Lab 5: Cloud Virtualisation Technologies

Cloud virtualisation technologies, such as virtual machines and containers, create abstraction layers over computer hardware, to divide a single computer—processors, memory, storage, network, etc. — into multiple smaller virtual units.

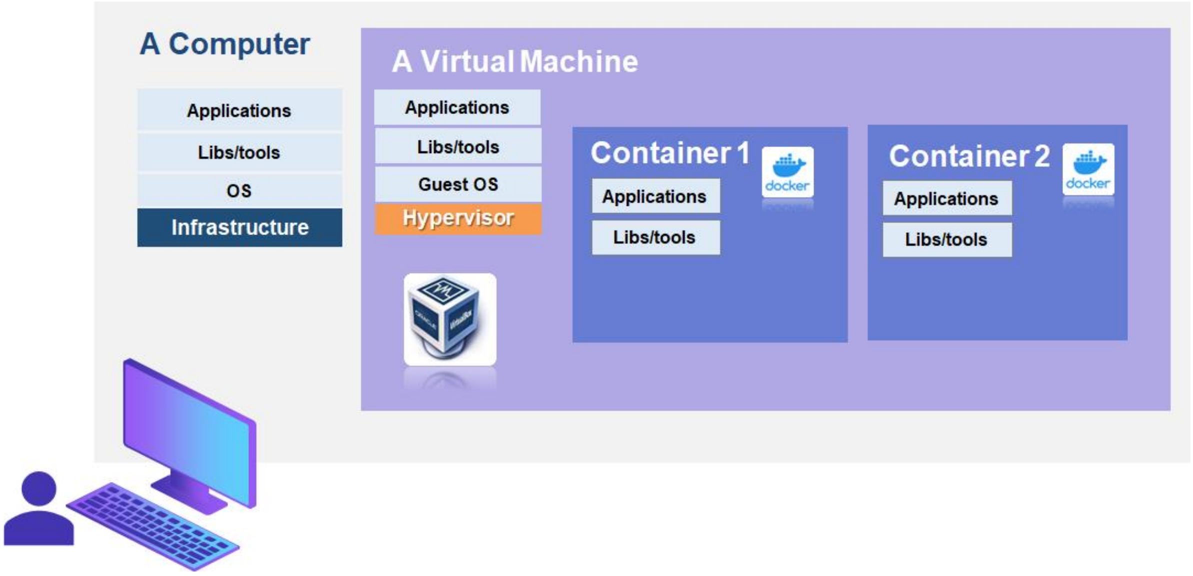
Through this lab session, students will explore virtual machines and docker containers by installing and running them on a local machine and learn how to develop and run applications on top of them.

Topics covered

This lab will cover two parts:

- Part 1, to install VirtualBox software and set up a virtual machine in your local computer and run data analytics; and
- Part 2, to install Docker software in a virtual machine and program the containers to run web server applications.

You will create a system architecture with two containers and a virtual machine on a local machine, as shown below.



A high-level view of the lab experimental environment

Learning Objectives

- Exploring the creation of a virtual machine using VirtualBox
- Inspecting and configuring the virtual machine environment
- Transferring files from and to the virtual machine
- Running a Python programme for data analytics
- Exploring creation of a container using Docker in the virtual machine
- Building a Docker image for creating a docker container
- Programming a docker container for a simple web service
- Saving the container image and starting a new container based on it

Duration

This lab takes approximately **100 minutes** to complete.

Part 1: Virtual Machine

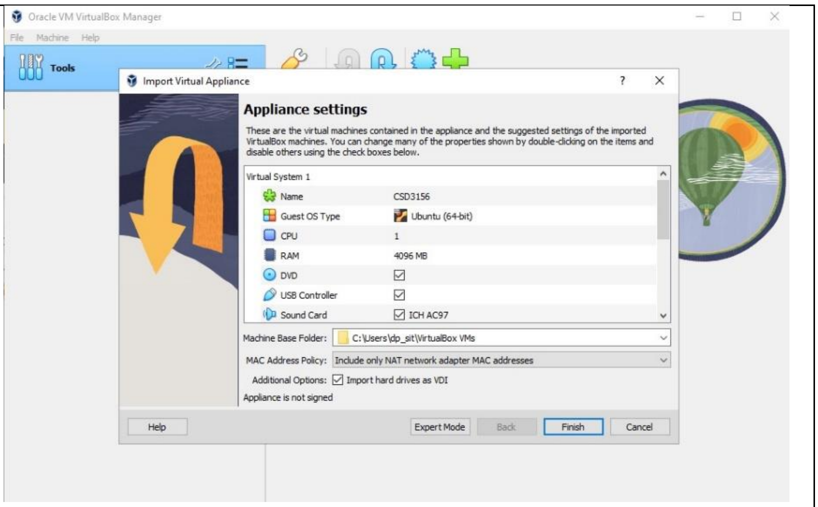
A virtual machine with Ubuntu Operating System (OS) will be created and started in your local machine, and you can program a python program to run data analytics applications.

Task 1: Create a Virtual Machine in a local computer

1. Open Oracle VirtualBox which has been installed in the lab computers.

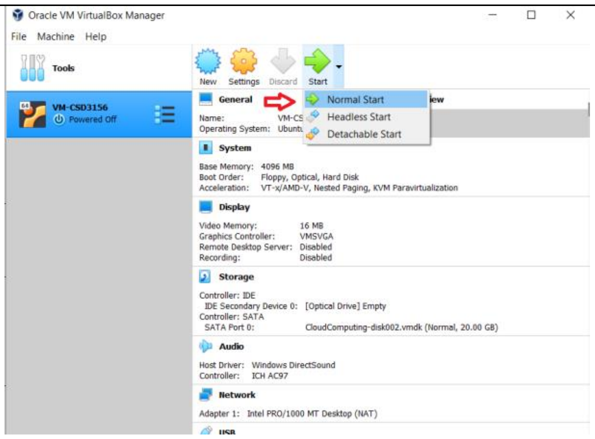
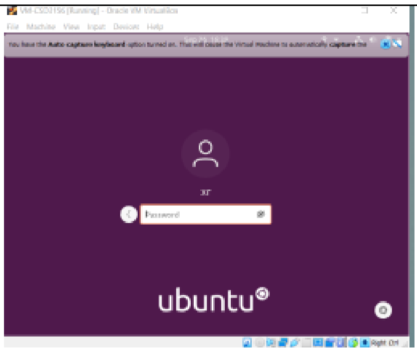
2. Install VirtualBox in your local machine.

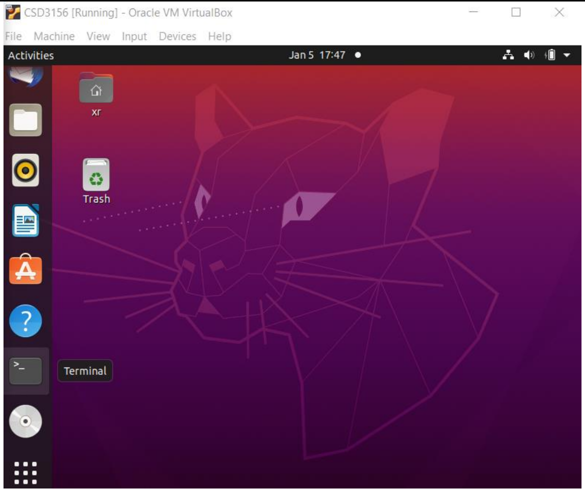
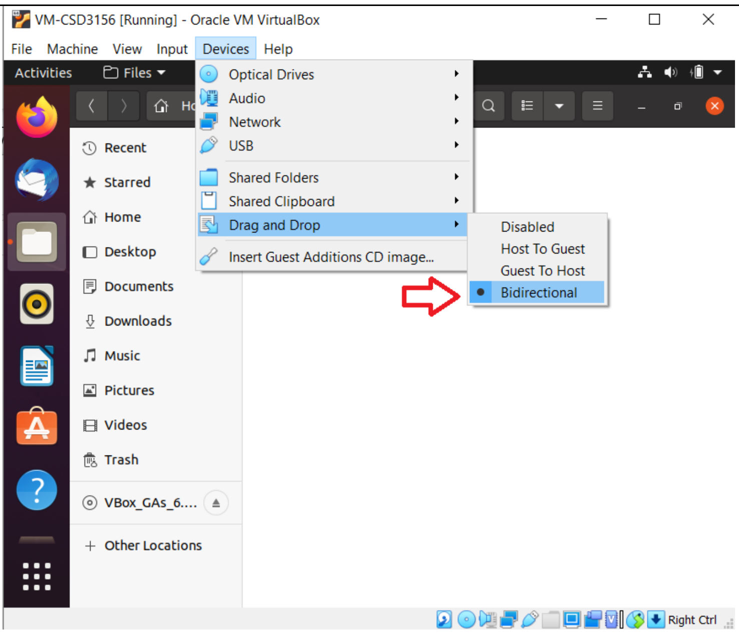
SN	Steps	Instructions
1)	Start a virtual machine from a local virtual machine image	You can find a virtual machine image file **CSD3156.ova** at the desktop machine c:_csd5136\CSD3156.ova, **or** c:\ICT_csd3156\CSD3156.ova. Double click CSD3156.ova. Virtual Box will be started automatically. Click “Finish”, and you will start to create a new VM on the VirtualBox tool.

		 Oracle VM VirtualBox Manager Import Virtual Appliance **Appliance settings** These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below. Virtual System 1 Name: CSD3156 Guest OS Type: Ubuntu (64-bit) CPU: 1 RAM: 4096 MB DVD: ☑ USB Controller: ☑ Sound Card: ☑ ICH AC97 Machine Base Folder: C:\Users\dp_sit\VirtualBox VMs MAC Address Policy: Include only NAT network adapter MAC addresses Additional Options: ☑ Import hard drives as VDI Appliance is not signed

Task 2: Setup the Virtual Machine

3. Start the virtual machine.

SN	Steps	Instructions
1)	Start the virtual machine	 Oracle VM VirtualBox Manager VM-CSD3156 Powered Off General Name: VM-CS Operating System: Ubuntu System Base Memory: 4096 MB Boot Order: Floppy, Optical, Hard Disk Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization Display Video Memory: 16 MB Graphics Controller: VMSVGA Remote Desktop Server: Disabled Recording: Disabled Storage Controller: IDE IDE Secondary Device 0: [Optical Drive] Empty Controller: SATA SATA Port 0: CloudComputing-disk002.vmdk (Normal, 20.00 GB) Audio Host Driver: Windows DirectSound Controller: ICH AC97 Network Adapter 1: Intel PRO/1000 MT Desktop (NAT) USB File Machine Help Tools New Settings Discard Start Normal Start Headless Start Detachable Start It may take a few minutes for the VM to start.
2)	Log into the virtual machine **Username:** xr **Password:** testtest	 VM-CSD3156 [Running] - Oracle VM VirtualBox You have the **Auto capture keyboard** option turned on. This will cause the Virtual Machine to automatically capture the xr File Machine View Input Devices Help Password ubuntu Right Ctrl

3)	Start a terminal and run command lines	 CSD3156 [Running] - Oracle VM VirtualBox File Machine View Input Devices Help Activities Jan 5 17:47 xr Trash Terminal You will have a new VM with Ubuntu OS. Run command lines for testing such as `$hostname -I`
4)	Enable drag & drop for file transfer to/from the virtual machine (Optional)	 VM-CSD3156 [Running] - Oracle VM VirtualBox File Machine View Input Devices Help Activities Files Optical Drives Audio Network USB Shared Folders Shared Clipboard Drag and Drop Insert Guest Additions CD image... Disabled Host To Guest Guest To Host ● Bidirectional Recent Starred Home Desktop Documents Downloads Music Pictures Videos Trash VBox_GAs_6.... Other Locations Right Ctrl You can try to drag and drop a file from local machine to the folder in the virtual machine.

Task 3: Run a python data analytics program in the VM

4. Install python and libraries.

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
```

Enter password: testtest

Deadsnakes PPA provides newer releases over the default Ubuntu repositories. Press “enter” to continue.

```
$ sudo apt update
```

```
$ sudo apt install python3.10
```

Install python 3.10

```
$ python3.10 --version
```

Check the version

Trouble shooting:

If there are system errors like “lock /var/lib/dpkg/lock” or “cache lock”, try the following command lines:

```
$sudo rm /var/lib/dpkg/lock
$sudo rm /var/lib/apt/lists/lock
$sudo rm /var/cache/apt/archives/lock
```

If it still does not work, try the following:

```
$sudo rm /var/lib/dpkg/lock-frontend
```

Install Pandas and NumPy libraries.

```
$ sudo apt install python3-pandas
```

```
$ sudo apt install python3-numpy
```

5. Go to the `/home/xr/test/Lab1-Part1` in the virtual machine, find data file and python file for the data analytics and run the python file

```
$ cd /home/xr/test/Lab1-Part1
```

```
$ cat data.csv
```

```
$ cat statistics.py
```

```
$ python3 statistics.py
```

Add the screenshot of Step 5 to the lab report.

(Optional) Program a new data analytics application and add the python codes & results to the lab report.

Part 2: Docker Container

Students will explore how to create docker containers in the Virtual Machine with Ubuntu OS created in Part 1, and program a Docker file to start docker containers running multiple web servers concurrently.

You need to be familiar with the following command lines for docker operations.

SN	Operation	Command line
1	Build a docker image	`$sudo docker build . -t <image_name:tag_name>`

2	Check the docker images	List all the docker images
		`$sudo docker images`
		Return the docker images ID
		`$sudo docker images -q`

3	Remove docker images	Remove a docker image of `<image_name/image_id>`
		`$sudo docker rmi -f <image_name/image_id>`
		Remove all dangling images
		`$sudo docker image prune`
		Remove all container images
		`$sudo docker rmi $(sudo docker images -q)`

4	Create & run a container	`$sudo docker run --name <container_name> <image_name:tag_name>`

5	Check the containers	List all the containers
		`$sudo docker ps -a`
		List the active containers
		`$sudo docker ps`

6	Interact with a container	`$sudo docker exec -it <container_name> /bin/bash`

7	Update a container image	`$sudo docker commit <container_name> <image_name:tag_name>`

8	Transfer a file from/to the host to/from a container	`$sudo docker cp <local_path> <container_name>:<container_path>`
		`$sudo docker cp <container_name>:<container_path> <local_path>`

9	Start/stop a container	`$sudo docker start <container_name>`
		`$sudo docker stop <container_name>`

10	Remove containers	Remove a container of `<container_name>`
		`$sudo docker rm <container_name>`
		Remove all inactive containers
		`$sudo docker rm $(sudo docker ps --filter status=exited -q)`

Task 1: Install Docker on Ubuntu in a VM

6. Check the version of the docker installed in the Ubuntu VM

```
$docker --version
```

Check if Docker works

```
$sudo systemctl status docker
```

If docker is not active, then enable it

```
$sudo systemctl enable --now docker
```

Clean up the existing docker containers and images

```
$sudo docker rm $(sudo docker ps --filter status=exited -q)
```

Test if docker is connecting with the Docker hub by running hello-world

```
$sudo docker run hello-world
```

If successfully, you should be able to see "*Hello from Docker!*"

Task 2: Program a Dockerfile for httpd web server

7. Go to folder `/home/xr/test/Lab1-Part2`

```
$cd /home/xr/test/Lab1-Part2/dockerfile
```

```
$cd httpd
```

```
$cat index.html
```

You should be able to see the following content in the index.html.

```
<!DOCTYPE html>

<html>

<body>

<h1>Hello, world!</h1>

<p class="subtitle">This is an example web page from Web Server 1.</p>

</body>

</html>
```

Or you can use Vim command to create your own web page index.html file (Optional).

VIM commands:

- Press “i” to change to insert mode
- Press “esc”, then enter “:w” to save the file without exiting vim
- Press “esc”, then enter “:wq” to save the file and exit vim

8. Build a docker image using the Dockerfile for a httpd web server.

```
$cat Dockerfile
```

```
# Getting the base image
FROM httpd:latest

#Copy webpage file to webserver
COPY index.html htdocs/.
```

Create an image using the Dockerfile with a name “myhttpd” and tag “v1”

```
$sudo docker build . -t myhttpd:v1
```

Check the images created

$ ______________________

Please fill in the blank with a Linux Docker command to check the Docker images that have been created.

9. Start the container by running the container image (named “myhttpd” with tag “v1”) in a detached mode and name the container as “web1”

```
$sudo docker run -d --name web1 myhttpd:v1
```

10. Check the webpage from web server 1.

Check the process running
$______________________

Please fill in the blank with a Linux Docker command to check the Docker processes that have been started.

Add the above command to Result 2 in lab report.

If the container named “web1” is active, check its ip address

```
$sudo docker inspect web1|grep “IPAddress”
```

Get the <Ip address> of container web1 and open a web browser and open the url http://<Ip_address> (**Please note that the url is not “https”**)

Add the screenshot of the web browser to Result 2 in lab report.

Task 3: Start another web server container based on a same image

11. Please fill the blank to start another container by running the container image named “myhttpd” with tag “v1” in a detached mode and name the new container as “web2”.

$____

Please note how fast to start a new container based on a docker image comparing with starting a vm.

Add the above command to Result 3 in lab report.

12. Check the webpage from web server 2.

Check the process running. If the container named “web1” is active, check its ip address

```
$sudo docker inspect web2|grep “IPAddress”
```

Get the <Ip address> of container web2 and open a web browser and open the url http://<Ip_address>

13. Interact with the container web2

```
$sudo docker exec -it web2 /bin/bash
```

Go to htdocs director at the container

```
root@containerID$cd /usr/local/apache2/htdocs
```

Install vim at the container

```
root@containerID$apt-get update
root@containerID$apt-get install vim
root@containerID$vim index.html
```

Change “Web Server 1” into “Web Server 2” in the index.html

```
<!DOCTYPE html>

<html>

<body>

<h1>Hello, world!</h1>

<p class="subtitle">This is an example web page from Web Server 2.</p>

</body>

</html>
```

Refresh the webpage with the ip_address of web2 container
http://<Ip_address>

Troubleshooting: If it is still not changed to Web Server 2, run

```
$sudo docker stop web2
$sudo docker start web2
```

Exit from the container

```
root@containerID$exit
```

Task 4: Save the updated container image

14. Save the updated web2 container into a new image with `<image_name:tag>`

```
$ sudo docker commit web2 myhttpd:v2
```

check the Docker images, and there will be one new image of *myhttpd* with a tag *v2* created.

15. Run a new container called web_new using the new image

```
$sudo docker run -d --name web_new myhttpd:v2
```

Check the process running. If the container named "web_new" is active, check its ip address

```
$sudo docker inspect web_new|grep "IPAddress"
```

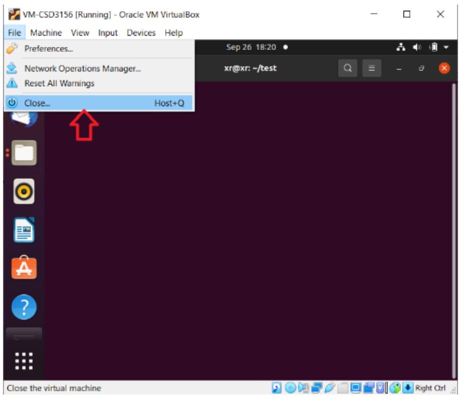
Get <Ip_address > of the web_new container and open the url http://<Ip_address>

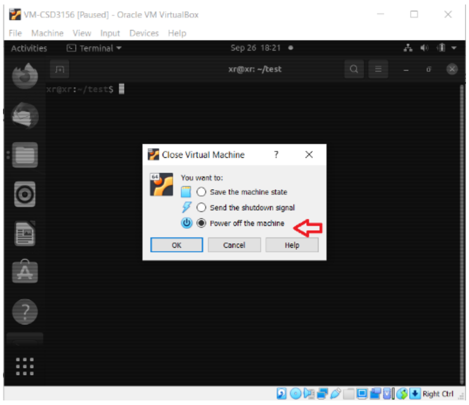
You can find that the index.html has "web server 2". It is because the index.html file in web2 server has been successfully saved into the updated container image *myhttpd* with a tag *v2*.

Add the screenshot of the web browser to Result 3 in the lab report.

Task 5: Stop the vm

16. Stop the virtual machine by select the File->Close and choose Power off the machine.





```
xr@xr:~/test$
```

Close Virtual Machine

You want to:

- ○ Save the machine state
- ○ Send the shutdown signal
- ● Power off the machine

Add one or two sentences to summarize what you have learned from Lab 5 into the lab report.

Lab complete

Congratulations! You have completed the lab.

Please submit the lab report to the x-site by the end of 3 Mar 2026 (Tuesday).

Thank you!