

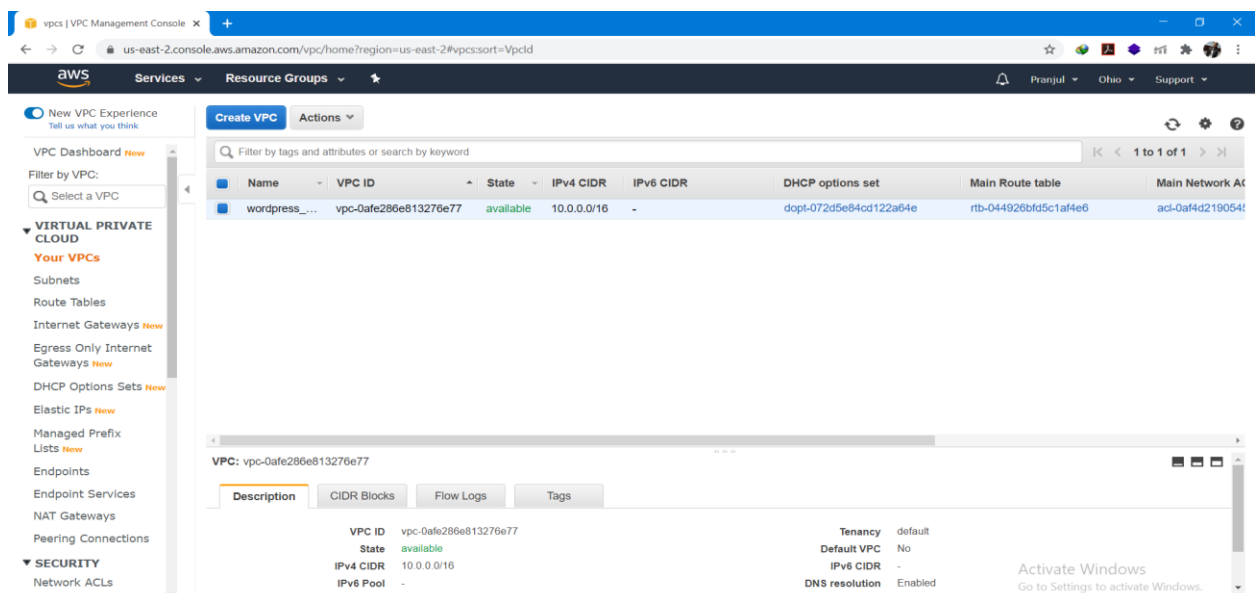
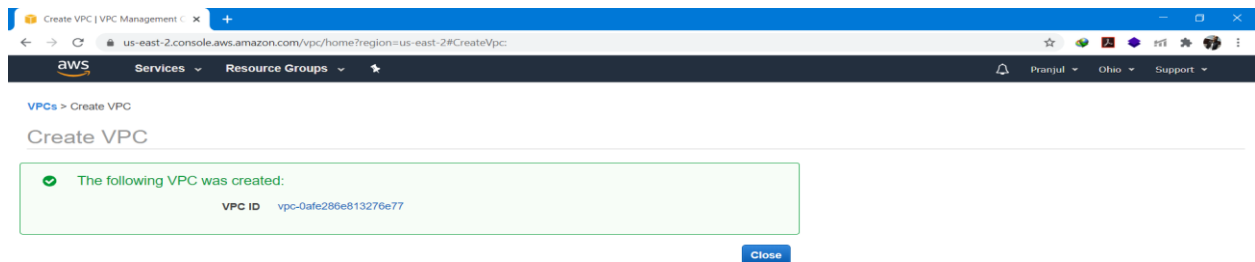
2-Tier Application - Wordpress and MySQL

- Launch 1 EC2 Linux instance and configure MySQL/Mariadb server on it (use your choice for password and user names).
- Launch 1 EC2 Linux instance and configure Wordpress Application server on it.
- Use previously configured MySQL or Mariadb database server for connection with private IP address of instances.

