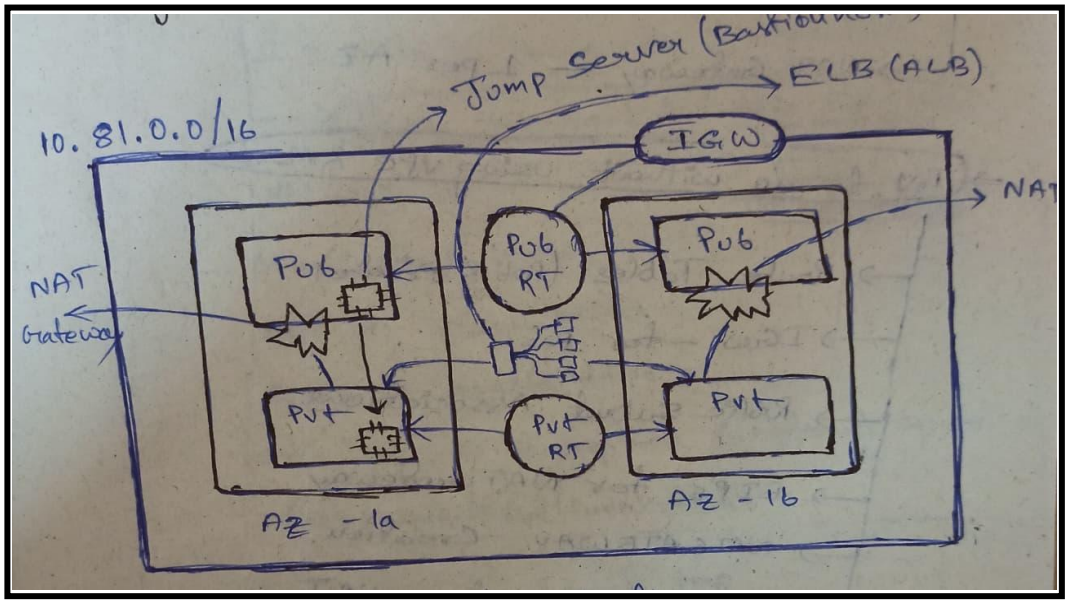
Building a Highly Available 2-Tier Architecture on AWS

A Practical Implementation Using VPC, Subnets, ALB, ASG, NAT Gateway & Bastion Host



AWS 2-Tier Architecture

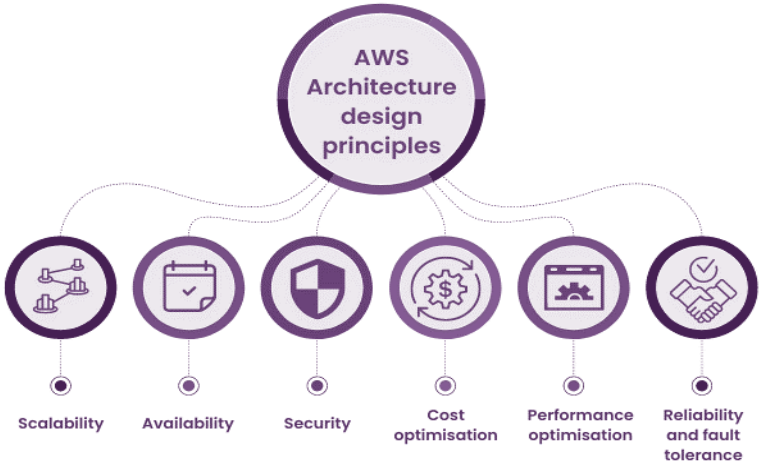
Overview:

This document presents a step-by-step walkthrough of designing and deploying a highly available 2-tier architecture on AWS. It covers core networking and compute components including VPC, public/private subnets, Application Load Balancer (ALB), Auto Scaling Group (ASG), NAT Gateway, Bastion Host, route tables, and security groups — all implemented across multiple Availability Zones for resilience and scalability.

1) Introduction

Modern applications demand high availability, security, and scalable infrastructure. AWS provides the foundational building blocks to design architectures that can automatically adapt to traffic, withstand failures, and ensure seamless user experiences.

