

Hindi Vidya Prachar Samiti's

RAMNIRANJAN JHUNJHUNWALA COLLEGE

OF ARTS, SCIENCE & COMMERCE

(EMPOWERED AUTONOMOUS)

CLOUD COMPUTING



Name : Niraj Ashok Gupta

Roll No: 703.

Class : MSc Data Science and Artificial Intelligence part I (Semester II)



Ramniranjan Jhunjhunwala College of Arts, Science and Commerce

Department of Data Science and Artificial Intelligence

CERTIFICATE

This is to certify Niraj Ashok Gupta of Msc. Data Science and Artificial Intelligence Roll No 703 has successfully completed the practical of CLOUD COMPUTING during the Academic Year 2023-2024.

Date :

Prof. Sujata Kotian
Prof-In-Charge

External Examiner

INDEX

Sr No.	Practical Name	Date	Signature
1	Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows.	Feb 2, 2024	
2	Install Xen Server and connect with Xen Client.	Feb 7, 2024	
3	Install ExSi Server and connect with vSphere.	Feb 7, 2024	
4	Install a C compiler in the virtual machine created using virtual box and execute a simple program	Feb 9, 2024	
5	To attach a virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine.	Feb 9, 2024	
6	Transfer files from one host machine to another virtual machine	Feb 14, 2024	
7	Installing and configuring the required platform for Google App Engine	Feb 14, 2024	
8	Using the software like JDK 1.7/1.8. design and develop web application using MVC Framework.	Feb 16, 2024	
9	Installing and configuring Dockers in local host and running multiple images on a Docker Platform.	Feb 16, 2024	
10	Install and Configure Puppet	Feb 21, 2024	
11	Install and Configure IBM Bluemix	Feb 21, 2024	
12	Developing an ASP.NET web based application on the Azure platform	Feb 23, 2024	

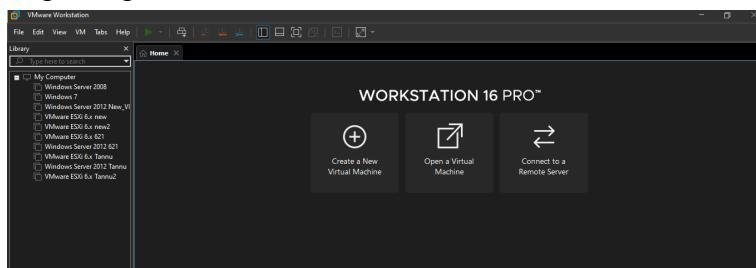
PRACTICAL 1

Aim: Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows.

→ Downloading Ubuntu

Step 1: Download the Ubuntu OS

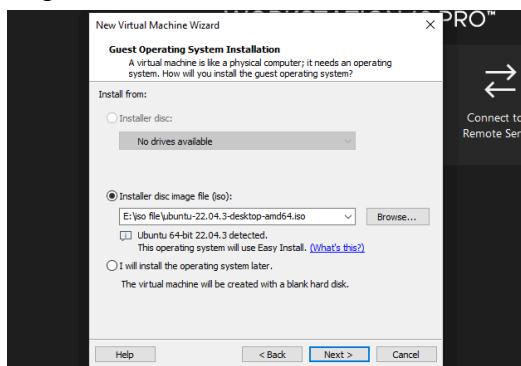
Step 2: Open VMware Workstation -> Click on create new virtual machine



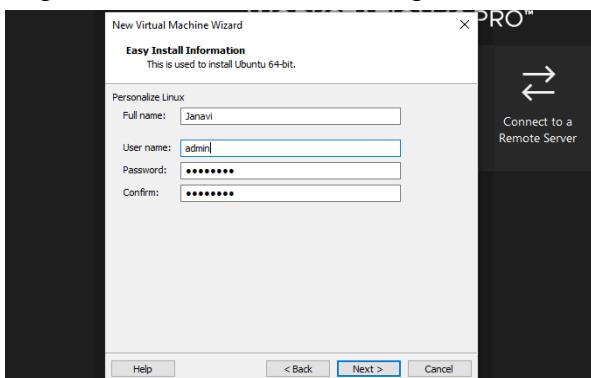
Step 3: Click Next



Step 4: Browse the downloaded Ubuntu file and click next



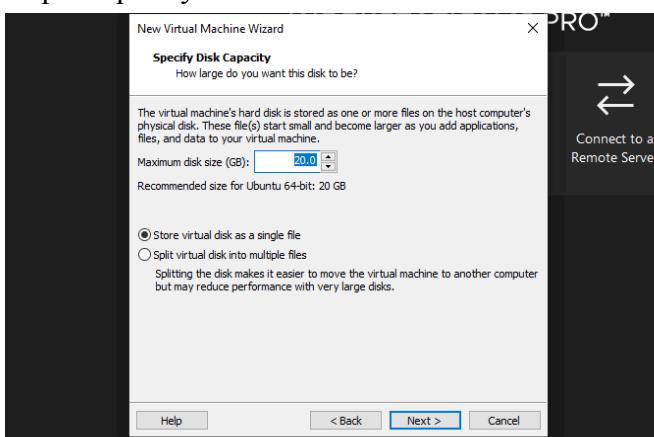
Step 5: Create an username and password and click next



Step 6: Choose the location to use your virtual machine and click next

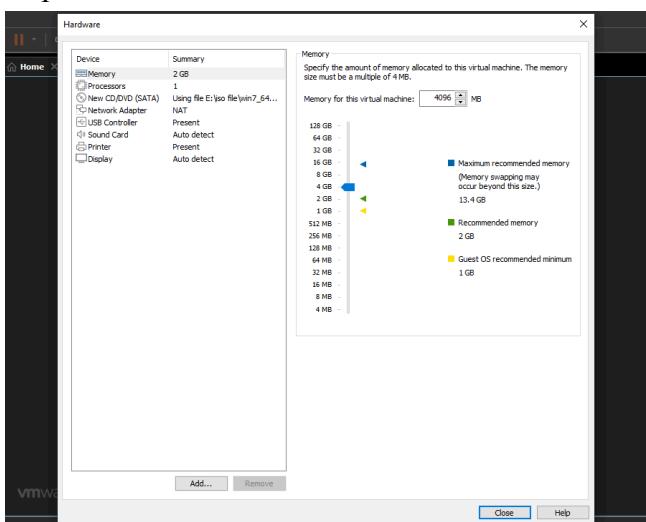
Step 7: Select create a new virtual disk and click next

Step 8: Specify the disk size and click next

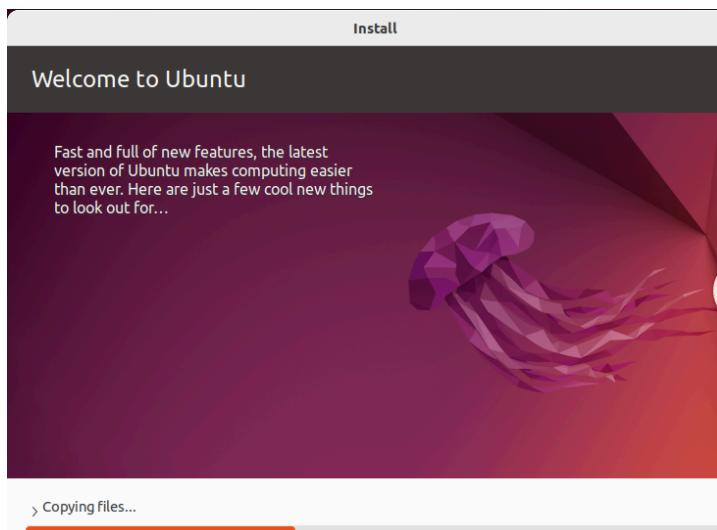
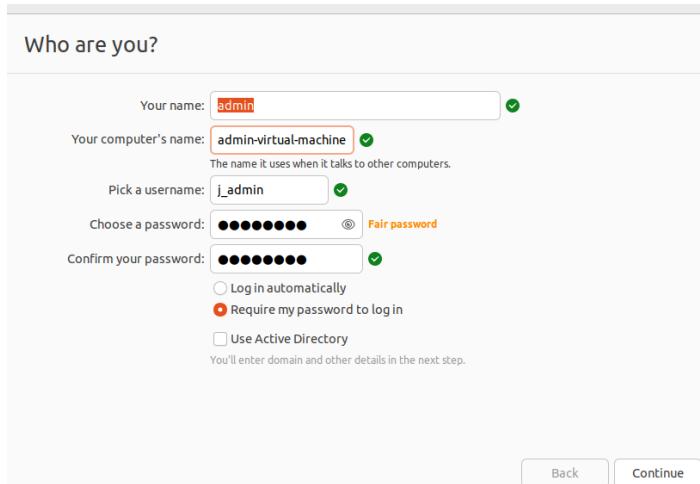


Step 9: Customize hardware

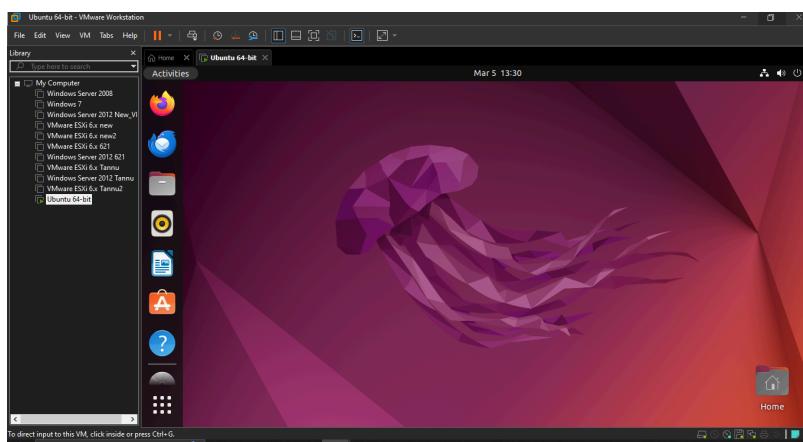
Step 10: Click Finish



Step 11: Installing Ubuntu on VMware and unzipping files

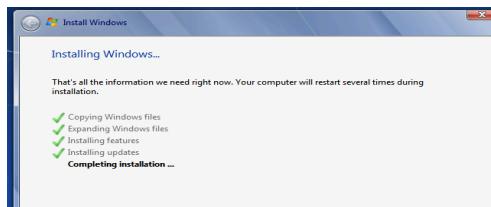
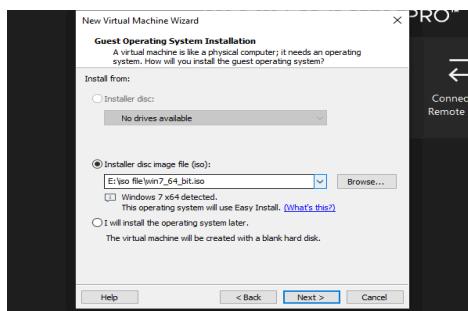


Step 25: Thus we have installed VMware Workstation with different flavours of linux on top of windows



→ Downloading Windows

(Steps for installing Windows OS in VMWare Workstation are the same as that of downloading and installing Ubuntu, with the only exception of selecting the proper iso file for the same!!)

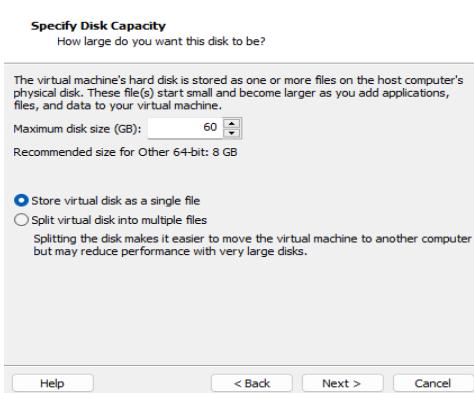
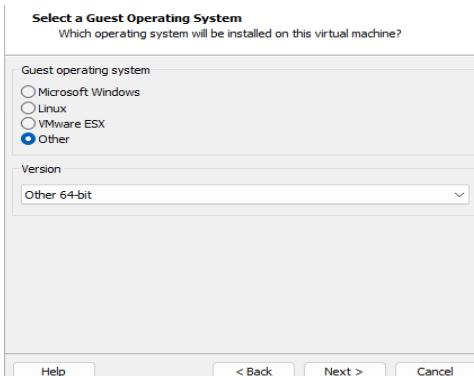
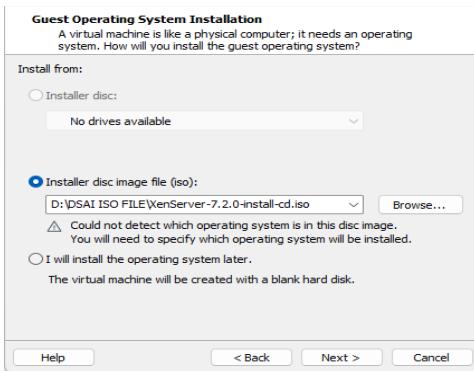


Signature: _____

PRACTICAL 2

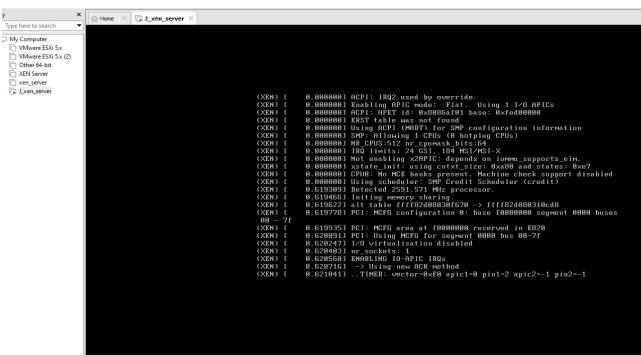
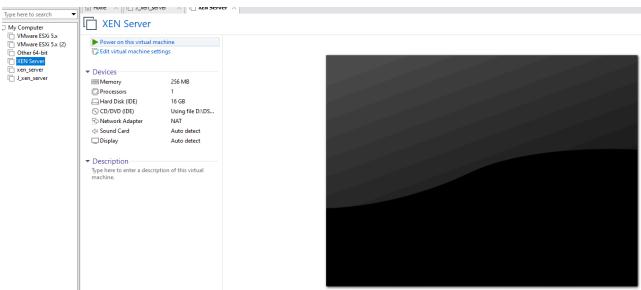
Aim: Install Xen Server and connect with Xen Client.

