# **Secure Two-Tier Architecture on AWS Using a Bastion Host**

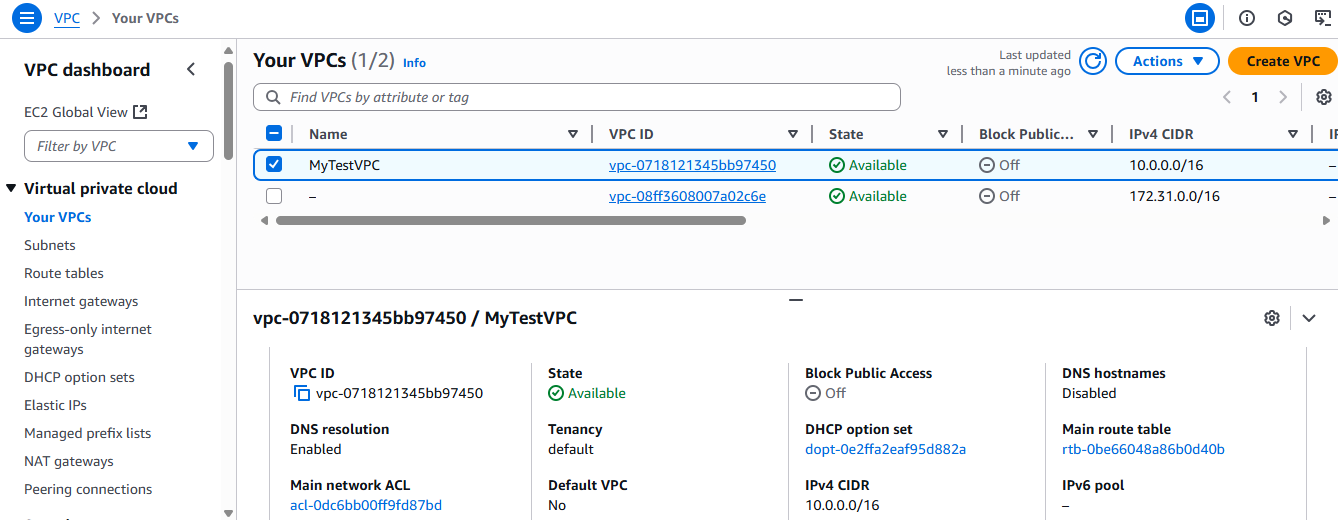
## **1. Overview**

This document details the design and implementation of a secure two-tier architecture on AWS, utilizing a bastion host for accessing a private EC2 instance. The purpose of this setup is to ensure that the private instance remains isolated from public internet access while still allowing secure administrative access via the bastion host.

