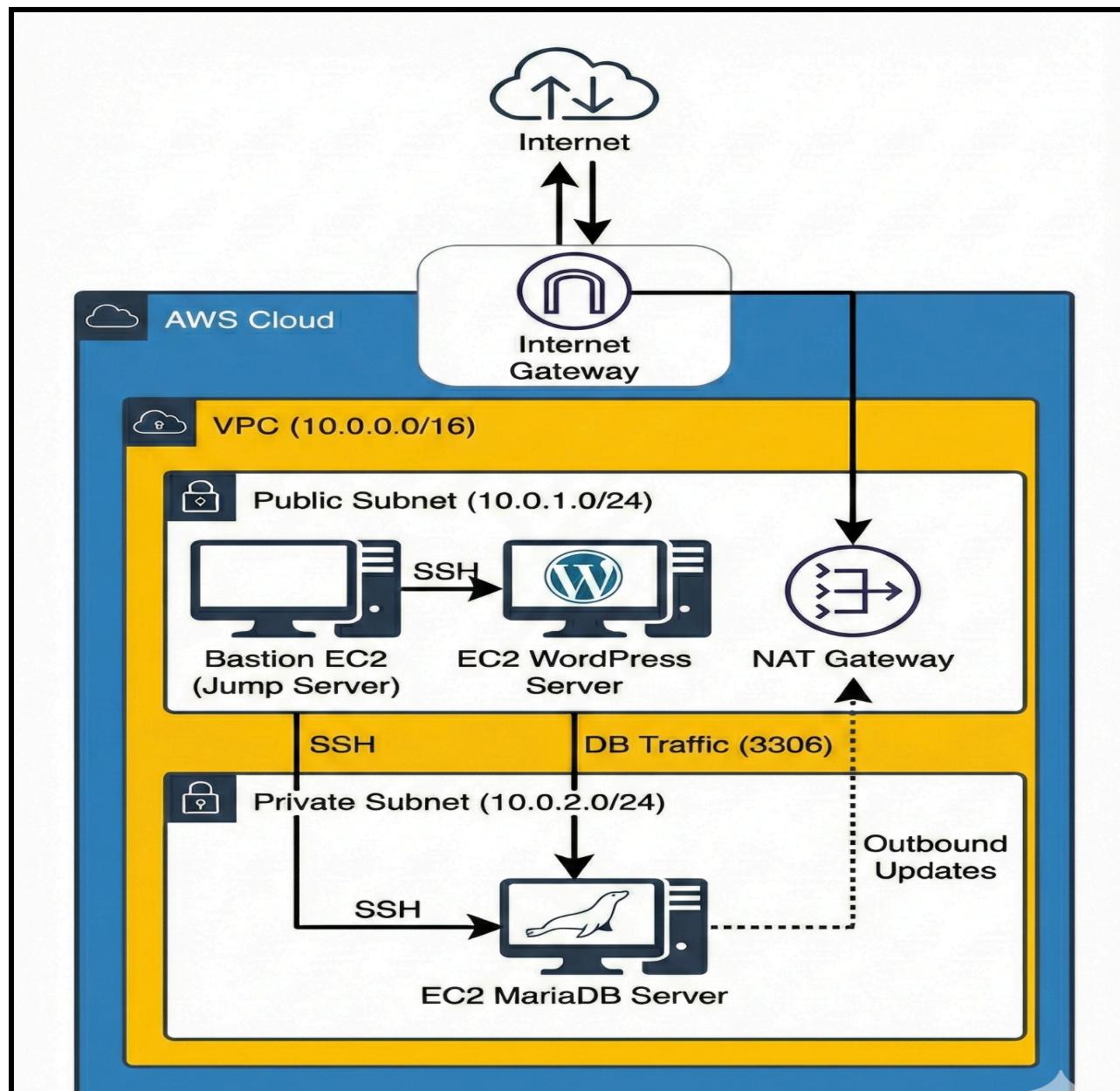


AWS Multi-Tier Architecture for WordPress with Bastion Host and NAT Gateway



1. Objective

The goal of this project is to deploy a secure, scalable WordPress application on AWS using a multi-tier architectural approach. By isolating the database in a private subnet and utilizing a Jump Server (Bastion Host), we ensure that sensitive infrastructure is protected from direct public internet exposure.

