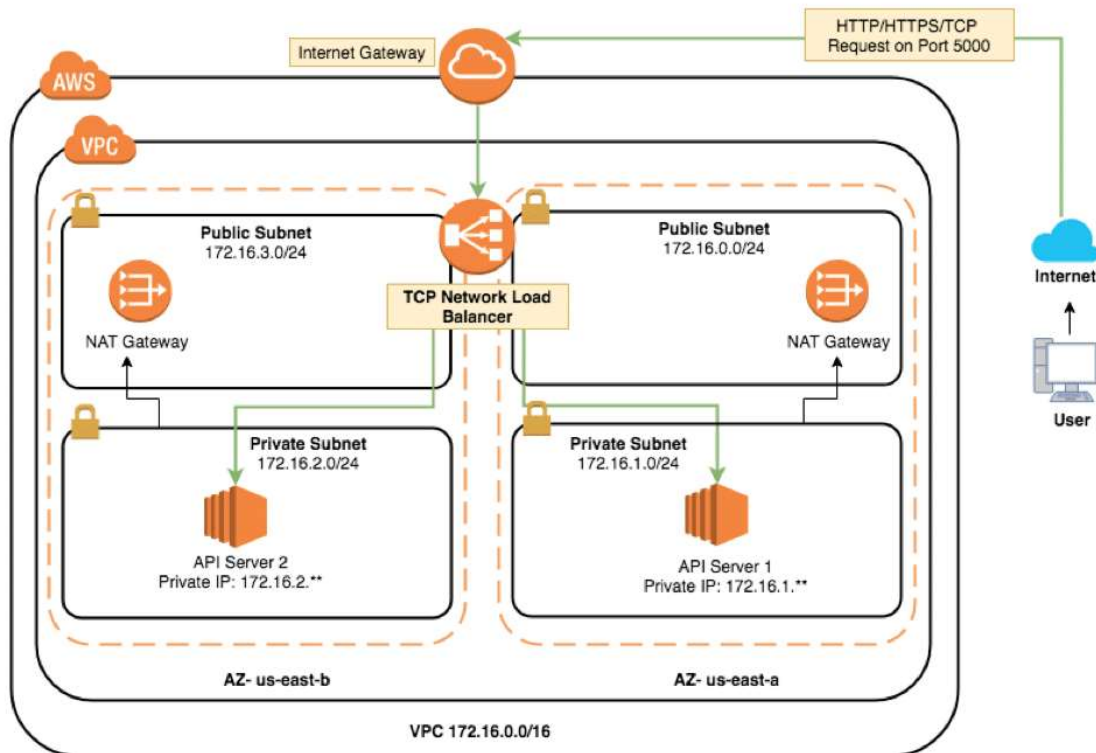


## How to Access Ec2 instance in a Private Subnet from the Internet

If you're here, you're probably experiencing a common issue: trying to access a certain port on an EC2 Instance located in a private subnet of the Virtual Private Cloud (VPC). A couple of months ago, we got a call from one of our customers that was experiencing the same issue. They wanted to open up their API servers on the VPC to one of their customers, but they didn't know how. In particular, they were looking for a solution that wouldn't compromise the security of their environment. We realized this issue is not unique to our customer, so we thought a blog post explaining how we solved it would be helpful!

To provide some context, once you have an API server within your VPC, it is closed to the outside world. No one can access or reach that server because of the strong firewall around it. There are a few ways around this, including Virtual Private Network (VPN) connections to your VPC, which allows you to open up private access. Unfortunately, this is not a viable solution if you need to open up your API server to the world, which was the case with our customer. The goal was to provide direct access from the internet outside the VPC for any user without VPN connection.

In order to solve this issue for our customer, one of the architecture changes we recommended was adding an internet-facing AWS TCP Network Load Balancer on the public subnet of the VPC. In addition to this load balancer, we also needed to create an instance-based target group.