(Steps for creating a Xen server VM are the same as that for Ubuntu and Windows, with the only exception of selecting the proper iso file for the same!!)

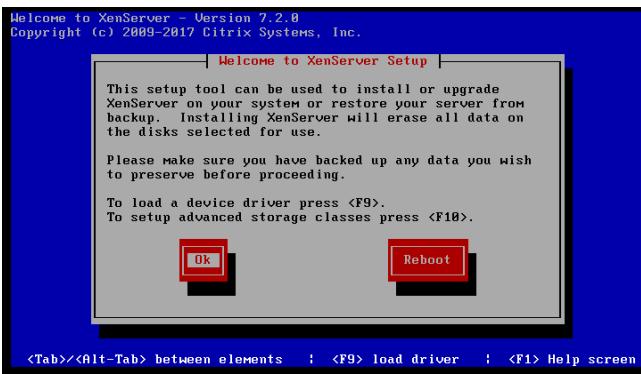


→ Steps for installing Xen server

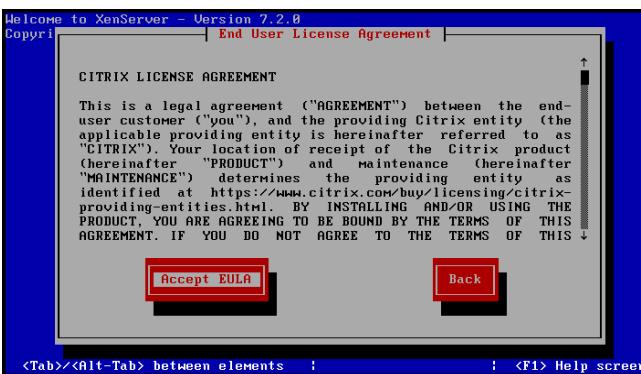
Step 1: Power on the virtual machine



Step 2: Click on Ok



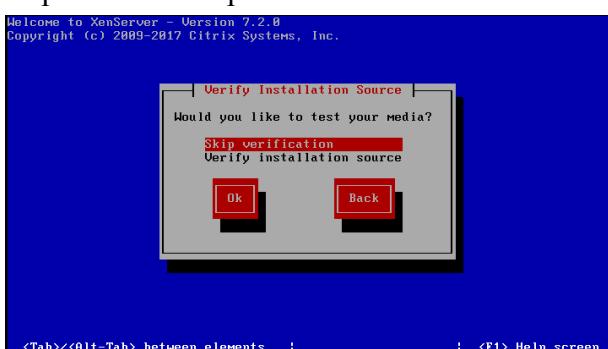
Step 3: Click on Accept EULA



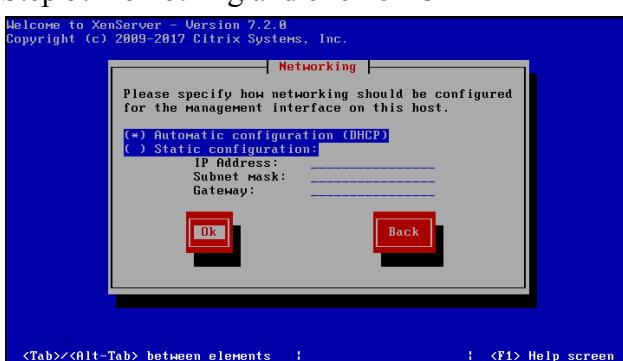
Step 4: Select Local media -> Click on Ok



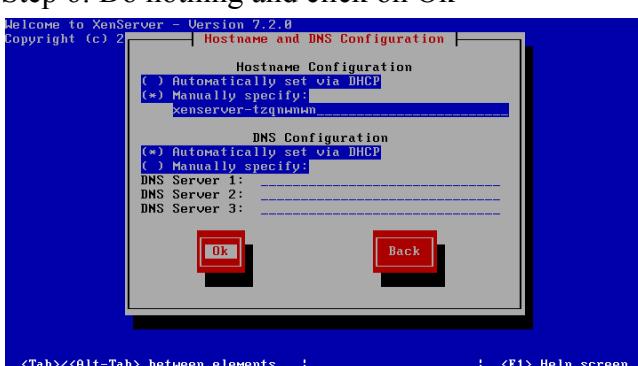
Step 4: Select Skip verification -> Click on Ok



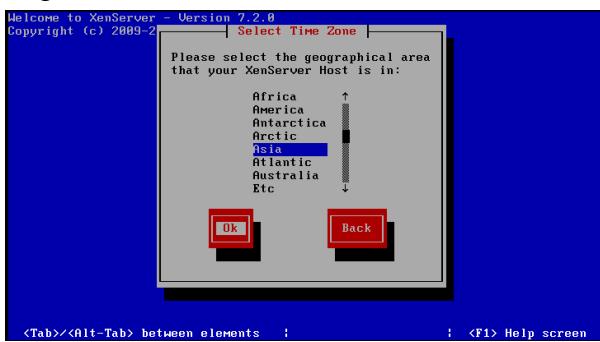
Step 5: Do nothing and click on Ok



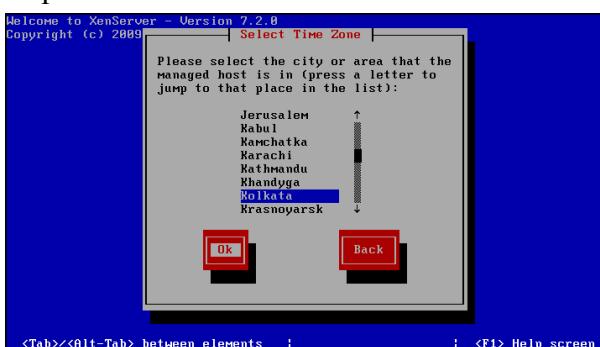
Step 6: Do nothing and click on Ok



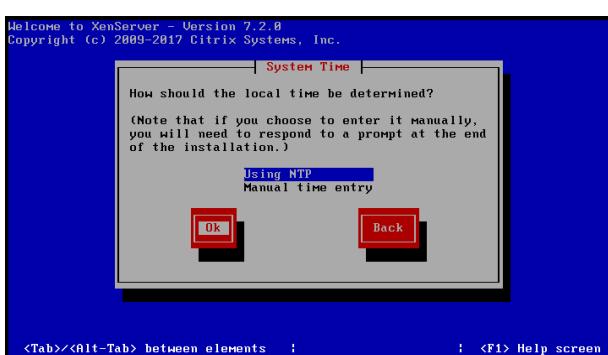
Step 7: Select Asia -> Click on Ok



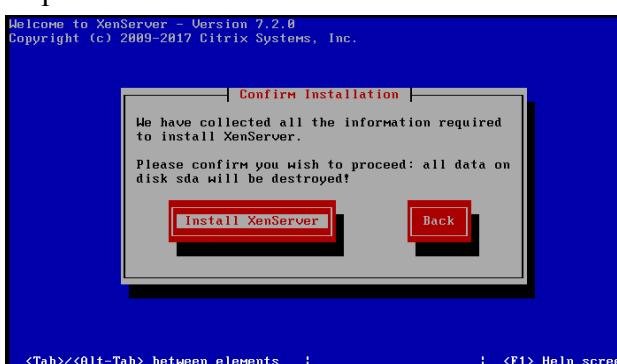
Step 8: Select Kolkata-> Click on Ok



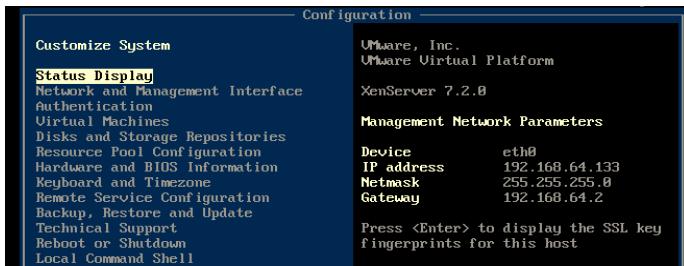
Step 9: Select Using NTP -> Click on Ok



Step 10: Install the Xen server

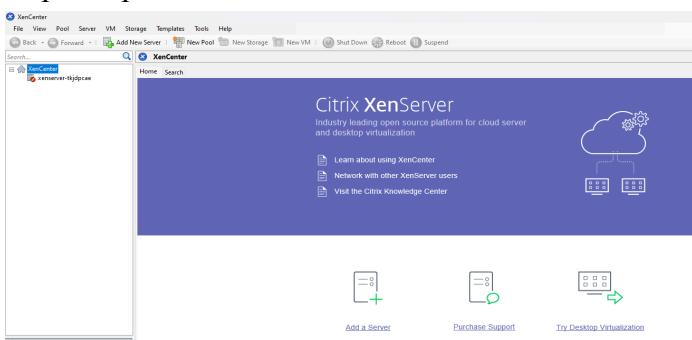


Step 11: Copy your system's IP address from over here

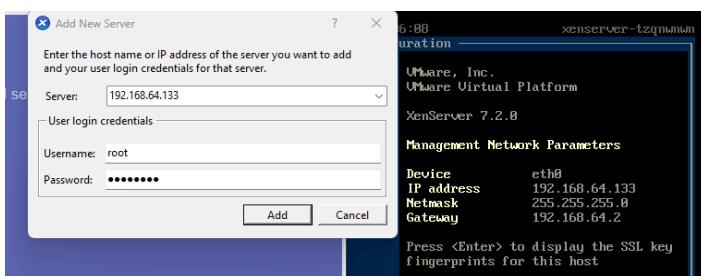


→ Steps for connecting Xen server with the Xen client

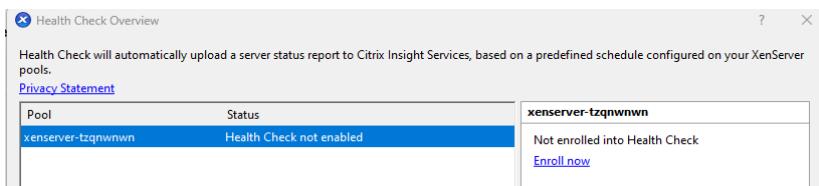
Step 1: Open Citrix XenCenter -> Click on Add a Server



Step 2: Enter the IP address you'd copied along with a suitable username & password ->
Click on Add



Now, you've got your desired result!!

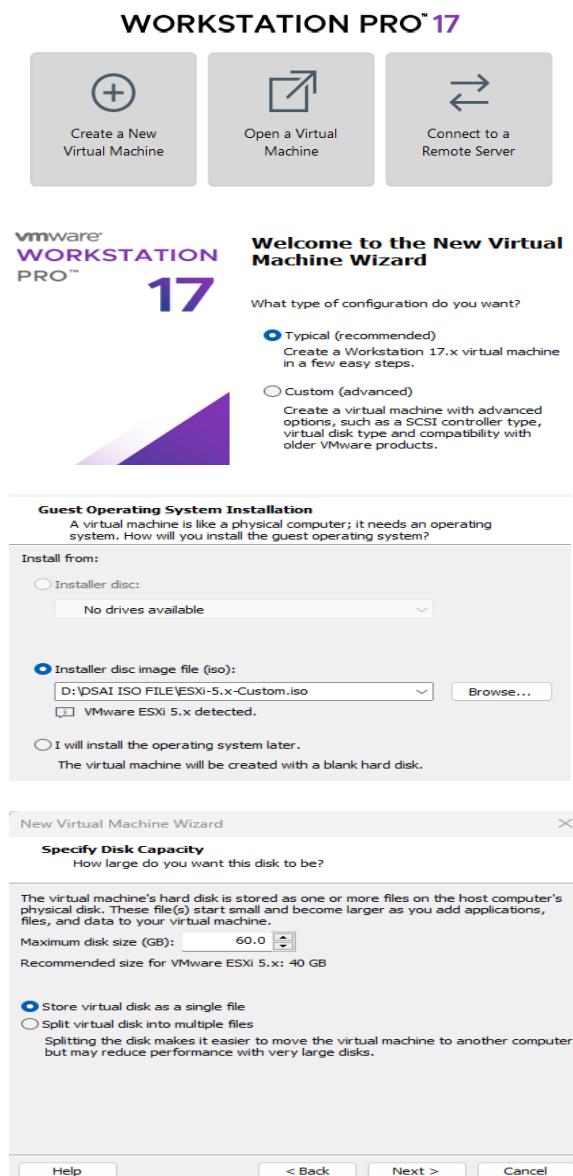


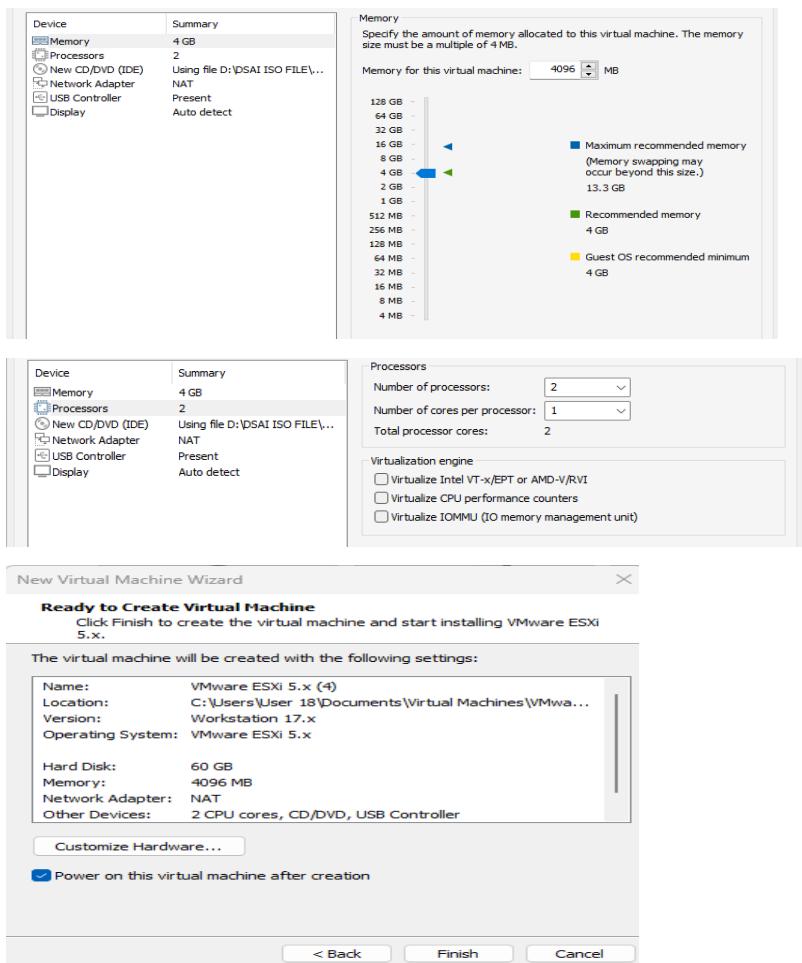
Signature: _____

PRACTICAL 3

Aim: Install ExSi Server and connect with vSphere.