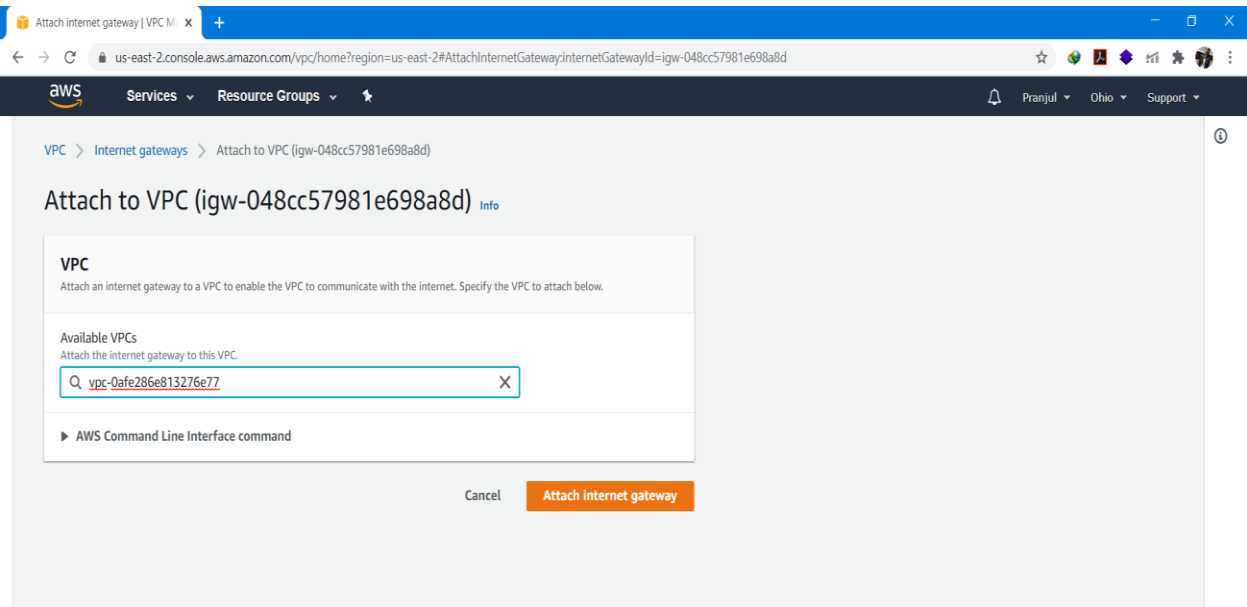
Steps:

Create Virtual Private Cloud(VPC)

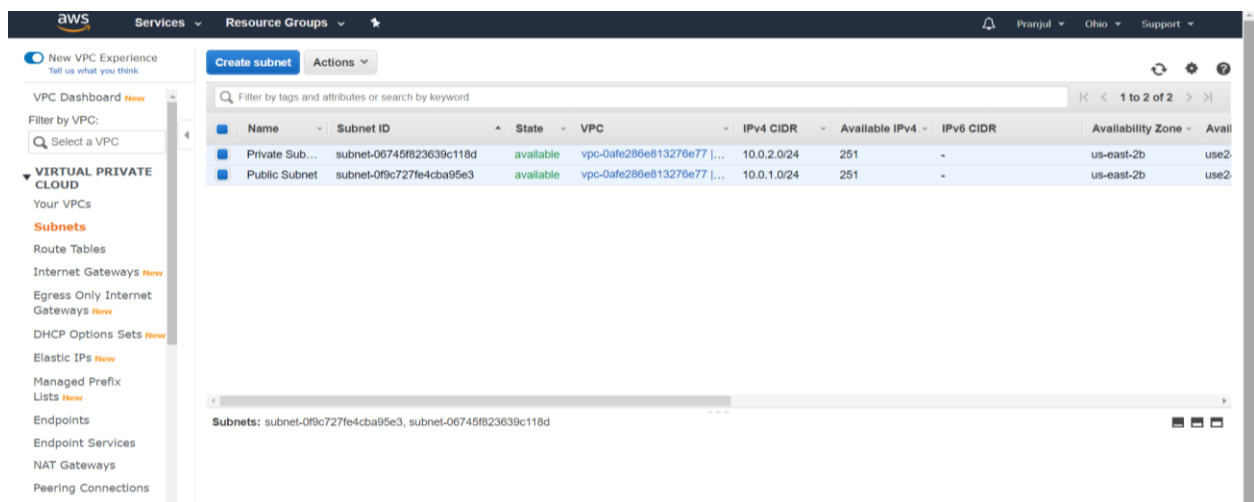
- 1- Launch VPC
- 2- Create VPC and enter the name tag, IPv4 CIDR block
- 3- Now VPC is created successfully



