# EXPERIMENT:1

**Objective**:- Implement the Remote Method Invocation(RMI).

**Program**:-

• Create the Remote Interface:-

## ➢ ReceiveMessageInterface.java

|  |
| --- |
| **import java.rmi.\*;**    **public interface ReceiveMessageInterface extends Remote**    **{**    **void receiveMessage(String x) throws RemoteException;**  **}** |

• Create the Client Application:-

## ➢ RmiClient.java

|  |
| --- |
| **import java.rmi.\*; import java.rmi.registry.\*; import java.net.\*;**    **public class RmiClient**  **{**  **static public void main(String args[])**  **{**  **ReceiveMessageInterface rmiServer;**  **Registry registry;**  **String serverAddress=args[0];**  **String serverPort=args[1];**  **String text=args[2];**  **System.out.println("sending "+text+" to "+serverAddress+":"+serverPort); try{**  **// get the gregistryh** |

**//registry=LocateRegistry.getRegistry(serverAddress,(new**

**Integer(serverPort)).intValue());**

**registry = LocateRegistry.getRegistry(serverAddress,**

**Integer.parseInt(serverPort));**

**// look up the remote object**

**rmiServer = (ReceiveMessageInterface)(registry.lookup("rmiServer"));**

**// call the remote method rmiServer.receiveMessage(text);**

**}**

**catch(RemoteException e)**

**{**

**e.printStackTrace();**

**}**

**catch(NotBoundException e)**

**{**

**e.printStackTrace();**

**}**

**}**

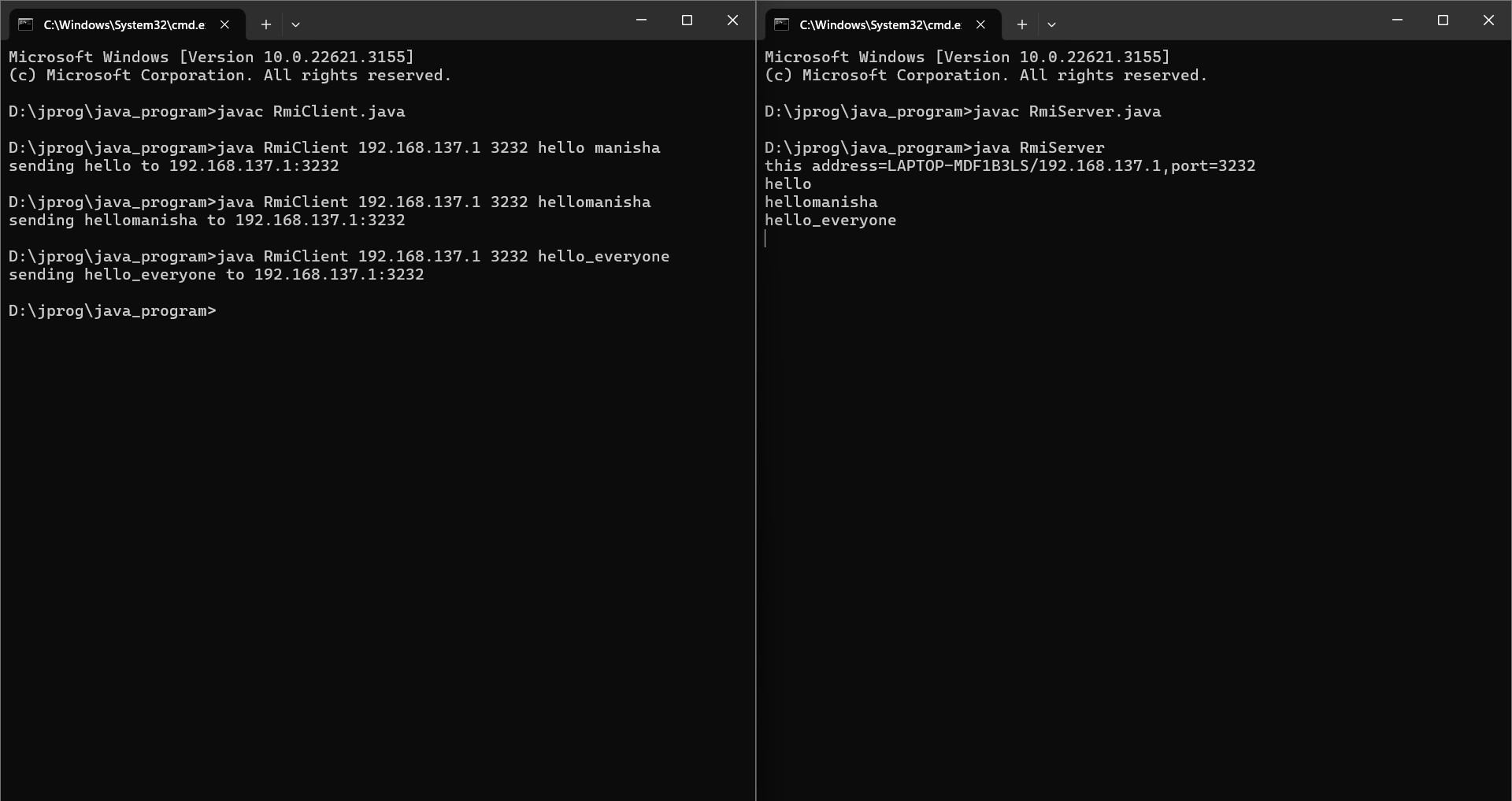
**}**

• Create the Server Application:-

## ➢ RmiServer.java

|  |
| --- |
| **import java.rmi.\*; import java.rmi.registry.\*; import java.rmi.server.\*; import java.net.\*;**    **public class RmiServer extends java.rmi.server.UnicastRemoteObject implements ReceiveMessageInterface**  **{**  **int thisPort;**  **String thisAddress;**  **Registry registry; // rmi registry for lookup the remote objects.**    **// This method is called from the remote client by the RMI. // This is the implementation of the gReceiveMessageInterfaceh. public void receiveMessage(String x) throws RemoteException** |
| **{**  **System.out.println(x);**  **}**    **public RmiServer() throws RemoteException**  **{ try{**  **// get the address of this host.**  **thisAddress= (InetAddress.getLocalHost()).toString();**  **}**  **catch(Exception e){**  **throw new RemoteException("can't get inet address.");**  **}**  **thisPort=3232; // this port(registryfs port)**  **System.out.println("this address="+thisAddress+",port="+thisPort); try{**  **// create the registry and bind the name and object. registry = LocateRegistry.createRegistry( thisPort ); registry.rebind("rmiServer", this);**  **}**  **catch(RemoteException e){**  **throw e;**  **}**  **}**    **static public void main(String args[])**  **{ try{**  **RmiServer s=new RmiServer();**  **}**  **catch (Exception e) {**  **e.printStackTrace();**  **System.exit(1);**  **}**  **}**  **}** |

## Output:-



# EXPERIMENT:2

**Objective**:- Installation of any Hypervisor(VmwarePlayer/ Vmware workstation/ Oracle Virtual Box).

**Introduction**:- VMware develops virtualization Software. Virtualization software creates an abstraction layer over computer hardware that allows the hardware elements of a single computer — processors, memory, storage, and more — to be divided into multiple virtual computers, commonly called virtual machines (VMs).

Each virtual machine runs its own operating system (OS) and behaves like an independent computer, even though it is running on a portion of the actual underlying computer hardware. A VM is a software-based representation of a physical computer. An operating system (OS) running in a VM is called a guest OS.

## Installation:-

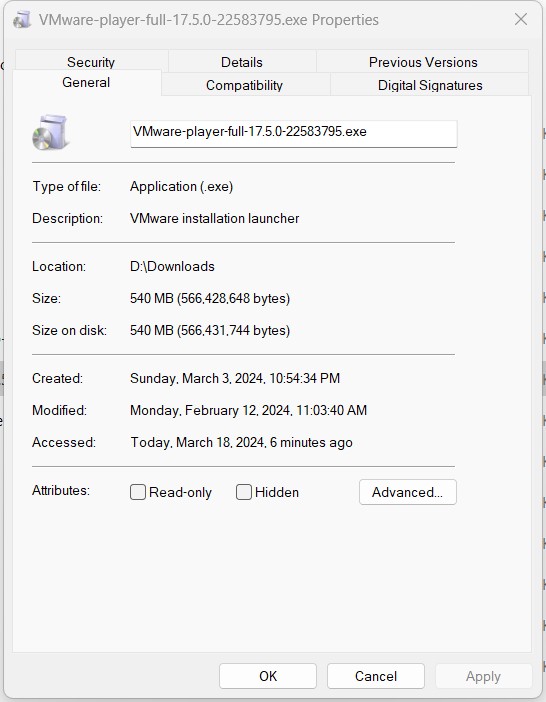
Steps to install Vmware workstation:

**Step 1:-** Installing VMware Workstation from given below link. There are two options for downloading one is Windows and other for Linux. My Base Operating System is Windows11, So I choose for VMware for Windows. If Your Base OS is Linux go and choose VMware for Linux Link.

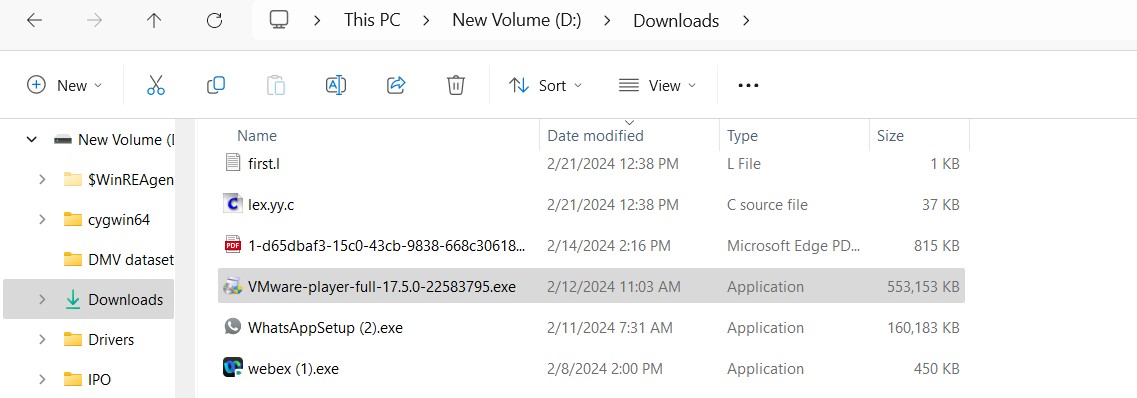
[https://www.vmware.com/content/vmware/vmware-publishedsites/us/products/workstation-player/workstation-player-evaluation.html.html](https://www.vmware.com/content/vmware/vmware-published-sites/us/products/workstation-player/workstation-player-evaluation.html.html)



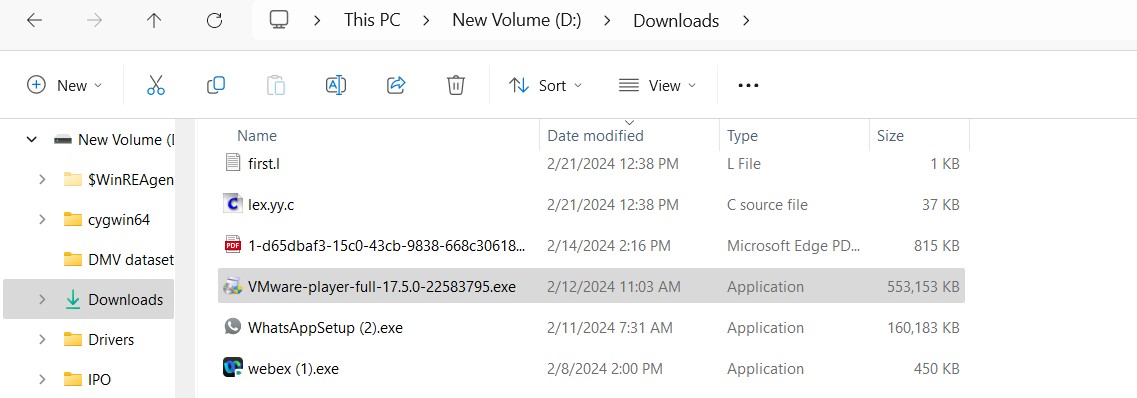
**Step 2:-** Check your VMware Properties.



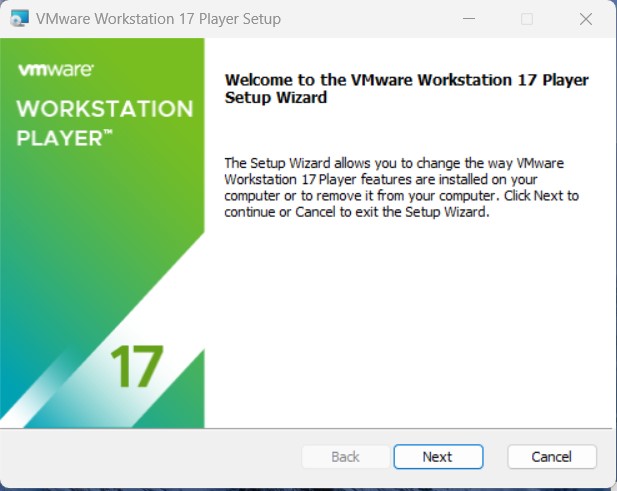
**Step 3:-** Go to Download Folder.



**Step 4:-** Click the VMware downloaded File and Install it.

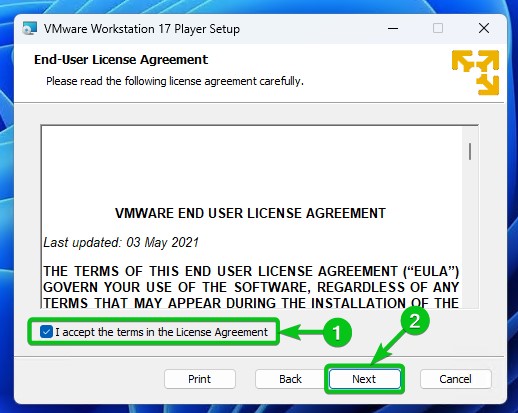


Step 5:- Click on VMware Software and Click on Next to the Installation wizard.



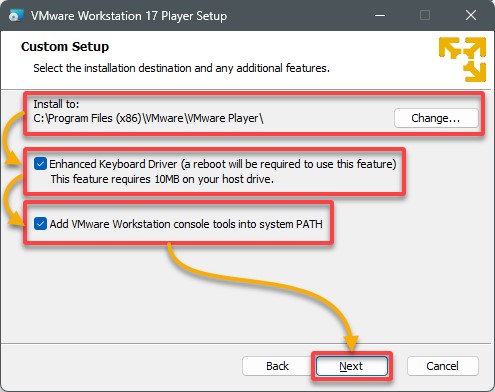
**Step 6:-** Read and Accept the VMware End User license agreement.

Click Next to Continue.



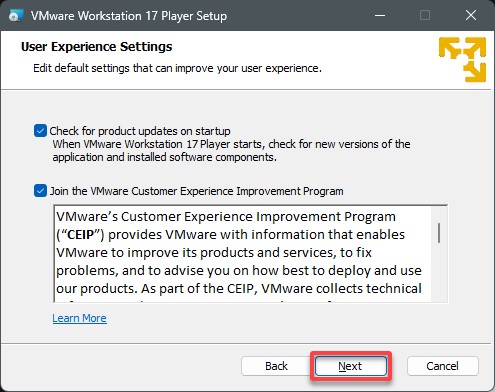
**Step 7:-** Specify the Installation directory. You can also enable Enhance keyboard driver here.

Click Next to continue.



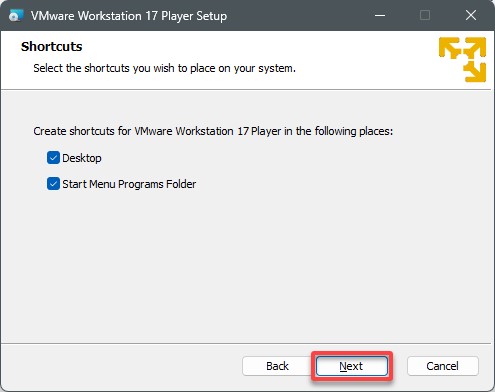
**Step 8:-** You can enable product startup and join the VMware Customer experience Improvement program here.

Click Next to Continue.

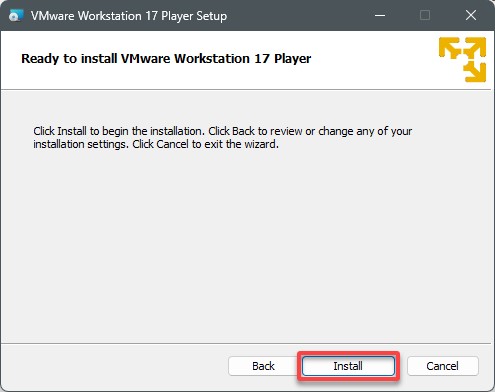


**Step 9:-** Select the shortcuts you want to create for easy access to VMware Workstation.

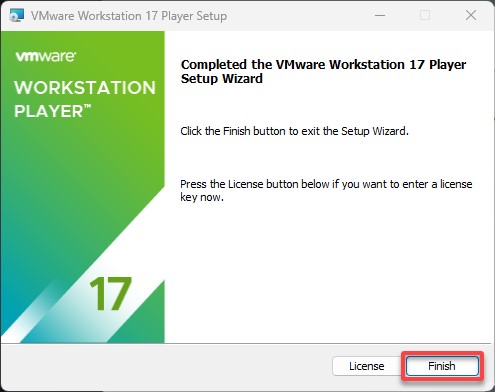
Click Next to Continue.



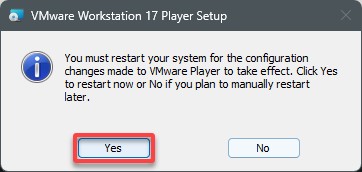
**Step 10:-** Click Install button to start the installation.



