

# AWS Project VPC with public-private subnet in production

This example demonstrates how to create a VPC that you can use for servers in a production environment. To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both Availability Zones.

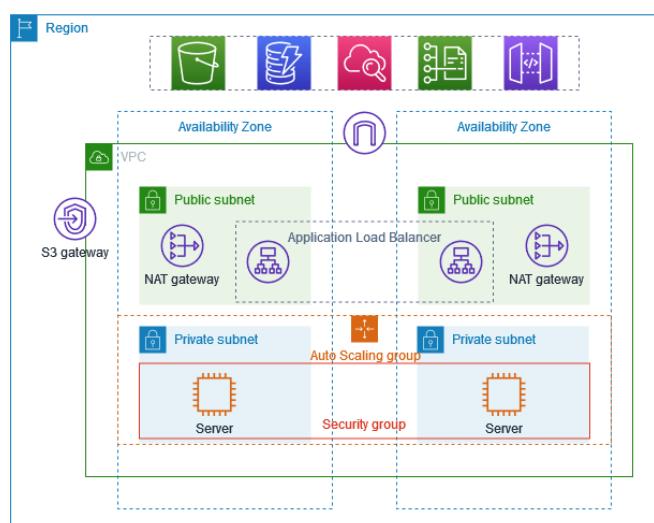
## Overview

The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway. The servers can connect to Amazon S3 by using a gateway VPC endpoint.

## Architecture Overview

The project creates a robust cloud infrastructure using the following AWS services:

- **VPC (Virtual Private Cloud)** - Isolated network environment
- **Public & Private Subnets** - Network segmentation across multiple AZs
- **EC2 Auto Scaling** - Automatic scaling based on demand
- **Application Load Balancer** - Traffic distribution and high availability
- **Target Groups** - Health monitoring and routing
- **EC2 -Bastion Host** - Secure access to private resources
- **Security Groups** - Network-level security control



## -components Breakdown

### 1. VPC (Virtual Private Cloud)

- CIDR Block: 10.0.0.0/16
- Purpose: Isolated network environment for all resources
- Features:
  - DNS resolution enabled
  - DNS hostnames enabled
  - Dedicated tenancy (optional)

