



SAVEETHA SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CSA15 – CLOUD COMPUTING AND BIG DATA ANALYTICS
LABORATORY**

INDEX

| S.N O | DATE | EXPERIMENT NAME | MARK S | SIGN |
|----------|------|---|-----------|------|
| 1. | | Create a simple cloud software application and provide it as a service using any Cloud Service Provider to demonstrate Software as a Service (SaaS). | | |
| 2. | | Create a Virtual Machine with 1 vCPU, 2GB RAM and 15GB storage disk using a Type 2 Virtualization Software | | |
| 3. | | Create a Virtual Hard Disk and allocate the storage using VM ware Workstation | | |
| 4. | | Create a Snapshot and Cloning of a VM and Test it by loading the Previous Version/Cloned VM | | |
| 5. | | Demonstrate Infrastructure as a Service (IaaS) by Creating a Virtual Machine using a Public Cloud Service Provider (Azure/GCP/AWS), configure with minimum CPU, RAM, and Storage and Launch the VM image. | | |
| 6. | | Create a Simple Web Application using Java or Python and host it in any Public Cloud Service Provider (Azure/GCP/AWS) to demonstrate Platform as a Service (PaaS) | | |
| 7. | | Create a Storage service using any Public Cloud Service Provider (Azure/GCP/AWS) and check the public accessibility of the stored file to demonstrate Storage as a Service | | |
| 8. | | Create a SQL storage service and perform a basic query using any Public Cloud Service Provider (Azure/GCP/AWS) to demonstrate Database as a Service (DaaS) | | |
| 9. | | Perform the basic configuration setup for Installing Hadoop 2.x like Creating the HDUSER and SSH localhost | | |
| 10. | | Install Hadoop 2.x and configure the Name Node and Data Node. | | |
| 11. | | Launch the Hadoop 2.x and perform MapReduce Program for a Word Count problem | | |

EXP NO 1: CREATE A SIMPLE CLOUD SOFTWARE APPLICATION AND PROVIDE IT AS A SERVICE USING ANY CLOUD SERVICE PROVIDER TO DEMONSTRATE SOFTWARE AS A SERVICE (SAAS).

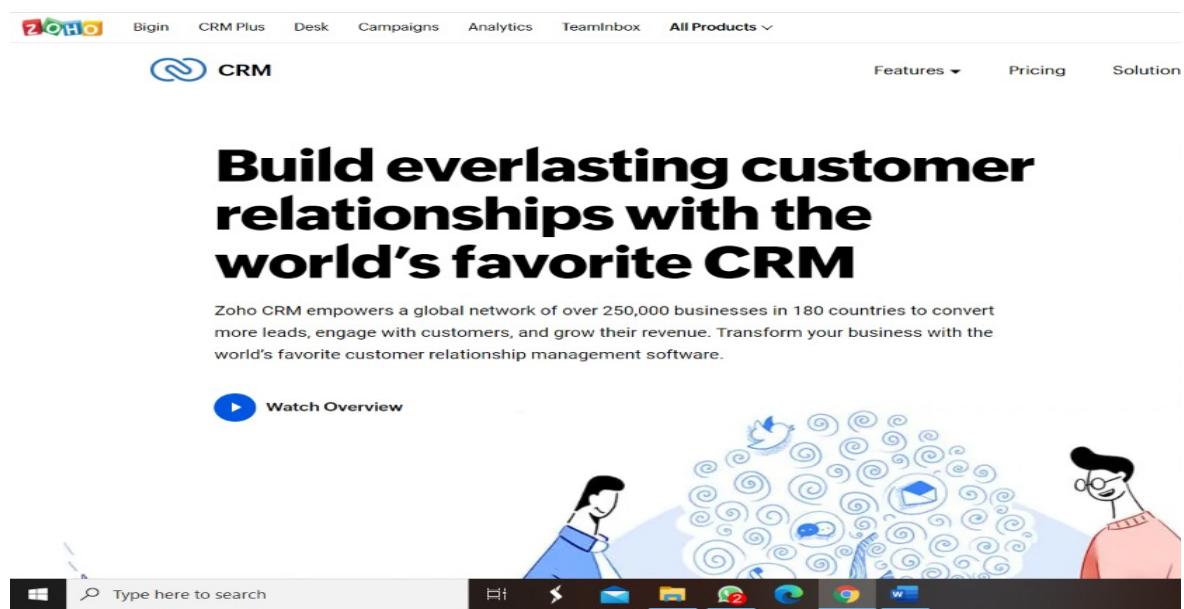
DATE:

AIM:

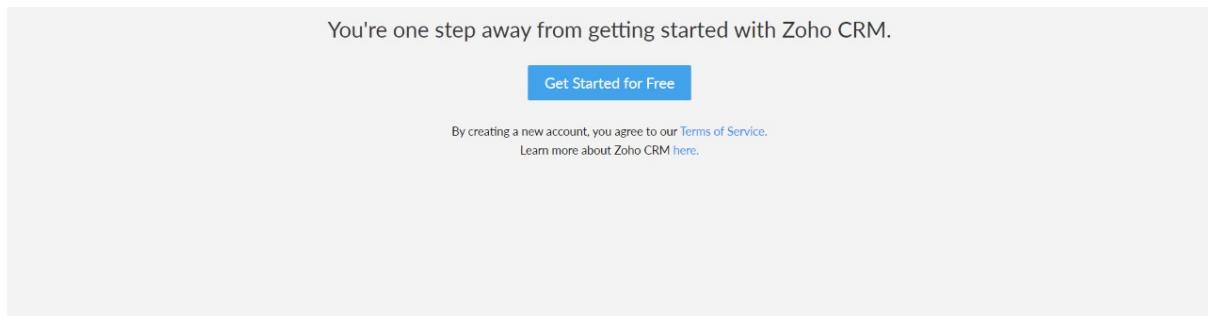
PROCEDURE:

IMPLEMENTATION:

STEP1: GOTO ZOHO.COM



STEP 2: LOGINTO THE ZOHO.COM



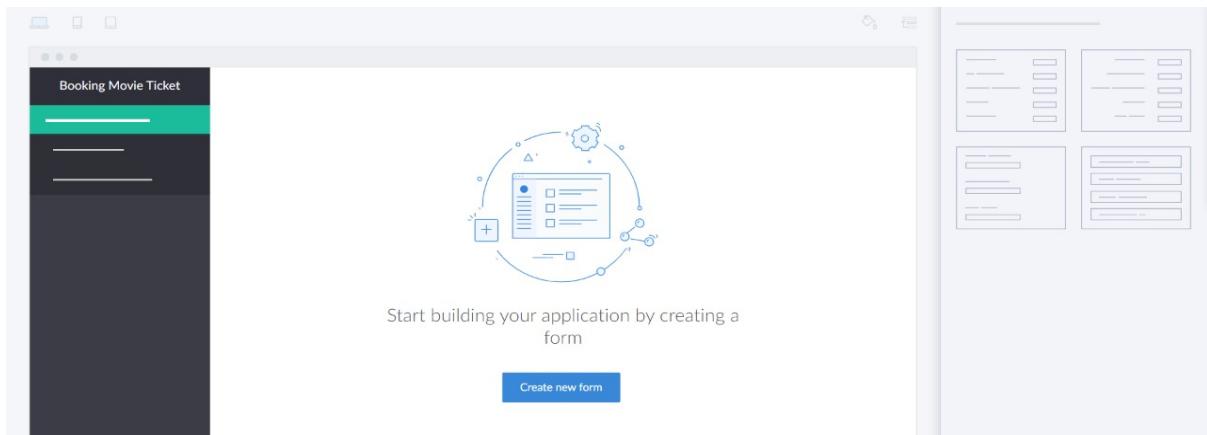
STEP 3: SELECT ONE APPLICATION

A screenshot of the "Create Application" page. At the top left is a search bar with placeholder text "Start typing to search for applications...". To the right are filter buttons for "All", "Information Technology", "Business", "Sales & Marketing", "Educ...", and "More". Below the search bar are eight application cards arranged in two rows of four. The first card, "Create from scratch", is highlighted with a dashed border. The other seven cards represent different applications: "Sales Management", "Order Management", "Employee Management", "IT Asset Tracker", "Event Management", "Course Planner", and "Expenses". Each card has a small icon, a title, and a brief description.

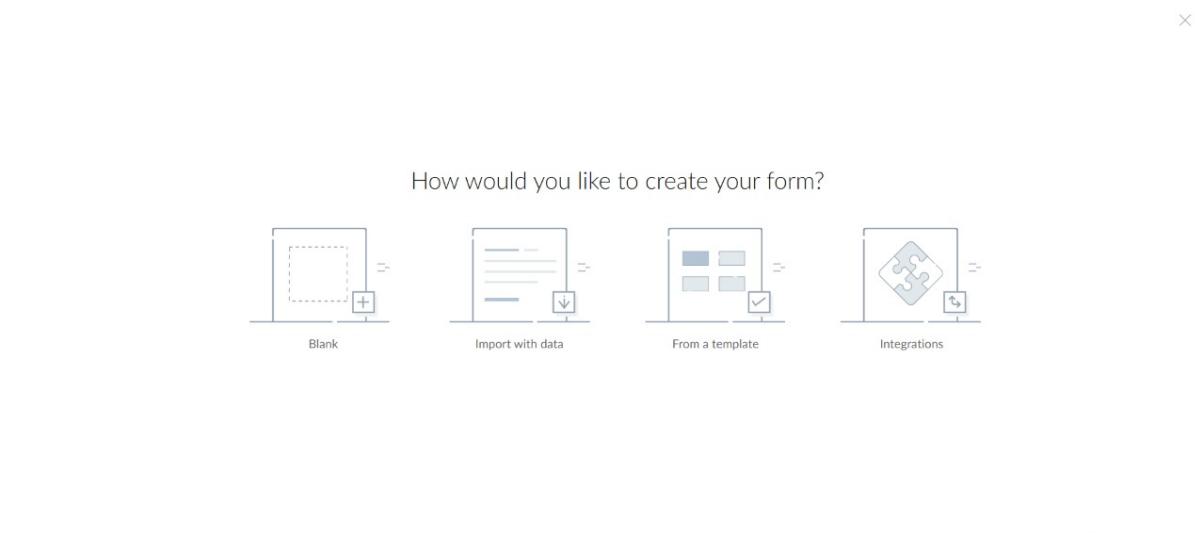
STEP 4: ENTER APPLICATION NAME

A screenshot of the "Create Application" page, similar to the previous one but with a modal dialog open. The dialog is titled "Enter Application Name" and contains a text input field with placeholder text "Examples: Campaign Monitor, Order Management" and a blue "Create" button below it. The rest of the page is dimmed, and the application cards are grayed out, indicating they are not currently selectable while the dialog is open.

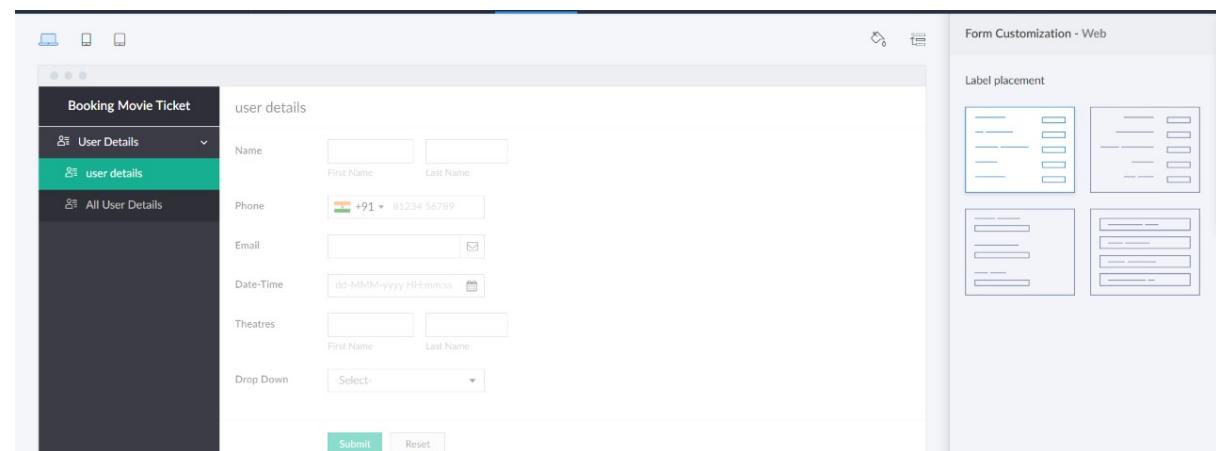
STEP 5: CREATED NEW APPLICATION



STEP 6: SELECT ONE FORM



STEP 7: THE SOFTWARE HASE BEEN CREATED.



Booking Movie Ticket
user details

Basic Fields

| | | | |
|--|-------------|--|------------|
| | Name | | Email |
| | Address | | Phone |
| | Single Line | | Multi Line |
| | Number | | Date |
| | Time | | Drop Down |

Field Properties

Field name: Name

Field link name: Name

Validation: Mandatory

Display Fields

Prefix: First Name

Last Name: Last Name

Suffix: Suffix

Data Privacy

Done

RESULT:

EXPNO 2: CREATE A VIRTUAL MACHINE WITH 1 VCPU, 2GB RAM AND 15GB STORAGE DISK USING A TYPE 2 VIRTUALIZATION SOFTWARE

DATE:

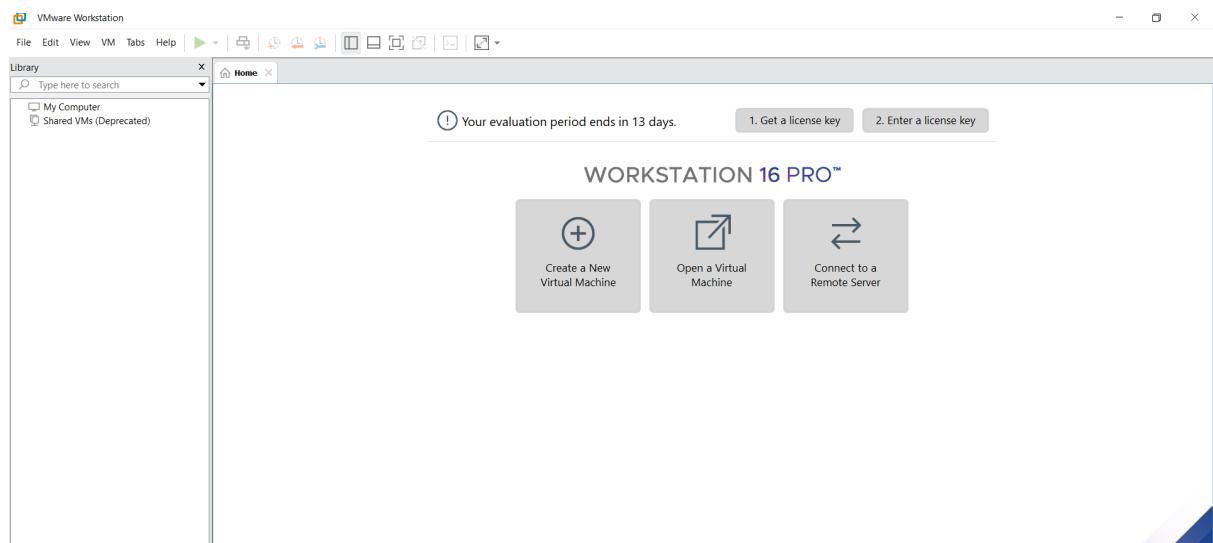
AIM:

PROCEDURE:

IMPLEMENTATION:

STEP 1:

DOWLOAD VMWARE WORKSTATION AND INSTALLED AS TYPE 2 HYPERVISOR

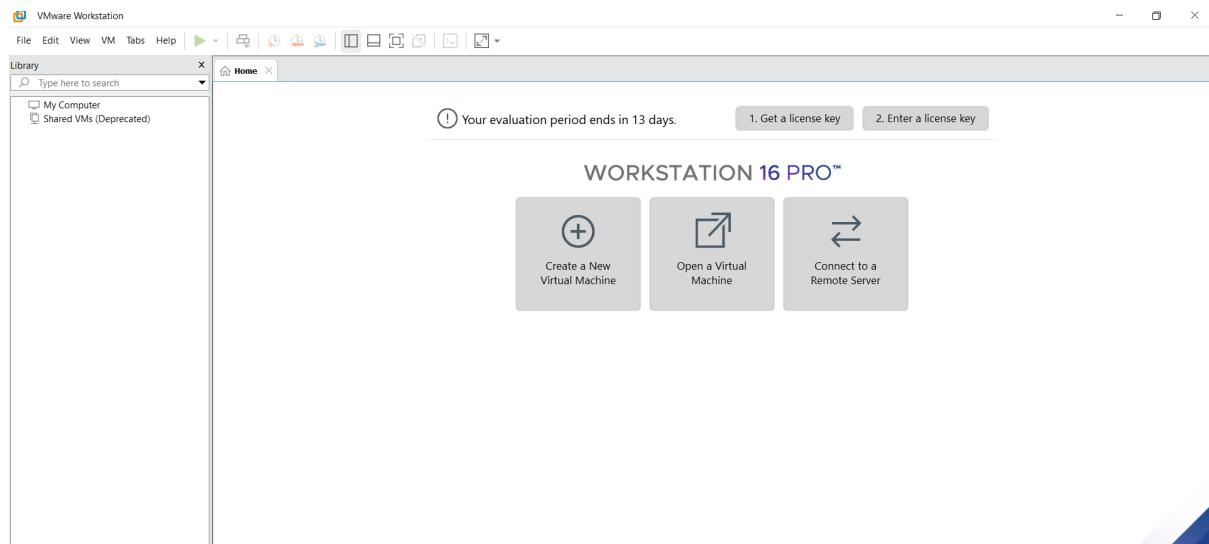


STEP2: DOWNLOAD UBUNTU OR TINY OS AS ISO IMAGE FILE

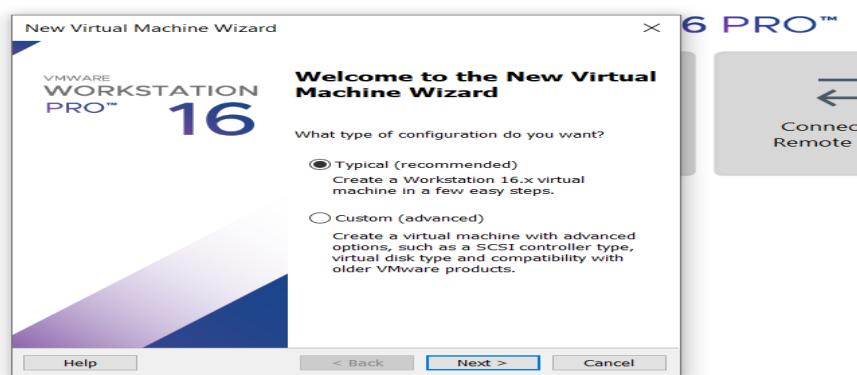
Index of /11.x/x86/release/

| | |
|---------------------------|-----------------------------|
| .. | |
| distribution_files/ | |
| src/ | |
| Core-11.1.iso | 09-Feb-2020 11:50 - |
| Core-11.1.iso.md5.txt | 03-Dec-2019 11:14 - |
| Core-11.1.iso.zsync | 01-Apr-2020 07:49 14757888 |
| Core-current.iso | 01-Apr-2020 07:49 48 |
| CorePlus-11.1.iso | 01-Apr-2020 07:49 50639 |
| CorePlus-11.1.iso.md5.txt | 01-Apr-2020 07:50 14757888 |
| CorePlus-11.1.iso.zsync | 01-Apr-2020 07:50 216006656 |
| CorePlus-current.iso | 01-Apr-2020 07:50 52 |
| TinyCore-11.1.iso | 01-Apr-2020 07:50 369358 |
| TinyCore-11.1.iso.md5.txt | 01-Apr-2020 07:50 216006656 |
| TinyCore-11.1.iso.zsync | 01-Apr-2020 07:50 19922944 |
| TinyCore-current.iso | 01-Apr-2020 07:50 52 |
| | 01-Apr-2020 07:50 68301 |
| | 01-Apr-2020 07:50 19922944 |

STEP 3: IN VMWARE WORKSTATION->CREATE NEW VM



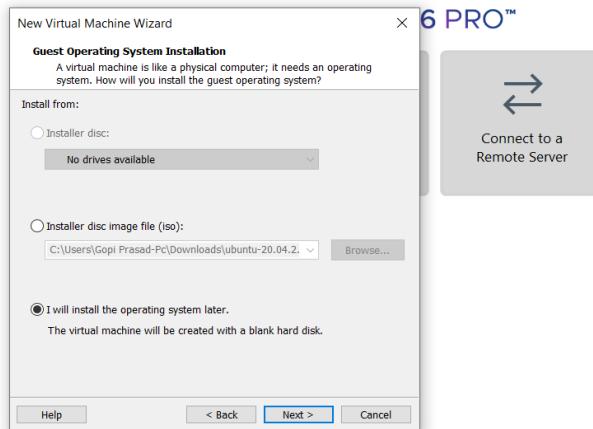
STEP 4: DO THE BASIC CONFIGURATION SETTINGS.



! Your evaluation period ends in 12 days.

1. Get a license key

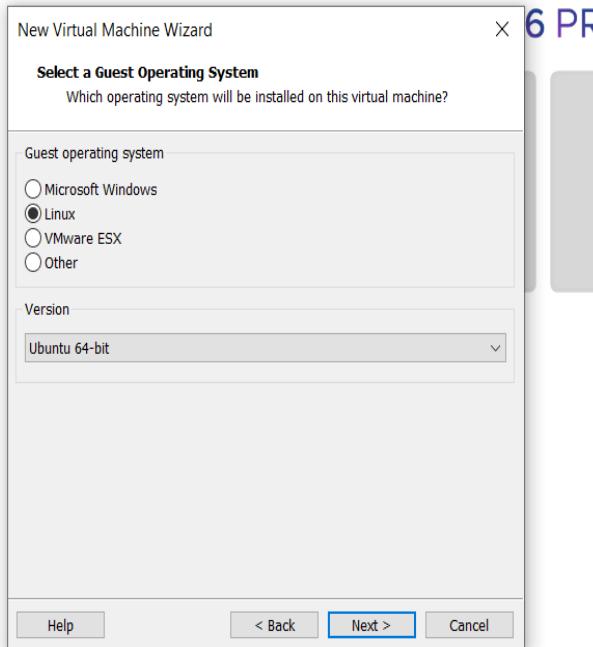
2. Enter a license key



6 PRO™



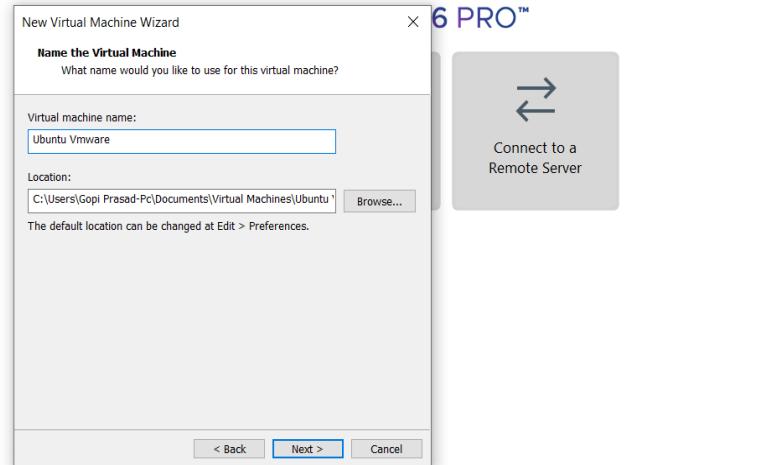
Connect to a
Remote Server



6 PRO™



Connect to a
Remote Server

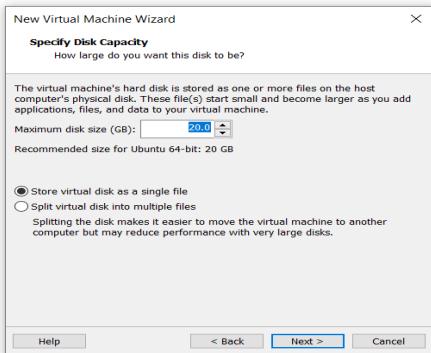


6 PRO™



Connect to a
Remote Server

! Your evaluation period ends in 12 days. [1. Get a license key](#) [2. Enter a license key](#)

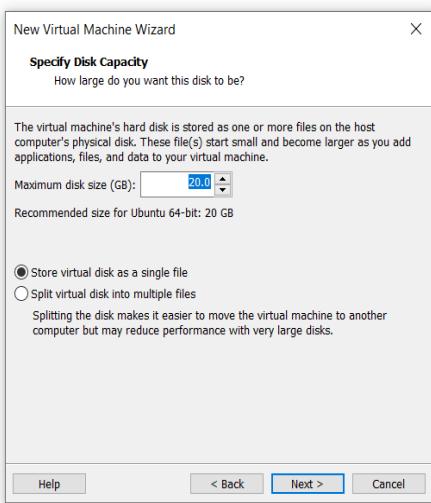


6 PRO™



Connect to a
Remote Server

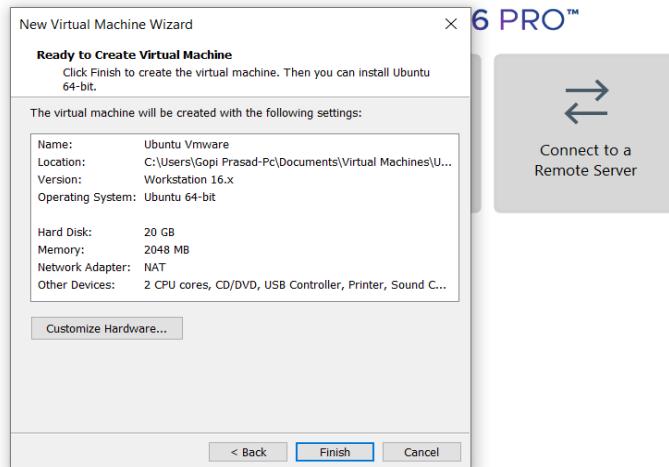
! Your evaluation period ends in 12 days. [1. Get a license key](#) [2. Enter a license key](#)



6 PRO™



Connect to a
Remote Server

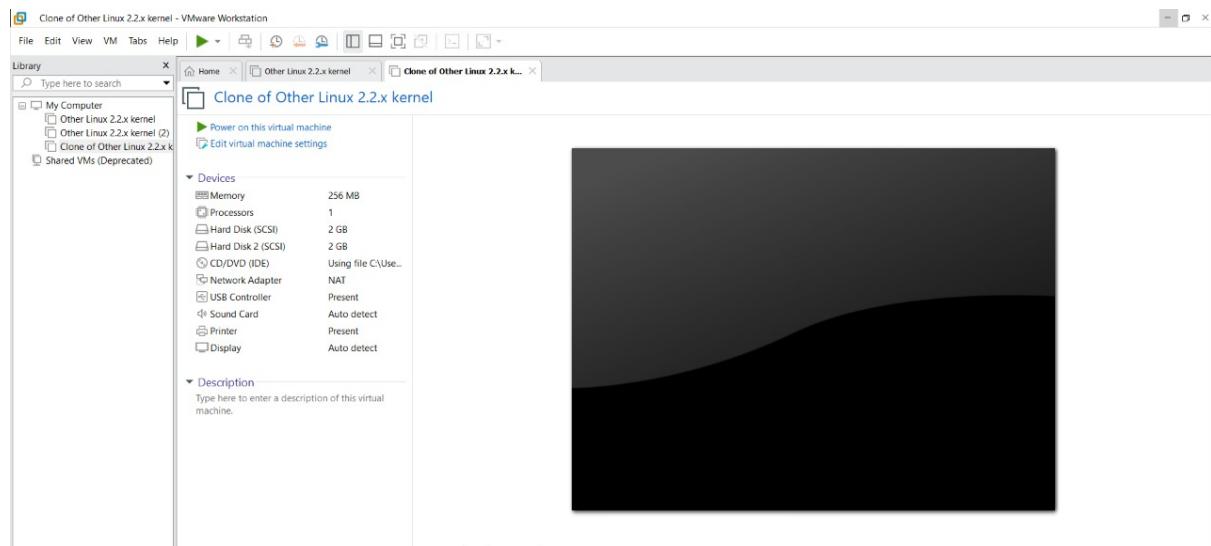


6 PRO™

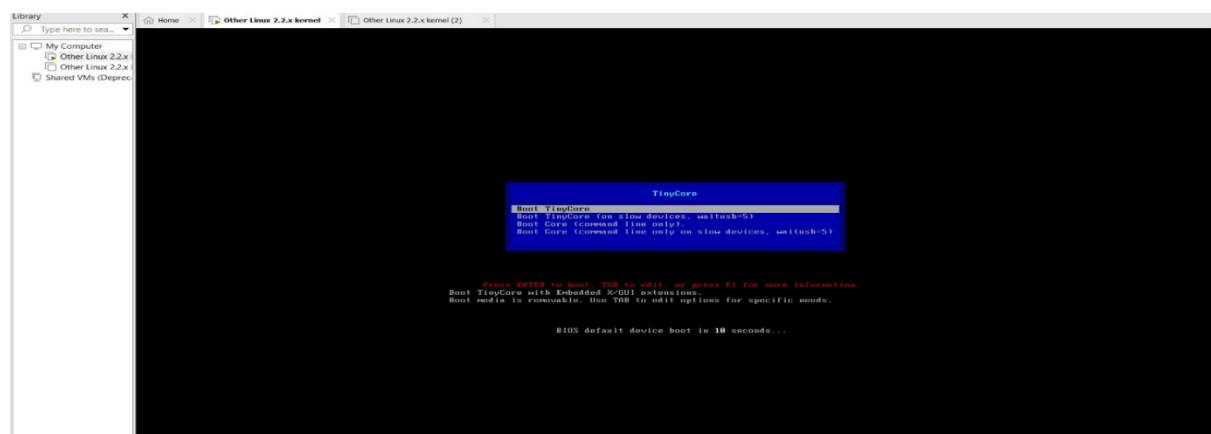


Connect to a
Remote Server

STEP 4: CREATED TINYOS VIRTUAL MACHINE



STEP 5: LAUNCH THE VM



RESULT

EXP 3: CREATE A VIRTUAL HARD DISK AND ALLOCATE THE STORAGE USING VM WARE WORKSTATION

DATE:

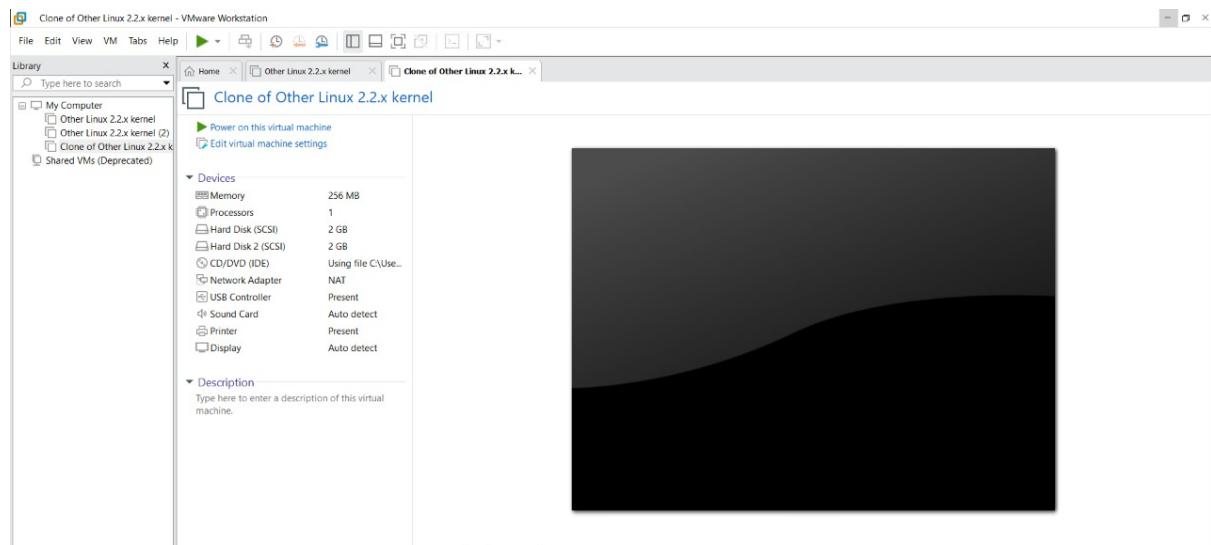
AIM:

PROCEDURE:

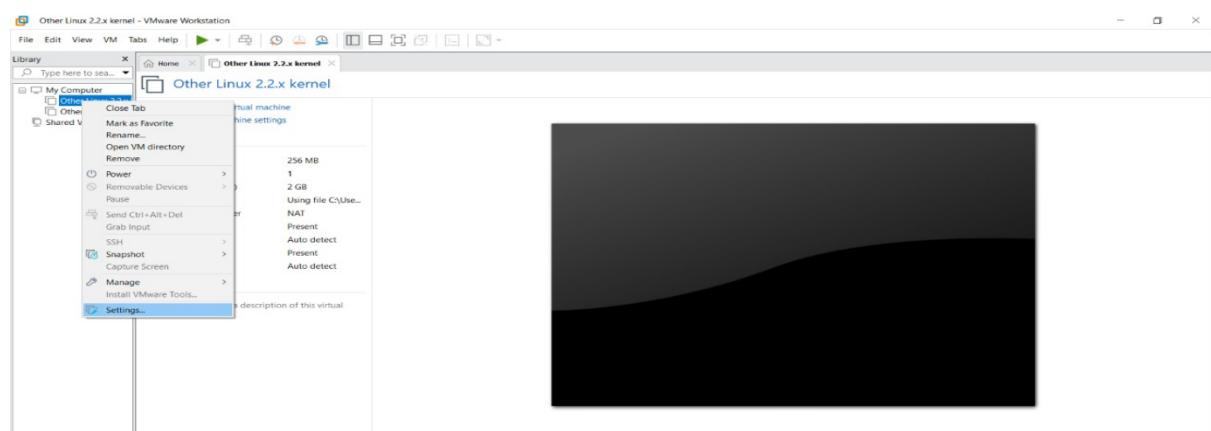
IMPLEMENTATION:

STEP 1:

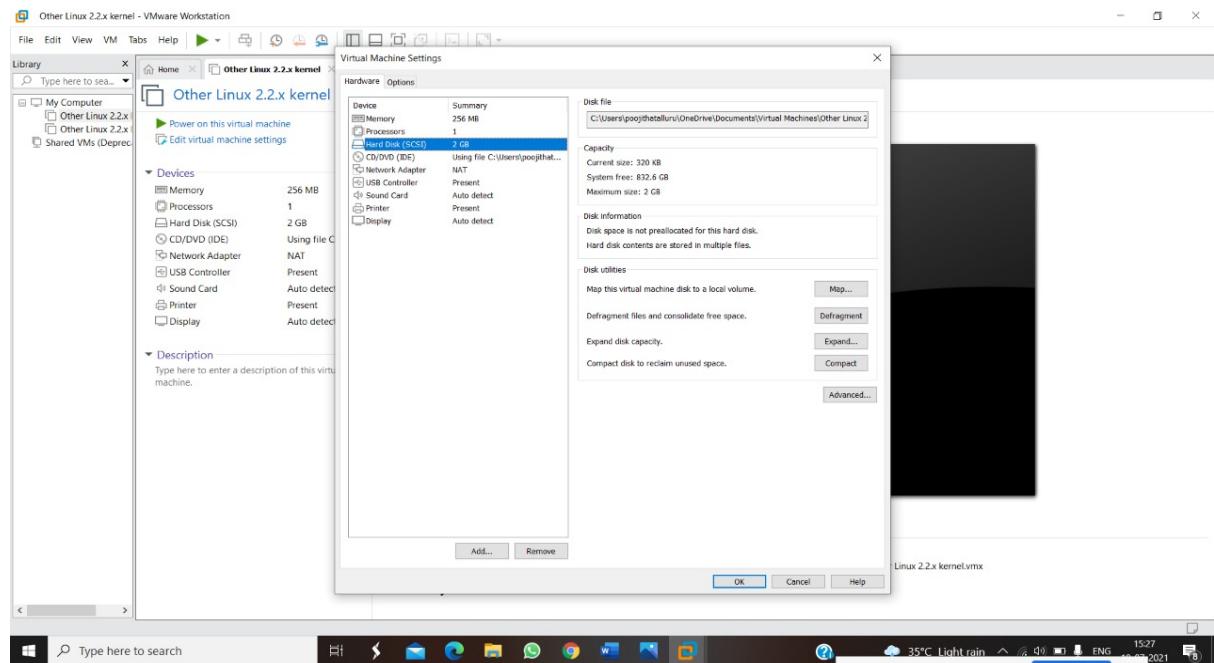
GOTO VM WARE WORKSTATION



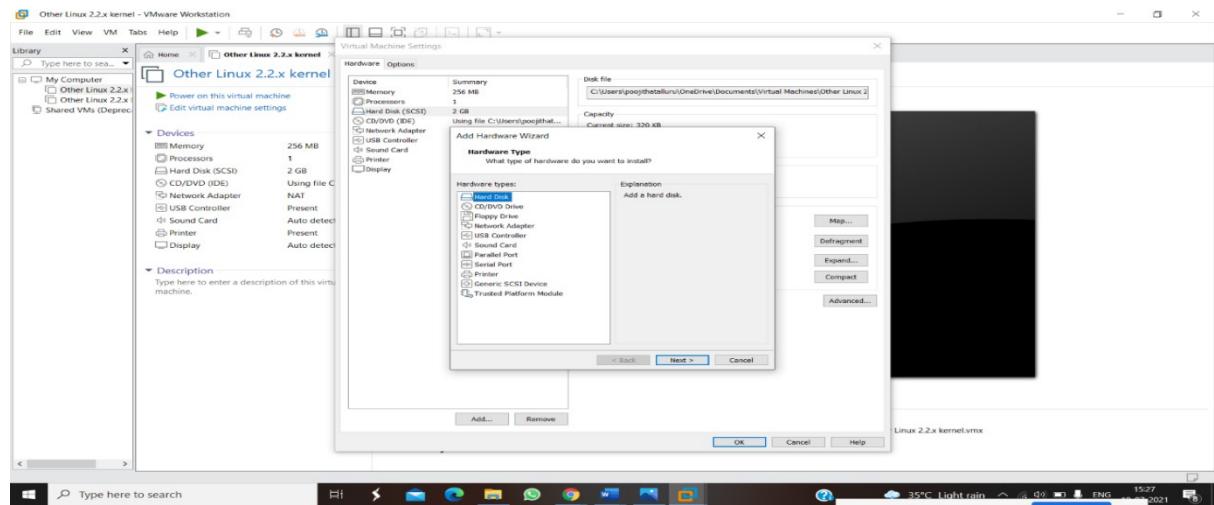
STEP2: RIGHT CLICK THE VM AND GOTO THE SETTINGS

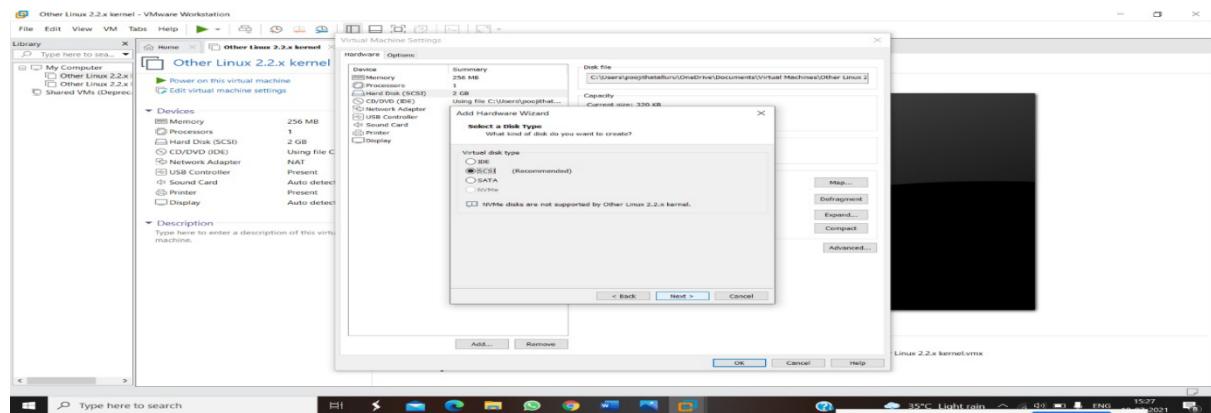


STEP 3: ADD HARDWARE WIZARD AND SELECT SCSI AND CLICK NEXT

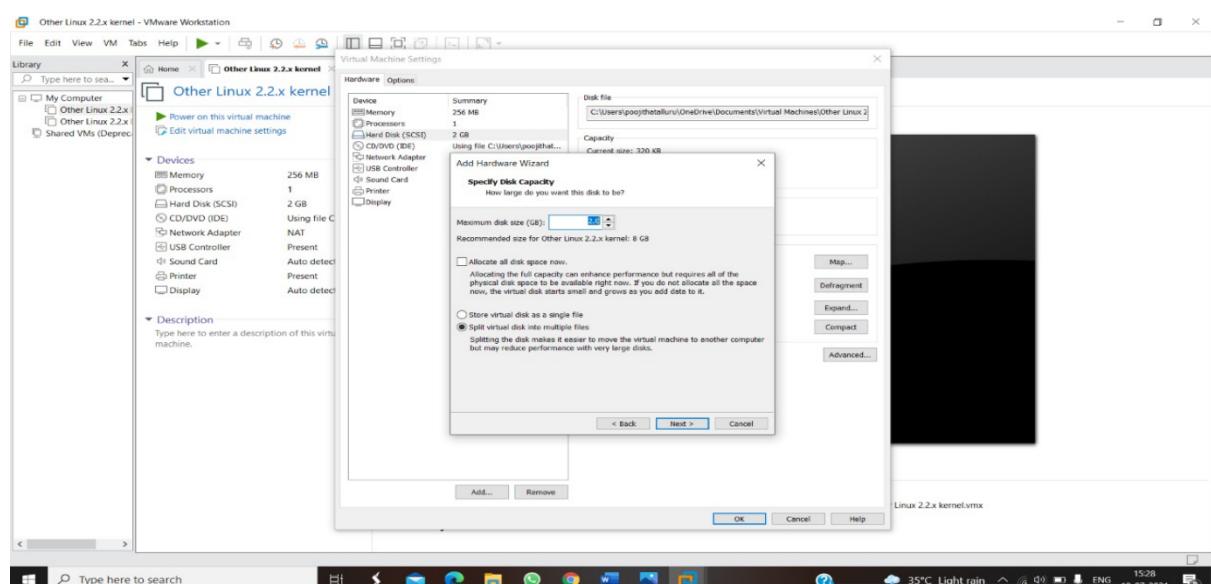


STEP 4: CREATE NEW VIRTUAL DISK

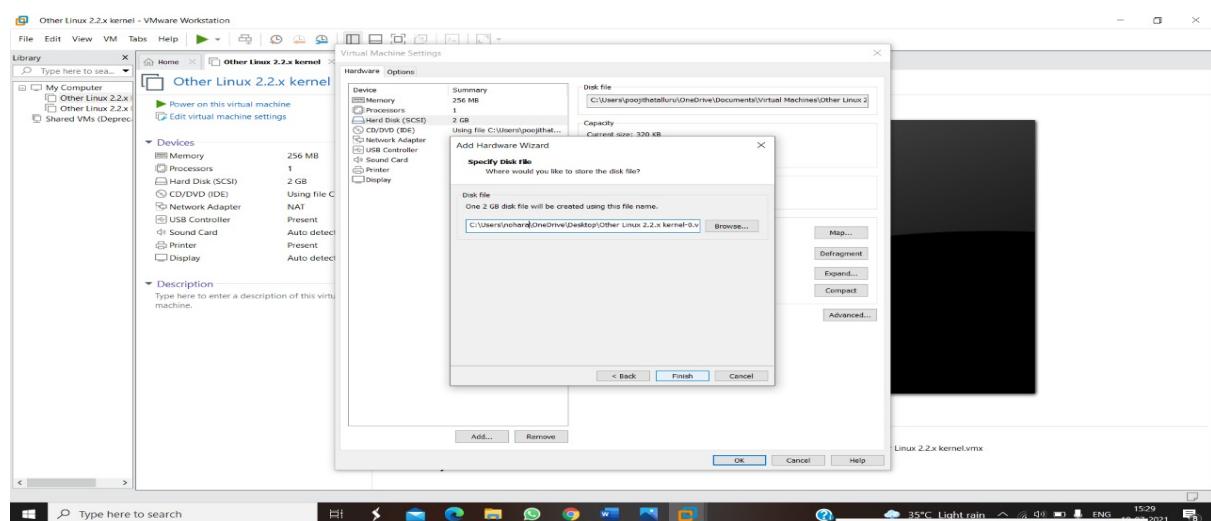


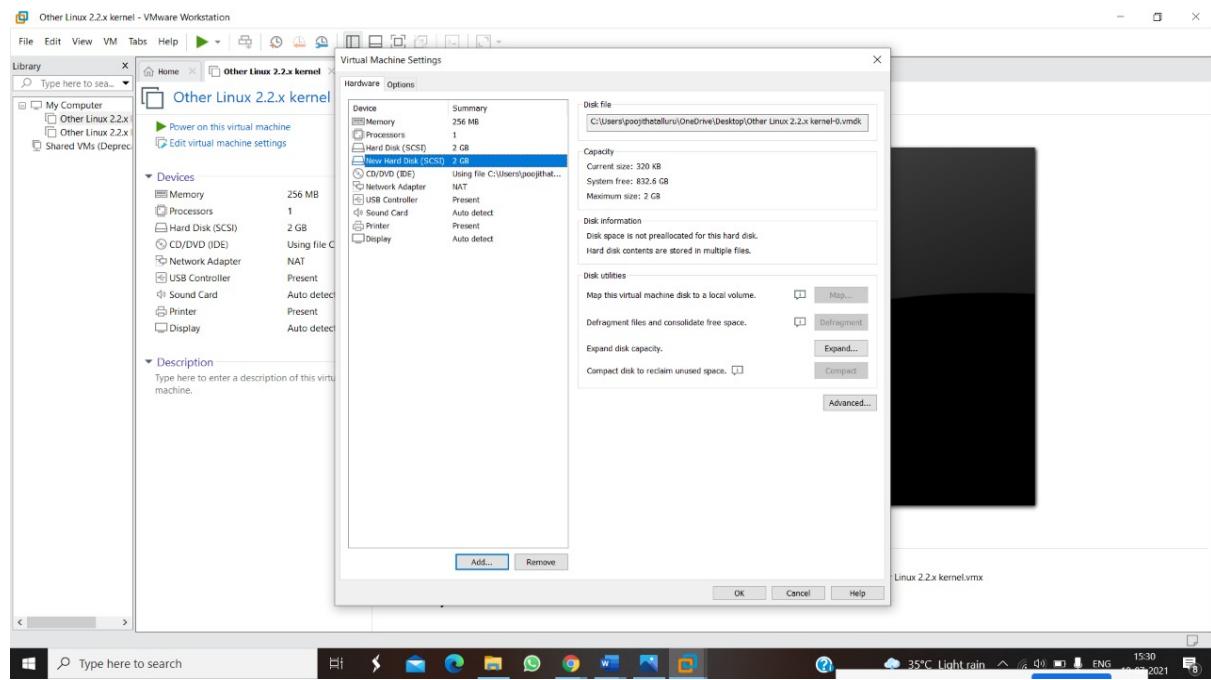


STEP 5: SELECT THE DISK SIZE AS 2.0. AND SELECT SPLIT VIRTUAL DISK INTO MULTIFILES.



STEP 6: GIVE NAME AND CLICK THE FINISH





RESULT:

EXPNO 4: CREATE A SNAPSHOT AND CLONING OF A VM AND TEST IT BY LOADING THE PREVIOUS VERSION/CLONED VM

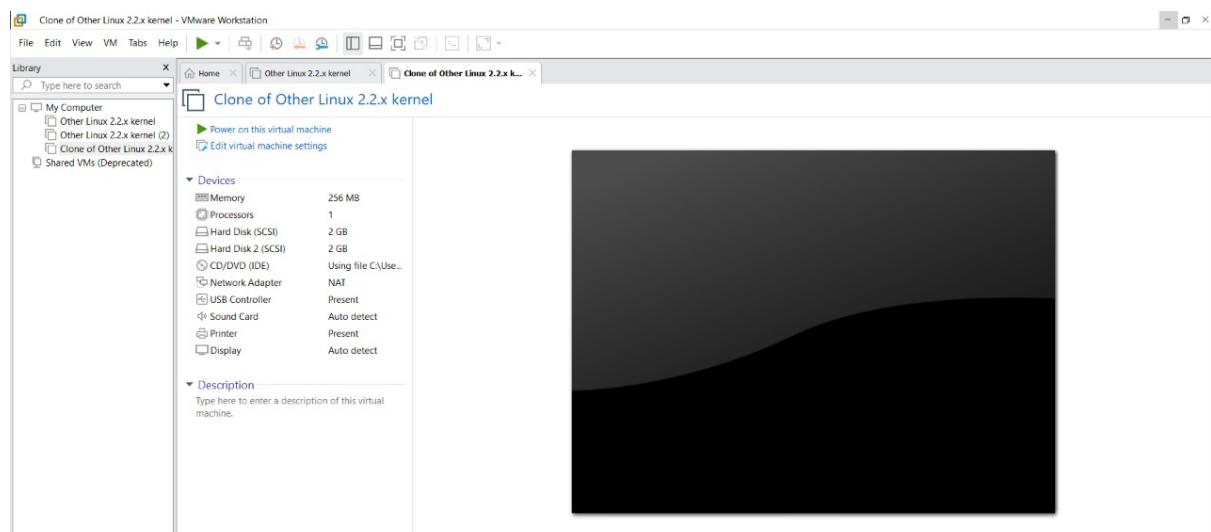
DATE:

AIM:

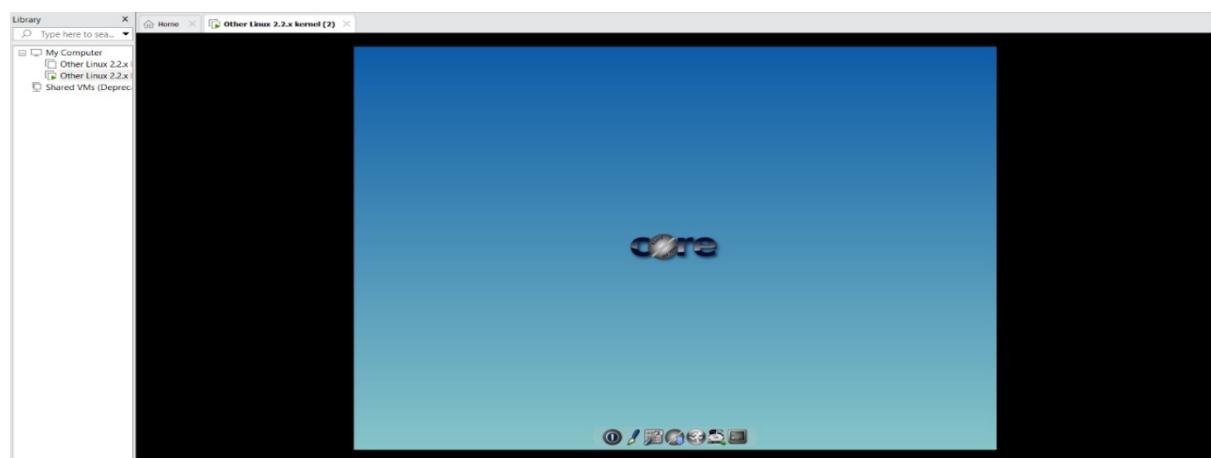
PROCEDURE:

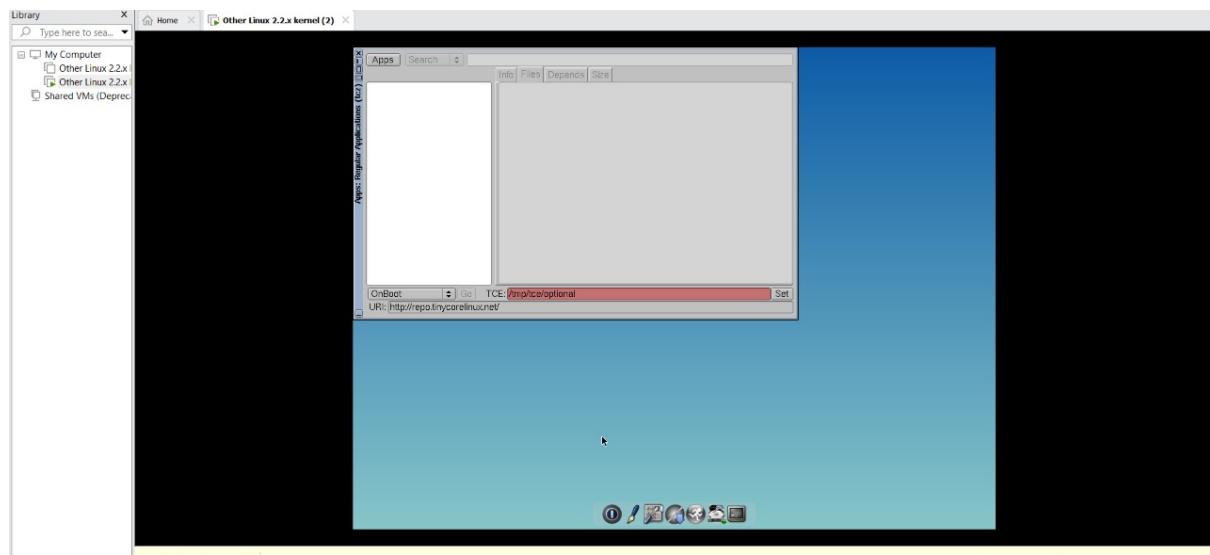
IMPLEMENTATION:

STEP 1: GOTO VMWARE WORKSTATION

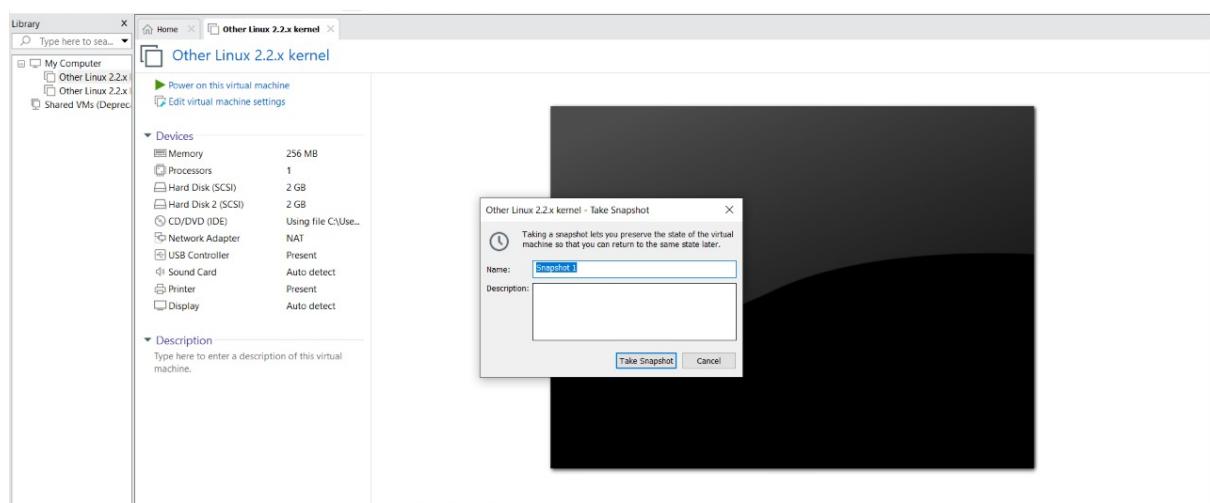
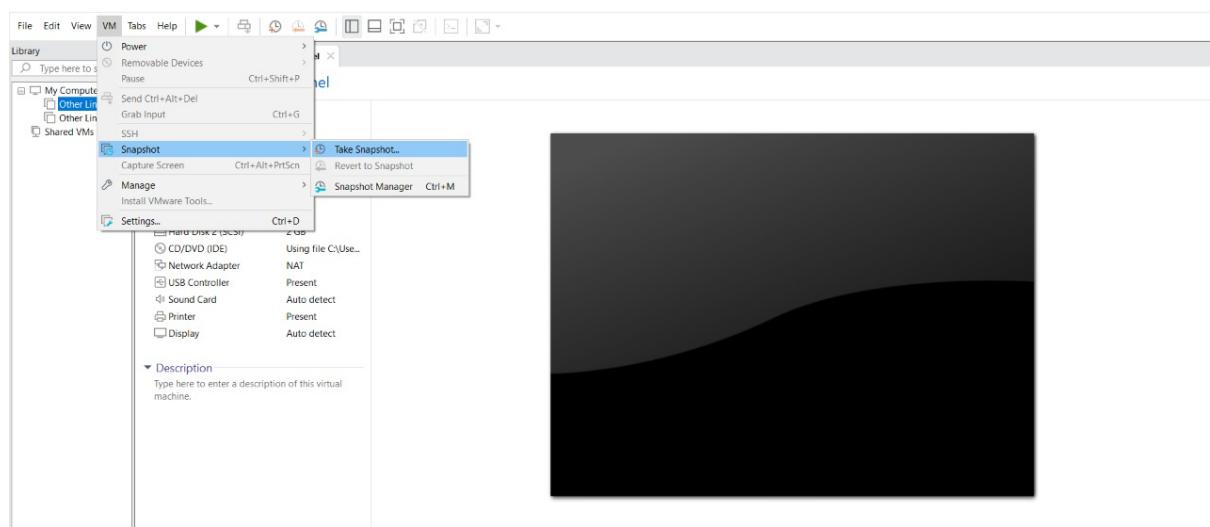


STEP 2: CREATE FILES ON DESKTOP

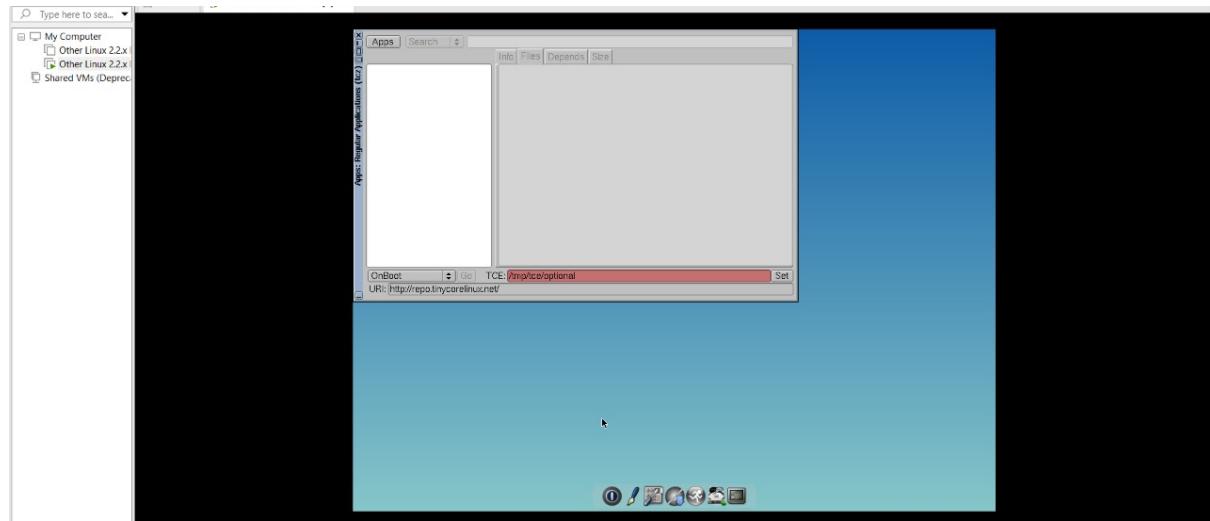




STEP 2: CLICK ON VM AND SELECTS SNAPSHOT-> TAKE SNAPSHOT.