(Steps for creating an ESXI VM are the same as that for Ubuntu and Windows, with the only exception of selecting the proper iso file for the same, and disabling the option “Virtualize Intel VT-x/EPT or AMD-v/RVI” in the Customize hardware -> Processors section!!)



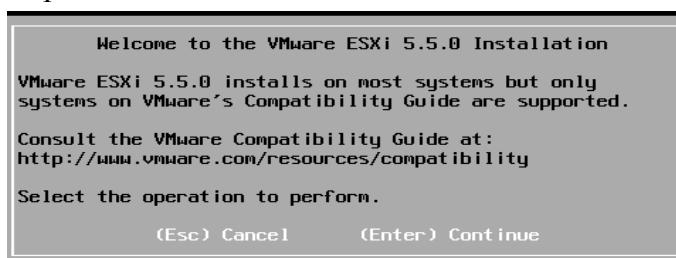


→ Steps for installing ESXI server

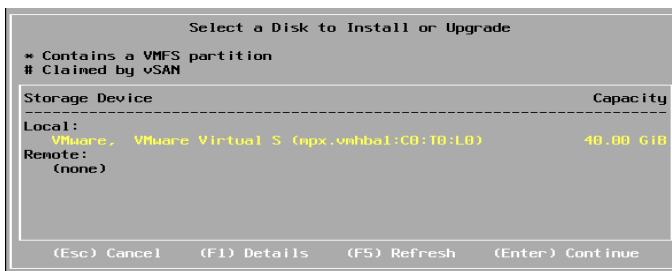
A similar interface will appear



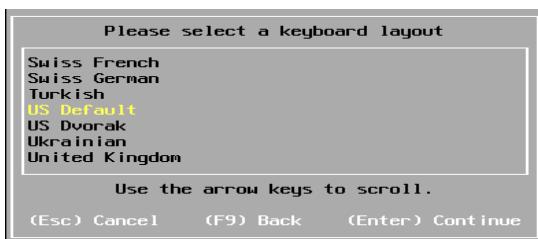
Step 1: Hit enter



Step 2: Hit enter



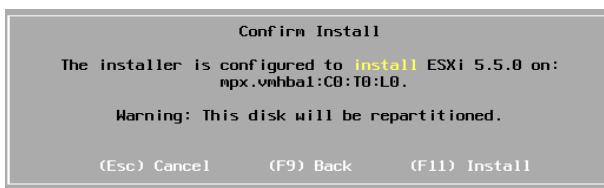
Step 3: Select US Default -> Hit enter



Step 4: Default username is root; set a suitable password -> Hit enter



Step 5: Hit F11 to install ESXI server

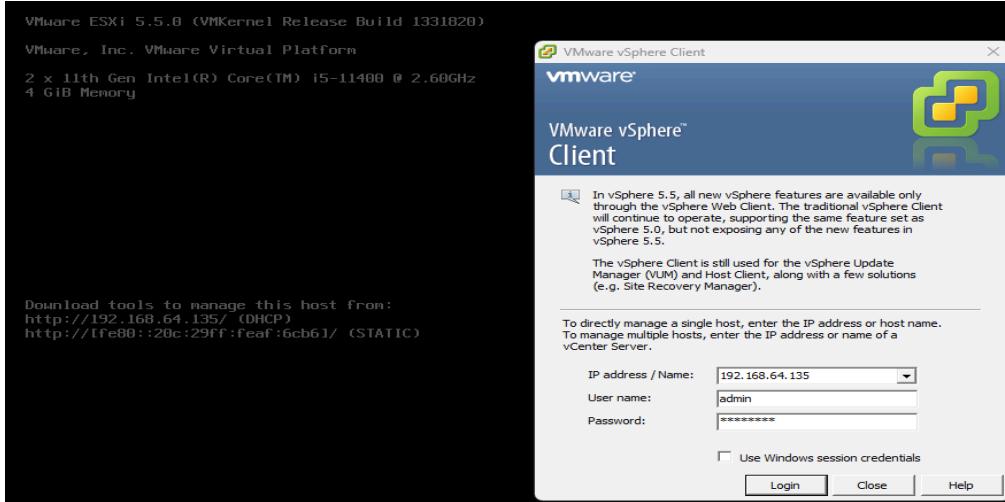


Step 6: Hit enter to reboot



→ Steps for connecting with vSphere client

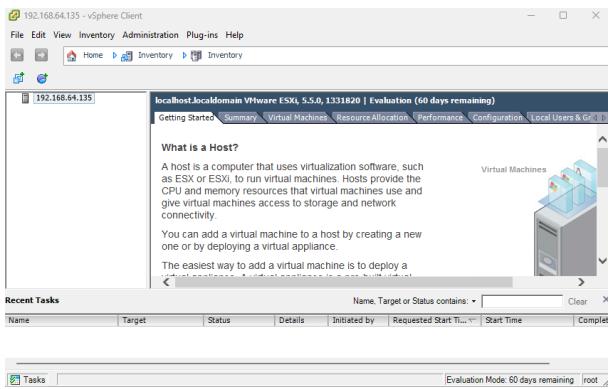
Step 1: Open VMware vSphere Client -> Enter the ip address displayed on your ESXI server -> Enter the username and password you'd set -> Click on Login



Step 2: Click on Ignore



Here's the desired output!

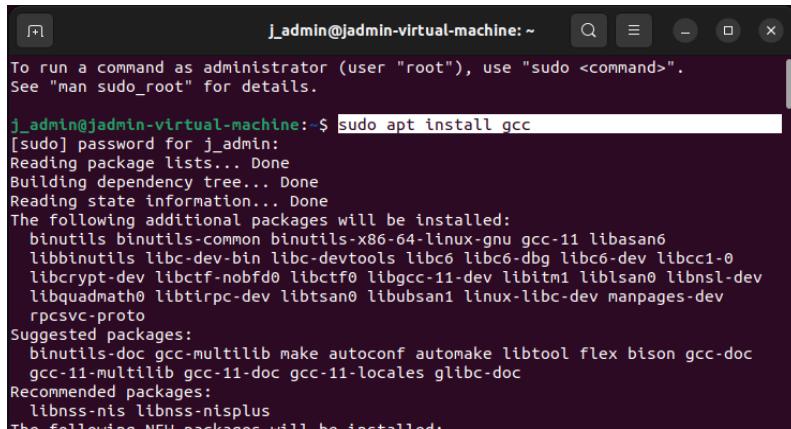


Signature: _____

PRACTICAL 4

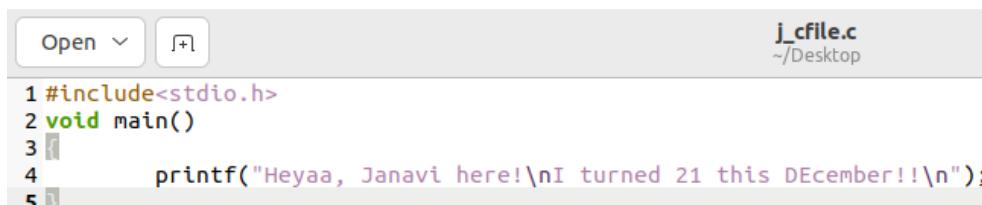
Aim: Install a C compiler in the virtual machine created using virtual box and execute a simple program.

Step 1: Open the terminal on Ubuntu and install C compiler – “sudo apt install gcc”



```
j_admin@admin-virtual-machine:~$ sudo apt install gcc
[sudo] password for j_admin:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu gcc-11 libasan6
libbinutils libc-dev-bin libc-devtools libc6 libc6-dbg libc6-dev libcc1-0
libcrypt-dev libctf-nobfd0 libctf0 libgcc-11-dev libitm1 liblsan0 libnsl-dev
libquadmath0 libtirpc-dev libtsan0 libubsan1 linux-libc-dev manpages-dev
rpcsvc-proto
Suggested packages:
binutils-doc gcc-multilib make autoconf automake libtool flex bison gcc-doc
gcc-11-multilib gcc-11-doc gcc-11-locales glibc-doc
Recommended packages:
libnss-nis libnss-nisplus
The following NEW packages will be installed:
```

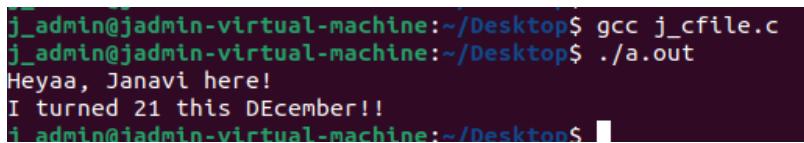
Step 2: Once you’re done with the installation, open the editor and type a simple C program and save it



```
j_cfile.c
~/Desktop
1 #include<stdio.h>
2 void main()
3 {
4     printf("Heyaa, Janavi here!\nI turned 21 this DEcember!!\n");
5 }
```

Step 3: Compile and run the C program

- To compile the program: gcc file_name.c
- To run the program: ./a.out



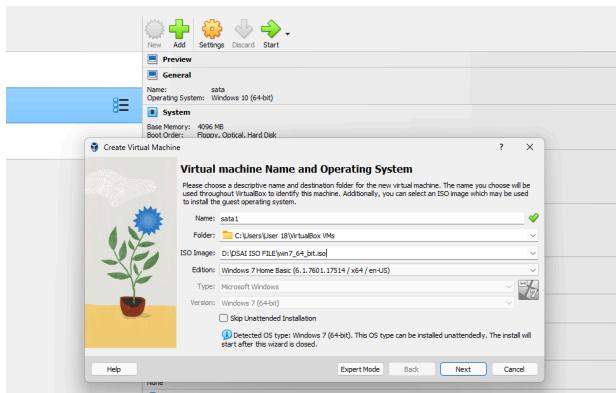
```
j_admin@admin-virtual-machine:~/Desktop$ gcc j_cfile.c
j_admin@admin-virtual-machine:~/Desktop$ ./a.out
Heyaa, Janavi here!
I turned 21 this DEcember!!
j_admin@admin-virtual-machine:~/Desktop$
```

Signature: _____

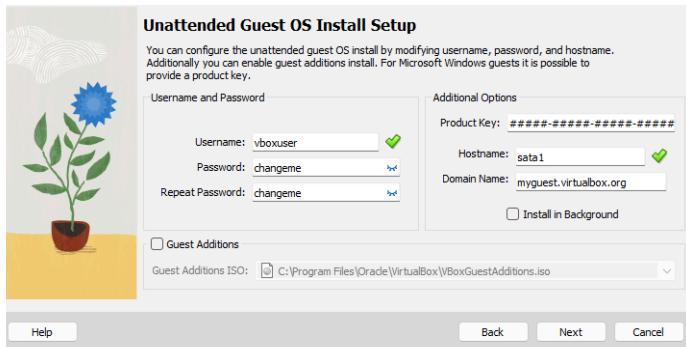
PRACTICAL 5

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

Step 1: Open VirtualBox -> New virtual machine -> Give a suitable name to it -> ISO Image -> Other -> Windows 7 iso file -> Next



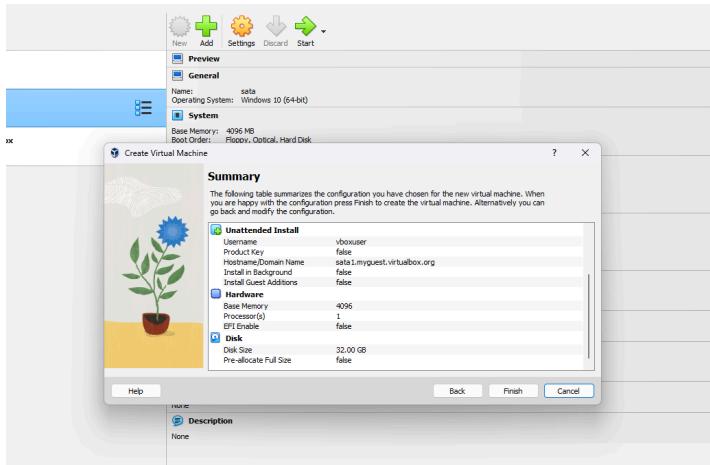
Step 2: Set suitable username and password -> Next



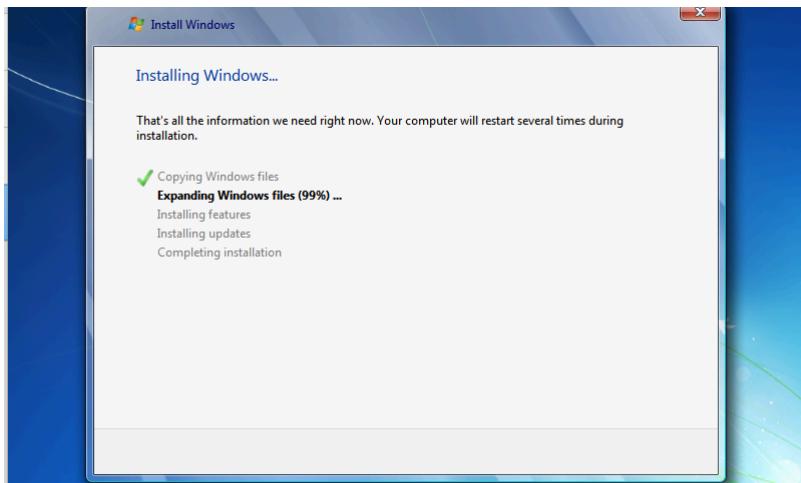
Step 3: Set the base memory to 4 MB -> Next



