



CLOUD+

A beginner's guide to creating an EC2 instance
inside a custom VPC and connecting via SSH

SUMMARY

This project covers the process of setting up a custom VPC, launching an EC2 instance inside it, and connecting to the server using SSH. The steps include creating a VPC with a subnet, assigning a public IP, opening required ports, and generating a key pair. Finally, PuTTY was used as the SSH client to complete the connection, since the default SSH client on Windows was unavailable due to update issues.

Tebogo

EDUV7276134

Table of Contents

Introduction

Console Page

Name the VPC

Assign IPv4 CIDR Block

Assign 1 AZ and 1 Subnet

VPC Creation Done

Create EC2 Instance

Add Tags

Software Image

Instance Type

Create a new key pair

Use the VPC created above

Assign the Subnet from VPC

Ensure the Public IP is auto assigned

Allow SSH and HTTP traffic from anywhere

EC2 running

Use PuTTY to connect to the EC2 instance

Connection complete

Project Illustration

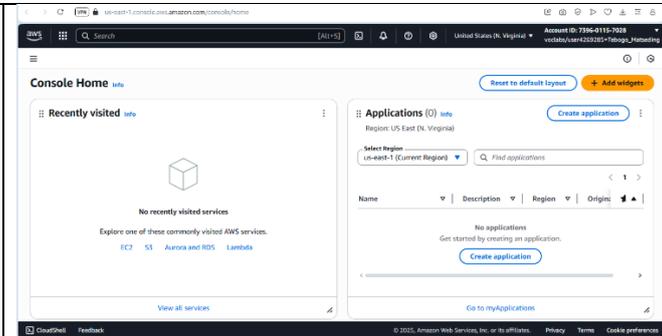
START

Introduction

In this project, we create a Virtual Private Cloud (VPC), launch an EC2 instance inside it, and connect to the server using SSH. This walkthrough shows how networking, instance creation, and secure access all fit together.

Console Page

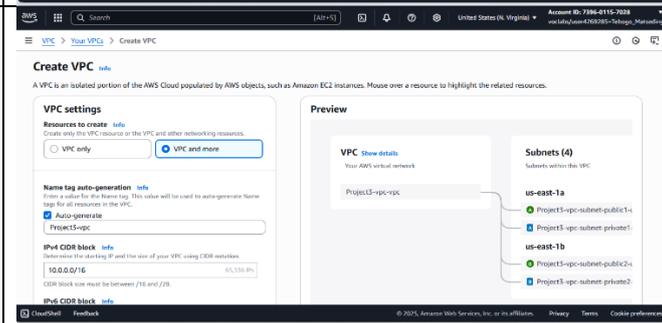
We begin in the AWS Management Console, the dashboard for accessing and managing all AWS services.



Name the VPC

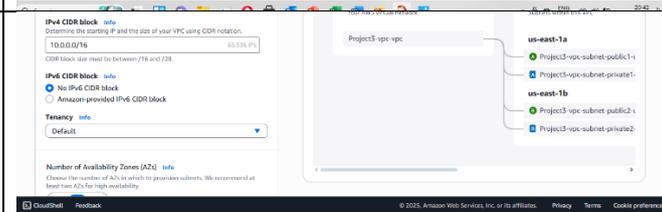
(Project3-vpc)

Inside the VPC section, we create a new Virtual Private Cloud and give it a meaningful name. This VPC will act as the private network for our instance.



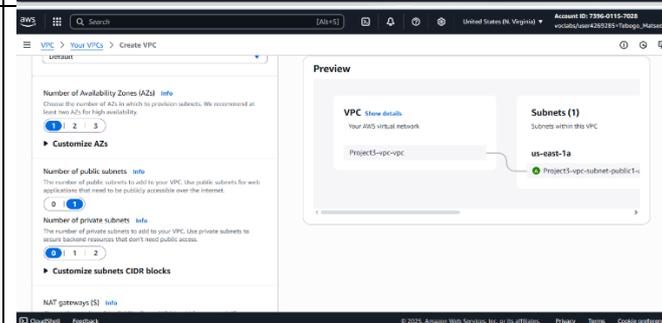
Assign IPv4 CIDR Block

We assign an IPv4 CIDR block (e.g., 10.0.0.0/16) to define the range of private IP addresses available inside the VPC.