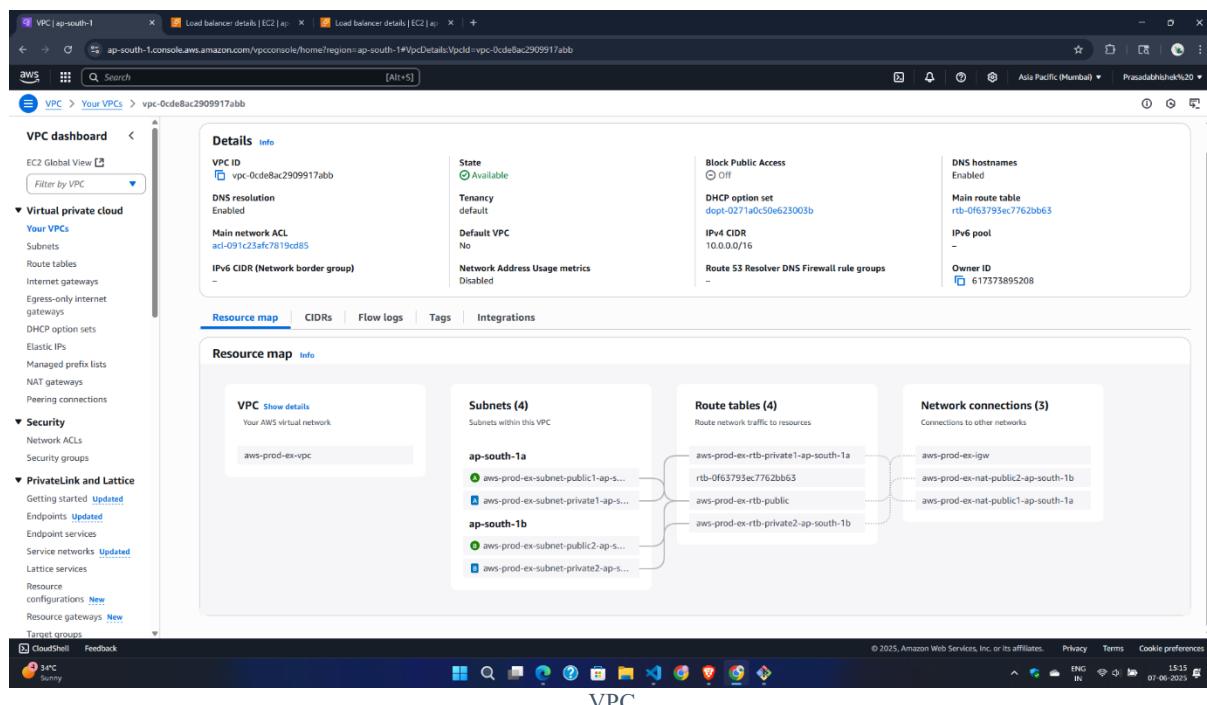
### 2. Subnets Configuration

#### Public Subnets

- Public Subnet 1: 10.0.1.0/24 (AZ-1a)
- Public Subnet 2: 10.0.2.0/24 (AZ-1b)
- Features:
  - Internet Gateway attached
  - Auto-assign public IP enabled
  - Hosts Load Balancer and Bastion Host

#### Private Subnets

- Private Subnet 1: 10.0.3.0/24 (AZ-1a)
- Private Subnet 2: 10.0.4.0/24 (AZ-1b)
- Features:
  - NAT Gateway for outbound internet access
  - Hosts application servers
  - No direct internet access



### 3. EC2 Auto Scaling Group

- Configuration:

- Minimum instances: 2
- Maximum instances: 6
- Desired capacity: 2
- Health check type: ELB
- Health check grace period: 300 seconds

- Scaling Policies:

- Scale up when CPU > 70%
- Scale down when CPU < 30%

- Launch Template:

- Instance type: t3.micro (or t2.micro for free tier)
- AMI: Amazon Linux 2
- User data script for web server setup

The screenshot shows the AWS EC2 Auto Scaling group details page for 'aws-prod-ex'. The left sidebar navigation includes 'EC2' (selected), 'Dashboard', 'Instances', 'Images', 'Elastic Block Store', 'Network & Security', and 'Load Balancing'. The main content area is titled 'aws-prod-ex Capacity overview' and displays the following information:

Desired capacity	Scaling limits (Min - Max)	Desired capacity type
2	1 - 4	Units (number of instances)

Date created: Sat Jun 07 2025 14:38:53 GMT+0530 (India Standard Time)

**Launch template** (Edit):  
Launch template: lt-0a507a68813446bf (aws-prod-ex)  
AMI ID: ami-0e35ddab05955cf5f  
Version: Default  
Description: proof of concepts for app deploying private subnet.  
View details in the launch template console.

**AMI ID**: ami-0e35ddab05955cf5f  
**Instance type**: t2.micro  
**Security groups**: sg-0fe7dd49a25b85022  
**Storage (volumes)**: -  
**Key pair name**: key-one  
**Owner**: arn:aws:iam::617373895208:root  
**Create time**: Sat Jun 07 2025 12:30:14 GMT+0530 (India Standard Time)  
**Request Spot Instances**: No

**Network** (Edit):  
Availability Zones: ap-south-1b, ap-south-1a  
Subnet ID: subnet-08a28c1aa02db84bd, subnet-07b7e13ae6cc973b  
Availability Zone distribution: Balanced best effort

At the bottom, there are links for CloudShell, Feedback, and a status bar showing 34°C, Sunny, ENG IN, 15:16, and 07-06-2025.