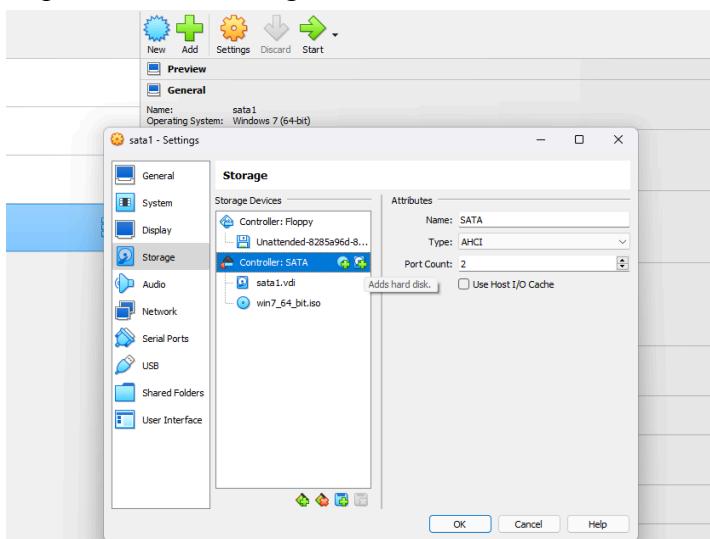
Step 4: Click on Finish



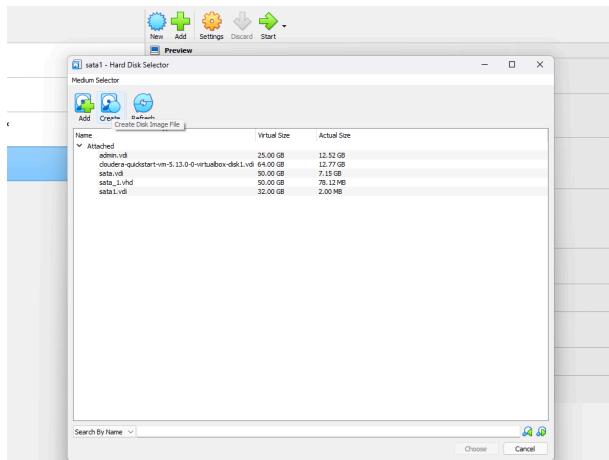
Step 5: Install Windows 7 OS



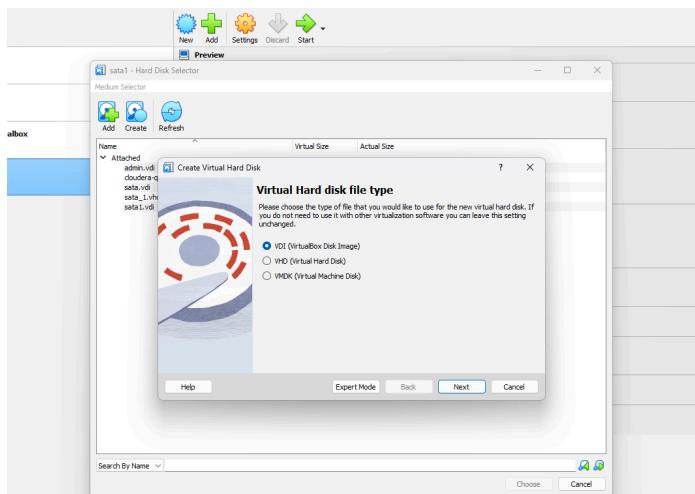
Step 6: Click on Storage -> Controller: SATA -> Click on the 2nd icon next to it



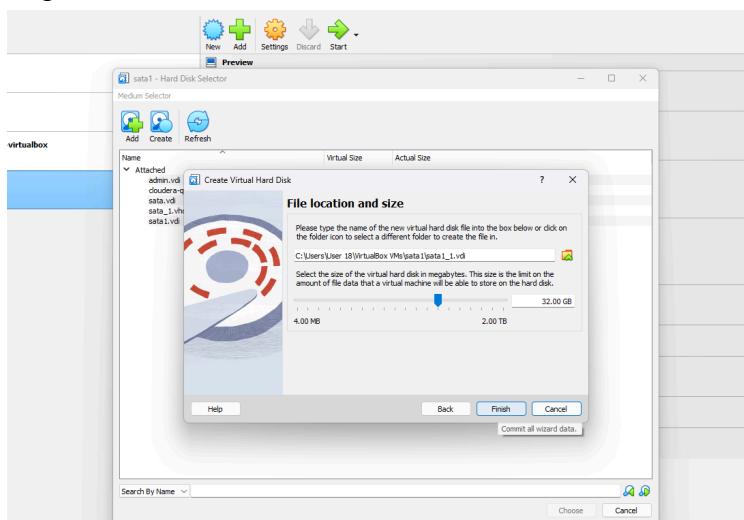
Step 7: Click on Create



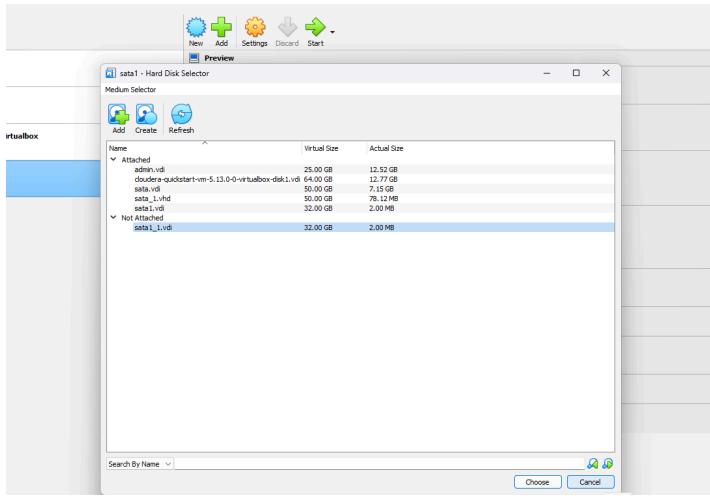
Step 8: Select Virtual Hard Disk (2nd option) -> Next



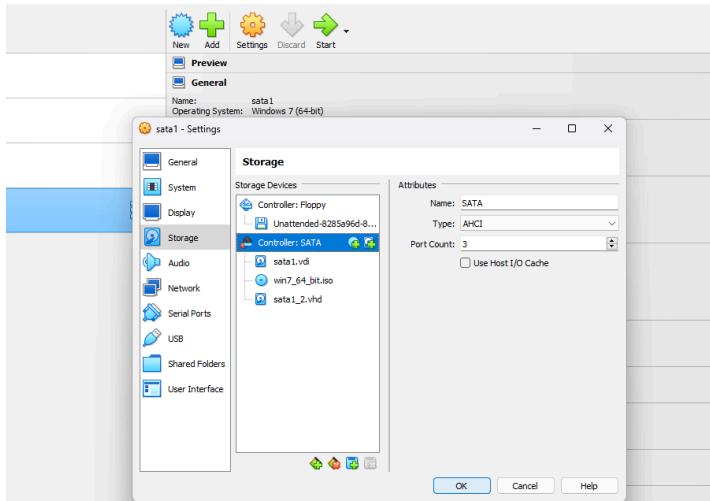
Step 9: Click on Finish



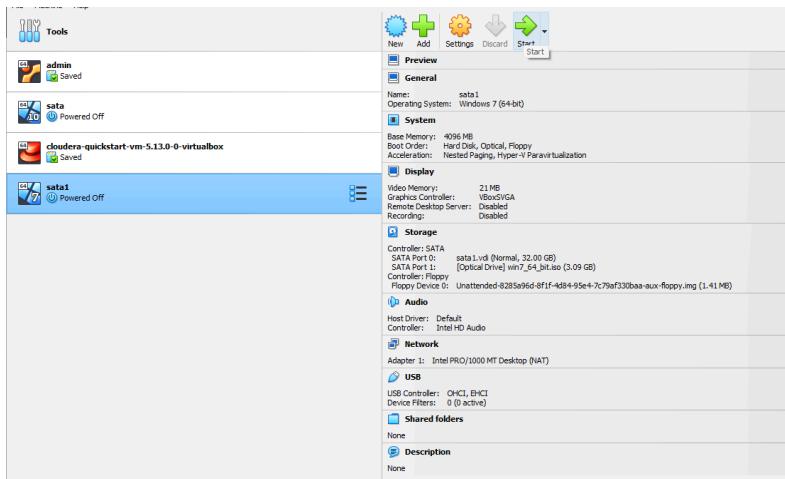
Step 10: Click on Choose



Step 11: Click on OK

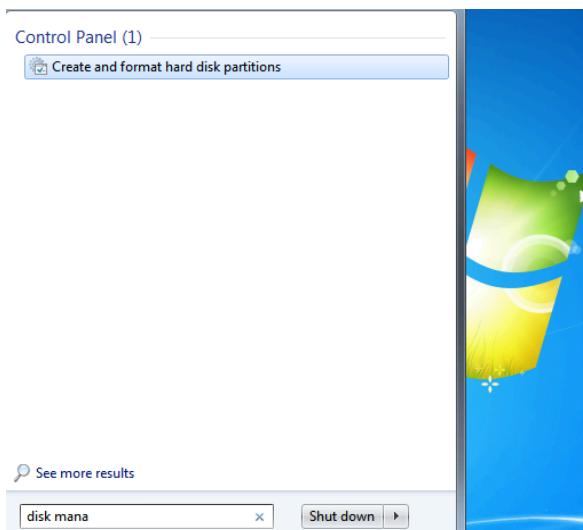


Step 12: Start your virtual machine

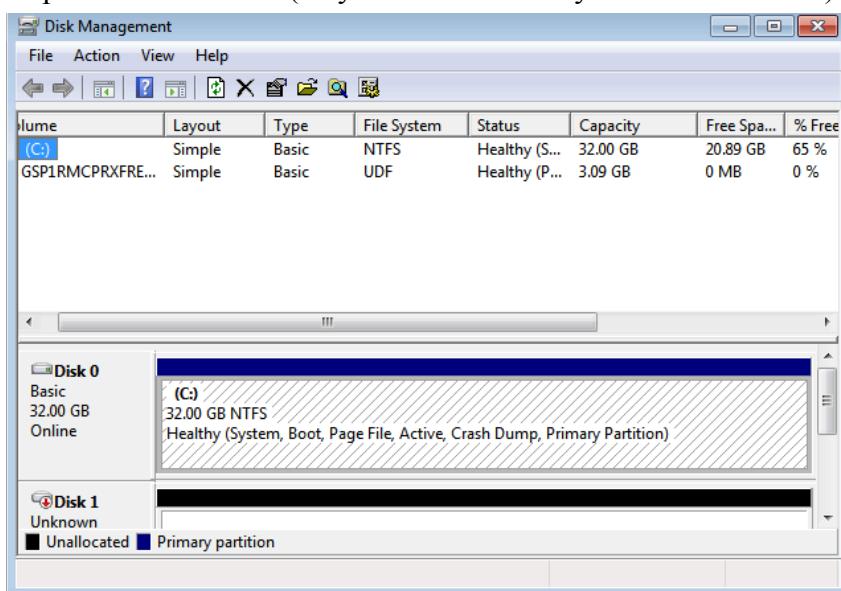




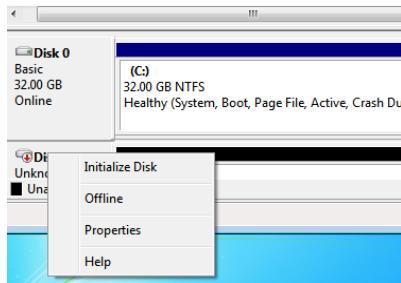
Step 13: Search for Disk Management -> Open Create and format hard disk partitions



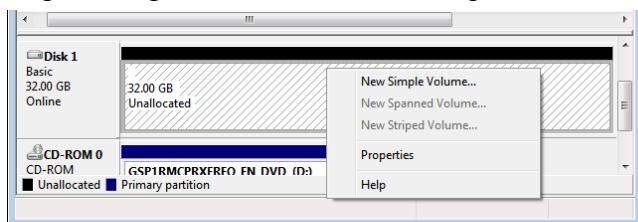
Step 14: Go to Disk 1 (Any disk with which you see a red arrow)



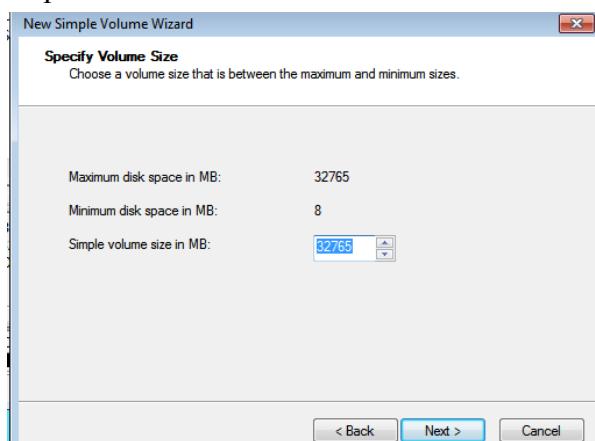
Step 15: Right click on Disk 1 -> Initialize disk



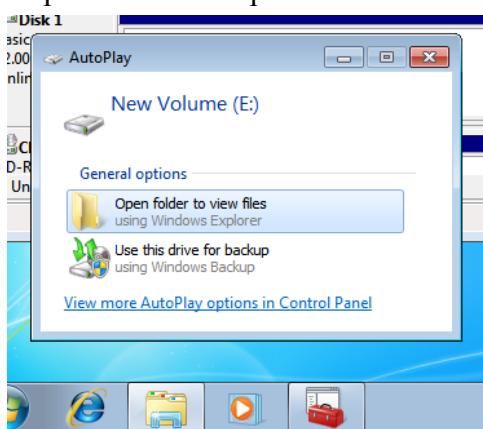
Step 16: Right click over it's vacant portion -> New Simple Volume



