

# VIRTUAL TESTBED

Document: Functional Specification

Student: Adam Kenneth Shibaba

Supervisor: James Egan

23/01/2022

## Abstract

The aim of this document is to highlight the functionality this virtual environment will perform. It will display functions and services that will be made available for the parties involved in managing and operating it such as actors. The information provided would also focus on how interaction would exist between users and the application as well as expected result. Finally, I will use this to mention the required deliverables.

# Contents

- Abstract..... 2
- Introduction ..... 4
  - Overview ..... 4
  - Functionality Specification Scope ..... 4
  - Technologies ..... 4
- Application Users ..... 5
  - Administrators ..... 5
  - Students ..... 5
  - Tutors ..... 5
- Use Case and Scenario ..... 6
  - Login Primary Actor..... 6
    - Preconditions: ..... 6
    - Main Scenario ..... 6
    - Result ..... 8
  - Login Primary Actor..... 9
    - Preconditions: ..... 9
    - Main Scenario ..... 9
    - Result ..... 14
  - Login Primary Actor..... 15
    - Preconditions: ..... 15
    - Main Scenario ..... 15
    - Result ..... 20
- Source codes ..... 21

# Introduction

## Overview

The document is going to focus on the development of a virtual environment. It is my hope to see that modules across Year 3 and 4 cybercrime and IT Security department will be incorporated into virtual machines to facilitate learning for students. The environment will contain efficient virtualization software and automation such as Docker and ansible. Furthermore, the project would include the use of a popular cloud platform to ensure convenient access, user experience, and security

## Functionality Specification Scope

In this project, I will present the implementation of a virtualized system that will contain several technologies to help store applications required for 3<sup>rd</sup> or 4<sup>th</sup> year cybercrime and IT Security modules. It assists students with access to several tools necessary for practical operations to help them achieve relevant IT skills. For example, some models such as forensics, Reverse engineering, networking, and Software Architecture utilize software to accomplish several tasks. Through this project, software can be stationed in a containerized environment or server, that students can access and explore. The use of automation software will create servers with containerized virtual machines and API to configure and deploy all tools. The system will showcase the efficiency of a type of virtualization that utilize resources of the Operating System rather than hardware.

## Technologies

- Ansible automation software
- Apache, httpd, ssh
- Docker container modules
- Linux operating system
- Linode
- LAN

## Application Users

### Administrators

The importance of administrators is crucial because they ensure privileges are deployed to users and security is maintained. Students operating VMs can unknowingly disrupt the network that virtual machines sit on, thereby disturbing their performance. Administrators would help prevent this because of their constant monitoring and supervising. Also, they put policies in place to help protect the system. This way, no resource is wasted.

### Students

Students would have regular interactions with virtual machines since they are limited with the privileges they have; this means they would not be able to perform modifications on VMs. Only administrators/tutors can enable this. In scenarios where the application they are working with requires ad-on, plugins or extensions, then administrator would grant them permission to include such tools. Students would not be able to view other VMs apart from their created VM

### Tutors

Tutors have higher permission over students. Such permissions allow them to assess practical work being done on students' VM, from a primary host machine. This way tutors can assist students with difficulties. Tutors would have permission to include ad-on, extensions or plug-ins required for tutorials, view, delete, assess and modify VMs.

# Use Case and Scenario

## Login Primary Actor

Administrator:

Preconditions:

Administrator has created server with docker containers, linode, and required services e.g webserver, https, mysql, php, and webapplications

Main Scenario

### 1. Administrator remote login

```
kenneth@kenneth-Lenovo-IdeaPad-S340-14API:~$ ssh moz@139.162.198.212
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:~$
```

### 2. View docker and docker images

```
moz@localhost:~$ sudo docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
6e2a87bcfd4   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          22 hours ago  Up 22 hours  0.0.0.0:11695->80/tcp, :::11695->80/tcp  intelligent_brahmagupta
d694234db74a   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          46 hours ago  Up 46 hours  0.0.0.0:11553->80/tcp, :::11553->80/tcp  suspicious_spence
e3228a6ccb9b   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          46 hours ago  Up 46 hours  0.0.0.0:11465->80/tcp, :::11465->80/tcp  fervent_mendeleev
```