Features of our diagram:

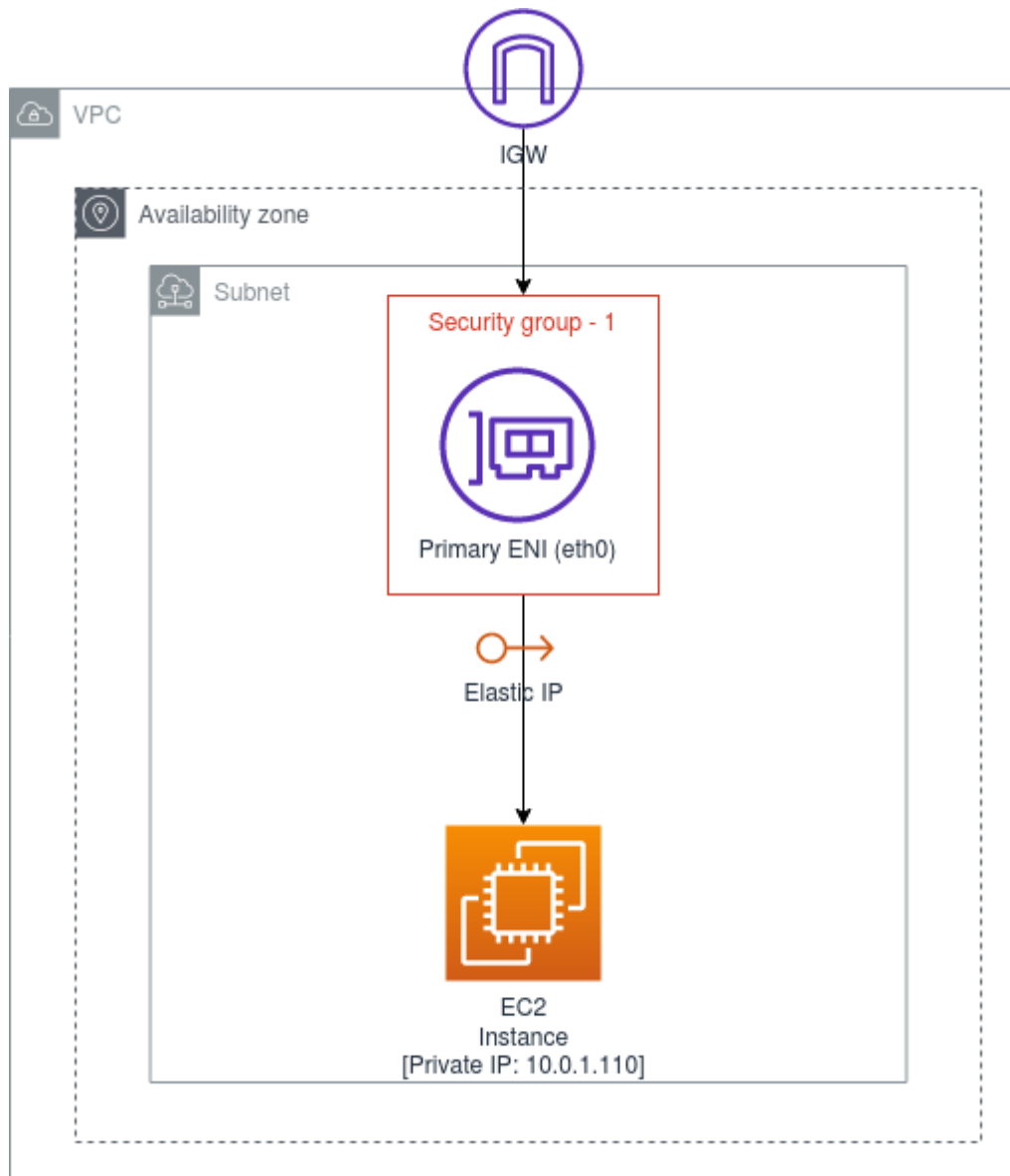
- Multi AZ: we used a private and public subnet in the same VPC in two different availability zones.
- Multi EC2 (API Servers): we deployed an API server in each private subnet in each availability zone.
- Multi NAT Gateways: a NAT gateway will allow the EC2 instances in the private subnets to connect to the internet and achieve high availability. We deployed one NAT gateway in the public subnets in each availability zone.
- TCP Load balancer health checks: a TCP load balancer will always redirect any user's requests to the healthy API servers. In case one AZ goes down, there will be another AZ that can handle any user's requests.

