Distributed and Cloud System Computing (5CS022)

Azure Web Hosting (Task 2)

Student Id: 2332244

Student Name: Naomi Thing

Group: L5CG4

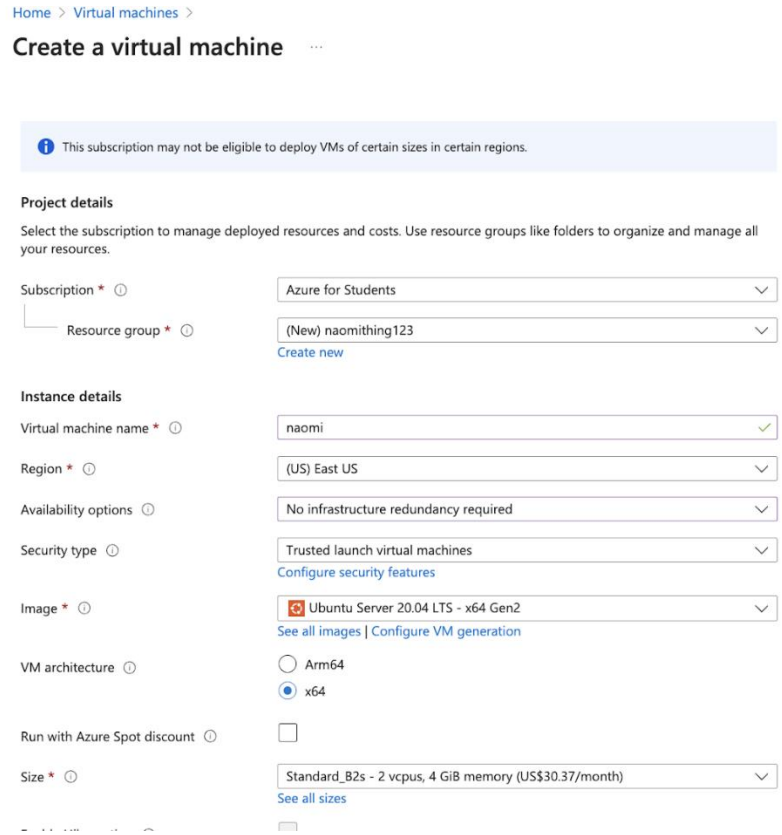
Module Leader: Mr. Deepshon Shrestha

Tutor: Mr. Deepshon Shrestha

Submitted Date: 18th 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.



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)

ⓘ 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

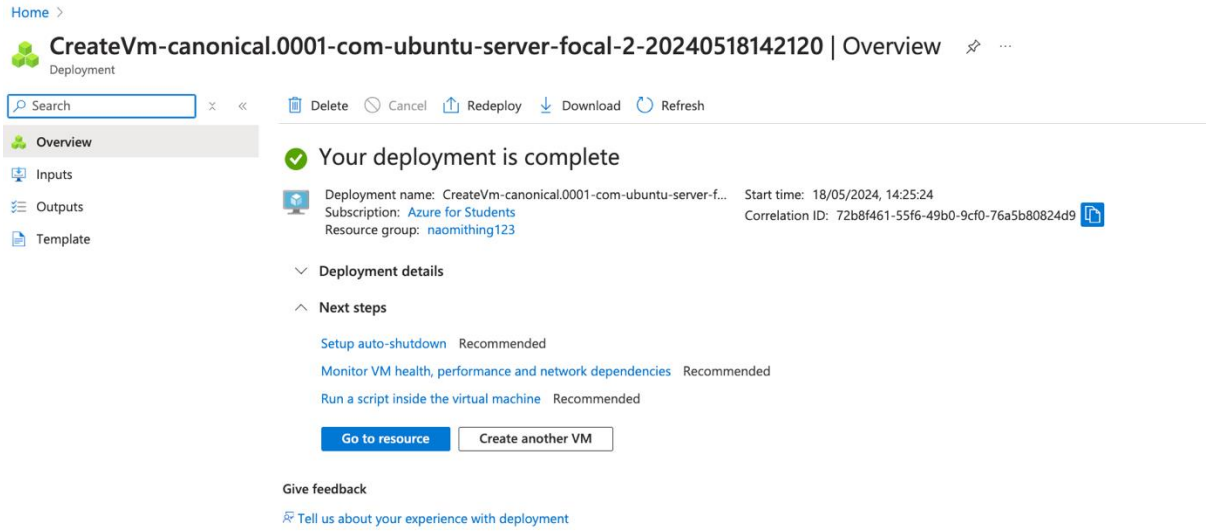
Alerts	Off
Boot diagnostics	On
Enable OS guest diagnostics	Off
Enable application health monitoring	Off

