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.

Tebo

[Course title]

Table of Contents

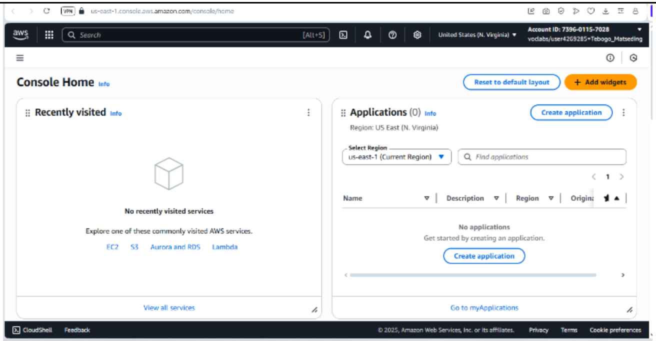
START

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.



us-east-1.console.aws.amazon.com/console/home

United States (N. Virginia) ▾ Account ID: 7396-0115-7028 voclabs/user4269285=Tebogo_Matseding

Console Home Info

Recently visited Info

No recently visited services

Explore one of these commonly visited AWS services.

Applications (0) Info

Region: US East (N. Virginia)

Select Region: us-east-1 (Current Region)

No applications

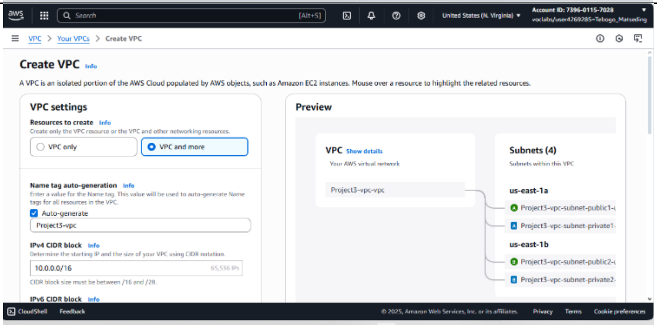
Get started by creating an application.

© 2025, Amazon Web Services, Inc. or its affiliates.

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.



VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

○ VPC only ● VPC and more

Name tag auto-generation Info

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☑ Auto-generate

Project3-vpc

IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

Preview

VPC Show details — Your AWS virtual network: Project3-vpc-vpc

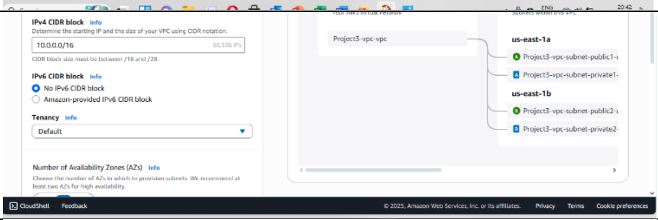
Subnets (4) — Subnets within this VPC

us-east-1a: Project3-vpc-subnet-public1-; Project3-vpc-subnet-private1-

us-east-1b: Project3-vpc-subnet-public2-; Project3-vpc-subnet-private2-

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.



IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

● No IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

Tenancy Info: Default

Number of Availability Zones (AZs) Info

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

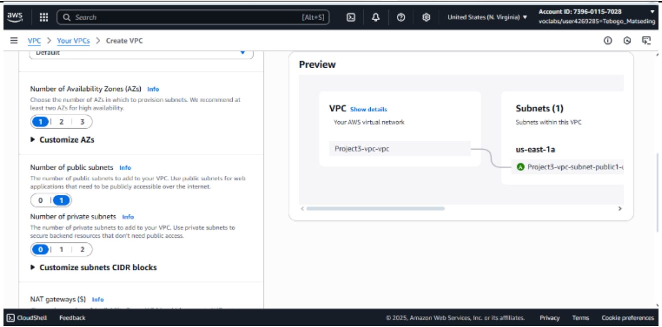
Project3-vpc-vpc

us-east-1a: Project3-vpc-subnet-public1-; Project3-vpc-subnet-private1-

us-east-1b: Project3-vpc-subnet-public2-; Project3-vpc-subnet-private2-

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 > Your VPCs > Create VPC

Number of Availability Zones (AZs) Info

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

[✓ 1] 2 3

▸ Customize AZs

Number of public subnets Info

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 [✓ 1]