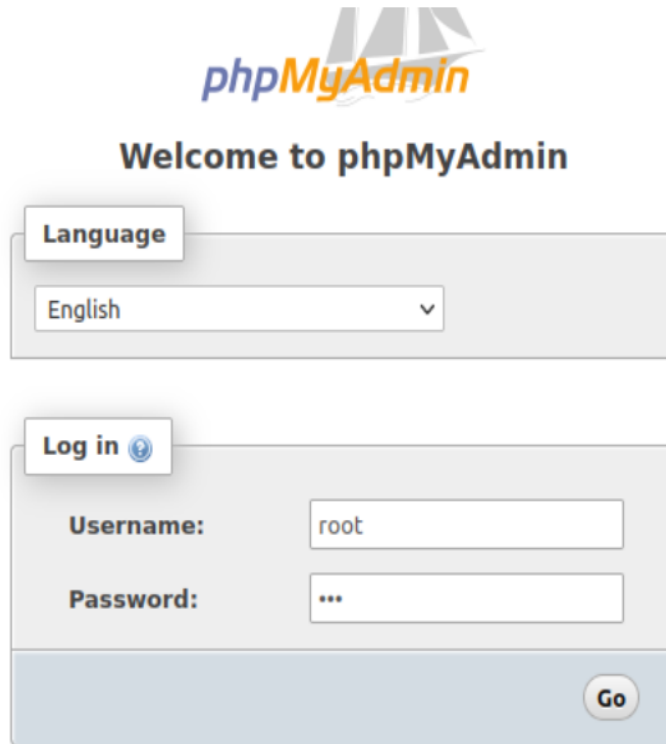
### 3. View running containers

```
moz@localhost:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
6e2a87bcfd4   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          21 hours ago  Up 21 hours  0.0.0.0:11695->80/tcp, :::11695->80/tcp  intelligent_brahmagupta
d694234db74a   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          45 hours ago  Up 45 hours  0.0.0.0:11553->80/tcp, :::11553->80/tcp  suspicious_spence
e3228a6ccb9b   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          46 hours ago  Up 46 hours  0.0.0.0:11465->80/tcp, :::11465->80/tcp  fervent_mendeleev
bd368d7513b3   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          2 days ago    Exited (0) 46 hours ago                    cranky_curran
8ac16e6bde6   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          2 days ago    Exited (0) 46 hours ago                    deterned_rldc
1d9c41ba2c65   lukaszlach/kall-desktop:xfce       "/lnit"                11 days ago   Exited (0) 46 hours ago                    laughing_satoshi
cd2f8c8b73e2   dorowu/ubuntu-desktop-lxde-vnc     "/startup.sh"          12 days ago   Exited (0) 46 hours ago                    nostalgic_pasteur
b763718d8ba6   lukaszlach/kall-desktop:xfce       "/lnit"                5 weeks ago   Exited (0) 46 hours ago                    affectionate_suth
```

### 4. Source codes

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

## 5. Database login



**phpMyAdmin**

### Welcome to phpMyAdmin

**Language**

English ▼

**Log in**

**Username:**

**Password:**

## 6. User credential table

+ Options		id	username	email	password	role
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	10	superadmin	superadmin@site.com	f91e15dbec69fc40f81f0876e7009648	0
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	11	saqlain	saqlain.sial05@gmail.com	81dc9bdb52d04dc20036dbd8313ed055	1
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	12	mo	moazam3005@gmail.com	bece27a540b3ed869571e6b4c73d1906	1
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	15	dan	dan@gmail.com	b421c83df77b00d587b62fe2591e5526	1
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	16	Test	test@test.com	c4ca4238a0b923820dcc509a6f75849b	1
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	17	ikram	ikram.arif05@gmail.com	202cb962ac59075b964b07152d234b70	1
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	18	kay	mytest@test.com	b421c83df77b00d587b62fe2591e5526	1
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	19	joe	student1@test.com	b421c83df77b00d587b62fe2591e5526	1

## 7. Quiz table

+ Options		id	user_id	module	task_1	task_2	task_3	task_4	task_5	date_created
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	10	FORENSICS	Create a directory called lab1: Command mkdir lab...	Create a file called student1: Command nano studen...				2022-04-04
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	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
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	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
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	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
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	5	18	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%	2022-04-05
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	6	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...	2022-04-08
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	7	19	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%	2022-04-17

Result

Successful

Administrator is able to

- Login remotely
- Deploy docker in server
- View docker and docker images
- View running containers
- View Source codes
- View or modify all tables in the database



## Login Primary Actor

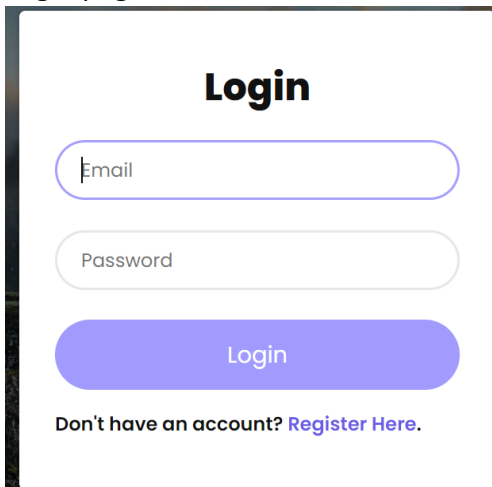
Tutor:

Preconditions:

Tutor has been provided an account and credentials to access the server

Main Scenario

1. Tutor will launch web application
2. Login page will load for tutor



**Login**

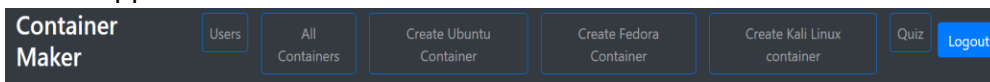
Email

Password

Login

Don't have an account? [Register Here.](#)

3. Enters login credential. Username and password
4. Login details are then validated by database
5. If validation is successful, then tutor will be granted access.
6. Access applications. View Containers created



Container Maker

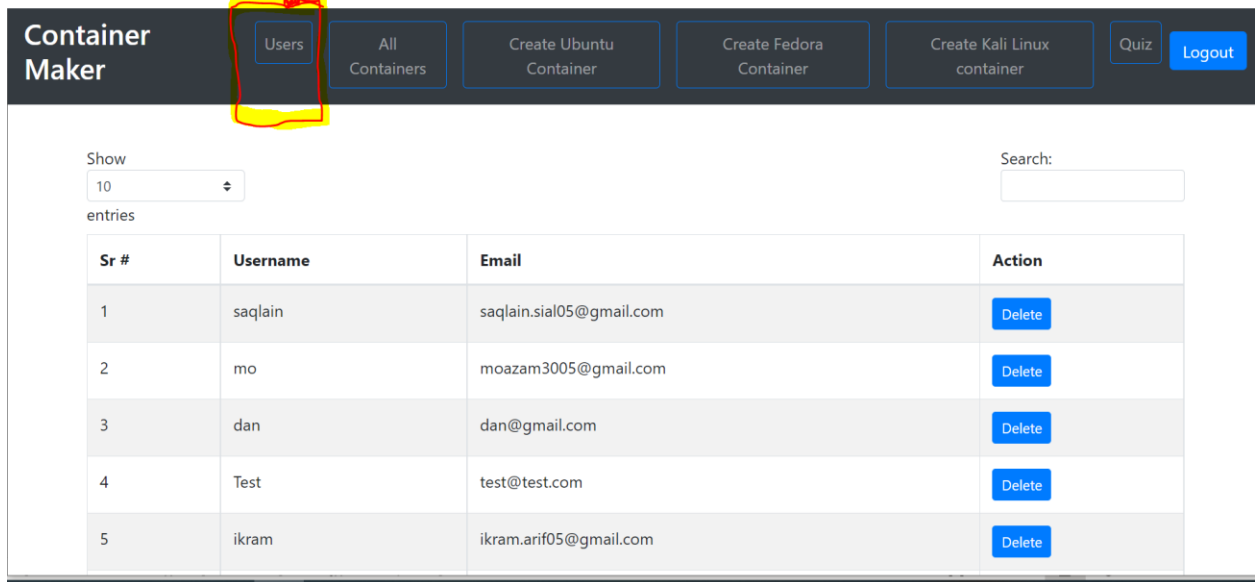
Users All Containers Create Ubuntu Container Create Fedora Container Create Kali Linux container Quiz Logout

Show  entries

Search:

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

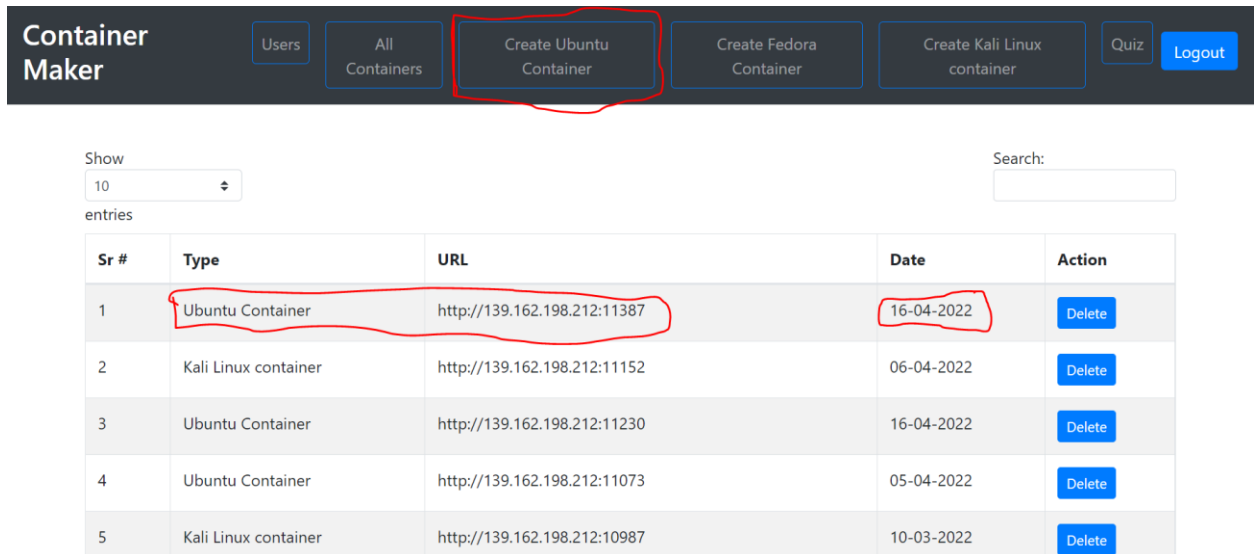
## 7. View Users and their containers



The screenshot shows the 'Container Maker' application interface. The 'Users' tab is highlighted with a yellow box. Below the navigation bar, there is a 'Show' dropdown set to '10' and a search field. The main content is a table with 5 entries, each with a 'Delete' button.

Sr #	Username	Email	Action
1	saqlain	saqlain.sial05@gmail.com	Delete
2	mo	moazam3005@gmail.com	Delete
3	dan	dan@gmail.com	Delete
4	Test	test@test.com	Delete
5	ikram	ikram.arif05@gmail.com	Delete