Advanced

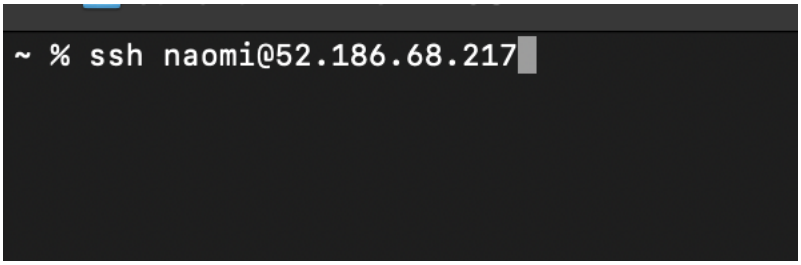
Extensions	None
VM applications	None
Cloud init	No
User data	No
Disk controller type	SCSI
Proximity placement group	None
Capacity reservation group	None

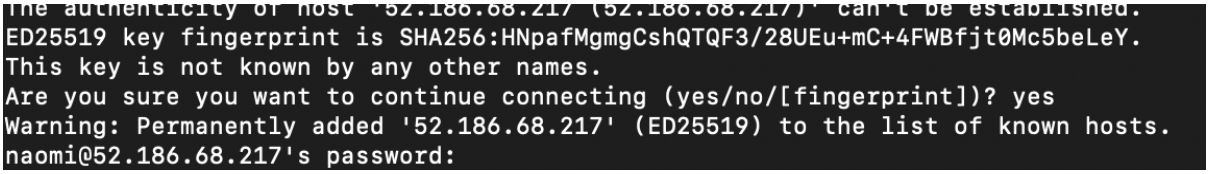
< Previous | Next > | Create

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



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.

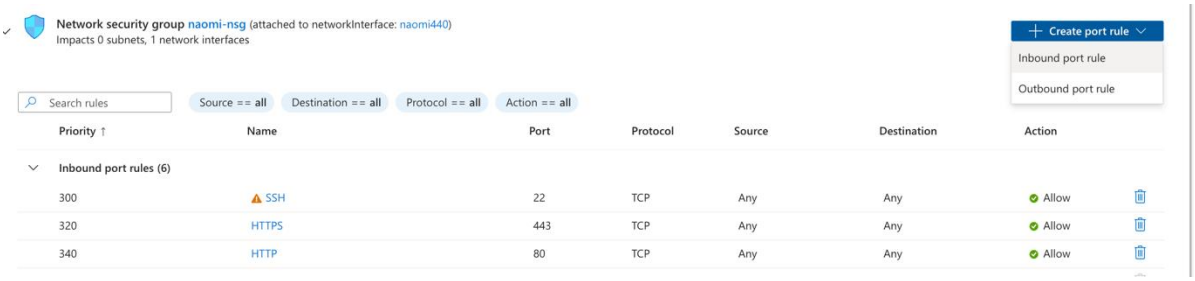




The next step is to install Webmin and Apache on our virtual machine. By installing these tools, we guarantee that our virtual system is fully capable of keeping and providing our project's online content effectively. This step is critical for showing a dependable and easily accessible hosting environment.

