

Task Achievement: Set up a Web Server on a Linux Machine

Name : Ahmed Ibrahim Nassar

At First, I created an ec2 instance on AWS

Go to Launch an instance

set the instance name

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

SE-Intern-Ins.

Add additional tags

Then we choose the OS version that the server will run on.

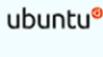
We chose Ubuntu Server 24.04 LTS 64 bit

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your Instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents **Quick Start**

Amazon Linux  macOS  **Ubuntu**  Windows  Red Hat  SUSE Linux  |  **Browse more AMIs**
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-0e2c8caa4b6378d8c (64-bit (x86)) / ami-0932ffb346ea84d48 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Username <small>i</small>	
64-bit (x86) 	ami-0e2c8caa4b6378d8c	ubuntu	Verified provider

Then create the pairing key with the pem file so that we can open the terminal from a Linux device (ssh)

Create key pair



Key pair name

Key pairs allow you to connect to your instance securely.

Enter key pair name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair

Private key file format

.pem

For use with OpenSSH

.ppk

For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

Create key pair

Then we choose the security group

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-0fc22addee8fe3f05f

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups [Info](#)

Select security groups

nassar sg-0a6e2ab48f0eca4ea

VPC: vpc-0fc22addee8fe3f05f

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Security Groups in AWS

Security Groups act as a virtual firewall for controlling inbound and outbound traffic to AWS resources like EC2 instances

Making the Website Public:

To make your website accessible to everyone on the internet, follow these steps:

1. Add an HTTP Rule:

- **Type:** HTTP
- **Port Range:** 80
- **Source:** 0.0.0.0/0 (allows all IP addresses).

Compare security group rules Info

Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic. You can select more security groups to view their inbound rules to help you to decide how to secure your instance from incoming traffic.

Common security groups

Select security groups ▾

nassar sg-0a6e2ab48f0eca4ea X
VPC: vpc-0fc22adee8fe3fb05

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Inbound rules (3)

Security group name	Security group ID	Type	Protocol	Port range	Source	Description
nassar	sg-0a6e2ab48f0eca4ea	HTTP	tcp	80	0.0.0.0/0	-
nassar	sg-0a6e2ab48f0eca4ea	ssh	tcp	22	41.47.195.232/32	-
nassar	sg-0a6e2ab48f0eca4ea	HTTPS	tcp	443	0.0.0.0/0	-

Then launch the instance.

Instances (1/1) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
SE-Intern-ins.	i-0ee01ec464677f9c1 (SE-Intern-ins.)	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-3-85-11-23.compute...	3.85.11.23	-	-

i-0ee01ec464677f9c1 (SE-Intern-ins.)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
IPv6 address	3.85.11.23 open address	Public IPv4 DNS
Answer private resource DNS name IPv4 (A)	Instance state	Private IPv4 addresses
IAM Role	Running	Public IPv4 DNS
IMDSv2	Instance type	Elastic IP addresses
Darnnif	t2.micro	AWS Compute Optimizer finding
	Instance ARN	Opt-in to AWS Compute Optimizer for recommendations. Learn more
	i-0ee01ec464677f9c1	Auto Scaling Group name
		Managed

Then we connect to the server from the device using the ssh, pairing key

with the command: `ssh -i anassar.pem ubuntu@ec2-3-85-11-23.compute-1.amazonaws.com`

```
a-nassar@a-nassar-VMware:~$ sudo ssh -i /home/a-nassar/Downloads/anassar.pem ubuntu@ec2-3-85-11-23.compute-1.amazonaws.com
[sudo] password for a-nassar:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Sun Dec 15 17:17:20 UTC 2024

System load: 0.0          Processes:           144
Usage of /:   42.5% of 6.71GB  Users logged in:    1
Memory usage: 65%          IPv4 address for enX0: 172.31.33.37
Swap usage:   0%

=> There are 4 zombie processes.

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

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

*** System restart required ***
Last login: Sun Dec 15 12:48:17 2024 from 41.47.195.232
ubuntu@ip-172-31-33-37:~$ S
```

First, to start this project, we need to:

- Update the Package Index: `sudo apt-get update`
- Install Apache: `sudo apt-get install apache2 -y`
- Install PHP: `sudo apt-get install php libapache2-mod-php php-mysql -y`
- Install MySQL: `sudo apt-get install mysql-server -y`

Verify that Apache, MySQL, and PHP are working properly after installation.