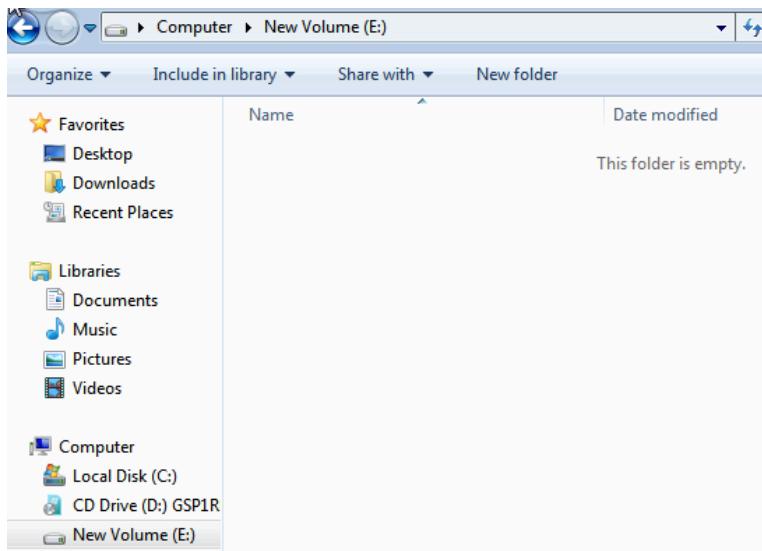
Step 17: Click on Next -> Finish



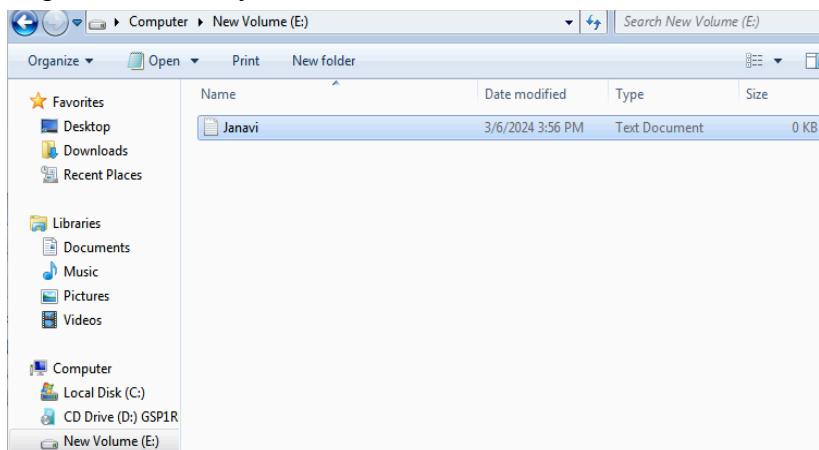
Step 18: Click on open folder to view files



Step 19: Go to File Manager -> Computer -> E Drive (You'll see that this newly created partition doesn't contain any folder or file)



Step 20: Create any folder or file in it



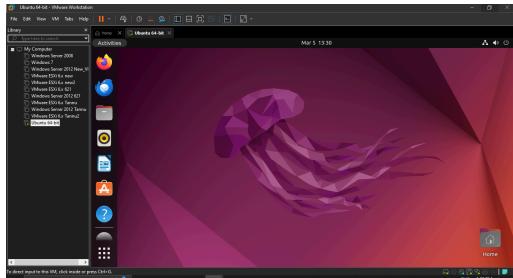
For the final result, you next need to shutdown your virtual machine, and again start it and go to File Manager -> Computer -> E Drive again. If your file or folder is successfully saved over there, you got it!!

Signature: _____

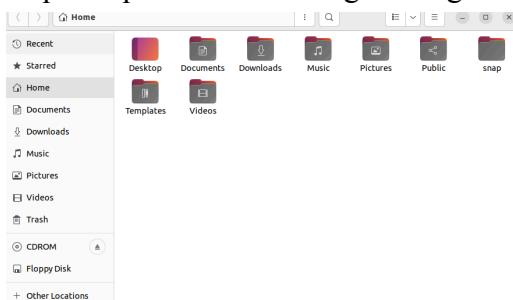
PRACTICAL 6

Aim: Transfer files from one host machine to another virtual machine.

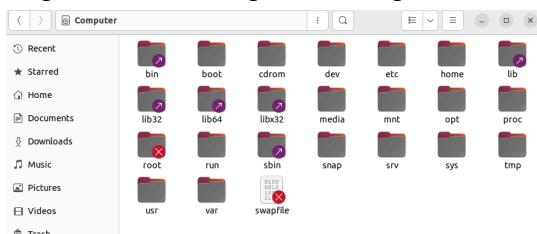
Step 1: Open the ubuntu in VM



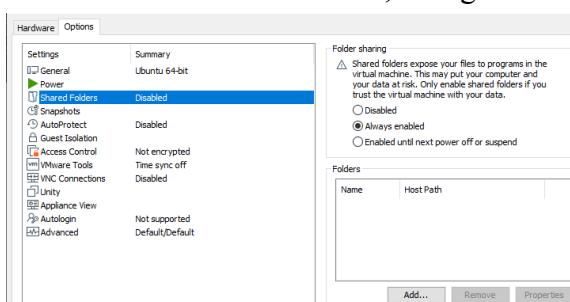
Step 2: Open the file manager and go to Other Locations



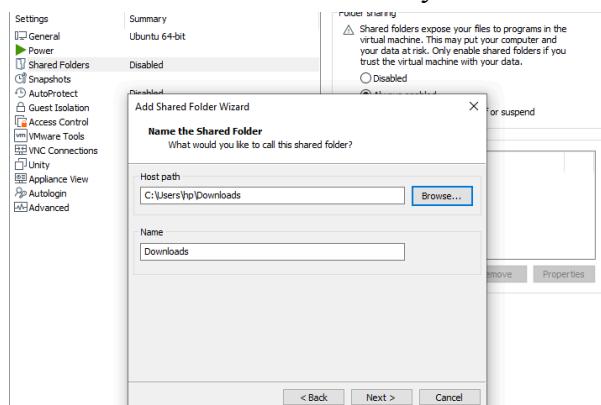
Step 3: Go to Computer and open the folder named “mnt”. Initially the folder will be empty



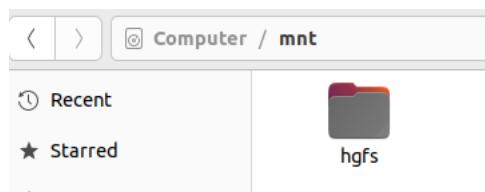
Step 4: Now right click ubuntu 64-bit(VM name) and select properties, then go to options tab. Then select the shared folders, change the radio button to “Always enabled” and click on Add



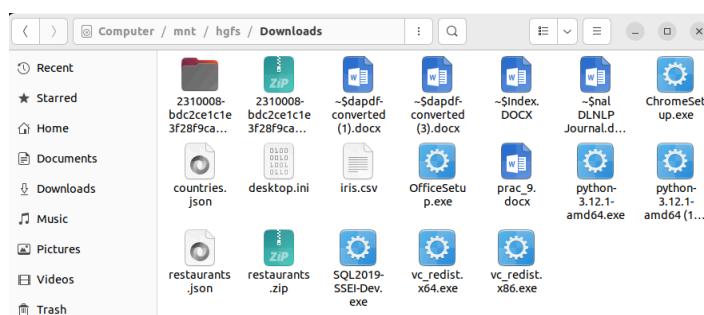
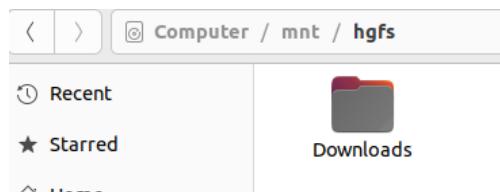
Step 5: Click Next and select the folder/file that is to be shared to VM in the Host path. Then, click on Ok and check Read Only and click Finish



Step 6: Now in the same location “Computer>mnt”, a folder named hgfs has been created; open this folder



And here, you've got your desired result; your selected files/folders are now added to the hgfs folder!!



Signature: _____

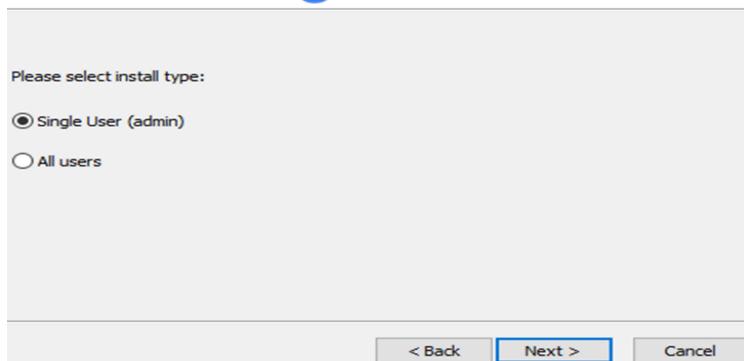
PRACTICAL 7

Aim: Installing and configuring the required platform for Google App Engine.

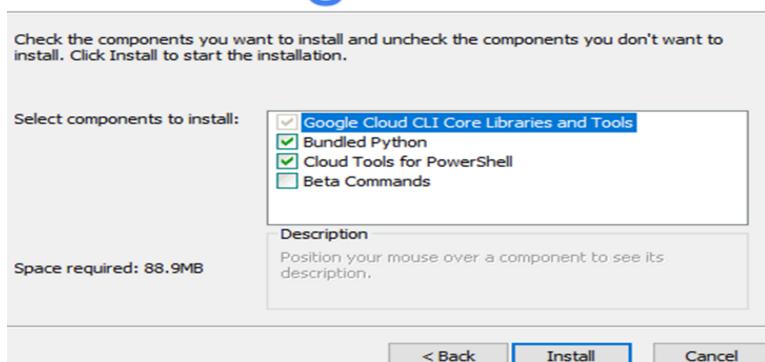
Step 1: Start installing Google SDK using it's base file or by searching “google cloud sdk installer.exe”

This PC > New Volume (D:)			
Name	Date modified	Type	Size
cloudera-quickstart-vm-5.13.0-0-virtualb...	23-10-2017 16:34	File folder	
Excel	13-02-2024 08:01	File folder	
iso file	11-03-2024 12:31	File folder	
USB Drive	01-12-2023 14:15	File folder	
cloudera-quickstart-vm-5.13.0-0-virtualb...	21-12-2023 09:22	WinRAR ZIP archive	57,55,879 KB
VirtualBox-7.0.10-158379-Win (3)	10-10-2023 13:09	Application	1,08,300 KB
GoogleCloudSDKInstaller	14-02-2024 07:54	Application	149 KB

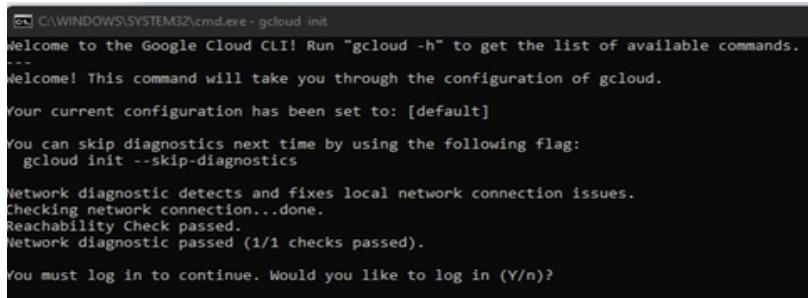
Step 2: Select single user -> Click next



Step 3: Click on install



Step 4: Install Google SDK by keeping the default settings and proceeding further >> Open the gcloud cmd



```
C:\WINDOWS\SYSTEM32\cmd.exe - gcloud init
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
...
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

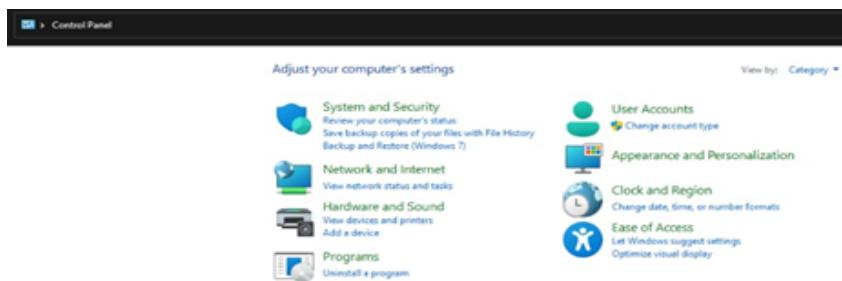
You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

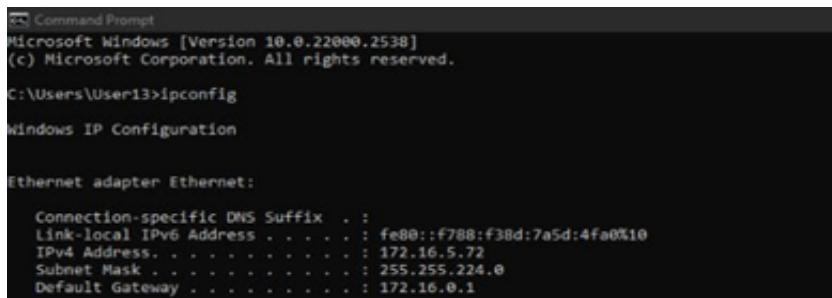
You must log in to continue. Would you like to log in (Y/n)?
```

Step 5: Configure the Network if it's required

- Go to “Control Panel” >> User Account >> Credential Manager >> Windows Credential >> Go to “Add a generic credential” >> In this it will ask for IPv4 Address



Step 6: Use “ipconfig” command in the normal CMD to get the IPv4 address



```
Command Prompt
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User13>ipconfig

Windows IP Configuration

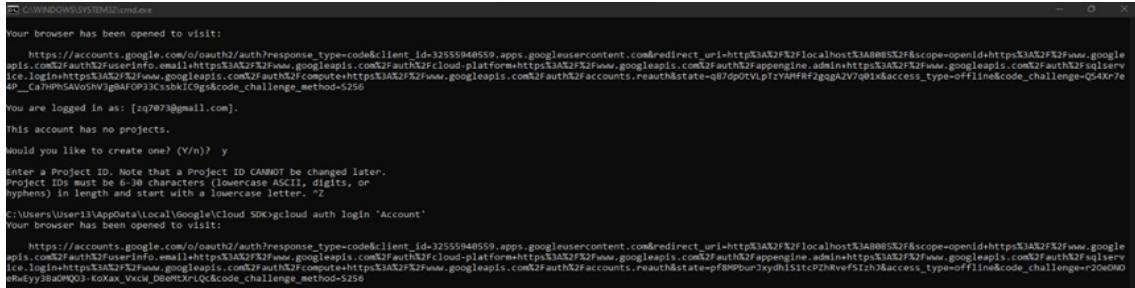
Ethernet adapter Ethernet:

  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::f788:f38d:7a5d:4fa0%10
  IPv4 Address. . . . . : 172.16.5.72
  Subnet Mask . . . . . : 255.255.224.0
  Default Gateway . . . . . : 172.16.0.1
```