Number of private subnets Info

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

[✓ 0] 1 2

▸ Customize subnets CIDR blocks

NAT gateways ($) Info

Preview

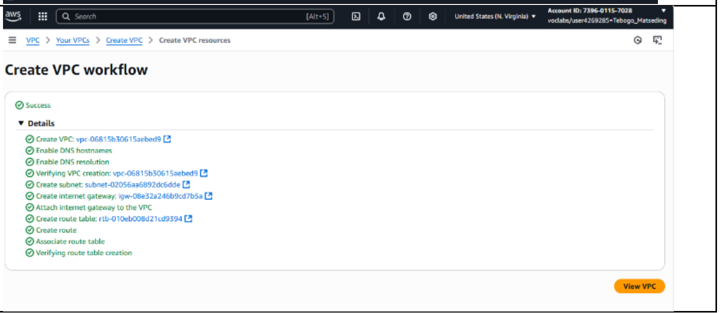
VPC Show details — Your AWS virtual network: Project3-vpc-vpc

Subnets (1) — Subnets within this VPC

us-east-1a: Project3-vpc-subnet-public1-

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.



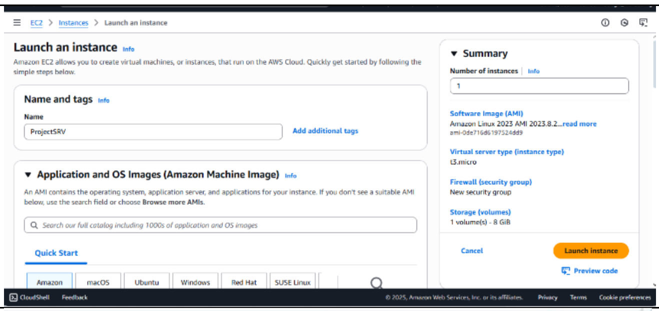
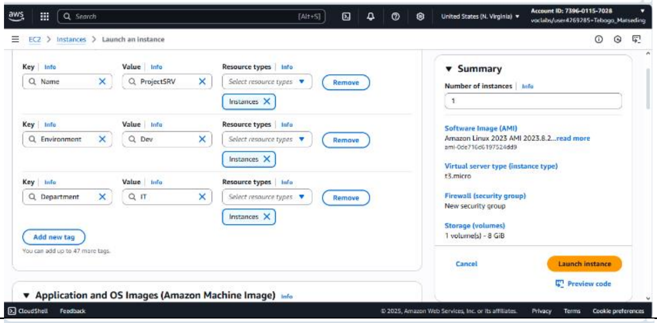
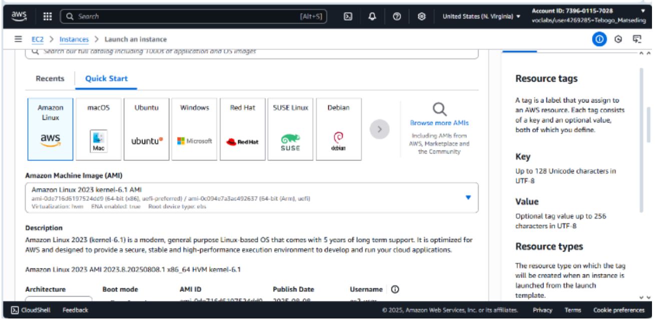
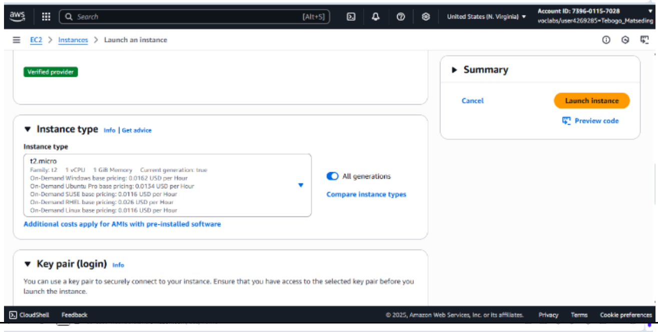
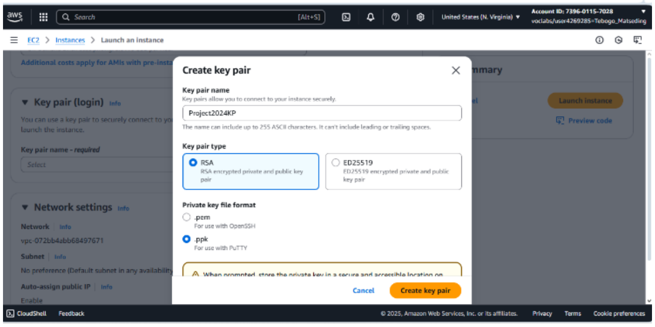
VPC > Your VPCs > Create VPC > Create VPC resources

