**Documentation**

sudo apt-get update

sudo apt-get install apache2

sudo systemctl status apache2

sudo apt-get install mysql-server

sudo systemctl start mysql

sudo systemctl enable mysql

sudo apt-get install php libapache2-mod-php php-mysql

sudo systemctl restart apache2

Verify Installation :

Create File info.php in this location to Verfiy

touch /var/www/html/info.php

Enter this link to verfiy <http://localhost/info.php>

2- To configure Apache to serve the website from the /var/www/html/ directory on Ubuntu, you typically need to adjust the default virtual host configuration. Here's how you can do it

sudo nano /etc/apache2/sites-available/000-default.conf

DocumentRoot /var/www/html

And restart Apache

sudo systemctl restart apache2

3- **Create a simple website that displays the message "Hello World!"; when accessed through a web browser.**

**cd /var/www/html/**

**sudo nano index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Hello World!</title>**

**</head>**

**<body>**

**<h1>Hello World!</h1>**

**</body>**

**</html>**

**sudo chmod 644 index.html**

**sudo systemctl restart apache2**

**4- Configure MySQL to create a new database, user, and password for the website.**

**sudo mysql -u root -p**

**CREATE DATABASE mywebsite\_db;**

**SET GLOBAL validate\_password.policy = 0;**

**CREATE USER 'mywebsite\_user'@'localhost' IDENTIFIED BY 'StrongPassword123!';**

**GRANT ALL PRIVILEGES ON mywebsite\_db.\* TO 'mywebsite\_user'@'localhost';**

**FLUSH PRIVILEGES;**

**EXIT;**

**5- Modify the website to use the newly created database to display a message that includes the visitor's**

**IP address and the current time.**

**USE mywebsite\_db;**

**CREATE TABLE visitor\_logs (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**ip\_address VARCHAR(45) NOT NULL,**

**visit\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

**2- cd /var/www/html**

**2- sudo gedit info.php**

**<?php**

**// Connect to MySQL database**

**$servername = "localhost";**

**$username = "mywebsite\_user";**

**$password = "StrongPassword123!";**

**$database = "mywebsite\_db";**

**$conn = new mysqli($servername, $username, $password, $database);**

**// Check connection**

**if ($conn->connect\_error) {**

**die("Connection failed: " . $conn->connect\_error);**

**}**

**// Get visitor's IP address**

**$ip\_address = $\_SERVER['REMOTE\_ADDR'];**

**// Insert visitor information into database**

**$sql = "INSERT INTO visitor\_logs (ip\_address) VALUES ('$ip\_address')";**

**$conn->query($sql);**

**// Display visitor's IP address and current time**

**echo "Hello visitor from " . $ip\_address . "!<br>";**

**echo "The current time is: " . date("Y-m-d H:i:s");**

**// Close database connection**

**$conn->close();**

**?>**

**6- when enter this website**

[**http://localhost/info.php**](http://localhost/info.php)

**The result is : Hello visitor from ::1!**

**The current time is: 2024-03-19 16:22:17**

**Task 2 :**

**1- cd /home/amany/muticloudDEVOPS**

**2- git init**

**3- nano .gitignore**

**4- # Ignore configuration files with passwords**

**config/passwords.ini**

**config/\*.secret**

**5- git add .gitignore**

**6- git commit -m "Add .gitignore to exclude sensitive files"**

**7- git push -u origin master**

**Task#3 Containerize your repo:**

**2- docker build -t my-apache-server:v1 .**

**3- docker run -p 8080:8080 my-apache-server**

**4- docker-compose up --build**

**Task 4 : Networking Basics**

**Explain what IP address used in these tasks and it's routing protocols:**

**When when enter the URL website for php using ip address for connect , java application when build docker file expose port 8080 use ip address to enter the website , IP addresses used in Docker Compose are typically assigned by Docker and are internal to the Docker network. Services can communicate with each other using these internal IP addresses.**

**Routing protocols such as BGP or OSPF are not used within Docker Compose, as it operates within a single host or a controlled network environment. Docker manages the routing internally using its networking features.**