Step 7: Setting IPv4 address, Username and password to set up the configuration (Mostly username and password is either admin or root)



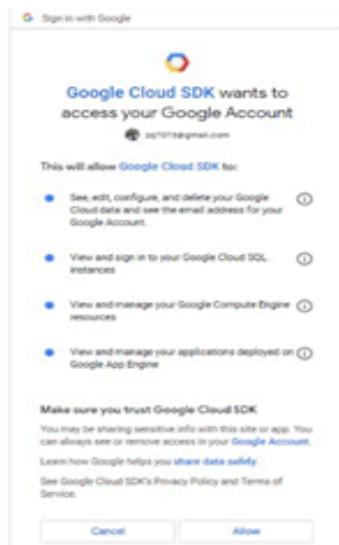
Step 8: After successfully establishing the configuration >> Go to gcloud cmd >> make use of “gcloud auth login ‘Account’ to log-in OR can even directly through the web page



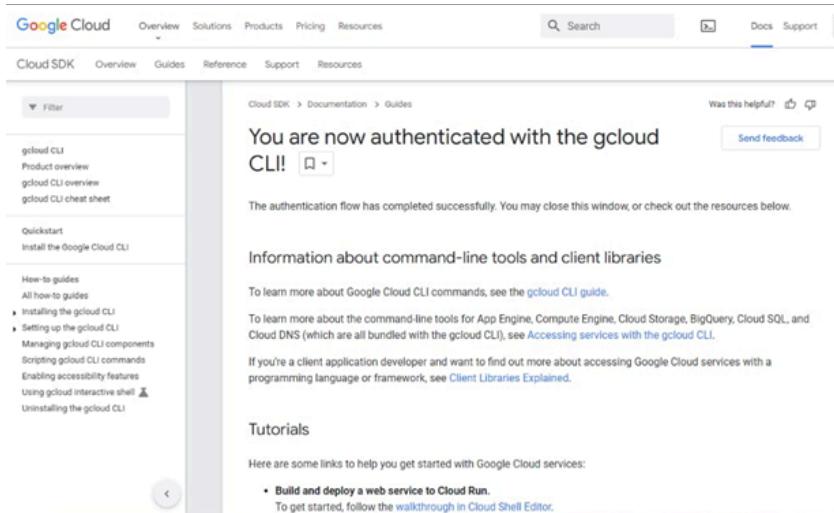
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.readonly&state=q87dp0VlpTzYAHf2gogA2V7q01x&access_type=offline&code_challenge_method=S4Xr7e4P_Cn7PhSAVo5HVigbAFOPj3csbk1Gg&code_challenge=5z56

You are logged in as: [zq7073@gmail.com].
This account has no projects.
Would you like to create one? (Y/n)? y
Enter a Project ID. Note that a Project ID CANNOT be changed later.
Project IDs must be 6-30 characters (lowercase ASCII, digits, or hyphens) in length and start with a lowercase letter. ?2
C:\Users\user13\AppData\Local\Google\Cloud SDK>gcloud auth login 'Account'
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.readonly&state=pf0Mpburxydh1S1tcP0Rvef51zh3&access_type=offline&code_challenge=r20eONO&uddy3BaQMQ3-KoKax_Vxw_DbeHrx/LQC&code_challenge_method=5z56

Step 9: Make use of google account (gmail) to login into the gcloud



Step 10: After successfully logging in >> Go to the “walkthrough in Cloud Shell Editor” under “Tutorials”



Google Cloud Overview Solutions Products Pricing Resources

Cloud SDK Overview Guides Reference Support Resources

Cloud SDK > Documentation > Guides

Was this helpful?  Send feedback

You are now authenticated with the gcloud CLI! 

The authentication flow has completed successfully. You may close this window, or check out the resources below.

Information about command-line tools and client libraries

To learn more about Google Cloud CLI commands, see the [gcloud CLI guide](#).

To learn more about the command-line tools for App Engine, Compute Engine, Cloud Storage, BigQuery, Cloud SQL, and Cloud DNS (which are all bundled with the gcloud CLI), see [Accessing services with the gcloud CLI](#).

If you're a client application developer and want to find out more about accessing Google Cloud services with a programming language or framework, see [Client Libraries Explained](#).

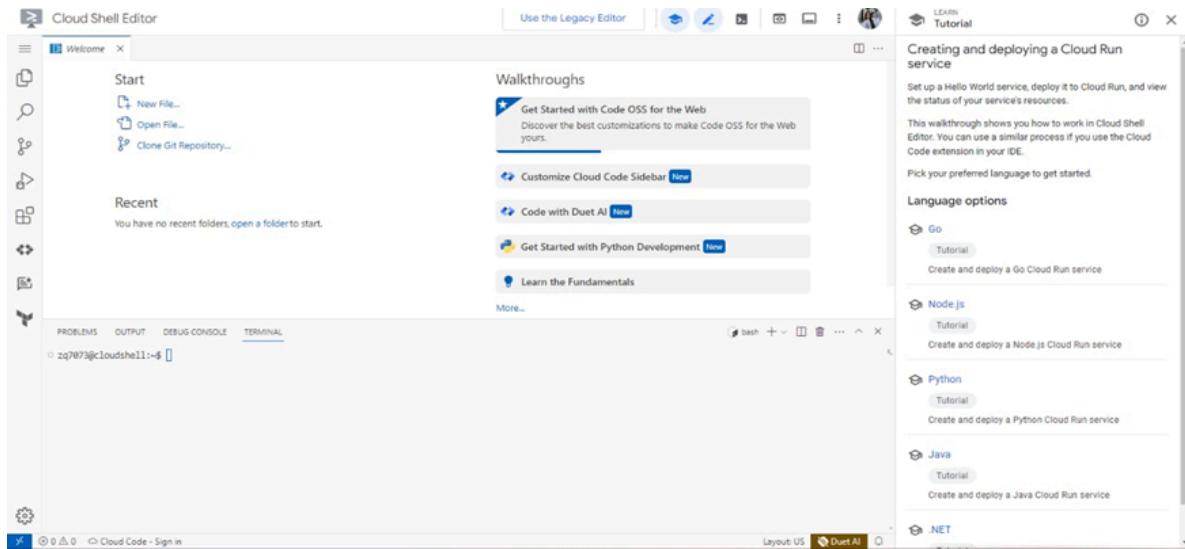
Tutorials

Here are some links to help you get started with Google Cloud services:

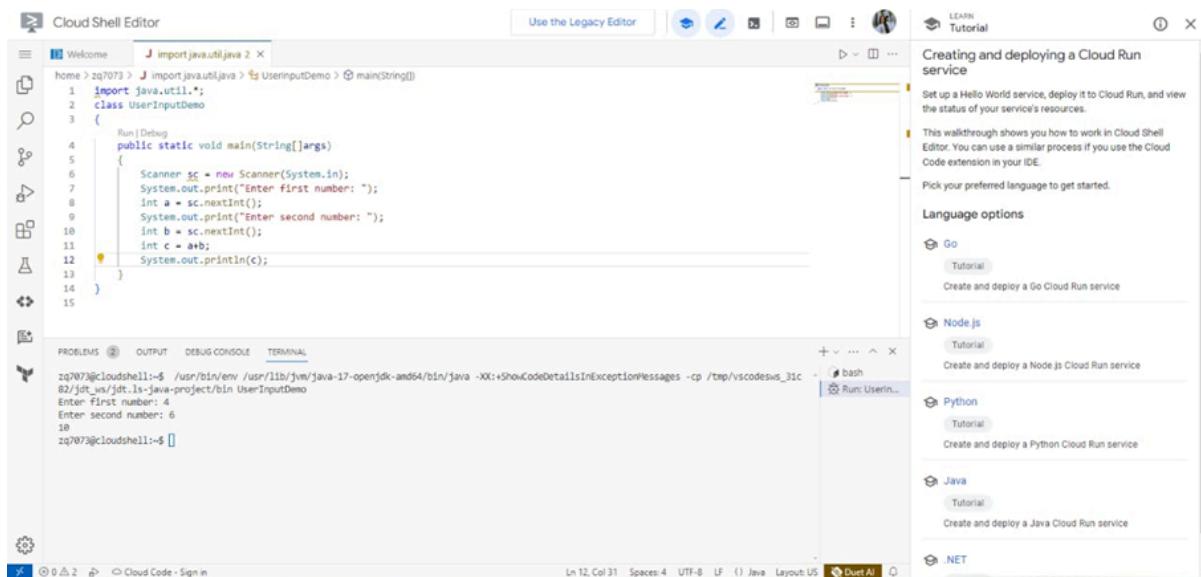
- Build and deploy a web service to Cloud Run.

To get started, follow the [walkthrough in Cloud Shell Editor](#).

Step 11: Create a new file of any of the choice either of java, python, c, etc



Step 12: After creating a new file >> Write a code in any particular language >> run it with the help of “RUN” button >> Output will be reflected in the TERMINAL



Signature: _____

PRACTICAL 8

Aim: Using the software like JDK 1.7/1.8. design and develop web application using MVC Framework.

→ Employee file (Model)

```
public class Employee {  
    // declaring the variables  
    private String EmployeeName;  
    private String EmployeeId;  
    private String EmployeeDepartment;  
  
    // defining getter and setter methods  
    public String getId() {  
        return EmployeeId;  
    }  
  
    public void setId(String id) {  
        this.EmployeeId = id;  
    }  
  
    public String getName() {  
        return EmployeeName;  
    }  
  
    public void setName(String name) {  
        this.EmployeeName = name;  
    }  
  
    public String getDepartment() {  
        return EmployeeDepartment;  
    }  
  
    public void setDepartment(String Department) {  
        this.EmployeeDepartment = Department;  
    } }
```

→ EmployeeView file (View)

```
public class EmployeeView {  
    // method to display the Employee details  
    public void printEmployeeDetails (String EmployeeName, String EmployeeId, String  
    EmployeeDepartment){  
        System.out.println("Employee Details: ");  
        System.out.println("Name: " + EmployeeName);  
        System.out.println("Employee ID: " + EmployeeId);  
        System.out.println("Employee Department: " + EmployeeDepartment);  
    }  
}
```

→ EmployeeController file (Controller)

```
public class EmployeeController {  
    // declaring the variables model and view  
    private Employee model;  
    private EmployeeView view;  
  
    // constructor to initialize  
    public EmployeeController(Employee model, EmployeeView view) {  
        this.model = model;  
        this.view = view;  
    }  
  
    // getter and setter methods  
    public void setEmployeeName(String name){  
        model.setName(name);  
    }  
  
    public String getEmployeeName(){  
        return model.getName();  
    }  
  
    public void setEmployeeId(String id){
```

```

        model.setId(id);
    }

    public String getEmployeeId(){
        return model.getId();
    }

    public void setEmployeeDepartment(String Department){
        model.setDepartment(Department);
    }

    public String getEmployeeDepartment(){
        return model.getDepartment();
    }

    // method to update view
    public void updateView() {
        view.printEmployeeDetails(model.getName(), model.getId(), model.getDepartment());
    }
}

```

→ **MVCMain file**

```

public class MVCMain {
    public static void main(String[] args) {

        // fetching the employee record based on the employee_id from the database
        Employee model = retriveEmployeeFromDatabase();

        // creating a view to write Employee details on console
        EmployeeView view = new EmployeeView();

        EmployeeController controller = new EmployeeController(model, view);
        controller.updateView();

        //updating the model data
    }
}

```

```

        controller.setEmployeeName("Nirnay");
        System.out.println("\n Employee Details after updating: ");

        controller.updateView();
    }

private static Employee retrieveEmployeeFromDatabase(){
    Employee Employee = new Employee();
    Employee.setName("Anu");
    Employee.setId("11");
    Employee.setDepartment("Salesforce");
    return Employee;
}
}

```

→ Now, we need to open command prompt in the folder where we've saved these file and compile our program.

- javac MVCMain.java

	Name	Date modified	Type	Size
✓ Today				
□	Employee.class	11-03-2024 11:37	CLASS File	1 KB
□	EmployeeController.class	11-03-2024 11:37	CLASS File	2 KB
□	EmployeeView.class	11-03-2024 11:37	CLASS File	1 KB
□	MVCMain.class	11-03-2024 11:37	CLASS File	1 KB
□	EmployeeView.java	11-03-2024 11:32	Java Source File	1 KB
□	MVCMain.java	11-03-2024 11:32	Java Source File	1 KB
□	Employee.java	11-03-2024 11:32	Java Source File	1 KB
□	EmployeeController.java	11-03-2024 11:32	Java Source File	2 KB

(After compilation, we can now see that 4 class files have been created inside the same folder).

→ Next step is to run our program, for getting the desired output.

- java MVCMain

```

Employee Details:
Name: Anu
Employee ID: 11
Employee Department: Salesforce

Employee Details after updating:
Employee Details:
Name: Nirnay
Employee ID: 11
Employee Department: Salesforce

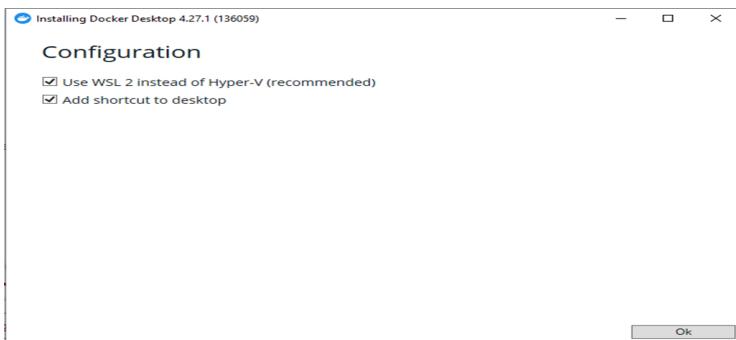
```

Signature: _____

PRACTICAL 9

Aim: Installing and configuring Dockers in local host and running multiple images on a Docker Platform.