In this document, I walk through the design of a Highly Available 2-Tier Architecture on AWS, where the application layer runs in private subnets and is securely accessed through a public-facing load balancer. By leveraging AWS networking services such as VPC, subnets, route tables, NAT Gateway, Bastion Host, and security groups—along with compute components like Auto Scaling Groups (ASG) and Application Load Balancer (ALB)—this architecture achieves both resilience and security.



What This Architecture Includes

- A dedicated VPC for isolated networking
- Public and private subnets across multiple AZs
- Application Load Balancer (ALB) for distributing traffic
- Auto Scaling Group (ASG) to maintain capacity and availability
- NAT Gateway for secure outbound traffic from private subnets
- Bastion Host (Jump Server) for controlled SSH access
- Route Tables for traffic flow control
- Security Groups for layered security

2) VPC & Subnet Design

To build a secure and highly available 2-tier architecture, all networking components are created inside a dedicated VPC named Prod2Tier-vpc with CIDR 10.81.0.0/16. This range gives enough IP space for multiple subnets and future expansion.

1. VPC & Availability Zones

- VPC Name: Prod2Tier-vpc
- CIDR: 10.81.0.0/16
- AZs Used: ap-south-1a & ap-south-1b

Using two Availability Zones ensures the application can continue running even if one zone experiences issues.

2. Subnet Layout

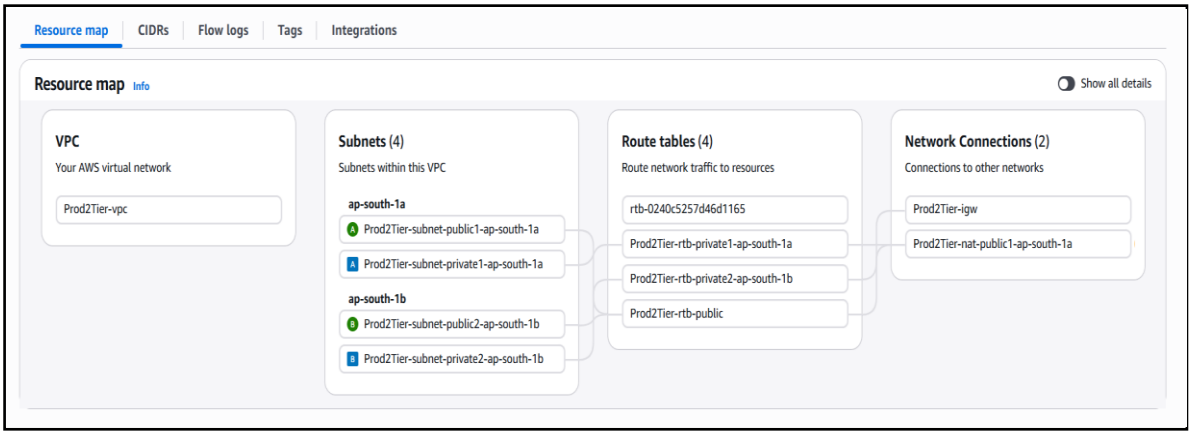
A total of 4 subnets are created — two public and two private — each placed in a different AZ for high availability.

Public Subnets

- Prod2Tier-subnet-public1-ap-south-1a → 10.81.1.0/24
- Prod2Tier-subnet-public2-ap-south-1b → 10.81.2.0/24

Private Subnets

- Prod2Tier-subnet-private1-ap-south-1a → 10.81.3.0/24
- Prod2Tier-subnet-private2-ap-south-1b → 10.81.4.0/24



3) EC2 Instances, Bastion Host & Access Flow

Bastion Host (Jump Server) — Public Subnet

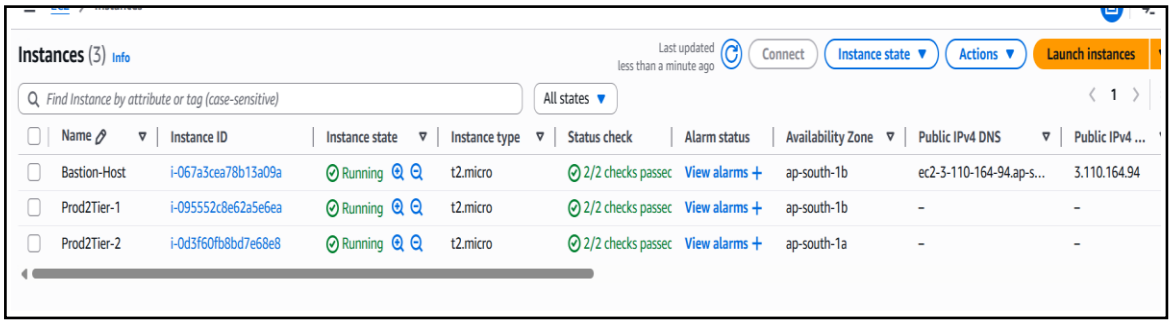
Name: Prod2Tier-bastion-ec2
Deployed in Public Subnet (ap-south-1a) with a public IP.