## Handling Elastic Network Interface(s) (ENIs) in AWS

## Elastic Network Interface (ENI)

ENI is a logical networking component in a VPC that represents a Virtual Network Card. It can have the following attributes.

1. A primary private IPv4 address
2. One or more secondary private IPv4 addresses
3. One Elastic IP Address (IPv4) per private IPV4 address
4. One public IPv4 address
5. One or more IPv6 addresses
6. One or more security groups
7. A MAC address
8. A source/destination check flag



## ENI — Key Features

1. Each instance in your VPC has a default network interface (the primary network interface — eth0) that assigns a private IPv4 address from the IPv4 address range of your VPC (See Figure 1).

2. You cannot detach the default (primary) network interface from an instance.
3. You can create an attach an additional/ secondary ENIs to an instance in your VPC. However, these ENIs should be created within the same availability zone of the EC2 instance that you are trying to attach your secondary ENI (See Figure 2). The number of network instances you can attach varies by instance type.
4. A Security Group is attached to an ENI not an EC2 instance. With this approach you can have multiple routes to the same EC2 instance with different security configurations.
5. You can create a network interface, attach it to an instance, detach it from an instance and attach it to another instance within the same Availability Zone. A network interface's attributes follow it as it is attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

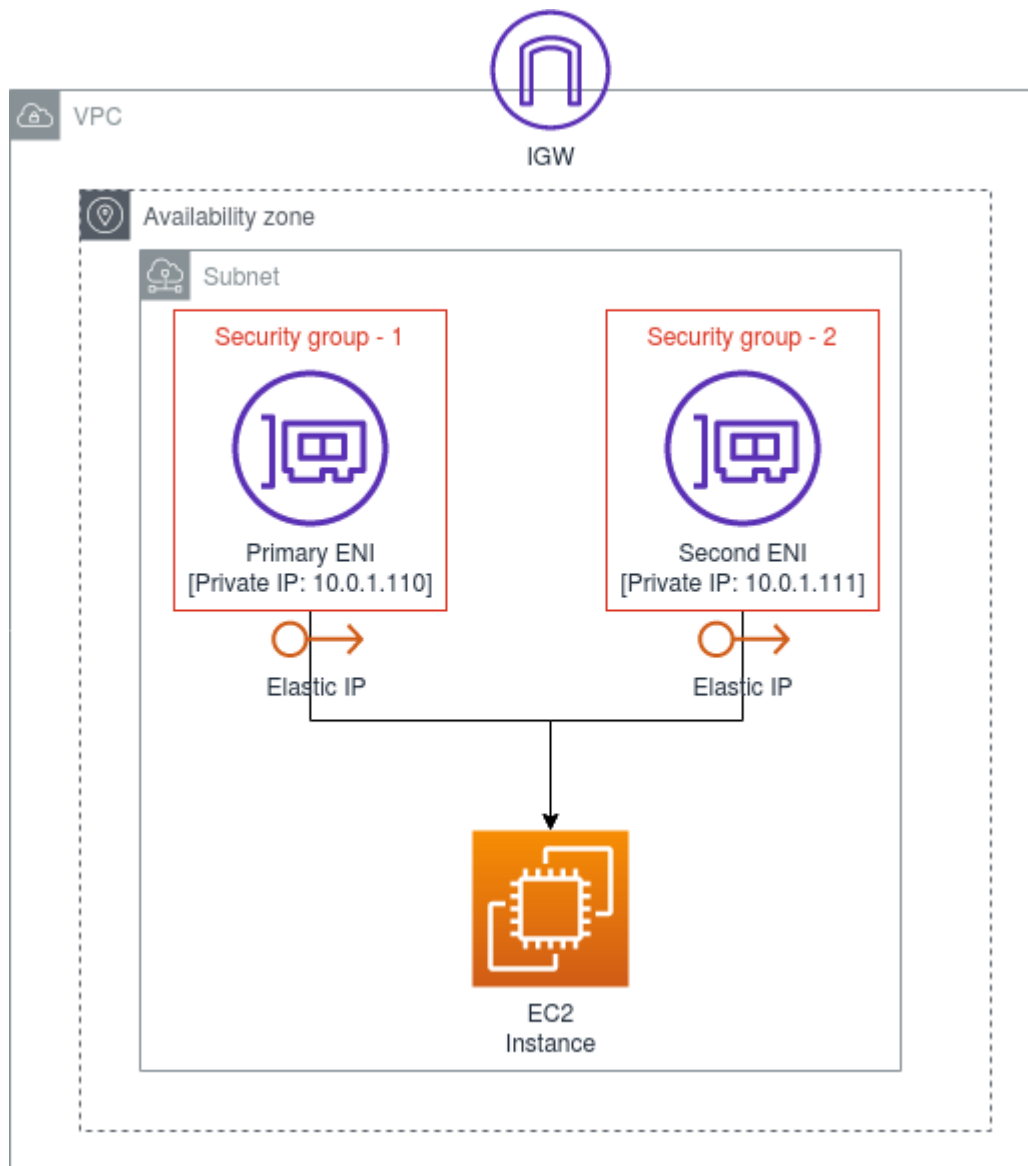


Figure 2 — Attaching additional ENIs to an EC2 instance

## The Scope

In this blog, we are primarily focusing on how we can create multiple ENIs for a single EC2 instance and attach multiple Elastic IPs for each ENI private IP. Using this approach you can create multiple routes to the same instance application installations.

We will discuss how we attach and detach ENIs to and from EC2 instances in a later blog.

## Task 1: Creating an EC2 instance and install Apache

Create an EC2 instance (Amazon Linux, t2.micro) and install Apache on it.

While creating the instance, create a Security Group (my-sg-1) and open port 80(HTTP) for future testing.

When you create an EC2 instance it does attach a default network interface (primary ENI — eth0) to it (See Figure 3).

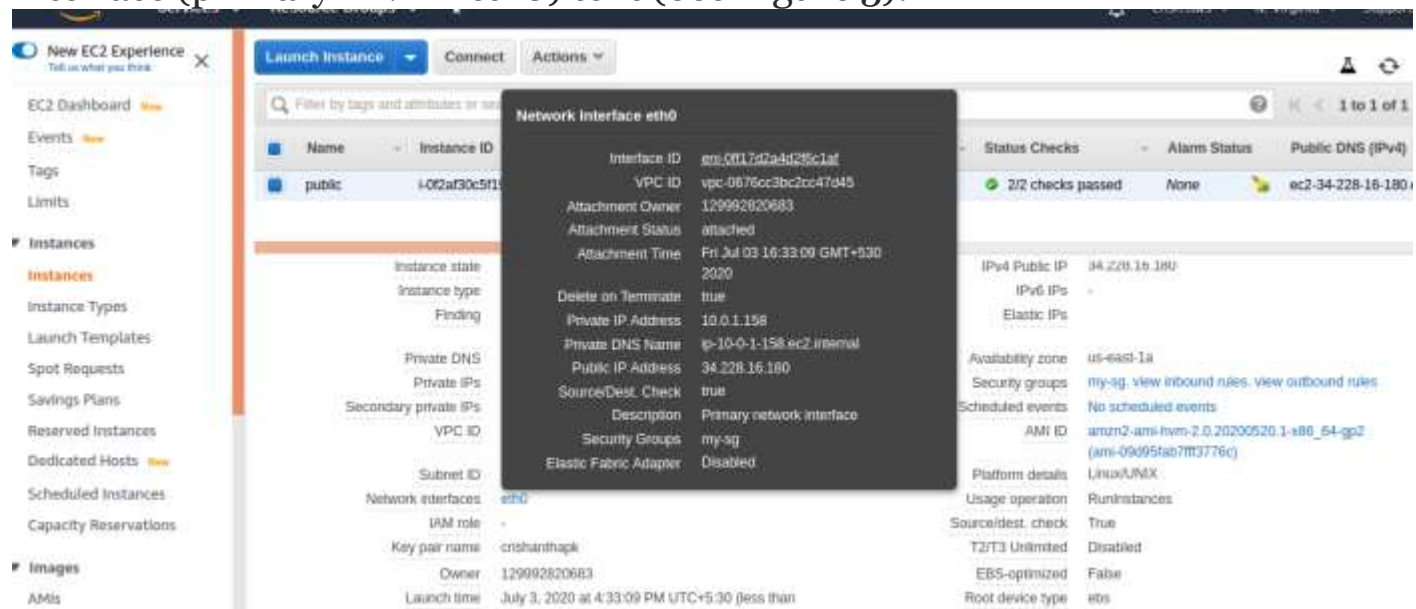


Figure 3 — The EC2 instance created with the default ENI

## Task 2: Allocate an Elastic IP

Now let's allocate a new Elastic IP to the primary ENI. (If your subnet is public and it is configured to have auto generated public IP(s) then this is not needed. But here we are going to showcase, how Elastic IP(s) can be allocated to ENIs)

