Task: Set up a Web Server on a Linux Machine

Overview:
You will set up a basic LAMP (Linux, Apache, MySQL, PHP) stack on a Linux server, host a simple PHP-based webpage, integrate database functionality, and make it accessible externally via a public URL. You will then use Git to manage your code and documentation, push it to GitHub, and write clear explanations of fundamental networking concepts.

Sub-task #1: Linux Server Simulation (LAMP Setup)

1. **Install Required Packages:**
    - Install Apache, MySQL, and PHP on the Linux machine.
    - Example (Debian/Ubuntu):
    - sudo apt-get update
    - sudo apt-get install apache2 mysql-server php libapache2-mod-php php-mysql
2. **Configure Apache:**
    - Ensure that the server is configured to serve the website from the /var/www/html/ directory.
    - Test by creating a simple index.html in /var/www/html/ and ensuring it is accessible via http://<server-ip>/.
3. **Create a Simple Website:**
    - Replace index.html with a PHP file (e.g. index.php) that displays "Hello World!".
    - Verify this by accessing http://<server-ip>/ in a web browser.
4. **Configure MySQL:**
    - Secure the MySQL installation (mysql_secure_installation or similar).
    - Create a new database (e.g. web_db) and a new MySQL user with a password. For example:
    - CREATE DATABASE web_db;
    - CREATE USER 'web_user'@'localhost' IDENTIFIED BY 'StrongPassword123';
    - GRANT ALL PRIVILEGES ON web_db.* TO 'web_user'@'localhost';
    - FLUSH PRIVILEGES;
5. **Modify the Website to Use the Database:**
    - Update your index.php to connect to MySQL and optionally create a simple table or fetch the current time.

- Display a message that includes the visitor's IP address ($_SERVER['REMOTE_ADDR']) and the current time.

6. **Testing Locally:**
    - Access the website again and verify that it now shows the visitor's IP address and the current time.
7. **Make the Website Publicly Accessible Through a Cloud Provider:**
    - If you have your Linux server hosted on a cloud provider (e.g., AWS, GCP, Azure, DigitalOcean), configure the necessary firewall rules and security groups to allow inbound HTTP (port 80) and optionally HTTPS (port 443) traffic.
    - Obtain a public IP address from your cloud provider or a domain name that points to your server's public IP address (e.g., using a service like Amazon Route 53, Google Domains, or Cloudflare).
    - Update your DNS settings to point your domain (e.g. http://your-domain.com) to your server's public IP.
    - Confirm that you can access the site externally using http://your-domain.com in a web browser.

Sub-task #2: Git & GitHub

1. **Initialize Git Locally:**
    - In your project directory, run:
    - git init
2. **Create a .gitignore File:**
    - Add a .gitignore file to exclude sensitive files (e.g., database credentials, config files) and unnecessary files.
3. **Commit Your Documentation & Source Code:**
    - Create a Markdown file (e.g. README.md) detailing every step you took to install, configure, test, and make your website publicly accessible.
    - Commit your documentation and the website files:
    - git add .
    - git commit -m "Initial commit: Add documentation and website files"
4. **Create and Push to a GitHub Repository:**
    - Create a new repository on GitHub.
    - Add the GitHub repo as a remote and push:

- git remote add origin <your-github-repo-url>
- git push -u origin main

Sub-task #3: Networking Basics

In your README.md, explain:

1. **IP Address**: What it is and its purpose in networking.
2. **MAC Address**: What it is, its purpose, and how it differs from an IP address.
3. **Switches, Routers, and Routing Protocols**: Basic definitions and their roles in a network.
4. **Remote Connection to Cloud Instance**: Steps you would take to connect to your cloud-based Linux instance from a remote machine (e.g., using SSH).

Deliverables

1. **Documentation (Markdown File)**:
    - Step-by-step instructions of the entire setup and configuration process, including how you made the website accessible externally.
    - Include your GitHub repository link.
    - Include explanations of IP addresses, MAC addresses, switches, routers, routing protocols, and instructions for connecting remotely via SSH.
2. **GitHub Repository**:
    - Containing your source code, configuration files (excluding sensitive data), and documentation.

Evaluation Criteria

- **Clarity & Completeness**: Steps should be clear and logically organized.
- **Correct Git Usage**: Properly structured repository, meaningful commit messages, and use of .gitignore.
- **Networking Understanding**: Accurate explanations demonstrating a fundamental understanding of networking concepts.
- **Linux Administration Skills**: Proper installation/configuration of LAMP stack, database setup, and ensuring external accessibility.

When Completed:
Provide the README.md and the GitHub repository link. Also, ensure the website can be accessed via the public URL you have configured.