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

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



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 *

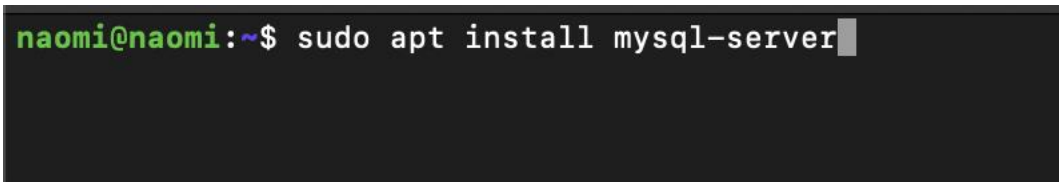
Webmin

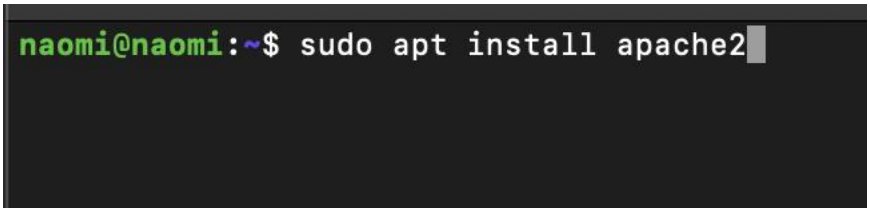
Description

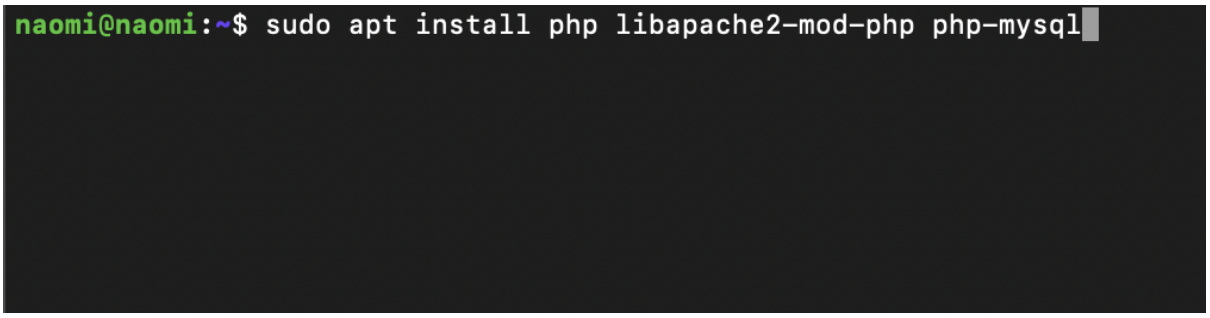
Webmin Control Panel

Add | Cancel | Give feedback