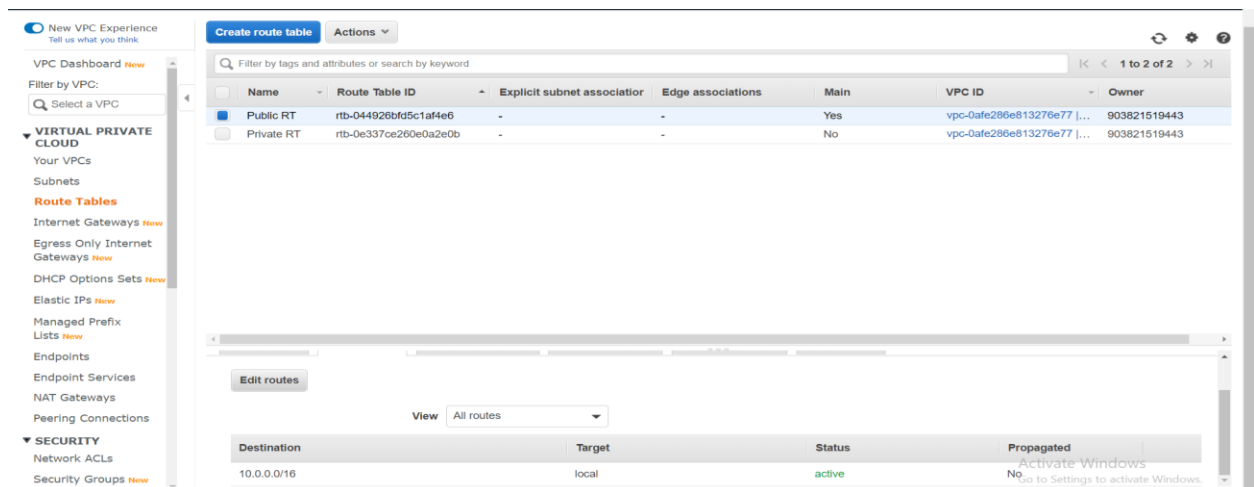
- 4- Create Internet Gateway
Give Name tag my_wordpress_gw. Click on create internet gateway
- 5- Attach to VPC



- 6- Create Subnets- public subnet and private subnet



- 7- Create route tables- Private RT and Public RT



8- Edit both public and private routes by clicking Edit routes > edit subnet associations

us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#EditRoutes:routeTableId=rtb-044926bfd5c1af4e6

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-048cc57981e698a8d		No

Add route

* Required

Cancel Save routes

Route Tables > Edit subnet associations

Edit subnet associations

Route table rtb-044926bfd5c1af4e6 (Public RT)

Associated subnets subnet-0f9c727fe4cba95e3

Filter by attributes or search by keyword

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0f9c727fe4cba95e3 Public Sub...	10.0.1.0/24	-	Main
subnet-06745f823639c118d Private Su...	10.0.2.0/24	-	Main

* Required

Cancel Save

New VPC Experience

VPC Dashboard

Filter by VPC: Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Managed Prefix Lists

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

SECURITY

Create route table Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
Public RT	rtb-044926bfd5c1af4e6	subnet-0f9c727fe4cba95e3	-	Yes	vpc-0afe286e813276e77	903821519443
Private RT	rtb-0e337ce260e0a2e0b	-	-	No	vpc-0afe286e813276e77	903821519443

Route Table: rtb-0e337ce260e0a2e0b

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No

Route Tables > Edit subnet associations

Edit subnet associations

Route table rtb-0e337ce260e0a2e0b (Private RT)

Associated subnets subnet-06745f823639c118d

Filter by attributes or search by keyword

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0f9c727fe4cba95e3 Public Sub...	10.0.1.0/24	-	rtb-044926bfd5c1af4e6
subnet-06745f823639c118d Private Su...	10.0.2.0/24	-	Main

* Required

Cancel Save

9- Go to Security Groups

The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The left sidebar contains navigation links for various AWS services, including Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, and a section for SECURITY (Network ACLs, Security Groups) and VIRTUAL PRIVATE NETWORK (VPN) (Customer Gateways, Virtual Private Gateways, Site-to-Site VPN). The main content area displays the 'Security Groups (1/1)' list with a table containing columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. The 'wordpress_sg' group is selected, showing its details: ID 'sg-0199bbc9100a5115f', name 'default', VPC ID 'vpc-0afe286e813276e77', and description 'default VPC security gr...'. Below the table, there are tabs for 'Details', 'Inbound rules', 'Outbound rules', and 'Tags'. The 'Inbound rules' tab is active, showing a list of rules. A note at the bottom states: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.'