➞ Why We Use a Bastion Host

We never expose private EC2 instances directly to the internet.
Instead, a Bastion Host acts as a secure entry point:

- Placed in the public subnet
- Accessible only via SSH from your IP
- From here, you "jump" into private EC2 instances
- All SSH access stays inside the VPC
- Stronger isolation and reduced attack surface

It enables controlled, auditable access to your private application servers.



Private EC2 Instances (Application Tier)

Name Pattern:
Prod2Tier-app-ec2-1a
Prod2Tier-app-ec2-1b

Deployed across private subnets in AZ 1a & 1b for high availability.

How They Are Accessed

Private EC2 instances do not have public IPs.
They are reachable only through:

1️⃣ Bastion Host (SSH Access)

Used for:

- Troubleshooting
- Code deployment
- Log inspection
- Configuration changes

2️⃣ Application Load Balancer (HTTP/HTTPS Traffic)

All user requests enter via ALB → Target Group → EC2.

No internet traffic reaches them directly.
No inbound SSH from the outside world.

Access Flow Summary

User → ALB → Target Group → Private EC2
Admin → Bastion Host → Private EC2 (SSH)

This design ensures:

- Security (no public exposure)
- Isolation (private subnets)
- Scalability (can add more EC2 via ASG)
- High availability (instances across two AZs)

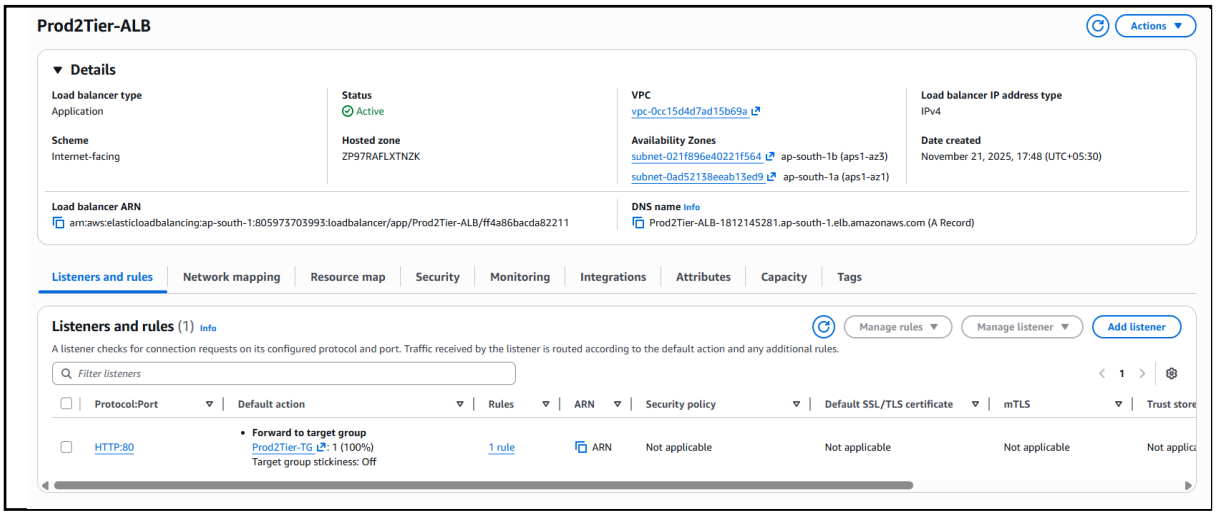
4) ALB & Auto Scaling Group

Application Load Balancer (ALB)

Name: Prod2Tier-ALB
Placed in public subnets across ap-south-1a & 1b to achieve multi-AZ high availability.