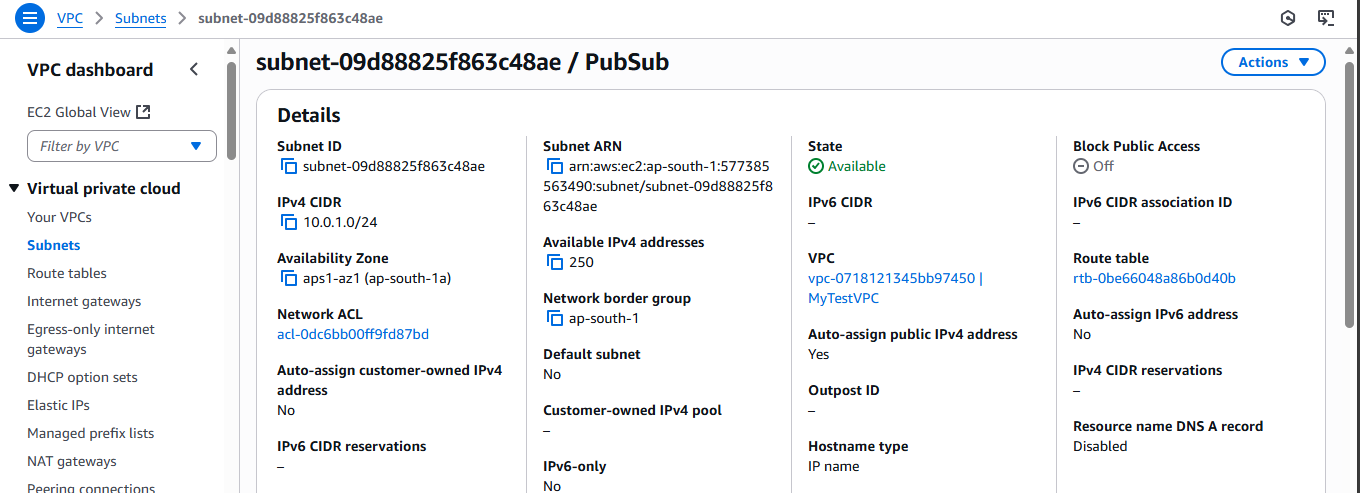
## **2. Architecture Design**

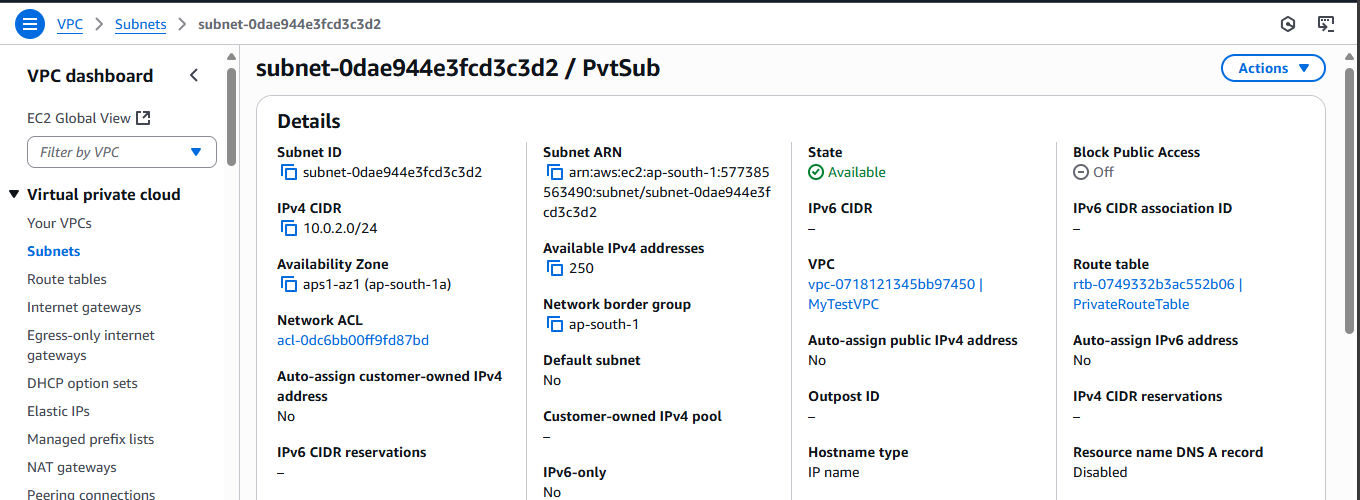
### **2.1 VPC Configuration**

* **VPC Name:** MyTestVPC
* **CIDR Block:** 10.0.0.0/16

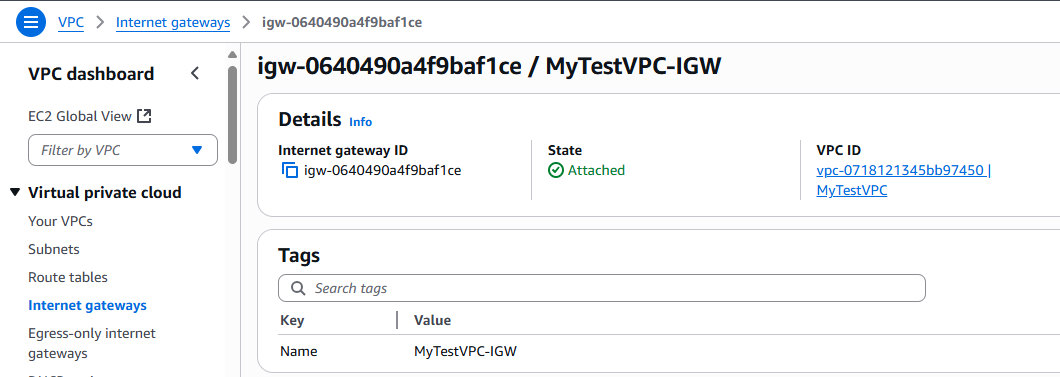


### **2.2 Subnets**

* **Public Subnet:** PubSub
  + CIDR Block: 10.0.1.0/24
  + Hosts the bastion host
  + Associated with a route table containing a route to the Internet Gateway  
    