Name	Security group ID	Security group name	VPC ID	Description	Owner
wordpress_sg	sg-0199bbc9100a5115f	default	vpc-0afe286e813276e77	default VPC security gr...	903821515

10- Edit inbound rules

The screenshot shows the 'Edit inbound rules' page for the 'wordpress_sg' security group. The page displays a table of inbound rules with columns: Type, Protocol, Port range, Source, and Description - optional. There are two rules listed: one for 'All traffic' from 'All' sources, and another for 'All traffic' from 'All' sources. The 'Add rule' button is visible. A note at the bottom states: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.'

Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	Custom	
All traffic	All	All	Custom	My task IP

The screenshot shows the 'Create network ACL' page. The left sidebar contains navigation links for various AWS services, including DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, and a section for SECURITY (Network ACLs, Security Groups) and VIRTUAL PRIVATE NETWORK (VPN) (Customer Gateways, Virtual Private Gateways, Site-to-Site VPN, Client VPN Endpoints). The main content area displays the 'Create network ACL' page with a table of network ACLs. The 'wordpress_rt' ACL is selected, showing its details: ID 'acl-0af4d219054540d12', name 'wordpress_rt', associated with '2 Subnets', default 'Yes', VPC ID 'vpc-0afe286e813276e77', and owner 'wordpress_VPC'. Below the table, there are tabs for 'Details', 'Inbound Rules', 'Outbound Rules', 'Subnet associations', and 'Tags'. The 'Inbound Rules' tab is active, showing a list of rules. A note at the bottom states: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.'

Name	Network ACL ID	Associated with	Default	VPC	Owner
wordpress_rt	acl-0af4d219054540d12	2 Subnets	Yes	vpc-0afe286e813276e77	wordpress_VPC

Network ACLs > Edit inbound rules

Edit inbound rules

Network ACL: `acl-0af4d219054540d12`

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	14.102.21.85/32	ALLOW

[Add Rule](#)

* Required [Cancel](#) [Save](#)

Create 2 EC2 instances

1- One is Wordpress and another is SQL database

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search:

Quick Start (0)

My AMIs (0)

AWS Marketplace (142)

Community AMIs (733)

Categories

All Categories

Infrastructure Software (11)

DevOps (109)

Business Applications (95)

Industries (8)

WordPress Certified by Bitnami and Automattic

★★★★★ (119) | 5.4.2-1 on Debian 10 | [Previous versions](#) | By [Bitnami](#)

Linux/Unix, Ubuntu 10 | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 7/7/20

WordPress is the world's most popular content management platform. It includes the new Gutenberg editor and over 45,000 themes and plugins. This image is certified by Bitnami as secure, up-to-date, and packaged using industry best practices, and approved by Automattic, the experts behind WordPress.

[More info](#)

[Select](#)

WordPress with NGINX and SSL Certified by Bitnami and Automattic

★★★★★ (10) | 5.4.2-3 on Debian 10 | [Previous versions](#) | By [Bitnami](#)

Linux/Unix, Ubuntu 10 | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 7/2/20

WordPress with NGINX and SSL enhances WordPress with SSL auto-configuration and the high-performance NGINX web server. This image is certified by Bitnami as secure, up-to-date, and packaged using industry best practices, and approved by Automattic, the experts behind WordPress.

[Select](#)