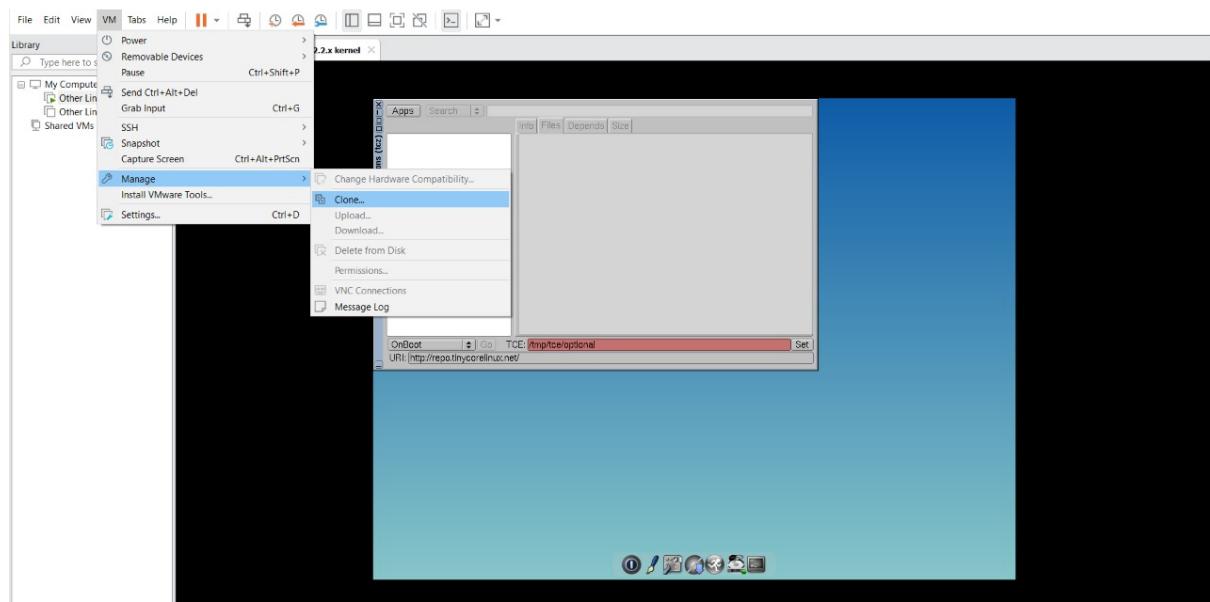


STEP 4: SNAPSHOT IS BEING DONE

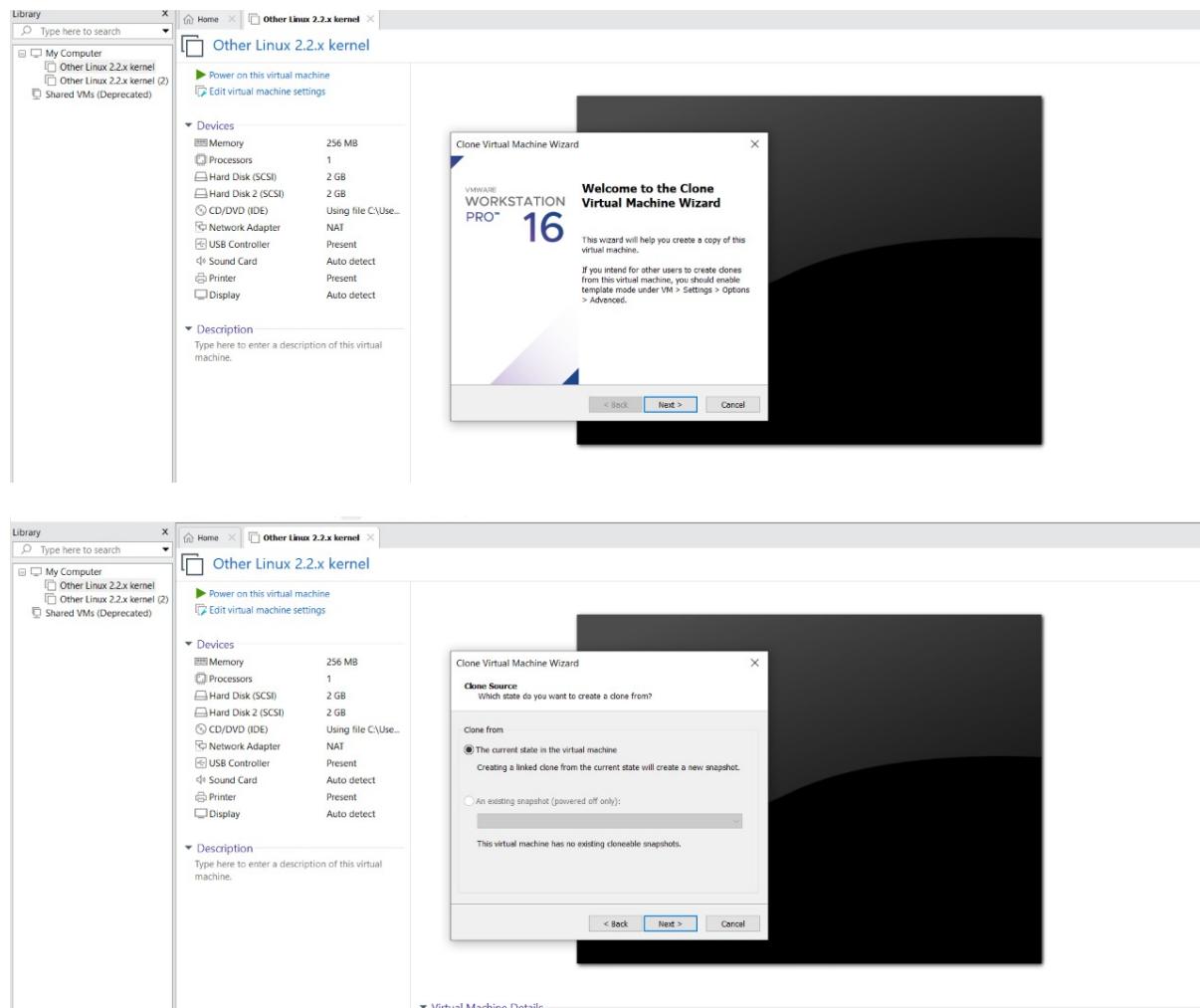


CLONING OF A VM

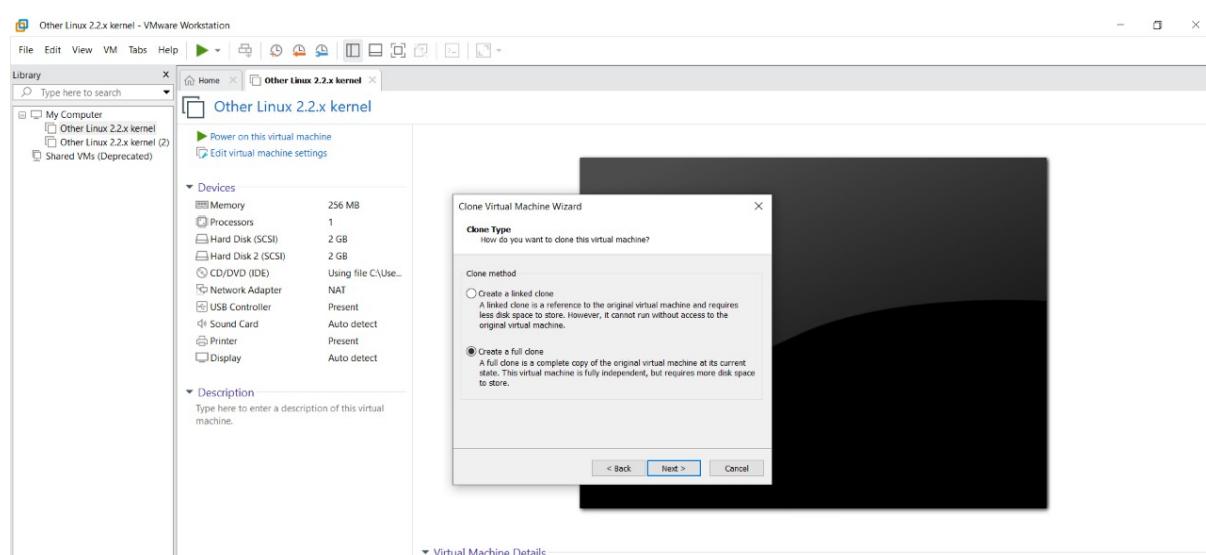
STEP 1: GO TO VM AND GOTO MANAGE AND CLICK CLONE

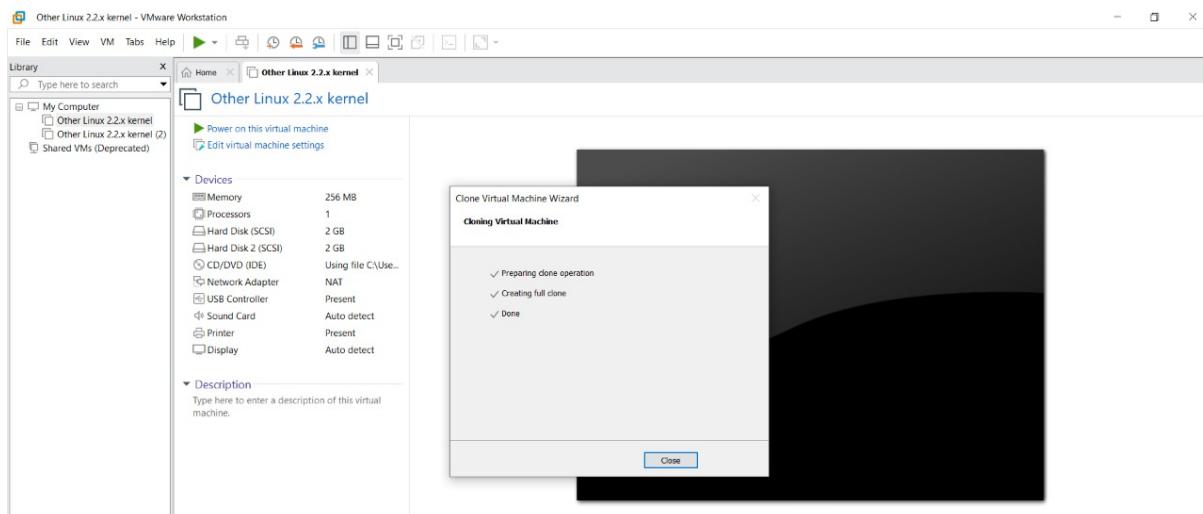


STEP 2: CLICK CLONE

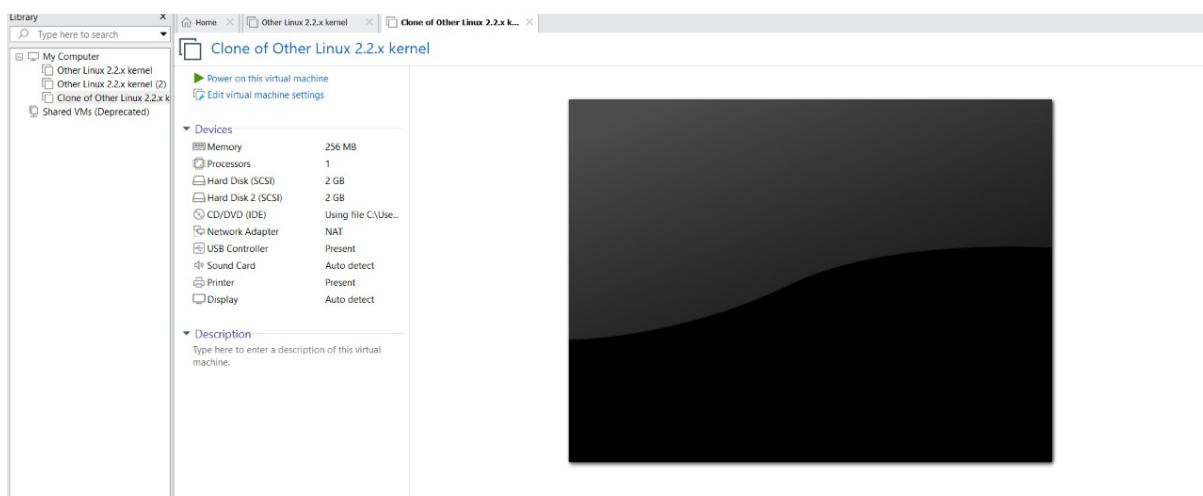


STEP 3: SELECT THE FULL CLONE





STEP 4: AFTER CLONE AGAIN OR VM IS OPENED.



RESULT:

5.DEMONSTRATE INFRASTRUCTURE AS A SERVICE(IAAS) BY CREATING A VIRTUAL MACHINE USING A PUBLIC CLOUD SERVICE PROVIDER(AZURE/GCP/AWS) CONFIGURE WITH MINIMUM CPU,RAM AND STORAGE AND LAUNCH THE VM IMAGE.

AIM:

Procedure:

Implementation:

STEP1:CREATE AN ACCOUNT IN MICROSOFT AZURE.

STEP2: GOTO RESOURCE GROUP AND CREATE A RESOURCE GROUP.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user account information. Below the navigation bar, the left sidebar displays the 'Resource groups' section, showing two existing resource groups: 'DefaultResourceGroup-CAU' and 'NetworkWatcherRG'. The main content area is titled 'Create a resource group' and is currently on the 'Review + create' step. A green banner at the top of this section indicates 'Validation passed.' The 'Basics' tab is selected, showing configuration details: Subscription (Azure for Students), Resource group (Record), and Region (East US). The 'Tags' section shows 'None'. At the bottom of the page, there are navigation links for 'Page 1 of 1', 'Create' (which is highlighted in blue), 'Previous', 'Next', and a link to 'Download a template for automation'.

STEP3: GIVE NECESSARY THINGS FOR RESOURCE GROUP.

Microsoft Azure

Resource groups

Create a resource group

Basics Tags Review + create

Project details

Subscription * Azure for Students

Resource group * Record

Resource details

Region * (US) East US

Review + create < Previous Next: Tags >

Microsoft Azure

Resource groups

Create a resource group

Basics Tags Review + create

Apply tags to your Azure resources to logically organize them by categories. A tag consists of a key (name) and a value. Tag names are case-insensitive and tag values are case-sensitive. [Learn more](#)

Name : Value : Resource

Review + create < Previous Next: Review + create >

STEP4: CREATE A VIRTUAL NETWORK FOR TO CREATE A VIRTUAL MACHINE .

The screenshot shows the Microsoft Azure portal's Resource groups page. At the top, there are navigation links for Home, Create, Manage view, Refresh, Export to CSV, Open query, Assign tags, and Feedback. Below the header, there are filter options for Subscription (all), Location (all), and Add filter. The main table lists three resource groups:

| Name | Subscription | Location | Actions |
|--------------------------|--------------------|-------------------|---------|
| DefaultResourceGroup-CAU | Azure for Students | Australia Central | ... |
| NetworkWatcherRG | Azure for Students | East US | ... |
| Record | Azure for Students | East US | ... |

At the bottom of the page, there are pagination controls: < Previous, Page 1 of 1, and Next >. The URL http://portal.azure.com/# is visible at the bottom.

STEP5: NOW CREATE A VIRTUAL MACHINE WITH UR IP ADDRESS AN USERNAME AND PASSWORD FOR YOUR VIRTUAL MACINE.

STEP6: AND YOUR VIRTUAL MACHINE IS DEPLOYED.

The screenshot shows the Microsoft Azure portal's CreateVm-MicrosoftWindowsServer.WindowsServer-2021-20210721104828 | Overview page. At the top, there are links for Home, Create, Manage view, Refresh, and a user profile. The main content area has tabs for Overview, Inputs, Outputs, and Template. The Overview tab is selected and displays the message "Your deployment is complete". It provides deployment details: Deployment name: CreateVm-MicrosoftWindowsServer.WindowsSe..., Start time: 7/21/2021, 10:52:14 AM, Subscription: Azure for Students, Correlation ID: a0f40b35-8270-49dc-bcf7-42eec66e5c61, and Resource group: Record. There are also sections for Deployment details (Download) and Next steps, which include Setup auto-shutdown, Monitor VM health, performance and network dependencies, and Run a script inside the virtual machine. At the bottom, there are buttons for Go to resource and Create another VM. On the right side, there are links for Security Center, Free Microsoft tutorials, and Work with an expert.

STEP7: NOW CONNECT THE VIRTUAL MACHINE AND DOWNLOAD THE RDP FILE TO OPEN YOUR WINDOWS VIRTUAL MACHINE.

Microsoft Azure Search resources, services, and docs (G+) srisaivenkatasubbarao1...
DEFAULT DIRECTORY

Home > CreateVm-MicrosoftWindowsServer.WindowsServer-201-20210721104828 >

Record-virtual

Virtual machine

Search (Ctrl+F) Connect Start Restart Stop Capture Delete Refresh Open in mobile

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Networking Connect Windows Admin Center (preview) Disks Size Security Advisor recommendations Extensions

Resource group (change) : Record Status : Running Location : East US Subscription (change) : Azure for Students Subscription ID : db4eee0b-1e34-4be0-9c9c-65cc8d398405 Tags (change) : Click here to add tags

Operating system : Windows (Windows Server 2019 Datacenter) Size : Standard DS1 v2 (1 vcpus, 3.5 GiB memory) Public IP address : 23.96.9.147 Virtual network/subnet : Record-vnet/default DNS name : Not configured

Properties Monitoring Capabilities (8) Recommendations Tutorials

Virtual machine

| | |
|------------------|--|
| Computer name | Record-virtual |
| Operating system | Windows (Windows Server 2019 Datacenter) |
| Publisher | MicrosoftWindowsServer |
| Offer | WindowsServer |
| Plan | 2019-Datacenter |
| VM generation | V1 |
| Agent status | Ready |

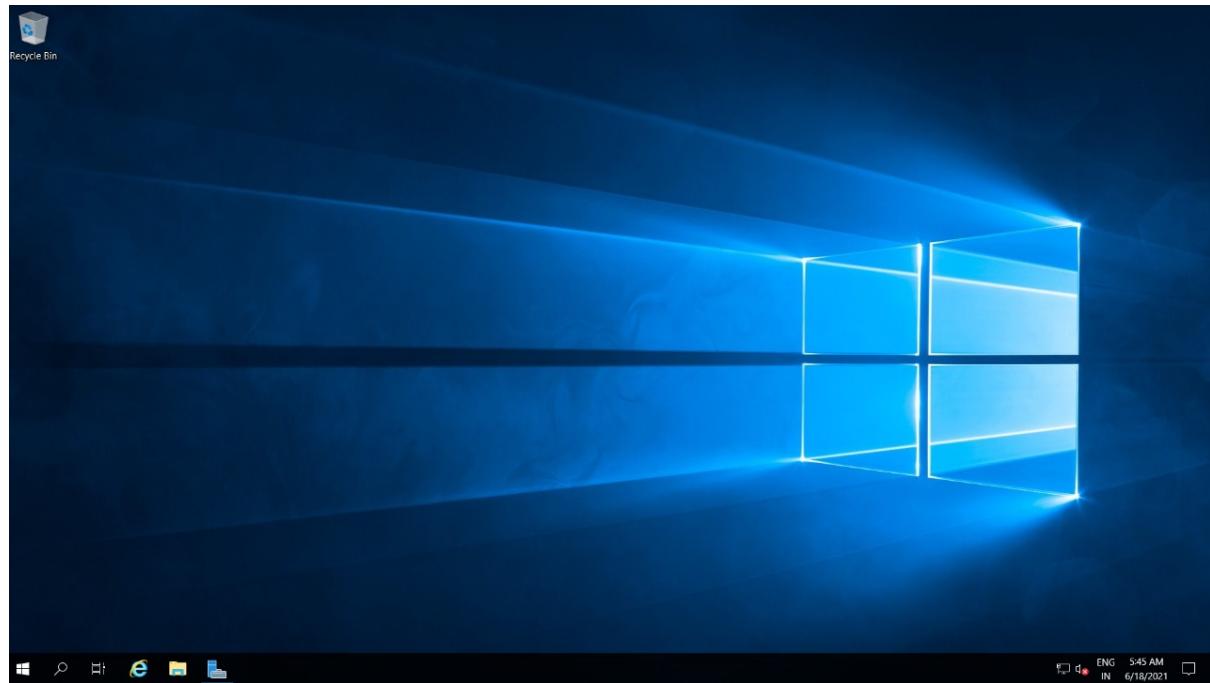
Networking

| | |
|---------------------------|---------------------|
| Public IP address | 23.96.9.147 |
| Public IP address (IPv6) | - |
| Private IP address | 10.0.0.4 |
| Private IP address (IPv6) | - |
| Virtual network/subnet | Record-vnet/default |
| DNS name | Configure |

JSON View

<https://portal.azure.com/#>

STEP8: CREATED A NEW WINDOWS VIRTUAL MACHINE.



RESULT:

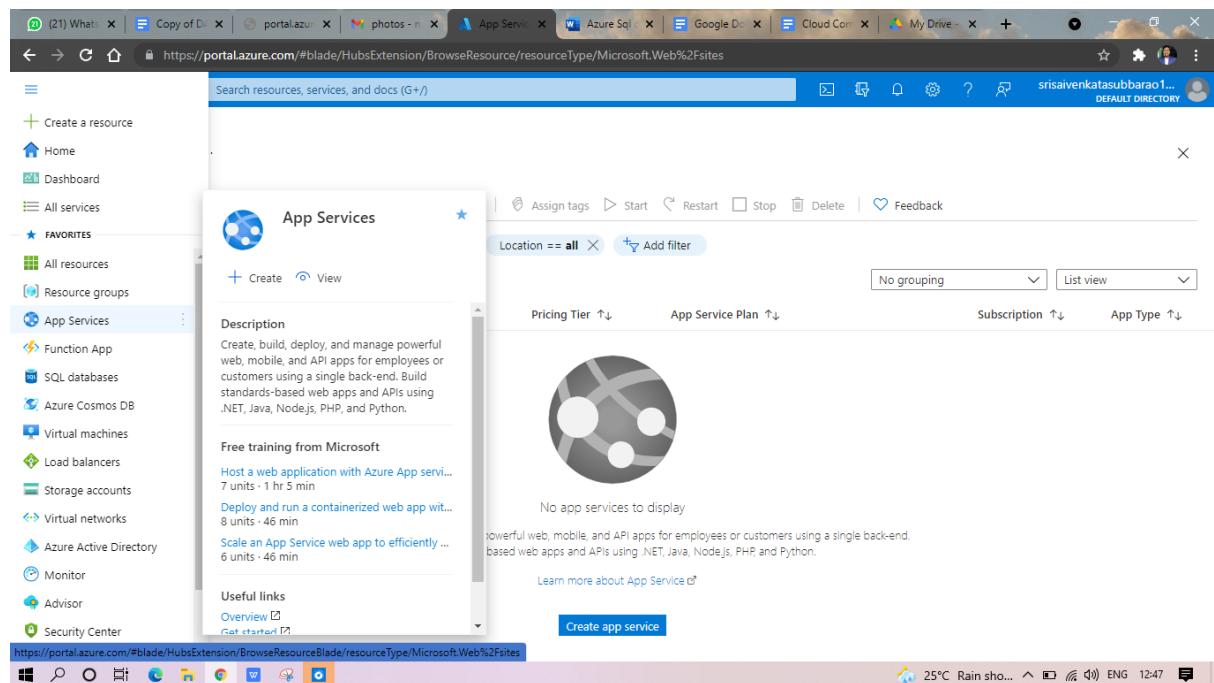
6.CREATE A SIMPLE WEB SITE USING ANY PUBLIC CLOUD SERVICE PROVIDER (AZURE/GCP/AWS) AND CHECK THE PUBLIC ACCESSIBILITY OF THE STORED FILE TO DEMONSTRATE STORAGE AS A SERVICE

AIM:

Procedure:

IMPLEMENTATION:

STEP1: FIRSTLY GO TO APPSERVICE TO CREATE AN WEBAPP.



STEP2: ENTER THE RESOURCE GROUP AND WEBAPP NAME AND REGION AND SELECT THE LINUX OS.

The screenshot shows the 'Create Web App' wizard in the Microsoft Azure portal. In the 'Project Details' section, 'Subscription' is set to 'Azure for Students' and 'Resource Group' is set to 'Record'. In the 'Instance Details' section, 'Name' is 'Record-app', 'Publish' is 'Code', 'Runtime stack' is 'Node 14 LTS', and 'Operating System' is 'Linux'. At the bottom, there are 'Review + create' and 'Next : Deployment (Preview) >' buttons.

STEP3: AFTER ENTER THE ALL THE NECESSARY THINGS CLICK THE REVIEW AND CREATE AND CLICK THE CREATE THE WEB APP.

The screenshot shows the 'Create Web App' wizard with the 'Review + create' tab selected. It displays the summary of the configuration: 'Web App by Microsoft' (Basic (B1) sku), 'Subscription' (db4eee0b-1e34-4be0-9c9c-65cc8d398405), 'Resource Group' (Record), 'Name' (Record-app), 'Publish' (Code), 'Runtime stack' (Node 14 LTS). Below this, the 'App Service Plan (New)' section shows 'Name' (ASP-Record-92e3), 'Operating System' (Linux), and 'Region' (Australia Central). At the bottom, there are 'Create', '< Previous', 'Next >', and 'Download a template for automation' buttons.