* **Private Subnet:** PvtSub
  + CIDR Block: 10.0.2.0/24
  + Hosts the private EC2 instance
  + No direct route to the internet

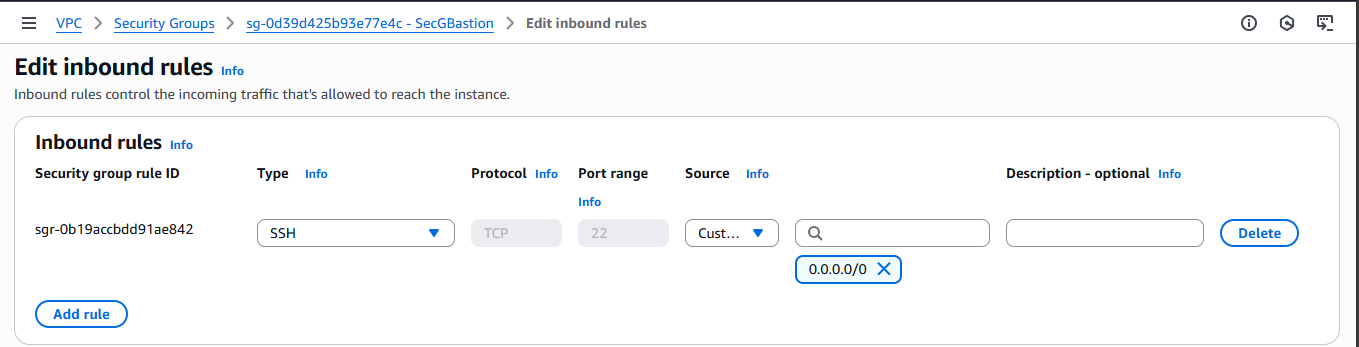
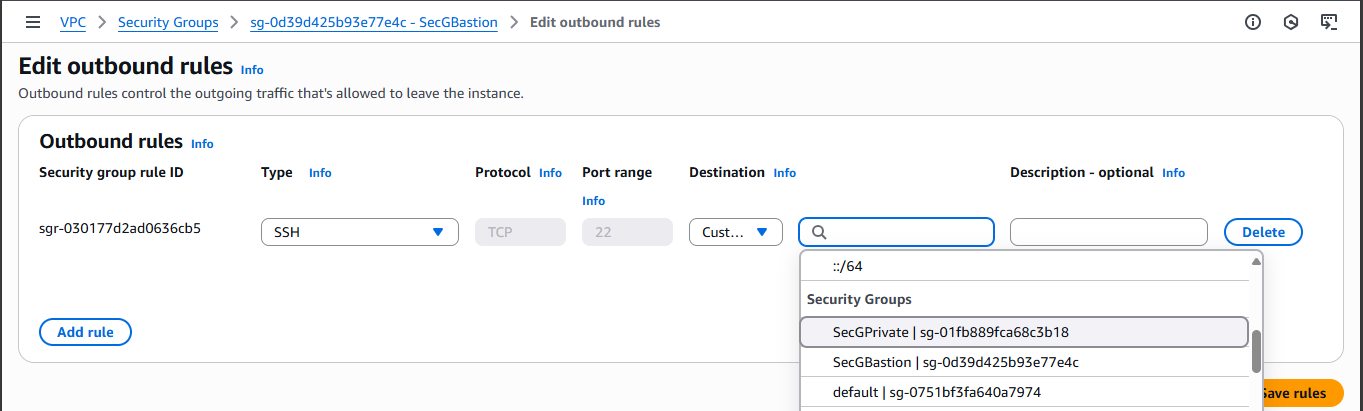