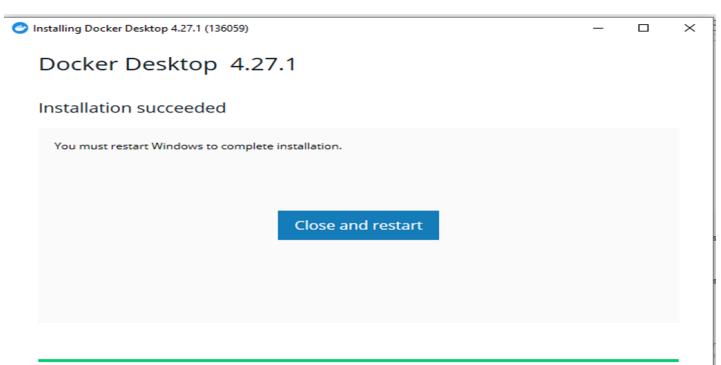
→ Installing Docker



Docker Desktop 4.27.1

Unpacking files...

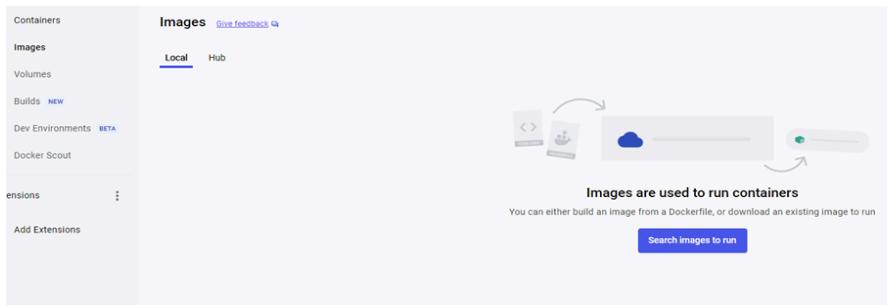
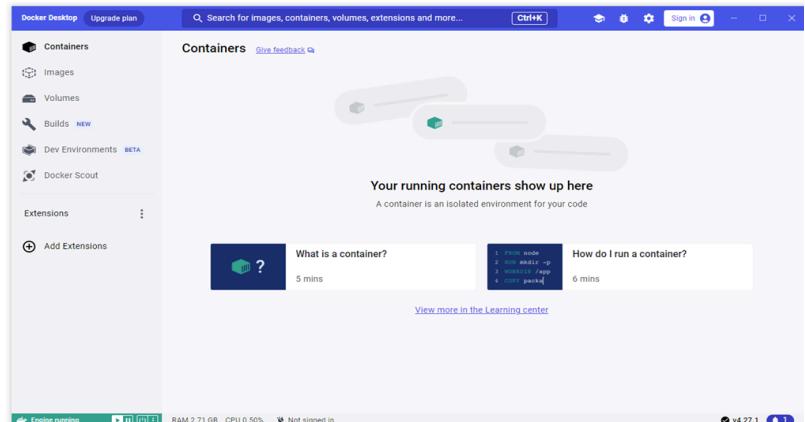
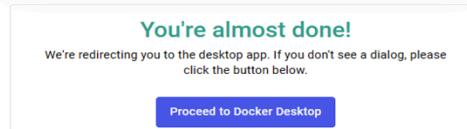
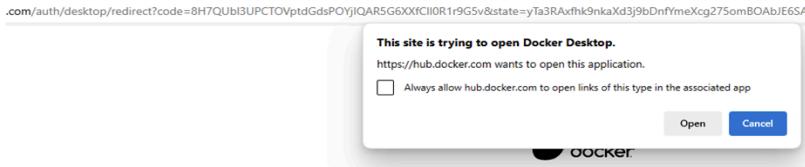
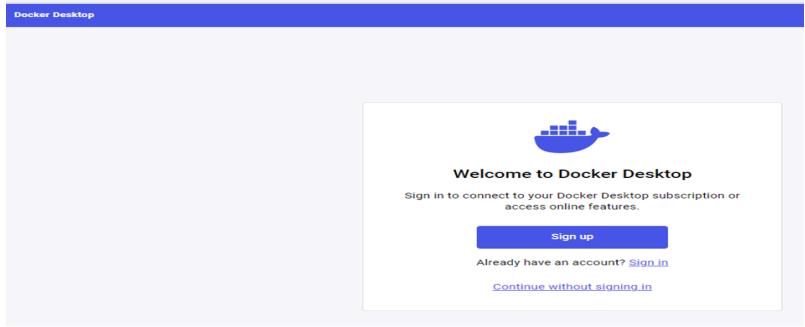
```
Unpacking file: resources/services.raw
Unpacking file: resources/linux-daemon-options.json
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/dvdpic.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/nanosnapshot_blob.bin
```



Complete the installation of Docker Desktop.

- Use recommended settings (requires administrator password)
Docker Desktop automatically sets the necessary configurations that work for most developers.
- Use advanced settings
You manually set your preferred configurations.

Finish



→ After installation, open command prompt and type the following commands:

- docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

- docker info

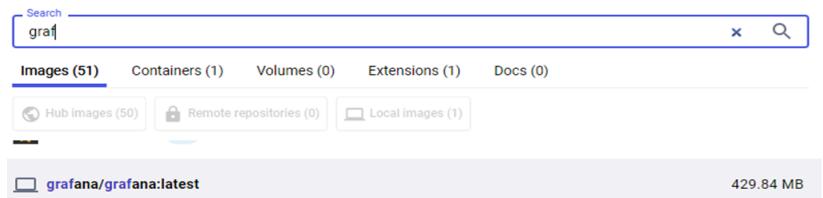
```
Client:
  Version:    25.0.2
  Context:    default
  Debug Mode: false
  Plugins:
    buildx: Docker Buildx (Docker Inc.)
      Version: v0.12.1-desktop.4
      Path:   C:\Program Files\Docker\cli-plugins\docker-buildx
    compose: Docker Compose (Docker Inc.)
      Version: v2.24.3-desktop.1
      Path:   C:\Program Files\Docker\cli-plugins\docker-compose
    debug: Get a shell into any image or container. (Docker Inc.)
      Version: 0.0.22
      Path:   C:\Program Files\Docker\cli-plugins\docker-debug
    dev: Docker Dev Environments (Docker Inc.)
      Version: v0.1.0
      Path:   C:\Program Files\Docker\cli-plugins\docker-dev-environment
```

- docker images

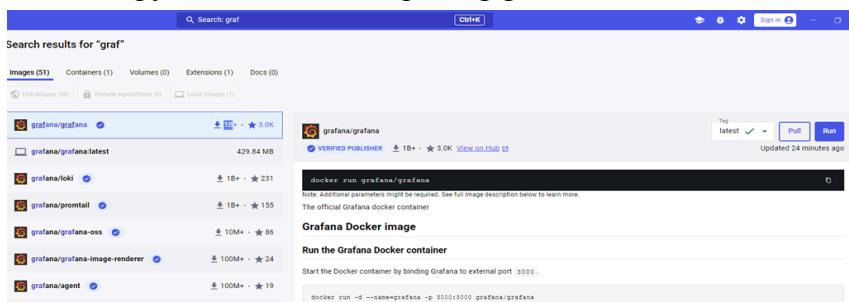
```
C:\Users\admin>docker images
REPOSITORY      TAG          IMAGE ID      CREATED     SIZE
```

- Adding images in the docker container
(Let's consider grafana)

1. Search for grafana in the docker desktop



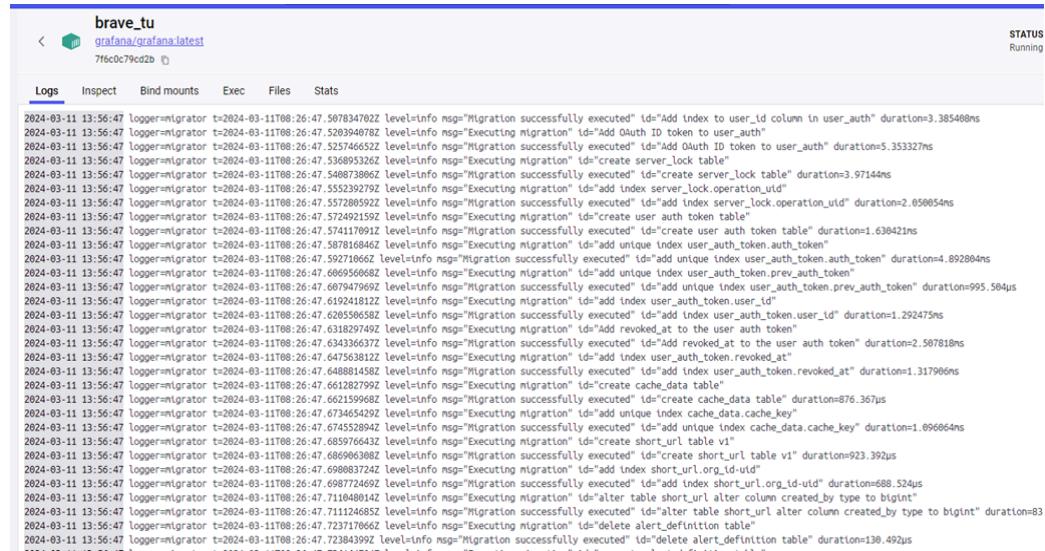
2. Copy the command for pulling grafana into docker



3. Paste it into the command prompt and hit enter

```
C:\Users\admin>docker pull grafana/grafana
Using default tag: latest
latest: Pulling from grafana/grafana
4abcf2066143: Pull complete
29fb09e11aa: Pull complete
81bf57c2e5a3: Pull complete
fb9f28e6477f: Pull complete
ae5a5eafe4: Pull complete
44cdf122dcfb: Pull complete
68de72da744b: Pull complete
ca8da122723b: Pull complete
```

4. After pulling, run it from Docker's GUI



The screenshot shows the Grafana logs in a Docker container. The logs output by the migration script show the creation of various database indexes and tables. Key log entries include:

- Migration successfully executed: "Migration successfully executed" "id=add index to user_id column in user_auth" duration=3.385408ms
- Add OAuth ID token to user_auth: "Add OAuth ID token to user_auth" duration=5.353327ms
- Create server_lock table: "Create server_lock table" duration=3.97144ms
- Create server_lock table: "Create server_lock table" duration=2.050054ms
- Create user_auth_token table: "Create user_auth_token table" duration=1.630421ms
- Add unique index user_auth_token.auth_token: "Add unique index user_auth_token.auth_token" duration=4.892804ns
- Add unique index user_auth_token.prev_auth_token: "Add unique index user_auth_token.prev_auth_token" duration=995.508μs
- Add unique index user_auth_token.user_id: "Add unique index user_auth_token.user_id" duration=1.292475ms
- Revoke user auth token: "Revoke user auth token" duration=2.507810ms
- Add unique index user_auth_token.revoked_at: "Add unique index user_auth_token.revoked_at" duration=1.317906ms
- Create cache_data_table: "Create cache_data_table" duration=876.367μs
- Add unique index cache_data.cache_key: "Add unique index cache_data.cache_key" duration=1.096064ms
- Create short_url_table_v1: "Create short_url_table_v1" duration=923.392μs
- Add index short_url.org_id-uid: "Add index short_url.org_id-uid" duration=688.524μs
- Alter table short_url alter column created_by type to bigint: "Alter table short_url alter column created_by type to bigint" duration=83
- Delete alert_definition table: "Delete alert_definition table" duration=130.492μs

→ And now, we can see that an image is added to the docker's image section

```
C:\Users\admin>docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
grafana/grafana     latest        c8b91775d855   5 days ago        430MB
```

Images <small>Give feedback</small>						
Local		Hub				
		Last refresh: 26 minutes ago				
		Search				
Name	Tag		Status	Created	Size	Actions
grafana/grafana	latest	In use	5 days ago	429.84 MB	▶ ⚙	⋮
c8b91775d855						

Signature: _____

PRACTICAL 10

Aim: Install and Configure Puppet.

→ Install puppet and open “Start command prompt with puppet”.

```
sc config puppet start=demand
```

```
[SC] ChangeServiceConfig SUCCESS
```

```
start puppet -- debug --logdest eventlog
```

```
puppet config print rundir
```

```
C:/ProgramData/PuppetLabs/puppet/var/run
```

```
puppet config set rundir /var/run/puppetlabs
```

```
puppet describe --list
```

```
These are the types known to puppet:
augeas      - @summary Apply a change or an array of change ...
cron        - @summary Installs and manages cron jobs
exec        - Executes external commands
file         - Manages files, including their content, owner ...
filebucket   - A repository for storing and retrieving file ...
group       - Manage groups
host         - @summary Installs and manages host entries
mount       - @summary Manages mounted filesystems, includi ...
notify      - Sends an arbitrary message, specified as a st ...
package     - Manage packages
resources   - This is a metatype that can manage other reso ...
schedule    - Define schedules for Puppet
scheduled_task - @summary Installs and manages Windows Schedul ...
selboolean  - @summary Manages SELinux booleans on systems ...
semodule    - @summary Manages loading and unloading of SEL ...
service     - Manage running services
ssh_authorized_key - @summary Manages SSH authorized keys
sshkey      - @summary Installs and manages ssh host keys
stage       - A resource type for creating new run stages
tidy        - Remove unwanted files based on specific crite ...
user        - Manage users
whit        - Whits are internal artifacts of Puppet's curr ...
yumrepo    - @summary The client-side description of a yum ...
zfs         - @summary Manage zfs
zone        - @summary Manages Solaris zones
zpool       - @summary Manage zpools
```

```
puppet describe file --providers
```

```
file
=====
Manages files, including their content, ownership, and permissions.
The `file` type can manage normal files, directories, and symlinks; the
type should be specified in the `ensure` attribute.
File contents can be managed directly with the `content` attribute, or
downloaded from a remote source using the `source` attribute; the latter
can also be used to recursively serve directories (when the `recurse`
attribute is set to `true` or `local`). On Windows, note that file
contents are managed in binary mode; Puppet never automatically translates
line endings.
**Autorequires:** If Puppet is managing the user or group that owns a
file, the file resource will autorequire them. If Puppet is managing any
parent directories of a file, the file resource autorequires them.
Warning: Enabling `recurse` on directories containing large numbers of
files slows agent runs. To manage file attributes for many files,
consider using alternative methods such as the `chmod_r`, `chown_r`,
or `recursive_file_permissions` modules from the Forge.
```

```
puppet describe user -s -m
```

```
user
=====
Manage users. This type is mostly built to manage system
users, so it is lacking some features useful for managing normal
users.
This resource type uses the prescribed native tools for creating
groups and generally uses POSIX APIs for retrieving information
about them. It does not directly modify `/etc/passwd` or anything.
**Autorequires:** If Puppet is managing the user's primary group (as
provided in the `gid` attribute) or any group listed in the `groups`
attribute then the user resource will autorequire that group. If Puppet
is managing any role accounts corresponding to the user's roles, the
```

```
puppet help
```

```
Usage: puppet <subcommand> [options] <action> [options]

Available subcommands:

Common:
  agent           The puppet agent daemon
  apply          Apply Puppet manifests locally
  config         Interact with Puppet's settings.
  help           Display Puppet help.
  lookup         Interactive Hiera lookup
  module         Creates, installs and searches for mod
  resource       The resource abstraction layer shell
```

```
puppet node find somenode.puppetlabs.lan --terminus plain --render-as yaml
```

```
--- !ruby/object:Puppet::Node
name: somenode.puppetlabs.lan
environment: production
parameters:
  aio_agent_version: 8.4.0
  dmi:
    manufacturer: ASUSTeK COMPUTER INC.
    product:
      name: ASUS EXPERTCENTER D500TC_D500TC
      serial_number: N4PFCG00T087160
      uuid: 7A70C1AD-EFB3-6642-9984-3C8554F854A9
  env_windows_installdir: C:\Program Files\Puppet\agent\windows
  facterversion: 4.5.2
  fips_enabled: false
  identity:
```

```
puppet facts find
```

```
{
  "name": "desktop-ffffbig0",
  "values": {
    "aio_agent_version": "8.4.0",
    "dmi": {
      "manufacturer": "ASUSTeK COMPUTER INC.",
      "product": {
        "name": "ASUS EXPERTCENTER D500TC_D500TC",
        "serial_number": "N4PFCG00T087160",
        "uuid": "7A70C1AD-EFB3-6642-9984-3C8554F854A9"
      }
    },
    "env_windows_installdir": "C:\\Program Files\\Puppet\\agent\\windows"
  }
}
```

Signature: _____

PRACTICAL 11

Aim: Install and Configure IBM Bluemix.