STEP4: AND OUR DEPLOYMENT IS COMPLETED.

The screenshot shows the Microsoft Azure WebApp Portal for the deployment named "Microsoft.Web-WebApp-Portal-1b6a401b-9ae6". The main message is "Your deployment is complete". Deployment details include name: Microsoft.Web-WebApp-Portal-1b6a401b-9ae6, Subscription: Azure for Students, Resource group: Record, Start time: 7/21/2021, 12:49:54 PM, Correlation ID: 76653cd2-c090-4d97-a1e5-21033aa42efc. The "Deployment details" section has a "Download" link. The "Next steps" section recommends managing deployments and protecting the app with authentication. A "Go to resource" button is at the bottom. On the right, there are links to Security Center, Free Microsoft tutorials, and Work with an expert.

STEP5: GOTO WEBSITE URL LINK.

The screenshot shows the Microsoft Azure App Service Overview for the "Record-app" service. It displays basic information like Resource group (Record), Status (Running), Location (Australia Central), Subscription (Azure for Students), and Subscription ID (db4ee0b1-e344-be09-9c9c-65cc8d398405). The URL is https://record-app.azurewebsites.net. The interface includes sections for "Essentials", "Diagnose and solve problems", "Application Insights", and "App Service Advisor". A JSON View link is also present.

STEP6: THIS IS OUR WEBAPP SERVICE.

The screenshot shows a browser window displaying the deployed Node.js application at https://record-app.azurewebsites.net. The page greets "Hey, Node developers!" and says "Your app service is up and running. Time to take the next step and deploy your code.". It includes a "Deployment Center" and "Quickstart" button. A cartoon character is shown working on a laptop with a "node" logo.

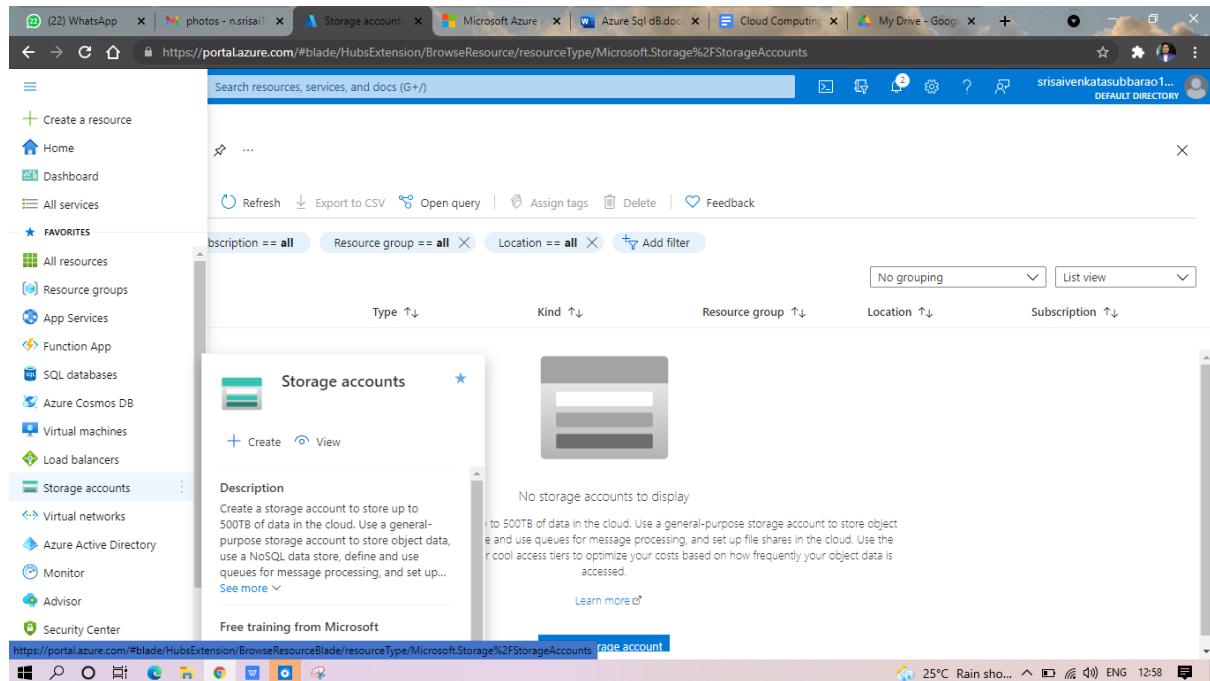
7.CREATE A STORAGE SERVICE USING ANY PUBLIC CLOUD SERVICE PROVIDER (AZURE/GCP/AWS) AND CHECK THE PUBLIC ACCESSIBILITY OF THE STORED FILE TO DEMONSTRATE STORAGE AS A SERVICE.

AIM:

PROCEDURE:

IMPLEMENTATION:

STEP1: OPEN AZURE AND GOTO STORAGE ACCOUNTS AND CREATE STOROAGE ACCOUNT



STEP2: ENTER THE RESOURC GROUP AND AND STORAGE ACCOUNT NAME AND REVIEW AND CREATE AND CLICK TH CREATE AND YOUR STORAGE ACCOUNT WILL BE DEPLOYED SUCESSFULLY.

The screenshot shows the Microsoft Azure Deployment Overview page for a deployment named "recordstoragesubbarao_1626852653220". The status is "Deployment succeeded". The deployment was made to the "Record" resource group. The deployment name is "recordstoragesubbarao_1626852653220", the subscription is "Azure for Students", and the start time is 7/21/2021, 1:00:57 PM. The correlation ID is 49e91964-e371-4019-ae37-954bf26dd89f. The page includes sections for "Deployment details" (with a download link), "Next steps", and "Feedback". On the right side, there are links to "Security Center", "Free Microsoft tutorials", and "Work with an expert".

STP3: AND OUR STORAGE ACCOUNT IS CREATED.

The screenshot shows the Microsoft Azure Storage Account Overview page for a storage account named "recordstoragesubbarao". The account is located in the "East US" region and is a "Standard/Hot" performance tier. It uses "Read-access geo-redundant storage (RA-GRS)" replication. The account kind is "StorageV2 (general purpose v2)". The provisioning state is "Succeeded" and it was created on 7/21/2021, 1:01:05 PM. The storage account has a "Record" tag. The page displays "Essentials" information and "Blob service" and "Security" settings. The "Blob service" section shows settings for hierarchical namespace, default access tier (Hot), blob public access (Enabled), and blob soft delete (Enabled / 7 days). The "Security" section shows settings for require secure transfer for REST API operations (Enabled), storage account key access (Enabled), and minimum TLS version (Version 1.2).

STEP5: GOTO STATIC WEBSITE

Resource group (change) : Gopi

Location : East US

Primary/Secondary Location : Primary: East US. Secondary: West US

Subscription (change) : Azure for Students

Subscription ID : 88bd0e11-e431-4a2a-8040-bdf7d22901aa

Disk state : Primary: Available; Secondary: Available

Tags (change) :

Properties **Monitoring** **Capabilities (7)** **Recommendations** **Tutorials** **Developer Tools**

Blob service

| | |
|------------------------|------------------|
| Hierarchical namespace | Disabled |
| Default access tier | Hot |
| Blob public access | Enabled |
| Blob soft delete | Enabled (7 days) |
| Container soft delete | Enabled (7 days) |
| Versioning | Disabled |

Security

| | |
|---|-------------|
| Require secure transfer for REST API operations | Enabled |
| Storage account key access | Enabled |
| Minimum TLS version | Version 1.2 |
| Infrastructure encryption | Disabled |

Networking

STEP6: AND ENABLE AND ENTER YOUR INDEX AND ERROR HTML FILENAMES.

Static website

Static website (Disabled) **Enabled**

An Azure Storage container has been created to host your static website.
swb

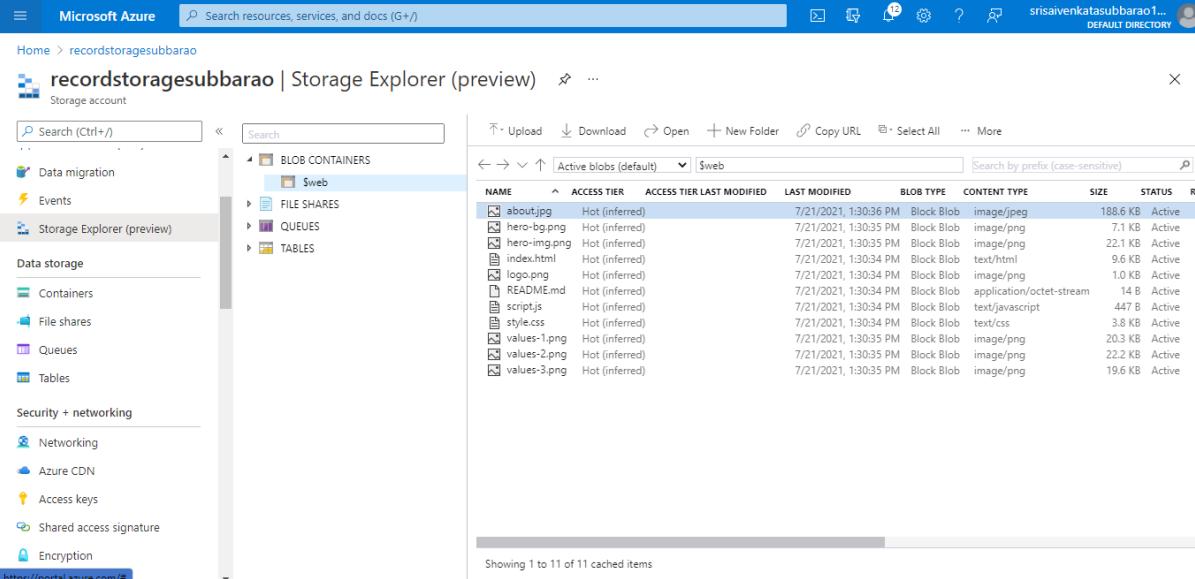
Primary endpoint

Secondary endpoint

Index document name

Error document path

STEP7: AND GOTO STORAGE EXPLORER (REVIEW) AND AND GOTO BLOB CONTAINERS AND WEB AND UPLOAD THE TWO HTML FILES INIT

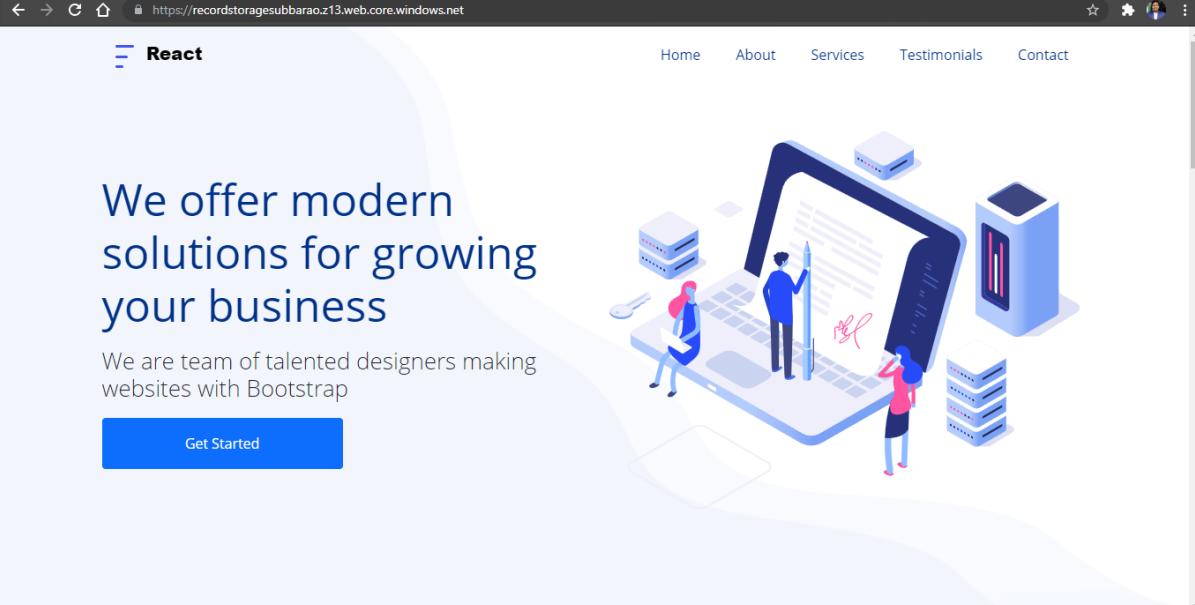


The screenshot shows the Microsoft Azure Storage Explorer interface. On the left, there's a sidebar with various options like Data migration, Events, Storage Explorer (preview), Data storage (Containers, File shares, Queues, Tables), and Security + networking (Networking, Azure CDN, Access keys, Shared access signature, Encryption). The main area shows a list of blob containers under 'BLOB CONTAINERS'. One container, '\$web', is selected. Inside '\$web', there are several files listed in a table:

| NAME | ACCESS TIER | ACCESS TIER LAST MODIFIED | LAST MODIFIED | BLOB TYPE | CONTENT TYPE | SIZE | STATUS | RI |
|--------------|----------------|---------------------------|-----------------------|------------|--------------------------|----------|--------|----|
| about.jpg | Hot (inferred) | | 7/21/2021, 1:30:36 PM | Block Blob | image/jpeg | 188.6 KB | Active | |
| hero-bg.png | Hot (inferred) | | 7/21/2021, 1:30:35 PM | Block Blob | image/png | 7.1 KB | Active | |
| hero-img.png | Hot (inferred) | | 7/21/2021, 1:30:35 PM | Block Blob | image/png | 22.1 KB | Active | |
| index.html | Hot (inferred) | | 7/21/2021, 1:30:34 PM | Block Blob | text/html | 9.6 KB | Active | |
| logo.png | Hot (inferred) | | 7/21/2021, 1:30:34 PM | Block Blob | image/png | 1.0 KB | Active | |
| README.md | Hot (inferred) | | 7/21/2021, 1:30:34 PM | Block Blob | application/octet-stream | 14 B | Active | |
| script.js | Hot (inferred) | | 7/21/2021, 1:30:34 PM | Block Blob | text/javascript | 447 B | Active | |
| style.css | Hot (inferred) | | 7/21/2021, 1:30:34 PM | Block Blob | text/css | 3.8 KB | Active | |
| values-1.png | Hot (inferred) | | 7/21/2021, 1:30:35 PM | Block Blob | image/png | 203 KB | Active | |
| values-2.png | Hot (inferred) | | 7/21/2021, 1:30:35 PM | Block Blob | image/png | 22.2 KB | Active | |
| values-3.png | Hot (inferred) | | 7/21/2021, 1:30:35 PM | Block Blob | image/png | 19.6 KB | Active | |

Showing 1 to 11 of 11 cached items

STEP8: AND AGAIN RETURN TO STATIC WEBSITE AND OPEN THE PRIMARY LINK AND YOUR WEB PAGE IS CREATED



The screenshot shows a web browser displaying a static website. The URL in the address bar is <https://recordstoragesubbarao.z13.web.core.windows.net>. The page has a header with the word 'React' and navigation links for Home, About, Services, Testimonials, and Contact. The main content features a large blue heading: 'We offer modern solutions for growing your business'. Below the heading, a subtext says: 'We are team of talented designers making websites with Bootstrap'. A large blue button labeled 'Get Started' is at the bottom left. To the right of the text, there's a 3D isometric illustration of a person working on a large tablet or screen, surrounded by server racks and network equipment.