### **2.3 Internet Gateway**

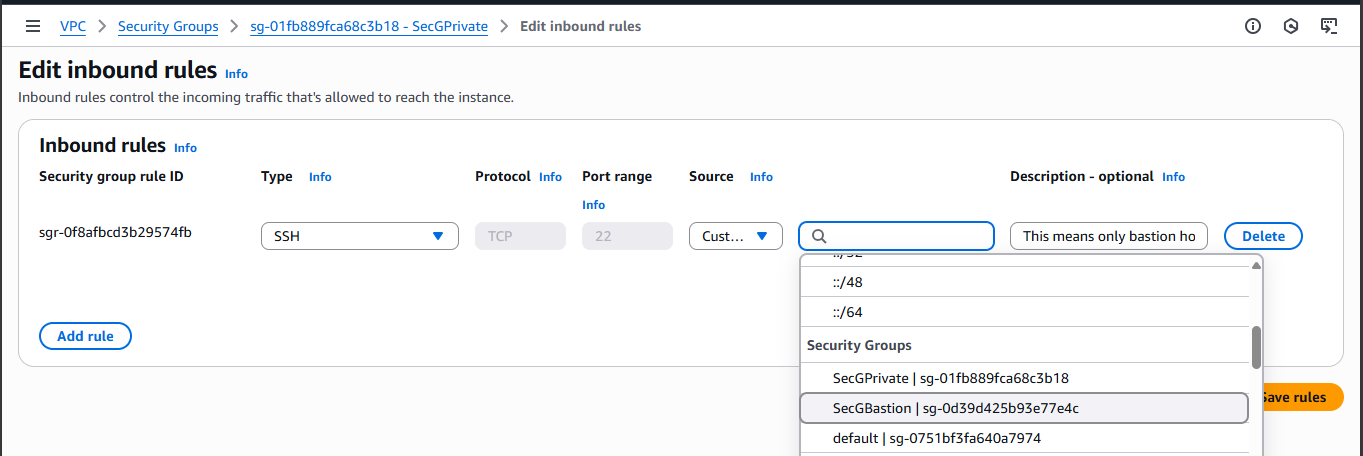
* **Name:** MyTestVPC-IGW
* Attached to MyTestVPC
* Associated with the route table for PubSub  
  

## **3. Security Groups**

### **3.1 SecGBastion (Bastion Host Security Group)**

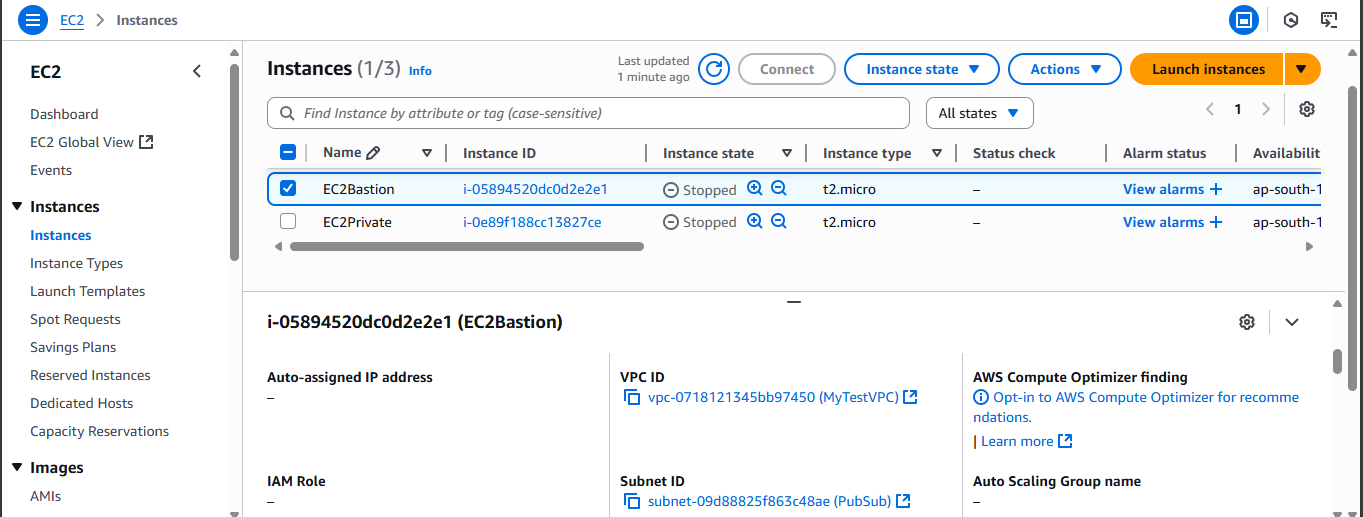
* **Inbound Rules:**
  + Allow SSH - TCP port 22  
    
