

# **Distributed and Cloud System Computing (5CS022)**

## **Azure Web Hosting (Task 2)**

**Student Id:** 2332244

**Student Name:** Naomi Thing

**Group:** L5CG4

**Module Leader:** Mr. Deepshon Shrestha

**Tutor:** Mr. Deepshon Shrestha

**Submitted Date:** 18<sup>th</sup> May, 2024.

## Project Configuration:

The first step, we create a virtual machine using Linux Ubuntu as the operating system. On the same page, we configure the operating system's login and password and enable port access to the virtual machine.

[Home](#) > [Virtual machines](#) >

### Create a virtual machine ...

 This subscription may not be eligible to deploy VMs of certain sizes in certain regions.

#### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="Azure for Students"/>
Resource group *	<input type="text" value="(New) naomithing123"/>

[Create new](#)

#### Instance details

Virtual machine name *	<input type="text" value="naomi"/>
Region *	<input type="text" value="(US) East US"/>
Availability options	<input type="text" value="No infrastructure redundancy required"/>
Security type	<input type="text" value="Trusted launch virtual machines"/>
Image *	<input type="text" value="Ubuntu Server 20.04 LTS - x64 Gen2"/>
VM architecture	<input checked="" type="radio"/> x64
Run with Azure Spot discount	<input type="checkbox"/>
Size *	<input type="text" value="Standard_B2s - 2 vcpus, 4 GiB memory (US\$30.37/month)"/>

[See all images](#) | [Configure VM generation](#)

[See all sizes](#)

### Administrator account

Authentication type ⓘ

- ☐ SSH public key  
☒ Password

Username \* ⓘ

naomi ✓

Password \*

..... ✓

Confirm password \*

..... ✓

### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ

- ☐ None  
☒ Allow selected ports

Select inbound ports \*

HTTP (80), HTTPS (443), SSH (22) ▾

**i** All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

## Create a virtual machine ...

✓ Validation passed

Place this virtual machine behind an existing load balancing solution?

No

Delete NIC when VM is deleted

Disabled

### Management

Microsoft Defender for Cloud

None

System assigned managed identity

Off

Login with Microsoft Entra ID

Off

Auto-shutdown

Off

Enable hotpatch

Off

Patch orchestration options

Azure-orchestrated patching (preview): patches will be installed by Azure

Reboot setting

Reboot if required

### Monitoring

Alerts

Off

Boot diagnostics

On

Enable OS guest diagnostics

Off

Enable application health monitoring

Off

### Advanced

Extensions

None

VM applications

None

Cloud init

No

User data

No

Disk controller type

SCSI

Proximity placement group

None

Capacity reservation group

None

< Previous

Next >

Create

Here, the deployment is complete. Following deployment, we use the VM's resources to learn more about the virtual OS.

The screenshot shows the Azure portal interface for a deployment. The top navigation bar includes 'Home >' and the deployment name 'CreateVm-canonical.0001-com-ubuntu-server-focal-2-20240518142120 | Overview'. Below the navigation bar, there's a search bar and a toolbar with actions: Delete, Cancel, Redeploy, Download, and Refresh. The left sidebar shows a list of tabs: Overview (selected), Inputs, Outputs, and Template. The main content area displays a green checkmark icon and the text 'Your deployment is complete'. Below this, it shows deployment details: Deployment name, Subscription (Azure for Students), Resource group (naomithing123), Start time (18/05/2024, 14:25:24), and Correlation ID. There are also sections for 'Deployment details' and 'Next steps' with recommended actions like 'Setup auto-shutdown', 'Monitor VM health, performance and network dependencies', and 'Run a script inside the virtual machine'. At the bottom, there are buttons for 'Go to resource' and 'Create another VM', and a 'Give feedback' link.

We may use the SSH network protocol to connect to our virtual operating system's Linux command line. The command line allows us to make and save the necessary changes to set up a hosting environment for the project.

```
~ % ssh naomi@52.186.68.217
```

```
The authenticity of host '52.186.68.217 (52.186.68.217)' can't be established.  
ED25519 key fingerprint is SHA256:HNpafMmgCshQTQF3/28UEu+mC+4FWBfjt0Mc5beLeY.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '52.186.68.217' (ED25519) to the list of known hosts.  
naomi@52.186.68.217's password:
```

The next step is to install Webmin and Apache on our virtual machine. By installing these tools, we guarantee that our virtual system is fully capable of keeping and providing our project's online content effectively. This step is critical for showing a dependable and easily accessible hosting environment.

```
[naomi@52.186.68.217's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1064-azure x86_64)

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

System information as of Sat May 18 09:24:24 UTC 2024

System load:  0.18           Processes:            115
Usage of /:   5.1% of 28.89GB Users logged in:         0
Memory usage: 7%           IPv4 address for eth0: 10.1.0.4
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

naomi@naomi:~$
```

```
naomi@naomi:~$ wget http://www.webmin.com/jcameron-key.asc
```