- Install IBM bluemix and open the command prompt.

ibmcloud

```
NAME:
  ibmcloud - A command line tool to interact with
  Find more information at: https://ibm.biz/cli-do

USAGE:
  [environment variables] ibmcloud [global options]
  [options]

VERSION:
  2.23.0+eca5d10-2024-01-31T20:36:03+00:00
```

ibmcloud help

```
NAME:
  ibmcloud - A command line tool to interact
  Find more information at: https://ibm.biz/c

USAGE:
  [environment variables] ibmcloud [global options]

VERSION:
  2.23.0+eca5d10-2024-01-31T20:36:03+00:00
```

ibmcloud plugin repo-plugins

Status	Name	Description
Not Installed	container-registry[cr]	. Manage IBM Cloud Container Registry
Not Installed	container-service[kubernetes]	.595... Manage IBM Cloud Kubernetes Service
Not Installed	cloud-functions[wsk/function]	6... Manage Cloud Functions
Not Installed	cloud-internet-services[cis]	5.8.... Manage Cloud Internet Service
Not Installed	dbaas-cli[dbaas]	. Manage Hyper Protect DBaaS cluster
Not Installed	cloud-databases[cdb]	9... Manage Cloud databases
Not Installed	key-protect[kp]	. Manage encryption keys on IBM Cloud
Not Installed	doi[doi]	

ibmcloud plugin install code-engine

```
Looking up 'code-engine' from repository
Plug-in 'code-engine[ce] 1.49.5' found
Attempting to download the binary file
  58.66 MiB / 58.66 MiB [=====]
=====] 100.00% 37s
61513728 bytes downloaded
Installing binary...
OK
Plug-in 'code-engine 1.49.5' was successfully installed
```

```
ibmcloud plugin install -a
```

```
The following plug-ins will be installed:  
* 'container-registry[cr] 1.3.5'  
* 'container-service[kubernetes-service/b]  
* 'cloud-functions[wsk/functions/fn] 1.0.  
* 'cloud-internet-services[cis] 1.15.11'  
* 'dbaas-cli[dbaas] 2.2.5'  
* 'cloud-databases[cdb] 0.17.1'  
* 'key-protect[kp] 0.9.3'  
* 'doi[doi] 0.4.5'  
* 'tke 1.4.1'  
* 'cloud-object-storage[cos] 1.8.0'  
* 'event-streams[es] 2.5.1'  
* 'power-iaas[pi] 1.0.0'  
* 'vpc-infrastructure[infrastructure-serv
```

```
ibmcloud plugin install container-service@1.0.506 secrets-manager@0.1.25
```

```
Looking up 'container-service' from repository 'IBM Cloud'...  
Looking up 'secrets-manager' from repository 'IBM Cloud'...
```

```
ibmcloud plugin list
```

Plugin Name supported	Version	Status	Private endpoints
analytics-engine-v3[ae-v3]	2.5.0	false	
app-configuration	1.0.14	true	
atracker	0.4.9	true	
catalogs-management	2.3.29	true	
cbr	1.6.0	true	
cloud-databases[cdb]	0.17.1	false	
cloud-dns-services[dns/dns-svcs]	0.7.3	true	
cloud-functions[wsk/functions/fn]	1.0.79	false	
cloud-internet-services[cis]	1.15.11	true	
cloud-object-storage	1.8.0	false	
cloudant[cll]	0.2.0	true	

```
ibmcloud plugin update
```

```
Checking upgrades for all installed plug-ins from repository 'IBM Cloud'...  
No updates are available.
```

```
ibmcloud plugin install container-registry
```

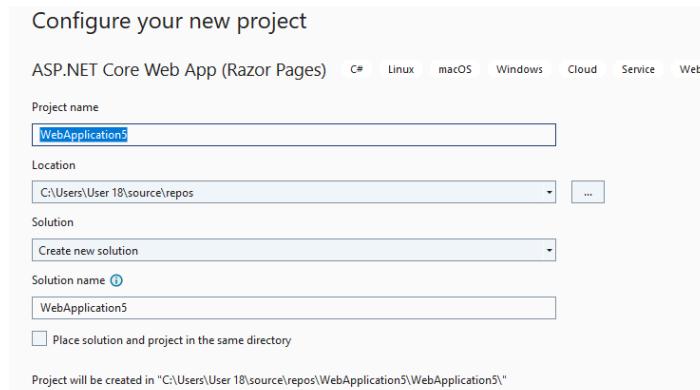
```
Looking up 'container-registry' from repository 'IBM Cloud'...  
Plug-in 'container-registry[cr] 1.3.5' found in repository 'IBM Cloud'  
Plug-in 'container-registry[cr] 1.3.5' was already installed. Do you want to re-install it or not? [y/N] > y  
Attempting to download the binary file...  
12.41 MiB / 12.41 MiB [======] 100.00% 0s  
13015040 bytes downloaded  
Installing binary...
```

Signature: _____

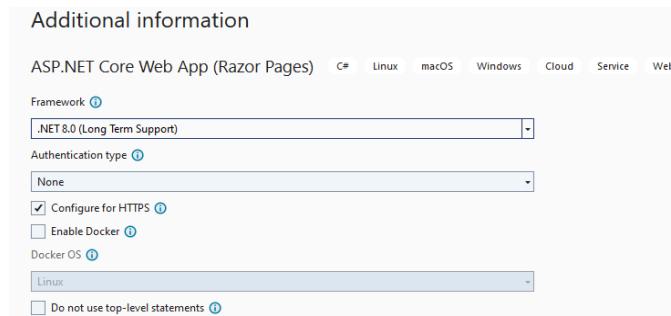
PRACTICAL 12

Aim: Developing an ASP.NET web based application on the Azure platform.

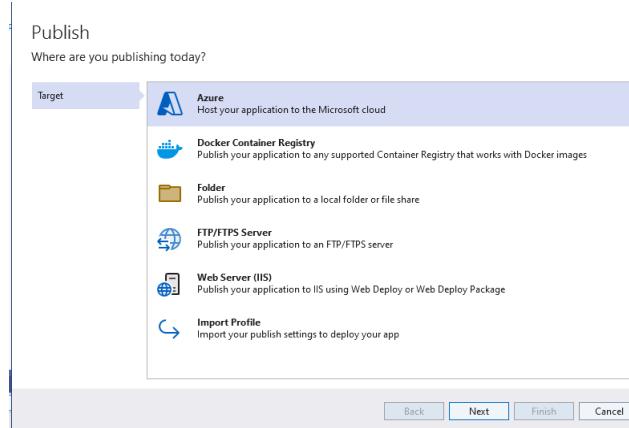
Step 1: Open Visual Studio 2022 -> ASP.NET Core Web App -> Give a suitable project name and location -> Next



Step 2: Next -> Finish



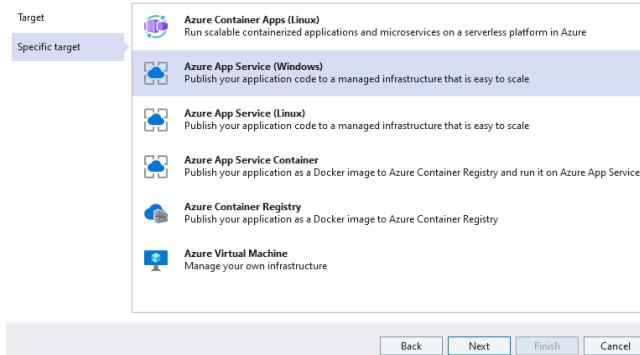
Step 3: Click on Publish -> Azure -> Next



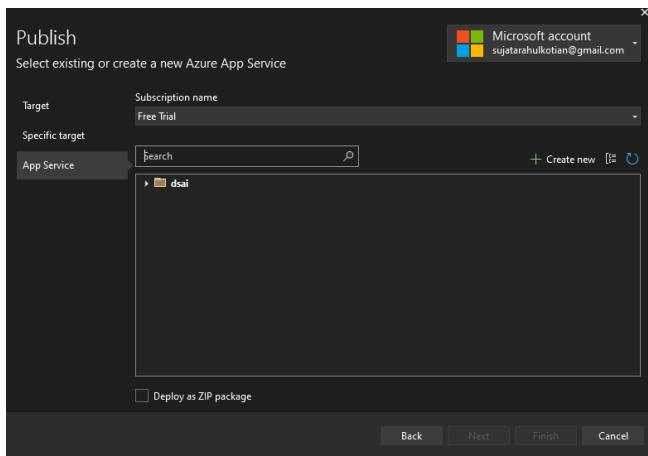
Step 4: Azure App Service (Windows) -> Finish

Publish

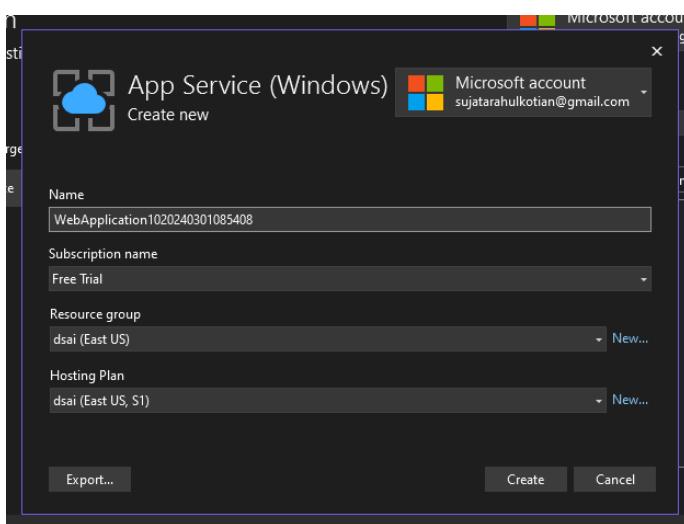
Which Azure service would you like to use to host your application?



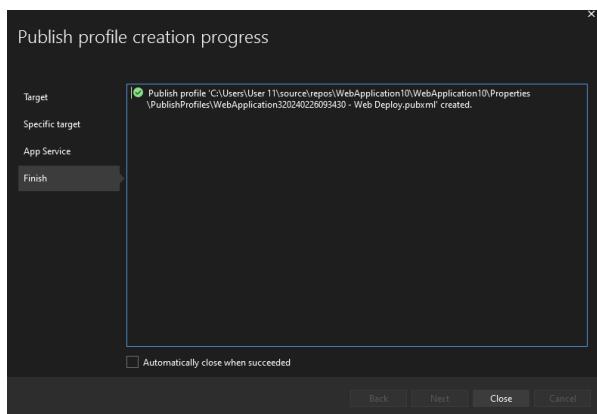
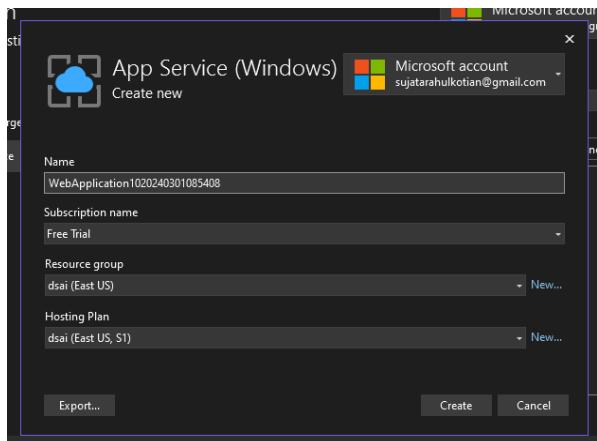
Step 5: Sign in to your Azure or Microsoft Account



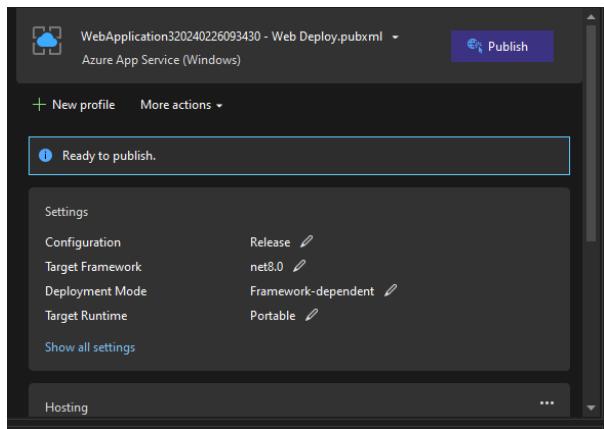
Step 6: Click on “+Create New” Option



Step 7: Click on “Create”



Step 8: Click on Publish



Signature: _____