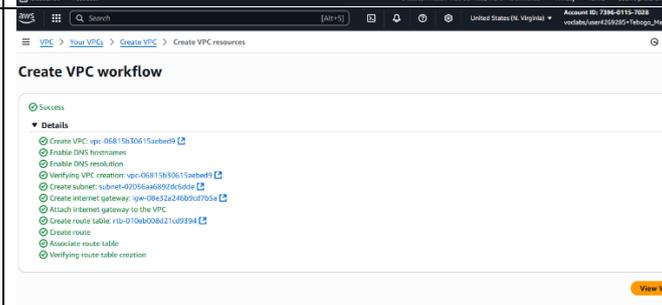
Assign 1 AZ and 1 Subnet

A subnet is created inside a specific Availability Zone. This subnet ensures our EC2 instance is tied to a particular physical data center within the region.



VPC Creation Done

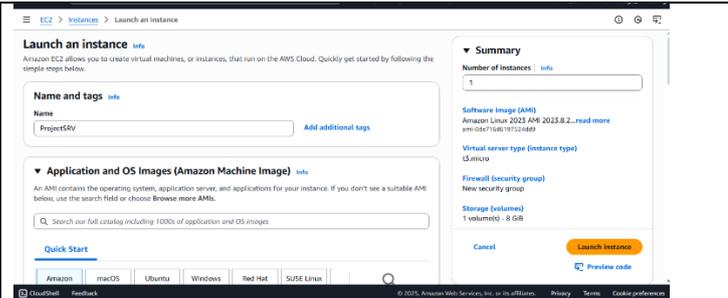
Once the VPC and subnet are configured, we complete the setup. Now we have a private environment to host our EC2 instance.



Create EC2 Instance

(Name ProjectSRV)

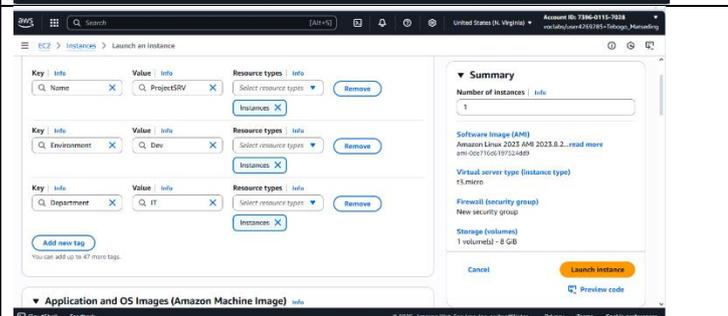
We move to the EC2 service and start the process of launching a new virtual server inside the VPC.



Add Tags

We add tags

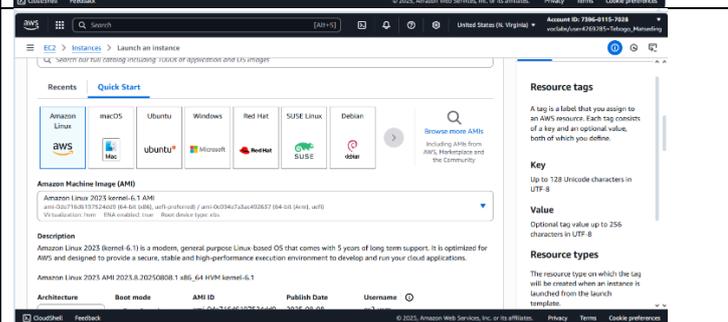
(like Name = MyProjectInstance) for easy identification and organization.