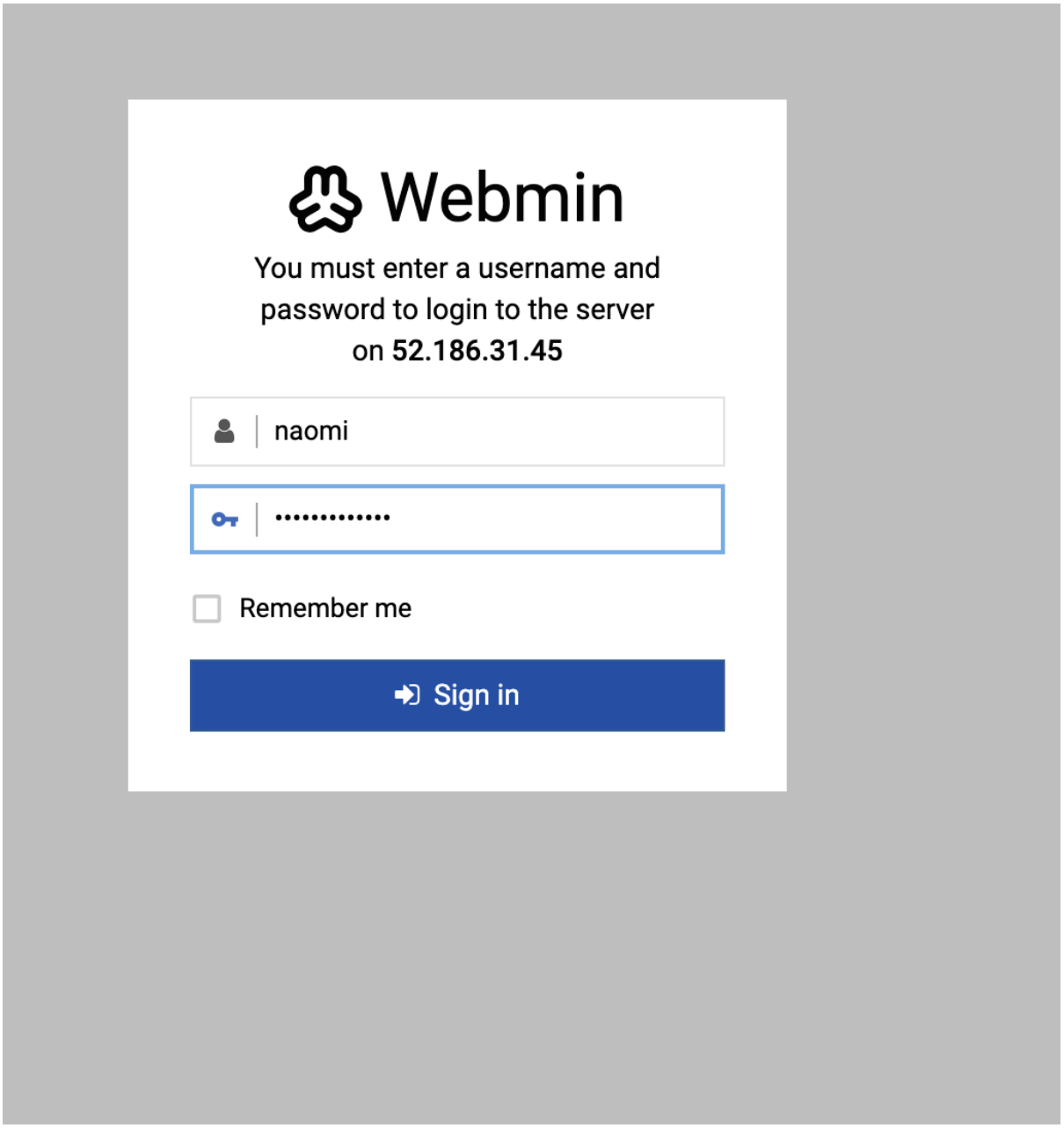
Then, I installed a LAMP stack (Linux, Apache, MySQL, and PHP), a popular web development software package. After upgrading the system, I installed the Apache web server, which handles HTTP requests. Then, I installed the MySQL database server for data administration, followed by PHP for dynamic content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