- Verify that Apache is working: Go to this link in my browser <http://3.85.11.23> {See default Apache welcome page.}
- Verify PHP is working: Create a PHP file to test it by using `sudo nano /var/www/html/info.php`

By adding the following code to the file:

```
<?php
phpinfo();
?>
```

And by opening the browser <http://3.85.11.23/info.php> You will see a page with information about your PHP settings.

- Verify MySQL is working: Log in to MySQL using the command `sudo mysql -u root -p`
To test the database connection using `SHOW DATABASES;`

```
ubuntu@ip-172-31-33-37:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 74
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
| web_db         |
+-----+
5 rows in set (0.05 sec)

mysql> EXIT;
Bye
```

Next, configure Apache to serve files from the /var/www/html/ directory (this directory is the default location from which Apache serves web pages).

Make sure Apache is using the /var/www/html/ directory from opining Apache default configuration file by
`sudo nano /etc/apache2/sites-available/000-default.conf`

And Searching for : `DocumentRoot /var/www/html`

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
```

After we have confirmed that the default Apache files are correct, we must restart it to make sure that everything is saved and working.

Restart by: `sudo systemctl restart apache2`

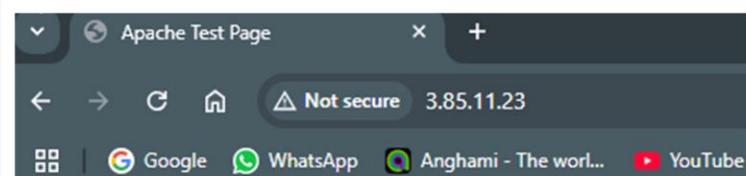
After making sure that everything is working properly, we start creating what is required to create our website.

First go to the directory /var/www/html/ by `cd /var/www/html/`

Create a file named index.html by `sudo nano index.html`

Create a simple website to test the file by adding this content to the file.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Apache Test Page</title>
  </head>
  <body>
    <h1>Apache is serving files correctly!</h1>
  </body>
</html>
```



Apache is serving files correctly!

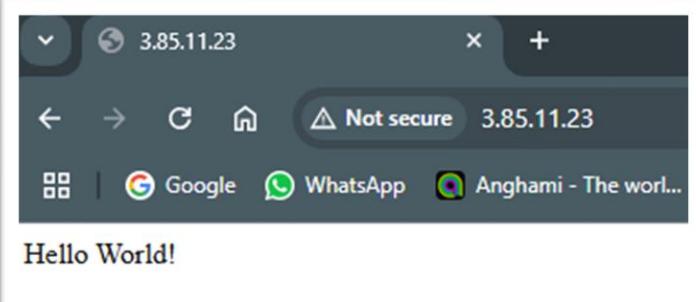
Create a Simple Website:-

Replace index.html with a PHP file (e.g. index.php) that displays "Hello World!".

At first go to /var/www/html/ by using `cd /var/www/html/`
delete index.html by using `sudo rm index.html`
Create a new PHP file named index.php by `sudo nano index.php`
Open the file using `nano` and put this code in it

```
<?php  
echo "Hello World!";  
?>
```

After saving the file, we enter the server's IP to find "Hello World!" on the browser page.



Configure MySQL:-

We ran the following command to run the MySQL security tool: `sudo mysql_secure_installation`

He asked me several questions:

Set the root password? (Y)
Remove anonymous users? (Y)
Disallow root login remotely? (Y)
Remove test database and access to it? (Y)
Reload privilege tables now? (Y)

After that, log in to MySQL using the root user using `sudo mysql -u root -p` insert my password

Create a new database using `CREATE DATABASE web_db;`

```
mysql> CREATE DATABASE web_db;  
Query OK, 1 row affected (0.02 sec)  
  
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| web_db |  
+-----+  
5 rows in set (0.00 sec)
```

`CREATE USER (web user)`

`CREATE USER 'web_user'@'localhost' IDENTIFIED BY 'Nassar@321456#';`

Grant all permissions to the user `web_user` on the database `web_db` by

`GRANT ALL PRIVILEGES ON web_db.* TO 'web_user'@'localhost';`

Update the privilege tables to activate the changes by `FLUSH PRIVILEGES;`

Log out of MySQL

Log in with new user {`web_user`} using `mysql -u web_user -p`

Make sure the database exists by `SHOW DATABASES;`

```
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| performance_schema |  
| web_db |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> exit  
Bye
```

Modify the Website to Use the Database:-

update index.php to:

1. Connect to MySQL.
2. Optionally create a table.
3. Fetch and display the visitor's IP address and the current time.

At first Update index.php

Open index.php by `sudo nano /var/www/html/index.php`

And add this code

```
<?php
    $servername = "localhost";
    $username = "web_user";
    $password = "Nassar@321456#";
    $dbname = "web_db";

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

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "CREATE TABLE IF NOT EXISTS visitors (
        id INT AUTO_INCREMENT PRIMARY KEY,
        ip_address VARCHAR(50) NOT NULL,
        visit_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    )";

    if ($conn->query($sql) === TRUE) {
        echo "Table 'visitors' is ready.<br>";
    } else {
        echo "Error creating table: " . $conn->error;
    }

    $visitor_ip = $_SERVER['REMOTE_ADDR'];
    $current_time = date('Y-m-d H:i:s');

    $sql = "INSERT INTO visitors (ip_address) VALUES ('$visitor_ip')";
    if ($conn->query($sql) === TRUE) {
        echo "Visitor logged successfully.<br>";
    } else {
        echo "Error logging visitor: " . $conn->error;
    }

    echo "Your IP address is: $visitor_ip<br>";
    echo "Current server time is: $current_time<br>";

    echo "Thank you 😊";
    $conn->close();
?>
```

Save and exit index.php

This code does the following:

- Connect to database: Uses server data (localhost), username, password, and database name to establish a connection with MySQL.
- Check connection: If the connection fails, an error message is displayed.
- Create table: A table named visitors This table contains:
 - Serial number (id).
 - Visitor's IP address (ip_address).
 - Visiting time (visit_time).
- Log visitor data:
 - Gets visitor's IP address and visiting time.
 - Stores IP address in database.
- Display visitor information:
 - Displays visitor's IP address.
 - Displays current time on server.
- Close connection: Terminates connection to database to free up resources.

Test your sit by using <http://3.85.11.23>

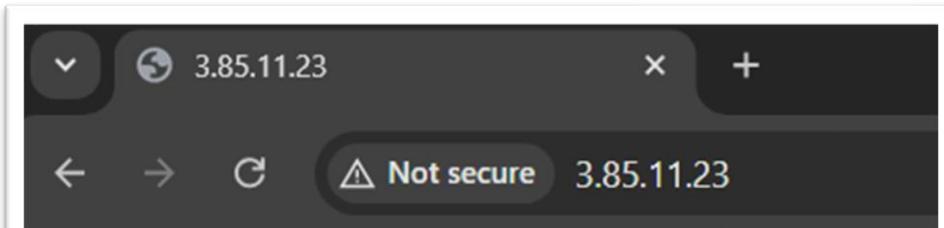


Table 'visitors' is ready.

Visitor logged successfully.

Your IP address is: 41.47.195.232

Current server time is: 2024-12-15 19:53:36

Thank you 😊

You can view the visitor log in the database from

Log in to My SQL `mysql -u web_user -p`

Then choose the database `web_db` `USE web_db;`

Then view the visitors' data with this command `SELECT * FROM visitors;`

And it will display the database.

Finally, the site is available to everyone through this link <http://3.85.11.23>

```
mysql> USE web_db;
Database changed
mysql> SELECT * FROM visitors;
+----+-----+-----+
| id | ip_address | visit_time |
+----+-----+-----+
| 1  | 41.47.195.232 | 2024-12-15 13:24:03 |
| 2  | 41.47.195.232 | 2024-12-15 13:24:06 |
| 3  | 41.47.195.232 | 2024-12-15 13:24:07 |
| 4  | 41.47.195.232 | 2024-12-15 13:24:08 |
| 5  | 41.47.195.232 | 2024-12-15 13:24:08 |
| 6  | 41.47.195.232 | 2024-12-15 13:24:08 |
| 7  | 41.47.195.232 | 2024-12-15 13:24:08 |
| 8  | 41.47.195.232 | 2024-12-15 13:24:09 |
| 9  | 41.47.195.232 | 2024-12-15 13:24:09 |
| 10 | 41.47.195.232 | 2024-12-15 13:24:10 |
| 11 | 41.47.195.232 | 2024-12-15 13:24:11 |
| 12 | 41.47.195.232 | 2024-12-15 13:24:21 |
| 13 | 41.47.195.232 | 2024-12-15 13:24:22 |
| 14 | 41.47.195.232 | 2024-12-15 13:24:22 |
| 15 | 41.47.195.232 | 2024-12-15 13:24:23 |
| 16 | 41.47.195.232 | 2024-12-15 13:24:24 |
| 17 | 41.47.195.232 | 2024-12-15 13:24:24 |
| 18 | 41.44.56.9   | 2024-12-15 13:25:22 |
| 19 | 41.47.195.232 | 2024-12-15 13:25:25 |
| 20 | 41.47.195.232 | 2024-12-15 13:31:06 |
+----+-----+-----+
20 rows in set (0.00 sec)

mysql> EXIT;
Bye
```

Git & GitHub:-

In my project directory, run git init

This command creates a hidden folder named .git that contains Git's version management files.