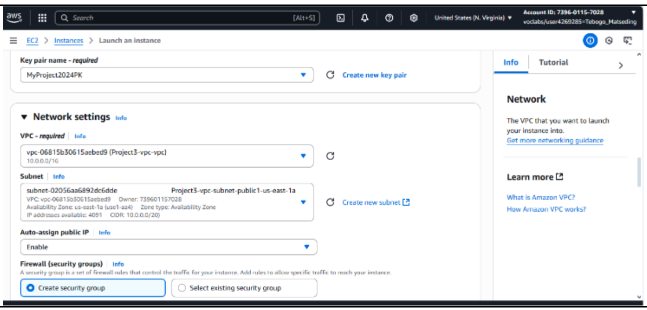
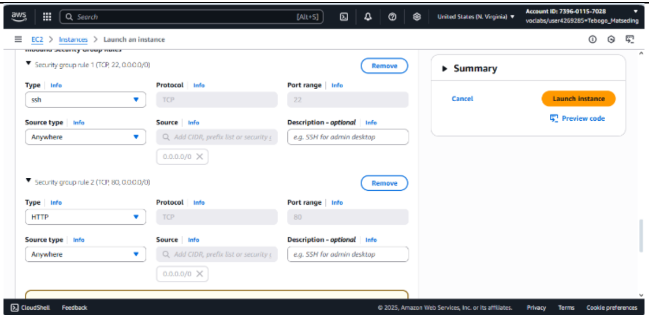
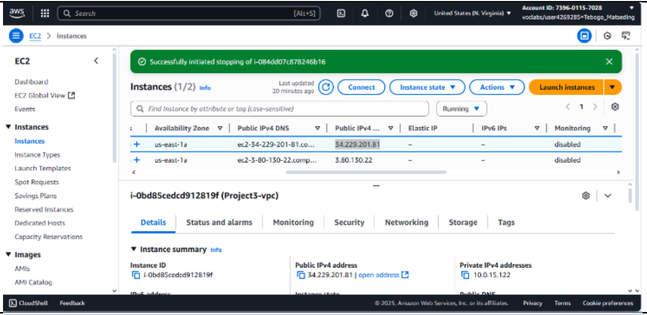
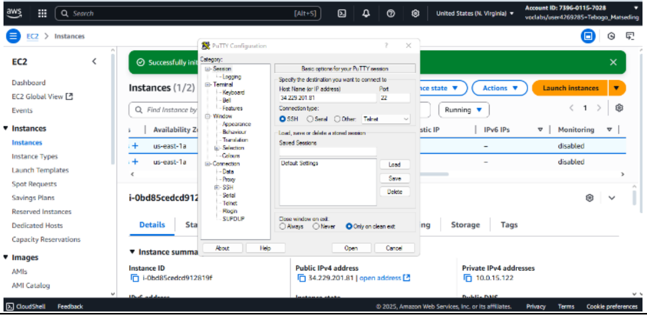
Create VPC workflow

✓ Success

▾ Details

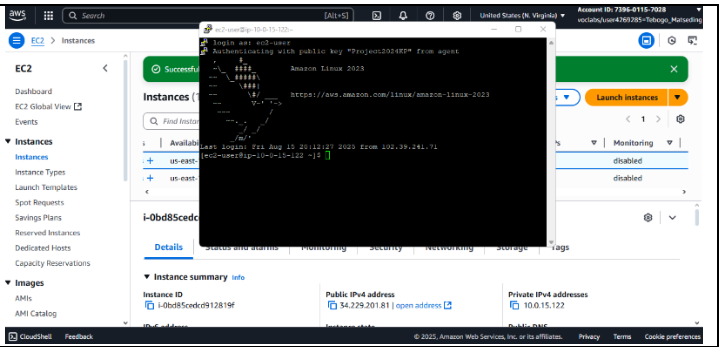
- ✓ Create VPC: vpc-06815b30615aebed9
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-06815b30615aebed9
- ✓ Create subnet: subnet-02056aa6892dc6dde
- ✓ Create internet gateway: igw-08e32a246b9cd7b5a
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: rtb-010eb008d21cd9394
- ✓ Create route
- ✓ Associate route table
- ✓ Verifying route table creation

