1. Objective

The objective of this task is to understand the concept of virtualization by installing VirtualBox and creating a Virtual Machine (VM) with a Linux operating system. The VM will then be used to host and deploy a React application, providing hands on experience with setting up and managing virtual environments for software development and deployment.

2. Background

Theory/Concepts:

- **Virtualization:** Virtualization allows multiple operating systems to run on a single physical machine by creating virtual versions of resources like servers.
- **Virtual Machine (VM):** A Virtual Machine (VM) is a software emulation of a computer, running an OS and applications in isolation.
- **Hypervisor:** A hypervisor, also known as a virtual machine monitor (VMM), is software that creates and runs virtual machines. There are two types of hypervisors:
    - **Type 1 (Bare-Metal) Hypervisor:** Runs directly on the host's hardware to control the hardware and manage guest operating systems (e.g., VMware ESXi, Microsoft Hyper-V, Proxmox).
    - **Type 2 (Hosted) Hypervisor:** Runs on a conventional operating system just as other computer programs do (e.g., Oracle VirtualBox, VMware Workstation).

  Oracle VM VirtualBox is a type 2 hypervisor that manages VMs which we will be using in this lab.

Context: This experiment will use Oracle VirtualBox, which is a Type 2 hypervisor. VirtualBox requires an existing operating system to be installed and can run alongside other applications on the host system. In this experiment the Host OS is the windows and guest OS is ubuntu.

3. Tools and Services

Software/Tools:

- Oracle VirtualBox
- ISO image of the ubuntu operating system to be installed.
- React application.

4. Experiment Setup

Step-by-Step Configuration:

1. **Download and Install VirtualBox:**
    - Download Oracle VirtualBox from the official website.
    - Follow the installation instructions for your operating system.
2. **Create a New Virtual Machine:**
    - Open VirtualBox and click on the "New" button.
    - Enter a name for your VM and select the type and version of the operating system you will install.

- Allocate memory (RAM) for the VM. A minimum of 2GB is recommended for most operating systems.
- Create a virtual hard disk. Choose the size and type of storage (dynamically allocated or fixed size).

3. **Configure the Virtual Machine:**
   - Go to the settings of the VM and configure the system settings, including the number of processors and the amount of video memory.
   - Attach the ISO image of the operating system to the VM by going to the "Storage" section and adding the ISO file to the optical drive.
4. **Service Deployment:**
   - Start the VM and follow the installation prompts of the operating system.
5. **Security Settings:**
   - Configure network settings, such as NAT or Bridged Adapter, depending on your requirements.
   - Set up user accounts and passwords during the OS installation.
6. Run the react application.

5. Execution

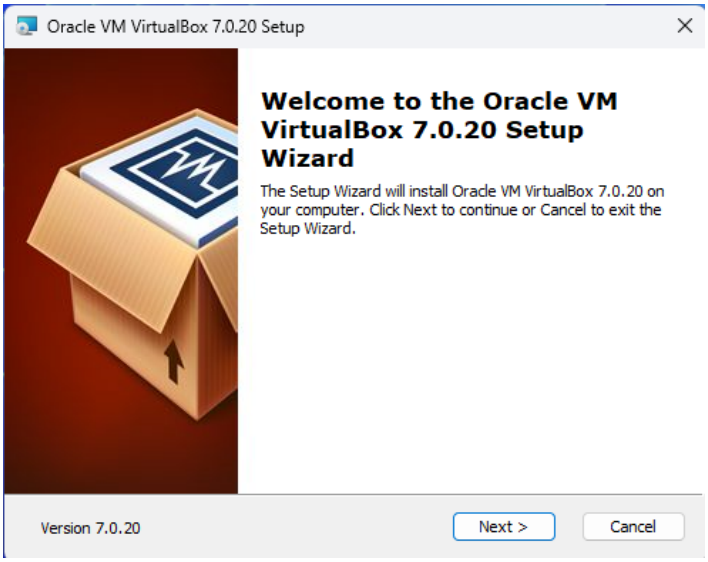
Tasks Performed:

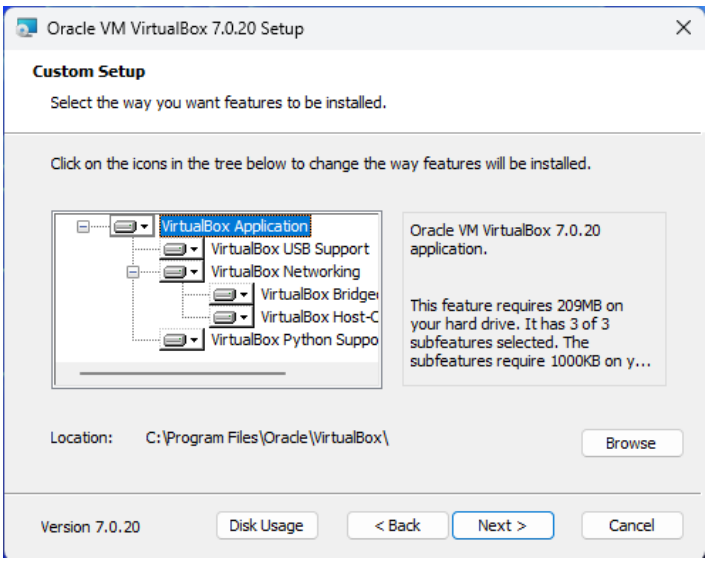
1. Install Oracle Virtual Box.
2. Create new VM.
3. Launch the VM and begin the OS installation.
4. Configure system settings and install necessary software within the VM.
5. Run react application with the backend.
6. Access react application:

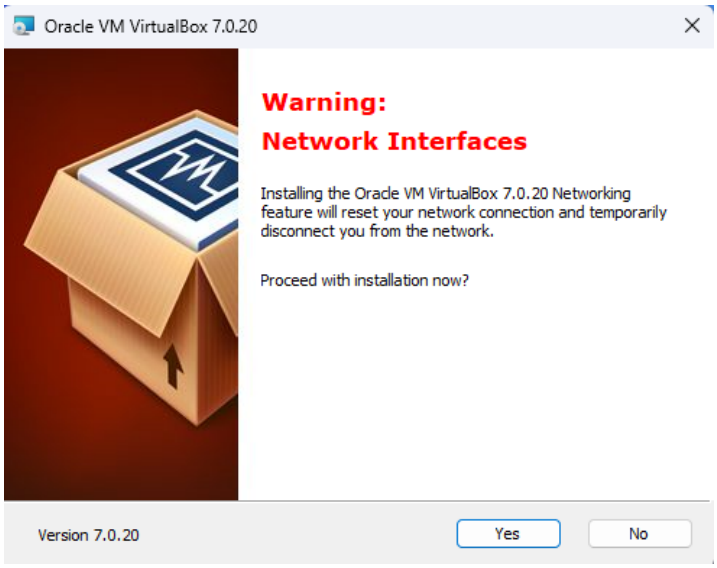
6. Observations

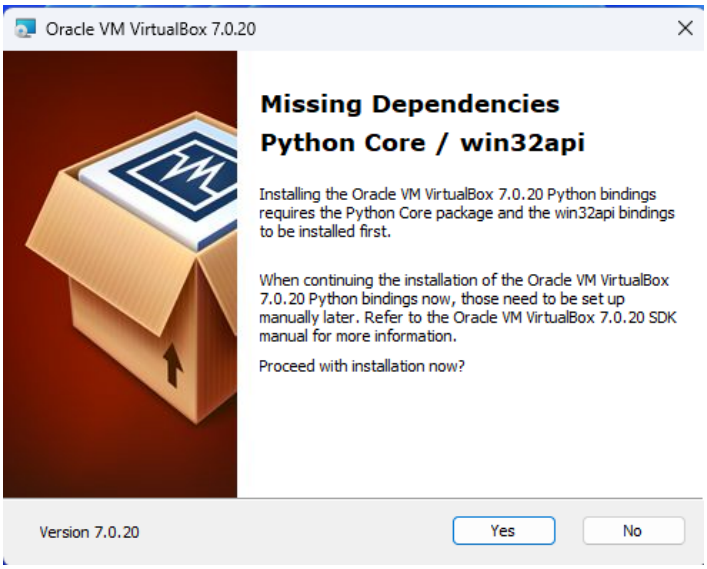
Data Collected:

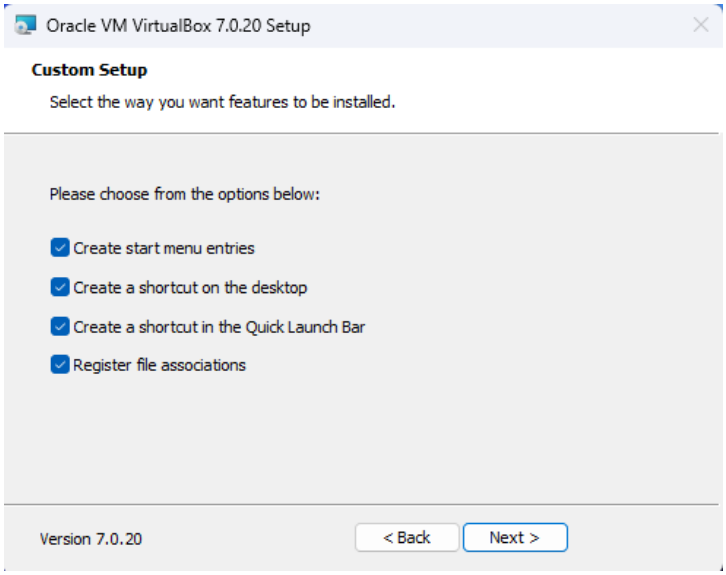
1. Installation of the Oracle Virtual Box:
   Start the installer and follow the instruction provided.

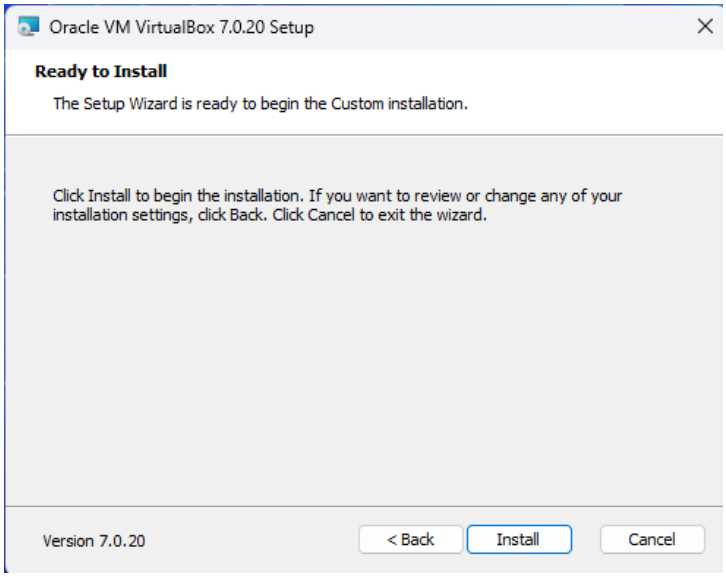






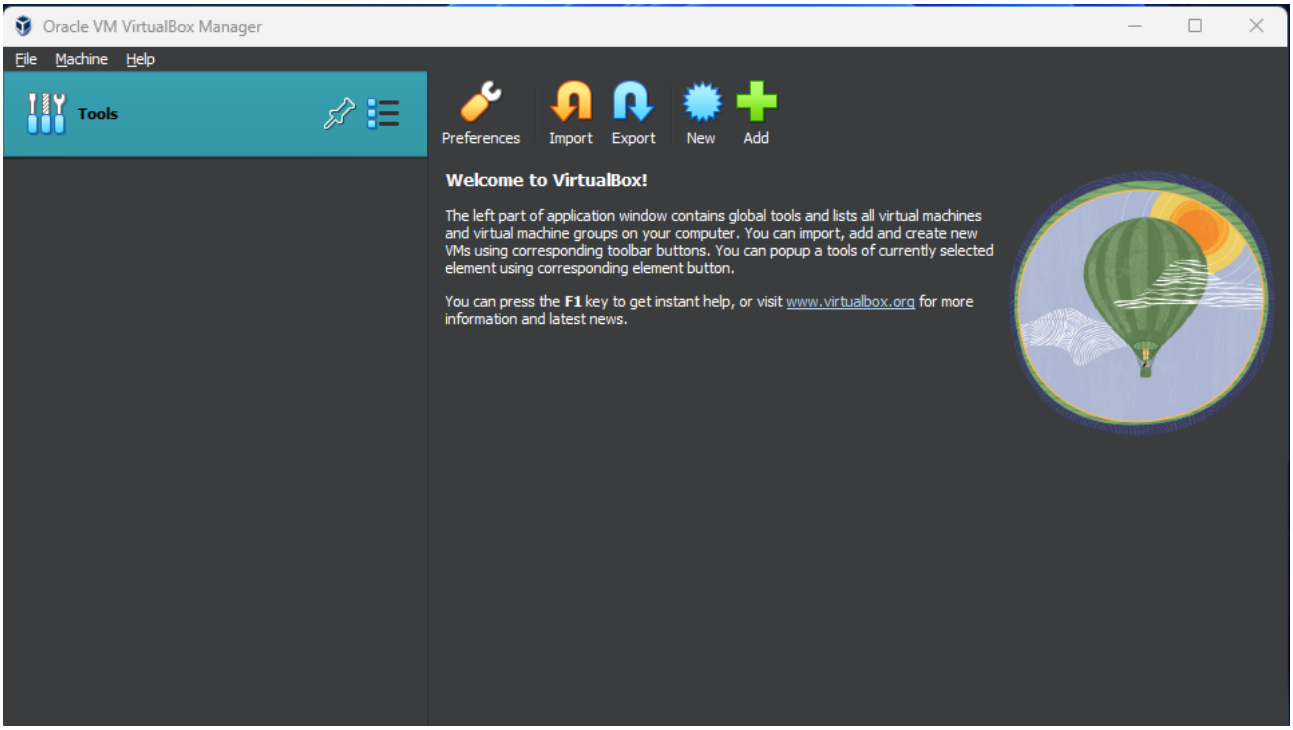






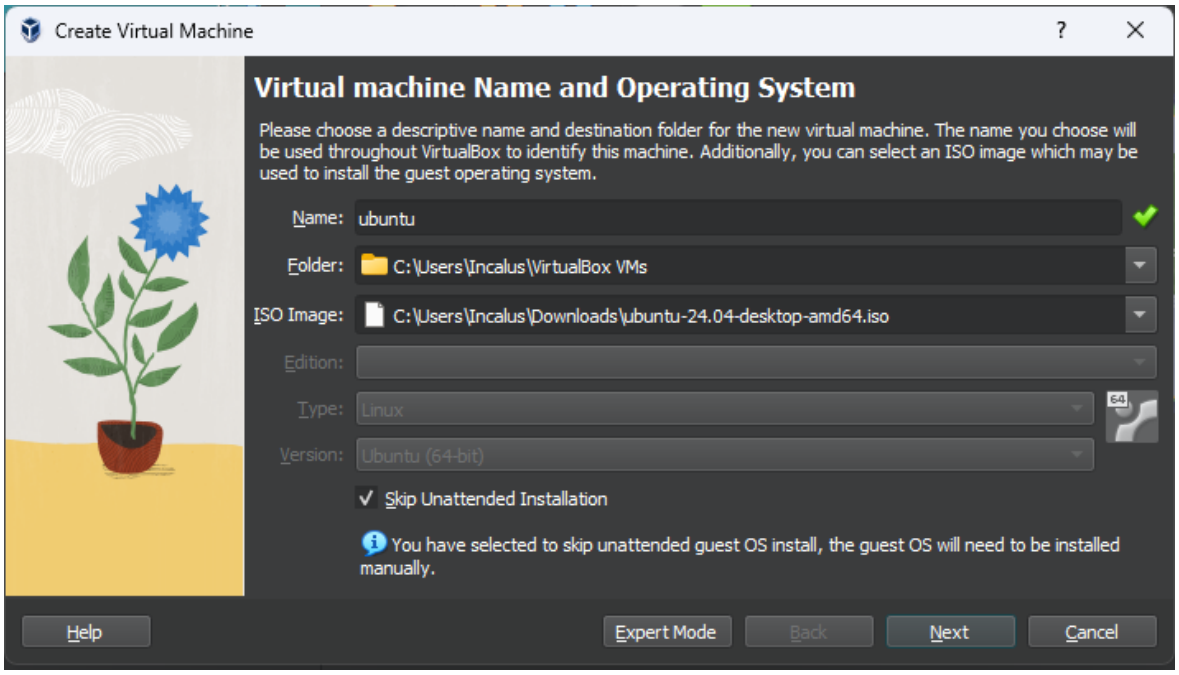
Here the installation of virtual box finished.

The UI for the virtual box is as follows:

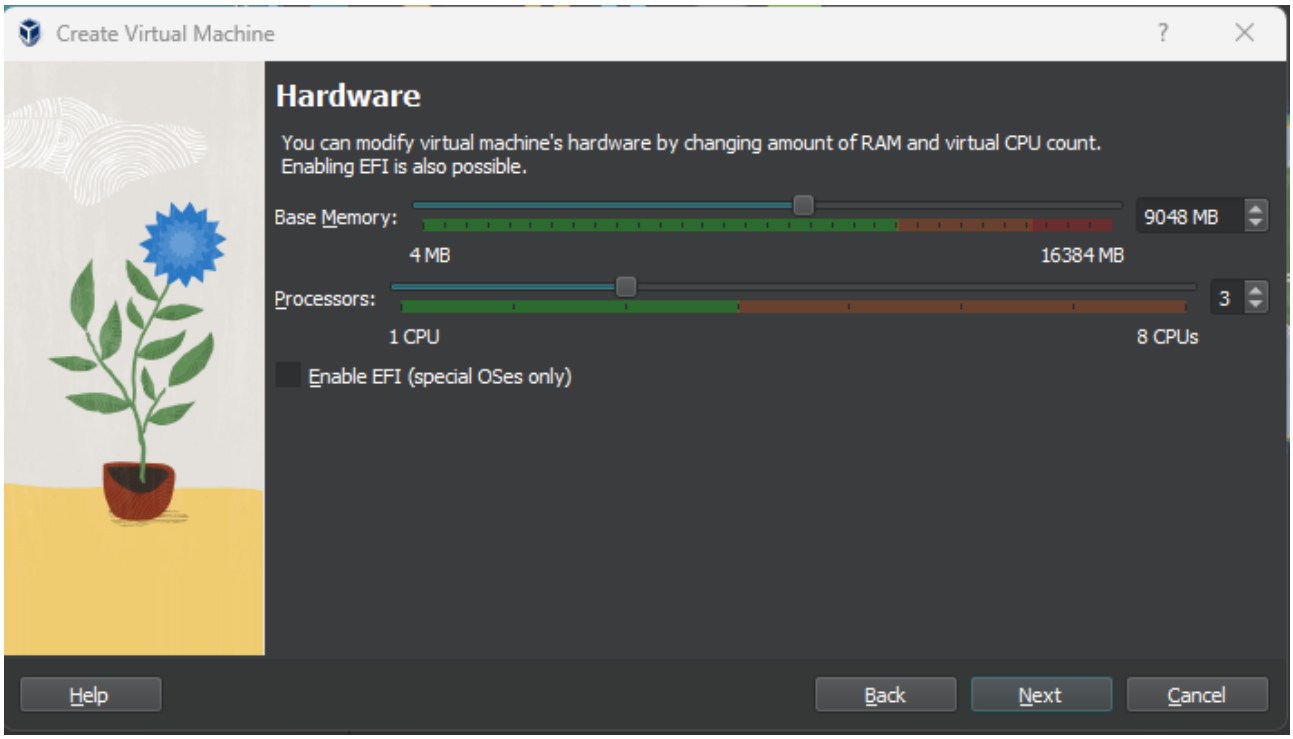


2. Create a virtual machine:

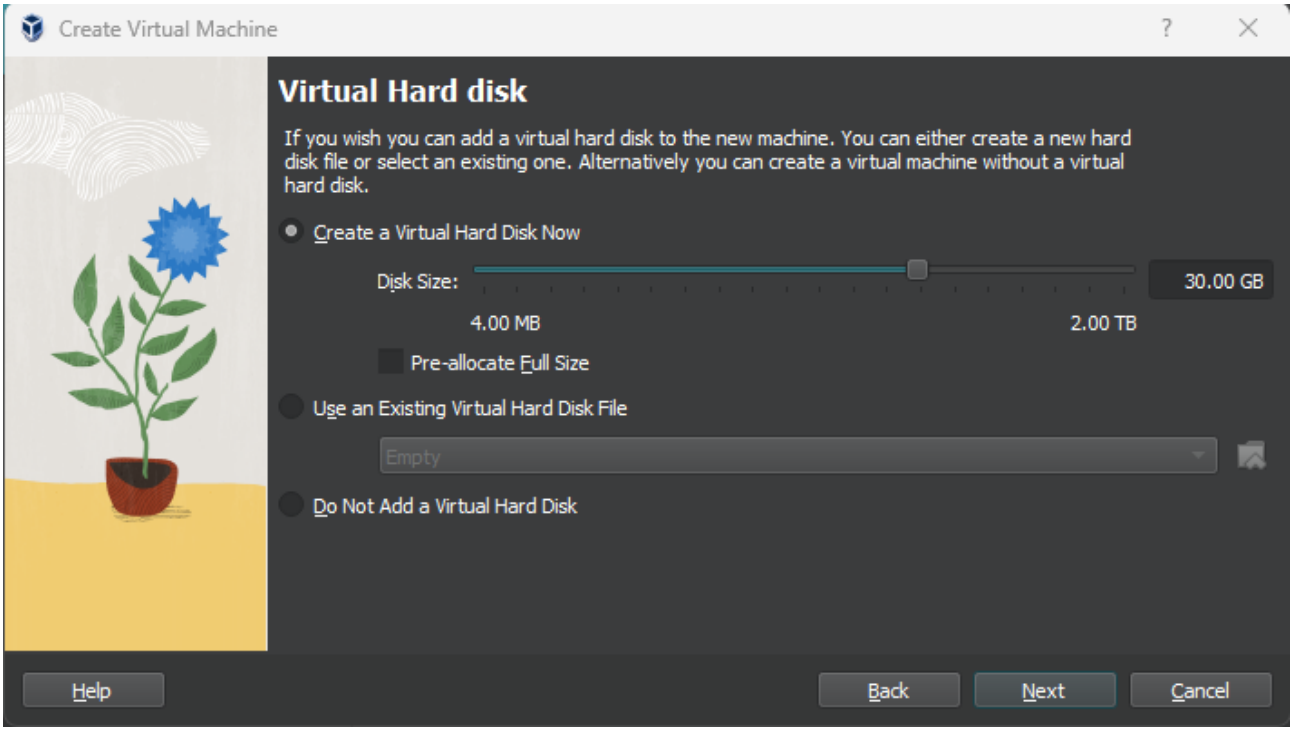
Click on new from the menu. And provide the necessary information required such as Name and location for the ISO file.



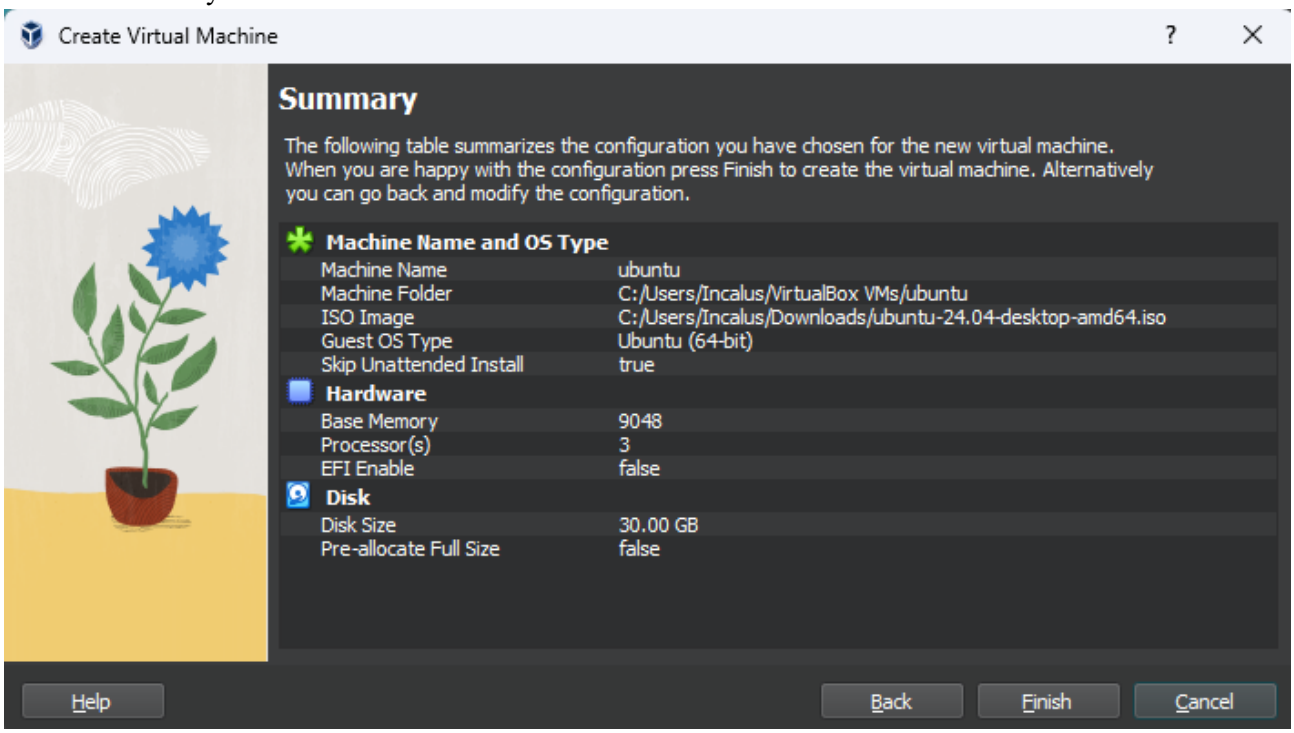
Provide how much RAM is requried (here 9GB) and the number of Proccessors (here 3) for the VM.



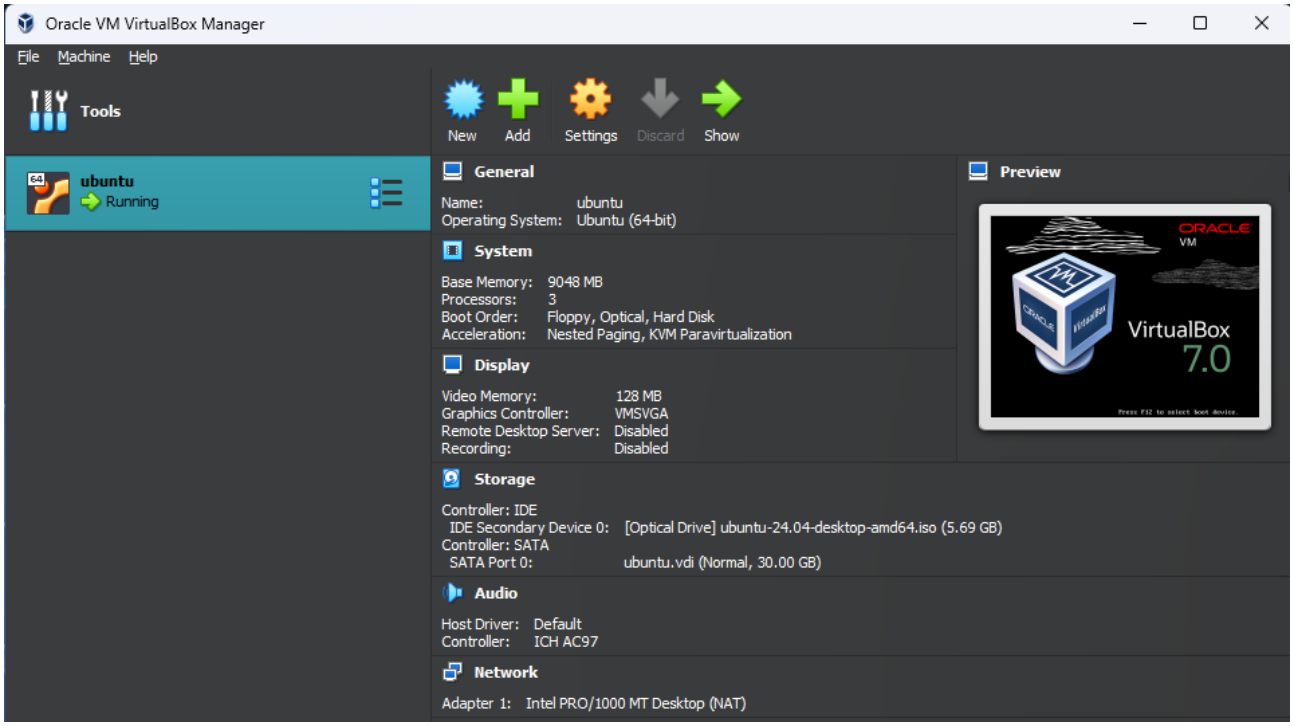
Provide the Disk space required for the VM (here 30 GB).



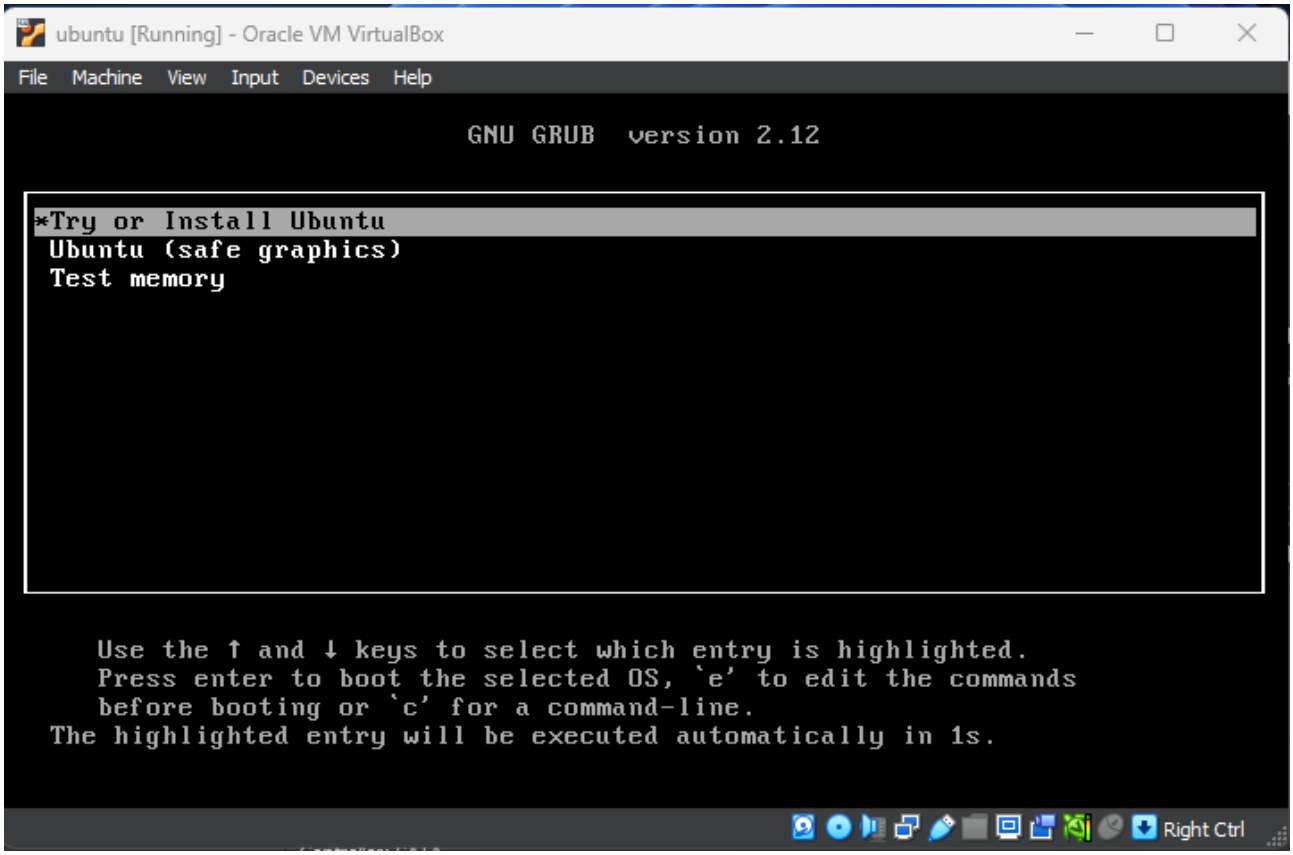
Here the summary for the VM created.

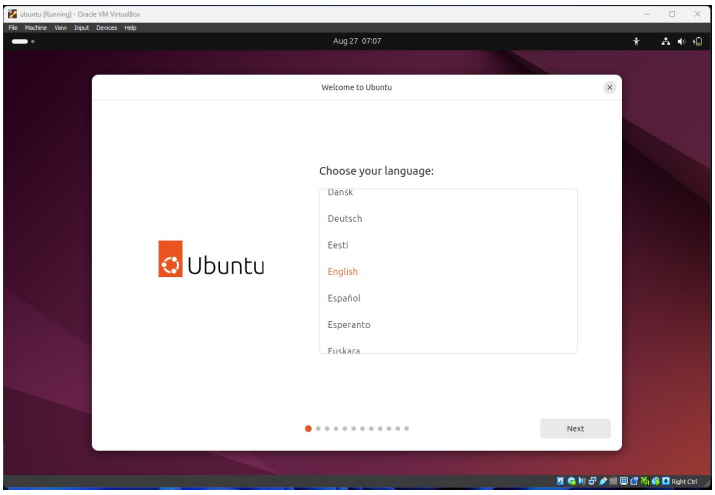


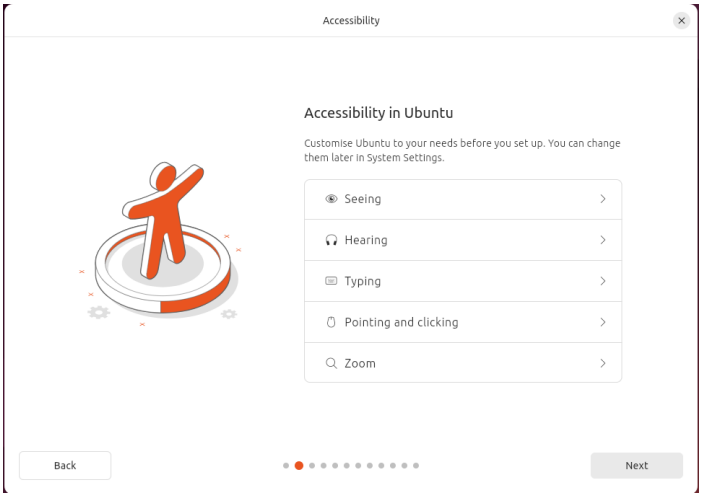
Then start the VM.

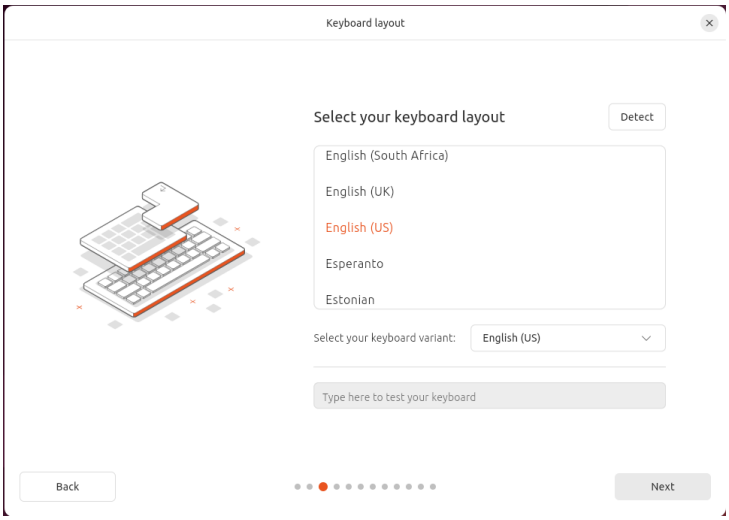


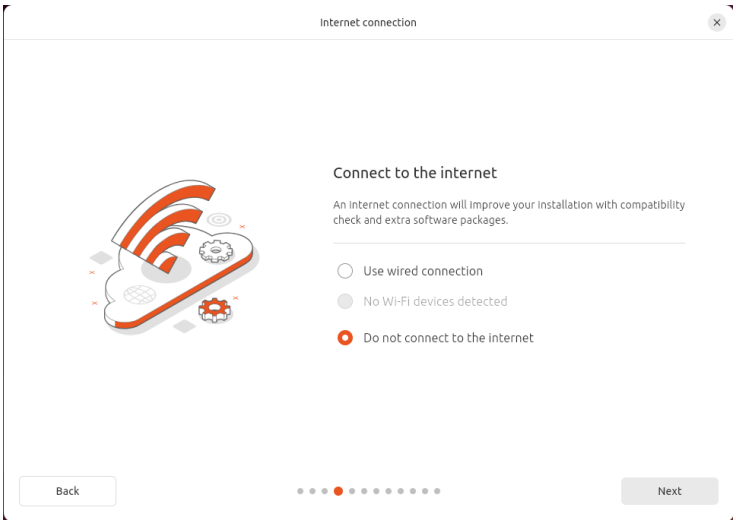
3. Install the ubuntu 24.04 on the VM.
   Start the VM and it will promt for the “Try or Install Ubuntu”. Select this option and follow the proccess for the installing OS on the VM.

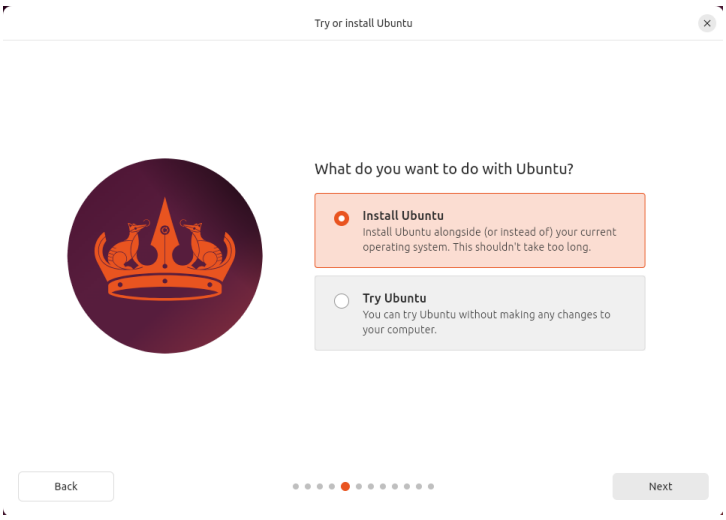


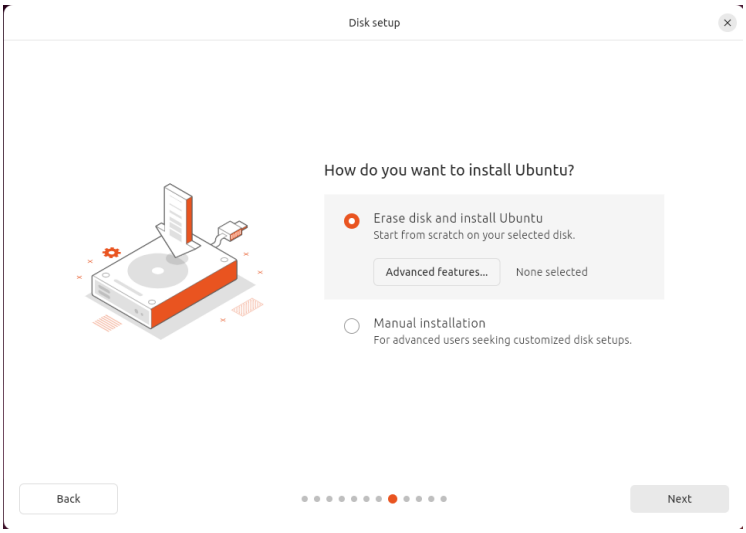




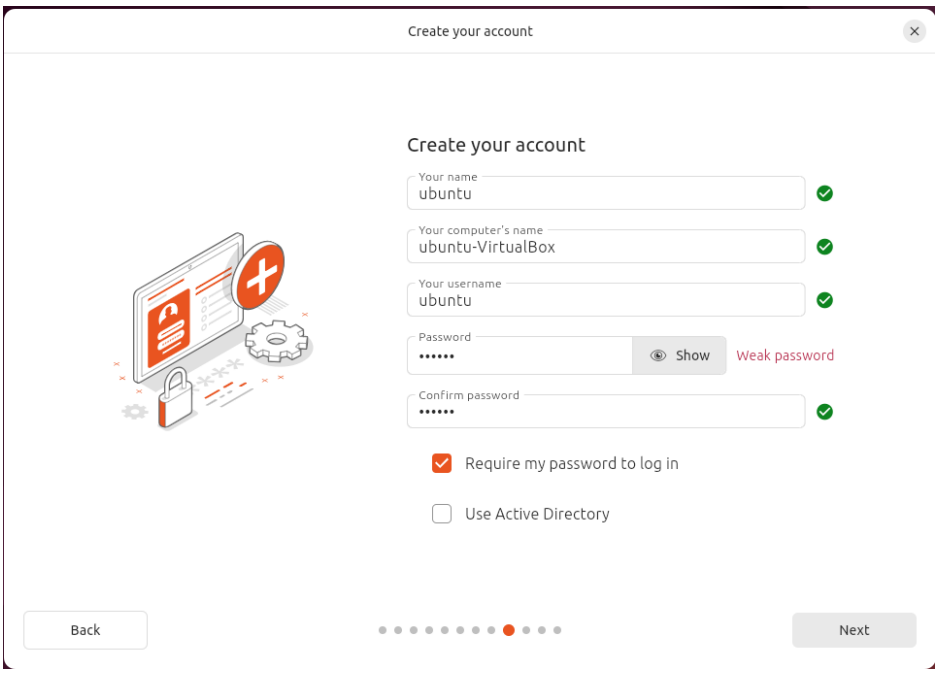




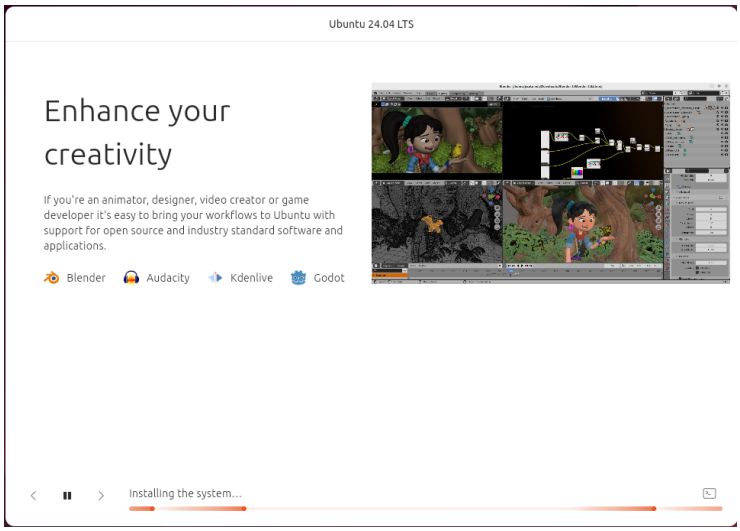


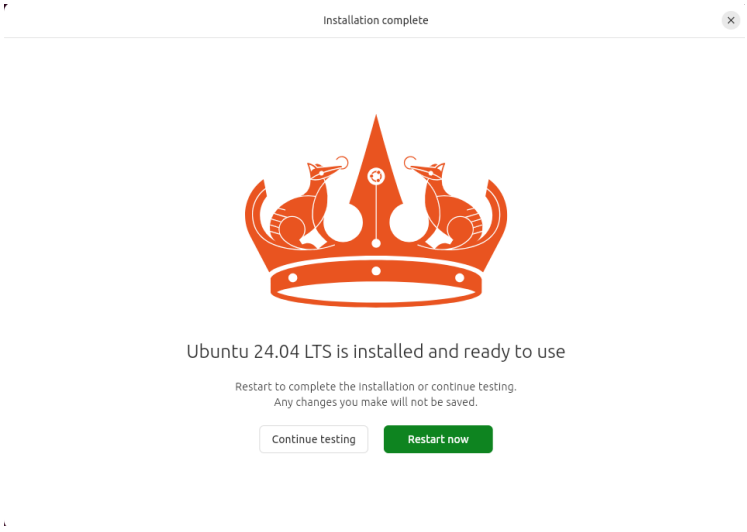


Give the information required for the name and password for the ubuntu login.

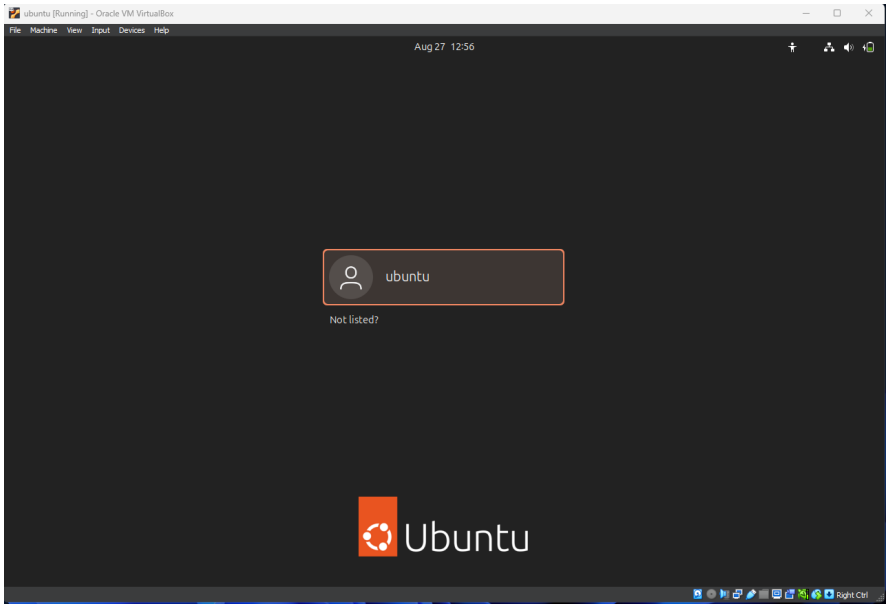


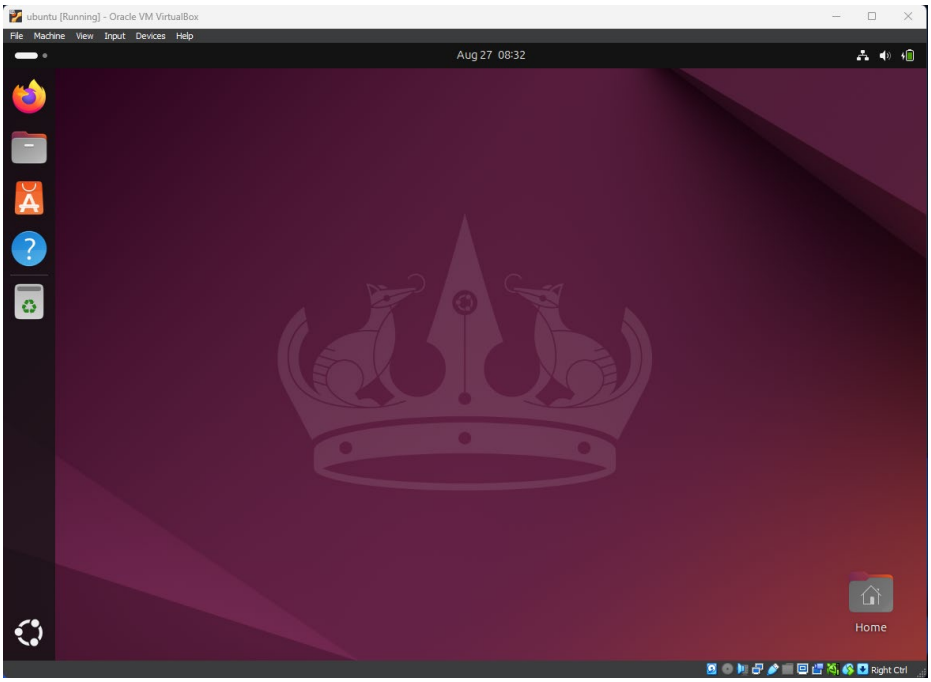
Now it install Ubuntu OS and after finishing installation it will prompt for restart. Just click restart now.





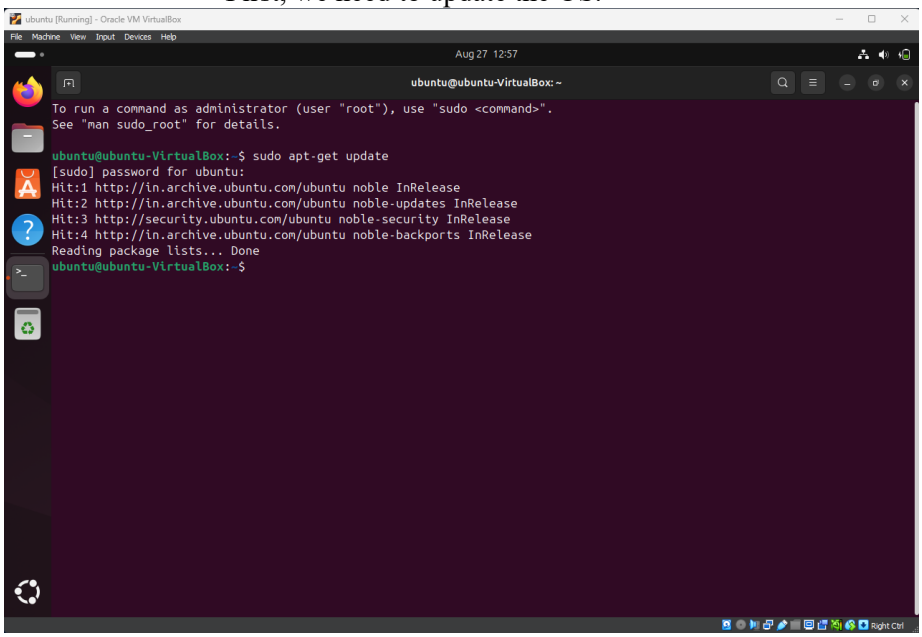
After restarting the login screen for the ubuntu will appere. Just provide the password you entered while installing the OS. And we will be on the Desktop of the ubuntu.

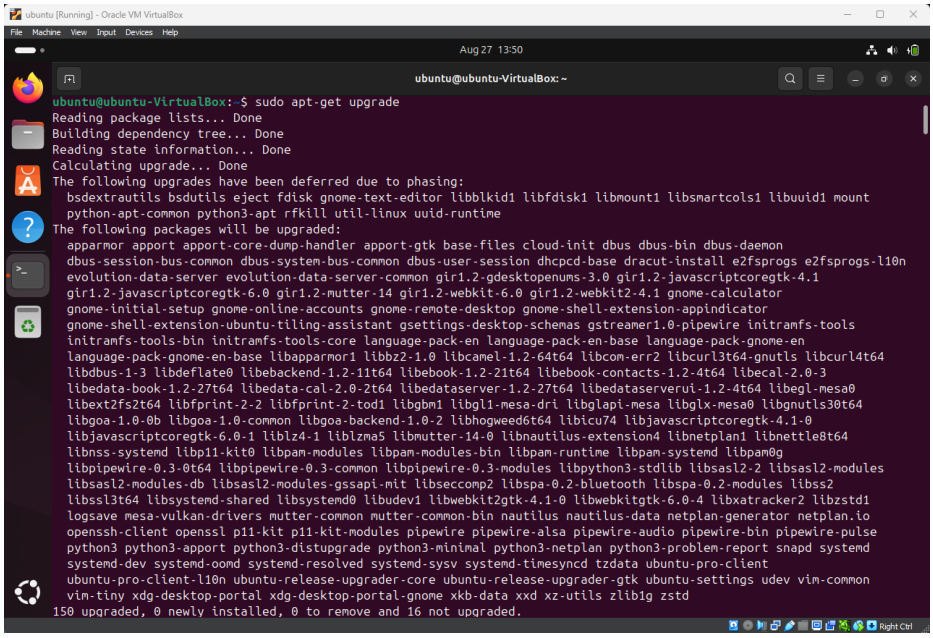


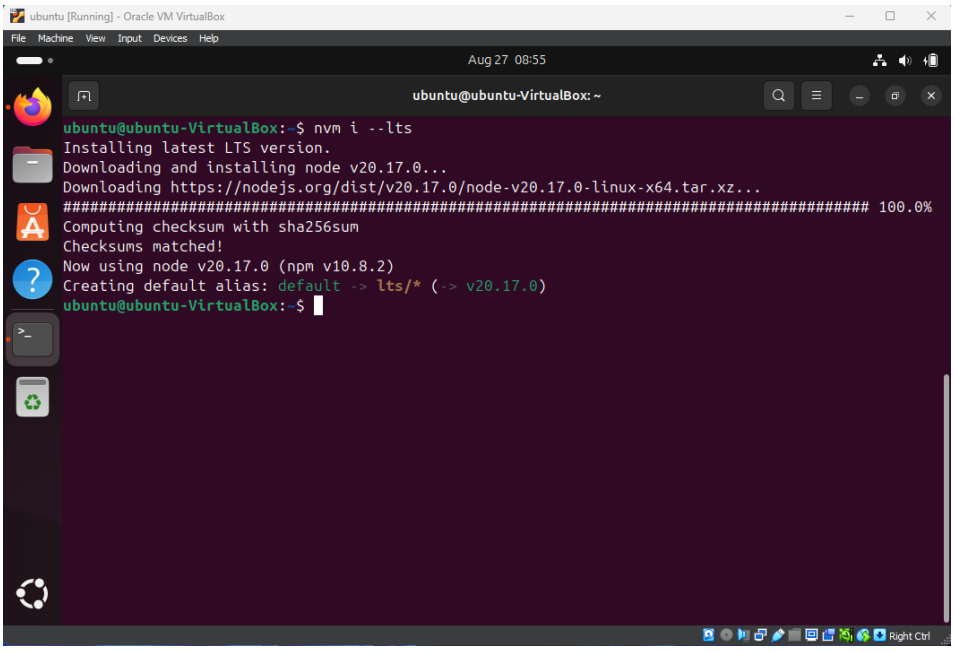


4. Installing prerequisites (node, docker)

- First, we need to update the OS.







- Install NodeJS

- Install Docker:

```
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... 61%

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 16 not upgraded.
Need to get 127 MB of archives.
After this operation, 461 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.20-1 [30.5 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu noble/main amd64 liberror-perl all 0.17029-2 [25.6 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 git-man all 1:2.43.0-1ubuntu7.1 [1,100 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 git amd64 1:2.43.0-1ubuntu7.1 [3,679 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 MB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.1.2-1~ubuntu.24.04~noble [14.6 MB]
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.1.2-1~ubuntu.24.04~noble [25.2 MB]
Get:11 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.1.2-1~ubuntu.24.04~noble [9,318 kB]
Get:12 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.1-1~ubuntu.24.04~noble [12.5 MB]
Fetched 127 MB in 15s (8,322 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 147935 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../01-containerd.io_1.7.20-1_amd64.deb ...
Unpacking containerd.io (1.7.20-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../02-docker-buildx-plugin_0.16.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../03-docker-ce-cli_5%3a27.1.2-1~ubuntu.24.04~noble_amd64.deb ...
```

5. Run the react application:

Starting the backend server:

```
ubuntu@ubuntu-VirtualBox:~/Desktop/Cloud-Lab-Exps/exp1/server$ npm run nodemon

> server@1.0.0 nodemon
> nodemon main.js

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node main.js`
Server is running on port 8000
Mongoose connected to mongodb://root:password@localhost:27017/task-manager?authSource=admin
MongoDB connected
```

Starting the Frontend react application:

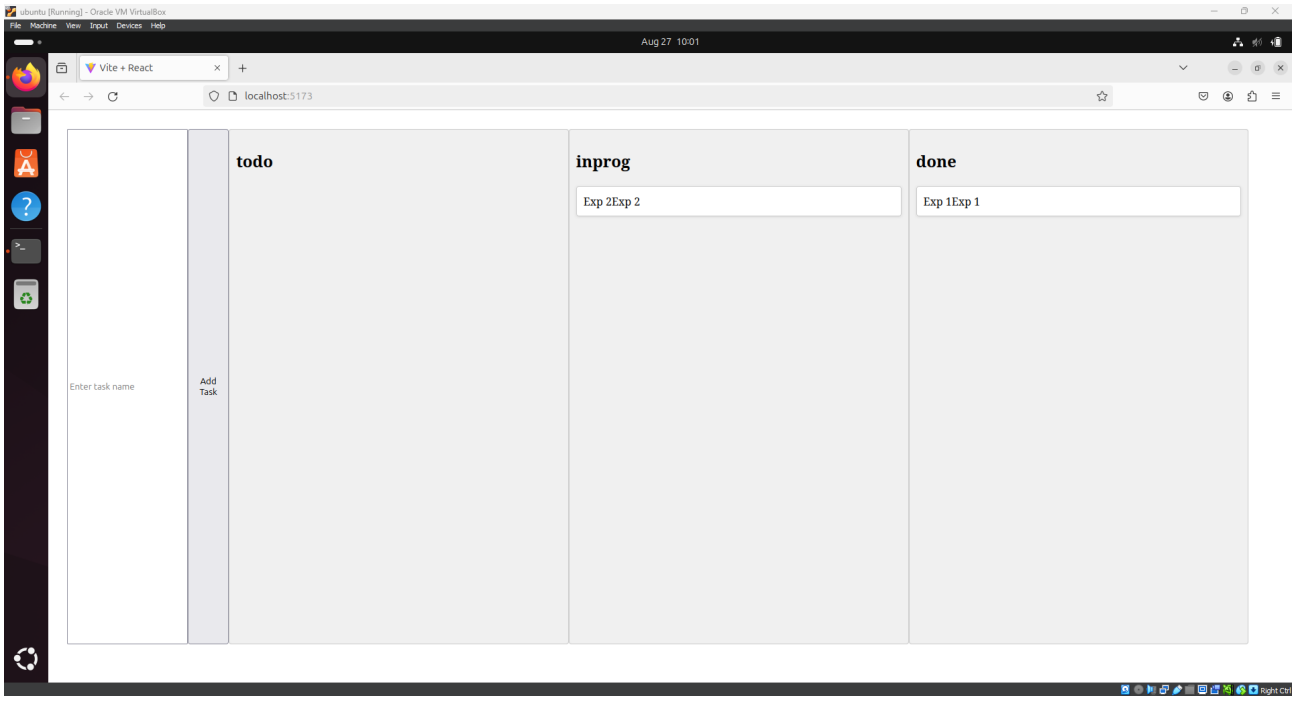
```
ubuntu@ubuntu-VirtualBox:~/Desktop/Cloud-Lab-Exps/exp1/client$ npm run dev

> client@0.0.0 dev
> vite


  VITE v5.4.2  ready in 206 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

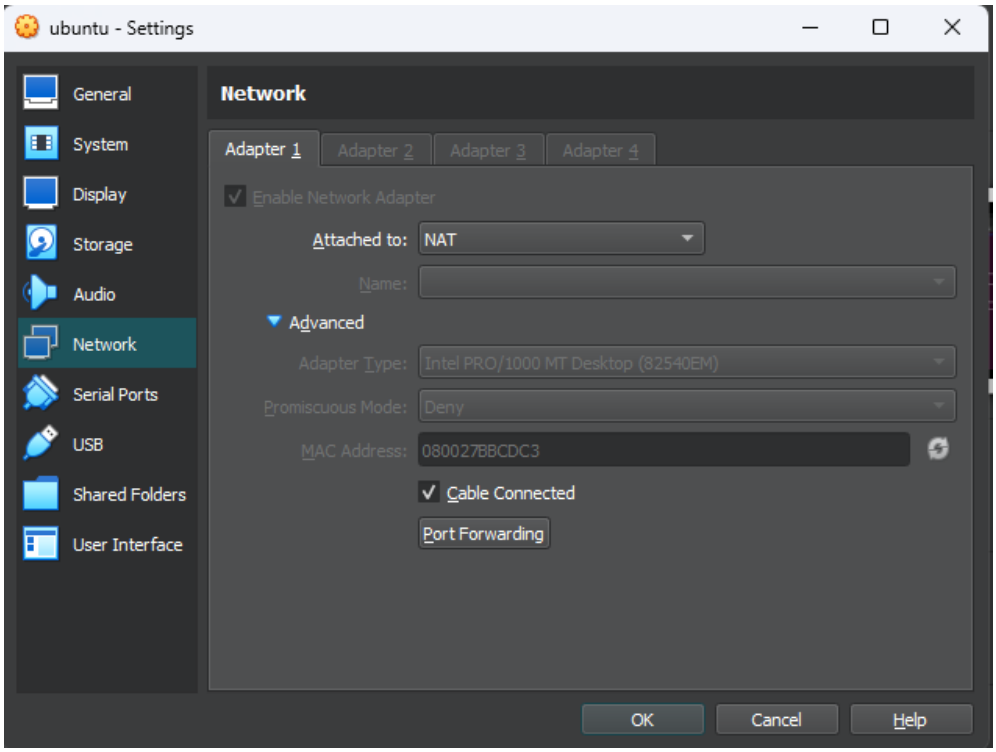
The react application UI.

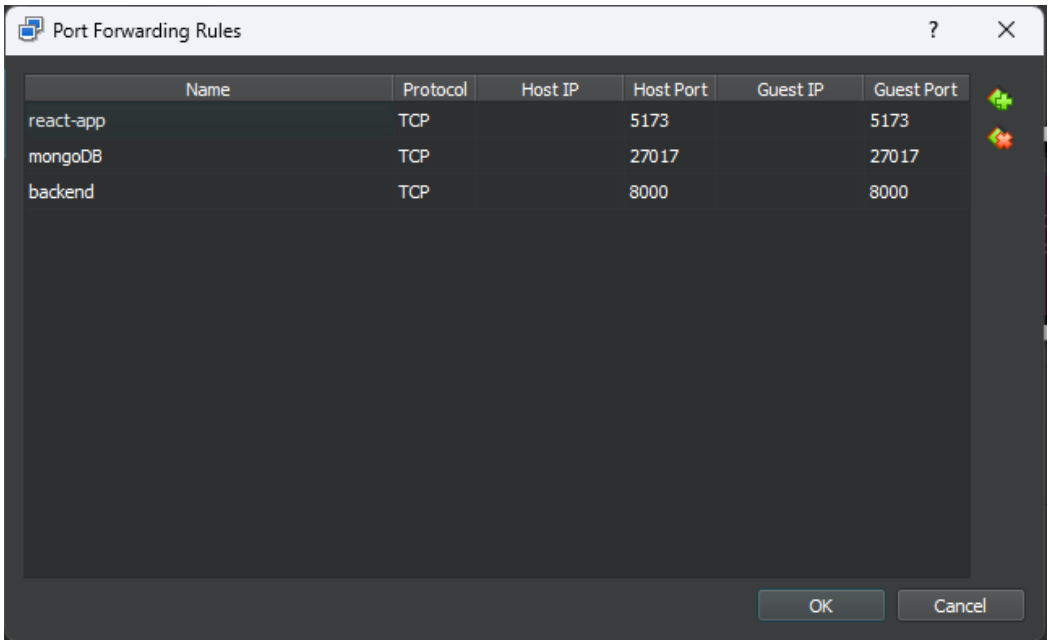


6. Access react application:

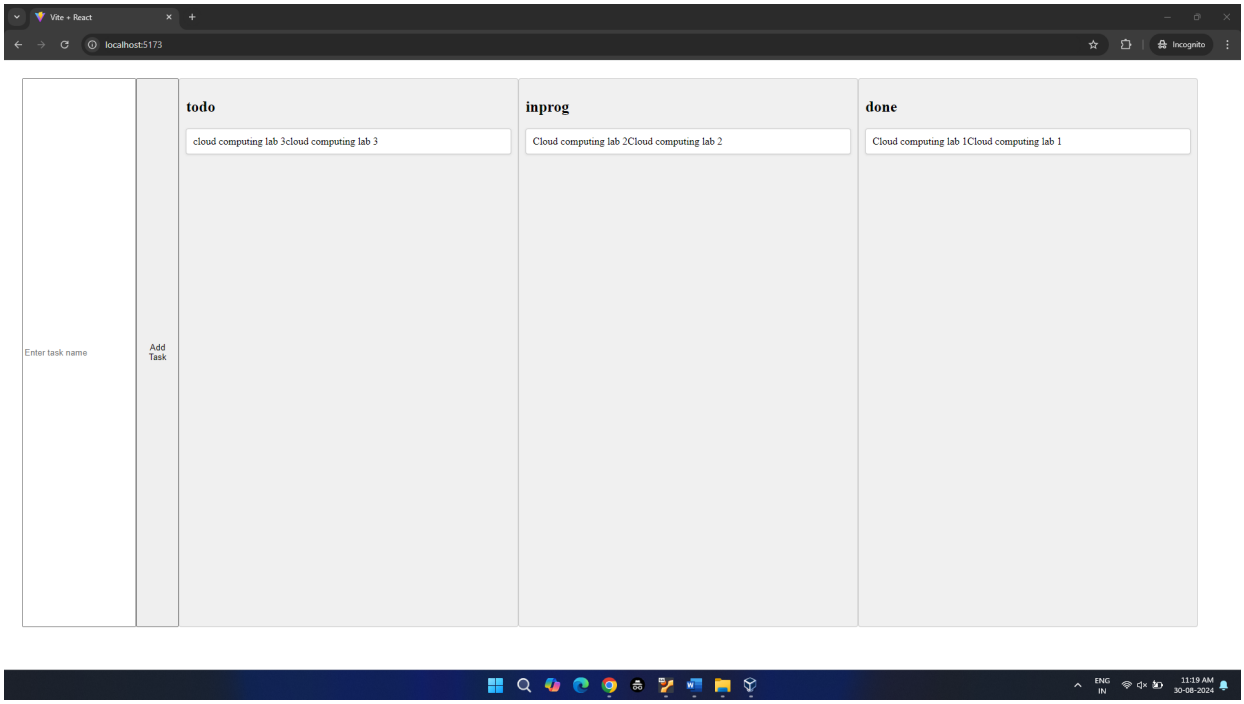
   For accessing the application from the host, we need to add port forwarding in the VM configuration. As out application is running on 5173, we need to forward port 5173 along with 8000 and 27017 as our backend and mongo data base is running on those ports.

   The option for port forwarding is in Setting => Network => Advanced => Port Forwarding





After adding port forwarding rules from given screenshot, we will be able to access the react application from the host machine.



7. Results

Virtualization was understood by installing Virtual Box and running a React Application.