### 4. Application Load Balancer (ALB)

- Type: Application Load Balancer
- Scheme: Internet-facing
- Listeners: HTTP (Port 80), HTTPS (Port 443)
- Availability Zones: Both AZ-1a and AZ-1b
- Features:
  - Cross-zone load balancing
  - Deletion protection

## Access logging (optional)

The screenshot shows the AWS CloudFront console with the 'aws-prod-ex' distribution selected. In the 'Listeners and rules' section, there is one rule defined:

ProtocolPort	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group aws-prod-ex:1 (100%)	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applicable

Load Balancer

## 5. Target Group

- Target Type:** Instance
- Protocol:** HTTP
- Port:** 80
- Health Check:**
  - Path: /health or /
  - Interval: 30 seconds
  - Timeout: 5 seconds
  - Healthy threshold: 2
  - Unhealthy threshold: 5

The screenshot shows the AWS CloudFront console with the 'aws-prod-ex' target group selected. The 'Targets' tab is active, showing the following data:

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	1	1	0	0	0

The 'Registered targets' section shows the following table:

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative o...	Override details	Launch...	Anomaly
i-06aeefc8861767d800		8000	ap-south-1a (a...)	Healthy	-	No override	No override is curren...	June 7, 20...	Normal
i-0bd9a5e930fd8dc0		8000	ap-south-1b (a...)	Unhealthy	Health checks failed	No override	No override is curren...	June 7, 20...	Normal

Target Group

## 6. Bastion Host (Jump Server)

The screenshot shows the AWS EC2 Instances page. The left sidebar includes sections for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays a table of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, and Elastic IP. Three instances are listed as Running: t2.micro. The instance **i-0f67e595cd5d8aab** is highlighted. Below the table, a "Select an instance" dropdown is open. The bottom navigation bar includes CloudShell, Feedback, and links for Privacy, Terms, and Cookie preferences.

## 7. Security Groups

- Load Balancer Security Group
- Web Server Security Group
- Bastion Host Security Group

The screenshot shows the AWS Security Groups page for the security group **sg-0fe7dd49a25b83022 - aws-prod-ex**. The left sidebar is identical to the previous EC2 screenshot. The main content area shows the security group details: name (aws-prod-ex), ID (sg-0fe7dd49a25b83022), owner (617373895208), description (allows ssh access), and VPC ID (vpc-0cde8ac2900917abb). Below this, the Inbound rules section is displayed, showing three entries for SSH access from 0.0.0.0/0. The bottom navigation bar includes CloudShell, Feedback, and links for Privacy, Terms, and Cookie preferences.

## Resources

The screenshot shows the AWS EC2 Resources page. On the left, a sidebar navigation menu includes sections for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area displays a summary of resources: Instances (running) 1, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 2, Instances 3, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 9, Snapshots 0, and Volumes 1. Below this, there are sections for Launch instance, Service health (Region: Asia Pacific (Mumbai), Status: This service is operating normally), Zones (listing zones ap-south-1a, ap-south-1b, ap-south-1c), Instance alarms (0 in alarm, 0 OK, 0 insufficient data), and Scheduled events (Asia Pacific (Mumbai), No scheduled events). A sidebar on the right provides information about EC2 Free Tier, offers for all AWS Regions, and monthly usage statistics for Linux EC2 Instances and Storage space on EBS. At the bottom, the status bar shows the URL https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Volumes, the date 07-06-2025, and system icons.

## Volume

The screenshot shows the AWS Volumes page. The sidebar navigation menu is identical to the one in the previous screenshot. The main content area displays a table titled "Volumes (3) Info" showing three volumes: vol-038b37020c5b081bd (gp3, 8 GB, 3000 IOPS, throughput 125, snapshot snap-007d50e..., created 2025/06/07 12:57 GMT+5...), vol-00529e07f9543fb5 (gp3, 8 GB, 5000 IOPS, throughput 125, snapshot snap-00a5570..., created 2025/06/07 14:38 GMT+5...), and vol-00fa2f9510a09fba8 (gp3, 8 GB, 3000 IOPS, throughput 125, snapshot snap-00a5570..., created 2025/06/07 14:38 GMT+5...). Below the table, a section titled "Fault tolerance for all volumes in this Region" shows a "Snapshot summary" with 0 / 3 recently backed up volumes / Total # volumes. A note indicates "Data Lifecycle Manager default policy for EBS Snapshots status" and "No default policy set up | Create policy". The status bar at the bottom shows the URL https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Volumes, the date 07-06-2025, and system icons.