OK

```
naomi@naomi:~$ sudo apt-key add jcameron-key.asc
```

```
[naomi@naomi:~$ sudo bash
[root@naomi:/home/naomi# echo "deb http://download.webmin.com/download/repository sarge contrib" >> /etc/apt/sources.list
[root@naomi:/home/naomi# exit
exit
naomi@naomi:~$
```

```
[naomi@naomi:~$ sudo apt update
Get:1 http://azure.archive.ubuntu.com/ubuntu focal
Get:2 http://azure.archive.ubuntu.com/ubuntu focal
```

✓ **Network security group naomi-nsg** (attached to networkInterface: naomi440)  
Impacts 0 subnets, 1 network interfaces

Search rules

Source == all Destination == all Protocol == all Action == all

Priority ↑	Name	Port	Protocol	Source	Destination	Action
Inbound port rules (6)						
300	SSH	22	TCP	Any	Any	Allow
320	HTTPS	443	TCP	Any	Any	Allow
340	HTTP	80	TCP	Any	Any	Allow

+ Create port rule

Inbound port rule

Outbound port rule

In addition, we established port 10000 to allow access to Webmin, which provides a simple interface for administering the system and its file storage. This setup is required for storing and accessing the project's files, ensuring that Webmin is always obtainable of efficient system administration and project-related operations.



## Add inbound security rule

naomi-nsg



Source ⓘ

Any



Source port ranges \* ⓘ

\*

Destination ⓘ

Any



Service ⓘ

Custom



Destination port ranges \* ⓘ

10000



Protocol



Any



TCP



UDP



ICMP

Action



Allow



Deny

Priority \* ⓘ

350



Name \*

Webmin



Description

Webmin Control Panel

Add

Cancel



Give feedback

Then, I installed a LAMP stack (Linux, Apache, MySQL, and PHP), a popular web development software package. After upgrading the system, I installed the Apache web server, which handles HTTP requests. Then, I installed the MySQL database server for data administration, followed by PHP for dynamic content and server-side scripting. This arrangement provides a solid foundation for creating and distributing web applications.

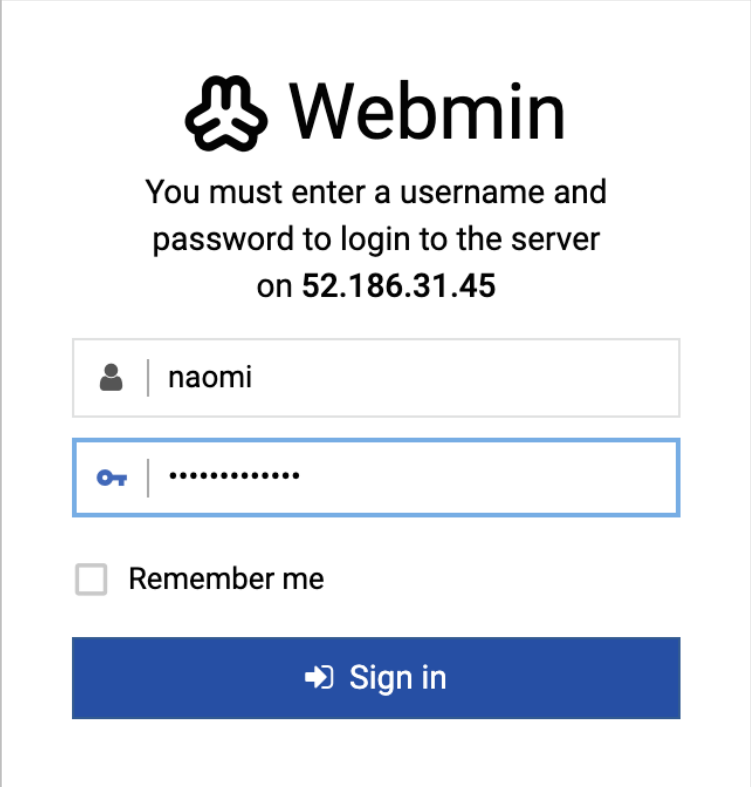
```
naomi@naomi:~$ sudo apt install mysql-server
```

```
naomi@naomi:~$ sudo apt install apache2
```

```
naomi@naomi:~$ sudo apt install php libapache2-mod-php php-mysql
```



Then, logging into Webmin.