Create a .gitignore file that excludes sensitive and unnecessary files from tracking by `touch .gitignore`

Access the file by `nano .gitignore`,

And add files to ignore them

`.env`

`config.php`

`*.log`

Then save the file changes

Using the following command, all files (except those excluded in .gitignore) are added to the tracking area.

`git add .` {The dot (.) means "add all files in this folder"}

`git commit -m "Initial your commit"`

then Create a new repository on GitHub.

The local Git repository is linked to the GitHub repository on the server, using the following command:

`git remote add origin https://github.com/ain931/my-lamp-project.git`

After connecting git to github you need to push the code to GitHub by.

`git push -u origin main`

After the push, he asked me for my username and password, so I entered them.

```
ubuntu@ip-172-31-33-37:/var/www/html$ sudo git push -u origin main
Username for 'https://github.com': ain931
Password for 'https://ain931@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 1.03 KiB | 1.03 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ain931/my-lamp-project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

GitHub sync:

to update code and GitHub sync

run this command in the directory `git add .` to add all files in this folder after updated thin write the commit by

`git commit -m "Your Updates"` Then push Updates at github by `git push`.

{If you want to pull the updates from GitHub run this command `git pull origin main`}

Networking Basics:-

What does an IP address mean?

An IP address is a unique number for each device connected to a network, such as a computer or mobile phone, so that they can communicate with each other.

Its benefits in networks:

- Device identification: It uniquely identifies each device on the network.
- Location identification: It identifies the location of the device so that the data is delivered correctly.
- Data routing: It helps the router send information to the device that needs it.
- Facilitating communication: It allows devices to talk to each other on the network or the Internet.

Types of IP addresses:

IPv4 such as (192.168.1.1), and this is a limited number.

IPv6 such as (2001:0db8:85a3::8a2e:0370:7334), and this is a very large number that suits the needs of the era.

Its importance:

Without it, devices will not be able to communicate or send data on the network or the Internet.

What does MAC Address mean?

The MAC Address is a fixed and unique number written on the network card (NIC) in any device. This number is given by the manufacturer and does not change

Its benefit:

Local device identification: It identifies the device on the local network (LAN).

Data routing: Switches use it to send data to the correct device within the network.

Work at the local network level: It works in small networks such as those in a home or company.

point	IP Address	MAC Address
Task	The device number on the network (like a home address).	The device number itself (like a fingerprint).
Scope	Works on Internet or between networks.	Works on LAN.
Stability	It changes depending on the network or router.	Stability and does not change.

MAC identifies the device itself.

IP identifies the device on the network.

The two work together to deliver data to your device in the right way

What does Switch mean?

A Switch is a device that connects devices on the same local area network (LAN) together. Its function is to receive data from one device and send it to the correct device based on the MAC Address.

What does Router mean?

A Router is a device that connects networks together. Its function is to direct data from one network to another (such as from the home network to the Internet) using the IP Address.

What does Routing Protocols mean?

These are rules or protocols that the Router uses to choose the best path to deliver data between networks.

Examples include:

OSPF (chooses the fastest path).

BGP (for large networks such as the Internet).

Switch: Connects devices within the same network.

Router: Connects networks together and directs data.

Routing Protocols: Determine the correct path for data between networks.

Finally, the verification sources

Github Repositories link : <https://github.com/ain931/my-lamp-project>

Website link : <http://3.85.11.23>

Screen from pushing the project README file:

```
ubuntu@ip-172-31-33-37:/var/www/html$ sudo git add README.md
ubuntu@ip-172-31-33-37:/var/www/html$ sudo git commit -m "README.md file upload"
[main a42be37] README.md file upload
Committer: root <root@ip-172-31-33-37.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 185 insertions(+)
create mode 100644 README.md
ubuntu@ip-172-31-33-37:/var/www/html$ sudo git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 2.37 KiB | 2.37 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ain931/my-lamp-project.git
    a903b66..a42be37 main -> main
ubuntu@ip-172-31-33-37:/var/www/html$ █
```