icons';

const Navbar = ({ isLoggedIn, setIsLoggedIn, setModalIsOpen }) => {
  const [isNavOpen, setIsNavOpen] = useState(false); // New state for toggling navbar
  const location = useLocation();

  const handleLogout = () => {
    setIsLoggedIn(false);
    localStorage.removeItem('isLoggedIn');
    window.location.href = '/';
  };

  const toggleNav = () => {
    setIsNavOpen(!isNavOpen);
  };

  const closeNav = () => {
    setIsNavOpen(false);
  };

  return (
    <nav className="navbar">
      <div className="logo-place">
        <Link to="/" className="logo-link" onClick={closeNav}>
          
        </Link>
      </div>
      <div className={`nav-links ${isNavOpen ? 'open' : ''}`>
        {isLoggedIn && location.pathname !== "/login" && (
          <Link to="/login" className="nav-link btn-prominent"
onClick={closeNav}>Login</Link>
        )}
        {isLoggedIn ? (
          <div>
            {location.pathname === "/add-employee" ? (
              <button onClick={() => { setModalIsOpen(true); closeNav(); }}
className="nav-link btn-prominent">Add Employee</button>
            ) : (
              <Link to="/add-employee" className="nav-link btn-prominent"
onClick={closeNav}>Add Employee</Link>
            )}
            <button onClick={() => { handleLogout(); closeNav(); }}

```



```

className="nav-link btn-prominent">Logout</button>
      </>
    ) : (
      location.pathname === "/login" && (
        <Link to="/" className="nav-link btn-prominent"
onClick={closeNav}>
          <FontAwesomeIcon icon={faArrowLeft} /> Back
        </Link>
      )
    )}
  </div>
  <div className="nav-toggle" onClick={toggleNav}>
    <FontAwesomeIcon icon={isNavOpen ? faTimes : faBars} />
  </div>
</nav>
);
};

const App = () => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const [modalsOpen, setModalsOpen] = useState(false);
  const [employees, setEmployees] = useState([]);
  const [isSendPayslipEnabled, setIsSendPayslipEnabled] = useState(false);

  useEffect(() => {
    const loggedInStatus = localStorage.getItem('isLoggedIn');
    if (loggedInStatus === 'true') {
      setIsLoggedIn(true);
    }
  }, []);

  const handleLogin = () => {
    setIsLoggedIn(true);
    localStorage.setItem('isLoggedIn', 'true');
  };

  const handleAddEmployee = (newEmployee) => {
    setEmployees(prevEmployees => [...prevEmployees, newEmployee]);
    setModalsOpen(false); // Close the modal after adding an employee
  };

  const handleUpdateEmployee = (updatedEmployee) => {
    setEmployees(prevEmployees =>
      prevEmployees.map(emp =>
        emp.email === updatedEmployee.email ? updatedEmployee : emp
      )
    );
  };

  const handleDeleteEmployee = (employeeToDelete) => {
    setEmployees(prevEmployees =>
      prevEmployees.filter(emp => emp.email !== employeeToDelete.email)
    );
  };
};

```

```

const handleCheckboxChange = (selectedEmployees) => {
  setIsSendPayslipEnabled(selectedEmployees.length > 0);
};

return (
  <Router>
    <Navbar
      isLoggedIn={isLoggedIn}
      setIsLoggedIn={setIsLoggedIn}
      setModalsOpen={setModalsOpen}
      isSendPayslipEnabled={isSendPayslipEnabled}
    />
    <Routes>
      <Route path="/" element={isLoggedIn ? <Navigate to="/add-employee" /> :
<Home isLoggedIn={isLoggedIn} />} />
      <Route path="/send-payslip" element={isLoggedIn ? <SendPayslip /> :
<Navigate to="/login" />} />
      <Route
        path="/add-employee"
        element={
          isLoggedIn ? (
            <Addemployee
              employees={employees}
              onAddEmployee={handleAddEmployee}
              onUpdateEmployee={handleUpdateEmployee}
              onDeleteEmployee={handleDeleteEmployee}
              modalsOpen={modalsOpen}
              setModalsOpen={setModalsOpen}
              onCheckboxChange={handleCheckboxChange}
            />
          ) : (
            <Navigate to="/login" />
          )
        }
      />
      <Route
        path="/login"
        element={
          isLoggedIn ? <Navigate to="/add-employee" /> : <Login
onLogin={handleLogin} />
        }
      />
    </Routes>
  </Router>
);
};

export default App;

```

Output Screenshots

AAGTECH

LOGIN

Sending Payslip amplified

Automate Sending Payslip with ease to every workers in the company

GET STARTED

Dashboard

Academic

Applied Labs

Relationship Details

Placement Details

Fee payment

COE

Feedback

Skill

Dashboard

NA

0

Earned Reward Points

0

Penalty

0

Current Reward Points

Class

07/11/2024

Thursday

November 7, 2024

Big Data Analytics

Alameia Mangal

Timing: 09:00 AM - 09:50 AM

Cryptography And Network Security

Jander P

Timing: 09:50 AM - 10:40 AM

85% Faster Sending

AAGTECH

← BACK

Login

aagtech@gmail.com

Login

Hello, Friend!

Enter your details and continue your journey with us.

AAG

TECH

ADD EMPLOYEE

LOGOUT

Employee List

☐

Name: Aaron S

Email: aaronsimon2612@gmail.com

Position: Software Developer

Salary: 95000

☐

Name: Allen

Email: fuegoallen333@gmail.com

Position: Software Developer

Salary: 85000

☐

Name: Abdul Ajiz

Email: ajiz123@gmail.com

Position: Software Developer

Name:

Enter employee name

Email:

Enter employee email

Position:

Enter employee position

Salary:

Enter employee salary

Add Employee

AAG

TECH

ADD EMPLOYEE

LOGOUT

Employee List

SEND

☐

Name: Aaron S

Email: aaronsimon2612@gmail.com

Position: Software Developer

Salary: 95000

☐

Name: Allen

Email: fuegoallen333@gmail.com

Position: Software Developer

Salary: 85000

☐

Name: Abdul Ajiz

Email: ajiz123@gmail.com

Position: Software Developer

Conclusion

The Employee Management System developed for this project demonstrates the effective use of cloud computing and serverless architecture in fulfilling organizational needs for efficient employee data management. By integrating a React.js frontend with AWS services such as Lambda, SQS, SES, and API Gateway, the system provides a scalable, secure, and efficient solution for handling employee records, payroll, and notifications. MongoDB offers flexible data storage, while AWS Amplify ensures high availability and seamless deployment.

This system showcases the advantages of serverless computing—automatic scaling, minimal operational overhead, and cost efficiency—making it suitable for organizations of various sizes. Automated payroll and notification modules improve productivity by reducing manual tasks and ensuring timely communication with employees.

Overall, the Employee Management System highlights how modern cloud technologies can drive operational efficiency, demonstrating a practical and adaptable approach to meeting business needs through scalable, robust, and secure cloud-based applications.