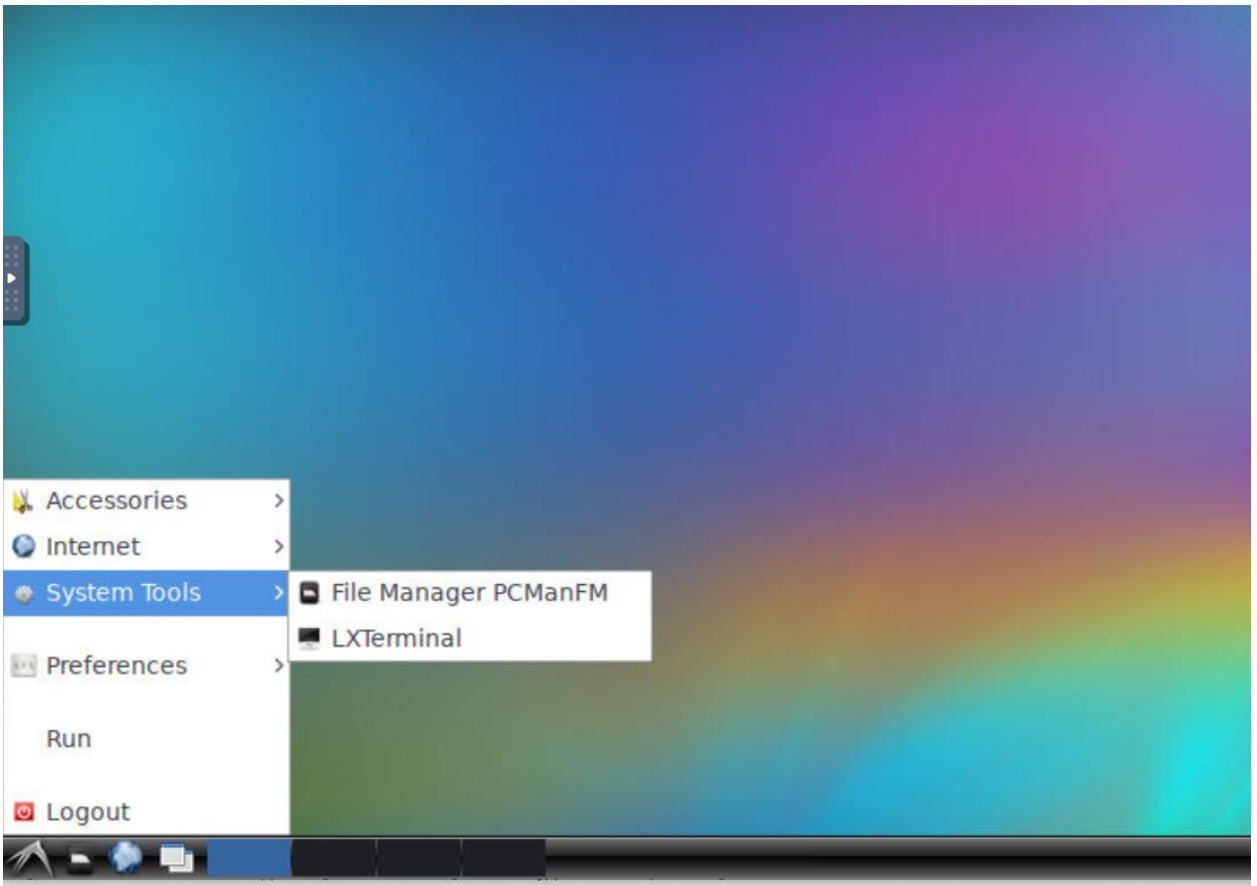
## 8. Create VM



The screenshot shows the 'Container Maker' application interface. The 'Create Ubuntu Container' tab is highlighted with a red box. Below the navigation bar, there is a 'Show' dropdown set to '10' and a search field. The main content is a table with 5 entries, each with a 'Delete' button. The first entry is circled in red.

Sr #	Type	URL	Date	Action
1	Ubuntu Container	http://139.162.198.212:11387	16-04-2022	Delete
2	Kali Linux container	http://139.162.198.212:11152	06-04-2022	Delete
3	Ubuntu Container	http://139.162.198.212:11230	16-04-2022	Delete
4	Ubuntu Container	http://139.162.198.212:11073	05-04-2022	Delete
5	Kali Linux container	http://139.162.198.212:10987	10-03-2022	Delete

9. Access students VM or Created VM



## 10. Terminal

```
root@8ac16e6bbde6: ~  
File Edit Tabs Help  
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:~#
```

## 11. Check for completed tasks



## 12. View Tasks/Quiz

Container Maker Users All Containers Create Ubuntu Container Create Fedora Container Create Kali Linux container Quiz

Start Quiz

Show 10 entries

Search:

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

## 13. Delete VMs

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

How to delete element 139.162.198.212 says are you sure?

OK Cancel

Lesson Three: Create... Online

Create Kali Linux container

Search:

	URL	Date	Action
iner	http://139.162.198.212:11553	16-04-2022	<span>Delete</span>
ainer	http://139.162.198.212:10987	10-03-2022	<span>Delete</span>
ner	http://139.162.198.212:1053/vnc.html Password:hieghai8At	26-02-2022	<span>Delete</span>
iner	http://139.162.198.212:10428	26-02-2022	<span>Delete</span>

entries

Sr #	Type	URL	Date	Action
1	Ubuntu Container	http://139.162.198.212:11553	16-04-2022	<a href="#">Delete</a>
2	Kali Linux container	http://139.162.198.212:10987	10-03-2022	<a href="#">Delete</a>
3	Fedora Container	http://139.162.198.212:1053/vnc.html Password:hieghai8At	26-02-2022	<a href="#">Delete</a>

Showing 1 to 3 of 3 entries

[Previous](#) [1](#) [Next](#)

## Result

Successful

Tutor is able to

- View All users and All containers
- Create VM
- Access students VM
- Enter terminal
- Check for completed tasks
- View tasks/Quiz
- Delete VMs
- Logout

Login Primary Actor

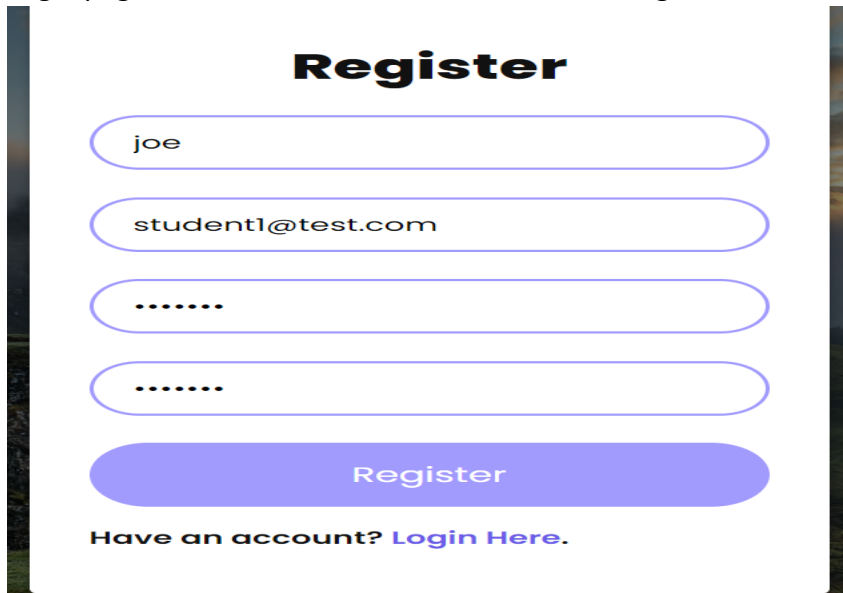
Student:

Preconditions:

Student will have to register before he or she can sign in

Main Scenario

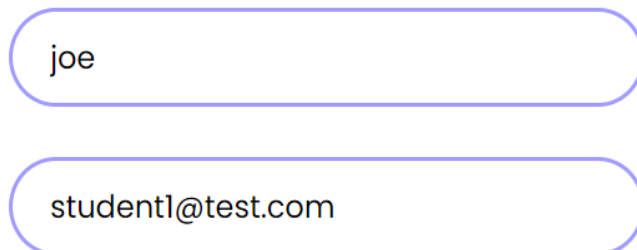
1. Student will launch web application
2. Login page will load for student. Student would register



The screenshot shows a registration form titled "Register". It contains four input fields: a username field with "joe", an email field with "student1@test.com", and two password fields, both containing six dots. Below the fields is a blue "Register" button. At the bottom, there is a link: "Have an account? [Login Here.](#)"

Wops! Email Already Exists.

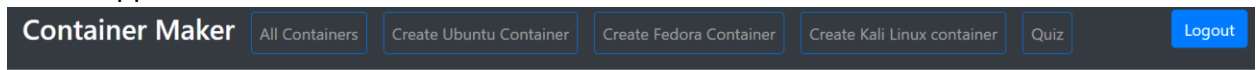
## Register



This partial screenshot shows the top two input fields of the registration form: the username field containing "joe" and the email field containing "student1@test.com".

3. Enters login credential. Student email and password if the email type does not exist and registration is complete

4. Login details are then validated by database
5. If validation is successful, then student will be granted access.
6. Access applications

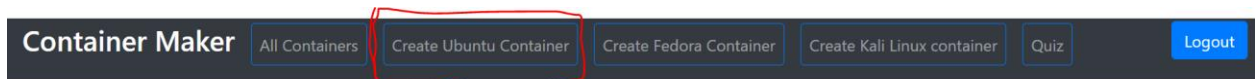


Show: 10 entries

Search:

Sr #	Type	URL	Date
No data available in table			

7. Student create VM



Show: 10 entries

Search:

Sr #	Type	URL	Date
No data available in table			

Show: 10 entries

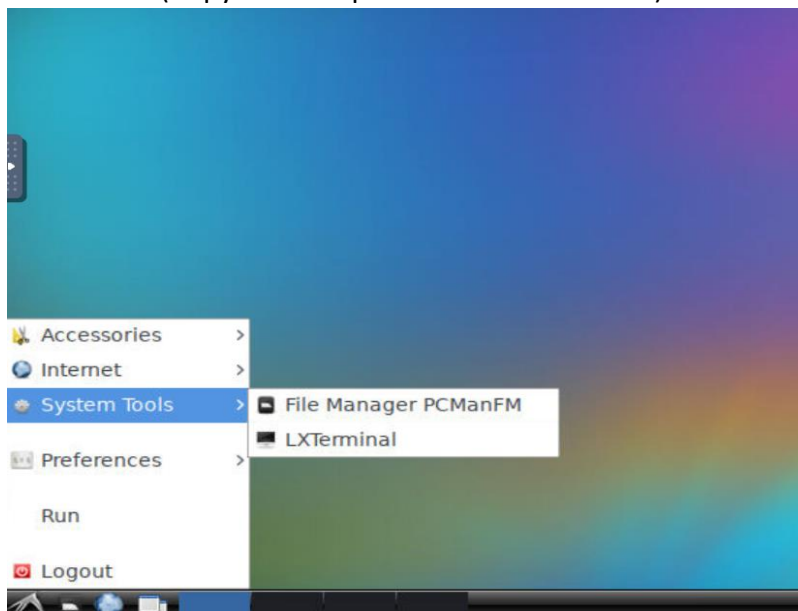
Search:

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

Showing 1 to 1 of 1 entries

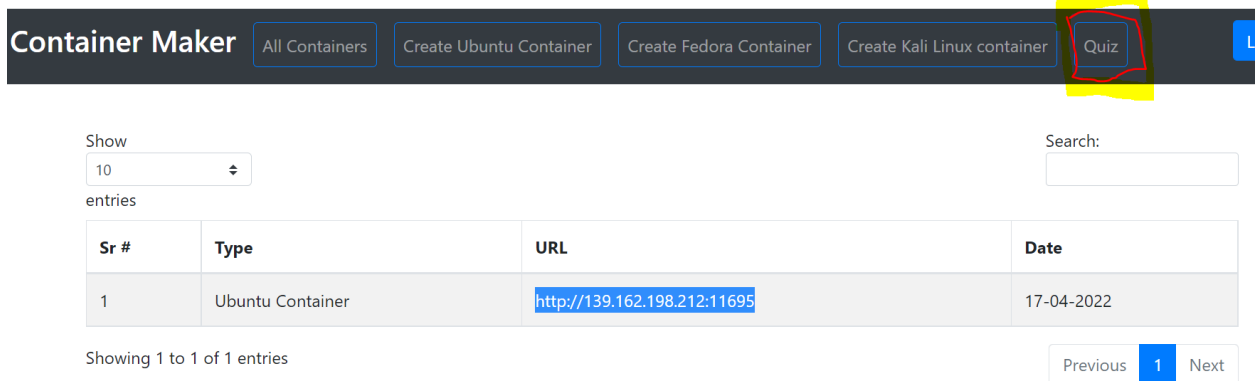
Previous 1 Next

8. Launch VM (Copy link and paste in a new browser)





## 9. Access Quiz



Container Maker All Containers Create Ubuntu Container Create Fedora Container Create Kali Linux container Quiz

Show 10 entries

Search:

Sr #	Type	URL	Date
1	Ubuntu Container	<a href="http://139.162.198.212:11695">http://139.162.198.212:11695</a>	17-04-2022

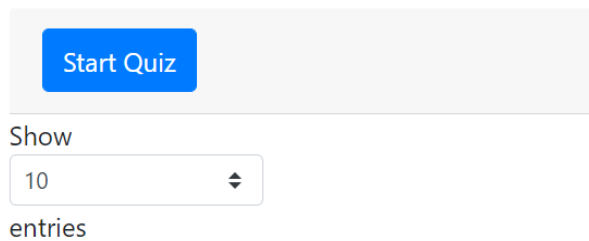
Showing 1 to 1 of 1 entries

Previous 1 Next

## 10. Quiz/Module menu



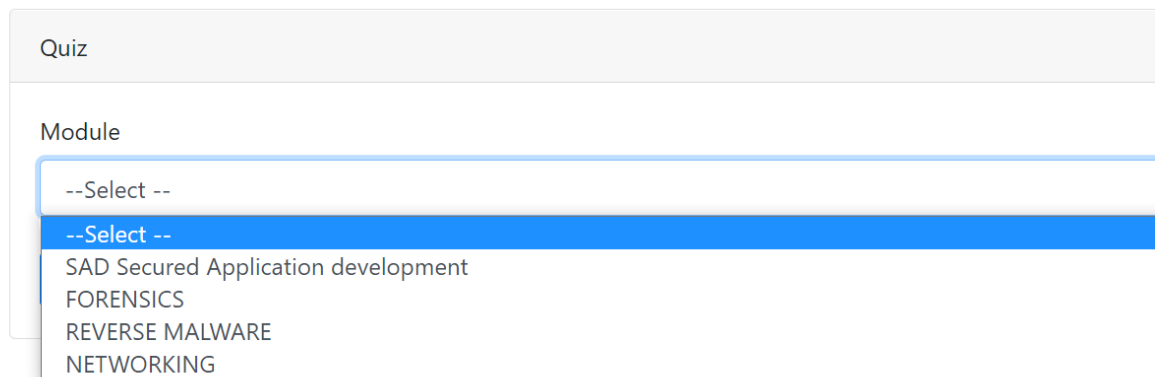
Container Maker All Containers Create U



Start Quiz

Show 10 entries

## 11. Select module/quiz



Quiz

Module

- Select --
- Select --
- SAD Secured Application development
- FORENSICS
- REVERSE MALWARE
- NETWORKING

## 12. Complete tasks and submit

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 would confirm completion by checking students VM and if completed tutor will grade. See bullet number 12 in tutors use case.

### Quiz

Wow! Quiz Completed.

### Module

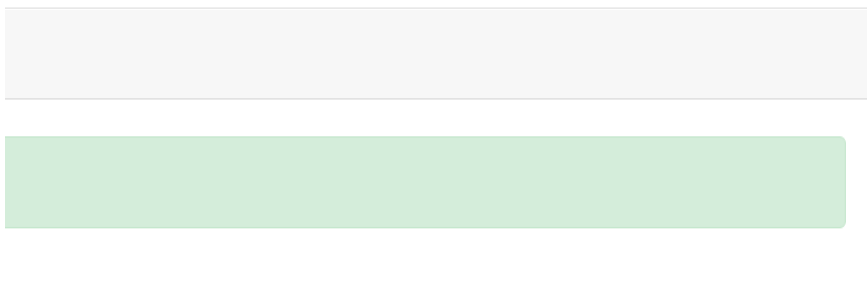
--Select--

Submit

```
student2@6e2a87dbcf4: ~/SAD1
File Edit Tabs Help
su: warning: cannot change directory to /home/student1: No such file or director
y
$ ^Z
root@6e2a87dbcf4:~# sudo adduser student2
Adding user `student2' ...
Adding new group `student2' (1001) ...
Adding new user `student2' (1001) with group `student2' ...
Creating home directory `/home/student2' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for student2
Enter the new value, or press ENTER for the default
  Full Name []: joe
  Room Number []: 216
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
root@6e2a87dbcf4:~# sudo su - student2
student2@6e2a87dbcf4:~$ mkdir SAD1
student2@6e2a87dbcf4:~$ cd SAD1/
student2@6e2a87dbcf4:~/SAD1$
```

13. Log out once tutor confirms completion

Create Fedora Container   Create Kali Linux container   Quiz   Logout



Result

Successful.