Key Components

- Listener
- Target Group
- Health Checks

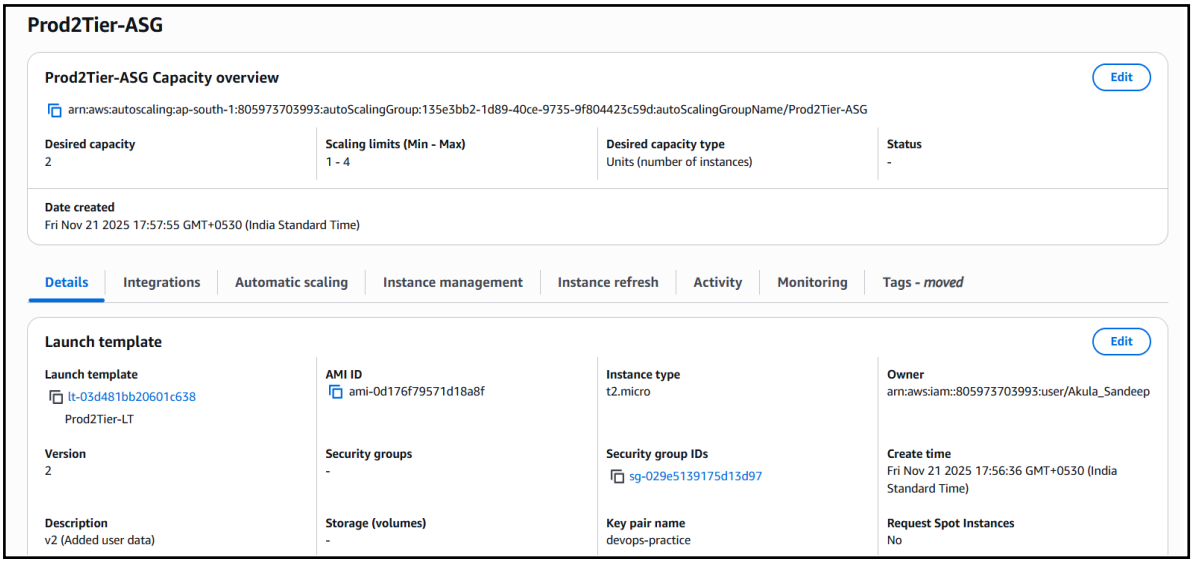


Auto Scaling Group (ASG)

Name: Prod2Tier-asg
Placed in private subnets across 1a & 1b to ensure backend redundancy.

Key Configuration

- Launch Template
- Desired Capacity / Min / Max
- Scaling Policies



How ALB + ASG Work Together

The combination delivers a highly available, fault-tolerant architecture:

- ALB distributes incoming traffic
- ASG ensures backend EC2 count is maintained
- ALB health checks remove unhealthy instances
- ASG replaces them automatically
- Traffic continues seamlessly during scaling events

Result:
A highly available, self-healing, production-ready 2-tier architecture.

5) Final Summary & Conclusion

This Highly Available AWS 2-Tier Architecture demonstrates a production-ready design using core AWS services.
By combining public and private subnets, ALB, ASG, NAT, and a Bastion Host, the environment achieves:

- Secure access control (Bastion for SSH, no public EC2 exposure)
- High availability across ap-south-1a & ap-south-1b
- Scalable application layer using Auto Scaling Group
- Efficient traffic distribution via Application Load Balancer
- Structured network segmentation using VPC, Subnets, Route Tables, IGW, and NAT
- Tier isolation ensuring the application tier stays private and protected
- Self-healing ability — ASG automatically replaces unhealthy instances

This architecture follows AWS best practices for security, availability, and scalability, making it ideal for hosting production-level applications.

- Key Takeaways

  - Private EC2 instances should never be exposed directly to the internet
  - ALB + ASG provide load balancing + fault tolerance + automatic scaling
  - Using a single NAT Gateway reduces cost while maintaining private subnet outbound access
  - Clear naming conventions make the setup organized and easier to maintain
  - Multi-AZ deployment ensures uptime even during AZ-level failures

---

- Closing Note

Building this environment helped reinforce concepts like network isolation, traffic flow, security layers, and scaling strategies on AWS.
Happy to assist anyone who is working on similar architectures or wants to learn more about AWS cloud design!

Thank you for reading — hope this helps your AWS journey! 🚀