**Step 11:-** Click Finish to exit the wizard.



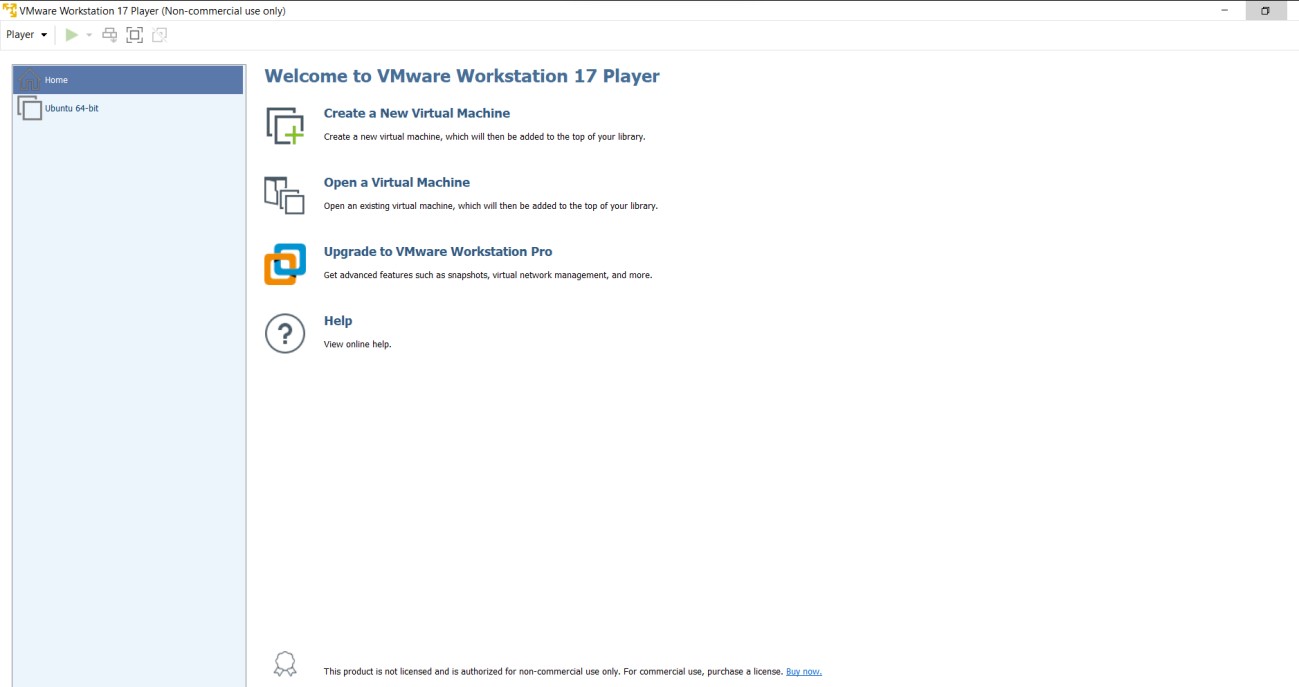
**Step 12:-** Lastly, click Yes to restart your computer for the changes.



**Step 13:-** That’s it we have successfully installed VMware Workstation 17 Player.

Now you can start the VMware Workstation 17 Player by clicking on the shortcut on Desktop.

Below is the Home screen of the VMware Workstation player which you will see every time when you start Workstation.



**Result:-**  VMware Workstation successfully setup and installed.

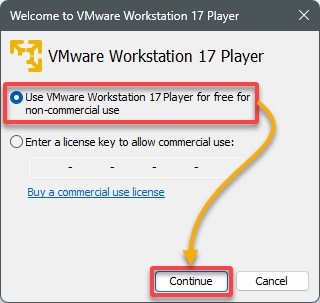
# EXPERIMENT:3

**Objective**:- Create a Virtual Machine of LinuxOS (Ubuntu) on Windows OS (ex. Windows11).

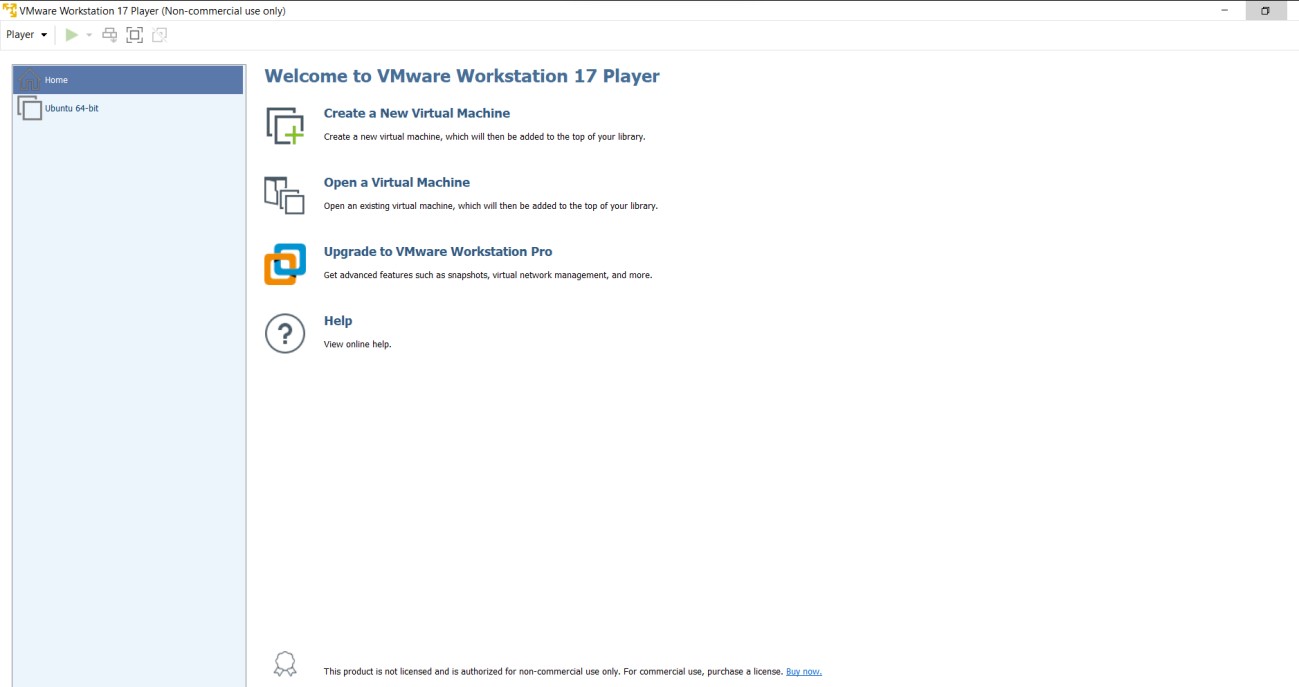
## Creation Steps:-

To create a VM in the VMware Workstation Player, follow these steps:

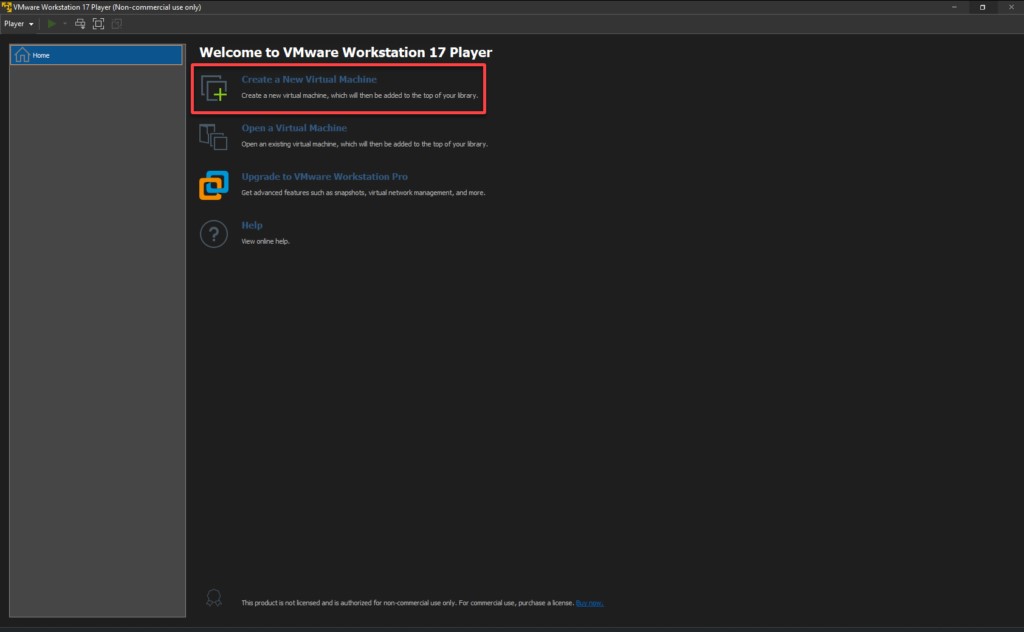
1. Launch VMware Workstation Player from your desktop shortcut or through the Start menu.
2. When prompted, select the first option to use the free version of VMware Workstation Player (or choose the second option if you have a license key) and click Continue.



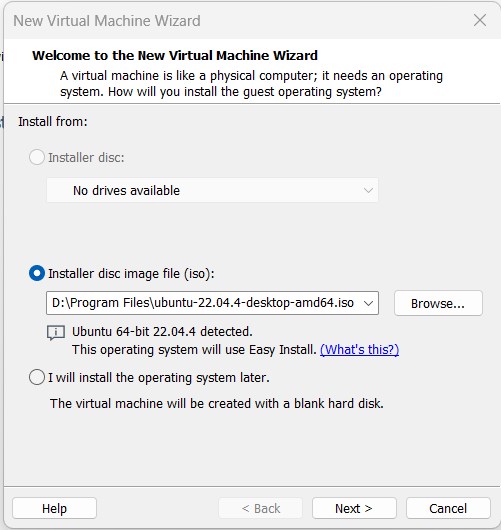
The VMware Workstation Player’s main window appears, as shown below.



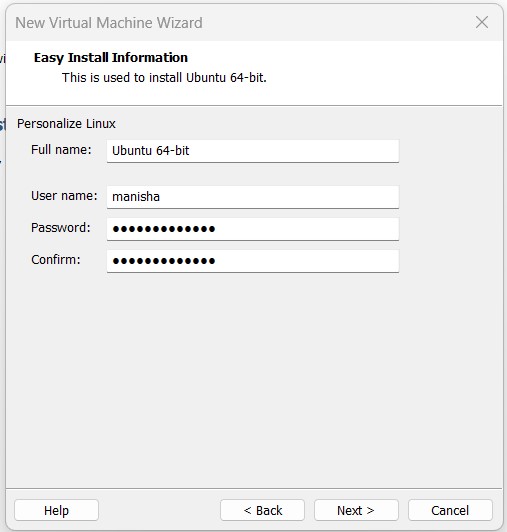
1. Next, click Create a New Virtual Machine, which opens a wizard to configure a new VM .



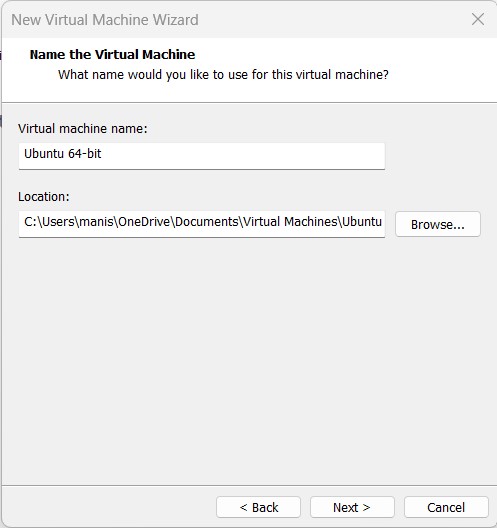
1. Now, select the second option, browse your ISO image file for the OS you wish to install on your VM, and click Next.



1. In the next screen, set the new OS administrator account credentials and click Next to confirm.

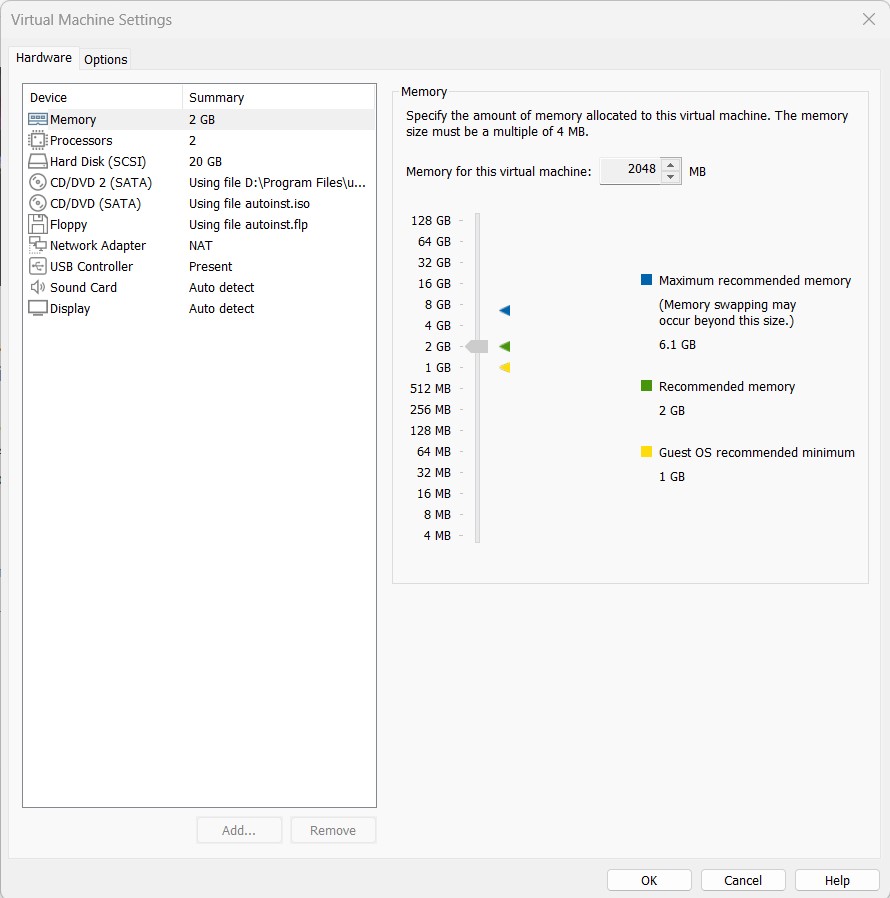
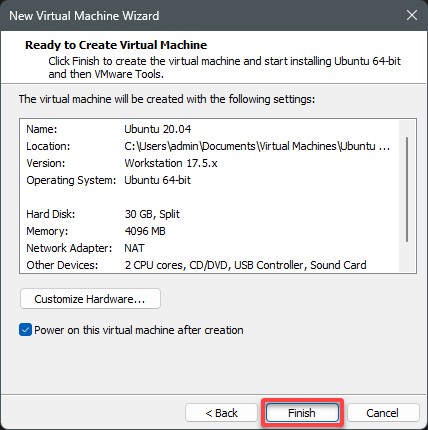


1. Provide a meaningful name for your VM (i.e., Ubuntu 22.04), choose a storage path for the VM files, and click Next.

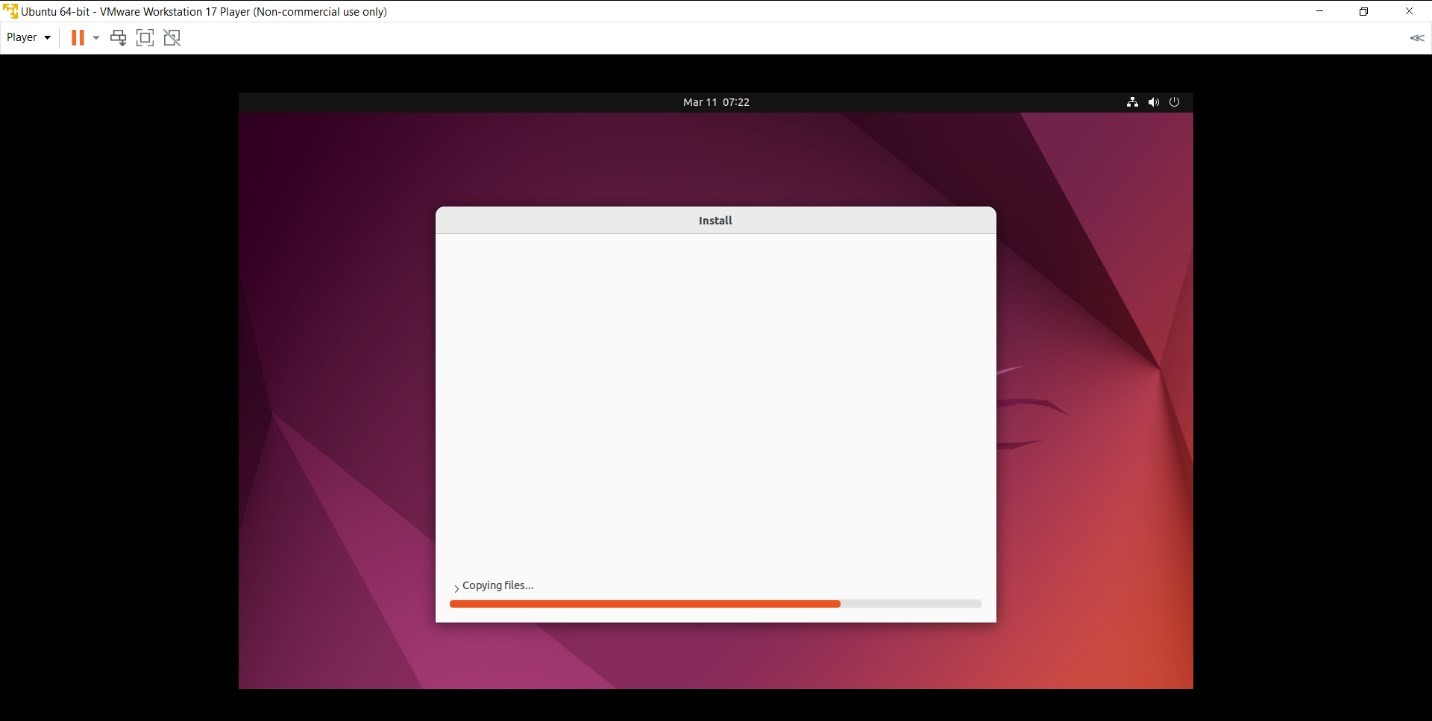


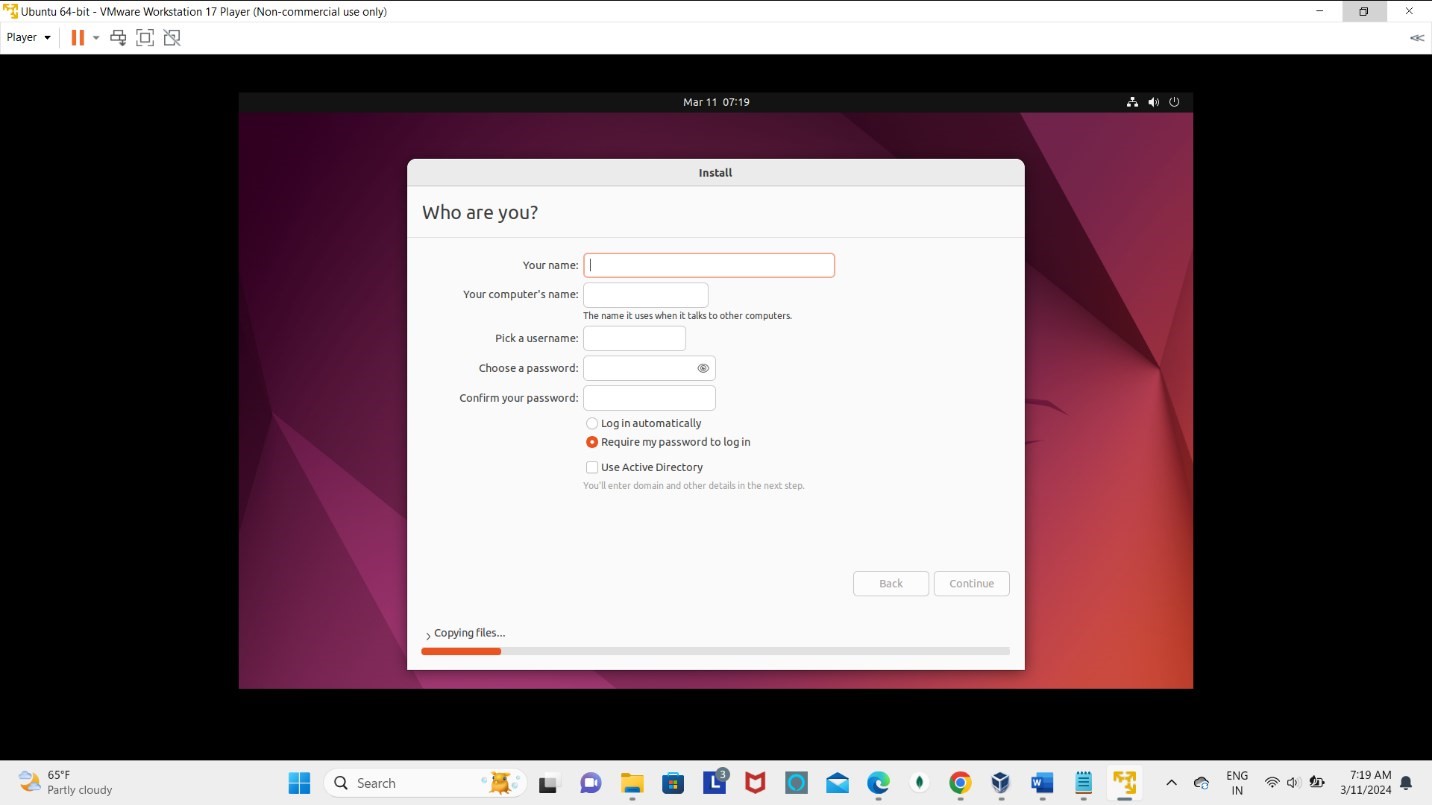
1. Decide on the amount of disk space to allocate to the VM and the maximum size the virtual hard disk can grow to, and click Next.
2. Now, click Finish to create the VM with your chosen settings, or click

Customize Hardware to adjust the hardware specifications before finalizing.



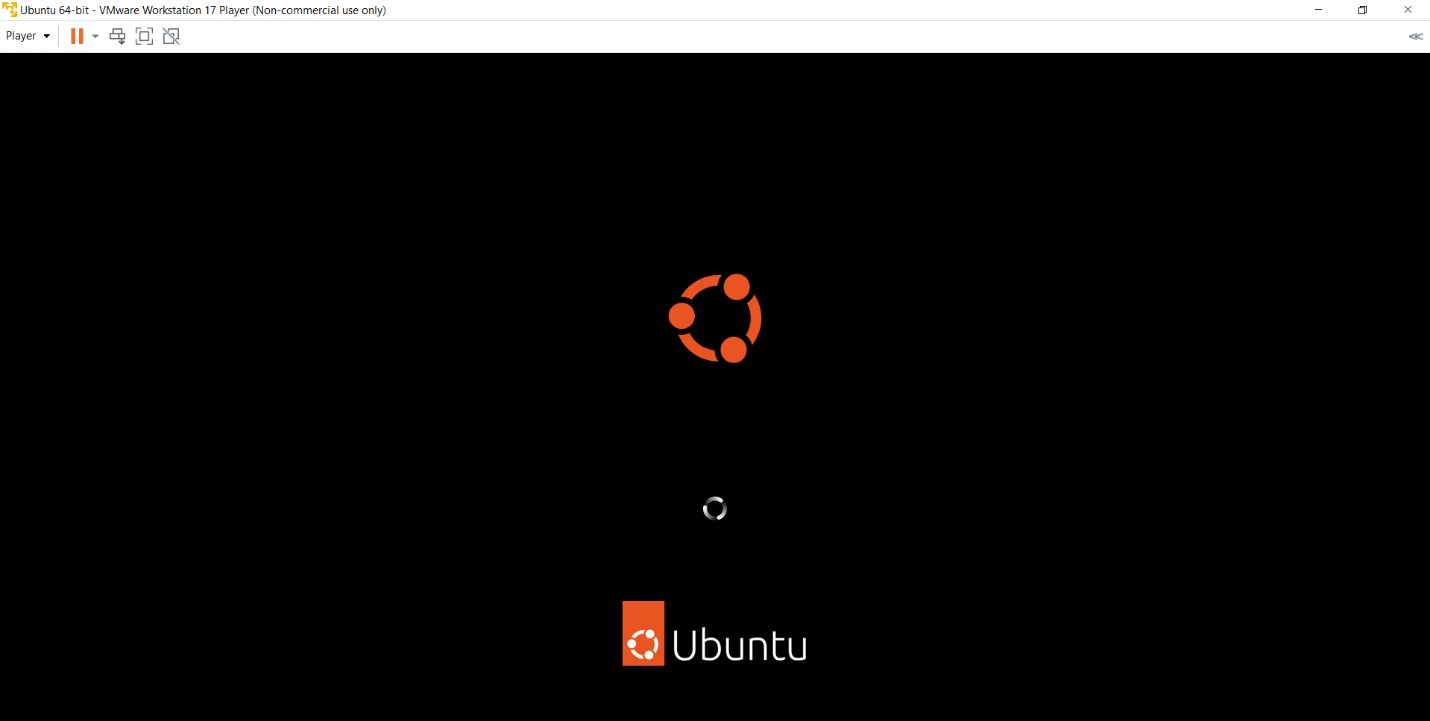
1. Ultimately, proceed by following the on-screen prompts to install the OS, replicating the process as if setting up a new computer.

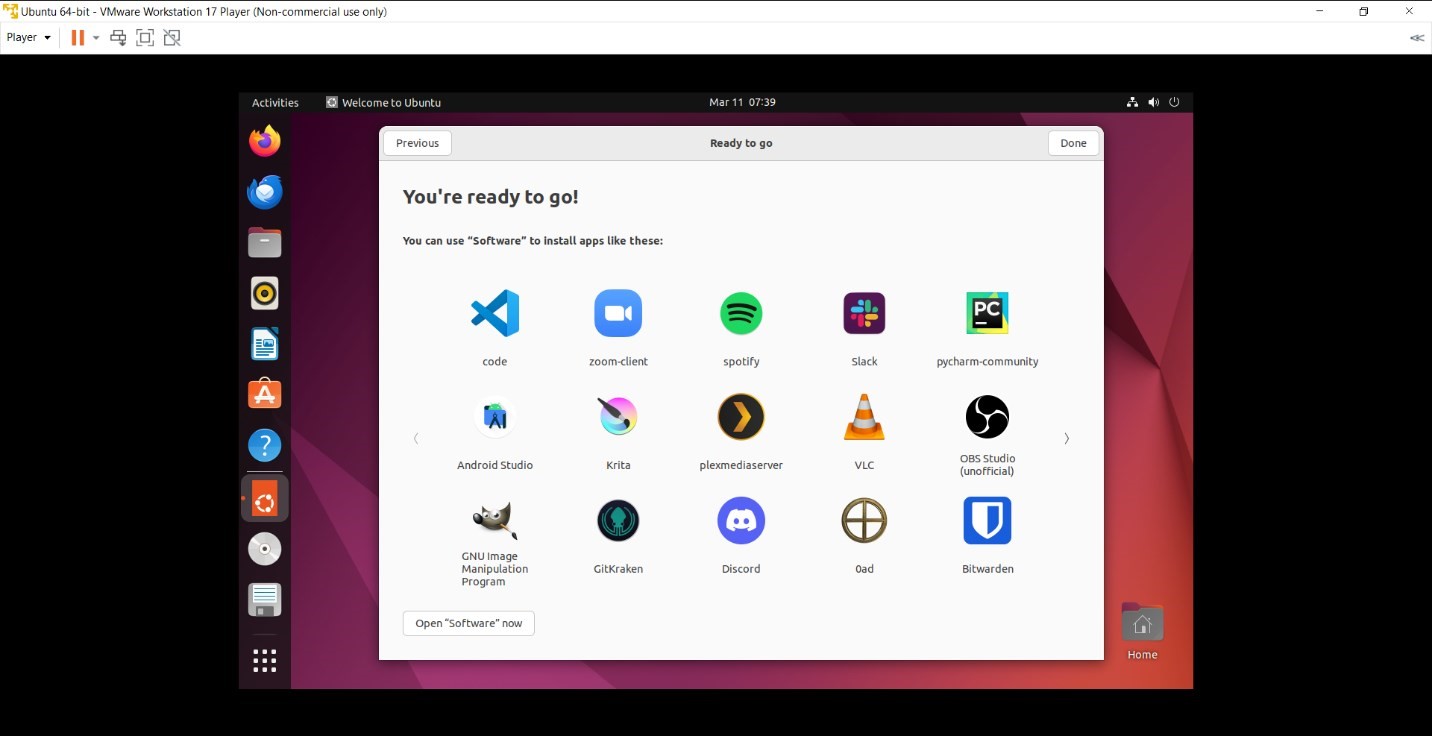


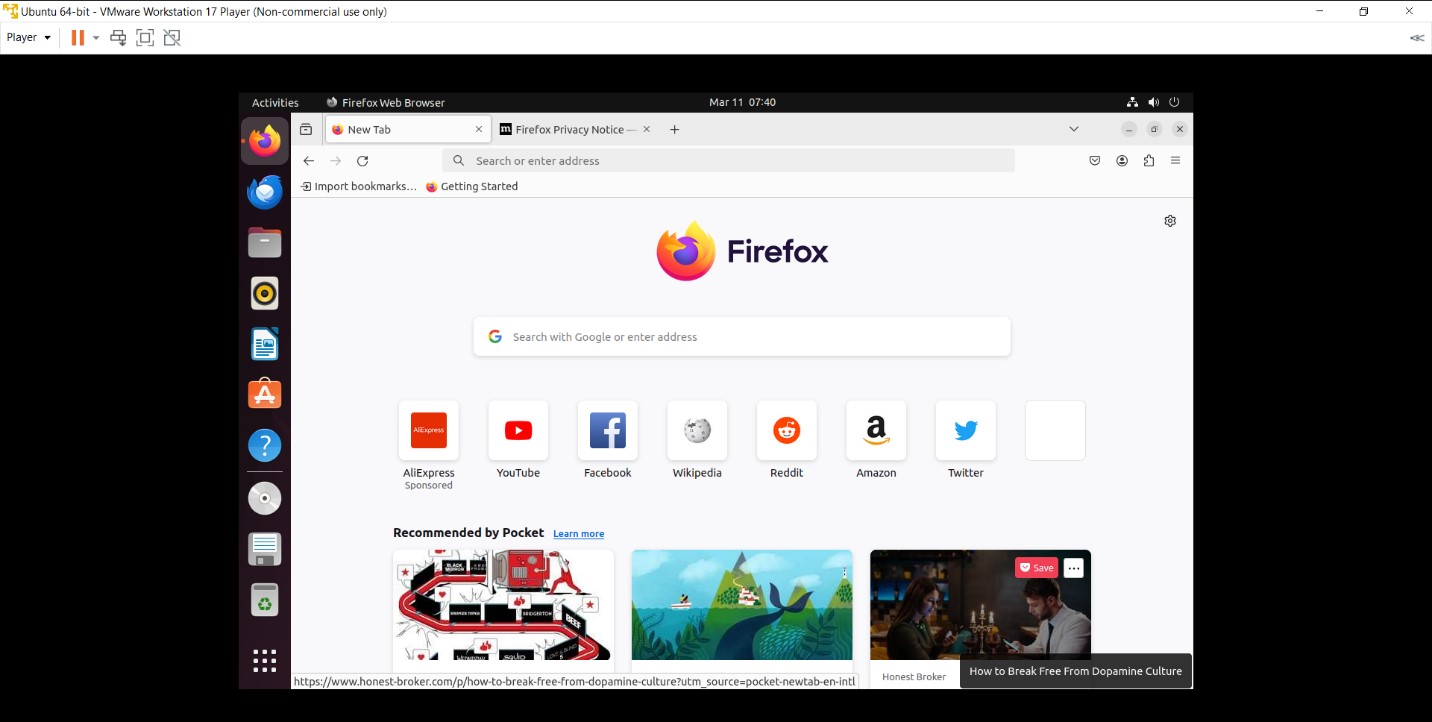


If everything goes according to plan, you now have a fully functioning Ubuntu Linux VM on your Windows PC, as shown below.

Enjoy the versatility of your new virtual machine and the opportunities it offers. Be it for testing, learning, and running different OSes and applications in an isolated and controlled environment.







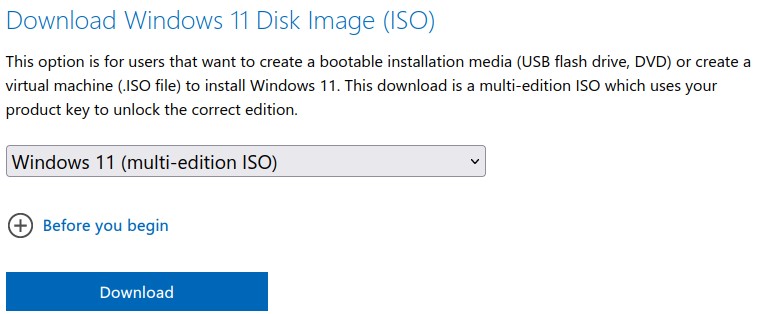
**Result:-** Linux OS (Ubuntu) successfully create on windows 11 as a guest OS.

# EXPERIMENT:4

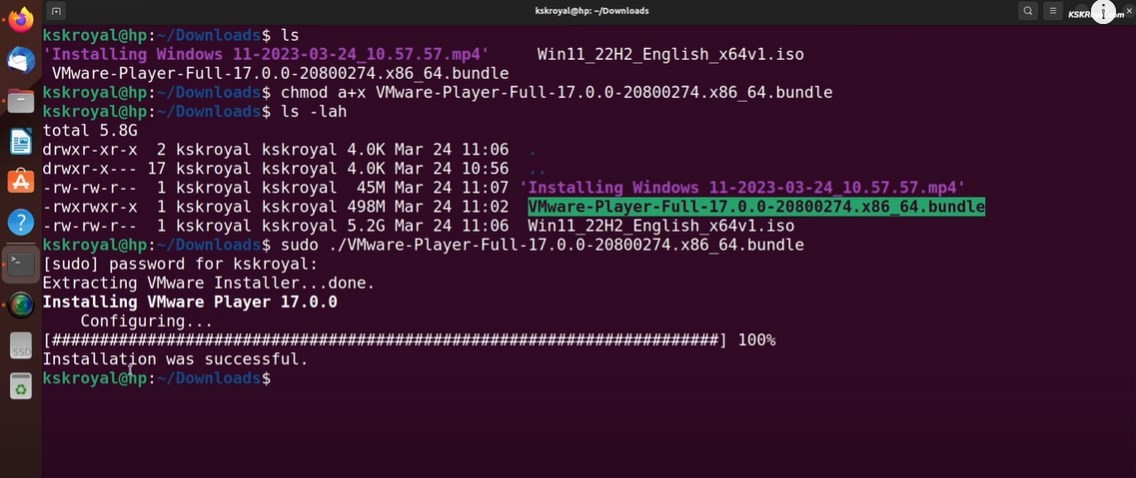
**Objective**:- Create a Virtual Machine of Windows (ex. windows 11) on LinuxOS (Ubuntu).

## Creation Steps:-

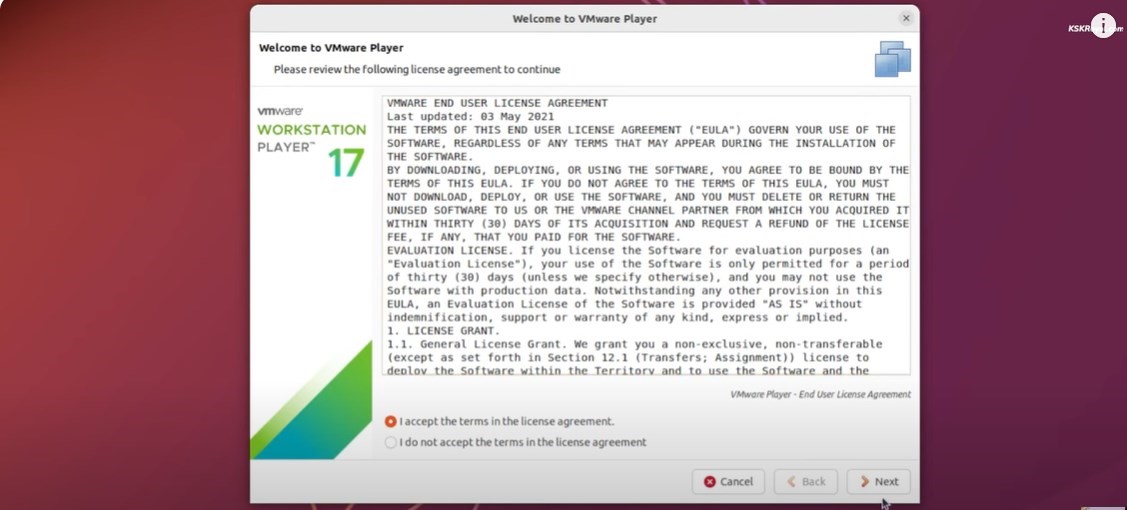
1. To start, we need to download a Windows 11 Disk Image (ISO) from the official website.



1. Run following commands to install Vmware Player in Ubuntu:-



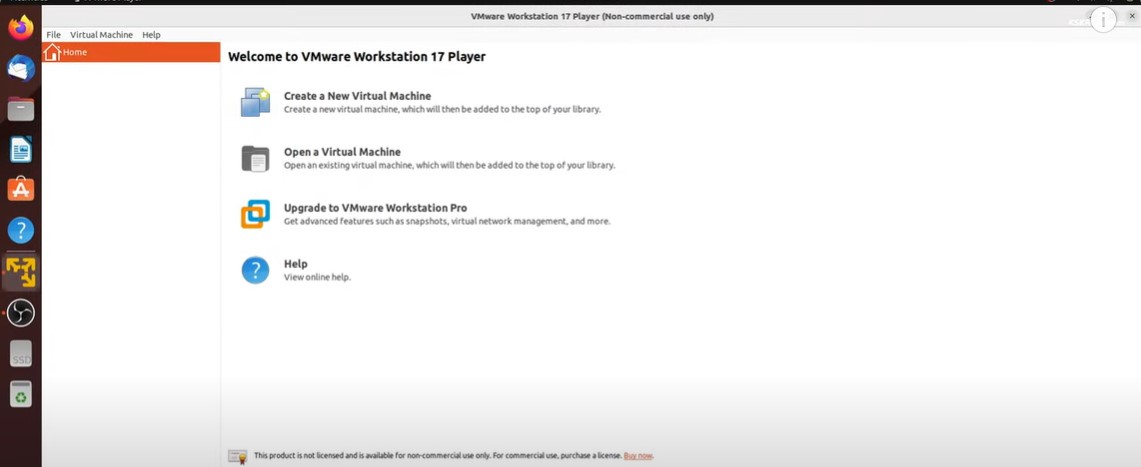
1. Then go to the Vmware Player and accept the license agreement:-



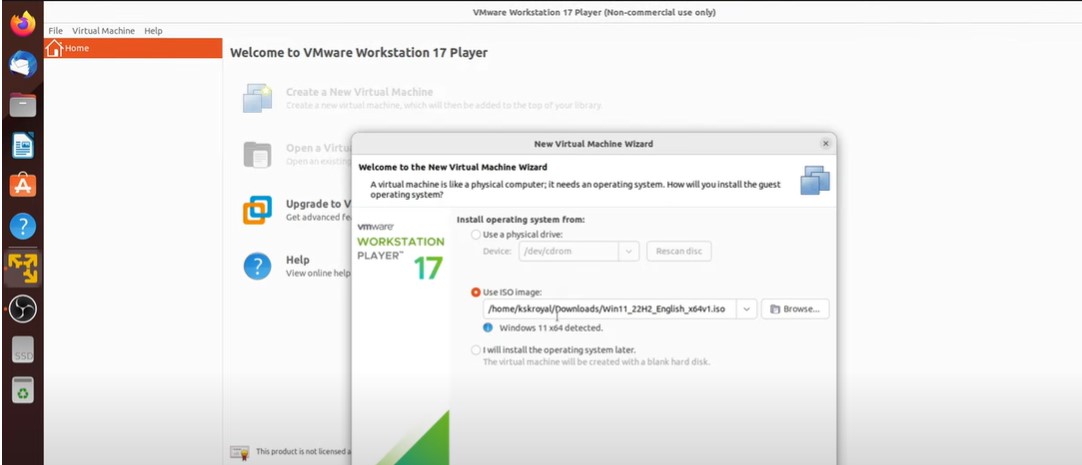
1. Select free foe non-commercial use and set password:-



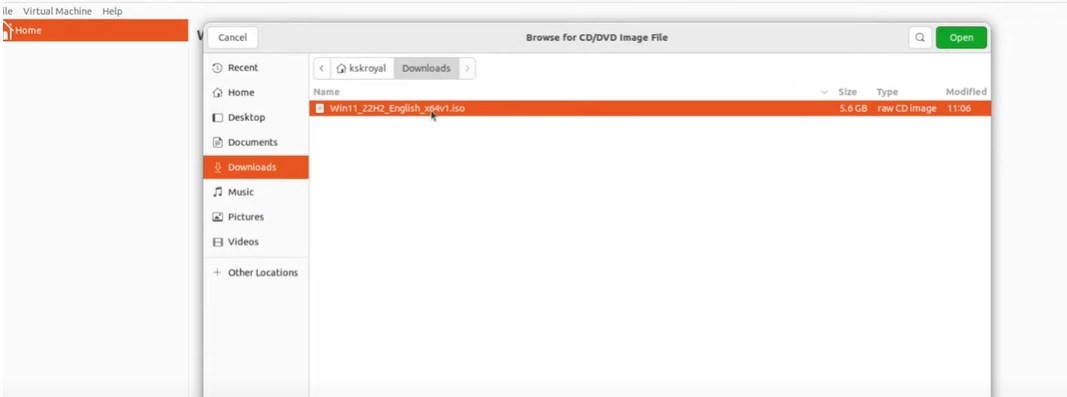
1. Start Vmware Workstation 17 Player to create virtual machine:-



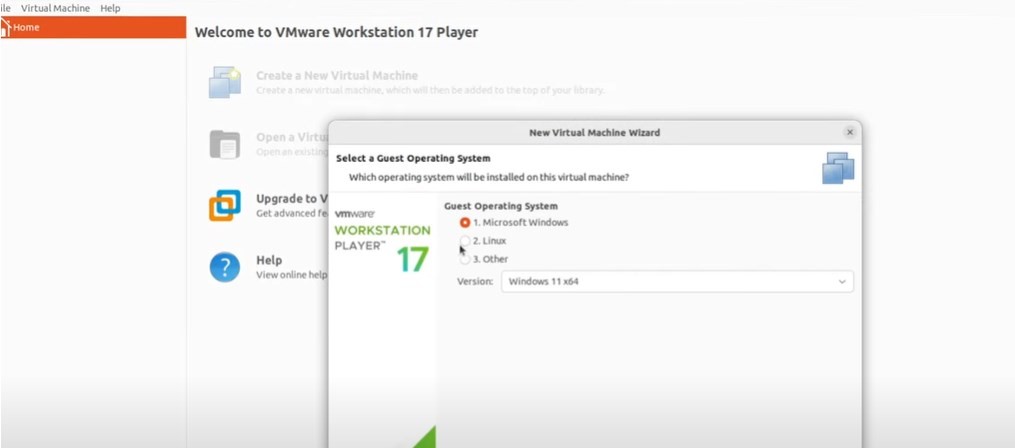
1. Click on Create a New Virtual Machine and select Windows 11 iso file:-



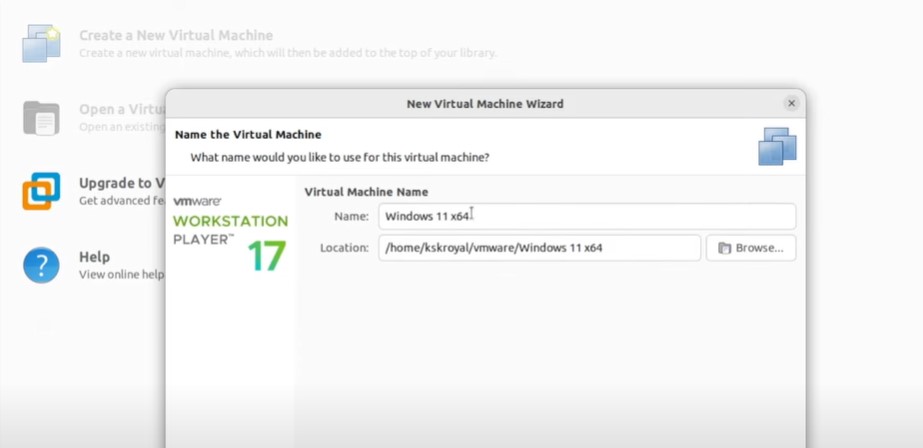
1. Select Windows 11 iso file from downloads:-



1. Select Microsoft Windows as Guest OS:-



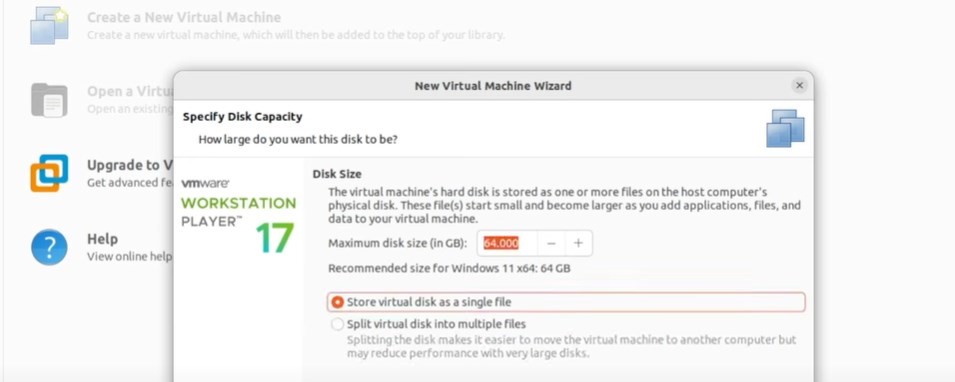
1. Create vietual machine name:-



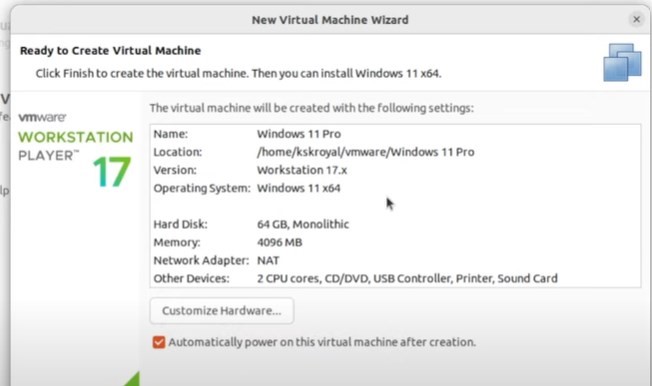
1. Create password that must be strong and confirm password (Note: remember the password):-



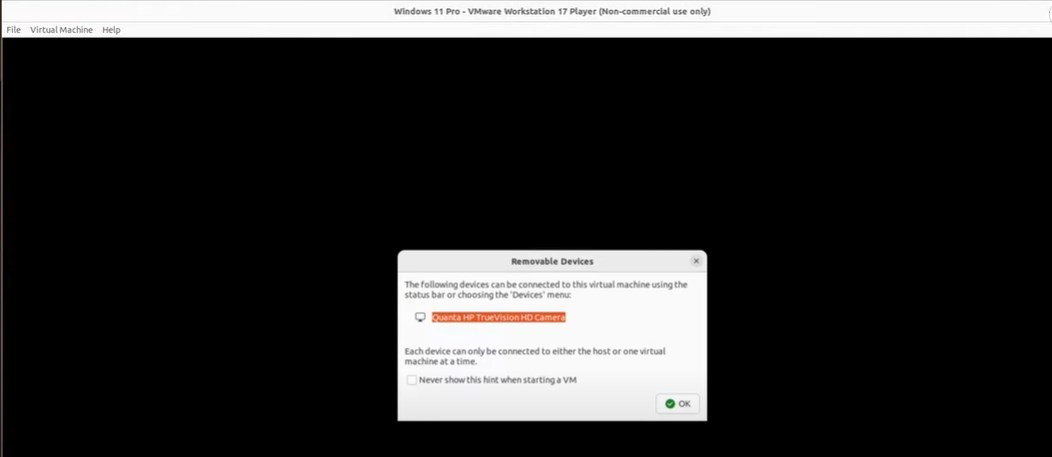
1. Set up the disk size of virtual machine according to your system configuration:-



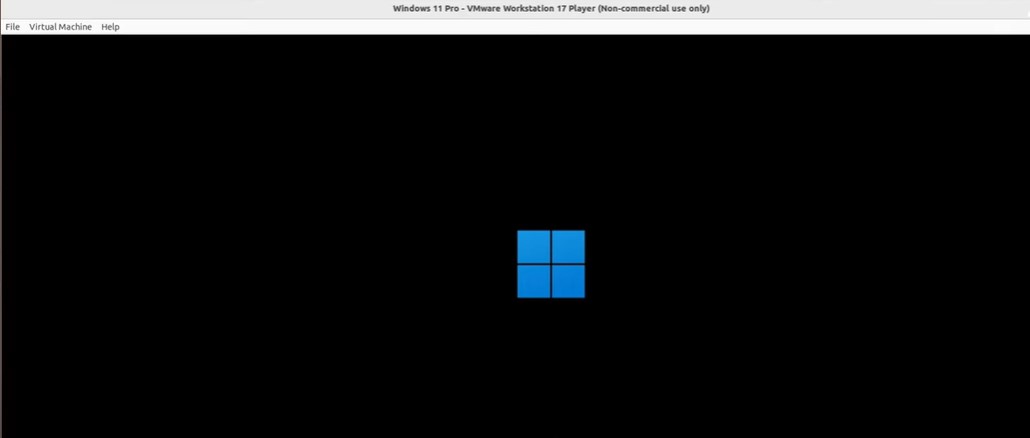
1. New virtual machine wizard that you want to set up:-



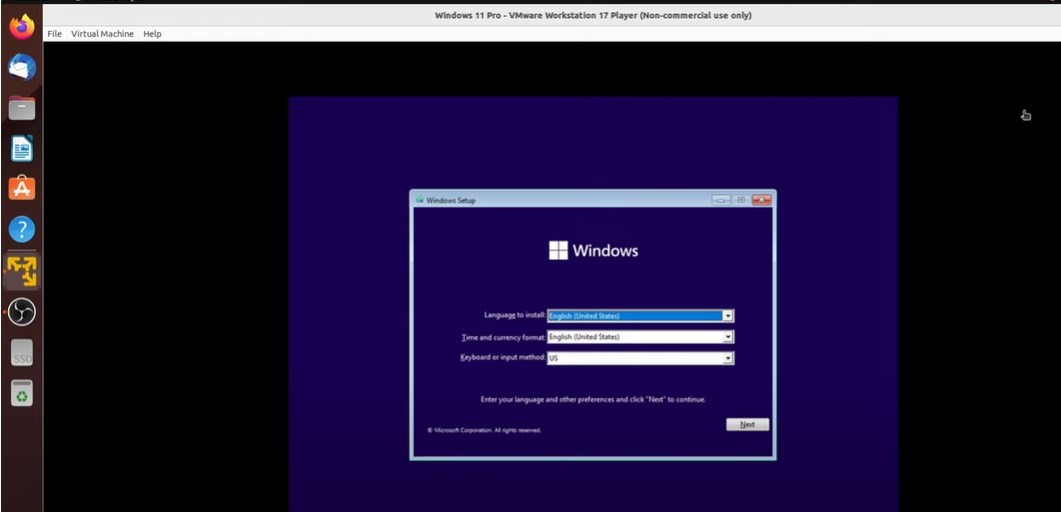
1. Window that open after creating a new virtual machine of windows 11:-



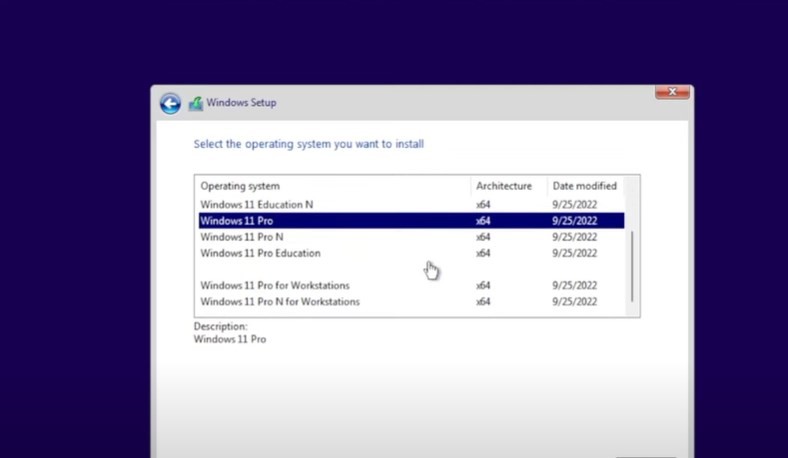
1. Processing to install windows 11:-



1. Windows setup:-



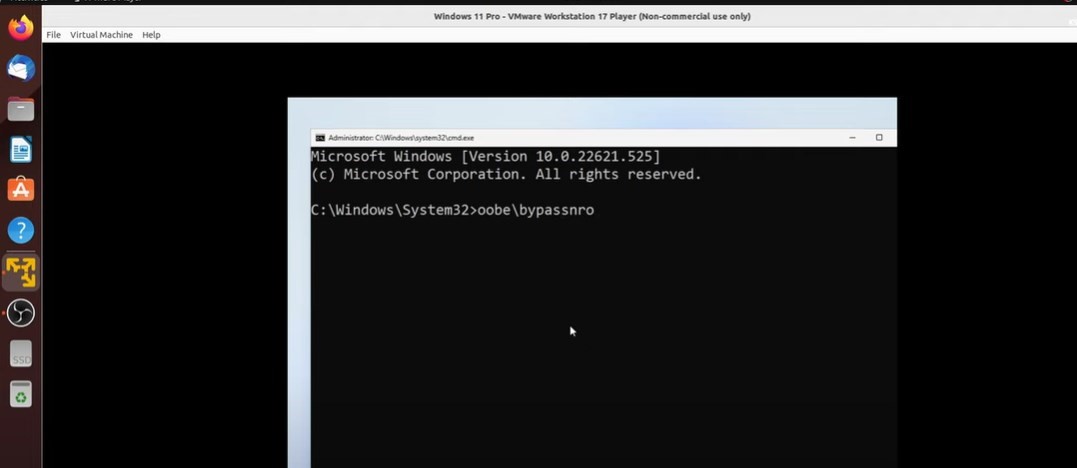
1. Select windows 11 pro:-



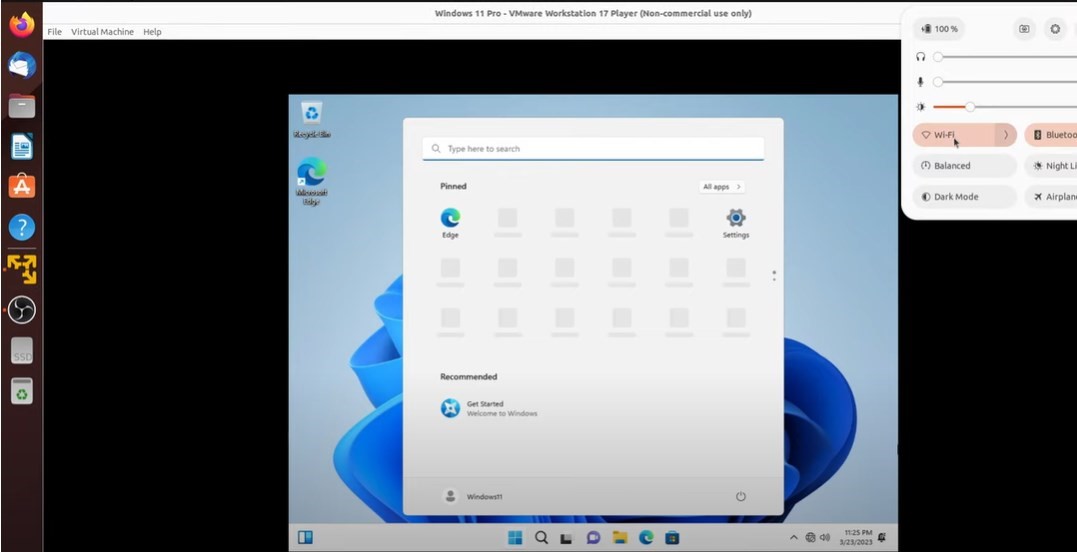
1. Install windows :-



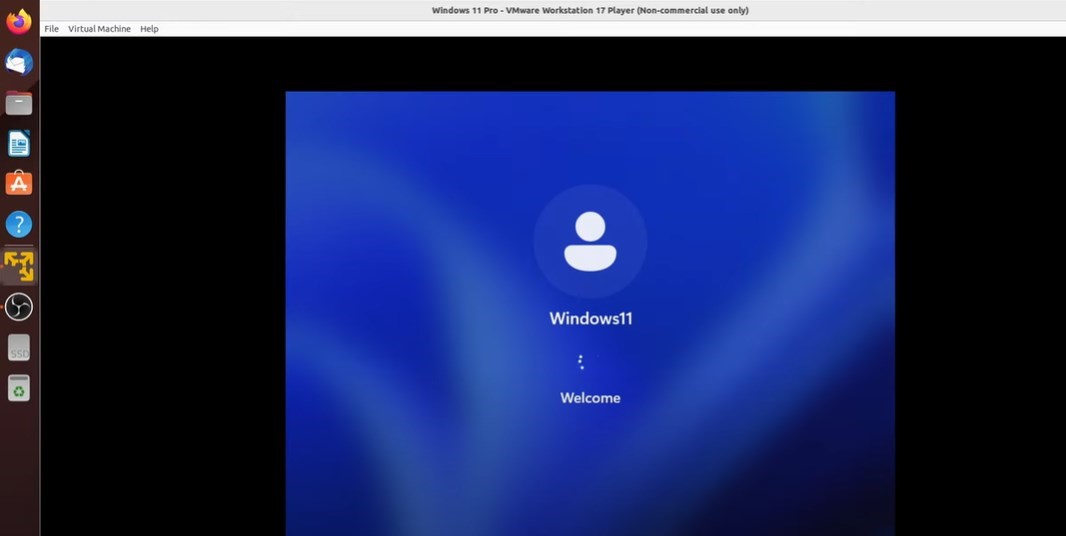
1. Run the below command on cmd and setup the your country, language and keyboard type:-



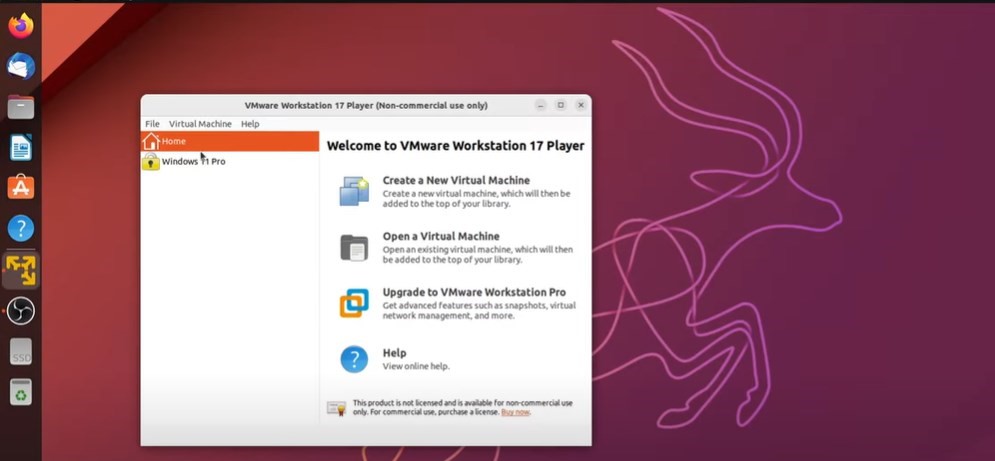
1. Then go to the wi-fi and start the wi-fi:-



1. This the the windows 11 home page after successfully creating the virtual machine:-



1. Restart the system and see the changes:-



**Result:-** Windows 11 successfully create on Linux OS (Ubuntu) as a guest OS.

# EXPERIMENT:5

**Objective**:- Installation of Vmware vSphere Hypervisor and Creatation of Vmware ESXi 7 virtual machine.

**Introduction**:- vSphere Hypervisor:-

The vSphere Hypervisor (formerly known as ESXi), is the free edition of vSphere offering the bare-metal architecture for best possible performance. It installs during boot-time of the Hypervisor host.

The following components traditionally run on the Hypervisor platform:

**Index** - MySQL Server stores the recording and management data

**Dashboard** - Single ISR Dashboard for both the Administrator and User

**FACE API**

**Installing vSphere Hypervisor:-**

**Prerequisites:-**

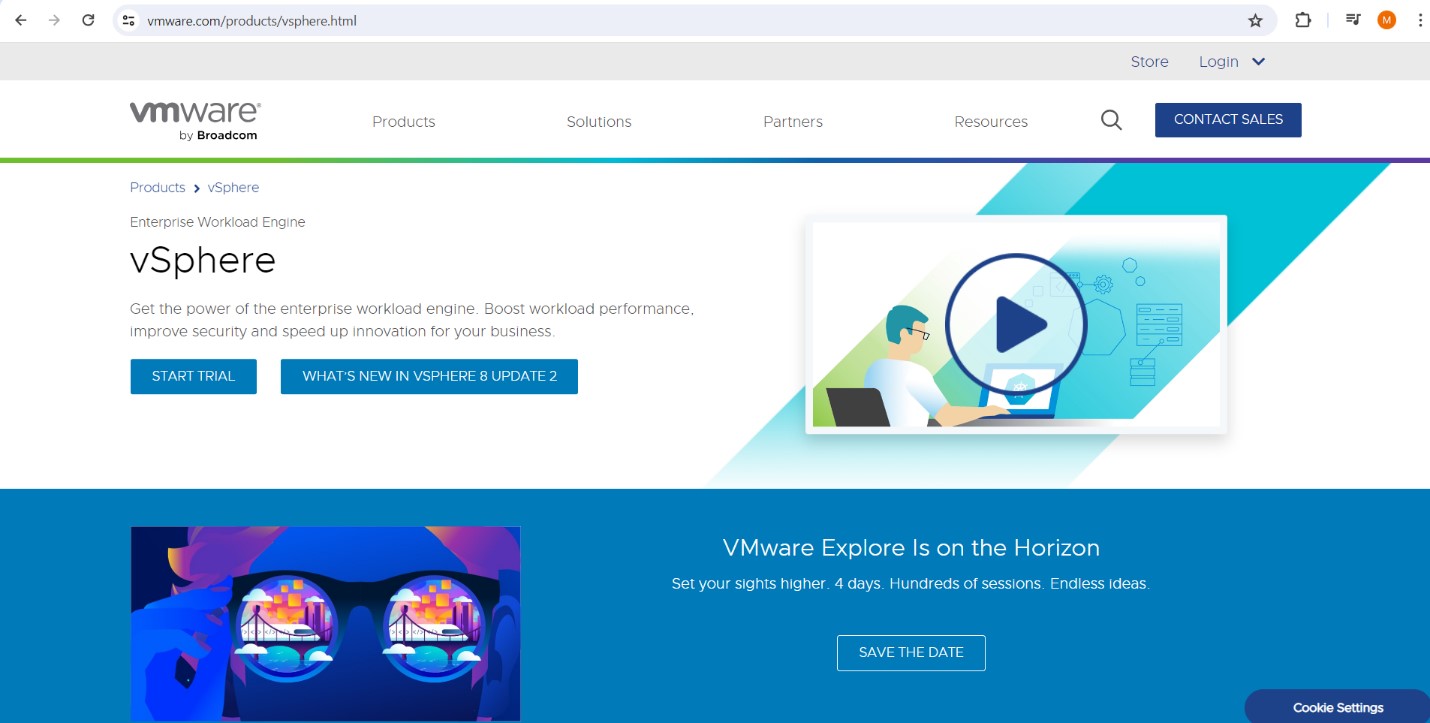
Create a VMware Customer Connect account.

## Procedure:-

To install vSphere Hypervisor:

1. Open a web browser and enter the following URL to navigate to the VMware download page:

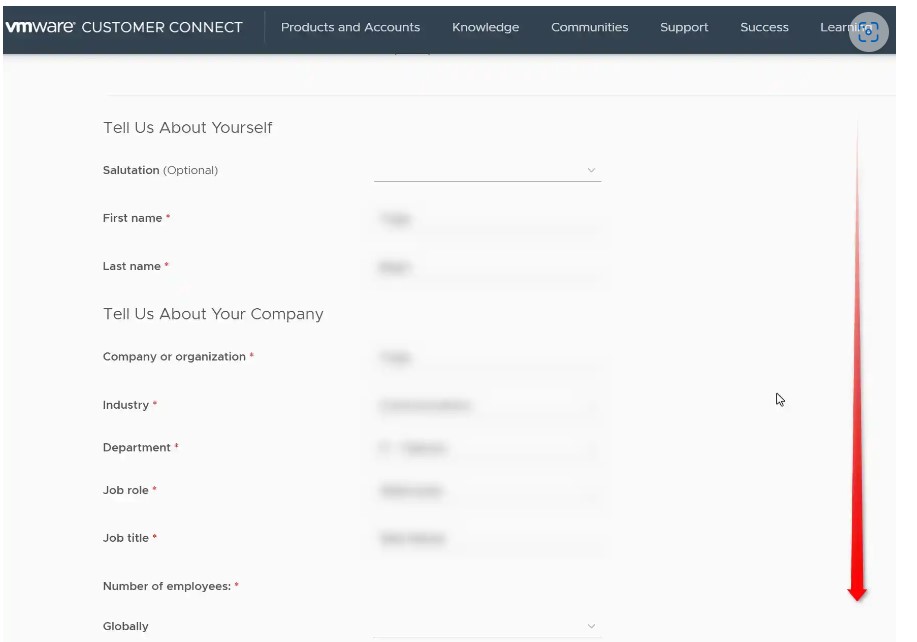
<https://my.vmware.com/web/vmware/evalcenter?p=free-esxi5&lp=default>



1. Download the ESXi 7 update 1 file to your server.

**Note:** You may need to login into the VMware download page with a user name and password before downloading the file. If not already registered, please register and then login to download the applicable file.

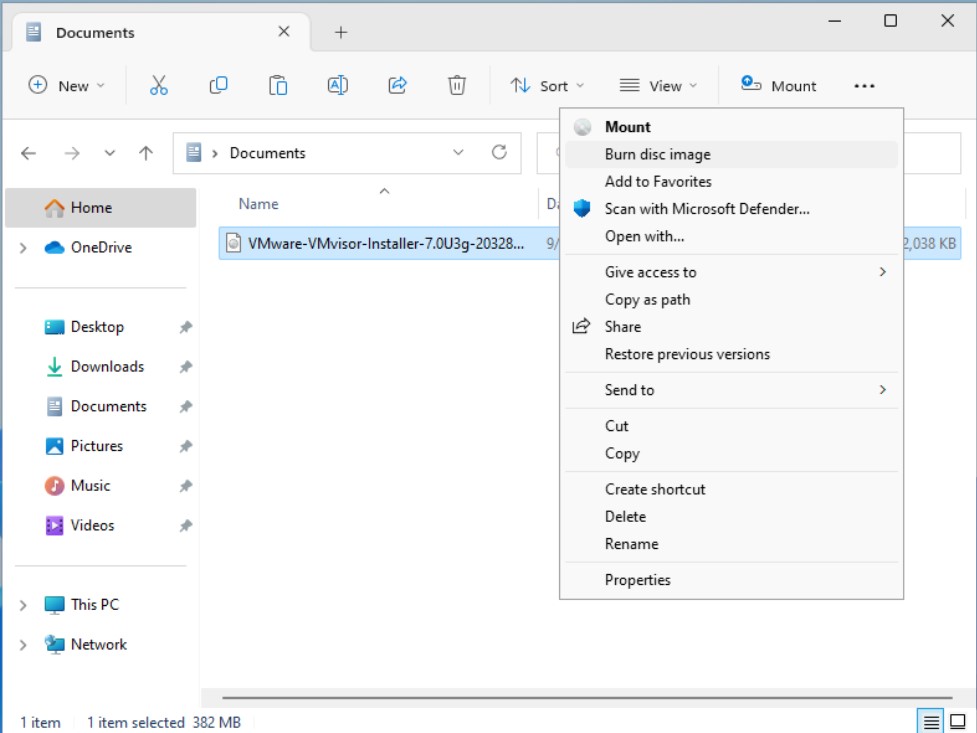




1. After downloading ISO of VMware ESXi 7, Burn it to a DVD with a burning application to create an installation disk.

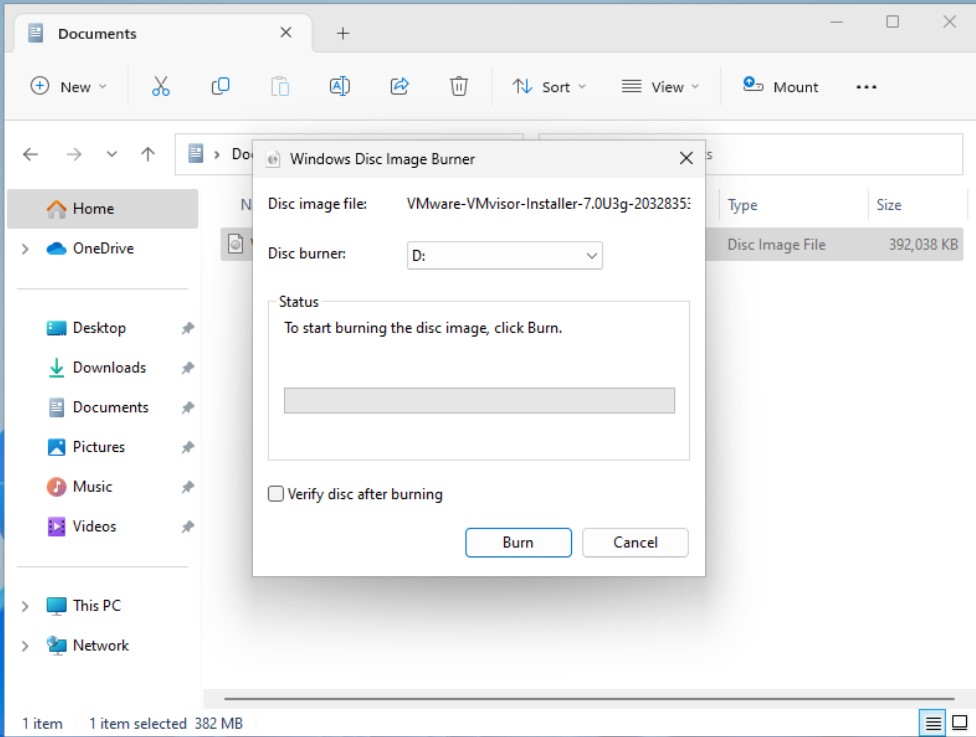
If you create it with the function of Windows 10/11, refer to follows.

1. Insert an empty DVD media in DVD drive. Next, right-click the ISO image of VMware ESXi 7 and select [Burn Disk image] in the menu.

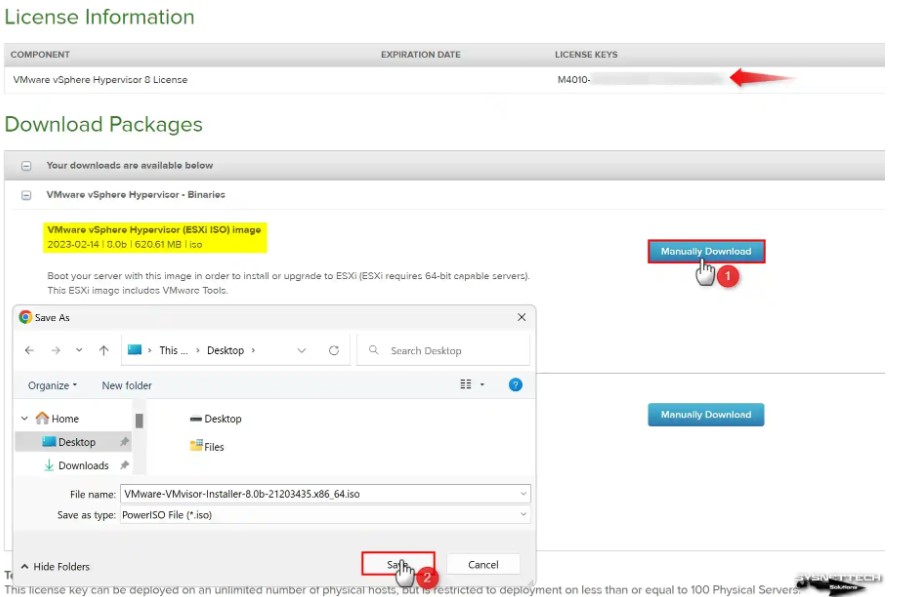


1. Click [Burn] button and start burning.

After finishing burning, eject the disk and proceed to the installation step.



1. Boot the server from the ESXi 7 update 1 CD you just created.
2. At the prompt, press Enter to proceed with the installation.
3. Press <F11> to accept the ESXi 7 license.



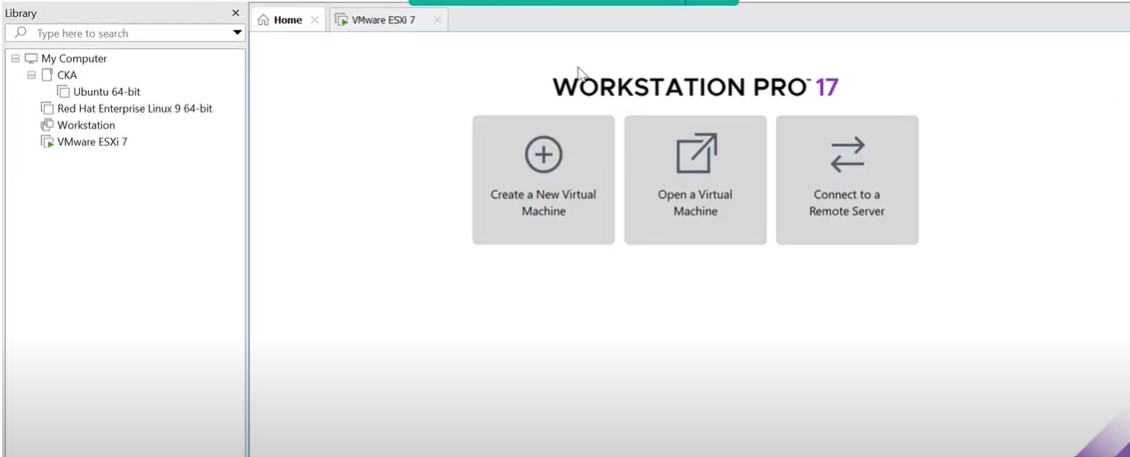
1. At the Select a Disk menu, press Enter to confirm the remote storage device and continue.
2. Press <F11> to install the ESXi 7 update 1.



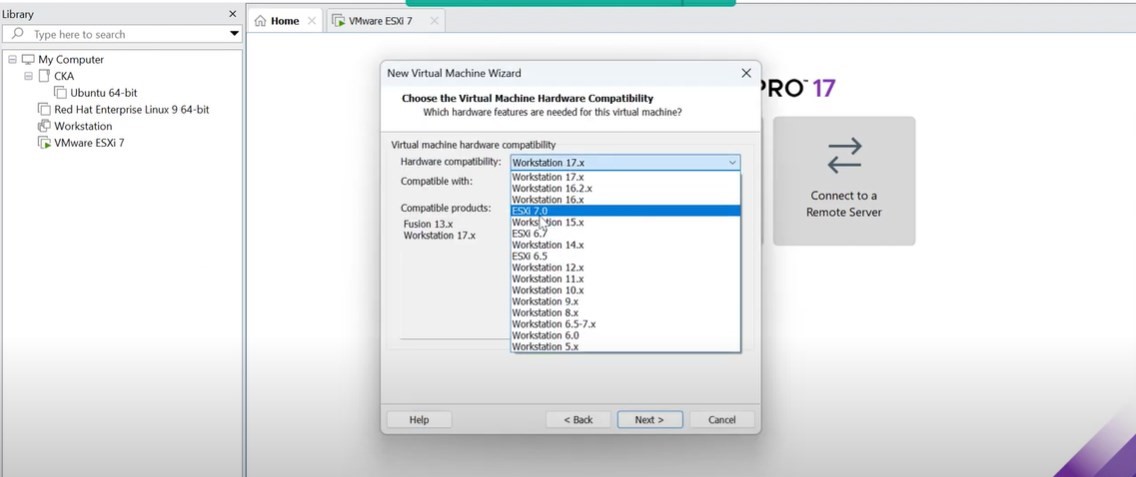
1. When the installation is complete, remove the CD and press Enter to reboot the server.
2. Configure the vSphere Hypervisor using the procedures in Configuring vSphere Hypervisor.

**Creatation of Vmware ESXi 7 virtual machine:-**

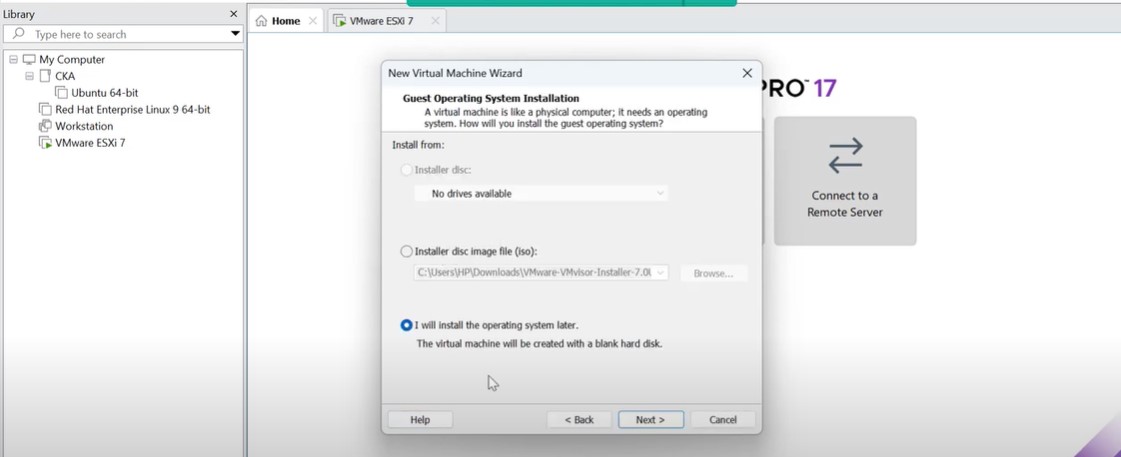
1. Go to the Vmware workstation and click on create a new virtual machine:-



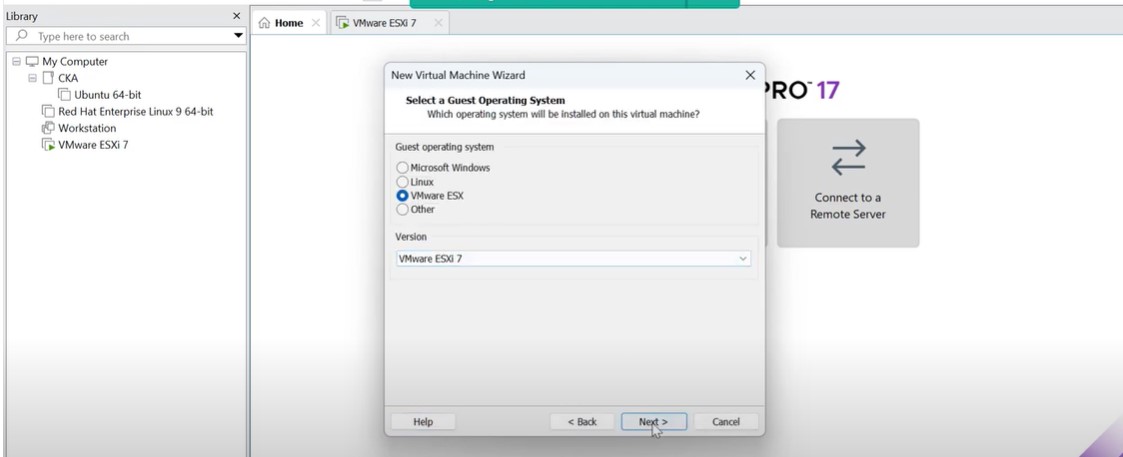
1. Select the ESXi 7.0 for hardware compatibility:-



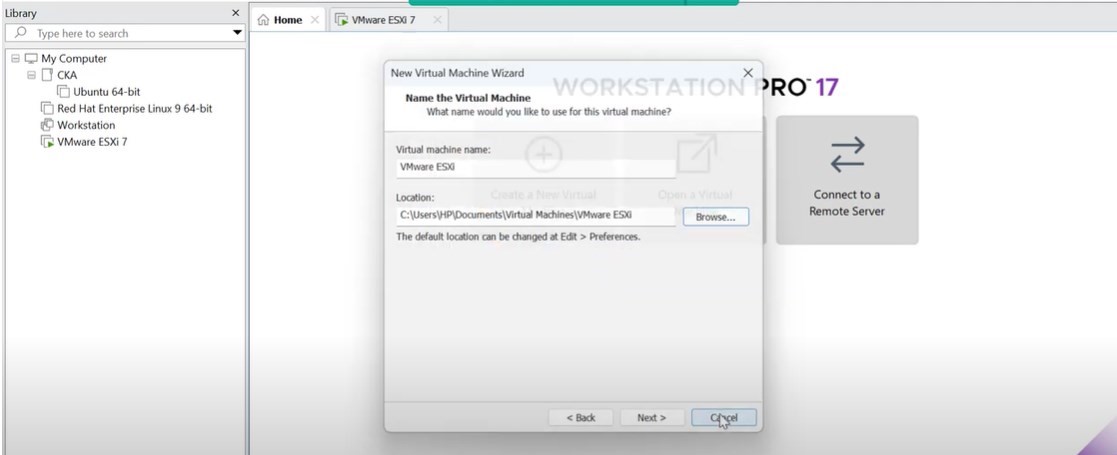
1. After selecting this go to the browse and choose the installed iso file:-



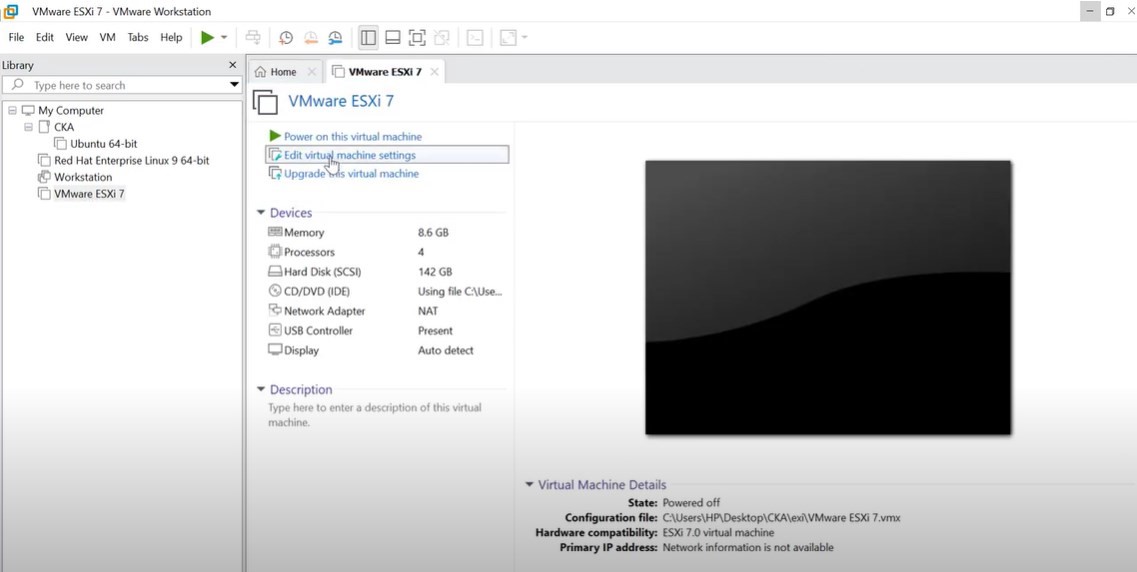
1. Select Vmware ESX:-



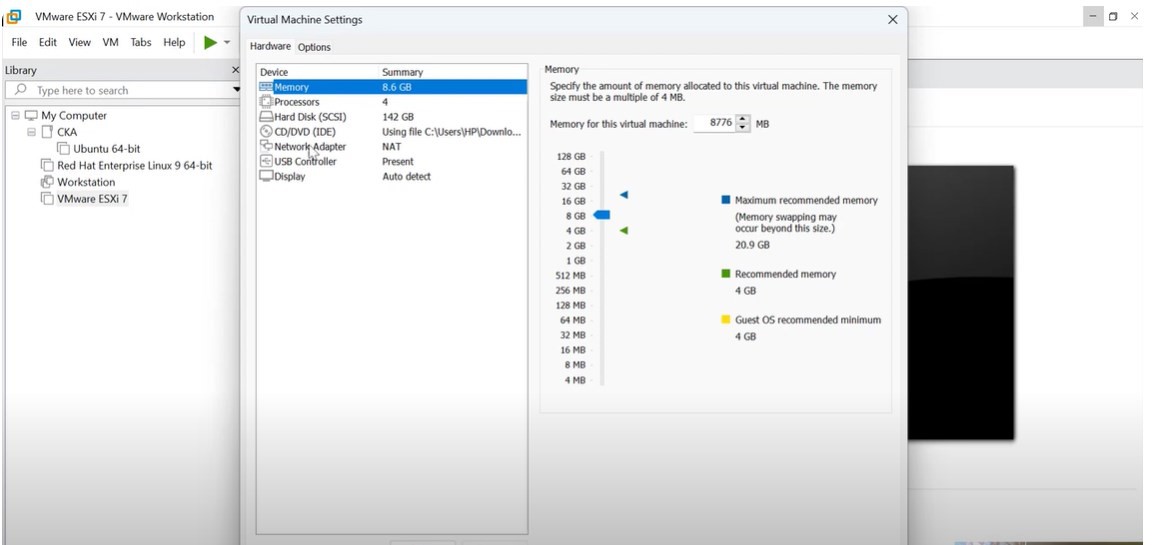
1. Choose the location of the file:-



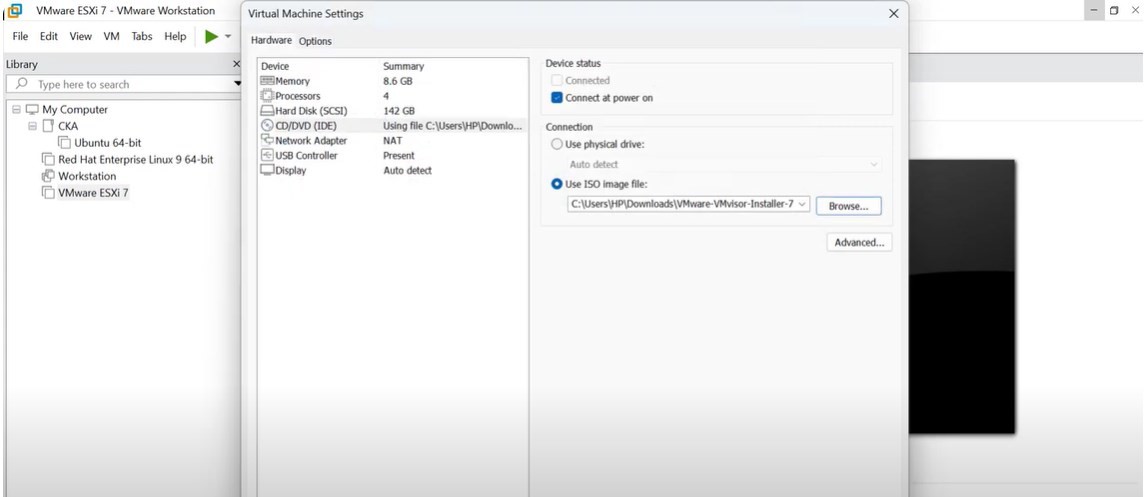
1. After successfully creating this click on the Vmware ESXi 7 and go to the edit virtual machine settings:-



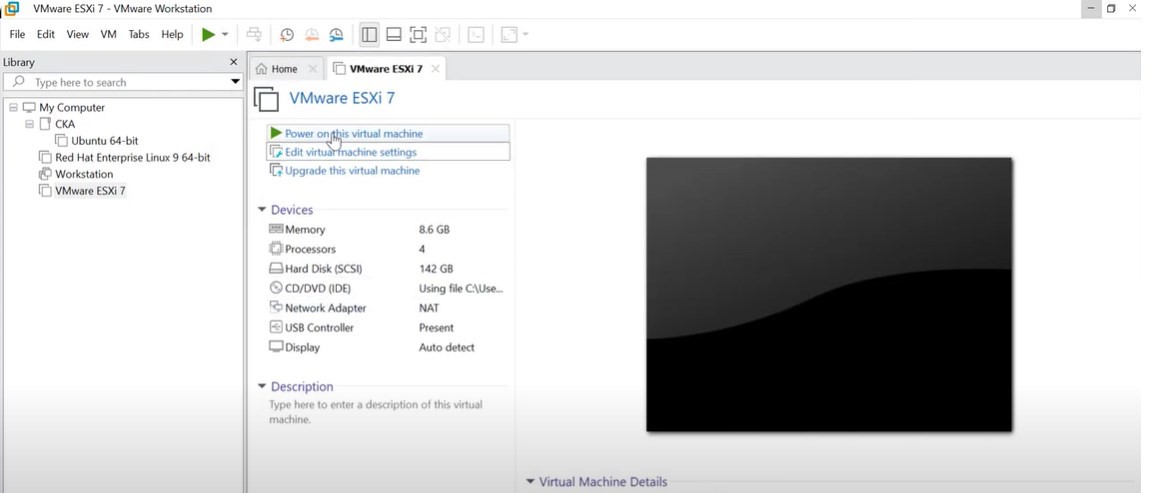
1. Go to the CD/DVD (IDE) option:-



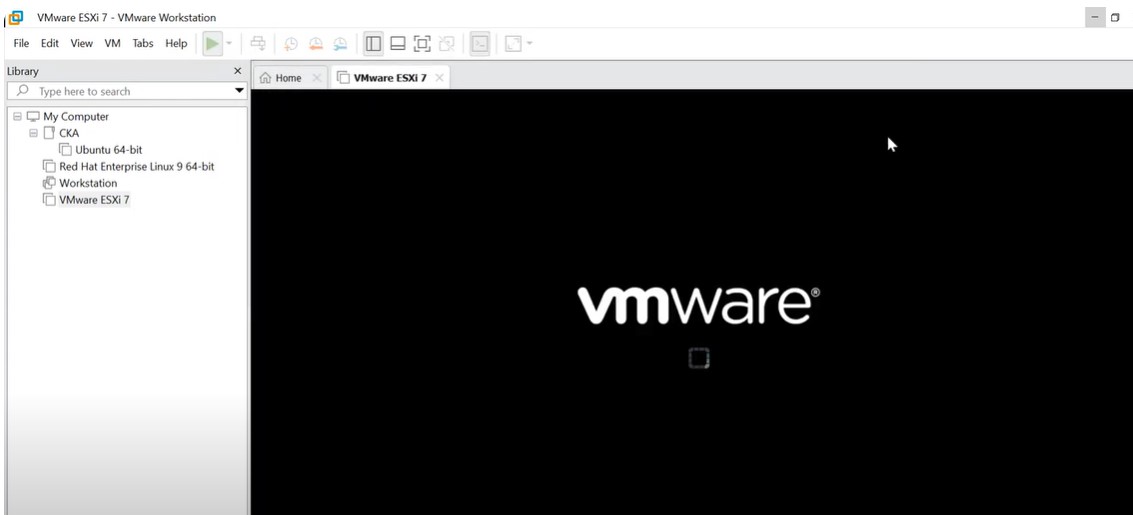
1. Choose the installed iso file:-



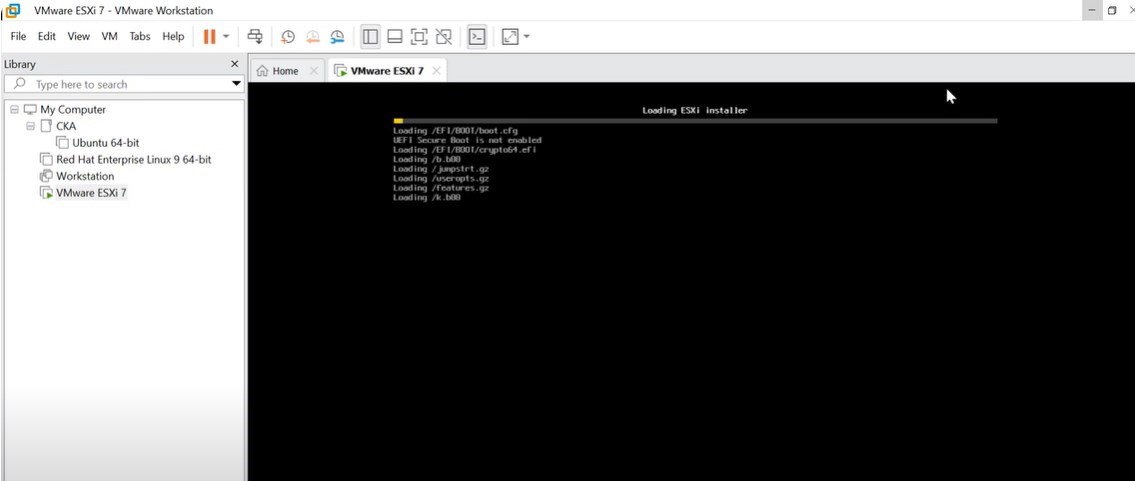
1. After this go to the power on this virtual machine option and click it:-



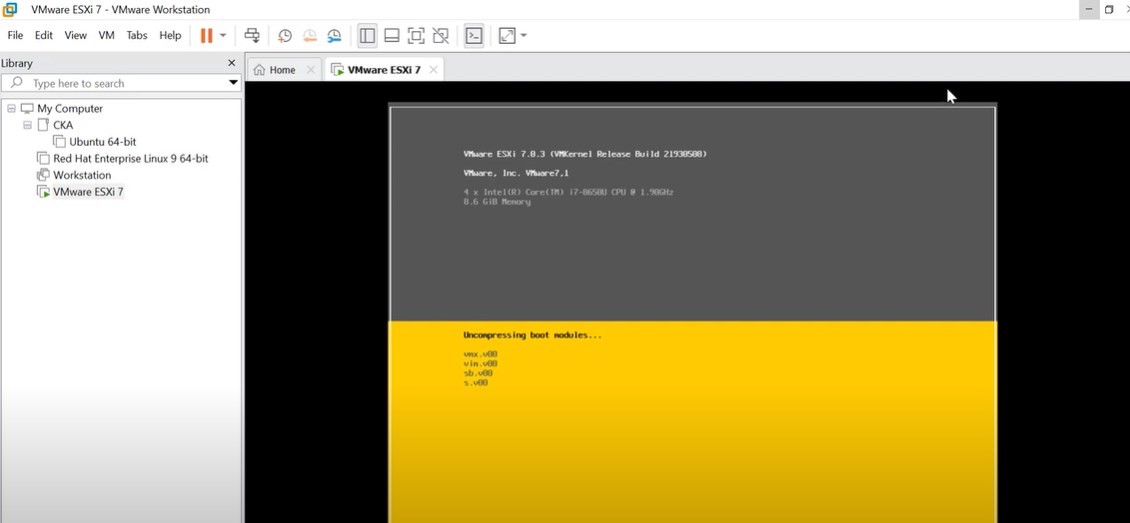
1. Then this window will appear:-



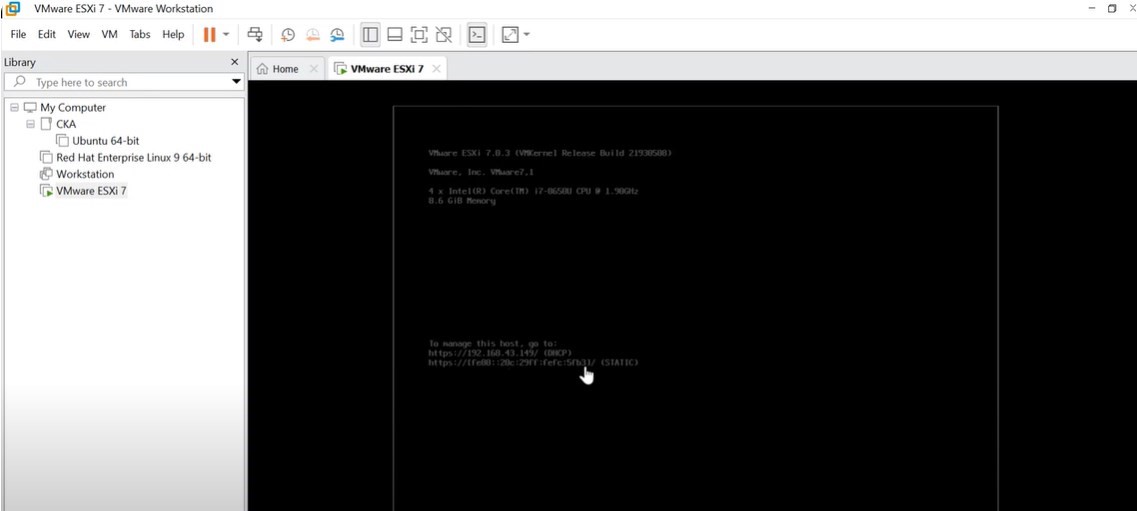
1. Loading the ESXi installer:-



1. After that processing the boot modules:-

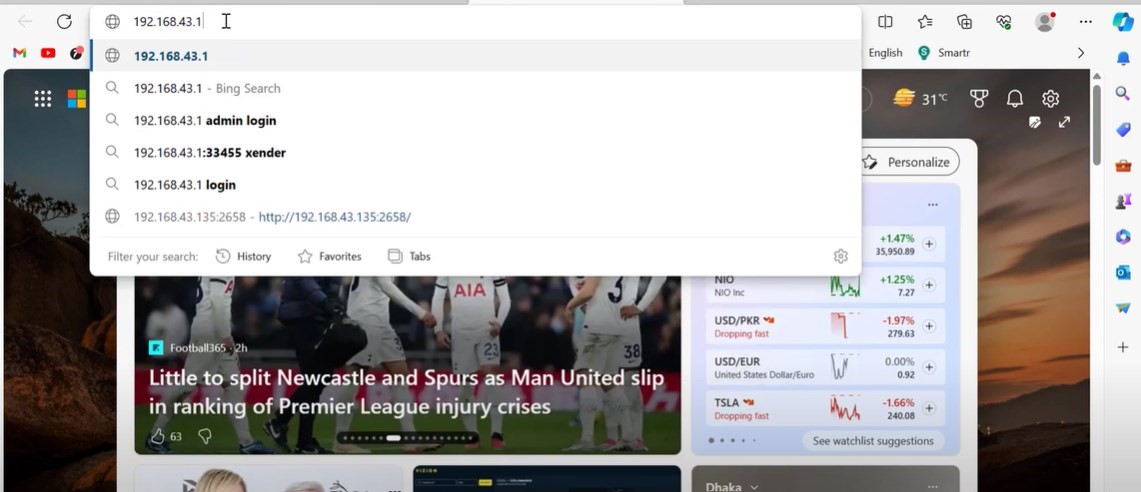


1. After successfully installing the ESXi then it can generate an IP address for your ESXi machine:-

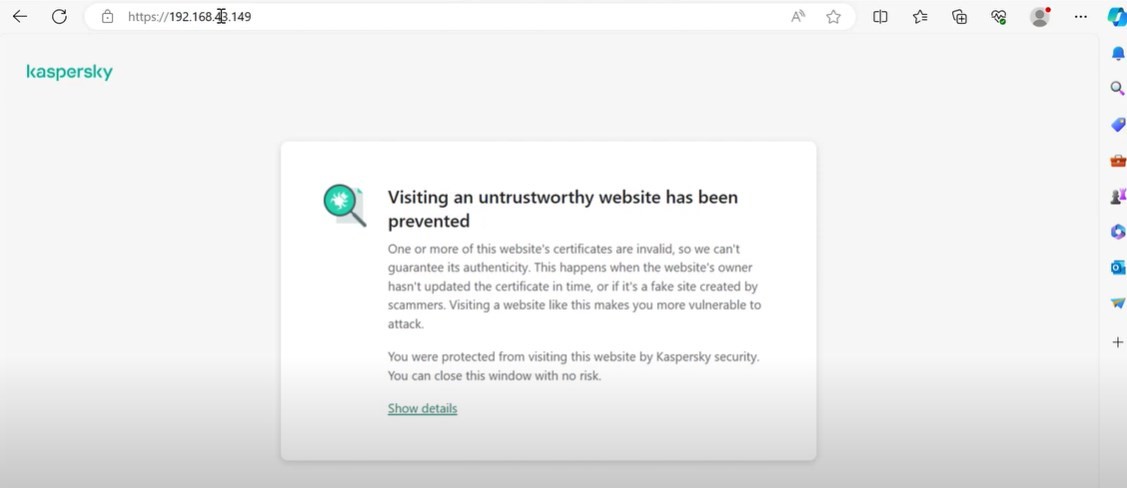


## After that go to the browser:-

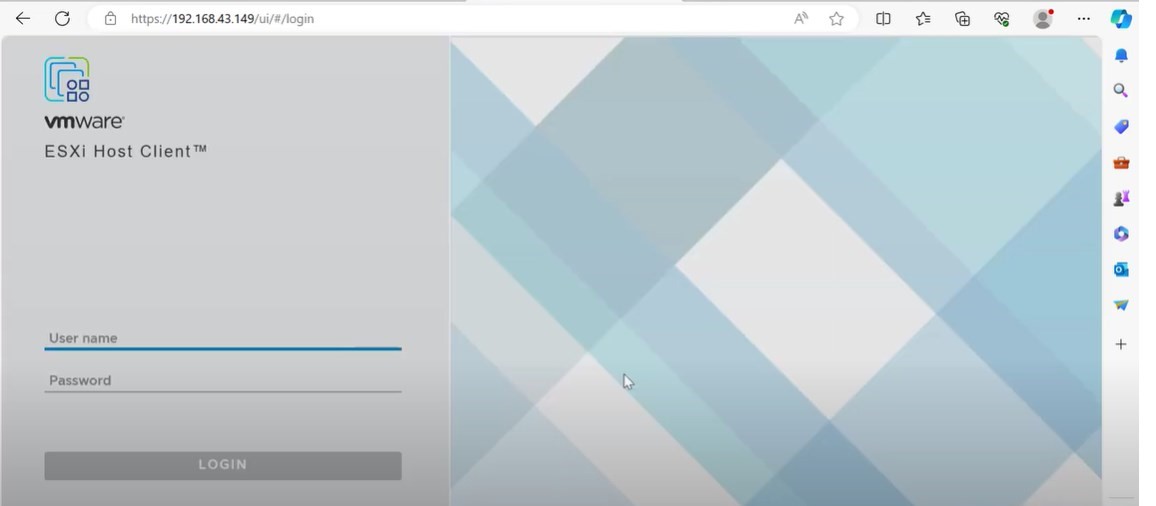
1. Type the created IP address:-



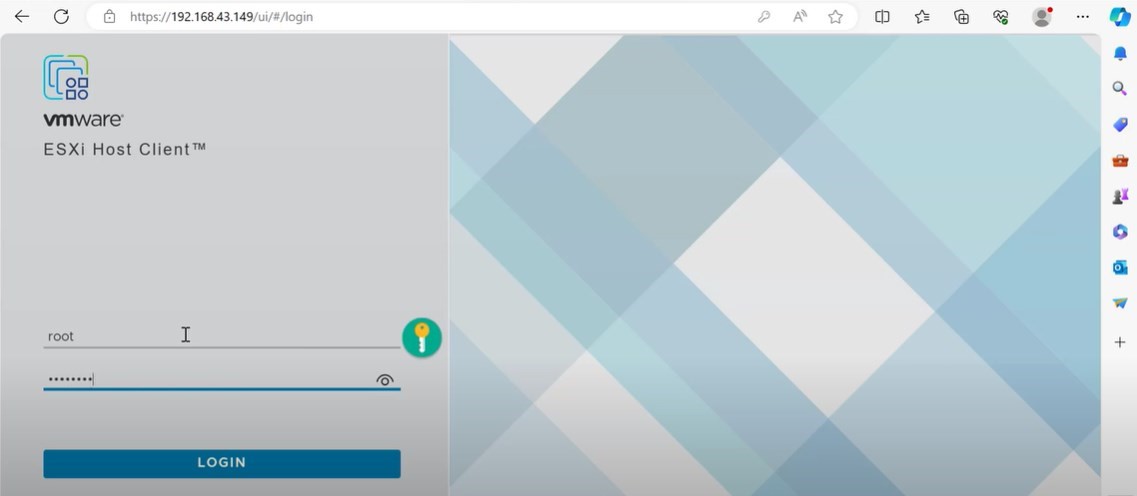
1. Go to the IP address and click on show details and click on continue button:-



1. Type your user name and password:-



1. After typing user name and password click on login button:-



1. After that we can forwarding to the vmware client page.

**Result:-** vmware vShpere ESXi 7 successfully installed and created.

# EXPERIMENT:6

**Objective**:- Working in cloud9 to demonstrate different language.

## Introduction:-

AWS Cloud9 is an integrated development environment, or IDE.

The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers, and a built-in terminal. It contains a collection of tools that you use to code, build, run, test, and debug software, and helps you release software to the cloud.

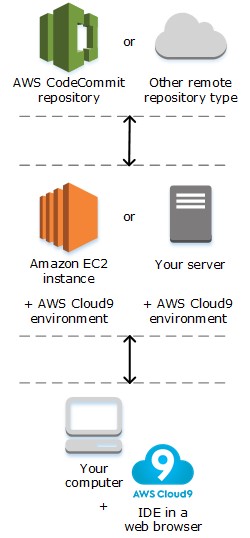
You access the AWS Cloud9 IDE through a web browser. You can configure the IDE to your preferences. You can switch color themes, bind shortcut keys, enable programming language-specific syntax coloring and code formatting, and more.

## Theory:-

How does AWS Cloud9 work:-

The following diagram shows a high-level overview of how AWS Cloud9 works.

From the diagram (starting at the bottom), you use the AWS Cloud9 IDE, running in a web browser on your local computer, to interact with your AWS Cloud9 environment. A computing resource (for example, an Amazon EC2 instance or your own server) connects to that environment. Finally, your work is stored in an **AWS CodeCommit repository or other type of remote repository**.



## AWS Cloud9 environments:-

An AWS Cloud9 environment is a place where you store your project's files and where you run the tools to develop your applications.

Using the AWS Cloud9 IDE, you can:

* Store your project's files locally on the instance or server.
* Clone a remote code repository—such as a repo in AWS CodeCommit—into your environment.
* Work with a combination of local and cloned files in the environment.

You can create and switch between multiple environments, with each environment set up for a specific development project. By storing the environment in the cloud, your projects no longer need to be tied to a single computer or server setup. This enables you to do things such as easily switch between computers and more quickly onboard developers to your team.

## Steps :-

Using Cloud9 is a great choice for demonstrating different programming languages. Here's a general guide on how you can set up and utilize Cloud9 for this purpose:

1. **Sign Up/Login**:- If you haven't already, sign up for a Cloud9 account or log in using your existing credentials. By following this link:-

Open <https://portal.aws.amazon.com/billing/signup>

1. **Create a New Environment**:- Once logged in, you can create a new environment by clicking on the "Create Environment" button.

Choose the type of environment you want (e.g., Ubuntu, Amazon Linux, etc.). The choice depends on the specific requirements of the languages you plan to demonstrate.

1. **Configure Environment**:- After selecting the environment type, you can configure it by giving it a name and description.

You can also choose the size of the instance depending on your requirements.

1. **Accessing the IDE**:- Once your environment is created, you'll be directed to the Cloud9 integrated development environment (IDE) where you can write, edit, and run code.

1. **Language Installation**:- Depending on the languages you want to demonstrate, you may need to install them if they're not already available in the environment.

For example, if you want to demonstrate Python, Node.js, Java, etc., you might need to install them using package managers like apt, npm, or by downloading and installing directly.

1. **Creating Files**:- You can create new files for each language you want to demonstrate. For example, create a file named hello\_world.py for Python, hello\_world.js for JavaScript, etc.

* + To create a new file, click "File" -> "New File" and choose the appropriate file extension (e.g., ".py" for Python, ".js" for JavaScript).
  + Write your code, and Cloud9 will provide language-specific suggestions as you type.

1. **Writing Code**:- Write the code for each language in the corresponding file you created. For example, print "Hello, World!" in Python, JavaScript, etc.

1. **Running Code**:- Cloud9 allows you to run the code directly within the IDE. Simply select the file containing the code you want to execute and click the "Run" button.

You'll see the output of your code in the console window within the IDE.

1. **Demonstration:-** You can now demonstrate the functionality of different languages by executing the code you've written and explaining how each language works.

1. **Saving and Sharing**:- Make sure to save your work periodically to avoid losing any changes.

You can also share your Cloud9 environment with others if you want them to see or collaborate on your demonstration.

## Example (Python):-

Create a new file named hello\_world.py.

Write the following code:

print("Hello, world!")

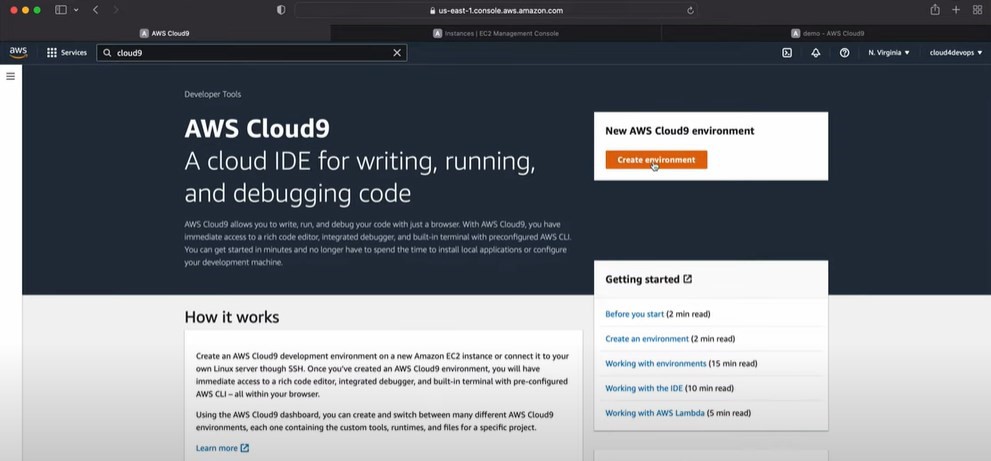
In the terminal pane, navigate to the directory containing hello\_world.py.

Run the code using python **hello\_world.py**.

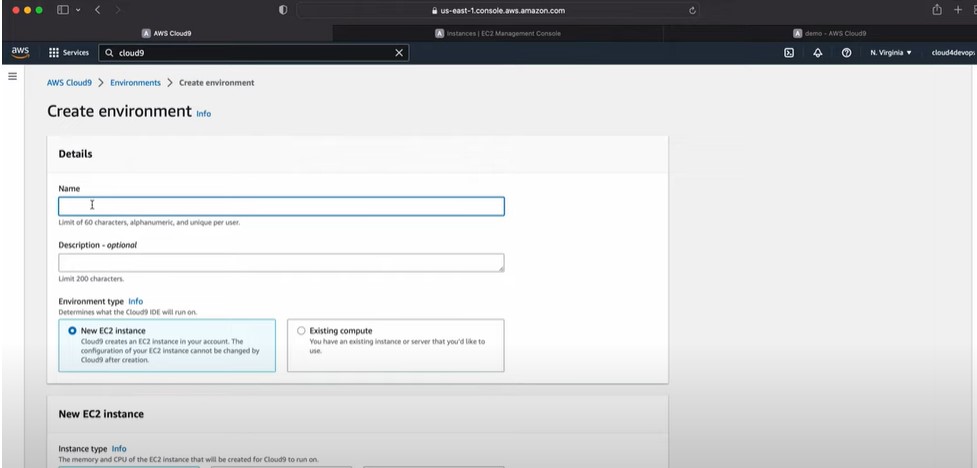
The terminal will print **"Hello, world!"**.

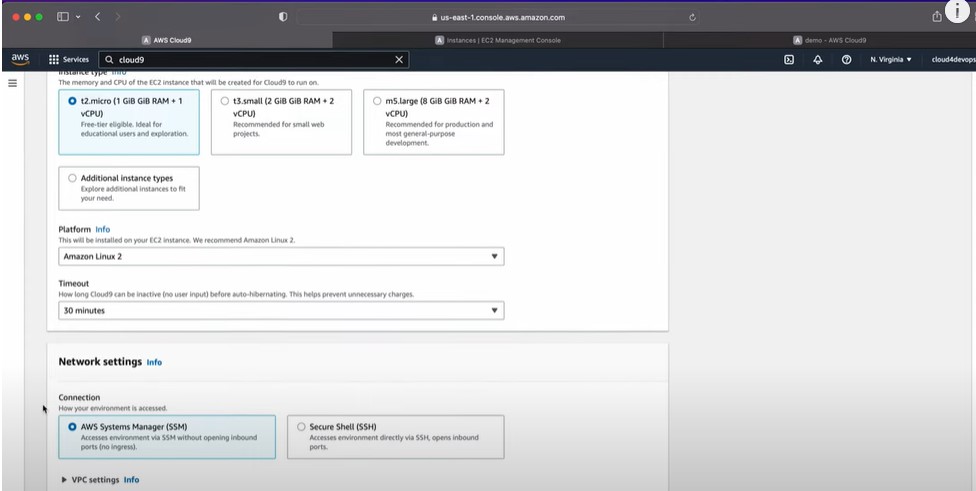
## Creating a New Environment steps:-

1. Click on the create environment:-

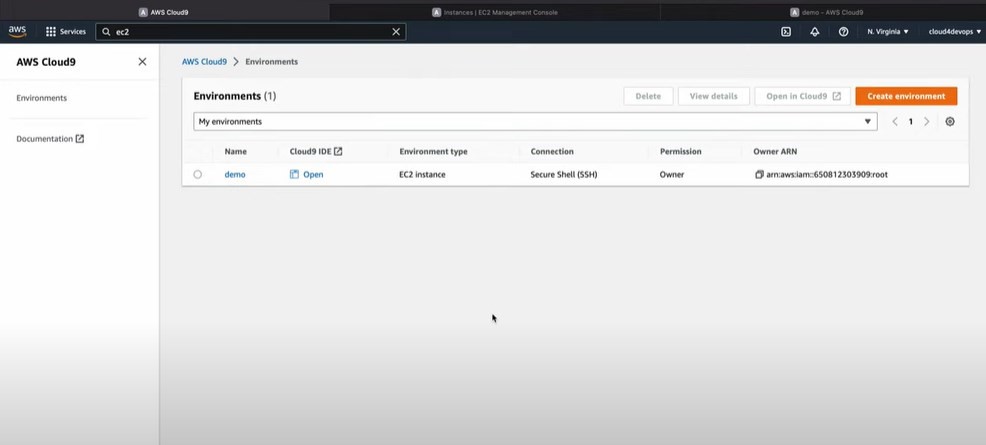


1. Fill the required details to create new environment:-

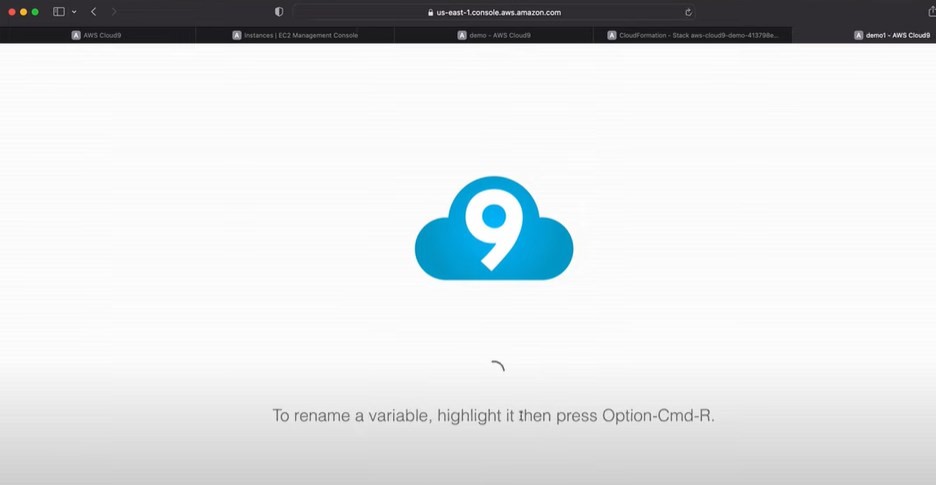




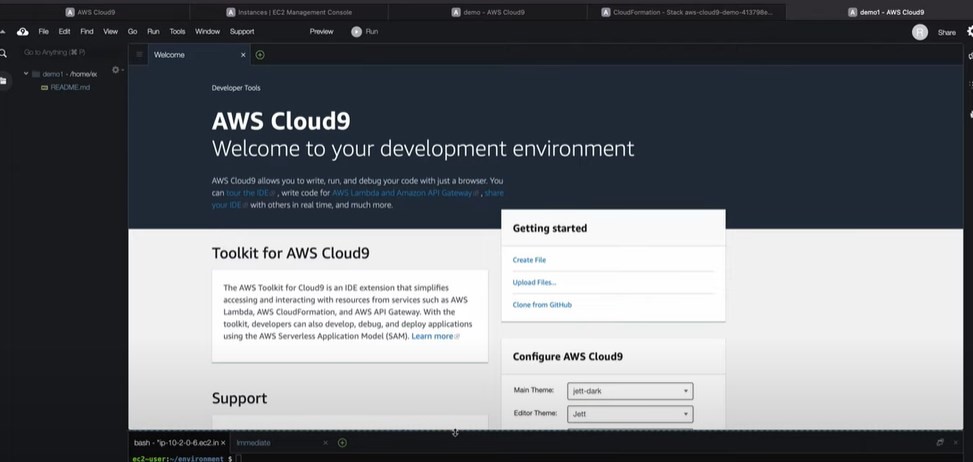
1. At the end click on the create button.
2. Succesfully cretaed the new environment.



1. Click on the open :-



1. Last terminal where you can work to demonstrate the different language.



**Result**:- Successfully demonstrate different language in cloud9.

# EXPERIMENT:7

**Objective**:- Working in codenvy to demonstrate provisioning and scaling of a website.

**Introduction**:- Codenvy IDE:-

By using this IDE you code, build, deploy and test your projects right in the cloud. It supports a wide range of programming languages and frameworks and allows you to share your workspace with colleagues. You can easily clone your environment, use tons of different plugins and project management tools for web application development, without ever leaving your IDE.

## o Setting Up Your Codenvy Workspace:-

1. **Create a Codenvy Account** (if you don't have one): Sign up for a free trial or use a paid plan depending on your needs.

[(https://chrome.google.com/webstore/detail/codenvy/lefigjbiimiemfhjmibbg emkpenelmag)](https://chrome.google.com/webstore/detail/codenvy/lefigjbiimiemfhjmibbgemkpenelmag)

1. **Create a New Workspace**: Choose a workspace type that aligns with your project requirements (e.g., Node.js, Java, Python). This sets up the basic environment with necessary tools pre-installed.

o **Building Your Website**:

1. **2Clone or Import Your Code**: If you have existing website code, import it from a Git repository or upload it directly into the workspace. Codenvy integrates seamlessly with Git for version control.

1. **Dockerfile for Provisioning**: Create a Dockerfile that defines the environment for your website. This file specifies the operating system, dependencies, and application installation steps. Here's a basic example:

## Dockerfile

FROM node:16-alpine

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

EXPOSE 3000 # Expose the port your website runs on

CMD [ "npm", "start" ] # Command to start the website

This Dockerfile uses a Node.js base image, installs dependencies, copies your code, exposes port 3000 (adjust if needed), and starts the website using npm start.

* **Provisioning the Website**:

* 1. **Build the Docker Image**: In the Codenvy terminal, navigate to the directory containing your Dockerfile and run docker build -t my-website-image . (replace my-website-image with your desired image name). This creates a Docker image encapsulating your website environment.

* 1. **Run a Container Instance**: Use docker run -d -p 8080:3000 my-websiteimage (replace 8080 with the desired external port) to launch a container from the built image. The -d flag runs the container in detached mode, and -p maps the container's port (3000) to the host's external port (8080, typically used for web traffic).

* **Scaling Your Website**:

* 1. **Horizontal Scaling (Adding More Instances)**: To handle increased traffic, you can run multiple container instances of your website image. Codenvy offers tools like Kubernetes for container orchestration, which can manage scaling automatically based on defined metrics.

* 1. **Vertical Scaling (Increasing Resources)**: If your website needs more processing power or memory, you can adjust the resource allocation for the container instance. Codenvy allows specifying CPU cores, memory limits, and disk space for containers.

* **Monitoring and Management**:

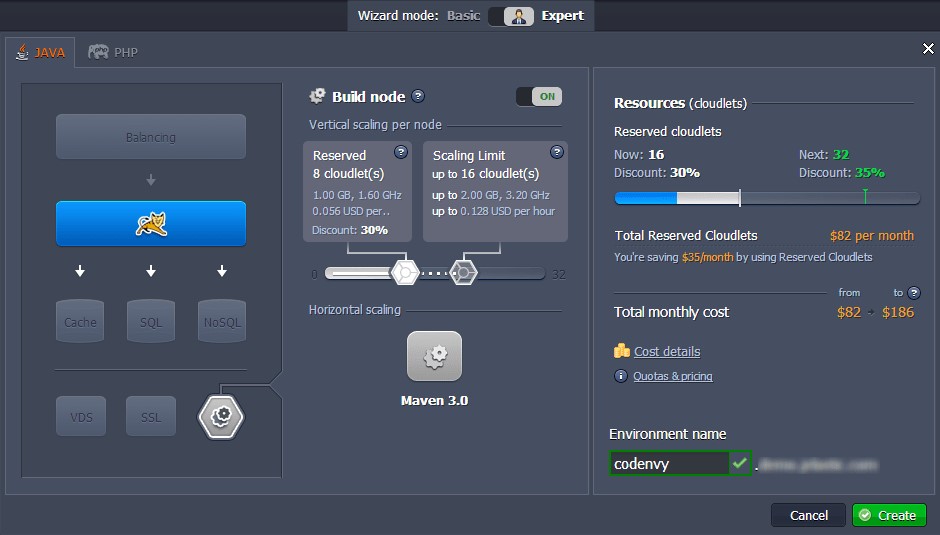
* 1. **Codenvy Dashboard**: Monitor your running container instances, resource usage, and logs within the Codenvy dashboard.

* 1. Use Codenvy's features or integrated tools like Kubernetes to dynamically scale instances up or down as needed.

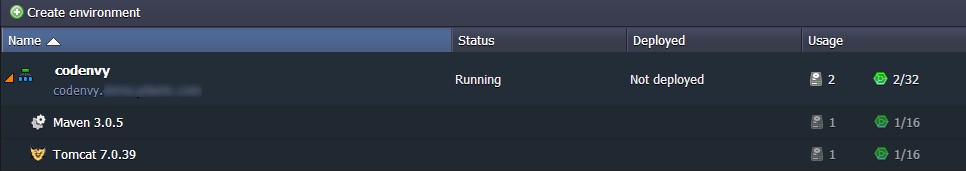
## Create environment:-

1. Log into your Jelastic account and click the **Create environment button**.

1. Select **Tomcat 7** (or any other Java application server) and specify the cloudlet consumption limits for it. Then pick Maven build node and set up the resource limits for this node. Next, type your environment name and click Create.



1. In a few minutes your environment with all the chosen nodes will be created.



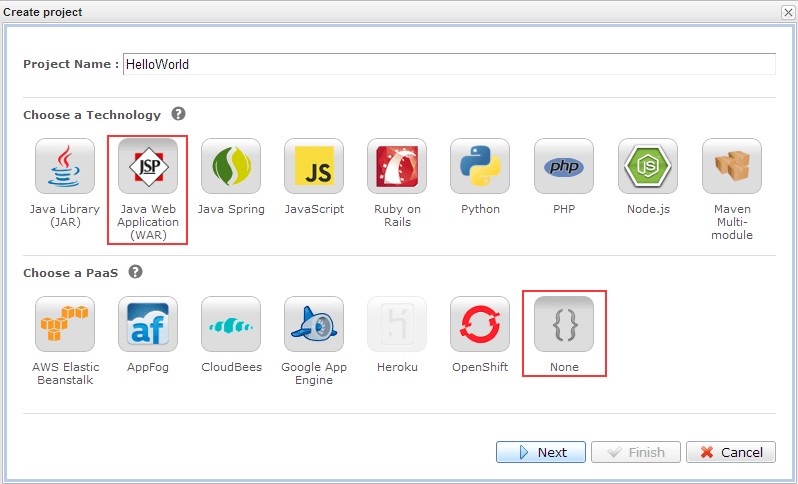
## Create Codenvy project:-

1. Log into your Codenvy workspace (or sign up and create one if you haven't done so yet).

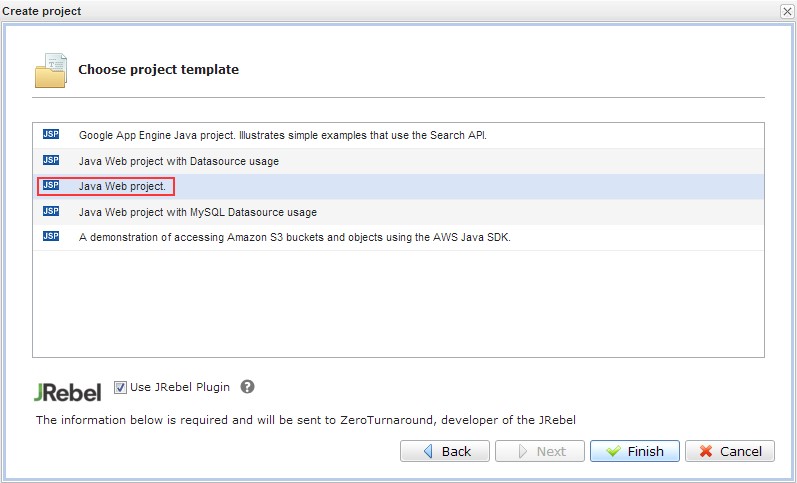
1. Click **Project > New > Create Project**... to start your own project or **Project > New > Import from GitHub...** to import an existing one from GitHub.



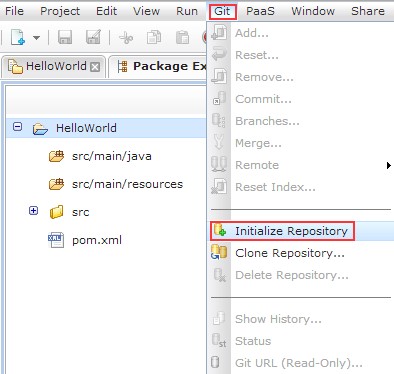
1. Select the type of your new project (Java Web Application in our case) and pick None in the available PaaS list.



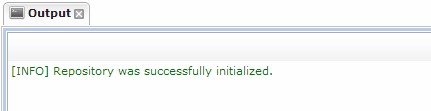
1. Select a suitable template for your Java project (e.g. Java Web Project) and click Finish.



1. When your project is successfully created (in this tutorial we use a simple HelloWorld app), initialize your local Git repository.



You'll see the output info, which confirms that your repository was successfully initialized.



**Result:**- Successfullt demonstrate provisioning and scaling of a website in codenvy.