## EC2 Instance

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, and Load Balancing.

The main content area displays the 'Instances (3) info' table with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
i-0bd9ae3e930fd08dc0	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	-	ap-south-1b	-	-	-
i-0f67e595cd5d8baab	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	-	ap-south-1a	ec2-13-233-84-48.ap-s...	13.233.84.48	-
i-06aefc8861767db06	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	-	ap-south-1a	-	-	-

At the bottom of the main content area, there is a dropdown menu labeled 'Select an instance'.

The status bar at the bottom right shows the date and time as 07-06-2025 and 15:17.

## Command

The terminal session starts with the user connecting via SSH to an Ubuntu 24.04.2 LTS instance. The user runs `ssh -i key-one.pem ubuntu@10.0.3.154` and is prompted to add the host to the known hosts. The user responds with 'yes'. The session then displays the welcome message for Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86\_64).

Documentation, Management, and Support links are provided. The system information section shows the following details:

- System load: 0.0
- Processes: 104
- Usage of /: 20.9% of 6.71GB
- Users logged in: 0
- Memory usage: 20%
- IPv4 address for enx0: 10.0.3.154
- Snap usage: 0%

The terminal also indicates that expanded security maintenance for applications is not enabled and lists available updates.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/<copyright>.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>". See "man sudo\_root" for details.

The user then lists the contents of the current directory, edits an index.html file using vim, and runs a Python web server on port 8000, serving the file 'index.html'.

The terminal status bar at the bottom right shows the date and time as 07-06-2025 and 15:20.

## Response

```
ubuntu@ip-10-0-3-154: ~
10.0.14.237 - - [07/Jun/2025 09:33:29] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:33:29] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:33:55] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:33:59] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:34:25] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:34:29] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:34:55] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:34:59] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:35:00] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:35:05] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:35:05] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:35:05] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:35:16] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:36:19] code 404, message File not found
10.0.14.237 - - [07/Jun/2025 09:36:19] "GET /favicon.ico HTTP/1.1" 404 -
10.0.27.174 - - [07/Jun/2025 09:36:25] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:36:29] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:36:55] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:36:59] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:37:00] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:37:05] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:37:29] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:37:55] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:37:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:38:20] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:38:29] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:38:55] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:38:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:39:00] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:39:29] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:39:53] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:39:59] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:40:25] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:40:29] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:40:55] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:40:59] "GET / HTTP/1.1" 200 -
`C
Keyboard interrupt received, exiting.
ubuntu@ip-10-0-3-154:~$ ls
index.html
ubuntu@ip-10-0-3-154:~$ vim index.html
ubuntu@ip-10-0-3-154:~$ rm index.html
ubuntu@ip-10-0-3-154:~$ vim index.html
ubuntu@ip-10-0-3-154:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.27.174 - - [07/Jun/2025 09:42:53] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:42:55] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:43:00] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:43:03] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:43:30] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:43:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:44:25] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:44:29] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:44:55] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:44:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:45:00] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:45:05] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:45:30] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:45:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:46:25] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:46:29] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:46:55] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:46:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:47:00] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:47:05] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:47:30] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:47:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:48:25] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:48:29] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:48:55] "GET / HTTP/1.1" 200 -
10.0.27.174 - - [07/Jun/2025 09:48:59] "GET / HTTP/1.1" 200 -
10.0.14.237 - - [07/Jun/2025 09:49:00] "GET / HTTP/1.1" 200 -
`C
```

Final OutPut :