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

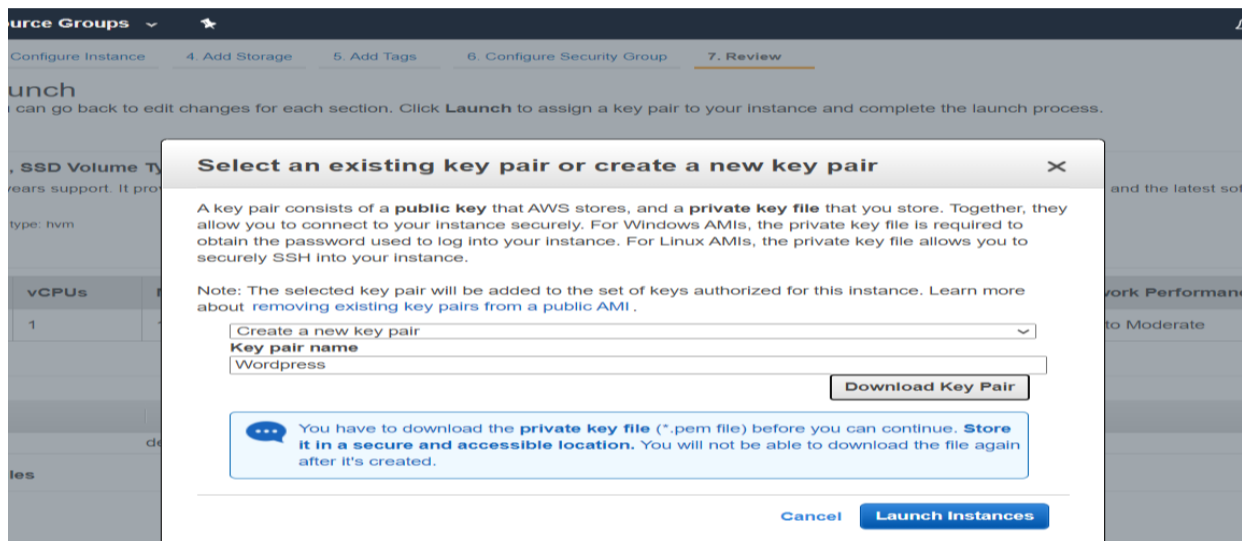
[Add Rule](#)

Warning

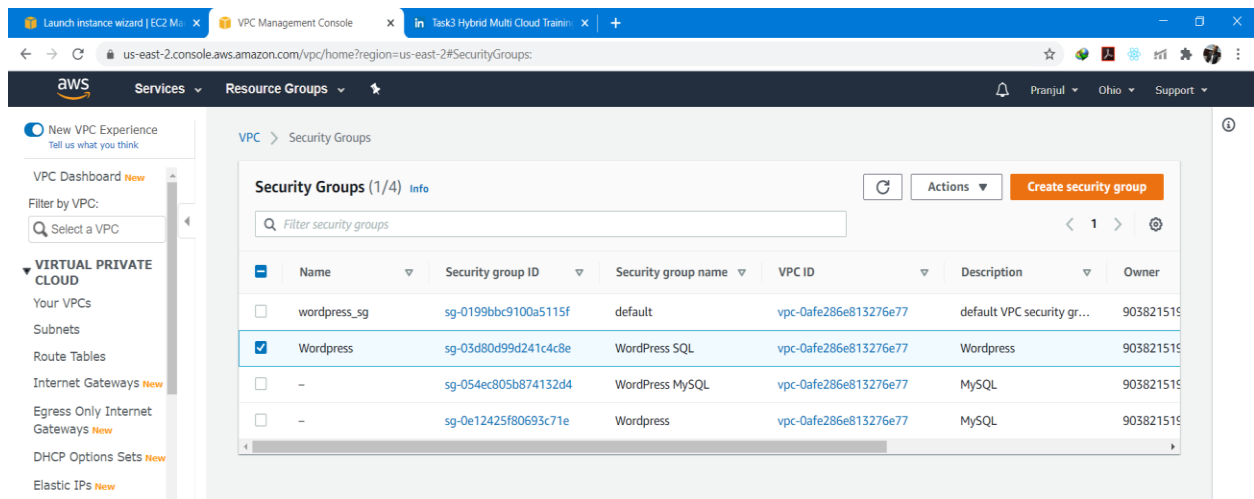
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