Key Features

- **Jump Server:** Secure SSH entry point for administrative tasks.
- **Three-Tier Logic:** Web tier in public subnets, Data tier in private subnets.
- **NAT Gateway:** Enables private instances to download updates without being reachable from the internet.
- **Enhanced Security:** Granular Security
- **Group rules to restrict traffic flow**

2. Architecture Overview

Network Flow

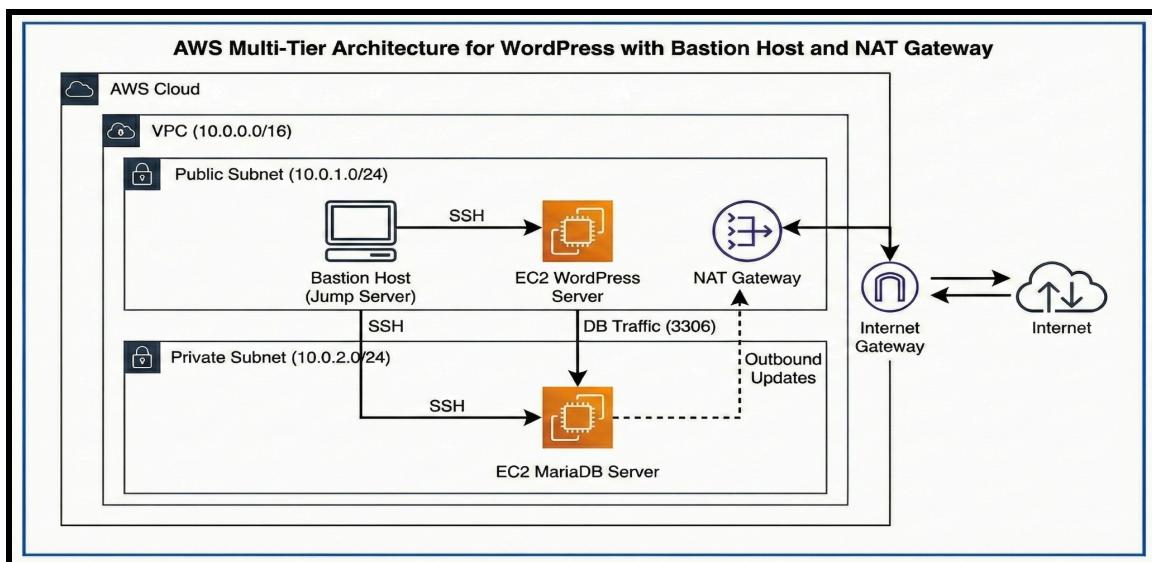
Internet <-> Internet Gateway <-> VPC

Public Subnet (10.0.1.0/24)

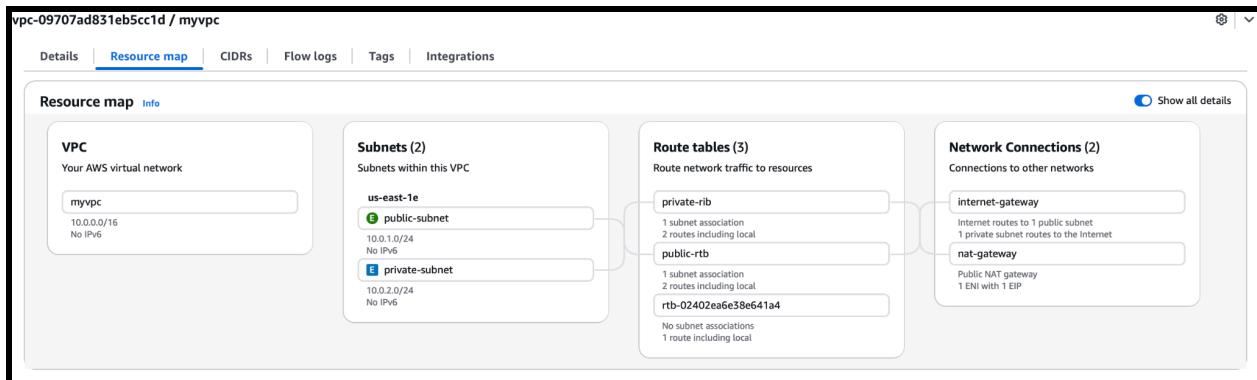
- **Jump Host:** SSH entry point.
- **WordPress EC2:** Running Apache + PHP.
- **NAT Gateway:** Attached to an Elastic IP for outbound private traffic.