* **Outbound Rules:**
  + The destination should be ‘SecGPrivate’ as shown in the image below.  
    

### **3.2 SecGPrivate (Private EC2 Security Group)**

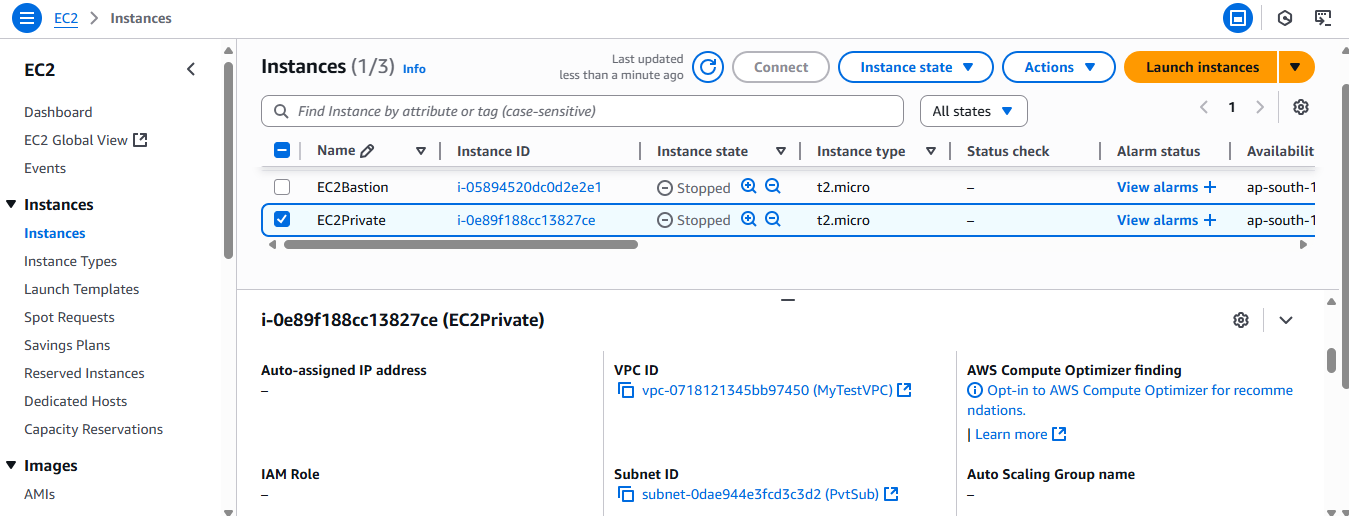
* **Inbound Rules:**
  + Allow SSH (TCP port 22) from the SecGBastion security group  
    

## **4. EC2 Instances**

### **4.1 Bastion Host (EC2Bastion)**

* Subnet: PubSub
* AMI: Amazon Linux 2 (Free Tier eligible)
* Instance Type: t2.micro or t3.micro (Free Tier eligible)
* Key Pair: MyKeyPair
* Security Group: SecGBastion
* Public IP: Enabled  
  