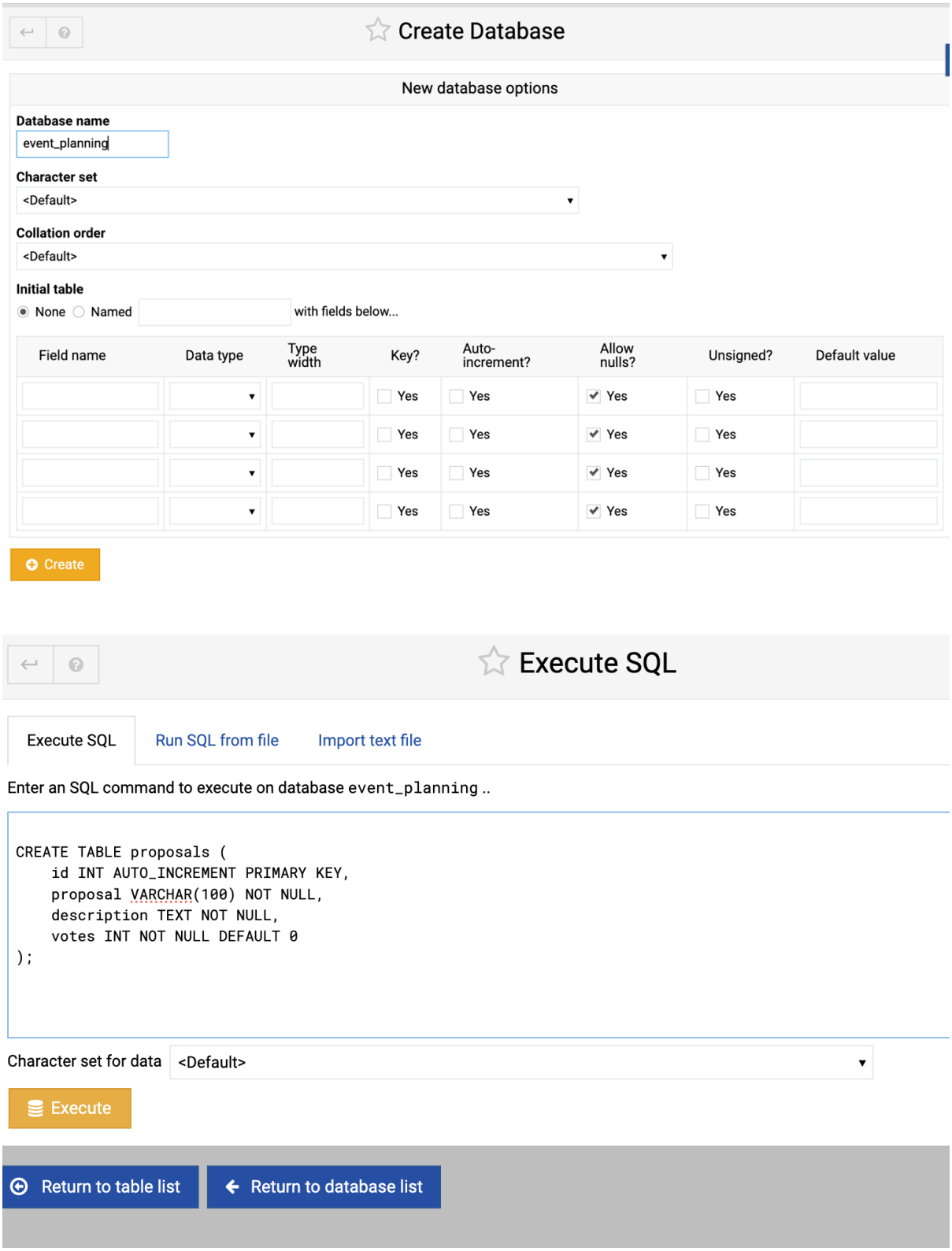


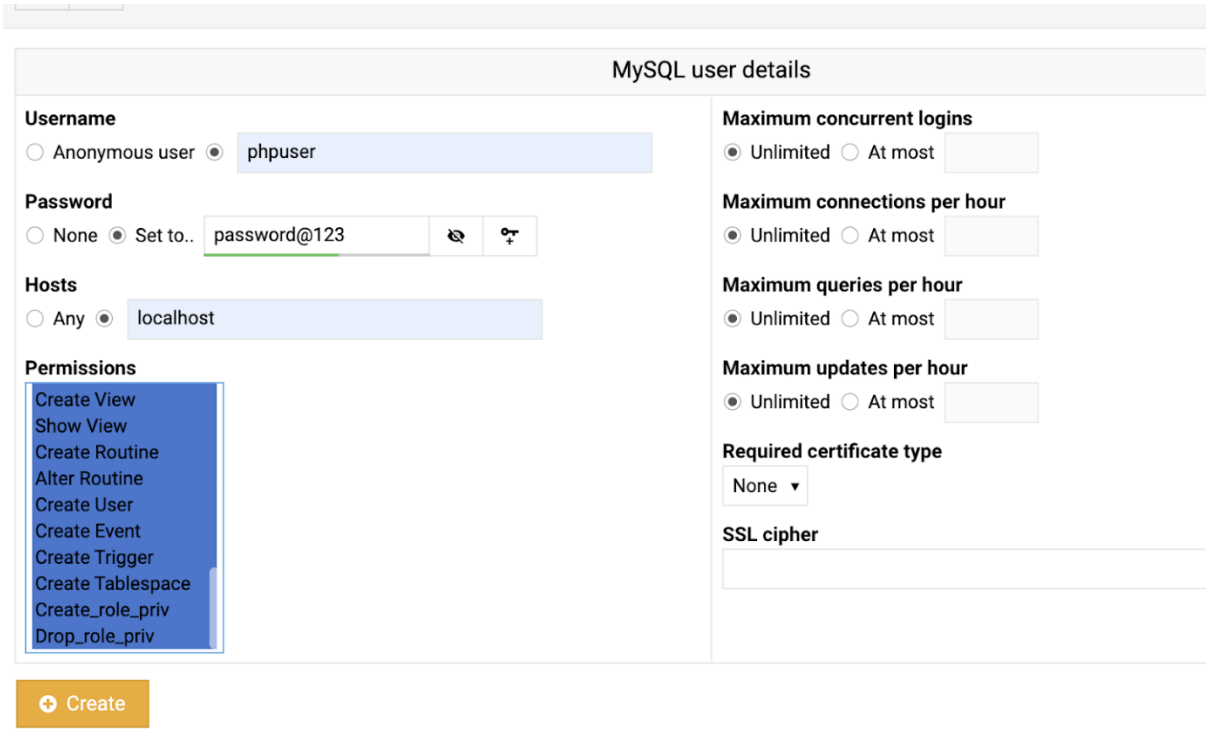


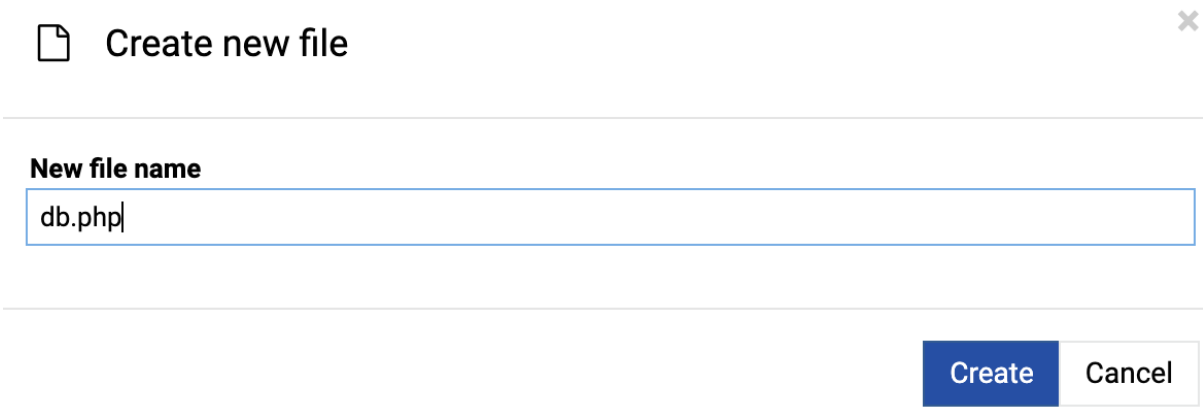
Then, logging into Webmin.



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




MySQL user details
Username
Anonymous user
phpuser
Password
None
Set to..
password@123
Hosts
Any
localhost
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
None
SSL cipher
Create