Student has been able to

- Register and login
- Create VM
- Launch VM
- Access Quiz
- Select module/quiz
- Complete tasks and submit
- Log out

## Source codes

### Config.php

```
Open  config.php
~/Documents/test-folder/docker_container/includes
1 <?php
2
3 $server = "localhost";
4 $user = "root";
5 $pass = "";
6 $database = "docker_app";
7
8 $conn = mysqli_connect($server, $user, $pass, $database);
9
0 if (!$conn) {
1     die('Connection Failed');
2 }
3
```

```
include 'includes/config.php';
session_start();
error_reporting(0);
if (isset($_SESSION['id'])) {
    header("Location: dashboard.php");
}
$error="";
if (isset($_POST['submit'])) {
    $email = $_POST['email'];
    $password = md5($_POST['password']);

    $sql = "SELECT * FROM users WHERE email='$email' AND password='$password'";
    $result = mysqli_query($conn, $sql);
    if ($result->num_rows > 0) {
        $row = mysqli_fetch_assoc($result);
        $_SESSION['id'] = $row['id'];
    }
}
```

### Logout.php

```
Open  logout.php
~/Documents/test-folder
1 <?php
2
3 session_start();
4 session_destroy();
5
6 header("Location: index.php");
7
8 ?>
```

## Test.php

```
test.php
~/Documents/test-folder
1 <?php
2
3 echo shell_exec('docker run -d -p 74:80 -v /dev/shm:/dev/shm dorowu/ubuntu-desktop-ixde-vnc');
4
5 ?>
```

## Users.php

```
users.php
~/Documents/test-folder
1 <?php
2 session_start();
3 if (!isset($_SESSION['id'])) {
4     header("Location: index.php");
5 }
6 if ($_SESSION['role']==1) {
7     header("Location: dashboard.php");
8 }
9
10
11 include 'includes/config.php';
12 $sql = "SELECT * from users where role=1";
13 $result = $conn->query($sql);
14
15 if(isset($_GET['id'])){
16     $conn->query("delete from users where id=".$_GET['id']);
17     $conn->query("delete from containers where user_id=".$_GET['id']);
18     header("Location: users.php");
19 }
20 ?>
21 <!DOCTYPE html>
22 <html lang="en">
23 <head>
24     <!-- Required meta tags -->
25     <meta charset="utf-8">
26     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
27
```

## Register.php

```
Open register.php
~/Documents/test folder

5 error_reporting(0);
6
7 session_start();
8
9 if (isset($_SESSION['username'])) {
10     header("Location: index.php");
11 }
12 $error = "";
13 if (isset($_POST['submit'])) {
14     $username = $_POST['username'];
15     $email = $_POST['email'];
16     $password = md5($_POST['password']);
17     $cpassword = md5($_POST['cpassword']);
18
19     if ($password == $cpassword) {
20         $sql = "SELECT * FROM users WHERE email='$email'";
21         $result = mysqli_query($conn, $sql);
22         if (!$result->num_rows > 0) {
23             $sql = "INSERT INTO users (username, email, password,role)
24                 VALUES ('$username', '$email', '$password',1)";
25             $result = mysqli_query($conn, $sql);
26             if ($result) {
27                 $error.= "<p class='alert alert-success'>Wow! User Registration Completed.</p>";
28                 $username = "";
29                 $email = "";
30                 $_POST['password'] = "";
31                 $_POST['cpassword'] = "";
32             } else {
33                 $error.= "<p class='alert alert-danger'>Wops! Something Wrong Went.</p>";
34             }
35         } else {
36             $error.= "<p class='alert alert-danger'>Wops! Email Already Exists.</p>";
37         }
38     }
39 }
40
```

```
function get_port($id)
{
    $port = "";
    if (strlen($id) == 1 || strlen($id) == 2) {
        $port = (rand(1, 10000));
    }
    if (strlen($id) == 3 | strlen($id) == 4) {
        $port = $id . (rand(1, 100));
    }
    if (strlen($id) == 5 | strlen($id) == 6) {
        $port = (rand(1, 100));
    }
    return $port;
}
```