RESULT:

8.CREATE A SQL STORAGE SERVICE AND PERFORM A BASIC QUERY USING ANY PUBLIC CLOUD SERVICE PROVIDER (AZURE/GCP/AWS) TO DEMONSTRATE DATABASE AS A SERVICE (DAAS)

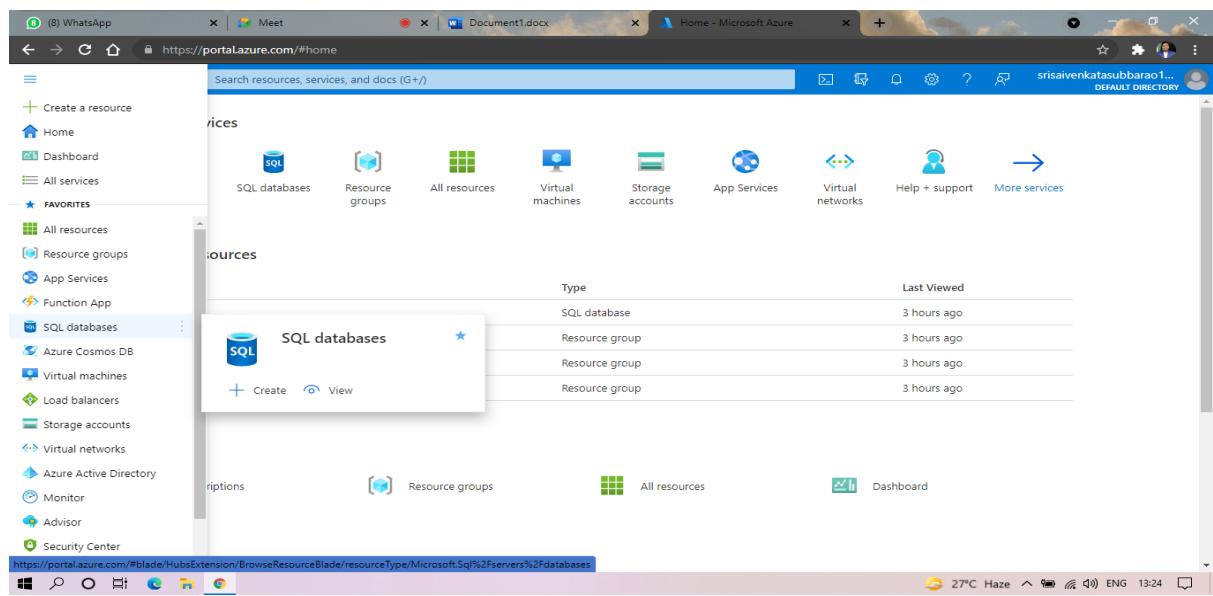
AIM:

PROCEDURE:

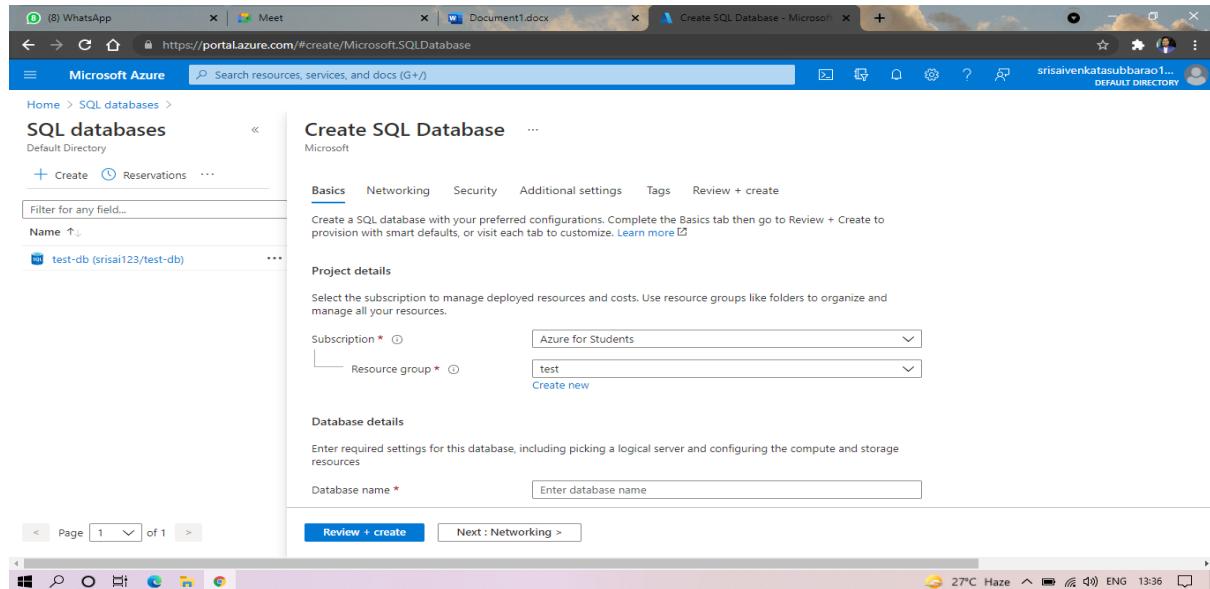
IMPLEMENTATION:

STEP1: GOTO AZURE AND GOTO SQLDATABASE.

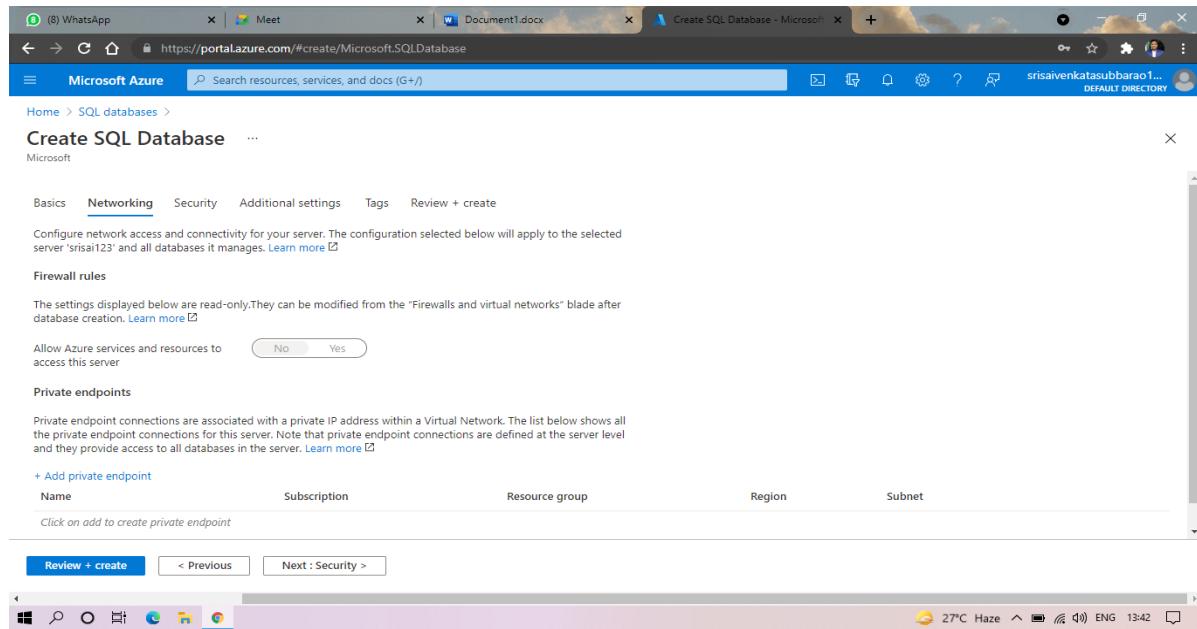
STEP 02:- Now Create a Sql Databse



STEP3: SELECT THE RESOURCE GROUP AND ENTER THE SERVERNAME THAT APPLICABLE.



STEP4: IN NETWORKING SELECT ALLOW AZURE SERVICES AND RESOURCES TO ACCESS THIS SERVER.



STEP5: IN ADDITIONAL SETTINGS SELECT SAMPLE.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#create/Microsoft.SQLDatabase>. The page title is "Create SQL Database". The "Additional settings" tab is selected. Under "Data source", the "None" button is selected. In the "Database collation" section, "SQL_Latin1_General_CI_AS" is chosen from a dropdown. The "Maintenance window" section notes that it's optional. At the bottom, there are "Review + create", "< Previous", and "Next : Tags >" buttons.

STEP6:AND THE SQL DATABASE IS DEPLOYED.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#blade/HubsExtension/DeploymentDetailsBlade/overview/id/%2Fsubscriptions%2Fdb4eee0b-1e34-4be0-9c9c-65cc8d398405%2Fresource...>. The page title is "Microsoft.SQLDatabase.newDatabaseExistingServer_155c16593e594aad | Overview". A success message box is displayed: "Deployment succeeded" with a green checkmark icon. It states: "Deployment 'Microsoft.SQLDatabase.newDatabaseExistingServer_15...' to resource group 'test' was successful." Below the message are "Go to resource" and "Pin to dashboard" buttons. On the left, there's a navigation menu with "Overview", "Inputs", "Outputs", and "Template" options. On the right, there are links to "Security Center", "Free Microsoft tutorials", and "Work with an expert". The taskbar at the bottom shows the date and time as "27°C Haze 1:50 PM" and "27°C Haze 13:50".

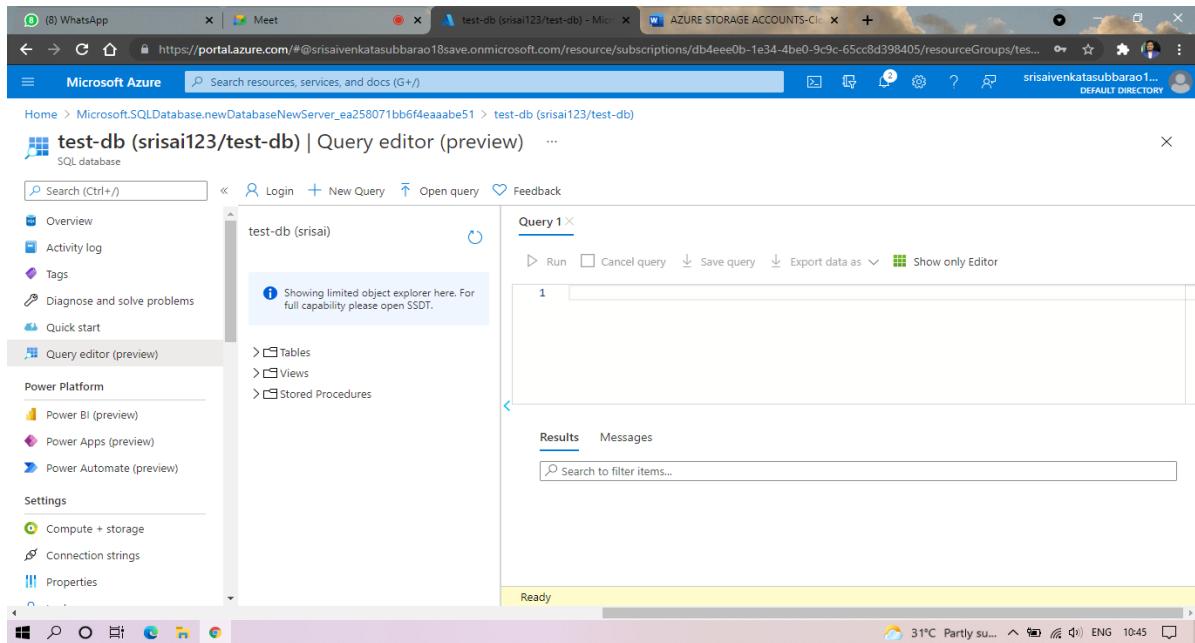
STEP7:AND NOW GOTO QUERY EDITOR.

The screenshot shows the Microsoft Azure portal interface. The main title bar says "test-db (srисai123/test-db)". The left sidebar has sections for Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The main content area displays "Essentials" information: Resource group (change) : test, Status : Paused, Location : West US 3, Subscription (change) : Azure for Students, Subscription ID : db4eee0b-1e34-4be0-9c9c-65cc8d398405, Tags (change) : Click here to add tags. It also shows Compute utilization with a graph from 0 to 100. At the bottom right, there's a "27°C Haze" weather icon and system status like ENG 13:52.

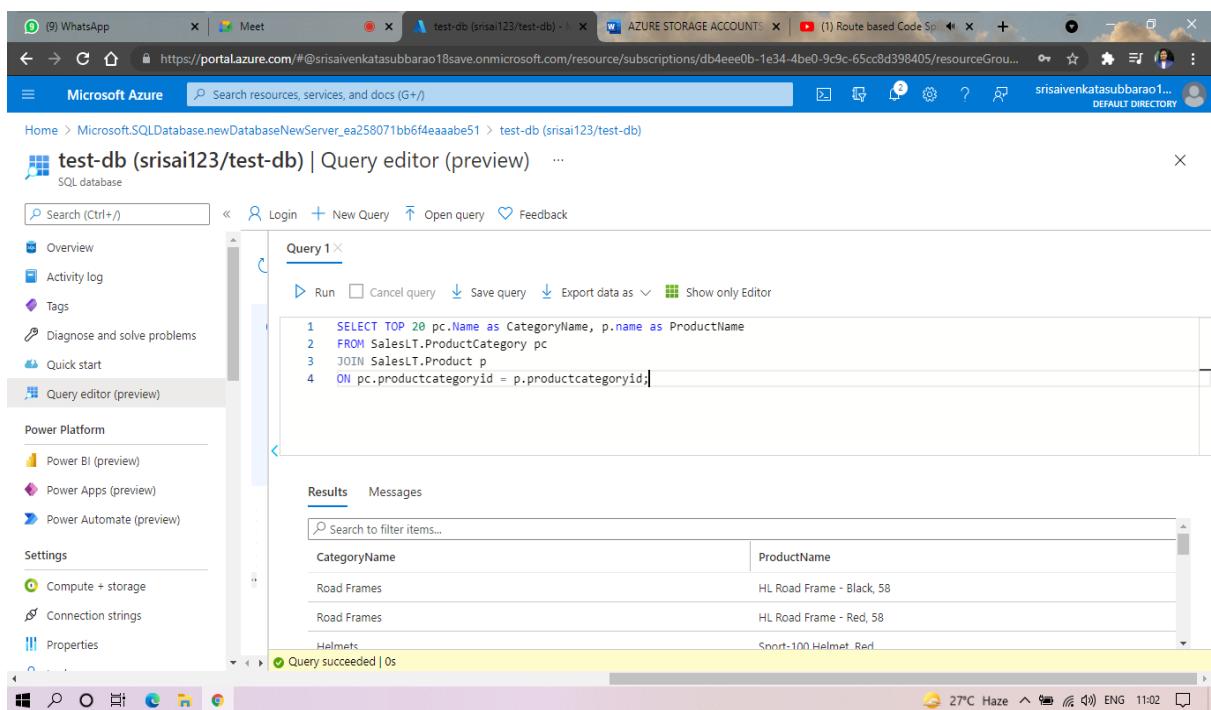
STEP8:AND NOW AGAIN LOGIN TO THE SQLDATADATABASE

The screenshot shows the Microsoft Azure portal interface with the "Query editor (preview)" tab selected. The left sidebar includes "Query editor (preview)" under the "Overview" section. The main area is titled "Welcome to SQL Database Query Editor". It shows two authentication options: "SQL server authentication" (Login: srисai, Password: [redacted]) and "Active Directory authentication" (Continue as srисaivenkatasubbarao18@s...). There is an "OK" button at the bottom. The bottom right corner shows the same weather and system status as the previous screenshot.

STEP: AND OUR TABLES WILL SHOWN AND TYPE THE QUERY TO EXECUTED.



STEP10: AND OUR OUTPUT IS READY.



RESULT:

EXP. 9: PERFORM THE BASIC CONFIGURATION SETUP FOR INSTALLING HADOOP 2.X LIKE CREATING THE HDUSER AND SSH LOCALHOST

AIM:

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.

[or]

```
$ sudo apt-get install default-jdk
```

```
$ java -version
```

Step 3 – Add a dedicated Hadoop user

```
$ sudo addgroup hadoop
```

```
$ sudo adduser --ingroup hadoop hduser
```

// Add hduser to sudo user group

```
$ sudo adduser hduser sudo
```

Step 4 – Install SSH and Create Certificates

```
$ sudo apt-get install ssh
```

```
$ su hduser
```

```
$ ssh-keygen -t rsa -P ""
```

```
// Set Environmental variables  
$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Step 5 – Check if SSH works

```
$ ssh localhost
```

Step 6 – Install Hadoop

```
// Extract Hadoop-2.7.2  
$ sudo tar xvzf hadoop-2.7.2.tar.gz
```

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

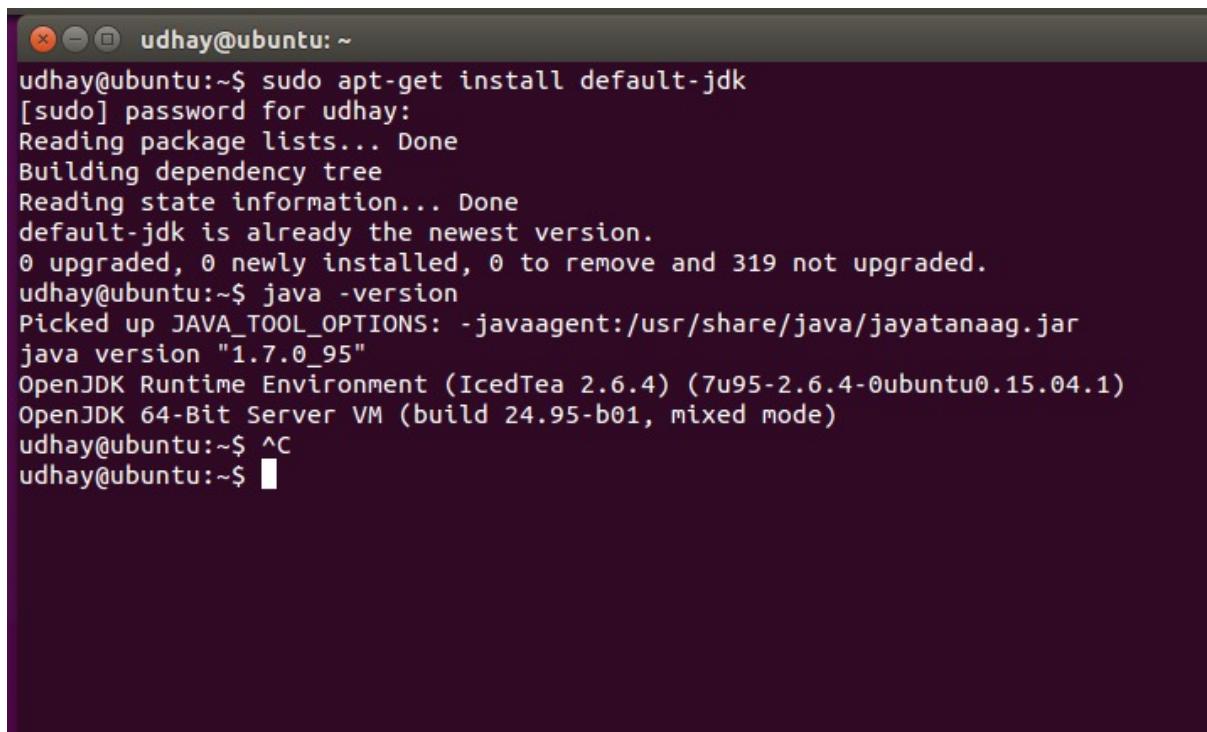
```
// Move the Hadoop folder to /usr/local/hadoop
```