Private Subnet (10.0.2.0/24)

- **MariaDB EC2:** No public IP; accessible only via Jump Host and Web Server.



3. Infrastructure Setup



STEP 1: VPC Creation

1. Navigate to the VPC Dashboard.
2. Select Create VPC.
3. Name: WP-VPC
4. IPv4 CIDR: 10.0.0.0/16

The screenshot shows the 'Your VPCs' list. A new VPC named 'myvpc' has been created, matching the specifications from the previous step.

Name	VPC ID	State	Encryption c...	Encryption control...	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-031eaf60467122985	Available	-	-	Off	172.31.0.0/16	-
myvpc	vpc-09707ad831eb5cc1d	Available	-	-	Off	10.0.0.0/16	-

STEP 2: Subnet Creation

- **Public Subnet:**
 - **Name: Public-Subnet**
 - **AZ: ap-south-1a**
 - **CIDR: 10.0.1.0/24**
- **Private Subnet:**
 - **Name: Private-Subnet**
 - **AZ: ap-south-1a**
 - **CIDR: 10.0.2.0/24**

The screenshot shows the 'Subnets' list. Two subnets have been created: 'public-subnet' and 'private-subnet', each associated with the 'myvpc' VPC and matching the specified CIDRs.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
public-subnet	subnet-064fcf0489c5301cd	Available	vpc-09707ad831eb5cc1d myvpc	Off	10.0.1.0/24	-
private-subnet	subnet-01ce5609fb8c82cba	Available	vpc-09707ad831eb5cc1d myvpc	Off	10.0.2.0/24	-

STEP 3: Internet Gateway (IGW)

1. Create IGW named WP-IGW.
2. Attach WP-IGW to WP-VPC.

Internet gateways (1/2) Info					
Find internet gateways by attribute or tag					
Name	Internet gateway ID	State	VPC ID	Owner	Actions
-	igw-0eecd723e07cf7c45	Attached	vpc-031eaf60467122985	767398111127	Actions
<input checked="" type="checkbox"/> internet-gateway	igw-0f9e2995d5d31fc84	Attached	vpc-09707ad831eb5cc1d myvpc	767398111127	Actions

STEP 4: Route Table Configuration

- **Public-RT:**
 - Routes: 0.0.0.0/0 -> WP-IGW
 - Association: Public-Subnet
- **Private-RT:**
 - Routes: 10.0.0.0/16 -> local (Initial state)
 - Association: Private-Subnet

Route tables (2/4) Info					
Last updated 1 minute ago Actions Create route table					
Find route tables by attribute or tag					
Name	Route table ID	Explicit subnet assoc...	Edge associa		
-	rtb-002442b59e47a54aa	-	-		
-	rtb-02402ea6e38e641a4	-	-		
<input checked="" type="checkbox"/> public-rtb	rtb-011a90470a07bff8e	subnet-064fc0489c5301...	-		
<input checked="" type="checkbox"/> private-rib	rtb-09bee00259a68e1ce	-	-		

UPDATE PUBLIC-RT ROUTES :

VPC > Route tables > rtb-011a90470a07bff8e > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	local	-	No	CreateRoute
	Internet Gateway	-		
	igw-0f9e2995d5d31fc84	-		

[Add route](#) [EditRoutes](#) [Cancel](#) [Preview](#) [Save changes](#)

STEP 5: NAT Gateway Setup

1. Elastic IP: Allocate a new EIP.
2. NAT Gateway: Name: nat-gateway
 - Subnet: Public-Subnet

- EIP: Select the allocated EIP.

Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability mode [Info](#)
Choose whether to deploy across all zones in the region or restrict to a single availability zone.

Regional - new
Scales automatically across all regional AZs, simplifying management for multi AZ deployments.

Zonal
Provides granular control within a specific availability zone, adhering to subnet level settings.

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.

Public

Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.
 [Allocate Elastic IP](#)

► Additional settings [Info](#)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> X	<input type="text" value="nat-gateway"/> X Remove

[Add new tag](#)
You can add 49 more tags.

3. Update Private-RT: Add route 0.0.0.0/0 -> NAT Gateway (nat-gateway).

Edit routes

Destination	Target	Status	Propagated	Route Origin	
10.0.0.0/16	<input type="text" value="local"/> X	<input checked="" type="radio"/> Active	No	CreateRouteTable	
<input type="text" value="0.0.0.0"/> X	<input type="text" value="NAT Gateway"/> X	-	No	CreateRoute	Remove
Add route					

[EditRoutes](#) [Cancel](#) [Preview](#) [Save changes](#)

4. Security & Compute

STEP6: Security Group Configuration

Security Group	Inbound Rules	Source
Jump-SG	SSH (22)	Your Public IP (/32)
WP-SG	SSH (22) HTTP (80)	Jump-SG 0.0.0.0/0
DB-SG	SSH (22) MySQL (3306)	Jump-SG WP-SG

sg-0e85b2734d997e7ff - jumpserver-sg

Details | **Inbound rules** | Outbound rules | Sharing | VPC associations | Tags

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0c840b898d395b6f1	IPv4	SSH	TCP	22	106.210.229.240/32	-

Basic details

Security group name [Info](#)
wordpress-sg

Name cannot be edited after creation.

Description [Info](#)
security group for Wordpress instance

VPC Info
vpc-09707ad831eb5cc1d (myvpc) [CreateSecurityGroup](#)

Inbound rules [Info](#)

Inbound rule 1 [Delete](#)

Type Info SSH	Protocol Info TCP	Port range Info 22
Source type Info Custom	Source Info <input type="text"/> sg-0e85b2734d997e7ff X	Description - optional Info

Inbound rule 2 [Delete](#)

Type Info HTTP	Protocol Info TCP	Port range Info 80
Source type Info Anywhere-IPv4	Source Info <input type="text"/> 0.0.0.0/0 X	Description - optional Info

Basic details

Security group name [Info](#)

mariadb-sg

Name cannot be edited after creation.

Description [Info](#)

to access mariadb privately

VPC [Info](#)

vpc-09707ad831eb5cc1d (myvpc)



Inbound rules [Info](#)

Inbound rule 1

[Delete](#)

Type [Info](#)

[CreateSecurityGroup](#)

MySQL/Aurora

TCP

Port range [Info](#)

3306

Source type [Info](#)

Custom

Source [Info](#)



Description - optional [Info](#)

sg-0d27faf74ac16c1d0



Inbound rule 2

[Delete](#)

Type [Info](#)

SSH

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Custom

Source [Info](#)



Description - optional [Info](#)

sg-0e85b2734d997e7ff



Step7: EC2 Instance Launch

General Specs: Amazon Linux 2023, t2.micro, wp-key.pem.

1. **Jump Host:** Public Subnet, Auto-assign Public IP: Enabled, SG: Jump-SG.
2. **WordPress Server:** Public Subnet, Auto-assign Public IP: Enabled, SG: WP-SG.
3. **MariaDB Server:** Private Subnet, Auto-assign Public IP: Disabled, SG: DB-SG.