The image shows the Webmin login interface. At the top, there is a Webmin logo (a stylized flower-like icon) followed by the text "Webmin". Below this, a message states: "You must enter a username and password to login to the server on 52.186.31.45". There are two input fields: the first is for the username, containing the text "naomi", and the second is for the password, containing a series of dots. Below the password field is a checkbox labeled "Remember me". At the bottom, there is a blue button with a right-pointing arrow and the text "Sign in".

**Webmin**

You must enter a username and password to login to the server on 52.186.31.45

☐ Remember me

[➔ Sign in](#)

Now, I moved on to creating database for the application that I am going to host.

Create Database

New database options

Database name

event\_planning

Character set

<Default>

Collation order

<Default>

Initial table

☒ None ☐ Named

with fields below...

Field name	Data type	Type width	Key?	Auto-increment?	Allow nulls?	Unsigned?	Default value
			<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
			<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
			<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
			<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	

+

 Create

Execute SQL

Execute SQL

Run SQL from file

Import text file

Enter an SQL command to execute on database event\_planning ..

```
CREATE TABLE proposals (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  proposal VARCHAR(100) NOT NULL,  
  description TEXT NOT NULL,  
  votes INT NOT NULL DEFAULT 0  
);
```

Character set for data

<Default>

 Execute

+

 Return to table list

←

 Return to database list

MySQL user details

Username

☐ Anonymous user
☒

Password

☐ None
☒ Set to..

Hosts

☐ Any
☒

Permissions

Create View

Show View

Create Routine

Alter Routine

Create User

Create Event

Create Trigger

Create Tablespace

Create\_role\_priv

Drop\_role\_priv

Maximum concurrent logins

☒ Unlimited
☐ At most

Maximum connections per hour

☒ Unlimited
☐ At most

Maximum queries per hour

☒ Unlimited
☐ At most

Maximum updates per hour

☒ Unlimited
☐ At most

Required certificate type

SSL cipher

Create



Create new file



New file name

db.php|

Create

Cancel

db.php (/var/www/html) ☆

```
1 <?php
2 $servername = "localhost";
3 $username = "phpuser";
4 $password = "password@123";
5 $dbname = "event_planning";
6
7 // Create connection
8 $conn = new mysqli($servername, $username, $password, $dbname);
9
10 // Check connection
11 if ($conn->connect_error) {
12     die("Connection failed: " . $conn->connect_error);
13 }
14 ?>
15 |
```

Create new file

New file name

index.php

Create

Cancel

```
index.php (/var/www/html) ☆ PHP utf-8
1 <?php
2 include 'db.php';
3
4 // Handle form submission to add a new proposal
5 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_proposal'])) {
6     $proposal = $_POST['proposal'];
7     $description = $_POST['description'];
8
9     $sql = "INSERT INTO proposals (proposal, description) VALUES ('$proposal', '$description')";
10    if ($conn->query($sql) === TRUE) {
11        echo "New proposal created successfully";
12    } else {
13        echo "Error: " . $sql . "<br>" . $conn->error;
14    }
15 }
16
17 // Handle voting
18 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['vote'])) {
19     $id = $_POST['id'];
20
21     $sql = "UPDATE proposals SET votes = votes + 1 WHERE id = $id";
```

## ⚙️ Change permissions

Mode:

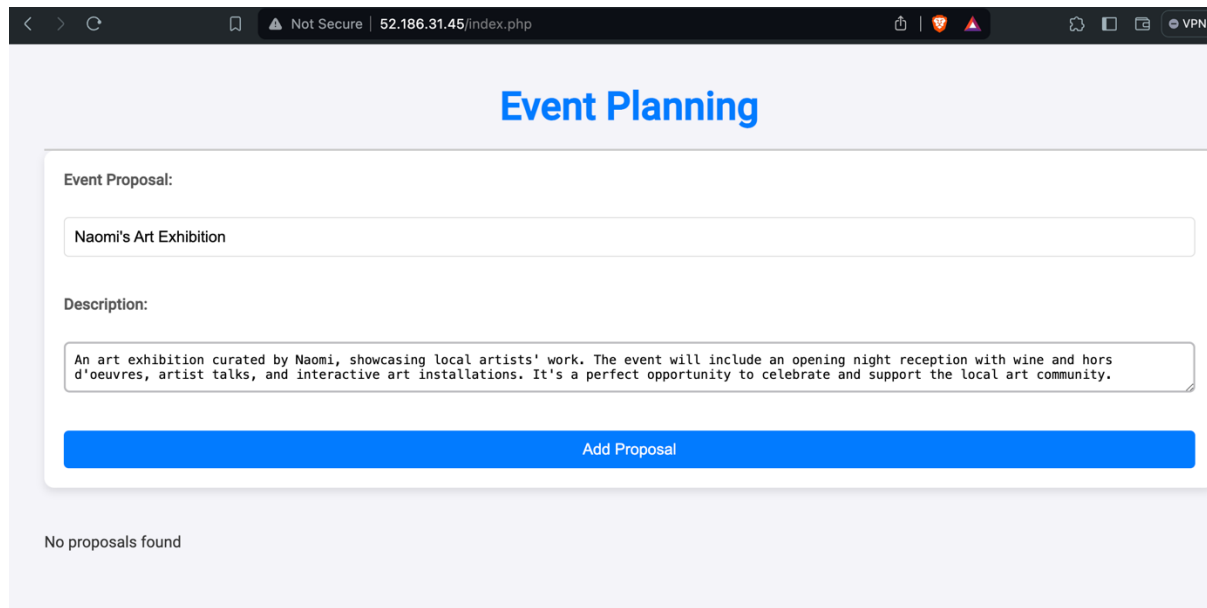
	Owner	Group	Others
Read	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Execute	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sticky bit	<input type="checkbox"/>		
Setgid	<input type="checkbox"/>		

