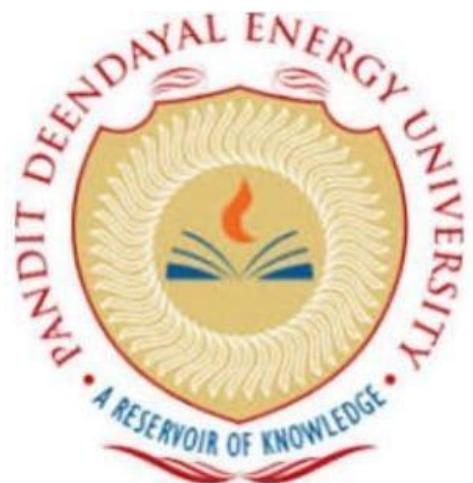


# Cloud Computing Laboratory (20CP322P)

Lab Report Submitted to

Pandit Deendayal Energy University, Gandhinagar

For



*Bachelor of Technology*

*in*

*Computer Science & Engineering Department*

*Submitted by **Dalsaniya Jemmy 21BCP319***

*Semester: VI Division: 5, Group: 10*

*Course Faculty*

*Dr. ABHINAV SHARMA*

*PANDIT DEENDAYAL ENERGY UNIVERSITY, GANDHINAGAR -*

*382007, India May-2024*

### List of Practicals

<b>Exp No.</b>	<b>Experiment Title</b>	<b>Date</b>	<b>Signature</b>
1	Study of Various Virtualization Software		
2	To Install and configure VMware WorkstationPro for creating the Virtual Machines		
3	To Install and configure VMware WorkstationPro or Oracle Virtual Box for creating Virtual Machines, cloning, and deleting VMs.		
4	Working with DOCKER		
5	Working with DOCKER		
6	Working with DOCKER		
7	Working with DOCKER		
8	Working with DOCKER - COMPOSE		

**Name : DALSANIYA JEMMY .V**  
**Roll no: 21BCP319**  
**Div : 5 G-10**

## **Lab 1 Assignment: Study of Various Virtualization Software**

**Aim :** Study of Various Virtualization Software such as VMware, QEMU, KVM, XEN Hypervisor and Oracle Virtual Box

Perform the following:

Compare their Virtualization type and methodologies.

Identify architectural difference.

List out various advantages and disadvantages.

Applications of virtualization software.

### **Team Members**

**1**

21BCP320  
Himanshi

**2**

21BCP325  
Shaily

**3**

21BCP321  
Ronit

**4**

21BCP319  
Jemmy

### **Introduction**



# What is Virtualization?

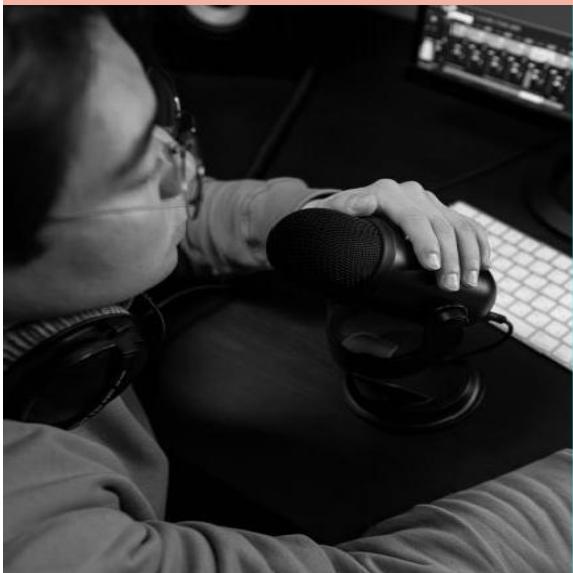
A SINGLE PHYSICAL COMPUTER WITH MULTIPLE VIRTUAL MACHINES RUNNING DIFFERENT OPERATING SYSTEMS.



Virtualization software creates virtual machines (VMs) on a single physical machine.

Each VM acts as a separate computer with its own operating system and applications.

This allows efficient resource utilization, flexibility, and isolation.



## Types of Virtualization and Methodology

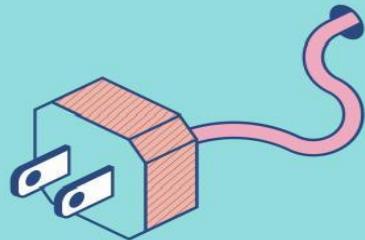
**TYPE 1 HYPERVISORS (BARE-METAL):** RUN DIRECTLY ON HARDWARE, PROVIDING HIGH PERFORMANCE AND ISOLATION. (EXAMPLES: VMWARE, XEN)

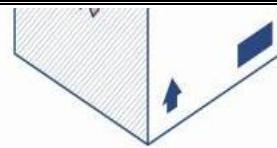
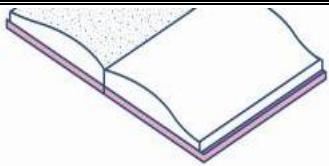
**TYPE 2 HYPERVISORS (HOSTED):** RUN ON TOP OF AN EXISTING OPERATING SYSTEM, OFFERING EASIER INSTALLATION BUT SLIGHTLY LOWER PERFORMANCE. (EXAMPLES: QEMU, ORACLE VIRTUAL BOX, VMWARE WORKSTATION)

**HYBRID OPTIONS:** SOME SOFTWARE, LIKE VMWARE WORKSTATION AND HYPERVERSOR, CAN FUNCTION AS BOTH TYPE 1 AND TYPE 2.

## Compare their Virtualization type and Methodologies

- VMware uses full virtualization with a Type-1 hypervisor and Type-2 hypervisor (Workstation).
- QEMU employs full-system emulation and supports both Type-1 and Type-2 hypervisor modes.
- KVM (Kernel-based Virtual Machine) is a Type-1 hypervisor that utilizes hardware virtualization extensions.
- XEN Hypervisor is a Type-1 hypervisor, providing paravirtualization for improved performance.
- Oracle VirtualBox is a Type-2 hypervisor, running on the host OS.



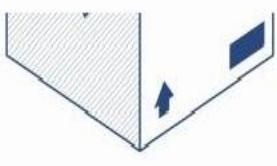
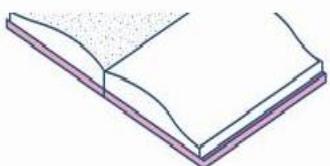


## VMWARE

VMware is a leading name in virtualization technology, allowing you to run multiple separate operating systems and applications on a single physical machine

## QEMU

QEMU is a machine emulator that can run operating systems and programs for one machine like CPU, memory and emulated devices on a different machine.

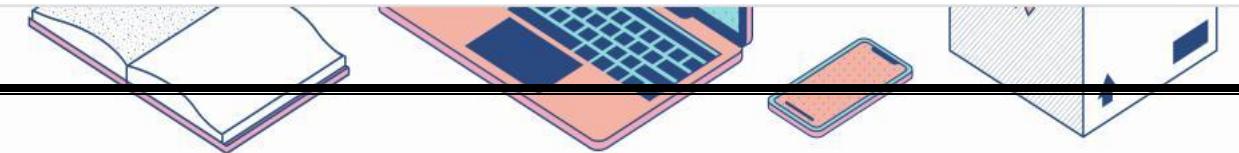


## KVM (KERNEL BASED VIRTUAL MACHINE)

KVM is a Type 1 hypervisor, meaning it runs directly on the hardware, independent of any host operating system. This offers several advantages over Type 2 hypervisors, such as higher performance and better isolation between VMs.

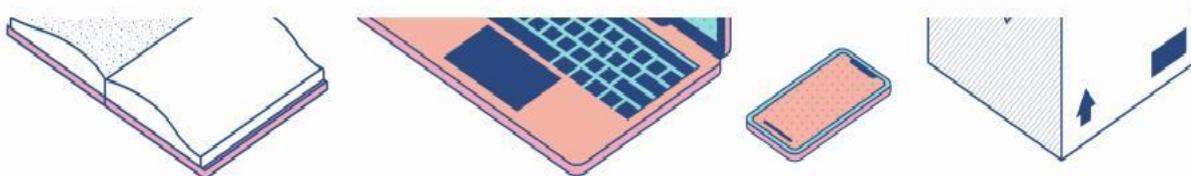
## XEN

Xen is a hypervisor managing multiple virtual machines on one computer without an extra operating system. It supports various OS like Linux and Windows. It creates "Domains" to manage virtual devices and tasks for other VMs.



## ORACLE VIRTUAL BOX

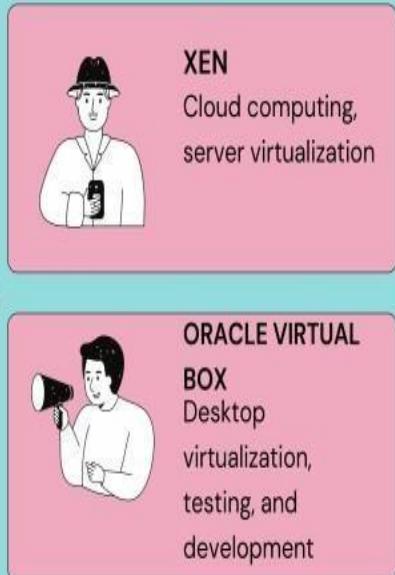
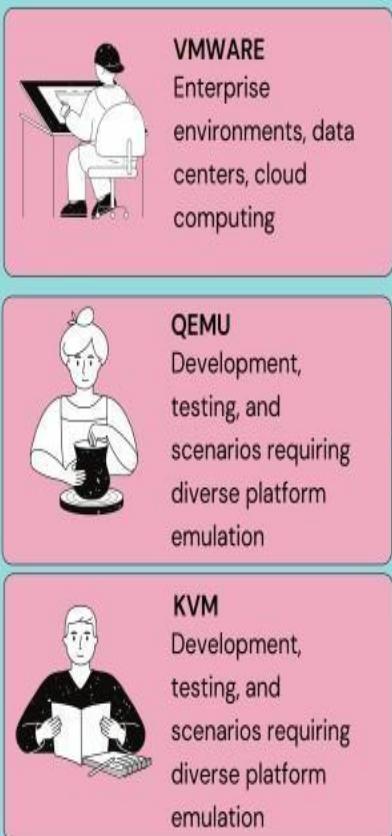
**It enables users to run multiple operating systems on a single physical machine. It provides a platform for creating and managing virtual machines (VMs) in which users can install and run different operating systems concurrently on their host system.**



## Applications of Virtualization

- **Server consolidation:** Combine multiple servers onto a single physical machine, saving space and resources.
- **Software development and testing:** Create isolated environments for development and testing, preventing conflicts with the production environment.
- **Cloud computing:** Build and manage cloud infrastructure using virtual machines.
- **Desktop virtualization:** Provide virtual desktops to users for improved security and manageability.
- **Education and training:** Provide students with access to different operating systems and applications in a controlled environment.
- **Personal use:** Run multiple operating systems or applications on a single computer for personal projects or entertainment.

## Applications



# THANK YOU



**Name : DALSANIYA JEMMY .V  
Roll no: 21BCP319  
Div : 5 G-10**

**Lab 2 Assignment: To Install and configure VMware Workstation Pro for creating the Virtual Machines**

**Aim :** Exploring VMWare Workstation Pro to create the virtual Machines

**Web 2:**

**1.Create Two or more Virtual Machine and assign resources.**

**Steps:**

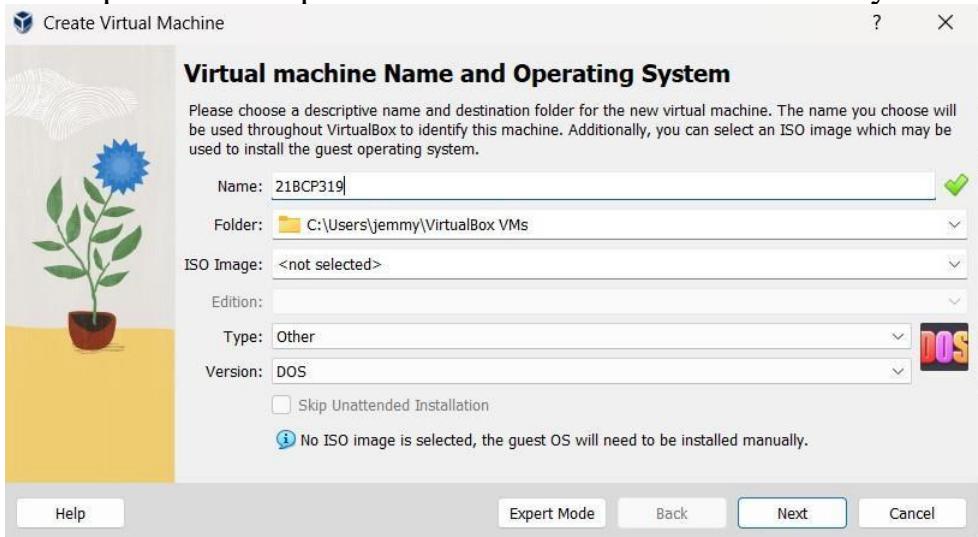
**● Step 1: Install VirtualBox**

Download and install Oracle VM VirtualBox from the official website: VirtualBox Downloads



## ● Step 2: Create Virtual Machines

- Open VirtualBox and click on "New" to create a new virtual machine.
- Provide a name for your virtual machine, choose the type and version of the operating system, and click "Next."
- Allocate memory (RAM) to the virtual machine. Choose an amount that suits your needs and click "Next."
- Create a virtual hard disk for the virtual machine. You can use the recommended size or adjust it according to your requirements. Click "Create."
- Choose the hard disk file type. The default VDI is usually fine. Click "Next."
- Decide whether to allocate the disk space dynamically or fixed. Dynamic allocation is recommended for saving space. Click "Next" and then "Create."
- Repeat these steps for each additional virtual machine you want to create.



**Create Virtual Machine**

**Hardware**

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 2048 MB (4 MB to 16384 MB)

Processors: 1 (1 CPU to 12 CPUs)

Enable EFI (special OSes only)

Help Back Next Cancel

Controller: SATA

**Create Virtual Machine**

**Virtual Hard disk**

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

Create a Virtual Hard Disk Now

Disk Size: 50.00 GB (4.00 MB to 2.00 TB)

Pre-allocate Full Size

Use an Existing Virtual Hard Disk File  
21BCP319 Puppy1.vdi (Normal, 2.00 GB)

Do Not Add a Virtual Hard Disk

Help Back Next Cancel

**Create Virtual Machine**

**Summary**

The following table summarizes the configuration you have chosen for the new virtual machine. When you are happy with the configuration press Finish to create the virtual machine. Alternatively you can go back and modify the configuration.

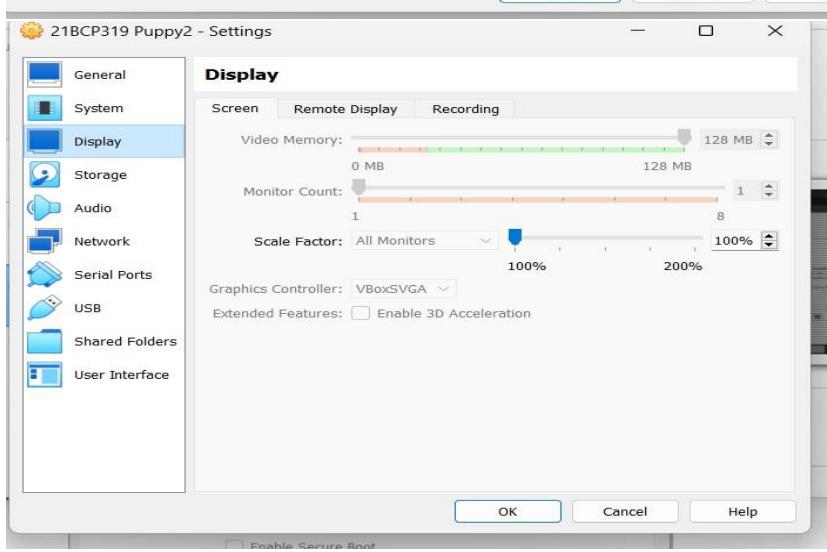
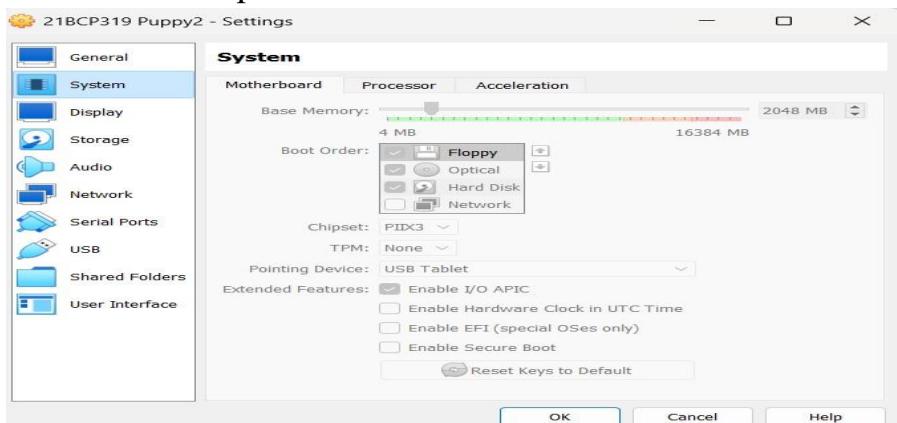
<b>Machine Name and OS Type</b>	
Machine Name	21BCP319
Machine Folder	C:/Users/jemmy/VirtualBox VMs/21BCP319
ISO Image	
Guest OS Type	Windows 10 (64-bit)
<b>Hardware</b>	
Base Memory	2048
Processor(s)	1
EFI Enable	false
<b>Disk</b>	
Disk Size	50.00 GB
Pre-allocate Full Size	false

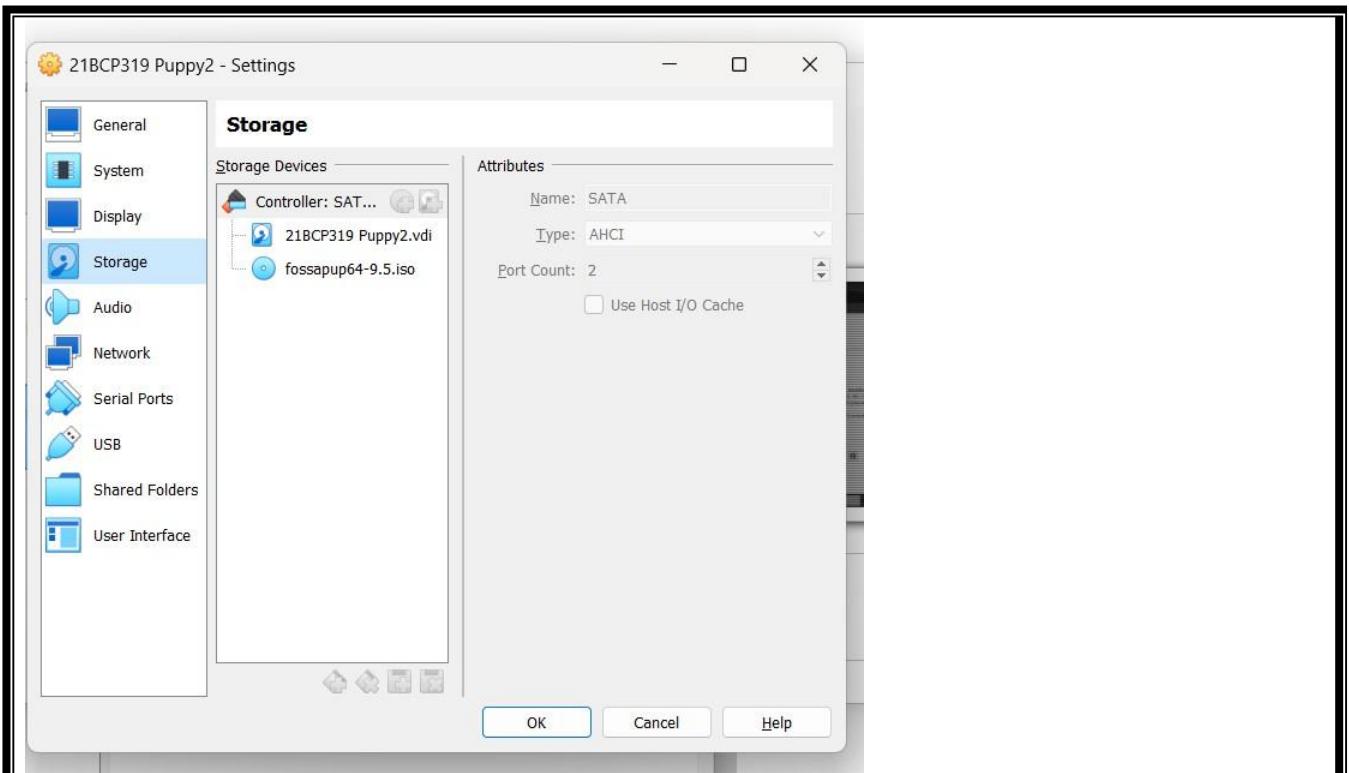
Help Back Finish Cancel

Controller: SATA

## ● Step 3: Assign Resources

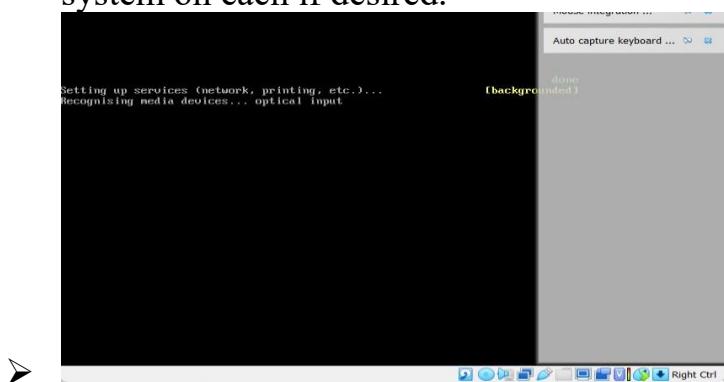
- With a virtual machine selected in VirtualBox, click on "Settings."
- In the "System" tab, adjust the number of processors and enable or disable the IO APIC based on your system's capabilities.
- In the "Display" tab, adjust the video memory and enable 3D acceleration if needed.
- In the "Storage" tab, you can attach ISO files to the virtual optical drive to install the guest operating systems later.
- In the "Network" tab, configure the network settings for your virtual machine.
- Repeat these steps for each virtual machine, adjusting the settings based on your resource requirements.





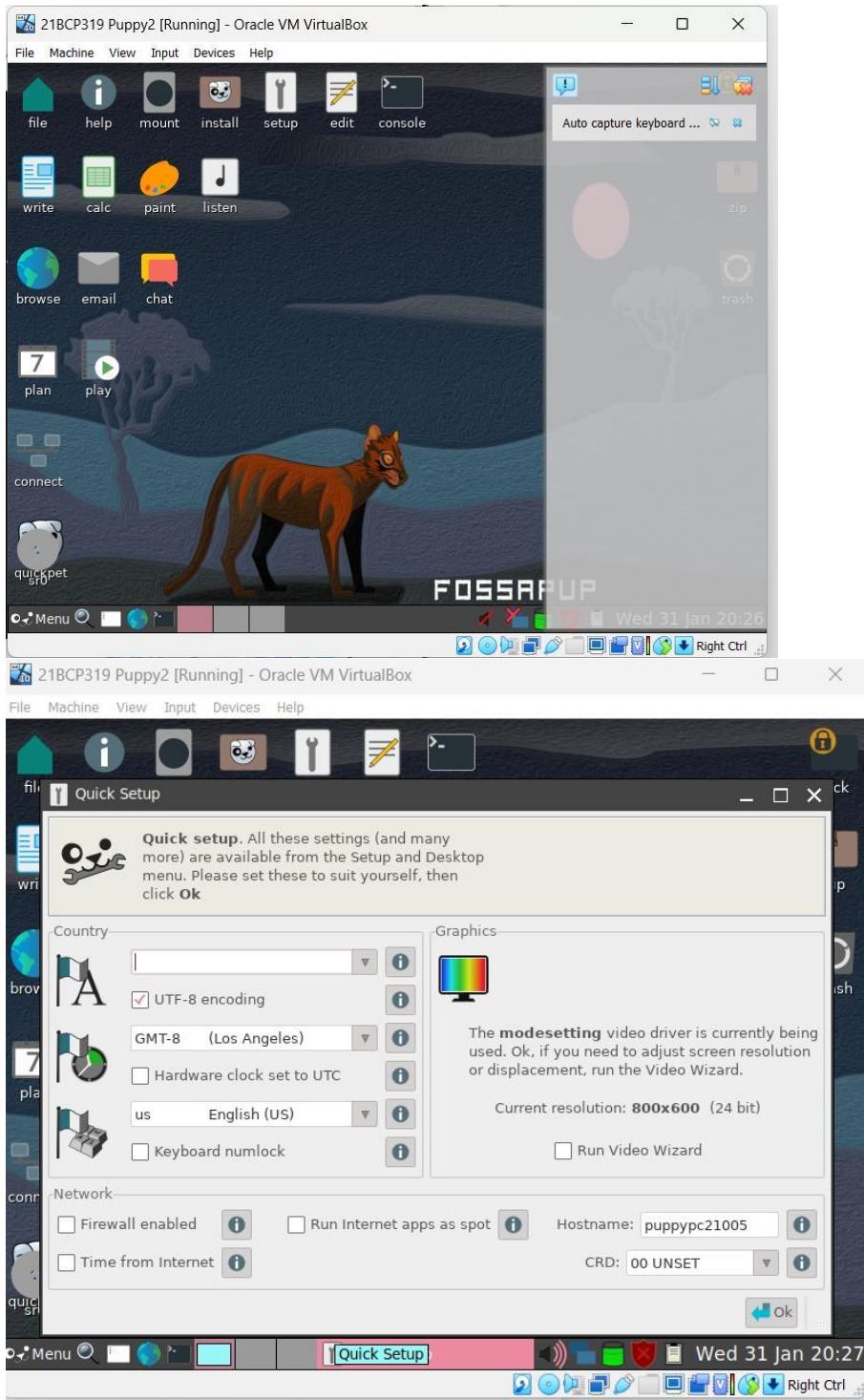
## ● Step 4: Install Guest Operating Systems

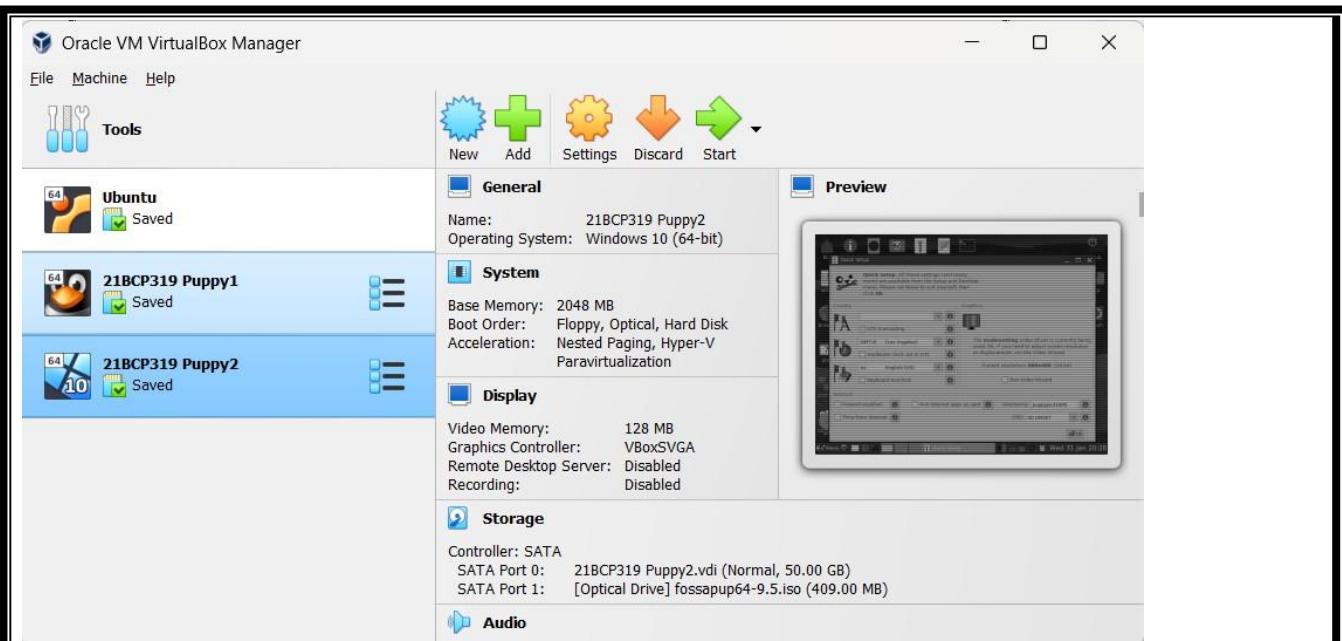
- Select a virtual machine and click "Start."
- VirtualBox will prompt you to select a startup disk. Choose the ISO file of the guest operating system you want to install.
- Follow the installation process for the selected guest operating system.
- Repeat these steps for each virtual machine, installing a different guest operating system on each if desired.



## ● Step 5: Post-Installation Configurations

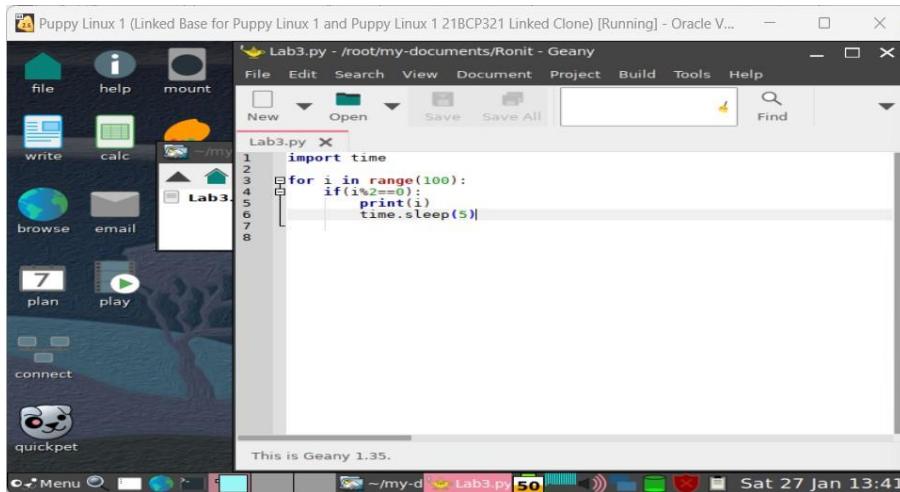
- Once the guest operating system is installed on each virtual machine, you may need to configure network settings, install guest additions, and set up any additional software or configurations as needed.





### 3. Run simple applications or programs on all the VMs.

#### 3.a) Run a simple Python script printing even numbers in VM-1 at 5 sec interval.



```
Puppy Linux 1 (Linked Base for Puppy Linux 1 and Puppy Linux 1 21BCP321 Linked Clone) [Running] - Oracle V...
System
puppypc17918
~/my-documents/Ronit
[NEW] | 1 |
root# ls
Choices  Downloads  my-applications  pkg  spot  Templates
conkyrc  ftdb  my-documents  puppy-reference  Startup  Web-Server
root# cd my-documents/Ronit/
root# python Lab3.py
0
2
4
6
8
10
12
-
Sat 27 Jan 2024 13:43:53 +08
Sat 27 Jan 13:43
```

### 3.b) Run a simple Python script printing odd numbers in VM-2 at 5 sec interval.

```
Puppy Linux [Running] - Oracle VM VirtualBox
Lab3.py - /root/my-documents/Ronit - Geany
File Edit Search View Document Project Build Tools Help
New Open Save Save All
Lab3.py X
1 import time
2
3 for i in range(100):
4     if(i%2!=0):
5         print(i)
6         time.sleep(5)
7
This is Geany 1.35.
Terminal
~/my-d... Lab3.py 51
Sat 27 Jan 13:40
```

```
Puppy Linux [Running] - Oracle VM VirtualBox
System
puppypc31700
~/my-documents/Ronit
[NEW] | 1 |
root# id
bash: id: command not found
root# ls
Choices  Downloads  my-applications  pkg  spot  Templates
conkyrc  ftdb  my-documents  puppy-reference  Startup  Web-Server
root# cd my-documents/Ronit/
root# python Lab3.py
1
3
5
7
9
11
13
15
-
Sat 27 Jan 13:42
```

## 4. View system configurations of each VM. Check whether it is same different from what you have created.

<b>General</b>	<b>Preview</b>
Name: 21BCP319 Puppy2 Operating System: Windows 10 (64-bit)	
<b>System</b>	
Base Memory: 2048 MB Boot Order: Floppy, Optical, Hard Disk Acceleration: Nested Paging, Hyper-V Paravirtualization	
<b>Display</b>	
Video Memory: 128 MB Graphics Controller: VBoxSVGA Remote Desktop Server: Disabled Recording: Disabled	
<b>Storage</b>	
Controller: SATA SATA Port 0: 21BCP319 Puppy2.vdi (Normal, 50.00 GB) SATA Port 1: [Optical Drive] fossapup64-9.5.iso (409.00 MB)	
<b>Audio</b>	
<b>General</b>	<b>Preview</b>
Name: 21BCP319 Puppy1 Operating System: Oracle Linux (64-bit)	
<b>System</b>	
Base Memory: 2048 MB Processors: 2 Boot Order: Floppy, Optical, Hard Disk Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization	
<b>Display</b>	
Video Memory: 16 MB Graphics Controller: VMSVGA Remote Desktop Server: Disabled Recording: Disabled	
<b>Storage</b>	
Controller: IDE IDE Secondary Device 0: [Optical Drive] fossapup64-9.5.iso (409.00 MB) Controller: SATA SATA Port 0: 21BCP319 Puppy1.vdi (Normal, 2.00 GB)	

## **Lab 3 Assignment: To Install and configure VMWare Workstation Pro or Oracle Virtual Box for creating Virtual Machines, cloning, and deleting VMs.**

### **Week 3:**

#### **1. Cloning of Virtual Machines: Full Cloning, Linked Cloning**

##### **❖ Full Cloning:**

Full cloning, in the context of virtualization, refers to the process of creating an exact and independent copy of a virtual machine (VM), including its entire virtual disk, configuration, and settings. When you perform a full clone, you essentially replicate the entire VM, resulting in a new instance that operates independently of the original.

- **Open VirtualBox:** Open Oracle VM VirtualBox and ensure the virtual machine you want to clone is powered off.
- **Select the VM:** Choose the virtual machine you want to clone in the VirtualBox Manager.

##### **● Clone the VM:**

Right-click on the selected VM.

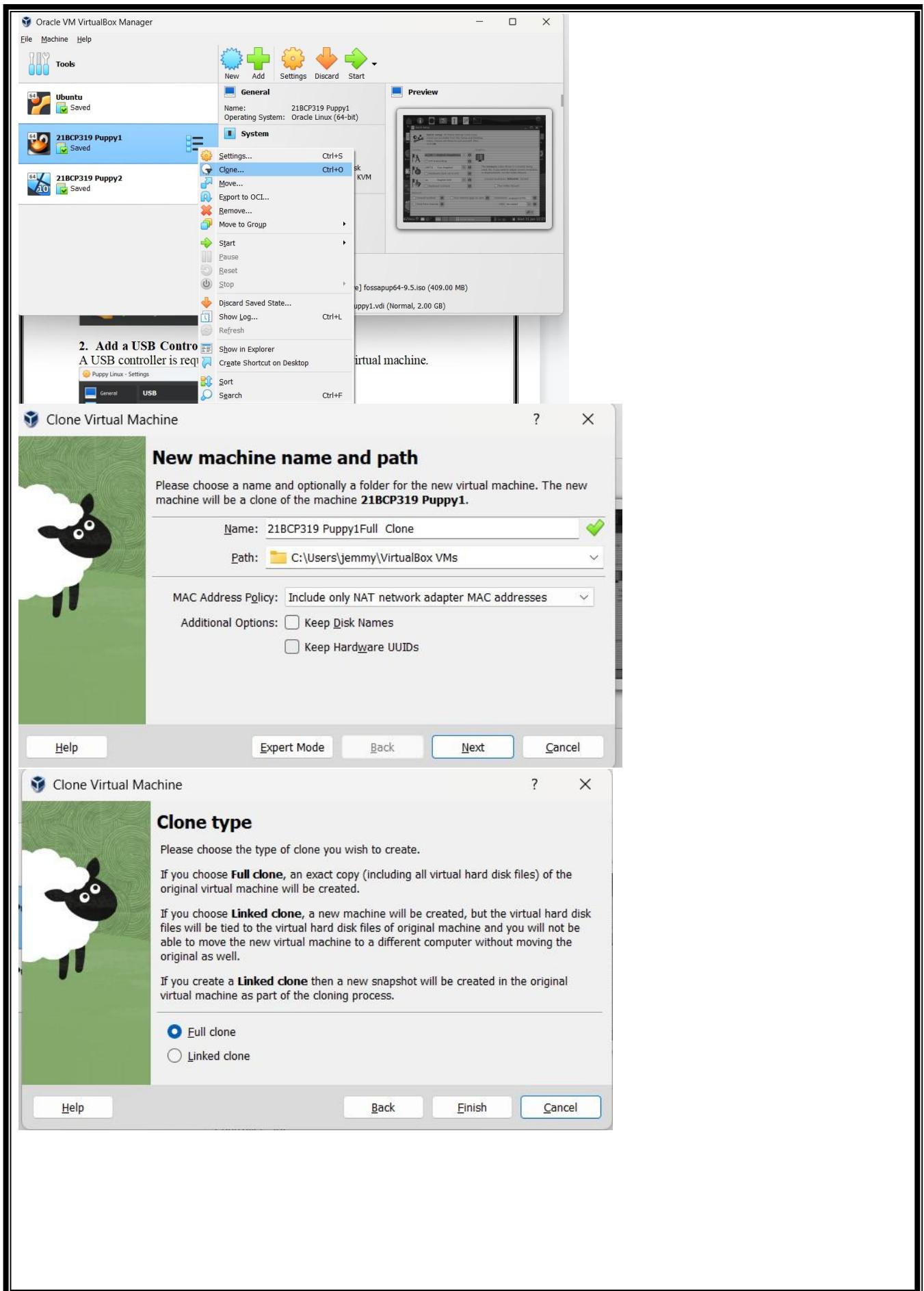
Click on "Clone..."

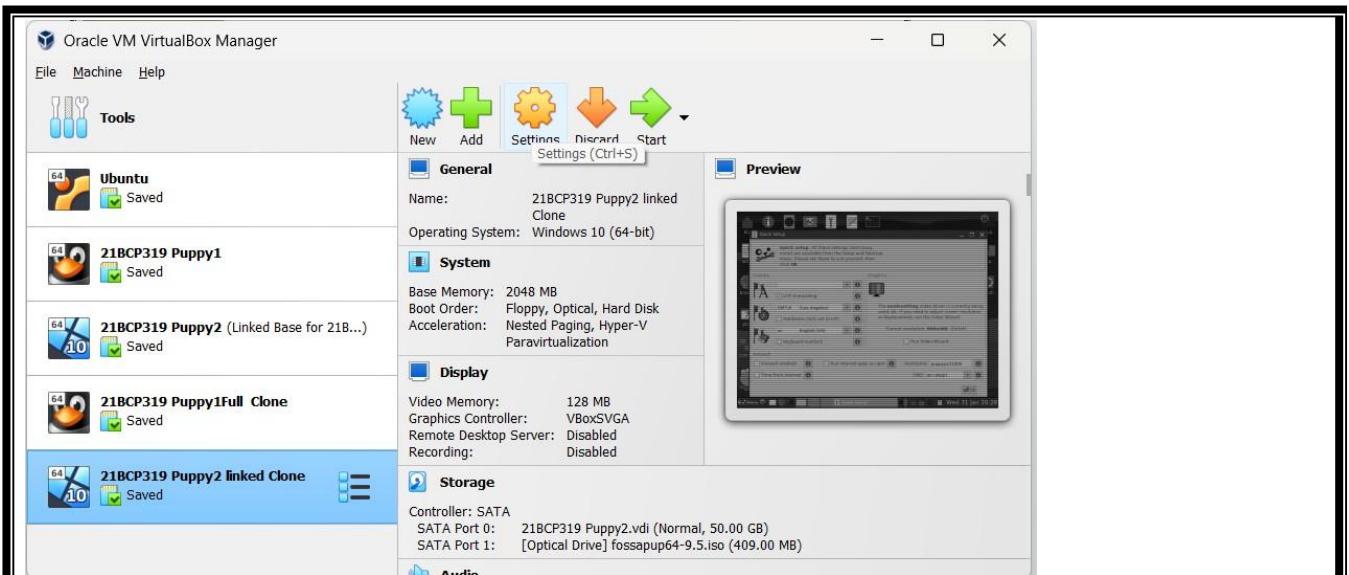
In the cloning wizard, provide a name for the clone.

Choose whether to create a full clone (which includes all the virtual disk data) or a linked clone (which shares the virtual disk with the original VM).

Click "Clone" to create the clone.

- **Review and Adjust Settings (if needed):** After cloning, you might want to review and adjust settings for the cloned VM, such as network settings, cloned disk sizes, etc.
- **Start the Cloned VM:** Start the cloned VM to verify that it's working as expected.





❖ **Linked Cloning:** Linked cloning is a virtualization technique that involves creating a new virtual machine (VM) that shares its virtual disk with an existing VM, known as the parent or source VM. In linked cloning, the new VM only stores the differences or changes made to the virtual disk compared to the original VM. This approach significantly reduces the amount of disk space required for each additional VM.

## ● Create a Snapshot (Optional but Recommended):

Before creating a linked clone, it's good practice to create a snapshot of the original VM. This way, you have a point to revert to if needed.

Right-click on the original VM and select "Snapshot."

## ● Select the VM:

Choose the virtual machine you want to create a linked clone of.

## ● Create Linked Clone:

Right-click on the selected VM.

Click on "Clone..."

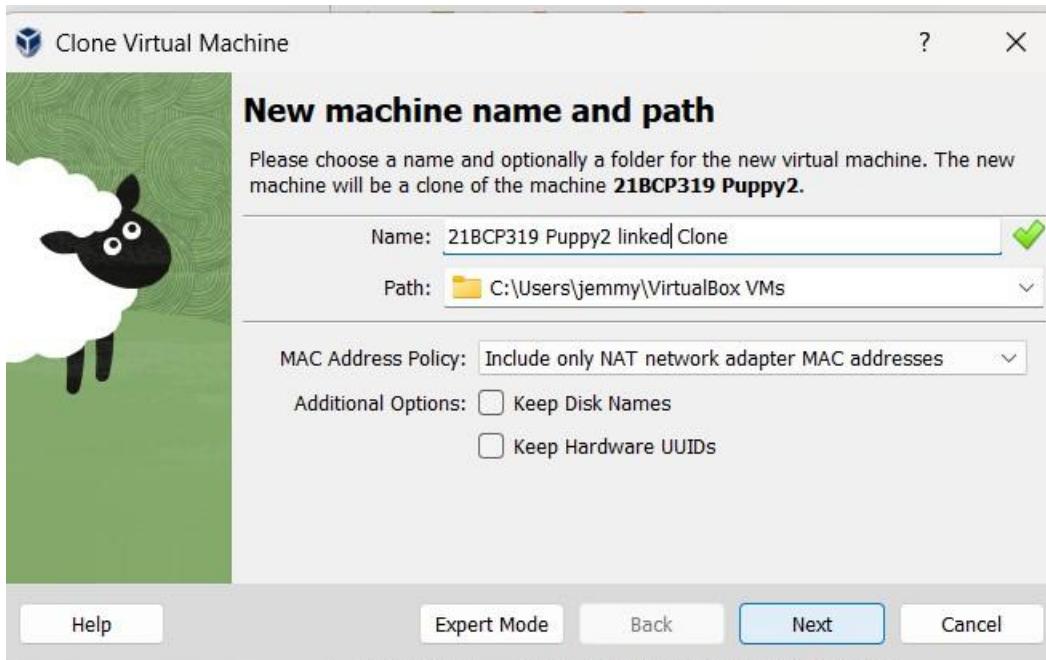
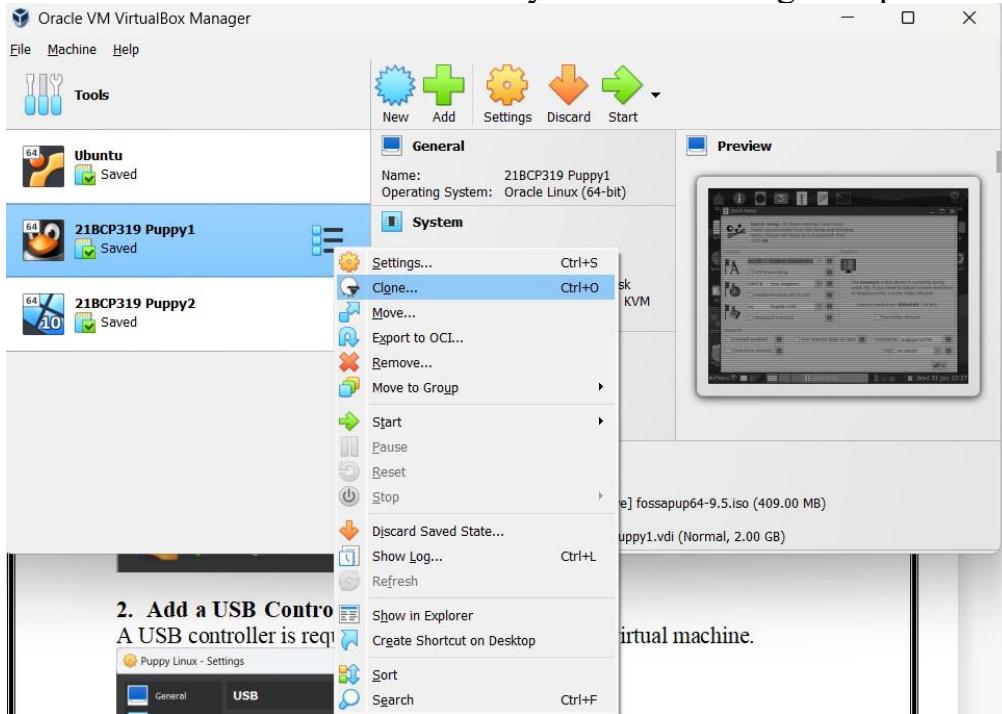
In the cloning wizard, provide a name for the clone.

Choose "Linked clone" under "Clone type."

Click "Clone" to create the linked clone.

## ● Start the Linked Cloned VM:

Start the linked cloned VM to verify that it's working as expected.



**Clone type**

Please choose the type of clone you wish to create.

If you choose **Full clone**, an exact copy (including all virtual hard disk files) of the original virtual machine will be created.

If you choose **Linked clone**, a new machine will be created, but the virtual hard disk files will be tied to the virtual hard disk files of original machine and you will not be able to move the new virtual machine to a different computer without moving the original as well.

If you create a **Linked clone** then a new snapshot will be created in the original virtual machine as part of the cloning process.

Full clone  
 Linked clone

Help Back Finish Cancel

**Oracle VM VirtualBox Manager**

File Machine Help

Tools New Add Settings Discard Start

Name	Description
Ubuntu	Saved
21BCP319 Puppy1	Saved
21BCP319 Puppy2 (Linked Base for 21B...)	Saved
21BCP319 Puppy1Full Clone	Saved
21BCP319 Puppy2 linked Clone	Saved

**General**

Name: 21BCP319 Puppy2 linked Clone  
 Operating System: Windows 10 (64-bit)

**System**

Base Memory: 2048 MB  
 Boot Order: Floppy, Optical, Hard Disk  
 Acceleration: Nested Paging, Hyper-V Paravirtualization

**Display**

Video Memory: 128 MB  
 Graphics Controller: VBoxSVGA  
 Remote Desktop Server: Disabled  
 Recording: Disabled

**Storage**

Controller: SATA  
 SATA Port 0: 21BCP319 Puppy2.vdi (Normal, 50.00 GB)  
 SATA Port 1: [Optical Drive] fossapup64-9.5.iso (409.00 MB)

**Audio**

2. Add a USB Controller to Virtual Machine

A USB controller is required to use a USB device in a virtual machine.

Ubuntu - Settings

General System Display Storage Audio Network Serial Ports

USB

Enable USB Controller

- USB 1.1 (OHCI) Controller
- USB 2.0 (OHCI + EHCI) Controller
- USB 3.0 (xHCI) Controller

USB Device Filters

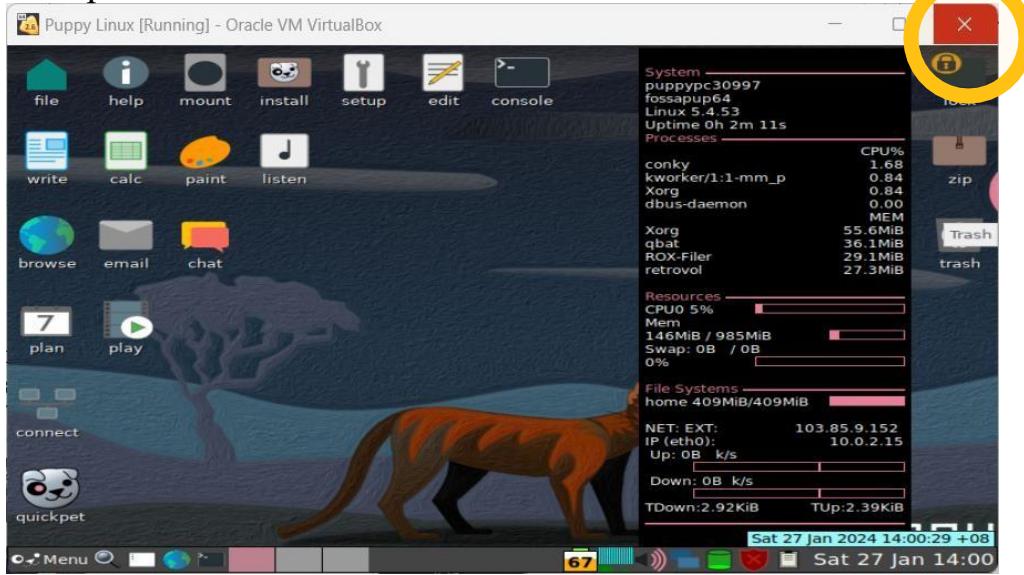
Shared Folders User Interface

OK Cancel Help

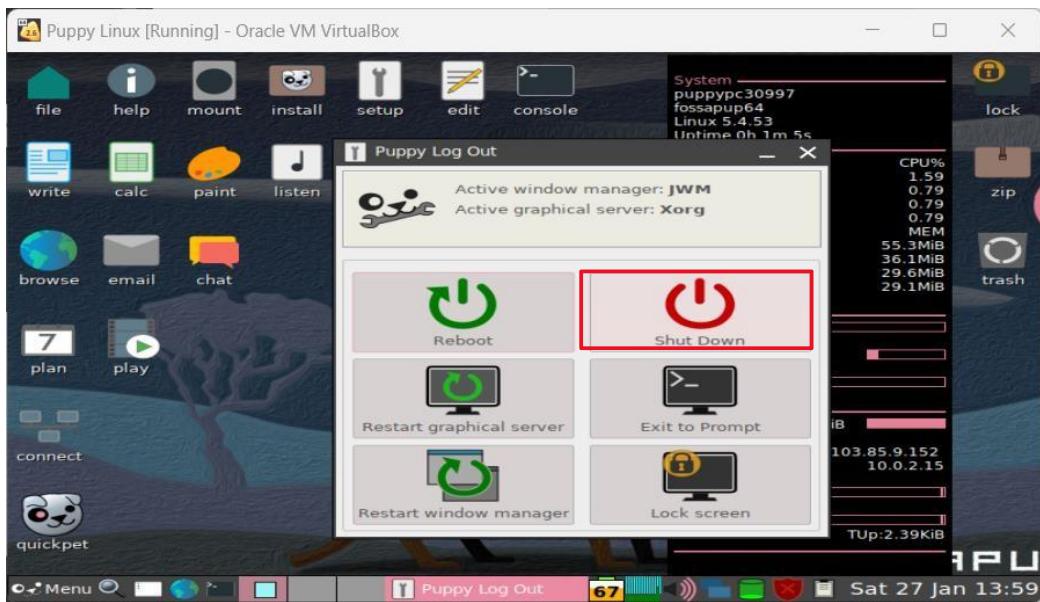
### 3. Closing Virtual Machines & Exiting VMM

#### A. Hard and Soft Power Off

**Hard power** : Hard power off is an abrupt way to shut down a virtual machine. It does not give the virtual machine any time to save its work in progress. You can select this option from the VM menu or from the toolbar.

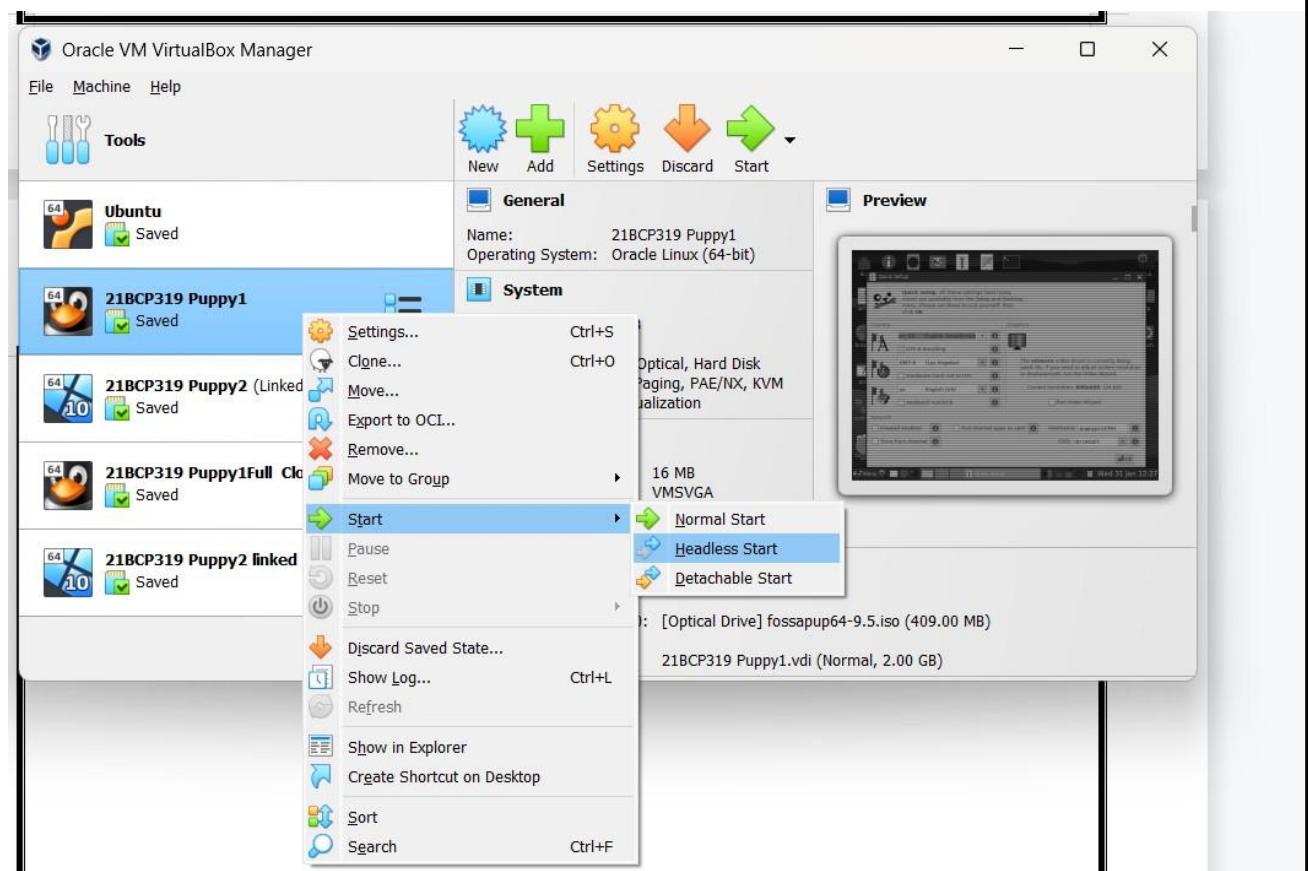


**Soft power off:** Soft power off sends a shut-down signal to the guest operating system, allowing it to shut down gracefully. This option is available from the VM menu or from the toolbar.



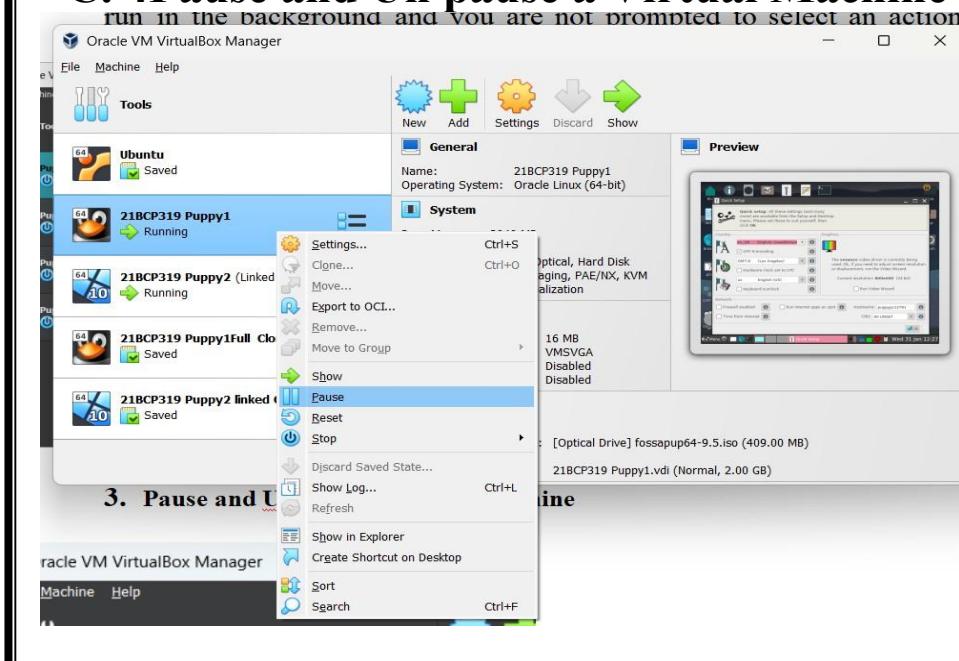
## B. Configure Virtual Machine to Run in the Background

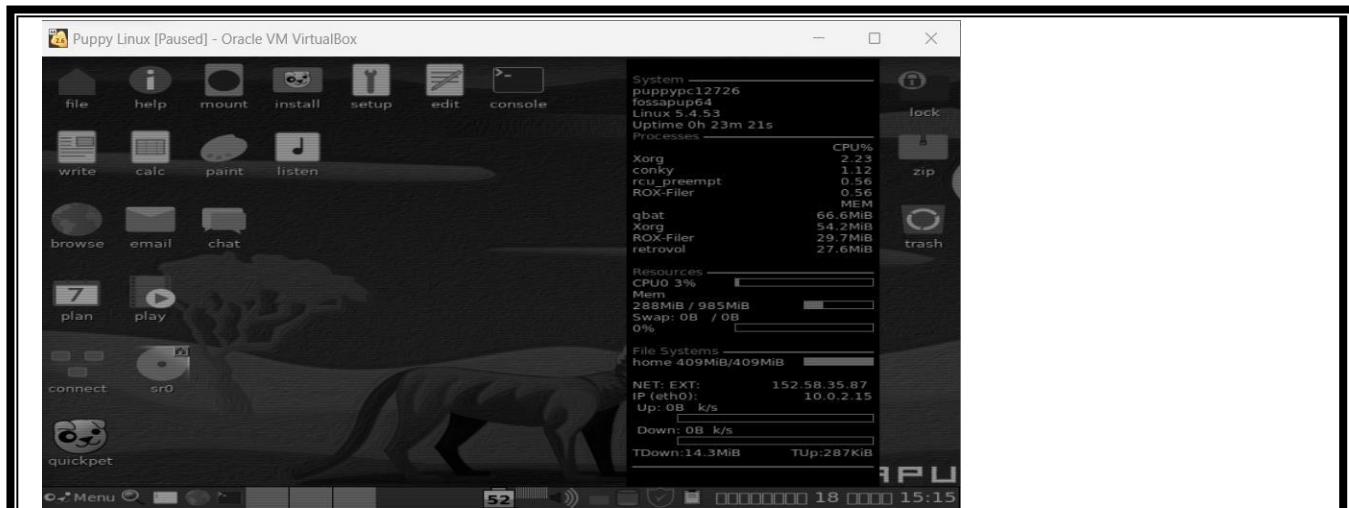
You can configure Workstation Pro preference settings so that virtual machines always run in the background and you are not prompted to select an action when you close powered-on virtual machines.



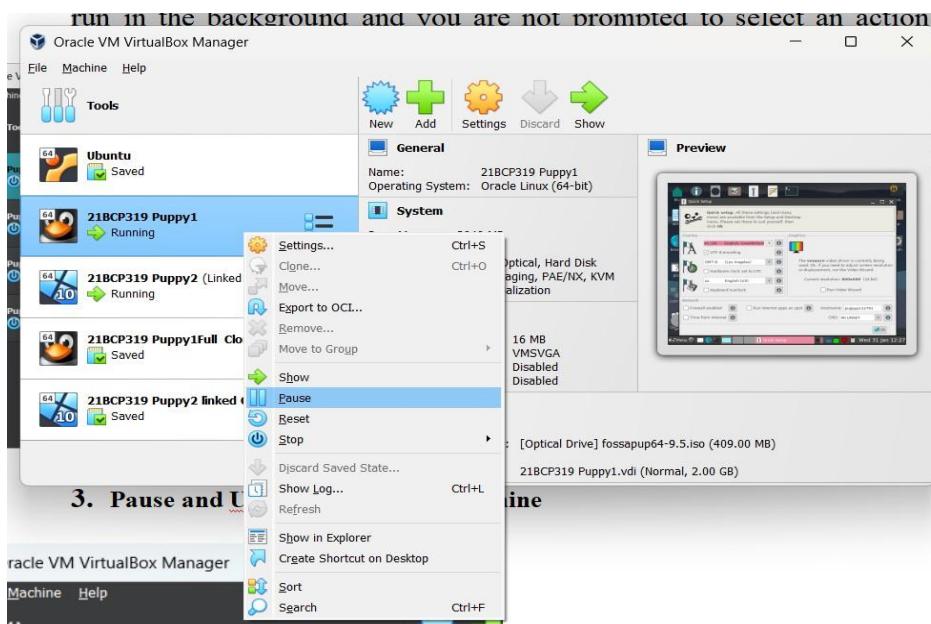
## C. 4Pause and Un pause a Virtual Machine :

run in the background and you are not prompted to select an action

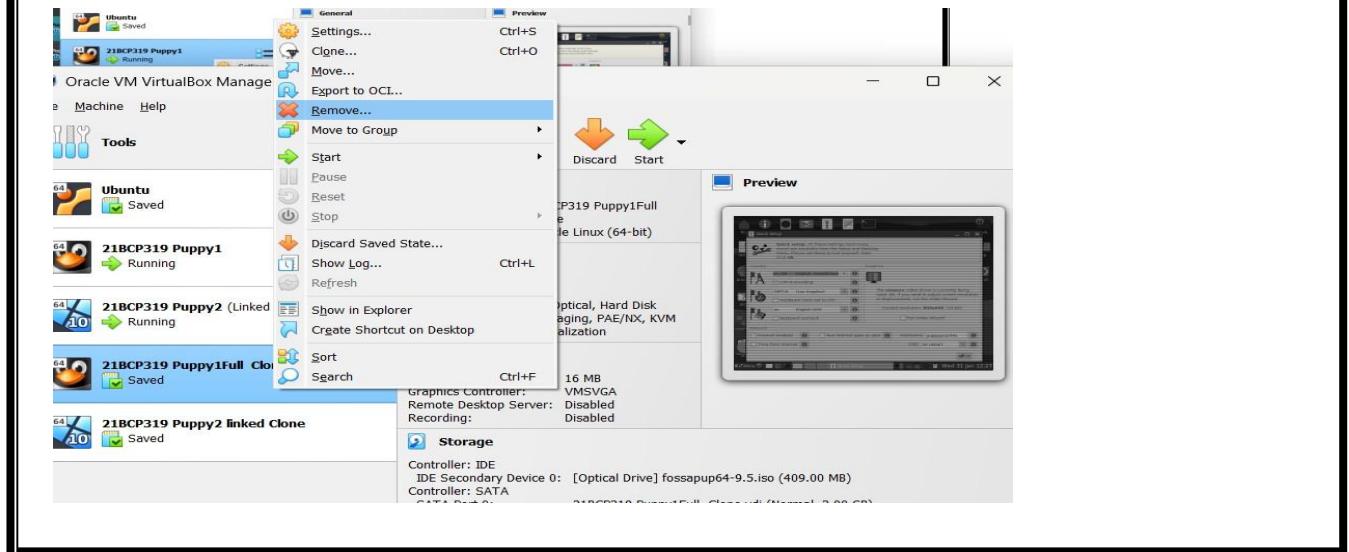


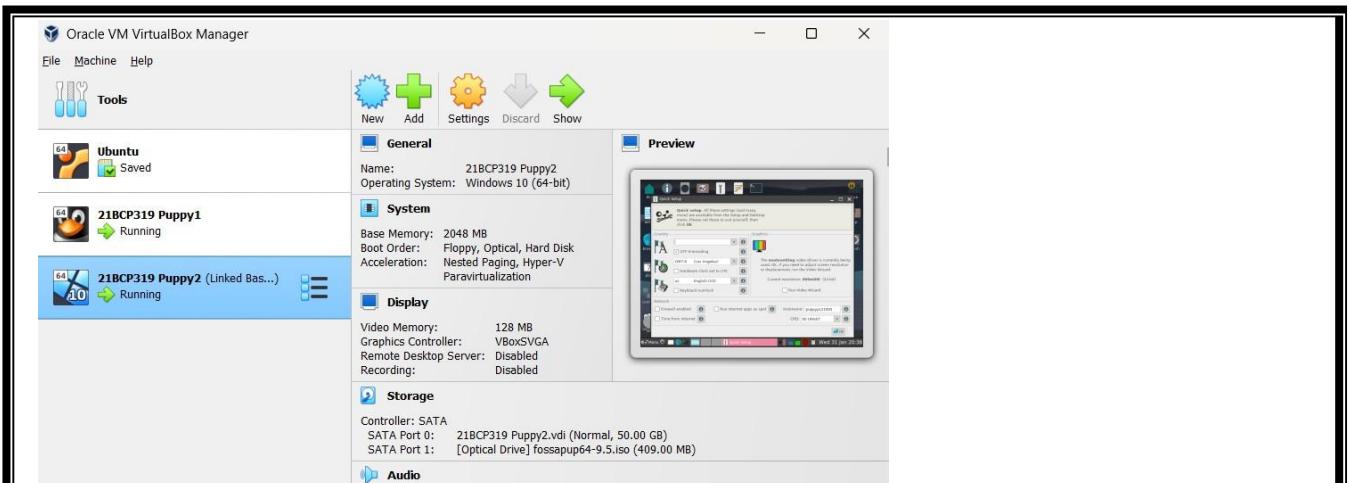


## 4. Pause all Virtual Machine without interacting Workstation Pro

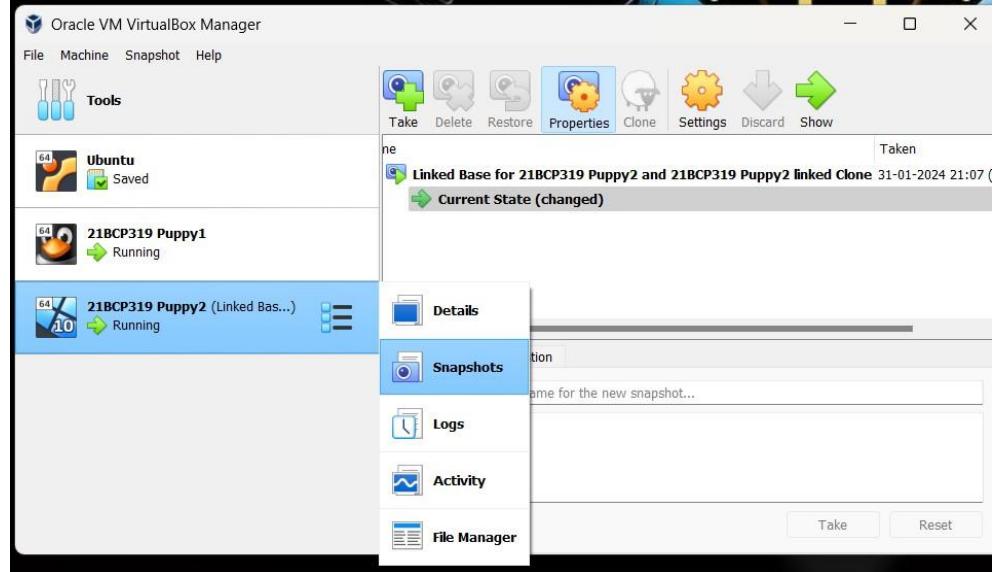
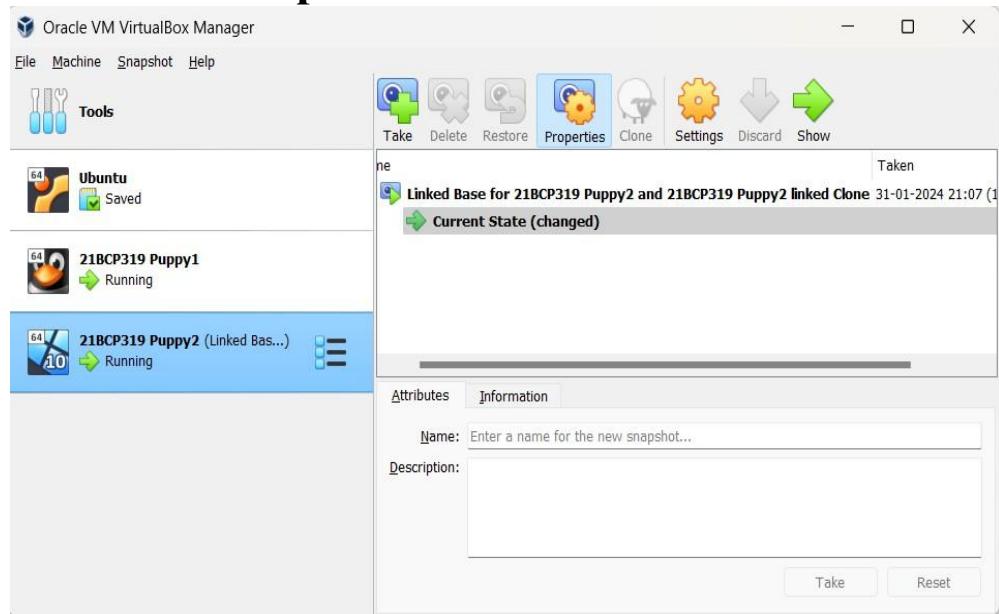


## 4. Delete a Virtual Machine from the disk





## 5. To take snapshot of Virtual Machines :



**Name : DALSANIYA JEMMY .V  
Roll no: 21BCP319  
Div : 5 G-10**

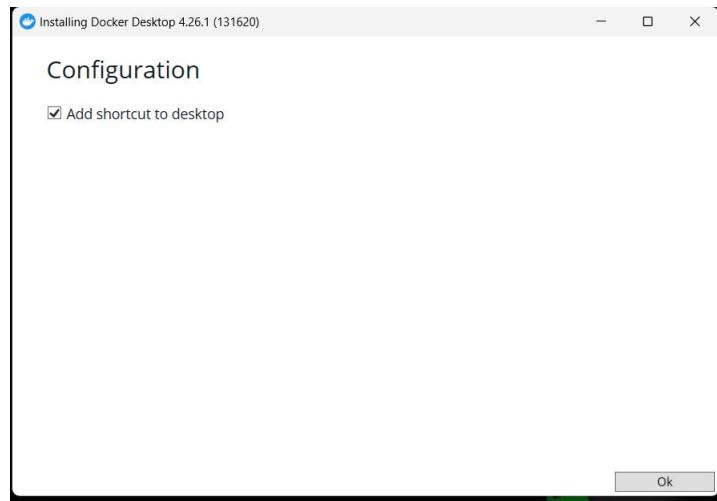
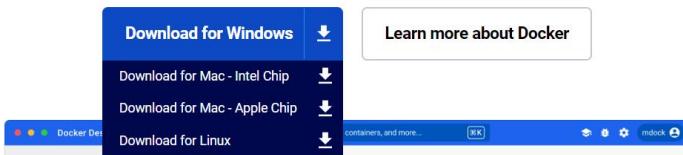
## **Lab 4 Assignment: Working with DOCKER**

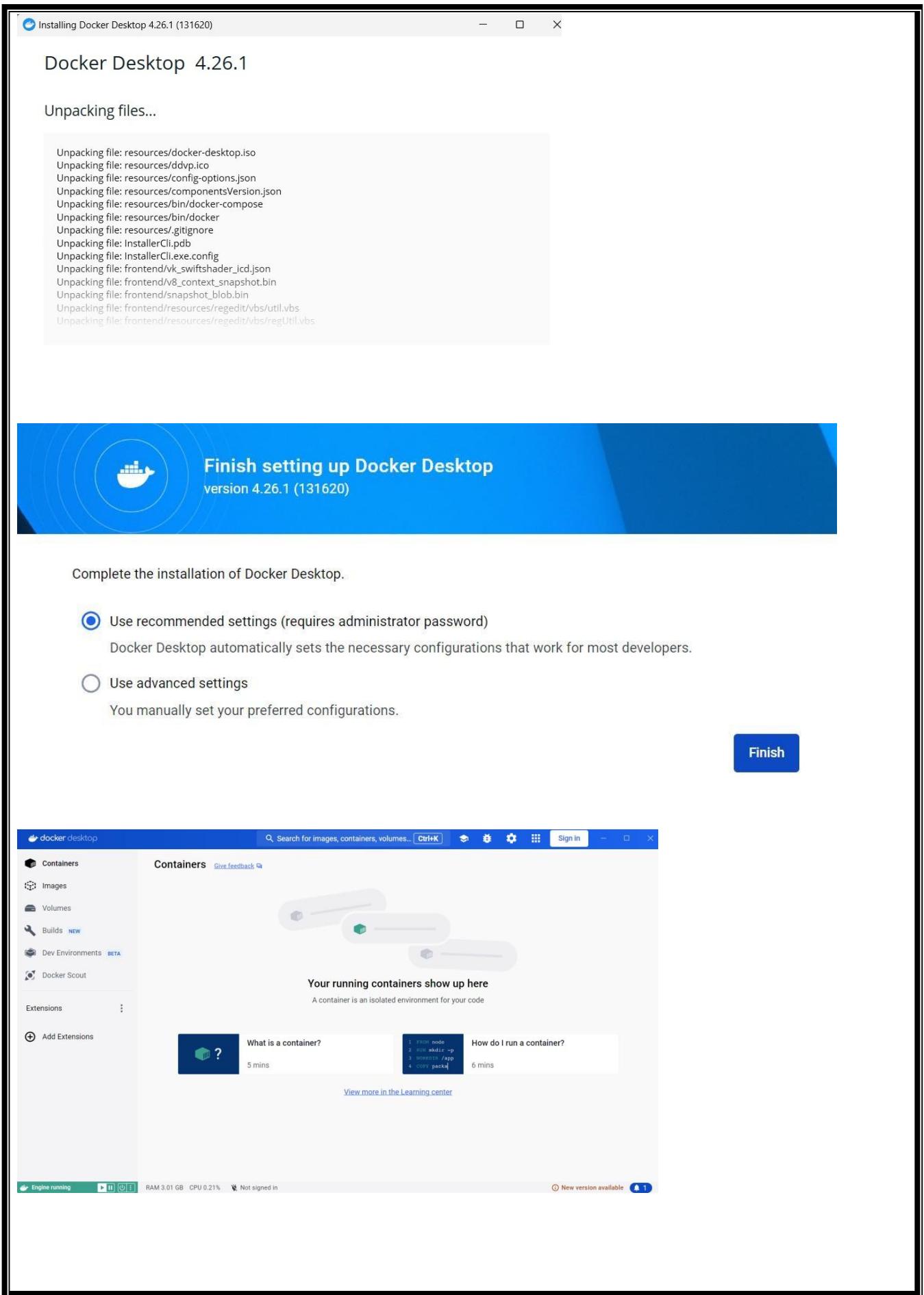
**Aim :** To Explore Docker Container

### **1. Install Docker in your environment**

# **Develop faster. Run anywhere.**

Build with the #1 most-used developer tool





## 2. Verify your Docker by looking up the version

```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\jemmy>docker --version
Docker version 25.0.3, build 4debf41
```

## 3. Run a hello-world container

```
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\jemmy>docker pull hello-world
```

```
C:\Users\jemmy>docker run hello-world
```

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

```
To generate this message, Docker took the following steps:
```

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

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

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

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

## 4. Verify the hello-world image

```
C:\Users\jemmy>docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
hello-world    latest       d2c94e258dcb   9 months ago   13.3kB
```

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	sweet_elba	hello-world ed4424dcde	Exited	N/A		3 minutes ago	

## 5. Get the sample application

**Code** Issues Pull requests Actions Security Insights

main 5 Branches 0 Tags Go to file Code

File	Description	Time
spec	first commit	7 months ago
src	first commit	7 months ago
README.md	first commit	7 months ago
package.json	Update package.json	4 months ago
yarn.lock	first commit	7 months ago

## 6. Run the application on the Docker

### a. Build the Docker Image

```

PS D:\Study\Sem 6\Cloud Computing> git clone https://github.com/docker/getting-started-app.git
Cloning into 'getting-started-app'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (25/25), done.
Receiving objects: 100% (68/68), 1.75 MiB | 995.00 KiB/s, done. Total 68 (delta 12), reused 10 (delta 10), pack-reused 33
Resolving deltas: 100% (12/12), done.
PS D:\Study\Sem 6\Cloud Computing> cd getting-started-app
PS D:\Study\Sem 6\Cloud Computing> touch Dockerfile
PS D:\Study\Sem 6\Cloud Computing> docker build -t getting-started .
ERROR: error during connect: this error may indicate that the docker daemon is not running: Get "http://%2F%2F.pipe%2Fdocker_engine/_ping": open //./pipe/docker_engine: The system cannot find the file specified.
PS D:\Study\Sem 6\Cloud Computing> docker build -t getting-started .
[+] Building 27.5s (11/11) FINISHED
-> [internal] load build definition from Dockerfile
-> >> transferring dockerfile: 188B
-> [internal] load .dockerignore
-> >> transferring context: 2B
-> resolving image config for docker.io/docker/dockerfile:1
-> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f38aa63b13dfccfa89046420e1781752bab202122f8f50032edf31be0021
-> [internal] load metadata for docker.io/library/node:18-alpine
-> [1/4] FROM docker.io/library/node:18-alpine@sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef8437672e1bd72f437
-> [internal] load build context
-> >> transferring context: 6.50MB
-> CACHED [2/4] WORKDIR /app
-> [3/4] COPY .
-> [4/4] RUN yarn install --production
-> exporting to image
-> >> exporting layers
-> > writing image sha256:d89def4768436efc2be97916f966d24bf380dada98bed8e2ff5aafec6a28529
-> > naming to docker.io/library/getting-started

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS D:\Study\Sem 6\Cloud Computing> docker run -dp 127.0.0.1:3000:3000 getting-started
f2e43bba0782b98f814ef5e475e7dc408e8e47d614b826e26027e95e6fbcbecf
PS D:\Study\Sem 6\Cloud Computing>

```

File Edit Selection View Go Run Terminal Help

D:\Study\Sem 6\Cloud Computing> git clone https://github.com/docker/getting-started-app.git

Cloning into 'getting-started-app'...  
remote: Enumerating objects: 68, done.  
remote: Counting objects: 100% (35/35), done.  
remote: Compressing objects: 100% (25/25), done.  
Receiving objects: 100% (68/68), 1.75 MiB | 995.00 KiB/s, done. Total 68 (delta 12), reused 10 (delta 10), pack-reused 33  
Resolving deltas: 100% (12/12), done.

PS D:\Study\Sem 6\Cloud Computing> cd getting-started-app

PS D:\Study\Sem 6\Cloud Computing\getting-started-app> touch Dockerfile

PS D:\Study\Sem 6\Cloud Computing\getting-started-app> docker build -t getting-started .

ERROR: error during connect: this error may indicate that the docker daemon is not running: Get "http://<ip>/v1/docker/engine/ping": open <ip>/v1/docker/engine: The system cannot find the file specified.

PS D:\Study\Sem 6\Cloud Computing\getting-started-app> docker build -t getting-started .

[+] Building 27.5s (11/11) FINISHED

>> [internal] load build definition from Dockerfile  
>> [internal] load dockerfile: 188B  
>> [internal] load .dockerignore  
>> [internal] transfer context: 2B  
>> resolve image config for docker.io/docker/dockerfile:1.1.0  
>> CACHED docker-image://docker.io/docker/dockerfile:1.1.0@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50032edf31be0012  
>> [internal] load metadata for docker.io/library/node:18-alpine  
>> [1/4] FROM docker.io/library/node:18-alpine@sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef8437672e1bd72f437  
>> [internal] load build context  
>> transfer context: 6.50MB  
>> CACHED [2/4] WORKDIR /app  
>> [3/4] COPY . .  
>> [4/4] RUN yarn install --production  
>> exporting to image  
>> exporting layers  
>> writing image sha256:d89def4768436fc2be97916f966d24bf380dada98becd8e2ff5aafec6a28529  
>> naming to docker.io/library/getting-started

What's Next?  
View a summary of image vulnerabilities and recommendations → docker scout quickview

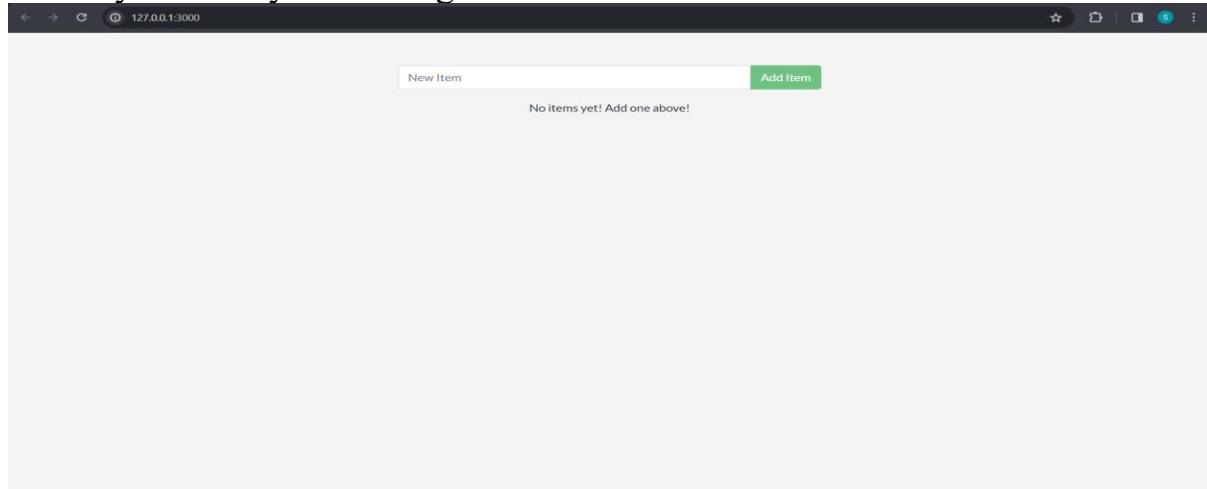
PS D:\Study\Sem 6\Cloud Computing\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-started d2e43bb0782b99f81a4fe5e475e7d40808e047de1ab26e26027e95efbfbcbeef

PS D:\Study\Sem 6\Cloud Computing\getting-started-app>

PS D:\Study\Sem 6\Cloud Computing\getting-started-app> docker run -dp 127.0.0.1:4000:3000 getting-started d2f8449c98e2ce22d756b229535c1f0a1858efdcf5bc235c998601b86266ffa7

PS D:\Study\Sem 6\Cloud Computing\getting-started-app>

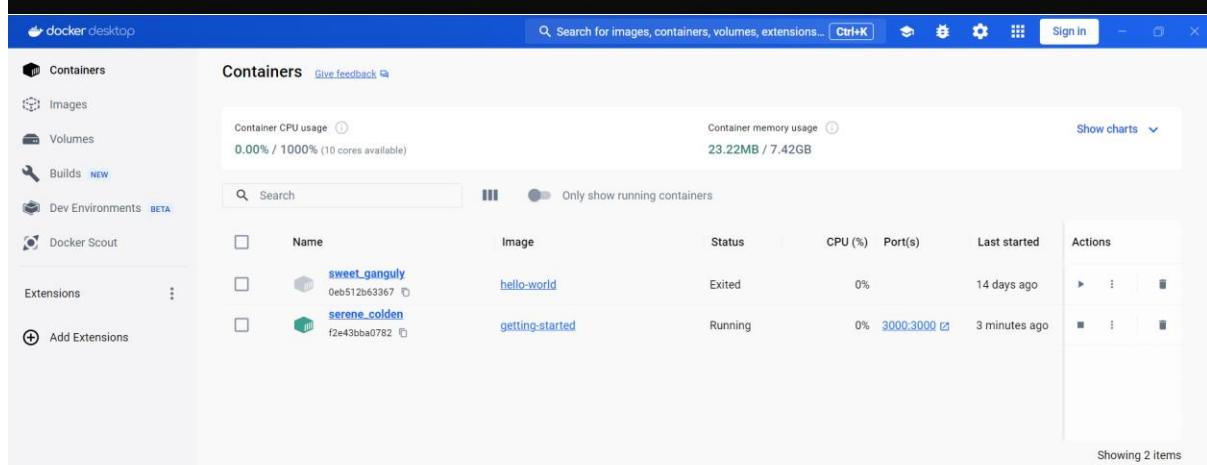
### b. Verify the newly build image



## 7. Run the Docker container

a. Run the container in the detached mode. Name the container “Test Sample”.

```
PS D:\Study\Sem 6\Cloud Computing> docker run -d --name TestSample ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
bcccd10f490ab: Already exists
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214
PS D:\Study\Sem 6\Cloud Computing> |
```



b. List all the running containers.

```
C:\Windows\System32\cmd.exe + ~
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

D:\Study\Sem 6\Cloud Computing>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9687572c2626 getting-started "docker-entrypoint.s..." About a minute ago Up About a minute 3000/tcp, 127.0.0.1:7001->7000/tcp nice_thompson
01015a4b9b92 getting-started "docker-entrypoint.s..." 2 minutes ago Up 2 minutes 3000/tcp, 127.0.0.1:7000->7000/tcp sweet_payne
f2e43bba0782 getting-started "docker-entrypoint.s..." 24 minutes ago Up 24 minutes 127.0.0.1:3000->3000/tcp serene_colden

D:\Study\Sem 6\Cloud Computing>docker stop 9687572c2626
9687572c2626

D:\Study\Sem 6\Cloud Computing>docker stop 01015a4b9b92
01015a4b9b92

D:\Study\Sem 6\Cloud Computing>docker stop f2e43bba0782
f2e43bba0782

D:\Study\Sem 6\Cloud Computing>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

D:\Study\Sem 6\Cloud Computing>
```

### c. Inspect the running containers

```
PS D:\Study\Sem 6\Cloud Computing> docker inspect TestSample
[{"Id": "8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214",
 "Created": "2024-04-04T09:21:10.261156223Z",
 "Path": "/bin/bash",
 "Args": [],
 "State": {
   "Status": "exited",
   "Running": false,
   "Paused": false,
   "Restarting": false,
   "OOMKilled": false,
   "Dead": false,
   "Pid": 0,
   "ExitCode": 0,
   "Error": "",
   "StartedAt": "2024-04-04T09:21:10.75163874Z",
   "FinishedAt": "2024-04-04T09:21:10.775011683Z"
 },
 "Image": "sha256:cab8bf26964cf2e80bae084d5983dab293fdb87485dc6445f3f7bbfc89d7459",
 "ResolvConfPath": "/var/lib/docker/containers/8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214/resolv.conf",
 "HostnamePath": "/var/lib/docker/containers/8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214/hostname",
 "HostsPath": "/var/lib/docker/containers/8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214/hosts",
 "LogPath": "/var/lib/docker/containers/8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214/8dfc458c159a1443b0337d2a7574d19e72f0002167010930a8814716ec718214/json.log",
 "Name": "/TestSample",
 "RestartCount": 0,
 "Driver": "overlay2",
 "Platform": "linux",
 "MountLabel": "",
 "ProcessLabel": "",
 "AppArmorProfile": "",
 "ExecIDs": null,
 "HostConfig": {
   "Binds": null,
   "ContainerIDFile": "",
   "LogConfig": {
     "Type": "json-file",
     "Config": {}
   }
 }
```

### d. Get the logs of “Test Sample” containers

## 8. Access the application

```
D:\Study\Sem 6\Cloud Computing>docker login
Authenticating with existing credentials...
Login Succeeded

D:\Study\Sem 6\Cloud Computing>docker push docker/getting-started
Using default tag: latest
The push refers to repository [docker.io/docker/getting-started]
An image does not exist locally with the tag: docker/getting-started

D:\Study\Sem 6\Cloud Computing>docker tag getting-started shaily21/getting-started

D:\Study\Sem 6\Cloud Computing>docker push shaily21/getting-started
Using default tag: latest
The push refers to repository [docker.io/shaily21/getting-started]
a5a1792fb6b9: Pushed
6dafb22190fb: Pushed
d8eca74bacfb: Pushed
b325b33b9813: Mounted from library/node
4a0d315ad53e: Mounted from library/node
29e213bad130: Mounted from library/node
d4fc045c9e3a: Mounted from library/node
latest: digest: sha256:344a4af9581b2d31d71491f6fdb7b0b39ef434151b11bf375a557c6000eb52e6 size: 1787
```

```
D:\Study\Sem 6\Cloud Computing>docker push docker/getting-started
Using default tag: latest
The push refers to repository [docker.io/docker/getting-started]
An image does not exist locally with the tag: docker/getting-started
```

## 9. Pushing and pulling an image to and from repository

The screenshot shows a web interface with a header bar containing the URL 'localhost:3000' and a search bar. The main content area displays a list of topics in a table format:

Topic	Description	Action
<b>CSS</b>	Cascading Style Sheet	
<b>Javascript</b>	A programming language	
<b>HTML</b>	Hyper Text Markup language	

At the top right of the main content area, there is a button labeled 'Add Topic'.

**Volumes**

Name	Status	Created	Size	Actions
todo-db	in use	1 day ago	604 kB	[More]

**Walkthroughs**

- Multi-container applications (8 mins)
- Persist your data between containers (3 mins)

**Images**

Optional settings

- Container name: (Random name generated if you do not provide one.)
- Ports: Host port: 3000, Container path: /etc/todos
- Volumes: Host path: todo-db, Container path: /etc/todos
- Environment variables: Variable: (empty), Value: (empty)

**Logs**

```
2024-03-08 21:53:11 Using sqlite database at /etc/todos/todo.db
2024-03-08 21:53:11 Listening on port 3000
```

**STATUS**  
Running (0 seconds ago)

**Docker Desktop - intelligent\_noether**

**Logs**

```
2024-02-15 15:36:37 Using sqlite database at /etc/todos/todo.db
2024-02-15 15:36:37 Listening on port 3000
2024-03-07 14:41:59 Using sqlite database at /etc/todos/todo.db
2024-03-07 14:41:59 Listening on port 3000
```

**Containers**

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
intelligent_murdock	getting-started	Created (128)	0%	3000:3000		[More]
eloquent_mclean	glowstation	Exited (1)	0%	7000:7000	22 days ago	[More]
bold_blackwell	glowstation	Created (128)	0%	3000:3000		[More]
interesting_bartik	getting-started	Created (128)	0%	3000:3000		[More]
serene_colden	getting-started	Exited	0%	3000:3000	22 days ago	[More]
sweet_ganguly	hello-world	Exited	0%		1 month ago	[More]
stupefied_satoshi	getting-started:latest	Running	0%	3000:3000	5 minutes ago	[More]
funny_mirzakhan	getting-started:latest	Running	0%	4000:3000	2 minutes ago	[More]

**Containers**

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
intelligent_murdock	getting-started	Created (128)	0%	3000:3000		[More]
eloquent_mclean	glowstation	Exited (1)	0%	7000:7000	22 days ago	[More]
bold_blackwell	glowstation	Created (128)	0%	3000:3000		[More]
interesting_bartik	getting-started	Created (128)	0%	3000:3000		[More]
serene_colden	getting-started	Exited	0%	3000:3000	22 days ago	[More]
sweet_ganguly	hello-world	Exited	0%		1 month ago	[More]
stupefied_satoshi	getting-started:latest	Running	0%	3000:3000	5 minutes ago	[More]
funny_mirzakhan	getting-started:latest	Running	0%	4000:3000	2 minutes ago	[More]

**todo-db**

**Data**

Container name	Image	Port	Target
suspicious_greider	getting-started:latest	4000	/etc
quizzical_noether	getting-started:latest	3000	/etc
stupefied_satoshi	getting-started:latest	3000	/etc
funny_mirzakhan	getting-started:latest	4000	/etc

Jd 21BCP319

Add Topic

## CSS

Cascading Style Sheet



## Javascript

A programming language



## HTML

Hyper Text Markup language



## JEMMY

21BCP319



**Name : DALSANIYA JEMMY .V**  
**Roll no: 21BCP319**  
**Div : 5 G-10**

## **Lab 5 Assignment: Working with DOCKER**

**Aim:** Create two dockers with front-end and back-end applications and connect these containers. The details of the containers are given below:

Web Application - Front-End + Back-End Application

- MariaDB - Back-End (one container)
- Wordpress - Front-End (different container)

Execute MariaDB Container First.

- Make the Container name Static like - my-mariadb-container
- Set the DB Root password using Env Variable - MARIADB\_ROOT\_PASSWORD
- Set the DB Username using Env Variable - MARIADB\_USER
- Set the DB password using Env Variable - MARIADB\_PASSWORD
- Set the DB Name using Env Variable - MARIADB\_DATABASE

```
D:\SEM 6\cloud\lab6>docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=Jemmy.mysql -d mysql:8.0.36
Unable to find image 'mysql:8.0.36' locally
8.0.36: Pulling from library/mysql
9a5c778f631f: Pull complete
ccc451c3fb55: Pull complete
db534de989c8: Pull complete
2c0dfd031fa2: Pull complete
d18a374d12e6: Pull complete
debd262f474e: Pull complete
b665a713d084: Pull complete
0f00ebfc82ae: Pull complete
fcbe1791ff51: Pull complete
084bd453daf0: Pull complete
1fad3f9b7ea3: Pull complete
Digest: sha256:866020f10a03e81055938f34a0ccdb3f3fa6c3b9a47e2dc49a6c4a5abe3d6ae
Status: Downloaded newer image for mysql:8.0.36
8f6206b9f7418eb3cd523aa71efd0d1f0812cc24c6b3ca237f1997ac5b42d85e
```

```
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.36 MySQL Community Server - GPL
```

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

```

mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)

mysql> create database wordpress;
Query OK, 1 row affected (0.01 sec)

mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpress |
+-----+
5 rows in set (0.00 sec)

```

```

D:\SEM 6\cloud\lab6>docker network ls
NETWORK ID      NAME          DRIVER      SCOPE
0d24149b460f    bridge        bridge      local
e6f5bed177c8    host          host       local
a5c398c74111    mern-docker_default  bridge      local
20dc2ae12f94    next-docker_default  bridge      local
d17fc34f08fd    none          null       local
f114b20d4e9e    vite-project_default  bridge      local

```

*Execute Wordpress Container -*

- Make the Container name Static like - `wordpress-container`
- Set the DB Container Name in Env Variable - `WORDPRESS_DB_HOST`
- Set the DB Name in Env Variable - `WORDPRESS_DB_NAME`
- Set the DB User in Env Variable - `WORDPRESS_DB_USER`
- Set the DB password in Env Variable - `WORDPRESS_DB_PASSWORD`
- Expose Front-End Container on port `8080/80`
- Access WebSite on `LocalHost/HostIP:PORT`

```
D:\SEM 6\cloud\lab6>docker run --name jemmy_wordpress --network mysql_network -d wordpress
Unable to find image 'wordpress:latest' locally
latest: Pulling from library/wordpress
8a1e25ce7c4f: Pull complete
5de14226e170: Pull complete
d5aaf617d1d2: Pull complete
d3ba065e262f: Pull complete
142ecae067f5: Pull complete
c1f1b407f749: Pull complete
6a1b2cfb806d: Pull complete
bc1f0226f3e0: Pull complete
994087df8b58: Pull complete
61717c19c02c: Pull complete
11fdc22aa1f1: Pull complete
6fe9d92b0a5a: Pull complete
471efce0ff50: Pull complete
6c5dc4209f58: Pull complete
1574a2d49f96: Pull complete
85e3d9381cf8: Pull complete
13dfc4135261: Pull complete
7776dbd0795b: Pull complete
565f6e0c08f0: Pull complete
ebae3f6276ad: Pull complete
bee8c49ccae9: Pull complete
Digest: sha256:e61ec4b26e768f5068fbf118e3701529fe6399d1910a66cafb4a01983e52863d
Status: Downloaded newer image for wordpress:latest
fb09594c7e258b75648e4b06e7e4a67813df20364084a809cdb617994343b785
```

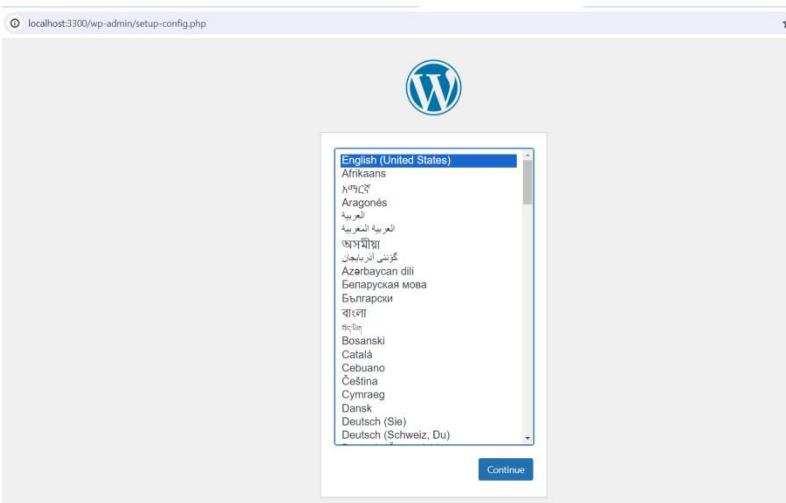
```
D:\SEM 6\cloud\lab6>docker network create mysql_network
f390abf1116df220b047286b03d4539ed84e693f9b0f5aa3f68e7c4cbdc47cf4
```

```
D:\SEM 6\cloud\lab6>docker network connect mysql_network some-mysql
```

```
D:\SEM 6\cloud\lab6>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fb09594c7e25 wordpress "docker-entrypoint.s..." 19 seconds ago Up 18 seconds 80/tcp jemmy_wordpress
8f6206b9f741 mysql:8.0.36 "docker-entrypoint.s..." 20 minutes ago Up 20 minutes 3306/tcp, 33060/tcp some-mysql
```

```
D:\SEM 6\cloud\lab6>docker run --name wordpress_jd --network mysql_network -dp 127.0.0.1:3300:80 wordpress
2ce0ed55c3a51ed9e3e4c921de78f9c22cd5ae528864249d4e8c45a98f767add
```

```
D:\SEM 6\cloud\lab6>docker run --name jemmy_wordpress --network mysql_network -dp 127.0.0.1:3300:80 wordpress
docker: Error response from daemon: Conflict. The container name "jemmy_wordpress" is already in use by container "fb09594c7e258b75648e4b06e7e4a67813df20364084a809cdb617994343b785". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```



Welcome to WordPress. Before getting started, you will need to know the following items.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

This information is being used to create a `wp-config.php` file. **If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? [Read the support article on `wp-config.php`.](#)

In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...

[Let's go!](#)

Below you should enter your database connection details. If you are not sure about these, contact your host.

<b>Database Name</b>	<input type="text" value="wordpress"/>
The name of the database you want to use with WordPress.	
<b>Username</b>	<input type="text" value="some-mysql"/>
Your database username.	
<b>Password</b>	<input type="password" value="password"/> <a href="#">Show</a>
Your database password.	
<b>Database Host</b>	<input type="text" value="localhost"/>
You should be able to get this info from your web host, if <code>localhost</code> does not work.	
<b>Table Prefix</b>	<input type="text" value="wp_"/>
If you want to run multiple WordPress installations in a single database, change this.	

[Submit](#)

**Name : DALSANIYA JEMMY .V  
Roll no: 21BCP319  
Div : 5 G-10**

## **Lab 6 Assignment: Working with DOCKER**

**Aim:** Create a customized docker image and perform the following

### **1. Create Docker File by own.**

```
FROM python:latest

WORKDIR /app

COPY hello_21BCP319.py /app

CMD ["python", "hello_21BCP319.py"]
```

### **2. Get the latest python image from Docker Repository.**

```
PS D:\SEM 6\cloud\lab7> docker pull python:latest
latest: Pulling from library/python
71215d55680c: Pull complete
3cb8f9c23302: Pull complete
5f899db30843: Pull complete
567db630df8d: Pull complete
d68cd2123173: Pull complete
63941d09e532: Pull complete
097431623722: Pull complete
09527fa4de8d: Pull complete
Digest: sha256:19973e1796237522ed1fcc1357c766770b47dc15854eafdda055b65953fe5ec1
Status: Downloaded newer image for python:latest
docker.io/library/python:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview python:latest
```

### **3) Write down a simple python program or get it from internet.**

```
py hello_21BCP319.py
1  print("hello 21BCP319!")
```

### **4) Run Python Program inside the Container.**

```
● PS D:\SEM 6\cloud\lab7> docker build -t app .
[+] Building 0.5s (8/8) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 143B
=> [internal] load metadata for docker.io/library/python:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/python:latest
=> [internal] load build context
=> => transferring context: 68B
=> [2/3] WORKDIR /app
=> [3/3] COPY hello_21BCP319.py /app
=> exporting to image
=> => exporting layers
=> => writing image sha256:b43b2234396665a0e3f8313e19ad50c94cf85ab46a9a28120880cd0e000beaad
=> => naming to docker.io/library/app

View build details: docker-desktop://dashboard/build/default/default/xkz5bndcg1pyqo6hqk3xi3kxg

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

```
● PS D:\SEM 6\cloud\lab7> docker run app
hello 21BCP319!
```

## 5) Tag and Push the Image to Docker Hub.

```
● PS D:\SEM 6\cloud\lab7> docker push app
Using default tag: latest
The push refers to repository [docker.io/library/app]
2c9245d6eb0c: Preparing
659fed5a3f44: Preparing
9adbc4b1428d: Preparing
f52093e4f67d: Preparing
1193f41e6b14: Preparing
e077e19b6682: Waiting
21e1c4948146: Waiting
68866beb2ed2: Waiting
e6e2ab10dba6: Waiting
0238a1790324: Waiting
denied: requested access to the resource is denied
● PS D:\SEM 6\cloud\lab7> docker login -u jemmy33 -p Jemmy.docker docker.io
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
```

## 6) Remove Image from Local and Again execute it from Hub.

```
E:\Study\Sem-6\Cloud\Lab\Lab7>
E:\Study\Sem-6\Cloud\Lab\Lab7>docker image rm app
Untagged: app:latest

● PS D:\SEM 6\cloud\lab7> docker run app
hello 21BCP319!
```

**Name : DALSANIYA JEMMY .V**  
**Roll no: 21BCP319**  
**Div : 5 G-10**

## **Lab 7 Assignment: Working with DOCKER**

**Aim:** Create three containers and access these containers from each other by using IP address and container name.

### **Download Busybox:**

```
PS D:\SEM 6\cloud\lab8> docker run -it --name 21bcn319_busybox busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
7b2699543f22: Pull complete
Digest: sha256:c3839dd800b9eb7603340509769c43e146a74c63dca3045a8e7dc8ee07e53966
Status: Downloaded newer image for busybox:latest
/ # ls
bin dev etc home lib lib64 proc root sys tmp usr var
/ #
```

### **1) Create three containers.**

- PS D:\SEM 6\cloud\lab8> docker run -d --name container\_a busybox sleep infinity  
5b0a34e7262fab47a8a17ed61e39b9a41f9671c3a48fd43324f1a53c46c68a7f
- PS D:\SEM 6\cloud\lab8> docker run -d --name container\_b busybox sleep infinity  
82e67262cb73d94c07e2bcec1c4e6eb31217b670f8ef787ab8ee999fa4e48e1c
- PS D:\SEM 6\cloud\lab8> docker run -d --name container\_c busybox sleep infinity  
755764b3f230e67cf99042d9bb92281ead175b6983a2a042e4f53a711aefa8e7
- PS D:\SEM 6\cloud\lab8>

### **Create new network and connect 3 container with network**

- PS D:\SEM 6\cloud\lab8> docker network create 21BCP319\_NET  
39a1c734a1c9aeaab257f0de57df7bb79173fc3ab2037b77a891674845048e54
- PS D:\SEM 6\cloud\lab8> docker network connect 21BCP319\_NET container\_a
- PS D:\SEM 6\cloud\lab8> docker network connect 21BCP319\_NET container\_b
- PS D:\SEM 6\cloud\lab8> docker network connect 21BCP319\_NET container\_c  
PS D:\SEM 6\cloud\lab8>

### **2) Access these containers using IP address**

```
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' container_a
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker run -e MYSQL_ROOT_PASSWORD=123456 --name mydbcontainer -d mariadb
78d655e56de4e9afa5c5a9eaf6428a701282163153b533cfe89b901a8360f16
PS E:\Study\Sem-6\Cloud\Lab\Lab8> ^C
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker run -d --name container_a busybox sleep infinity
d74e62cf90800958daef9f867af4046400c4ffa75fea3b38e5b6674ef0053d2
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker run -d --name container_b busybox sleep infinity
d0a02c09edbe03439cd4826249accc7e94442f66eb0693be22d91ceec26b0269
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker run -d --name container_c busybox sleep infinity
bcc97318011107acf35b14b1ca80813b7a9623b53edc49d0da579ef585a72b
```

```
PS E:\Study\Sem-6\Cloud\Lab\Lab8>
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' busybox_container1
Error: No such object: busybox_container1
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' container_a
172.21.0.2172.17.0.2
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker exec container_c ping 172.21.0.2172.17.0.2
ping: bad address '172.21.0.2172.17.0.2'
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker exec container_c ping 172.21.0.2
PING 172.21.0.2 (172.21.0.2): 56 data bytes
```

```
PING 172.21.0.2 (172.21.0.2): 56 data bytes
64 bytes from 172.21.0.2: seq=0 ttl=64 time=0.085 ms
64 bytes from 172.21.0.2: seq=1 ttl=64 time=0.093 ms
64 bytes from 172.21.0.2: seq=2 ttl=64 time=0.110 ms
64 bytes from 172.21.0.2: seq=3 ttl=64 time=0.100 ms
64 bytes from 172.21.0.2: seq=4 ttl=64 time=0.067 ms
64 bytes from 172.21.0.2: seq=5 ttl=64 time=0.099 ms
64 bytes from 172.21.0.2: seq=6 ttl=64 time=0.101 ms
64 bytes from 172.21.0.2: seq=7 ttl=64 time=0.133 ms
64 bytes from 172.21.0.2: seq=8 ttl=64 time=0.140 ms
64 bytes from 172.21.0.2: seq=9 ttl=64 time=0.097 ms
64 bytes from 172.21.0.2: seq=10 ttl=64 time=0.103 ms
64 bytes from 172.21.0.2: seq=11 ttl=64 time=0.097 ms
64 bytes from 172.21.0.2: seq=12 ttl=64 time=0.115 ms
64 bytes from 172.21.0.2: seq=13 ttl=64 time=0.107 ms
64 bytes from 172.21.0.2: seq=14 ttl=64 time=0.103 ms
64 bytes from 172.21.0.2: seq=15 ttl=64 time=0.093 ms
64 bytes from 172.21.0.2: seq=16 ttl=64 time=0.078 ms
64 bytes from 172.21.0.2: seq=17 ttl=64 time=0.106 ms
64 bytes from 172.21.0.2: seq=18 ttl=64 time=0.104 ms
64 bytes from 172.21.0.2: seq=19 ttl=64 time=0.096 ms
64 bytes from 172.21.0.2: seq=20 ttl=64 time=0.096 ms
64 bytes from 172.21.0.2: seq=21 ttl=64 time=0.105 ms
64 bytes from 172.21.0.2: seq=22 ttl=64 time=0.091 ms
64 bytes from 172.21.0.2: seq=23 ttl=64 time=0.106 ms
64 bytes from 172.21.0.2: seq=24 ttl=64 time=0.048 ms
64 bytes from 172.21.0.2: seq=25 ttl=64 time=0.163 ms
64 bytes from 172.21.0.2: seq=26 ttl=64 time=0.128 ms
64 bytes from 172.21.0.2: seq=27 ttl=64 time=0.105 ms
64 bytes from 172.21.0.2: seq=28 ttl=64 time=0.095 ms
64 bytes from 172.21.0.2: seq=29 ttl=64 time=0.091 ms
```

### 3) Access the container using the container name :

```
PS E:\Study\Sem-6\Cloud\Lab\Lab8>
PS E:\Study\Sem-6\Cloud\Lab\Lab8> docker exec container_b ping container_c
PING container_c (172.21.0.4): 56 data bytes
64 bytes from 172.21.0.4: seq=0 ttl=64 time=0.149 ms
64 bytes from 172.21.0.4: seq=1 ttl=64 time=0.111 ms
64 bytes from 172.21.0.4: seq=2 ttl=64 time=0.108 ms
64 bytes from 172.21.0.4: seq=3 ttl=64 time=0.168 ms
64 bytes from 172.21.0.4: seq=4 ttl=64 time=0.091 ms
64 bytes from 172.21.0.4: seq=5 ttl=64 time=0.182 ms
64 bytes from 172.21.0.4: seq=6 ttl=64 time=0.143 ms
64 bytes from 172.21.0.4: seq=7 ttl=64 time=0.166 ms
64 bytes from 172.21.0.4: seq=8 ttl=64 time=0.144 ms
64 bytes from 172.21.0.4: seq=9 ttl=64 time=0.209 ms
64 bytes from 172.21.0.4: seq=10 ttl=64 time=0.148 ms
64 bytes from 172.21.0.4: seq=11 ttl=64 time=0.092 ms
64 bytes from 172.21.0.4: seq=12 ttl=64 time=0.141 ms
64 bytes from 172.21.0.4: seq=13 ttl=64 time=0.175 ms
64 bytes from 172.21.0.4: seq=14 ttl=64 time=0.087 ms
64 bytes from 172.21.0.4: seq=15 ttl=64 time=0.152 ms
64 bytes from 172.21.0.4: seq=16 ttl=64 time=0.084 ms
64 bytes from 172.21.0.4: seq=17 ttl=64 time=0.132 ms
64 bytes from 172.21.0.4: seq=18 ttl=64 time=0.102 ms
64 bytes from 172.21.0.4: seq=19 ttl=64 time=0.121 ms
64 bytes from 172.21.0.4: seq=20 ttl=64 time=0.163 ms
64 bytes from 172.21.0.4: seq=21 ttl=64 time=0.143 ms
64 bytes from 172.21.0.4: seq=22 ttl=64 time=0.117 ms
64 bytes from 172.21.0.4: seq=23 ttl=64 time=0.169 ms
64 bytes from 172.21.0.4: seq=24 ttl=64 time=0.111 ms
64 bytes from 172.21.0.4: seq=25 ttl=64 time=0.168 ms
64 bytes from 172.21.0.4: seq=26 ttl=64 time=0.136 ms
64 bytes from 172.21.0.4: seq=27 ttl=64 time=0.173 ms
```

```
64 bytes from 172.21.0.4: seq=23 ttl=64 time=0.169 ms
64 bytes from 172.21.0.4: seq=24 ttl=64 time=0.111 ms
64 bytes from 172.21.0.4: seq=25 ttl=64 time=0.168 ms
64 bytes from 172.21.0.4: seq=26 ttl=64 time=0.136 ms
64 bytes from 172.21.0.4: seq=27 ttl=64 time=0.173 ms
64 bytes from 172.21.0.4: seq=28 ttl=64 time=0.115 ms
64 bytes from 172.21.0.4: seq=29 ttl=64 time=0.110 ms
64 bytes from 172.21.0.4: seq=30 ttl=64 time=0.073 ms
64 bytes from 172.21.0.4: seq=31 ttl=64 time=0.159 ms
64 bytes from 172.21.0.4: seq=32 ttl=64 time=0.116 ms
64 bytes from 172.21.0.4: seq=33 ttl=64 time=0.095 ms
64 bytes from 172.21.0.4: seq=34 ttl=64 time=0.118 ms
64 bytes from 172.21.0.4: seq=35 ttl=64 time=0.090 ms
64 bytes from 172.21.0.4: seq=36 ttl=64 time=0.100 ms
64 bytes from 172.21.0.4: seq=37 ttl=64 time=0.145 ms
64 bytes from 172.21.0.4: seq=38 ttl=64 time=0.119 ms
64 bytes from 172.21.0.4: seq=39 ttl=64 time=0.134 ms
64 bytes from 172.21.0.4: seq=40 ttl=64 time=0.114 ms
64 bytes from 172.21.0.4: seq=41 ttl=64 time=0.136 ms
64 bytes from 172.21.0.4: seq=42 ttl=64 time=0.222 ms
64 bytes from 172.21.0.4: seq=43 ttl=64 time=0.172 ms
64 bytes from 172.21.0.4: seq=44 ttl=64 time=0.130 ms
64 bytes from 172.21.0.4: seq=45 ttl=64 time=0.103 ms
64 bytes from 172.21.0.4: seq=46 ttl=64 time=0.183 ms
64 bytes from 172.21.0.4: seq=47 ttl=64 time=0.080 ms
64 bytes from 172.21.0.4: seq=48 ttl=64 time=0.144 ms
64 bytes from 172.21.0.4: seq=49 ttl=64 time=0.204 ms
64 bytes from 172.21.0.4: seq=50 ttl=64 time=0.182 ms
64 bytes from 172.21.0.4: seq=51 ttl=64 time=0.151 ms
64 bytes from 172.21.0.4: seq=52 ttl=64 time=0.194 ms
PS E:\Study\Sem-6\Cloud\Lab\Lab8> |
```

**Name : DALSANIYA JEMMY .V**  
**Roll no: 21BCP319**  
**Div : 5 G-10**

## **Lab 8 Assignment: Working with DOCKER - COMPOSE**

**Aim:** Re-Do the problem of Lab 5 using docker compose. Also do assign a static IP address to WordPress container using subnetting.

Create two docker containers with front-end and back-end applications and connect these containers using docker compose. The details of the containers are given below:

Web Application - Front-End + Back-End Application

- MariaDB - Back-End (one container)
- Wordpress - Front-End (different container) Execute MariaDB Container First.
  
- Container name Static like – ROLL\_NUMBER-mariadb-container
- Set the DB Root password using Env Variable - MARIADB\_ROOT\_PASSWORD
- Set the DB Username using Env Variable - MARIADB\_USER
- Set the DB password using Env Variable - MARIADB\_PASSWORD
- Set the DB Name using Env Variable - MARIADB\_DATABASE Execute Wordpress Container -
  
- Make the Container name Static like – ROLL\_NUMBER\_WORDPRESS-CONTAINER
- DB Container Name - ROLL\_NUMBER\_WORDPRESS\_DB\_HOST
- DB Name in Env Variable - WORDPRESS\_DB\_NAME
- DB User in Env Variable - WORDPRESS\_DB\_USER
- DB password in Env Variable - WORDPRESS\_DB\_PASSWORD
- Expose Front-End Container on port 8080/80
- Access WebSite on LocalHost/HostIP:PORT

Docker-compse.yaml

---

version: '3.1'

services:

wordpress:

image: wordpress restart: always ports:

- 8080:80

environment: WORDPRESS\_DB\_HOST: db

WORDPRESS\_DB\_USER: exampleuser WORDPRESS\_DB\_PASSWORD: examplepass

WORDPRESS\_DB\_NAME: exampledb volumes:

- wordpress:/var/www/html

db:

image: mysql:8.0 restart: always environment:

MYSQL\_DATABASE: exampledb MYSQL\_USER: exampleuser MYSQL\_PASSWORD: examplepass MYSQL\_RANDOM\_ROOT\_PASSWORD: '1'

volumes:

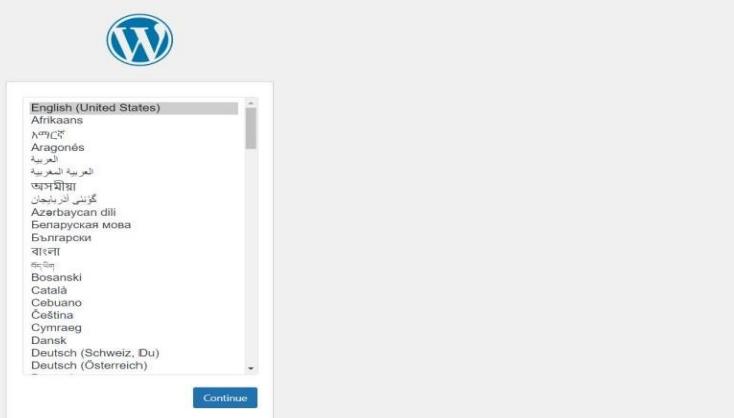
- db:/var/lib/mysql

volumes: wordpress: db:

````

```
E:\ JD :\college\sem6\Cloud\1\compose>docker compose up -d
[+] Running 2/2
  ✓ Container compose-db-1          Running
  ✓ Container compose-wordpress-1   Started
```

| Name        | Image        | Status        | CPU (%) | Port(s) | Last started  | Actions                                      |
|-------------|--------------|---------------|---------|---------|---------------|----------------------------------------------|
| compose     |              | Running (2/2) | 0.38%   |         | 2 minutes ago | <span>⋮</span> <span>⋮</span> <span>⋮</span> |
| db-1        | 7b2760098a86 | mysql:8.0     | 0.38%   |         | 4 minutes ago | <span>⋮</span> <span>⋮</span> <span>⋮</span> |
| wordpress-1 | aeebee3c2261 | wordpress     | 0%      | 8080:80 | 2 minutes ago | <span>⋮</span> <span>⋮</span> <span>⋮</span> |



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

|                          |                                                                            |
|--------------------------|----------------------------------------------------------------------------|
| Site Title               | Docker-Assignment                                                          |
| Username                 | User                                                                       |
| Password                 | dkIws7jAu*u8QfX#T7<br>Strong                                               |
| Your Email               | hello@gmail.com                                                            |
| Search engine visibility | <input type="checkbox"/> Discourage search engines from indexing this site |

**Important:** You will need this password to log in. Please store it in a secure location.

Double-check your email address before continuing.



Success!

WordPress has been installed. Thank you, and enjoy!

Username User

Password Your chosen password.

[Log in](#)

Username or Email Address

user

Password

  Remember Me[Log In](#)[Lost your password?](#)[← Go to Docker-Assignment](#)

Howdy, User

Screen Options ▾ Help ▾

## Dashboard

Welcome to WordPress!

[Learn more about the 6.5.2 version.](#)

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

[Add a new page](#)

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

[Open site editor](#)

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

[Edit styles](#)

**Site Health Status**

No information yet... Site health checks will automatically run periodically to gather information about your site. You can also [visit the Site Health screen](#) to gather information

**Quick Draft**

Title   
Content

Drag boxes here

Jemmy

```
E:\Ronit\college\sem6\Cloud\1\compose>docker compose down
[+] Running 3/3
✓ Container compose-wordpress-1   Removed
✓ Container compose-db-1          Removed
✓ Network compose_default         Removed
```