Software Image

(Amazon Linux image)

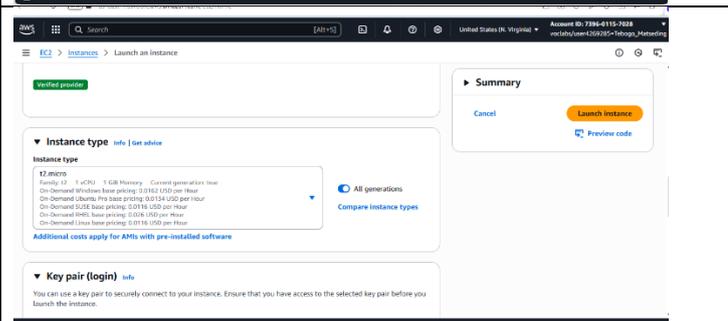
We select an Amazon Machine Image (AMI), such as Amazon Linux or Ubuntu. This defines the operating system of our EC2 instance.



Instance Type

(t2.micro)

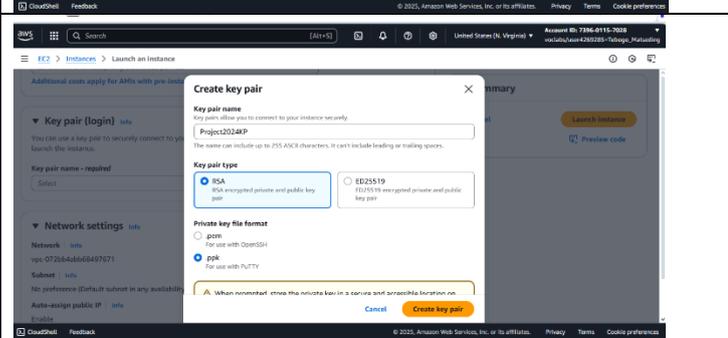
We choose an instance type (e.g., t2.micro or t3.medium) based on performance and cost needs. For school projects, smaller instance types are ideal.



Create a new key pair

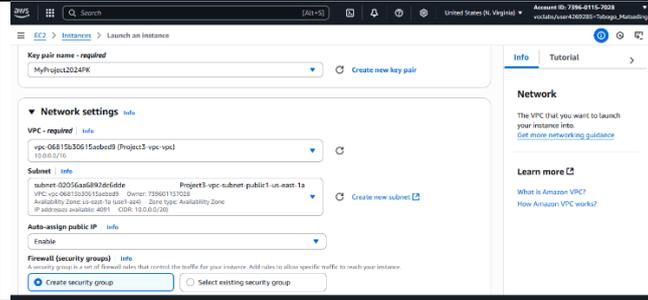
We generate a new key pair named MyProject2024KP using the .pk format. This is required for secure SSH access.

Note: The default SSH client on Windows could not be installed due to update issues, so PuTTY was used instead.



Use the VPC created above

During the launch process, we select the custom VPC created earlier to host the EC2 instance.

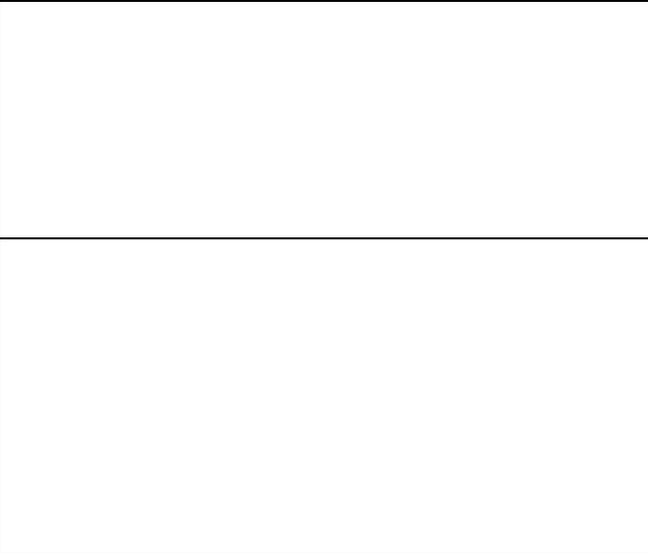


Assign the Subnet from VPC

We place the instance inside the subnet we created, tying it to a specific availability zone.