Apply to:

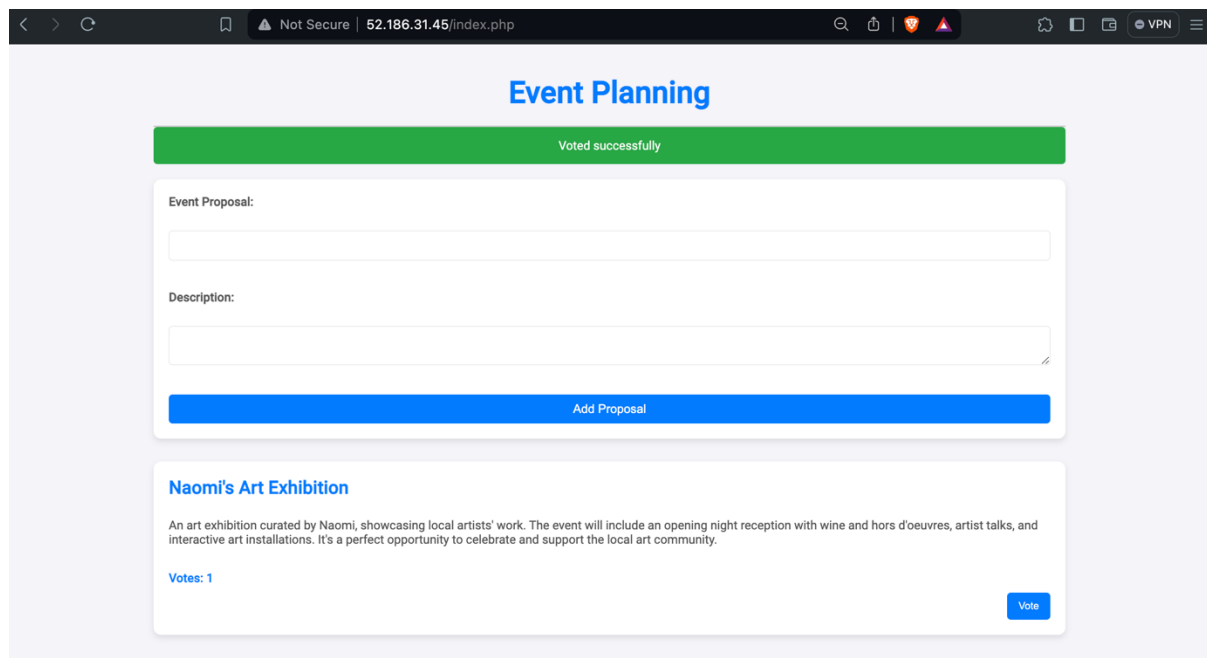
Change

Cancel

Then, here is the hosted project in the Azure virtual machine.



The screenshot shows a web browser window with the address bar displaying "Not Secure | 52.186.31.45/index.php". The page title is "Event Planning". Below the title, there is a form titled "Event Proposal:". The form has two input fields: "Event Proposal:" with the text "Naomi's Art Exhibition" and "Description:" with the text "An art exhibition curated by Naomi, showcasing local artists' work. The event will include an opening night reception with wine and hors d'oeuvres, artist talks, and interactive art installations. It's a perfect opportunity to celebrate and support the local art community." Below the form is a blue button labeled "Add Proposal". At the bottom of the page, it says "No proposals found".



The screenshot shows the same web browser window, but now with a green banner at the top that says "Voted successfully". Below the banner, the "Event Proposal:" form is still visible, but the "Add Proposal" button is now disabled. Below the form, there is a section titled "Naomi's Art Exhibition" with the same description as before. Below the description, it says "Votes: 1" and there is a blue button labeled "Vote".