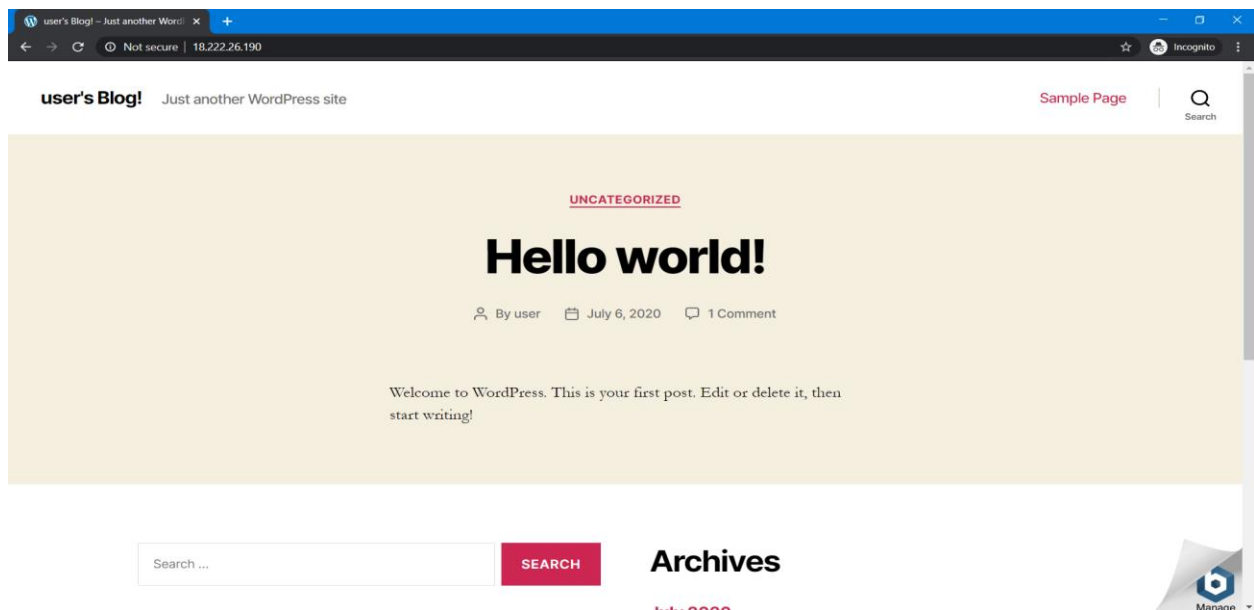
2- Review and launch



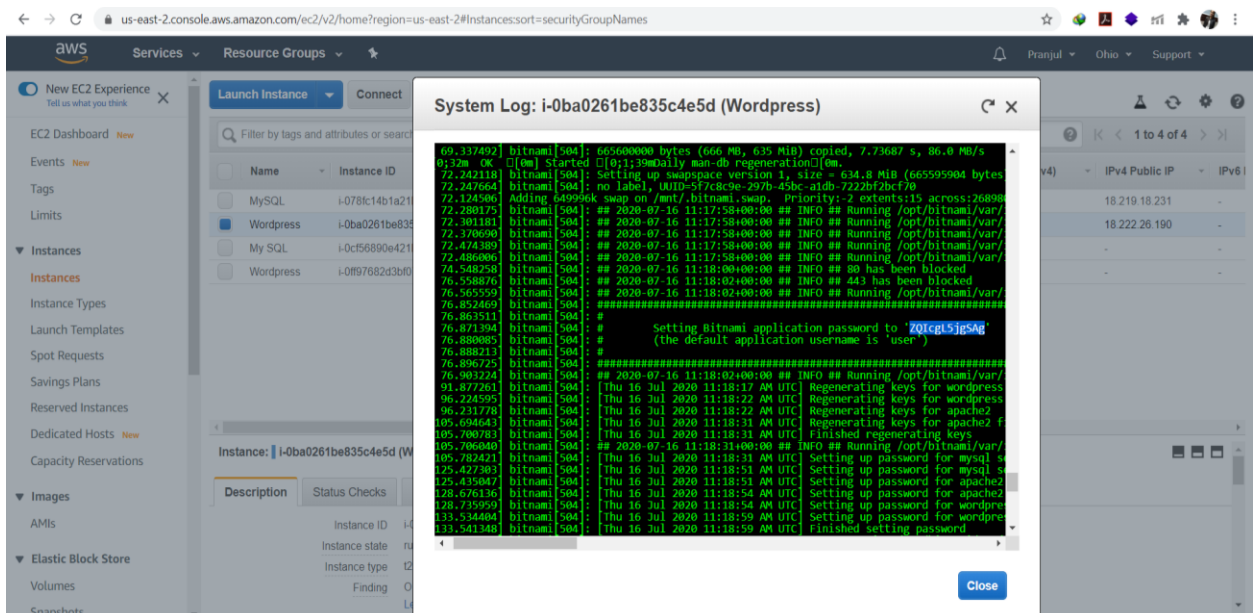