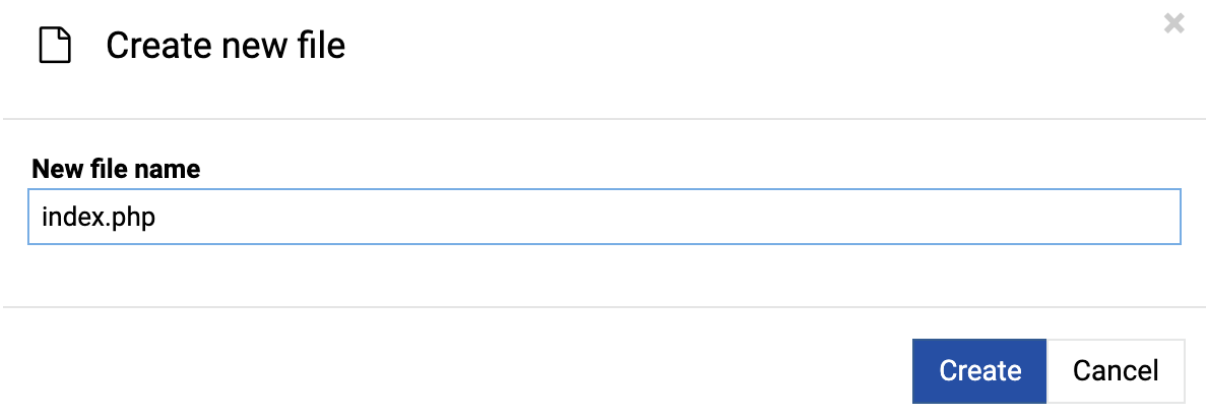

Create new file
New file name
db.php
Create
Cancel

db.php (/var/www/html)

```php
<?php
$servername = "localhost";
$username = "phpuser";
$password = "password@123";
$dbname = "event_planning";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```



index.php (/var/www/html) PHP utf-8

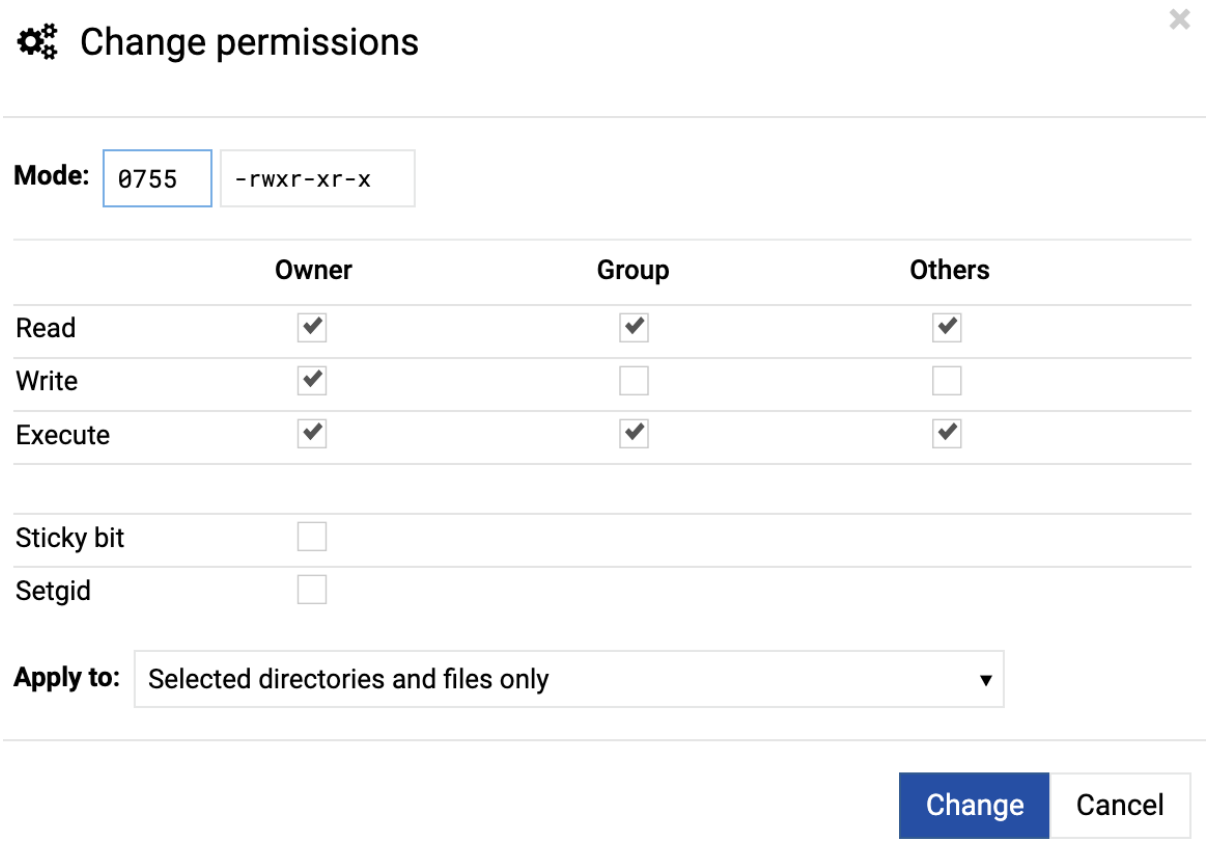
```php
<?php
include 'db.php';

// Handle form submission to add a new proposal
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_proposal'])) {
    $proposal = $_POST['proposal'];
    $description = $_POST['description'];

    $sql = "INSERT INTO proposals (proposal, description) VALUES ('$proposal', '$description')";
    if ($conn->query($sql) === TRUE) {
        echo "New proposal created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}

// Handle voting
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['vote'])) {
    $id = $_POST['id'];

    $sql = "UPDATE proposals SET votes = votes + 1 WHERE id = $id";
```



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