Instances (3) Info				
Last updated less than 1 hour ago				
Instances		Connect	Instance state	Actions
<input type="text"/> Find Instance by attribute or tag (case-sensitive)				All states
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	WP-Server	i-025b59c885fd019be	Running	t2.micro
<input type="checkbox"/>	DB-Server	i-072c0e596b24c06be	Running	t2.micro
<input type="checkbox"/>	jump-server	i-0bda1bc84e1ee3d37	Running	t2.micro

5. Deployment & Configuration

STEP 8: SSH Access Flow

Securely hop from your machine to the private infrastructure:

Set permissions

```
#chmod 400 wp-key.pem
```

Connect to Jump Host

```
#ssh -i wp-key.pem ec2-user@JUMP_PUBLIC_IP
```

```
mayurnikam@MAYURs-MacBook-Air Downloads % chmod 400 "key-2.pem"
mayurnikam@MAYURs-MacBook-Air Downloads % ssh -i "key-2.pem" ec2-user
@54.160.93.52
,      #
~\_\_ #####_          Amazon Linux 2023
~~ \_\#####\_
~~   \###|
~~     \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~       \~' '->
~~~      /
~~_. _/ \
~/_ / \
~/m/ '
```

Last login: Wed Jan 21 16:10:28 2026 from 106.210.229.240
[ec2-user@ip-10-0-1-121 ~]\$

From Jump Host to WordPress

```
ssh -i wp-key.pem ec2-user@WP_PRIVATE_IP
```

```
[ec2-user@ip-10-0-1-121 ~]$ chmod 400 key-2.pem
[ec2-user@ip-10-0-1-121 ~]$ ssh -i key-2.pem ec2-user@10.0.1.177
'      #_
~\_\_ #####_          Amazon Linux 2023
~~ \_\_#####\
~~   \###|
~~     \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '-->
~~~           /
~~_.  _/ \
~/_/_/ \
[m/'

[ec2-user@ip-10-0-1-177 ~]$
```

3. From Jump Host to MariaDB

```
ssh -i wp-key.pem ec2-user@DB_PRIVATE_IP
```

```
● ○ ● Downloads — ec2-user@ip-10-0-1-177:~ — ssh -i key-2.pem ec2-user@54.160.93.52 — 65x36
[ec2-user@ip-10-0-1-121 ~]$ ssh -i key-2.pem ec2-user@10.0.1.177
'      #_
~\_\_ #####_          Amazon Linux 2023
~~ \_\_#####\
~~   \###|
~~     \#/ ___ https://aws.amazon.com/linux/amazon-linux-20
23
~~       V~' '-->
~~~           /
~~_.  _/ \
~/_/_/ \
[m'

Last login: Wed Jan 21 16:14:56 2026 from 10.0.1.121
[ec2-user@ip-10-0-1-177 ~]$
```

STEP 9: MariaDB Installation (Private Subnet)

1. **Install Mariadb:** #sudo dnf update -y && sudo dnf install mariadb105-server -y
2. **Service:** #sudo systemctl start mariadb && sudo systemctl enable mariadb
3. **Secure:** #sudo mariadb-secure-installation
4. **Database Setup:**

```
CREATE DATABASE wpdb;
CREATE USER 'wpuser'@'%' IDENTIFIED BY 'password123';
GRANT ALL PRIVILEGES ON wpdb.* TO 'wpuser'@'%';
FLUSH PRIVILEGES;
```

```
[ec2-user@ip-10-0-1-177 ~]$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.5.29-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> CREATE DATABASE wpdb;
Query OK, 1 row affected (0.000 sec)

[ MariaDB [(none)]> CREATE USER 'wpuser'@'%' IDENTIFIED BY 'password123';
Query OK, 0 rows affected (0.002 sec)

[ MariaDB [(none)]> GRANT ALL PRIVILEGES ON wpdb.* TO 'wpuser'@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

STEP 10: WordPress Installation

1. **Packages:** sudo dnf install httpd php php-mysqlnd wget unzip -y
2. **Download:**

```
#cd /var/www/html
#sudo wget https://wordpress.org/latest.zip
#sudo unzip latest.zip
#sudo chown -R apache:apache wordpress
```