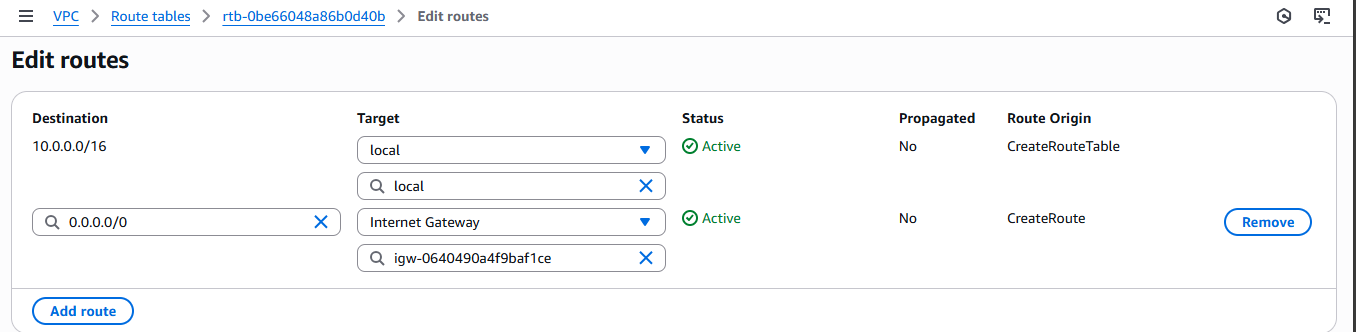
### **4.2 Private EC2 Instance (EC2Private)**

* Subnet: PvtSub
* AMI: Amazon Linux 2 (Free Tier eligible)
* Instance Type: t2.micro or t3.micro (Free Tier eligible)
* Key Pair: MyKeyPair
* Security Group: SecGPrivate
* Public IP: Disabled  
  

## **5. Routing Configuration**

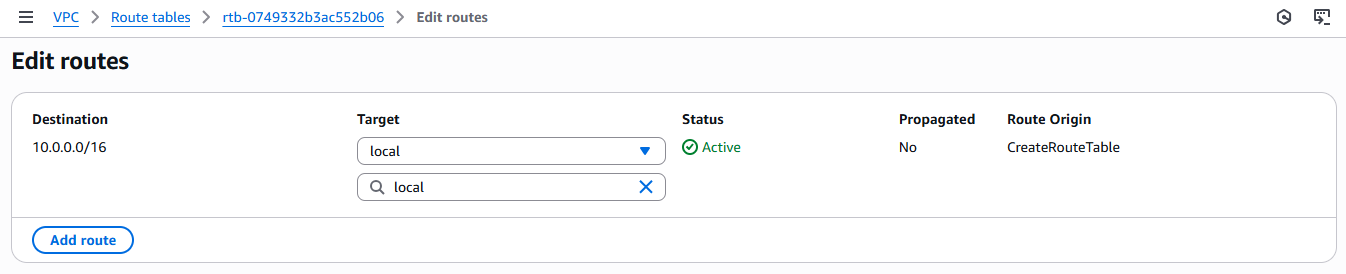
### **5.1 Public Subnet Route Table**

* Associated with PubSub
* Contains route:
  + 0.0.0.0/0 → MyTestVPC-IGW



### **5.2 Private Subnet Route Table**

* Associated with PvtSub
* No route to the internet



## **6. Access and Connectivity**

To access the private EC2 instance:

1. SSH into the bastion host using its public IP:

ssh -i MyEC2.pem ec2-user@<Bastion-Public-IP>

1. From the bastion host, SSH into the private EC2 instance using its private IP:

ssh -i MyEC2.pem ec2-user@<Private-EC2-Private-IP>



This ensures that the private instance is not exposed to the public internet.

## **7. Cleanup Steps**

To avoid incurring charges after testing, you can remove the following resources:

1. **Terminate EC2 Instances:**
   1. Terminate EC2Bastion and EC2Private.
2. **Delete EBS Volumes:**
   1. Check for and delete any unattached volumes.
3. **Remove Key Pairs:**
   1. If not reused elsewhere.
4. **Delete Security Groups:**
   1. SecGBastion and SecGPrivate, once instances are terminated.
5. **Delete Subnets, Route Tables, and Internet Gateway:**
   1. Disassociate and delete them from the VPC.
6. **Delete VPC:**
   1. After all associated resources are removed.
7. **Review Billing Dashboard:**
   1. Check for any remaining billable services such as snapshots or Elastic IPs.