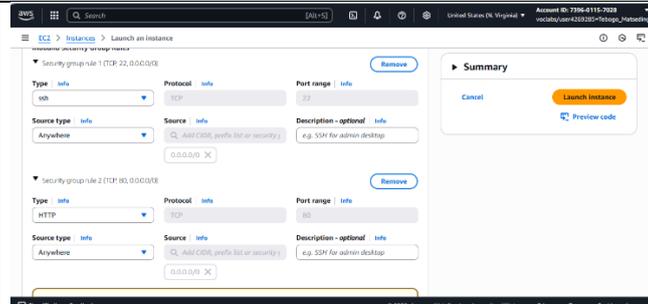
Ensure the Public IP is auto assigned

We configure the settings so that a public IP address is automatically assigned. This allows us to connect to the instance from the internet.



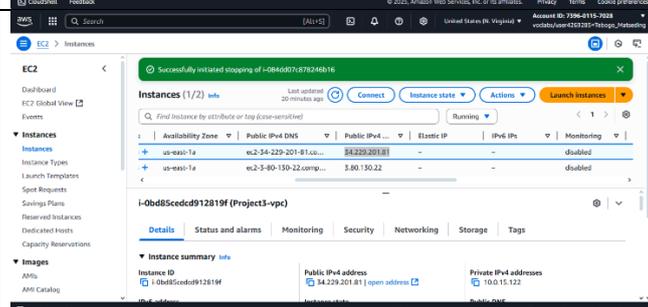
Allow SSH and HTTP traffic from anywhere

In the Security Group settings, we open port 22 (SSH) and port 80 (HTTP) to allow remote access and web traffic.



EC2 running

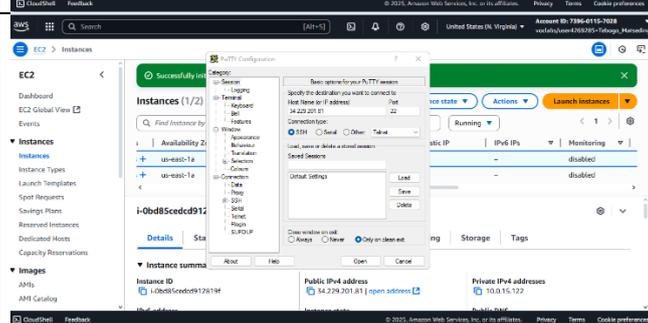
After reviewing and launching, the instance enters the running state. It now has both a private IP (inside the VPC) and a public IP (for external access).



Use PuTTY to connect to the EC2 instance

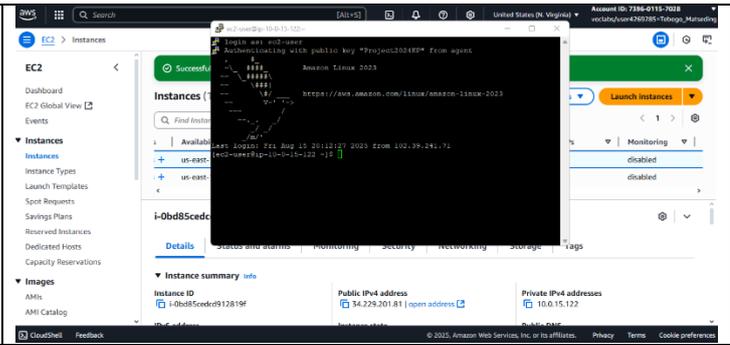
(via SSH using the created private key).

We open PuTTY, load the private key (.ppk), and connect to the instance using its public IP.



Connection complete

Once the session opens, we confirm that the SSH connection is successful. At this stage, we can run commands on the EC2 server.



Project illustration