Create EC2 Instance (Name ProjectSRV) We move to the EC2 service and start the process of launching a new virtual server inside the VPC.	 EC2 > Instances > Launch an instance Launch an instance 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 Name: ProjectSRV Add additional tags Application and OS Images (Amazon Machine Image) An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs. Search our full catalog including 1000s of application and OS images Quick Start: Amazon, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Summary Number of instances: 1 Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-0de716d6197524dd9 Virtual server type (instance type): t3.micro Firewall (security group): New security group Storage (volumes): 1 volume(s) - 8 GiB Cancel Launch instance Preview code © 2025, Amazon Web Services, Inc. or its affiliates.
Add Tags We add tags (like Name = MyProjectInstance) for easy identification and organization.	 United States (N. Virginia) Account ID: 7396-0115-7028 voclabs/user4269285=Tebogo_Matseding EC2 > Instances > Launch an instance Key: Name — Value: ProjectSRV — Resource types: Instances — Remove Key: Environment — Value: Dev — Resource types: Instances — Remove Key: Department — Value: IT — Resource types: Instances — Remove Add new tag You can add up to 47 more tags. Application and OS Images (Amazon Machine Image) Summary Number of instances: 1 Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-0de716d6197524dd9 Virtual server type (instance type): t3.micro Firewall (security group): New security group Storage (volumes): 1 volume(s) - 8 GiB Cancel Launch instance Preview code
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.	 EC2 > Instances > Launch an instance Recents / Quick Start: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian Browse more AMIs — Including AMIs from AWS, Marketplace and the Community Amazon Machine Image (AMI) Amazon Linux 2023 kernel-6.1 AMI ami-0de716d6197524dd9 (64-bit (x86), uefi-preferred) / ami-0c094e7a3ac492637 (64-bit (Arm), uefi) Virtualization: hvm ENA enabled: true Root device type: ebs Description Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications. Amazon Linux 2023 AMI 2023.8.20250808.1 x86_64 HVM kernel-6.1 Architecture Boot mode AMI ID Publish Date Username Resource tags A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define. Key Up to 128 Unicode characters in UTF-8 Value Optional tag value up to 256 characters in UTF-8 Resource types The resource type on which the tag will be created when an instance is launched from the launch template.
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.	 EC2 > Instances > Launch an instance Verified provider Instance type Instance type: t2.micro Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour All generations Compare instance types Additional costs apply for AMIs with pre-installed software Key pair (login) You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance. Summary Cancel Launch instance Preview code
Create a new key pair We generate a new key pair named MyProject2024KP using the .ppk file 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.	 EC2 > Instances > Launch an instance Create key pair Key pair name Key pairs allow you to connect to your instance securely. Project2024KP 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 Cancel Create key pair Network settings Network: vpc-072bb4abb68497671 Subnet: No preference (Default subnet in any availability Auto-assign public IP: Enable

Use the VPC created above During the launch process, we select the custom VPC created earlier to host the EC2 instance.	 EC2 > Instances > Launch an instance Key pair name - *required*: MyProject2024PK ▼ Network settings VPC - *required*: vpc-06815b30615aebed9 (Project3-vpc-vpc) 10.0.0.0/16 Subnet: subnet-02056aa6892dc6dde Project3-vpc-subnet-public1-us-east-1a VPC: vpc-06815b30615aebed9 Owner: 739601157028 Availability Zone: us-east-1a (use1-az4) Zone type: Availability Zone IP addresses available: 4091 CIDR: 10.0.0.0/20) Auto-assign public IP: Enable Firewall (security groups) 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 Network The VPC that you want to launch your instance into. Get more networking guidance Learn more What is Amazon VPC? How Amazon VPC works?
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 > Instances > Launch an instance ▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) Type: ssh Protocol: TCP Port range: 22 Source type: Anywhere Source: 0.0.0.0/0 Description - *optional*: e.g. SSH for admin desktop ▼ Security group rule 2 (TCP, 80, 0.0.0.0/0) Type: HTTP Protocol: TCP Port range: 80 Source type: Anywhere Source: 0.0.0.0/0 Description - *optional*: e.g. SSH for admin desktop Summary Cancel Launch instance Preview code
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).	 Successfully initiated stopping of i-084dd07c878246b16 Instances (1/2) Availability Zone | Public IPv4 DNS | Public IPv4 ... | Elastic IP | IPv6 IPs | Monitoring us-east-1a | ec2-34-229-201-81.co... | 34.229.201.81 | – | – | disabled us-east-1a | ec2-3-80-130-22.comp... | 3.80.130.22 | – | – | disabled i-0bd85cedcd912819f (Project3-vpc) Instance summary Instance ID: i-0bd85cedcd912819f Public IPv4 address: 34.229.201.81 | open address Private IPv4 addresses: 10.0.15.122
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.	 PuTTY Configuration Basic options for your PuTTY session Specify the destination you want to connect to Host Name (or IP address): 34.229.201.81 Port: 22 Connection type: ● SSH ○ Serial ○ Other: Telnet Saved Sessions: Default Settings Close window on exit: ○ Always ○ Never ● Only on clean exit Open Cancel Instance ID: i-0bd85cedcd912819f Public IPv4 address: 34.229.201.81 | open address Private IPv4 addresses: 10.0.15.122

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.