```
$ sudo mv hadoop-2.7.2 /usr/local/hadoop
```

```
// Assigning read and write access to Hadoop folder
```

```
$ sudo chown -R hduser:hadoop /usr/local/hadoop
```

Implementation:



The screenshot shows a terminal window titled "udhay@ubuntu: ~". It displays the output of several commands:

```
udhay@ubuntu:~$ sudo apt-get install default-jdk
[sudo] password for udhay:
Reading package lists... Done
Building dependency tree
Reading state information... Done
default-jdk is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 319 not upgraded.
udhay@ubuntu:~$ java -version
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
java version "1.7.0_95"
OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-0ubuntu0.15.04.1)
OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
udhay@ubuntu:~$ ^C
udhay@ubuntu:~$
```

```
udhay@ubuntu:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
ssh is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 319 not upgraded.
udhay@ubuntu:~$ su hduser
Password:
hduser@ubuntu:/home/udhay$
```

```
udhay@ubuntu:~$ su hduser
Password:
hduser@ubuntu:/home/udhay$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
/home/hduser/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:
09:0f:15:f2:b2:b7:5e:11:1a:6c:d3:2f:c3:09:02:15 hduser@ubuntu
The key's randomart image is:
+---[RSA 2048]---+
|   ..E.o.      |
|   . = .       |
|   = B o      |
|   O B +      |
|   . S * .    |
|   . . +     |
|   . .        |
|   . .        |
+-----+
hduser@ubuntu:/home/udhay$
```

```
hduser@ubuntu:/home/udhay$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
hduser@ubuntu:/home/udhay$ ssh localhost
```

```
Welcome to Ubuntu 15.04 (GNU/Linux 3.19.0-84-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com/
```

```
Last login: Thu Jul 15 22:00:14 2021 from localhost
hduser@ubuntu:~$
```

About the Cluster

Cluster Metrics

| | Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Total | VCores Reserved | Active Nodes | Standby Nodes |
|--|----------------|--------------|--------------|----------------|--------------------|-------------|--------------|-----------------|-------------|--------------|-----------------|--------------|---------------|
| | 0 | 0 | 0 | 0 | 0 | 0 B | 8 GB | 0 B | 0 | 8 | 0 | 1 | 0 |

Scheduler Metrics

| Scheduler Type | Scheduling Resource Type | Minimum Allocation |
|--------------------|--------------------------|-------------------------|
| Capacity Scheduler | [MEMORY] | <memory:1024, vCores:1> |

Cluster ID: 1626414170591
ResourceManager state: STARTED
ResourceManager HA state: active
ResourceManager HA zookeeper connection state: ResourceManager HA is not enabled.
ResourceManager RMStateStore: org.apache.hadoop.yarn.server.resourcemanager.recovery.NullRMStateStore
ResourceManager started on: Thu Jul 15 22:42:50 -0700 2021
ResourceManager version: 2.7.2 from b165c4fe8a74265c792ce23f546c64604acf0e41 by jenkins source checksum 2016-01-26T00:16Z
Hadoop version: 2.7.2 from b165c4fe8a74265c792ce23f546c64604acf0e41 by jenkins source checksum 2016-01-26T00:08Z

RESULT:

EXP. 10: INSTALL HADOOP 2.X AND CONFIGURE THE NAME NODE AND DATA NODE.

AIM:

PROCEDURE:

Step 7 - Modify Hadoop config files

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

1. `~/.bashrc`
2. `/usr/local/hadoop/hadoop-2.7.2/etc/hadoop/hadoop-env.sh`
3. `/usr/local/hadoop/hadoop-2.7.2/etc/hadoop/core-site.xml`
4. `/usr/local/hadoop/hadoop-2.7.2/etc/hadoop/hdfs-site.xml`
5. `/usr/local/hadoop/hadoop-2.7.2/etc/hadoop/yarn-site.xml`
6. `/usr/local/hadoop/hadoop-2.7.2/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-2.7.2
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=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export PATH=$PATH:/usr/local/hadoop/hadoop-2.7.2/bin
```

// Configure Hadoop Files

`$ cd /usr/local/hadoop/hadoop-2.7.2/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.xmsudo

```
$ cp /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/hadoop-2.7.2/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-2.7.2/bin
$ hadoop namenode -format
```

Step 9 - Start Hadoop

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

// Starting dfs services

```
$ start-dfs.sh
```

// Starting mapreduce services

```
$ start-yarn.sh
```

```
$ jps
```

Step 10 - Check Hadoop through web UI

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

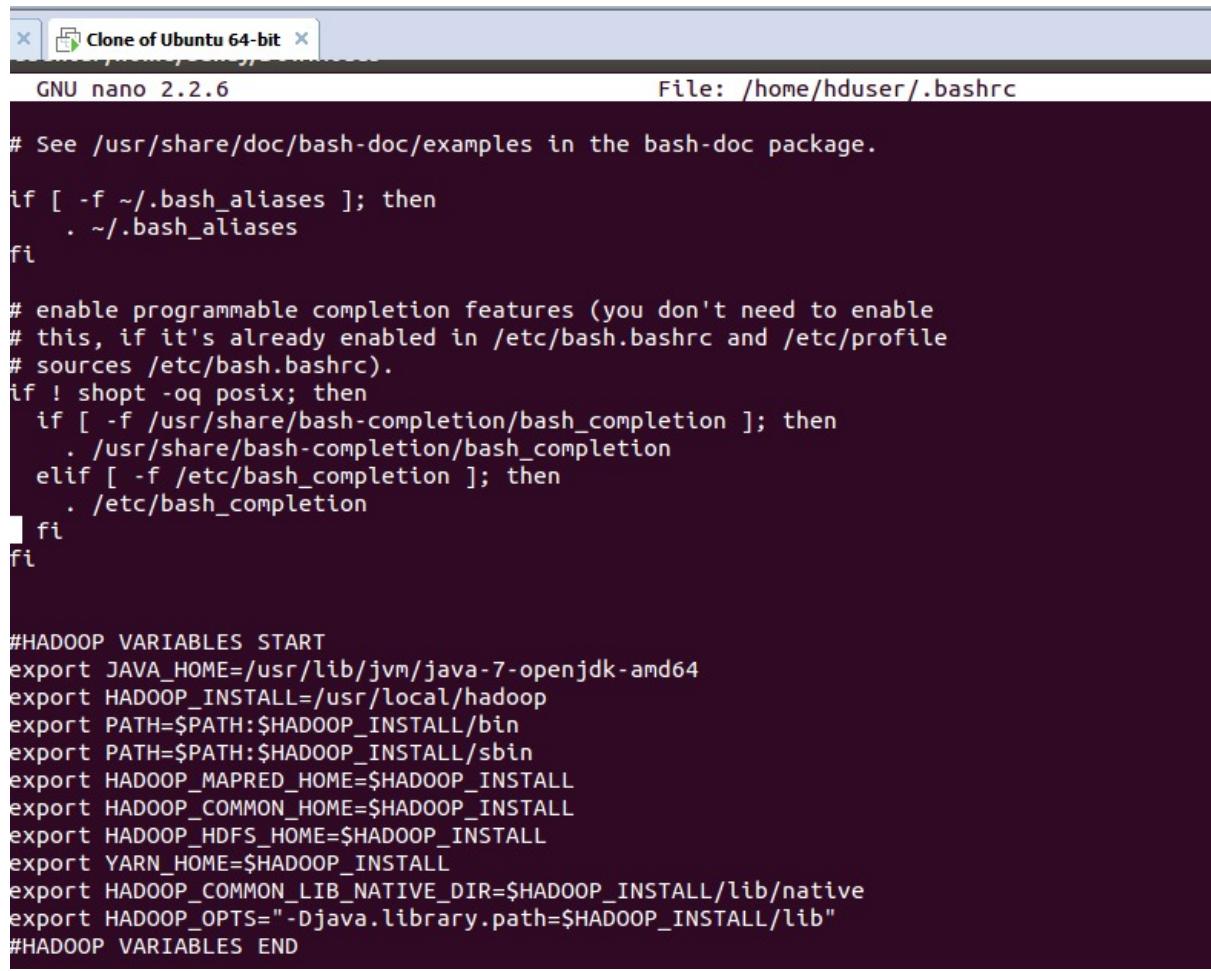
Go to browser type <http://localhost:50070> – Hadoop Namenode

Step 11 - Stop Hadoop

```
$ stop-dfs.sh
```

```
$ stop-yarn.sh
```

IMPLEMENTATION:



The screenshot shows a terminal window titled "Clone of Ubuntu 64-bit" with the file "/home/hduser/.bashrc" open in the nano editor. The code in the file is as follows:

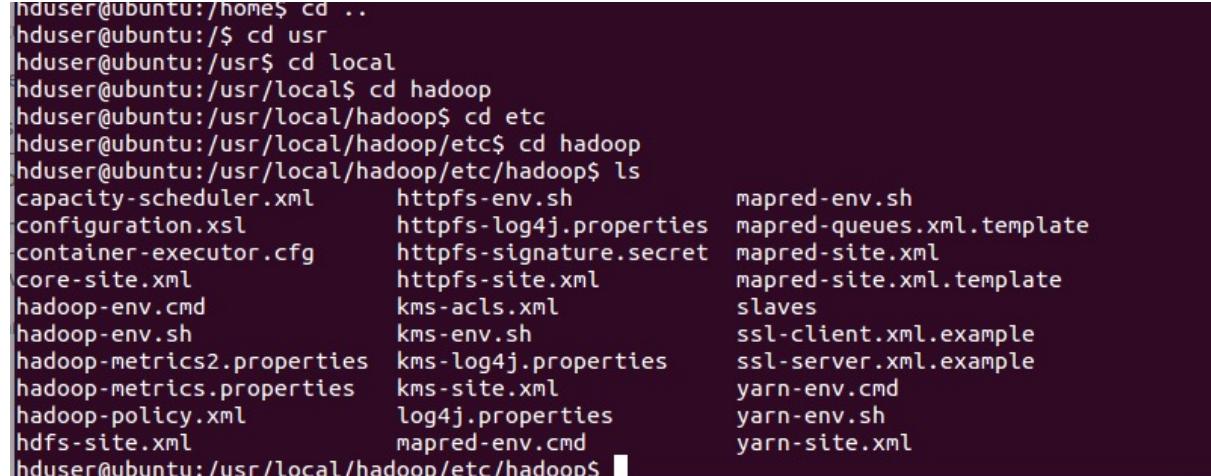
```
GNU nano 2.2.6
File: /home/hduser/.bashrc

# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

The screenshot shows a terminal window with the following command history and output:

```
hduser@ubuntu:/home$ cd ..
hduser@ubuntu:$ cd usr
hduser@ubuntu:/usr$ cd local
hduser@ubuntu:/usr/local$ cd hadoop
hduser@ubuntu:/usr/local/hadoop$ cd etc
hduser@ubuntu:/usr/local/hadoop/etc$ cd hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ ls
capacity-scheduler.xml      httpfs-env.sh          mapred-env.sh
configuration.xsl           httpfs-log4j.properties  mapred-queues.xml.template
container-executor.cfg       httpfs-signature.secret mapred-site.xml
core-site.xml                httpfs-site.xml        mapred-site.xml.template
hadoop-env.cmd              kms-acls.xml         slaves
hadoop-env.sh                kms-env.sh           ssl-client.xml.example
hadoop-metrics2.properties   kms-log4j.properties  ssl-server.xml.example
hadoop-metrics.properties    kms-site.xml         yarn-env.cmd
hadoop-policy.xml            log4j.properties     yarn-env.sh
hdfs-site.xml                mapred-env.cmd       yarn-site.xml
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$
```

```
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
GNU nano 2.2.6          File: hadoop-env.sh

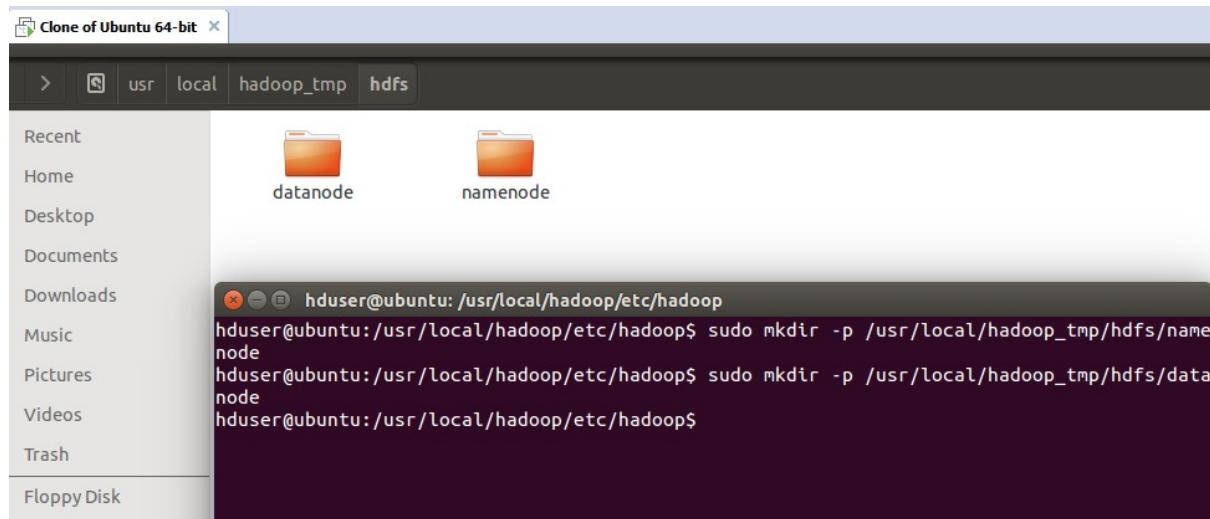
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export JAVA_HOME=${JAVA_HOME}

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
```



RESULT:

EXP. 11 LAUNCH THE HADOOP 2.X AND PERFORM MAPREDUCE PROGRAM FOR A WORD COUNT PROBLEM

AIM:

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 jar 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[] args) throws Exception {
        // TODO Auto-generated method stub
        if(args.length<2)
        {
            System.out.println("check the command line arguments");
        }
        JobConf conf=new JobConf(WordCount.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[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 - Create 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 file you created in **Step 9** (WordCountConfig).

Select an export destination (lets 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 **Permission** 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

| Name | Replication | Size | Last Modified | Permissions | Owner | Group |
|---------|-------------|------|------------------------|-------------|--------|------------|
| cloud | 0 | 0 B | 8/12/2016, 12:20:50 AM | drwxr-xr-x | hduser | supergroup |
| cse | 0 | 0 B | 8/11/2016, 1:47:41 AM | drwxr-xr-x | hduser | supergroup |
| folder | 0 | 0 B | 8/4/2016, 11:37:37 PM | drwxr-xr-x | hduser | supergroup |
| grid | 0 | 0 B | 8/11/2016, 9:52:15 PM | drwxr-xr-x | hduser | supergroup |
| output | 0 | 0 B | 8/11/2016, 9:54:38 PM | drwxr-xr-x | hduser | supergroup |
| project | 0 | 0 B | 8/11/2016, 11:54:23 PM | drwxr-xr-x | hduser | supergroup |
| tmp | 0 | 0 B | 8/4/2016, 11:40:37 PM | drwx----- | hduser | supergroup |

Step 12 – To view results in old Terminal

```
$hdfs dfs -cat /usr/local/hadoop/output/part-r-00000
```

```
hadoop1@ubuntu-1:~/project$ hadoop fs -cat /output/wordcount4/part-r-00000
.
a 1
and 1
as 1
count 1
counts 1
file 2
for 1
input 1
is 1
job 1
job. 1
map 1
returns 1
sample 1
takes 1
```

Browsing HDFS - Mozilla Firefox

Browsing HDFS

localhost:50070/explorer.html#/output

Search

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/output Go!

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|------------|--------|------------|------|-----------------------|-------------|------------|------------|
| -rw-r--r-- | hduser | supergroup | 0 B | 8/11/2016, 9:54:38 PM | 1 | 128 MB | _SUCCESS |
| -rw-r--r-- | hduser | supergroup | 44 B | 8/11/2016, 9:54:38 PM | 1 | 128 MB | part-00000 |

Step 13 - To Remove folders created using hdfs

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

RESULT: