STUDENT NAME: Adam Kenneth Shibaba

STUDENT ID: C00242437

PROJECT TITLE: Virtual Docker Testbed for Cybersecurity

SUPERVISOR: James Egan

SUBMISSION DATE: 26/04/2022

Abstract

The aim of this project is to provide an understanding of virtualization, how it facilitates IT operations and among several points, how it can be resourceful for online learning. The components that were used for this task include a cloud server, and docker containers to facilitate interaction between users and VMs, a database, and frontend application. The creation of this system is aimed at improving virtual lab exercises.

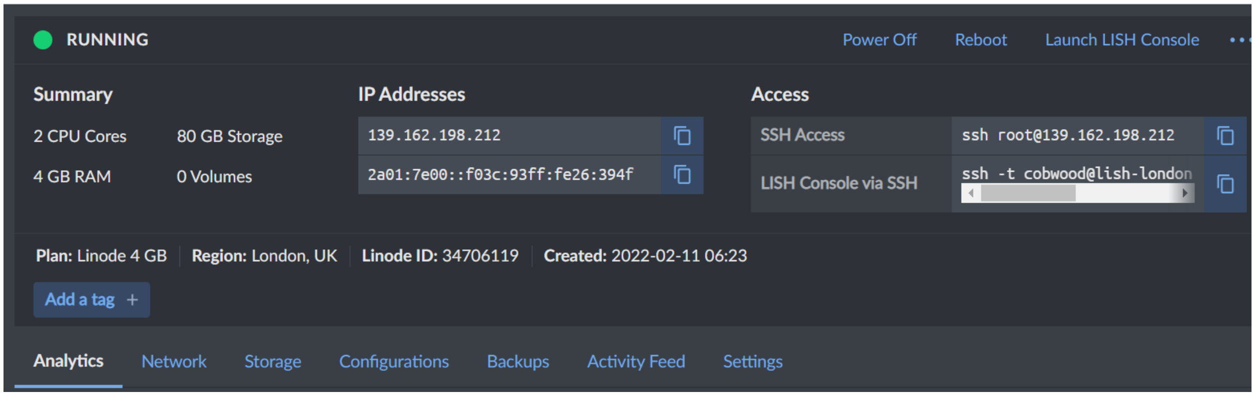
Contents

Introduction

This document describes the complete creation of a virtual docker testbed environment involving the use of containerization technology. It would aid in providing a comprehensive understanding how a virtual system would operate. It also includes how I was able to overcome challenges encountered, such as what worked, what did not, modifications and importantly, testing.

Project Description

After spending some good time seeking a way to start this project, I felt the first thing to begin with is where do I contain the services required to build the environment? And with the advances technology has made in virtualization, I decided to use cloud technology due to increased agility and flexibility. This is where Linode became very resourceful. With Linode, I created a server.



RUNNING

Summary		IP Addresses	Access	
2 CPU Cores	80 GB Storage	139.162.198.212	SSH Access	ssh root@139.162.198.212
4 GB RAM	0 Volumes	2a01:7e00::f03c:93ff:fe26:394f	LISH Console via SSH	ssh -t cobwood@lish-london

Plan: Linode 4 GB | **Region:** London, UK | **Linode ID:** 34706119 | **Created:** 2022-02-11 06:23

Services

- MySQL
- Php
- Docker
- Httpd
- Ssh

How to Access the Application

139.162.198.212

Paste the following IP Address into your browser, on the login page, enter the following student or tutor’s credentials:

Student

Username =	student001@test.com
Password =	Testbed

Tutor

Username =	superadmin@site.com
Password =	Pass@123

To Access the SQL Database, use the following URL: 139.162.198.212/phpmyadmin

And use the following credentials

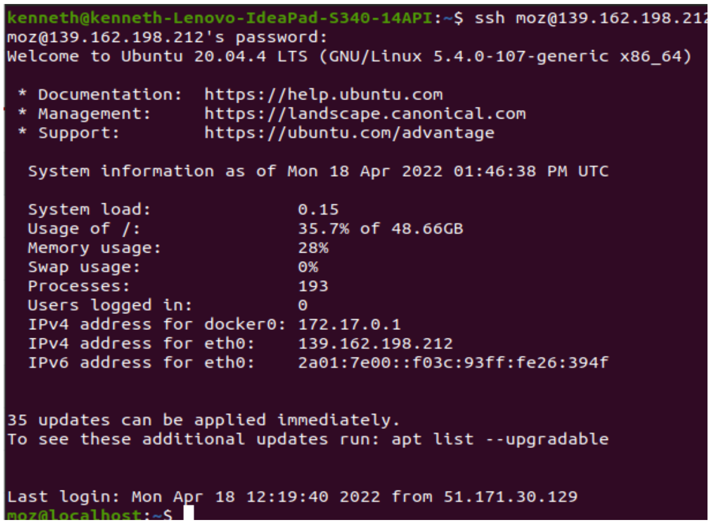
Username:	root
Password:	123

Chosen Distribution

Choosing a preferred distro to deploy is one of the initial steps I took in deploying this Instance. Linux OS was what I used for this task. Linode provided me with Linux distributions to choose from. This option allowed me to begin with a reliable Linux operating system including the creation of my own software stack from the ground up. For this project I selected the latest LTS release of Ubuntu (20.04 LTS currently).

Remote Access to Server

After creating the Linode server, the next step was to log in via ssh to start deploying the required services such as MySQL, Apache webserver, php and most importantly, docker. By using SSH to remotely log into the server I was able to install and configure a firewall, set time zone, set a limited user and encrypt SSH credentials to avoid unwanted access.
Appending SSH keys to my root user account allowed log in via ssh without a password possible. Keys include a private key which is stored in my local device and the public key which I uploaded into the server with scp (Secure copy protocol)



```
kenneth@kenneth-Lenovo-IdeaPad-S340-14API:~$ ssh moz@139.162.198.21
moz@139.162.198.212's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 18 Apr 2022 01:46:38 PM UTC

  System load:              0.15
  Usage of /:               35.7% of 48.66GB
  Memory usage:             28%
  Swap usage:               0%
  Processes:                193
  Users logged in:          0
  IPv4 address for docker0: 172.17.0.1
  IPv4 address for eth0:    139.162.198.212
  IPv6 address for eth0:    2a01:7e00::f03c:93ff:fe26:394f


35 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Mon Apr 18 12:19:40 2022 from 51.171.30.129
moz@localhost:~$ 
```

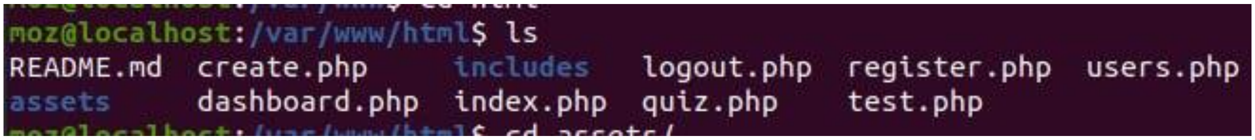
Shared CPU

I selected Shared CPU Compute Instance because it offers affordable virtual machines. It offered me a well-balanced set of services that are suitable for a variety of uses. Somethings that were beneficial about the Shared CPU included these specifications to its available plan

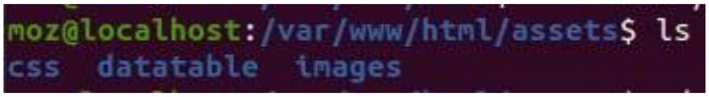
- Shared vCPU cores
- 100% SSD (Solid State Disk) storage
- 40 Gbps inbound network bandwidth
- Free inbound network transfer
- Dedicated IPv4 and IPv6 addresses
- Direct console access through Lish

Docker

The files that docker would integrate with are situated in the /var/www/html folder. This is where all php files/source codes that interact with docker are located such as register, logout, users, and quiz. Only the administrator would be able to access and modify these files or any configuration



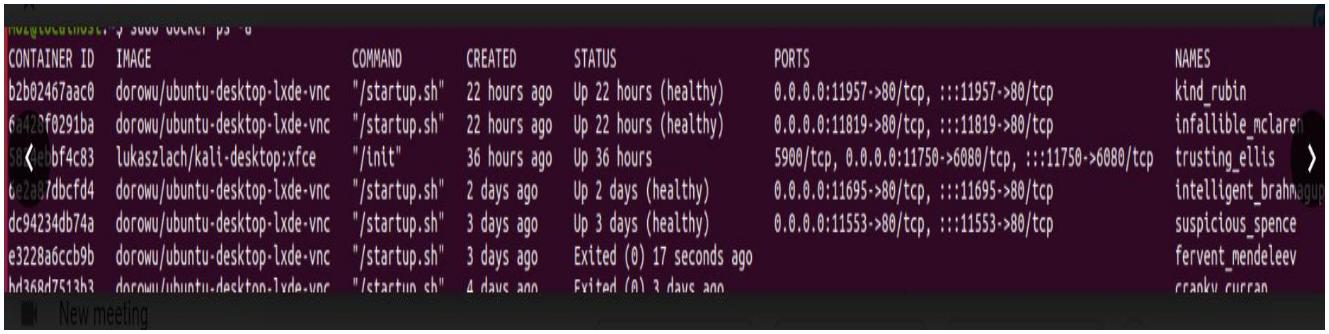
```
moz@localhost:/var/www$ cd html
moz@localhost:/var/www/html$ ls
README.md  create.php     includes   logout.php  register.php  users.php
assets     dashboard.php  index.php  quiz.php    test.php
moz@localhost:/var/www/html$ cd assets/
```



```
moz@localhost:/var/www/html/assets$ ls
css  datatable  images
```

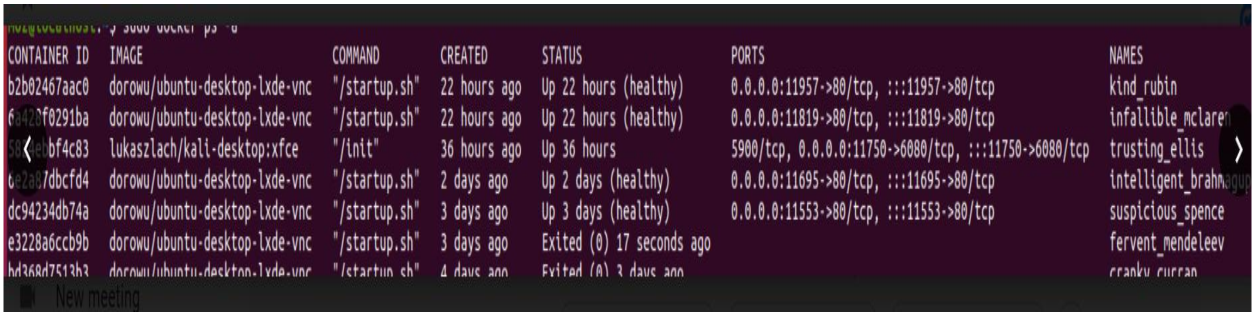
Containers

As mentioned only the administrator or person with the correct privileges would be able to view containers. Some of the information available would include a container ID, image, period created status, ports, and name. In the image below you can see how this is displayed.



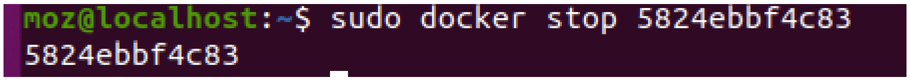
```
moz@localhost:~$ sudo docker ps -a
CONTAINER ID   IMAGE                          COMMAND          CREATED         STATUS                        PORTS                                                  NAMES
b2b02467aac0   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   22 hours ago    Up 22 hours (healthy)         0.0.0.0:11957->80/tcp, :::11957->80/tcp                kind_rubin
6a428f0291ba   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   22 hours ago    Up 22 hours (healthy)         0.0.0.0:11819->80/tcp, :::11819->80/tcp                infallible_mclaren
5824ebbf4c83   lukaszlach/kali-desktop:xfce   "/init"          36 hours ago    Up 36 hours                   5900/tcp, 0.0.0.0:11750->6080/tcp, :::11750->6080/tcp  trusting_ellis
6e2a87dbcfd4   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   2 days ago      Up 2 days (healthy)           0.0.0.0:11695->80/tcp, :::11695->80/tcp                intelligent_brahmagup
dc94234db74a   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   3 days ago      Up 3 days (healthy)           0.0.0.0:11553->80/tcp, :::11553->80/tcp                suspicious_spence
e3228a6ccb9b   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   3 days ago      Exited (0) 17 seconds ago                                                            fervent_mendeleev
bd368d7513b3   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   4 days ago      Exited (0) 3 days ago                                                                cranky_curran
```

In order to ensure and avoid resources such as CPU and the storage from being exhausted, administrators most especially would have to remove containers as soon as a class is over. This means tutors would have to delete the vm which triggered the container. Though this would require tutors to perform such action on the frontend, the same would apply to the backend. This way, resources of the server would be sufficient for upcoming tasks. Se images below



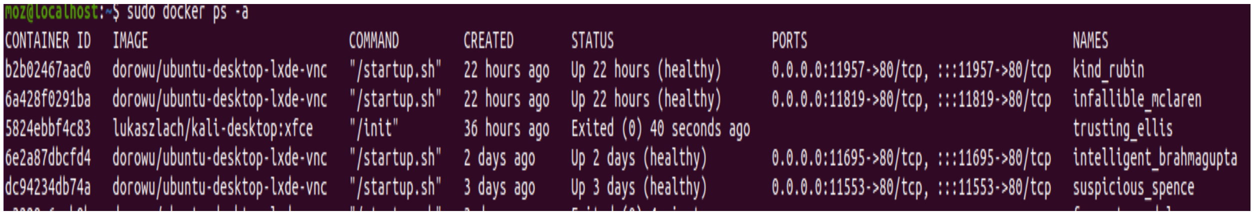
```
moz@localhost:~$ sudo docker ps -a
CONTAINER ID   IMAGE                          COMMAND          CREATED         STATUS                        PORTS                                                  NAMES
b2b02467aac0   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   22 hours ago    Up 22 hours (healthy)         0.0.0.0:11957->80/tcp, :::11957->80/tcp                kind_rubin
6a428f0291ba   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   22 hours ago    Up 22 hours (healthy)         0.0.0.0:11819->80/tcp, :::11819->80/tcp                infallible_mclaren
5824ebbf4c83   lukaszlach/kali-desktop:xfce   "/init"          36 hours ago    Up 36 hours                   5900/tcp, 0.0.0.0:11750->6080/tcp, :::11750->6080/tcp  trusting_ellis
6e2a87dbcfd4   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   2 days ago      Up 2 days (healthy)           0.0.0.0:11695->80/tcp, :::11695->80/tcp                intelligent_brahmagup
dc94234db74a   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   3 days ago      Up 3 days (healthy)           0.0.0.0:11553->80/tcp, :::11553->80/tcp                suspicious_spence
e3228a6ccb9b   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   3 days ago      Exited (0) 17 seconds ago                                                            fervent_mendeleev
bd368d7513b3   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"   4 days ago      Exited (0) 3 days ago                                                                cranky_curran
```

The image below shows the command to stop the container with the image ID



```
moz@localhost:~$ sudo docker stop 5824ebbf4c83
5824ebbf4c83
```

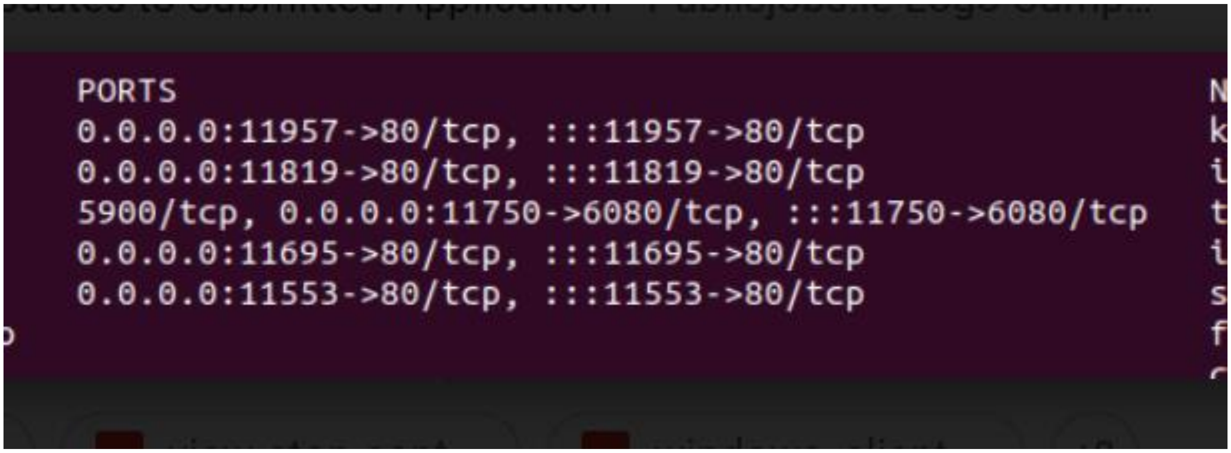
After completing the stop process, I can see in the image that the container has been stopped which is in the port section, on line three, there is no port number



```
moz@localhost:~$ sudo docker ps -a
CONTAINER ID   IMAGE                          COMMAND          CREATED        STATUS                    PORTS                                        NAMES
b2b02467aac0   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"    22 hours ago   Up 22 hours (healthy)     0.0.0.0:11957->80/tcp, :::11957->80/tcp     kind_rubin
6a428f0291ba   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"    22 hours ago   Up 22 hours (healthy)     0.0.0.0:11819->80/tcp, :::11819->80/tcp     infallible_mclaren
5824ebbf4c83   lukaszlach/kali-desktop:xfce   "/init"          36 hours ago   Exited (0) 40 seconds ago                                              trusting_ellis
6e2a87dbcfd4   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"    2 days ago     Up 2 days (healthy)       0.0.0.0:11695->80/tcp, :::11695->80/tcp     intelligent_brahmagupta
dc94234db74a   dorowu/ubuntu-desktop-lxde-vnc "/startup.sh"    3 days ago     Up 3 days (healthy)       0.0.0.0:11553->80/tcp, :::11553->80/tcp     suspicious_spence
```

Ports

When containers are triggered, they are provided with a port number which is how they would be communicated with by users from the front end. This potential of docker helps to promote isolation. I noticed that every time I created a VM for example kali Linux, the container it sits in is created and the ip address is assigned a port number. The importance of the port ensures the performance and speed of the VM is not disturbed. Also this promotes a great level of security.



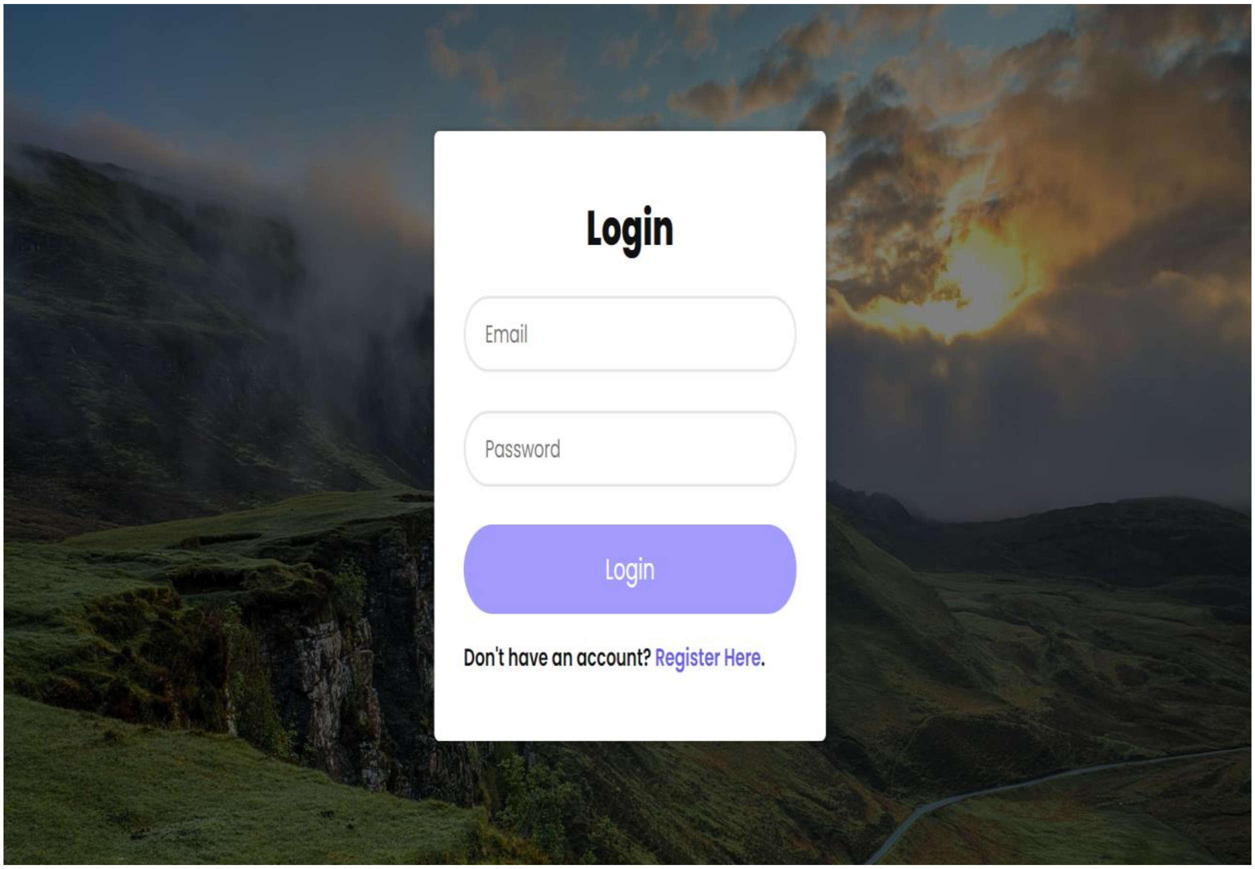
```
PORTS                                                     N
0.0.0.0:11957->80/tcp, :::11957->80/tcp                   k
0.0.0.0:11819->80/tcp, :::11819->80/tcp                   i
5900/tcp, 0.0.0.0:11750->6080/tcp, :::11750->6080/tcp     t
0.0.0.0:11695->80/tcp, :::11695->80/tcp                   i
0.0.0.0:11553->80/tcp, :::11553->80/tcp                   s
                                                          f
```

Web Application process

The image below is a depiction of the user login interface to access containers holding the required Linux os vm for the particular module he/she would be focusing on. The application has been designed to ensure operations in the application is based on privileges.

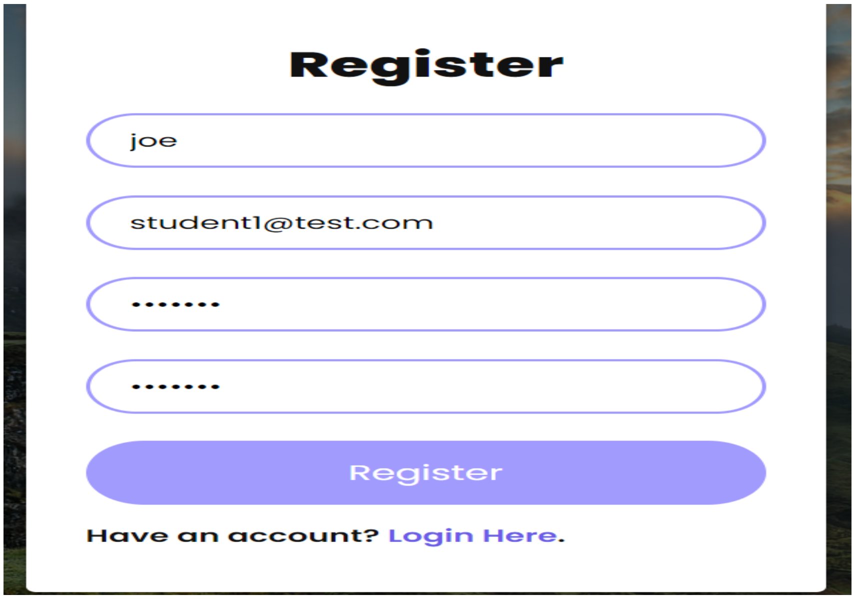
It illustrates how the software might be used by students to complete specific tutorials/lab activities. Tutors, and students would be the people allowed to access and use the application. The process would require students to register with the application in order to participate in lab activities. Registration would require students to meet up with several authentication conditions such as password type containing letters, numbers, and symbols. To get access to the page, a given link would be provided for students by tutors.

Students and tutors are welcomed with the login page



Students

Login process involves sign in or registration. A database stores all information entered by students regardless of the action. The app was designed with policies to ensure students who do not have an account must register to access functions of the application as well as meet the registration requirements such as password criteria. Once registration is complete and successful, students are now presented with a login page.



Register

joe

student1@test.com

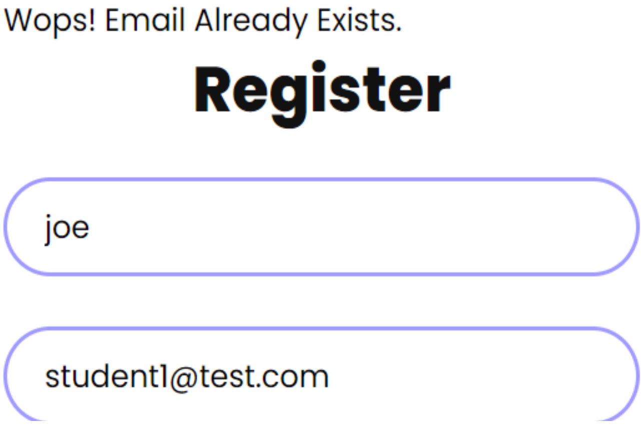
•••••••

•••••••

Have an account? Login Here.

Active users

The app notifies students with an existing account about its presence if they try registering again.



Wops! Email Already Exists.

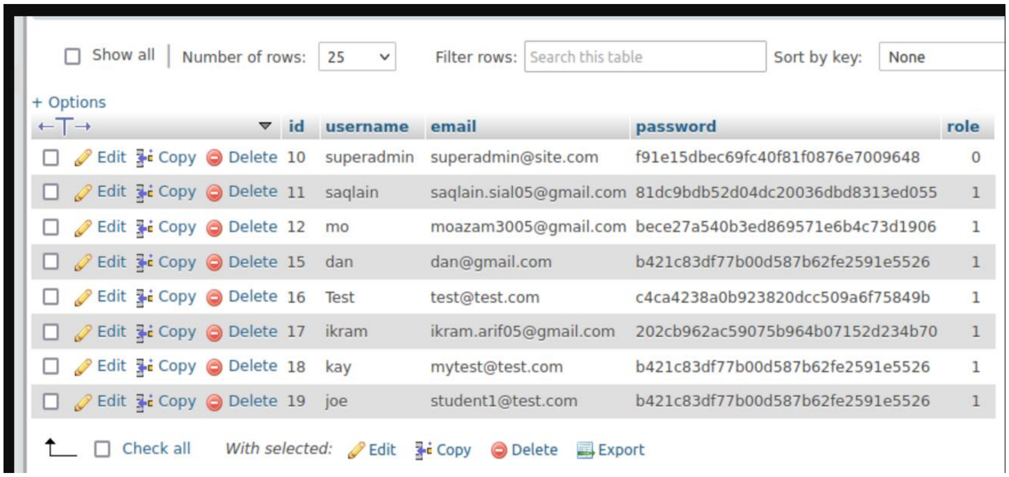
Register

joe

student1@test.com

User Table

After completing the registration or login process, user credentials are then added to the database. This way all login details are stored in the user table. Some of the fields include username, email, password, and role. The app contains hashing to ensure security of passwords.



□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

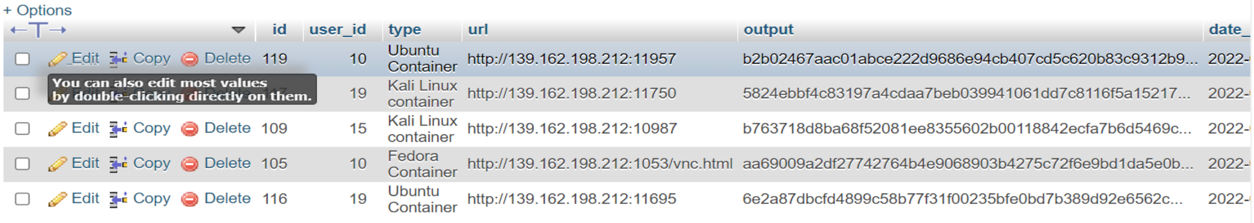
+ Options

		id	username	email	password	role
□	Edit Copy Delete	10	superadmin	superadmin@site.com	f91e15dbec69fc40f81f0876e7009648	0
□	Edit Copy Delete	11	saqlain	saqlain.sial05@gmail.com	81dc9bdb52d04dc20036dbd8313ed055	1
□	Edit Copy Delete	12	mo	moazam3005@gmail.com	bece27a540b3ed869571e6b4c73d1906	1
□	Edit Copy Delete	15	dan	dan@gmail.com	b421c83df77b00d587b62fe2591e5526	1
□	Edit Copy Delete	16	Test	test@test.com	c4ca4238a0b923820dcc509a6f75849b	1
□	Edit Copy Delete	17	ikram	ikram.arif05@gmail.com	202cb962ac59075b964b07152d234b70	1
□	Edit Copy Delete	18	kay	mytest@test.com	b421c83df77b00d587b62fe2591e5526	1
□	Edit Copy Delete	19	joe	student1@test.com	b421c83df77b00d587b62fe2591e5526	1

□ Check all *With selected:* Edit Copy Delete Export

Container Table

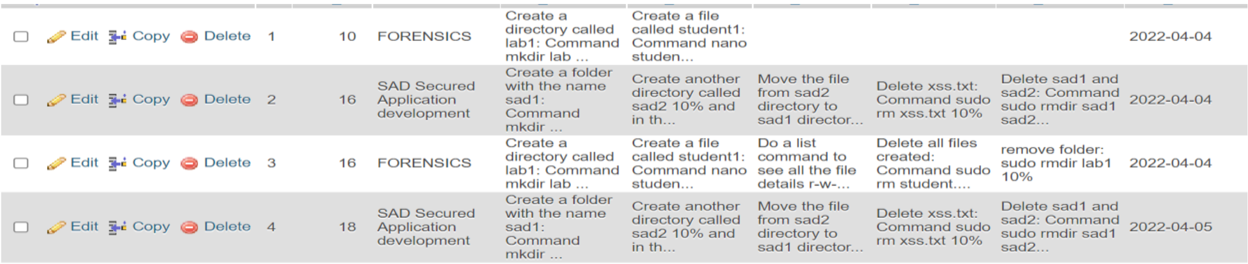
Every container that is entered on by the web application is stored in a database. Same applies to the quiz which can be viewed in the image below.



+ Options

		id	user_id	type	url	output	date_
□	Edit Copy Delete	119	10	Ubuntu Container	http://139.162.198.212:11957	b2b02467aac01abce222d9686e94cb407cd5c620b83c9312b9...	2022-
□	You can also edit most values by double-clicking directly on them.		19	Kali Linux container	http://139.162.198.212:11750	5824ebbf4c83197a4cdaa7beb039941061dd7c8116f5a15217...	2022-
□	Edit Copy Delete	109	15	Kali Linux container	http://139.162.198.212:10987	b763718d8ba68f52081ee8355602b00118842ecfa7b6d5469c...	2022-
□	Edit Copy Delete	105	10	Fedora Container	http://139.162.198.212:1053/vnc.html	aa69009a2df27742764b4e9068903b4275c72f6e9bd1da5e0b...	2022-
□	Edit Copy Delete	116	19	Ubuntu Container	http://139.162.198.212:11695	6e2a87dbcfd4899c58b77f31f00235bfe0bd7b389d92e6562c...	2022-

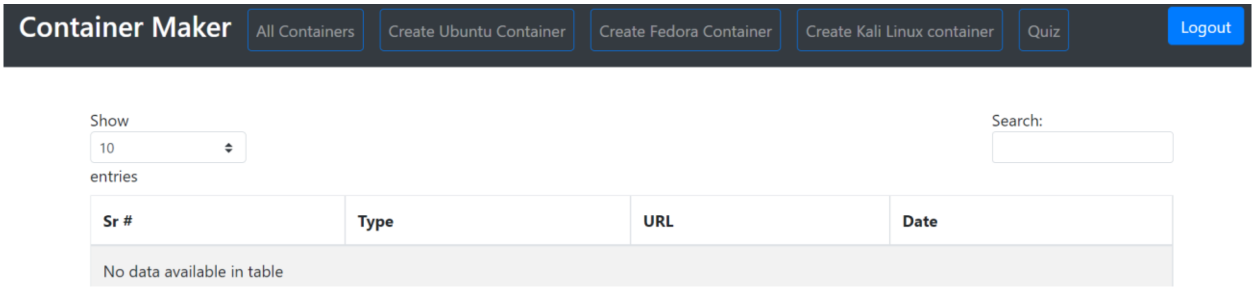
Quiz table



□	Edit Copy Delete	1	10	FORENSICS	Create a directory called lab1: Command mkdir lab ...	Create a file called student1: Command nano studen...				2022-04-04
□	Edit Copy Delete	2	16	SAD Secured Application development	Create a folder with the name sad1: Command mkdir ...	Create another directory called sad2 10% and in th...	Move the file from sad2 directory to sad1 director...	Delete xss.txt: Command sudo rm xss.txt 10%	Delete sad1 and sad2: Command sudo rmdir sad1 sad2...	2022-04-04
□	Edit Copy Delete	3	16	FORENSICS	Create a directory called lab1: Command mkdir lab ...	Create a file called student1: Command nano studen...	Do a list command to see all the file details r-w-...	Delete all files created: Command sudo rm student....	remove folder: sudo rmdir lab1 10%	2022-04-04
□	Edit Copy Delete	4	18	SAD Secured Application development	Create a folder with the name sad1: Command mkdir ...	Create another directory called sad2 10% and in th...	Move the file from sad2 directory to sad1 director...	Delete xss.txt: Command sudo rm xss.txt 10%	Delete sad1 and sad2: Command sudo rmdir sad1 sad2...	2022-04-05

Access containers

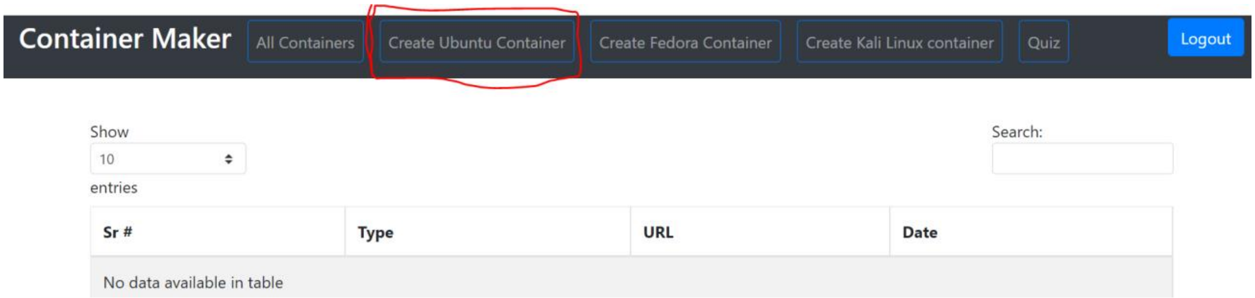
Once login is complete, students are directed to the next page which is where containers hold various Linux VMs. Students would have to select the preferred VM by clicking on the button which displays the Linux distro type. When any of these buttons are clicked on, a php code is executed which triggers the container to launch the VM. With a php plug-in known as php-ssh2 plugin, the code access the server via ssh, and this launches the container holding the VM intended to be used.



Show 10 entries

Search:

Sr #	Type	URL	Date
No data available in table			



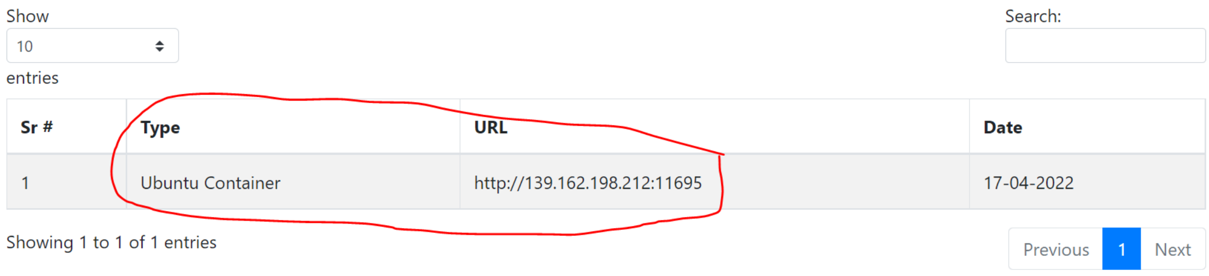
Show 10 entries

Search:

Sr #	Type	URL	Date
No data available in table			

Launching VM

Students would use the ip address that is created with its port number to access the vm. Once this is launched, they can now begin to work with the vm



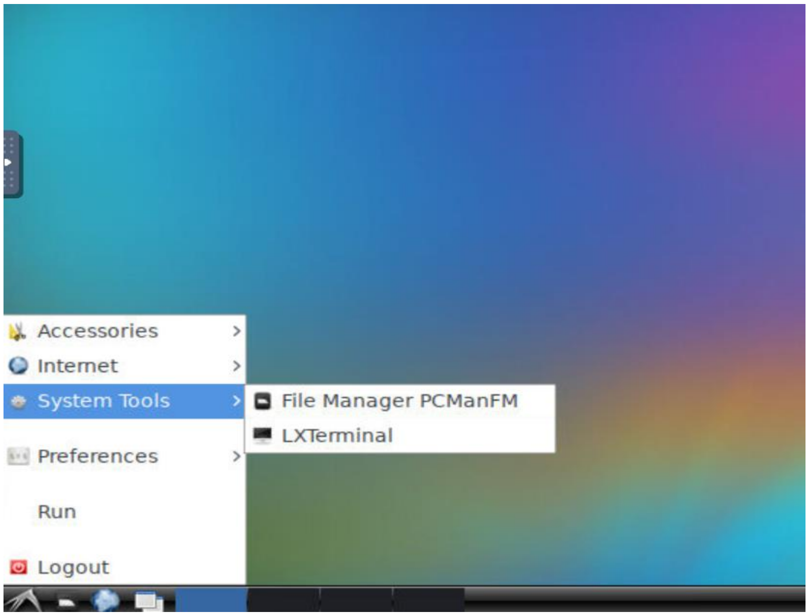
Show 10 entries

Search:

Sr #	Type	URL	Date
1	Ubuntu Container	http://139.162.198.212:11695	17-04-2022

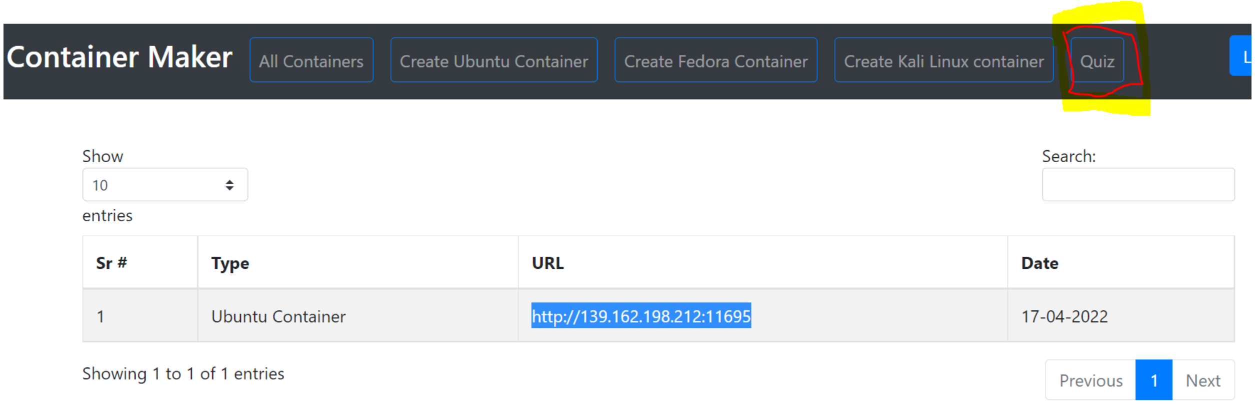
Showing 1 to 1 of 1 entries

The ubuntu VM is now active and running. From this point students can complete their given tasks.



Quiz

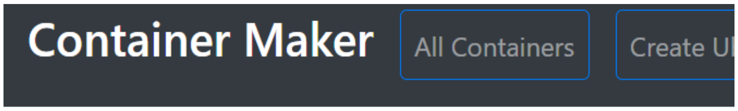
Accessing the quiz would require students to use the quiz button which would direct them to a drop-down list where they can select the module activity they have been given.

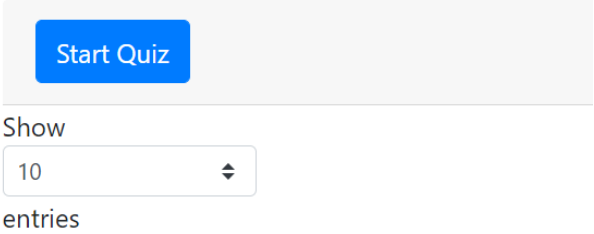


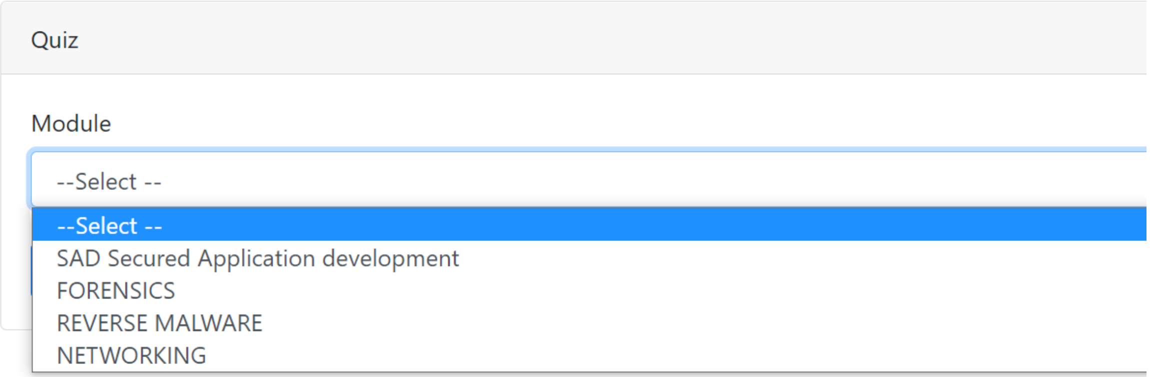
Sr #	Type	URL	Date
1	Ubuntu Container	http://139.162.198.212:11695	17-04-2022

Showing 1 to 1 of 1 entries

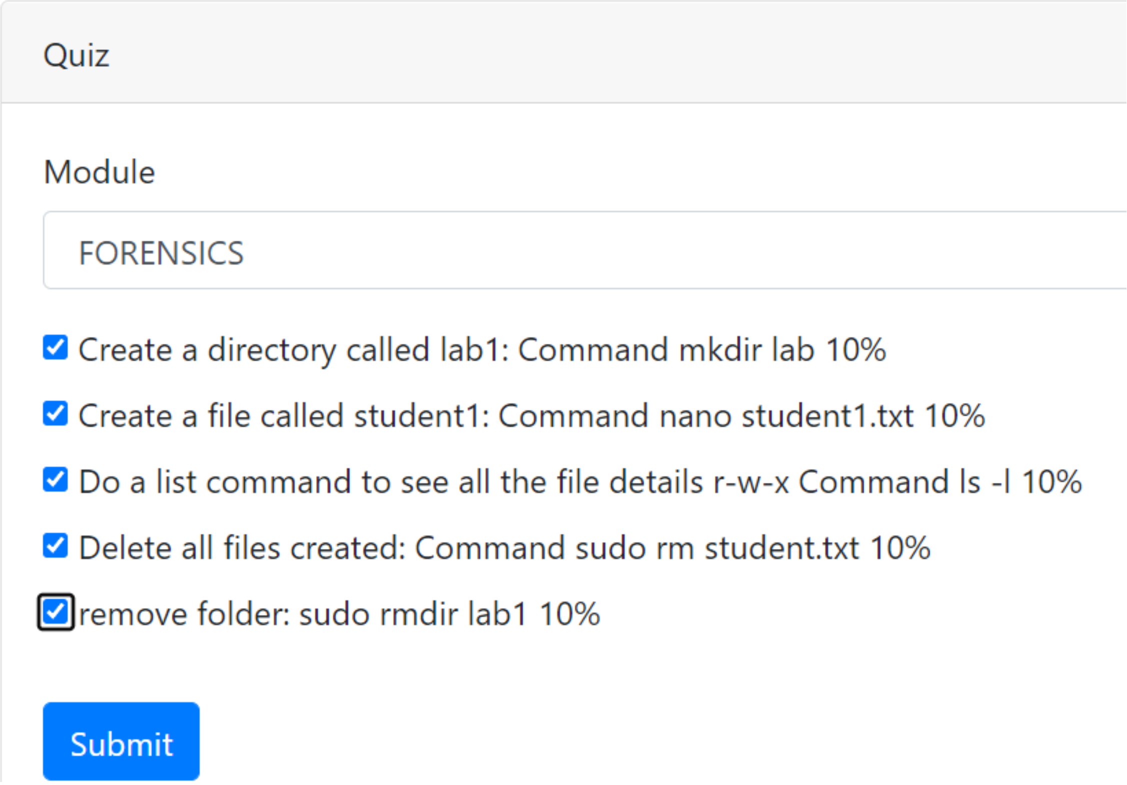
Students can now select the module quiz







The next stage would be for students to begin completing tasks and as soon as a task is complete, they tick each box until the end. This is then followed by clicking the submit button.



Quiz

Module

FORENSICS

☑ Create a directory called lab1: Command mkdir lab 10%

☑ Create a file called student1: Command nano student1.txt 10%

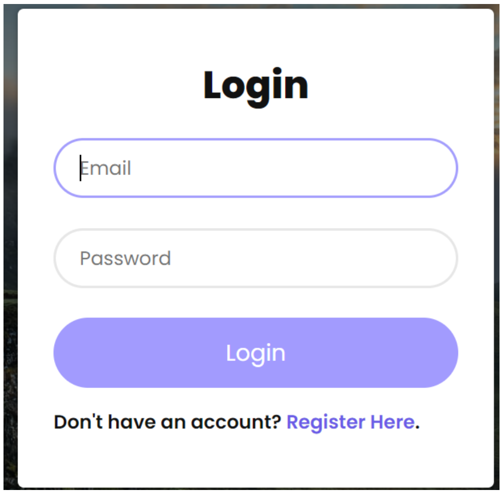
☑ Do a list command to see all the file details r-w-x Command ls -l 10%

☑ Delete all files created: Command sudo rm student.txt 10%

☑ remove folder: sudo rmdir lab1 10%

Tutors

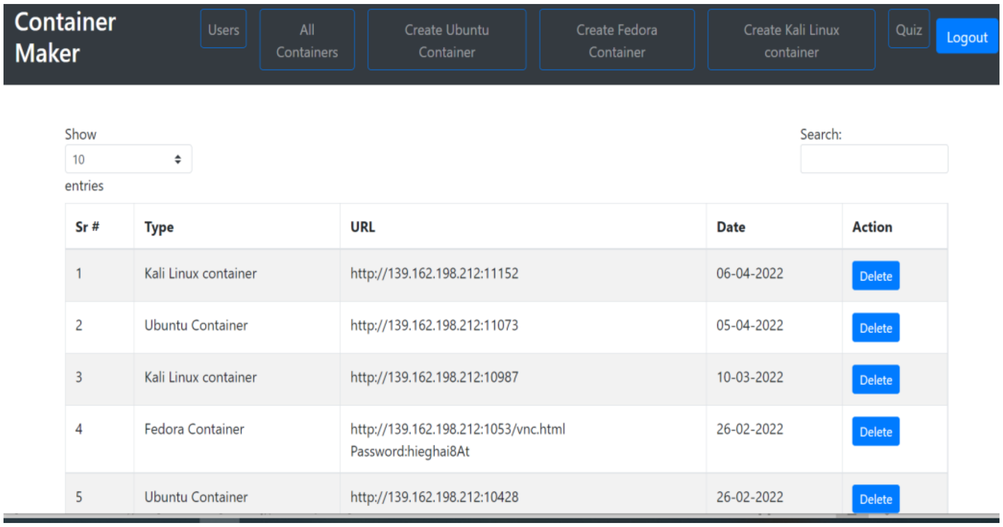
Tutors’ login credentials would be entered into the login section. Registration process would differ from students as tutors would have higher privilege than students. A login username would be given to them by the administrator, and they would create their password.



Don’t have an account? Register Here.

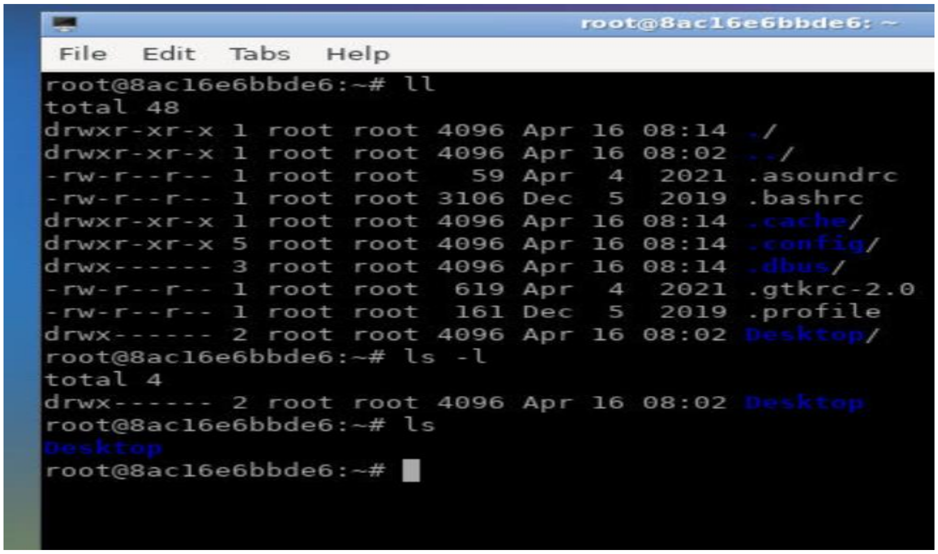
View Containers

As said, tutors would possess more rights than students and this includes able to view, access, examine, and delete all created containers by students including working students. They would also be able to create containers.



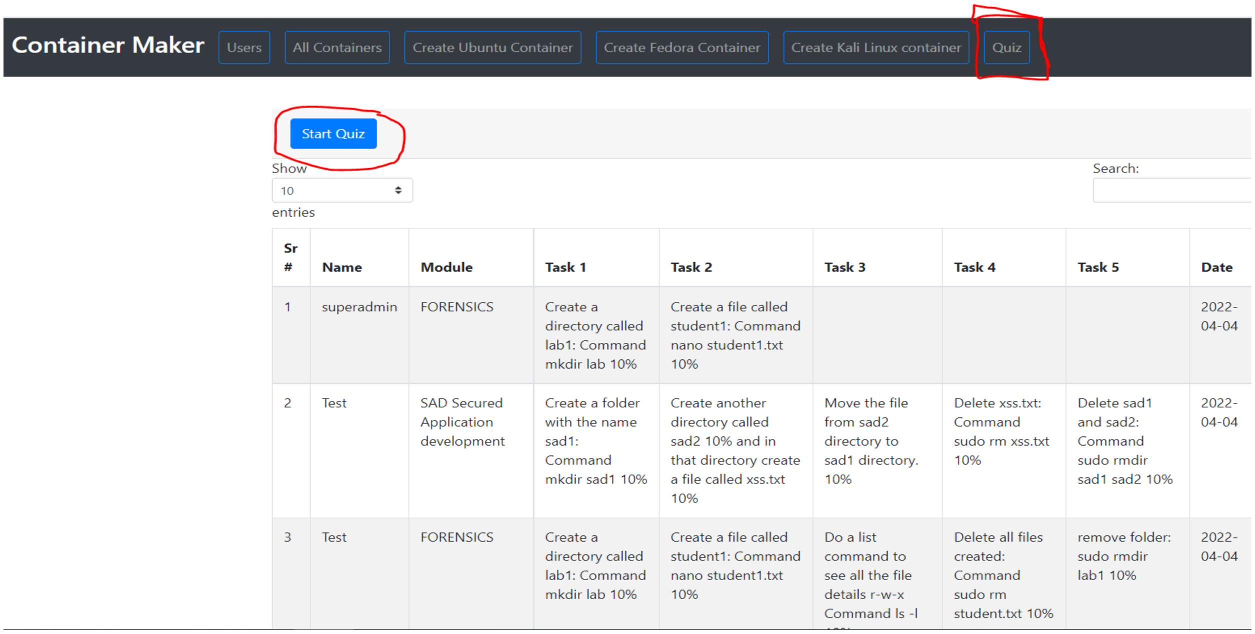
Sr #	Type	URL	Date	Action
1	Kali Linux container	http://139.162.198.212:11152	06-04-2022	Delete
2	Ubuntu Container	http://139.162.198.212:11073	05-04-2022	Delete
3	Kali Linux container	http://139.162.198.212:10987	10-03-2022	Delete
4	Fedora Container	http://139.162.198.212:1053/vnc.html Password:hieghai8At	26-02-2022	Delete
5	Ubuntu Container	http://139.162.198.212:10428	26-02-2022	Delete

The platform would enable tutors determine the progress of given tasks. This way they would be able to grade students based on the work accomplished.



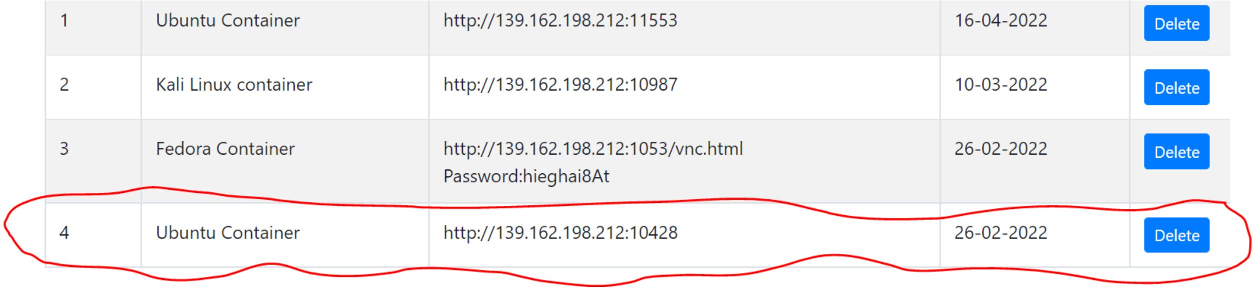
```
root@8ac16e6bbde6:~# ll
total 48
drwxr-xr-x 1 root root 4096 Apr 16 08:14 ./
drwxr-xr-x 1 root root 4096 Apr 16 08:02 ../
-rw-r--r-- 1 root root   59 Apr  4  2021 .asoundrc
-rw-r--r-- 1 root root 3106 Dec  5  2019 .bashrc
drwxr-xr-x 1 root root 4096 Apr 16 08:14 .cache/
drwxr-xr-x 5 root root 4096 Apr 16 08:14 .config/
drwx------ 3 root root 4096 Apr 16 08:14 .dbus/
-rw-r--r-- 1 root root  619 Apr  4  2021 .gtkrc-2.0
-rw-r--r-- 1 root root  161 Dec  5  2019 .profile
drwx------ 2 root root 4096 Apr 16 08:02 Desktop/
root@8ac16e6bbde6:~# ls -l
total 4
drwx------ 2 root root 4096 Apr 16 08:02 Desktop
root@8ac16e6bbde6:~# ls
Desktop
root@8ac16e6bbde6:~# 
```

Among the privileges tutors have would be to view tasks that students have submitted. This part of the application would make tutors responsibility of examining what students have done and grade them accordingly.



Sr #	Name	Module	Task 1	Task 2	Task 3	Task 4	Task 5	Date
1	superadmin	FORENSICS	Create a directory called lab1: Command mkdir lab 10%	Create a file called student1: Command nano student1.txt 10%				2022-04-04
2	Test	SAD Secured Application development	Create a folder with the name sad1: Command mkdir sad1 10%	Create another directory called sad2 10% and in that directory create a file called xss.txt 10%	Move the file from sad2 directory to sad1 directory. 10%	Delete xss.txt: Command sudo rm xss.txt 10%	Delete sad1 and sad2: Command sudo rmdir sad1 sad2 10%	2022-04-04
3	Test	FORENSICS	Create a directory called lab1: Command mkdir lab 10%	Create a file called student1: Command nano student1.txt 10%	Do a list command to see all the file details r-w-x Command ls -l	Delete all files created: Command sudo rm student.txt 10%	remove folder: sudo rmdir lab1 10%	2022-04-04

Once tutor complete their operation, they can remind students to stop running containers and tutors can remove containers. The administrator can also do this at the end of the day in order to prevent the server’s resources such as CPU memory, RAM, network, and storage from being over exhausted.



1	Ubuntu Container	http://139.162.198.212:11553	16-04-2022	Delete
2	Kali Linux container	http://139.162.198.212:10987	10-03-2022	Delete
3	Fedora Container	http://139.162.198.212:1053/vnc.html Password:hieghai8At	26-02-2022	Delete
4	Ubuntu Container	http://139.162.198.212:10428	26-02-2022	Delete

Comparison to Original Design and Specification

The platform this system was supposed to sit on was oracle VirtualBox. However, after observing several setbacks and challenges that began developing, I brought this to the attention of my tutor. I mentioned that a cloud environment would be a preferred option to utilize because it provided greater benefits such as agility and flexibility including resources compared to a virtual box.

Moving ahead into the project, several changes were made. It continued to remain a huge challenge to install docker in virtual box VM and this was due to the fact some dependencies were missing after the installation process and this always made it difficult to progress. One major reason was there wasn't enough memory in the VM which is what made me change to cloud technology. AWS was the initial platform but after doing concrete research I realized from a monetary perspective, it was going to involve high cost.

Further significant progress was faced with an issue. Connecting to a docker container through the web application began displaying a connection failure message. The reason for this was because the CPU threshold was very high. To be specific, over 90%. To overcome this issue, I logged into my Linode server and resized the CPU from 2gb to 4gb and not only was I able to connect but the speed, storage, and performance were excellent.

Also, I tried implementing automation by using a script to facilitate deletion of containers as soon as they are stopped but this task was very adamant to accomplish. After doing extensive research, I came to realize implementing such an idea on docker containers would not be able to be achieved. Since containers are running instances, once they are stopped and removed, a script would not have the reference it's supposed to execute. The solution to this problem was to maintain the initial method which involved remote log in to the server (administrator) and remove the stopped container.

Learning Outcomes

Having gone through such an experience has provided me a great wealth of knowledge. Working on this project exposed me to the potentials of containerization and virtualization technologies. Some the of information acquired during this period have given me a great level of confidence. This project helped me to explore some weaknesses in which I developed some strength from coding languages such as SQL, JavaScript, and php, to cloud technology and most significantly, containerization.

Technology Learning Outcomes

Completing this project gave me a great understanding and discipline in the area of IT administration and security. Some of my weaknesses were explored and manifested into strengths because of the enthusiasm this experience built in me. Research in cloud technology helped to see the importance it holds including benefits. My passion for interacting with CLI/shell is something I truly can’t believe. The potentials of Linux as a principal tool in virtualization has made me realize how important it is. Seeing how a web application interacts with the backend server to spin docker containers was a learning curve. Furthermore, being able to learn how to write and understand coding languages such as php and JavaScript especially when expected.

Personal Outcomes

This experience provided me the skills and mindset that are required when working on projects of this kind. I acquired great reading skills which was something I was not passionate about. During the time of the project, I engaged myself with several learning platforms that I continue utilizing to improve my knowledge in virtualisation and containerization technology which are strong areas of IT operations. Also, it helped me develop a proactive lifestyle. In my workplace, I noticed great improvement in the way I handle tasks (which are around IT administration and Security).

Also, due to such development (proactive), I am able to complete various tasks effectively. A great achievement the process helped me with.