3. **Config:**

```
#cd wordpress
#sudo cp wp-config-sample.php wp-config.php
#sudo vi wp-config.php
# Update DB_NAME, DB_USER, DB_PASSWORD, and DB_HOST (MariaDB Private IP)
```

```
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://developer.wordpress.org/advanced-administration/
wordpress/wp-config/
*
* @package WordPress
*/
// ** Database settings - You can get this info from your web host
** */
/** The name of the database for WordPress */
define( 'DB_NAME', 'wpdb' );

/** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'password123' );

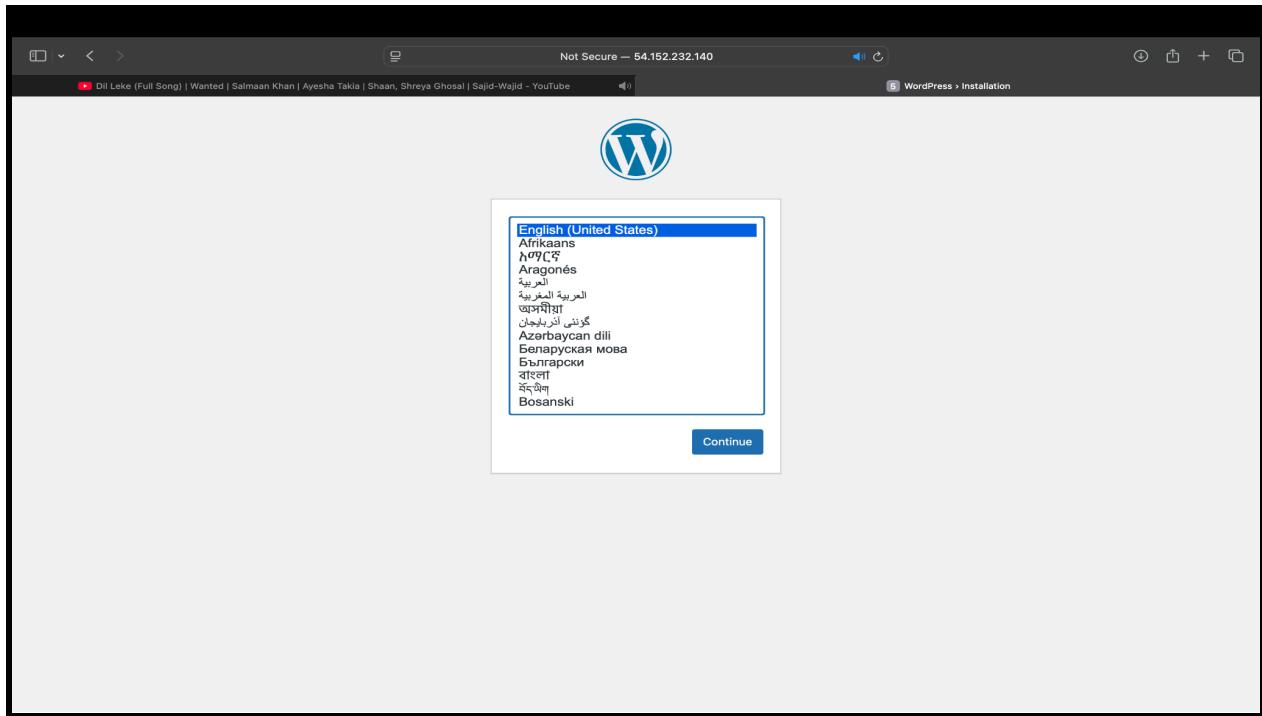
/** Database hostname */
define( 'DB_HOST', '10.0.2.172' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate the
```

Phase 12: Final Access



A screenshot of the WordPress dashboard. The title bar shows "Not Secure — 54.152.232.140" and the URL "Dashboard < mayur nikam — WordPress". The dashboard features a "Welcome to WordPress!" banner with the text "Learn more about the 6.9 version.". On the left, there's a sidebar with navigation links: Home, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and Collapse Menu. The main content area includes sections for "Author rich content with blocks and patterns", "Customize your entire site with block themes", and "Switch up your site's look & feel with Styles". It also shows "Site Health Status" with the message "No information yet..." and "At a Glance" with 1 Post and 1 Page. On the right, there's a "Quick Draft" section with fields for Title and Content.