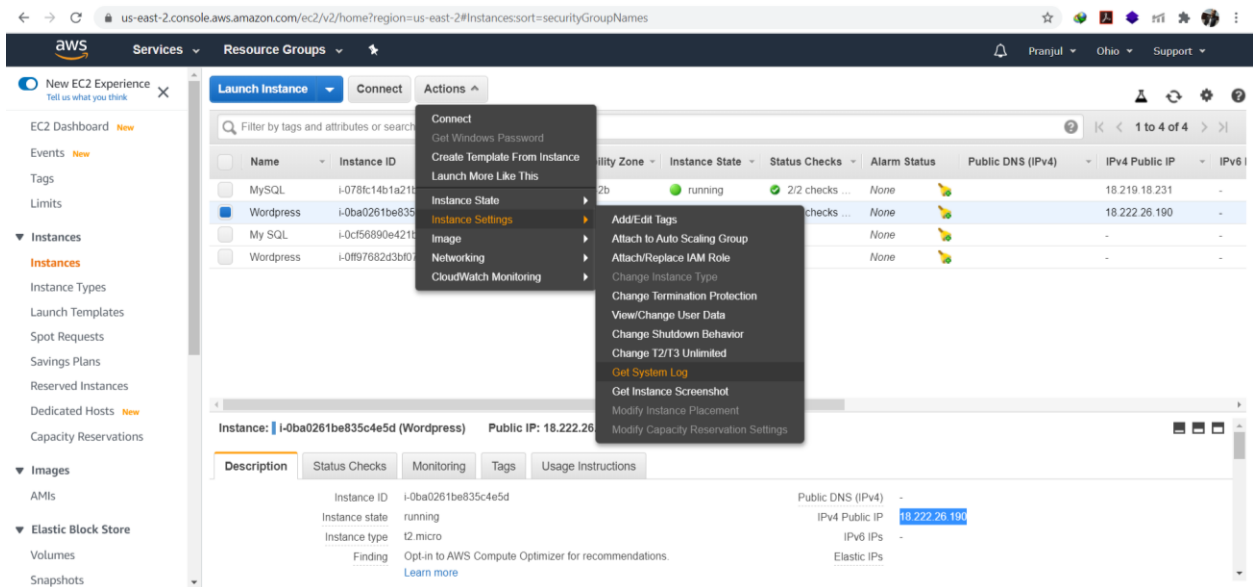
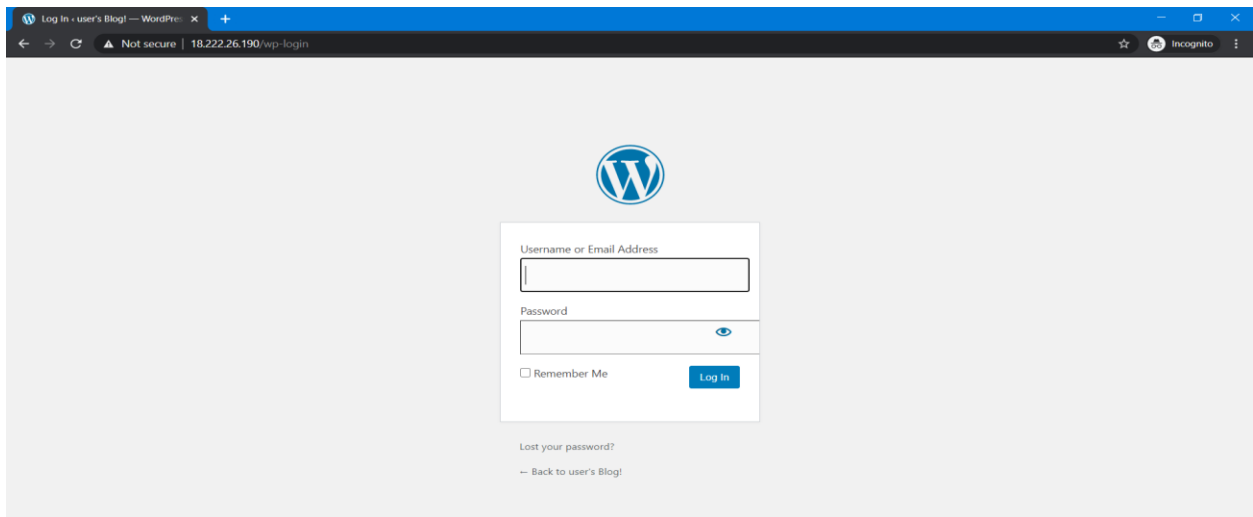
3- Now go to Security Groups and select the name wordpress