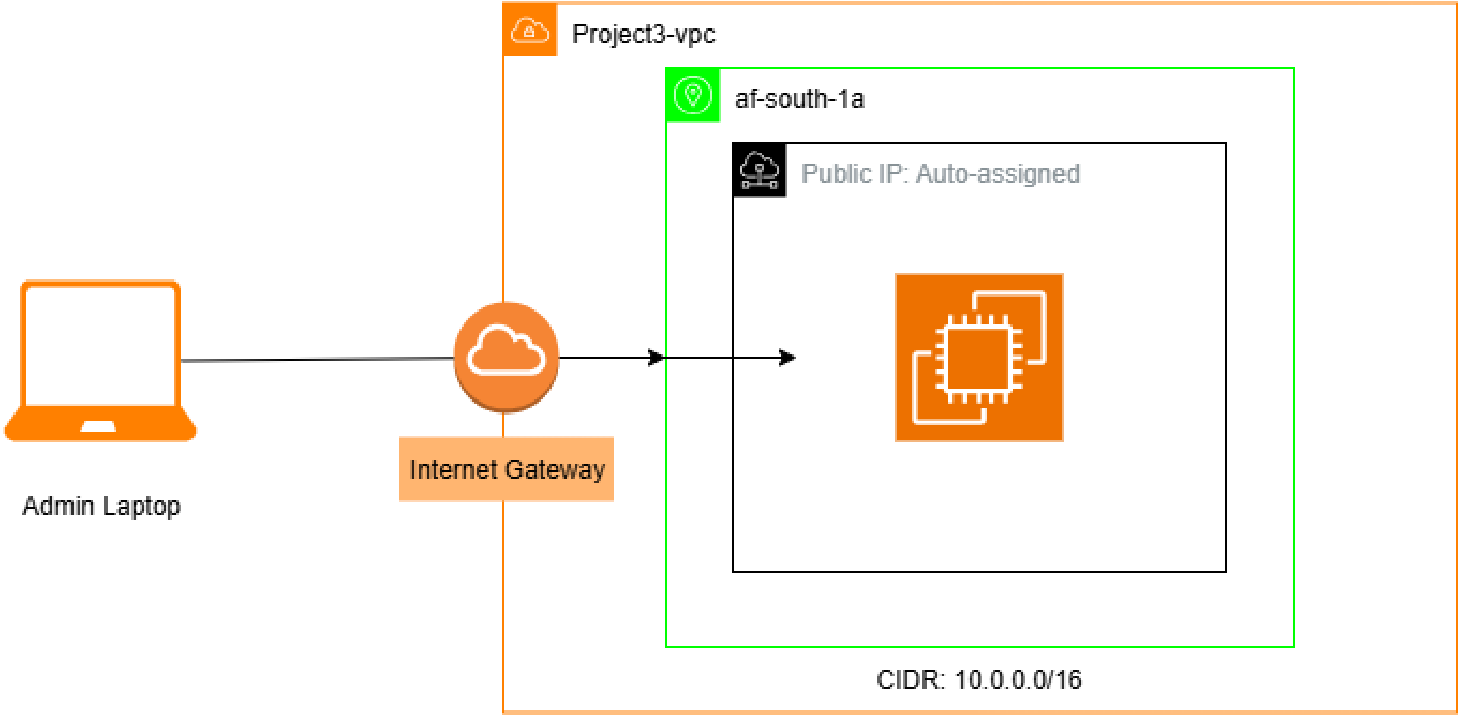
```
login as: ec2-user
Authenticating with public key "Project2024KP" from agent
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Aug 15 20:12:27 2025 from 102.39.241.71
[ec2-user@ip-10-0-15-122 ~]$ 
```

Instance ID: i-0bd85cedcd912819f

Public IPv4 address: 34.229.201.81 | open address

Private IPv4 addresses: 10.0.15.122

Project illustration



Security group

Inbound

SSH (Port 22) = 0.0.0.0/0

HTTP (Port 80) = 0.0.0.0/0

Outbound

All traffic = 0.0.0.0/0

EC2 Details

Name: ProjectSRV

AMI: Amazon Linux

Instance Type: t2micro

Key Pair: MyProject2024KP

Tags: Key=IT, Key=Dev