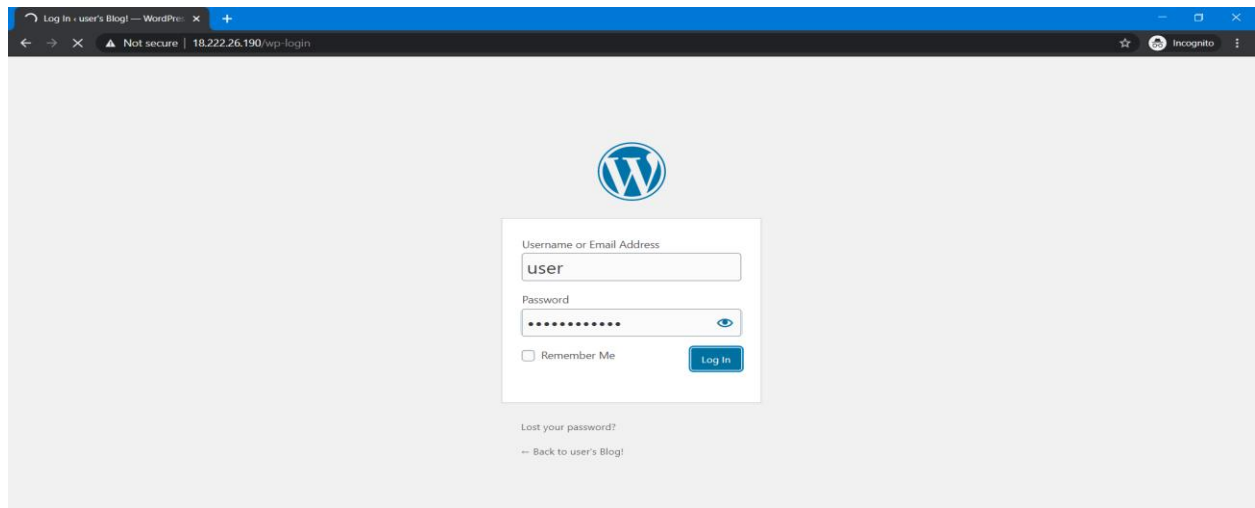


4- Now select the ec2 instance 'wordpress'.
Copy the public ip and paste the ip to view the wordpress website.

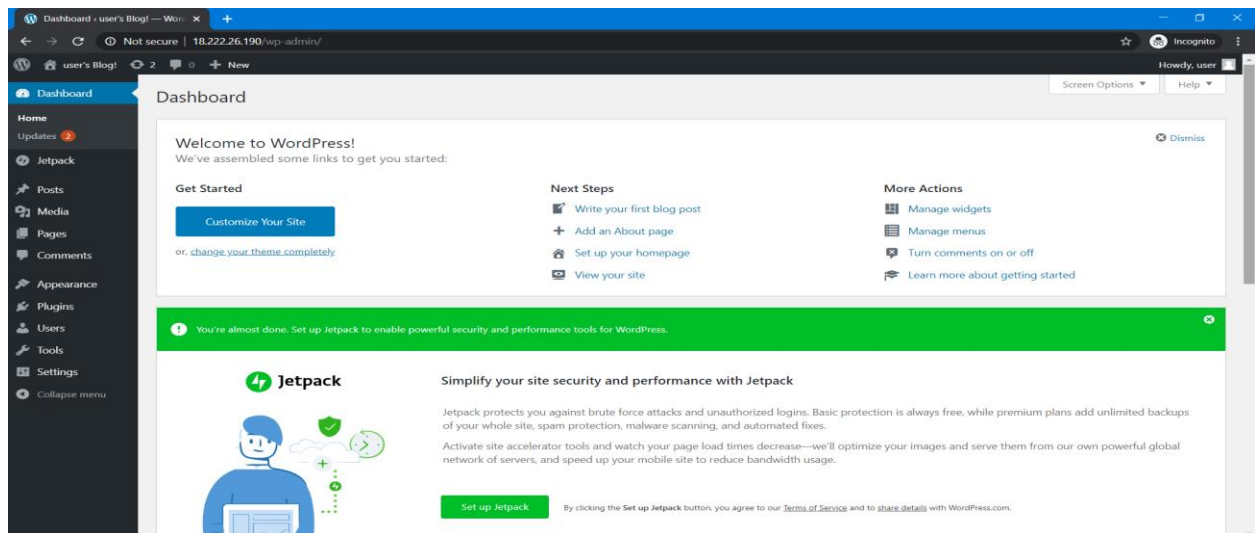


5- After enter the ip/wp-admin the login page will be appeared.





6- Now Dashboard is open



7- After doing changes we get this page.
Finally wordpress application is launched