Go to Elastic IP(s) → Click “Allocate Elastic IP Address” → Select Amazon Pool of IPv4 addresses → Click *Allocate* (See Figure 4)

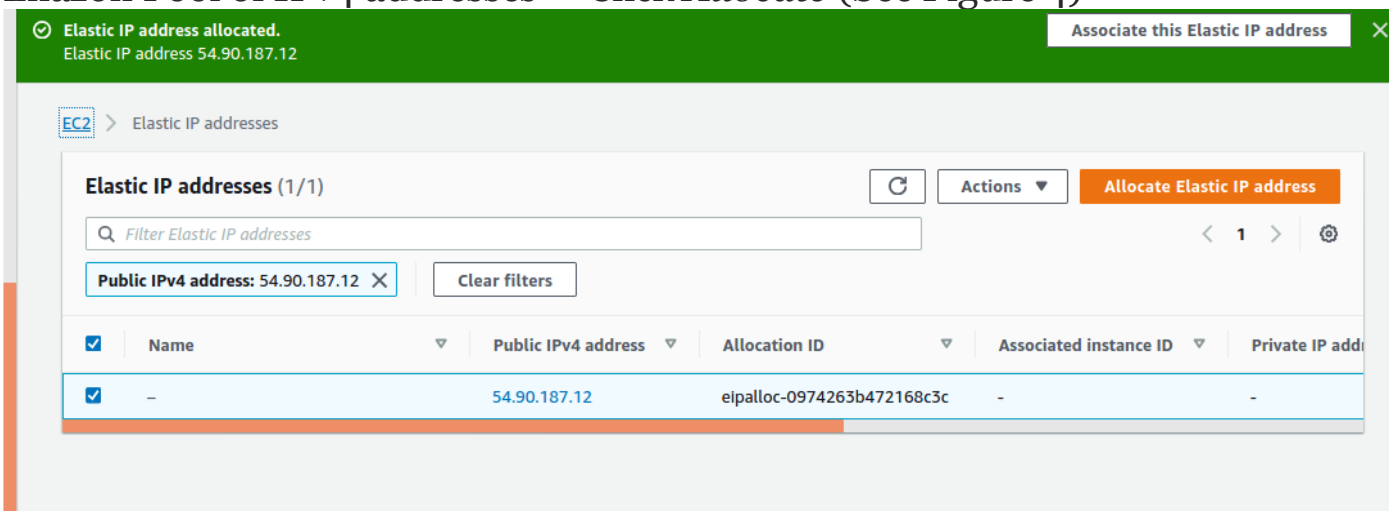


Figure 4 —Allocate a new Elastic IP

### Task 3: Associate the created Elastic IP to the primary/default ENI

Go to Elastic IP Addresses → Click *Actions* button and select *Associate Elastic IP* → Select *Resource Type* as *Network Interface* → Select the *Primary Network Interface* that you have created from the select Network Interface search box → Click *Associate*.



## Associate Elastic IP address


Choose the instance or network interface to associate to this Elastic IP address (54.90.187.12)

Elastic IP address: 54.90.187.12

### Resource type

Choose the type of resource with which to associate the Elastic IP address.

- ☐ Instance
- ☒ Network interface

 If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#).

### Network interface

 Choose a network interface

eni-0ff17d2a4d2f6c1af

### Private IP address

The private IP address with which to associate the Elastic IP address.

 Choose a private IP address

Figure 5 — Associate Elastic IP to the primary/ default ENI

## Task 4: Lets run the EC2 instance Apache installation using the Elastic IP via the primary / default ENI

Lets try to run the instance now with the assigned Elastic IP. You should see something similar to this. Here the index.html has a dummy text “Testing ENI” (See Figure 6).

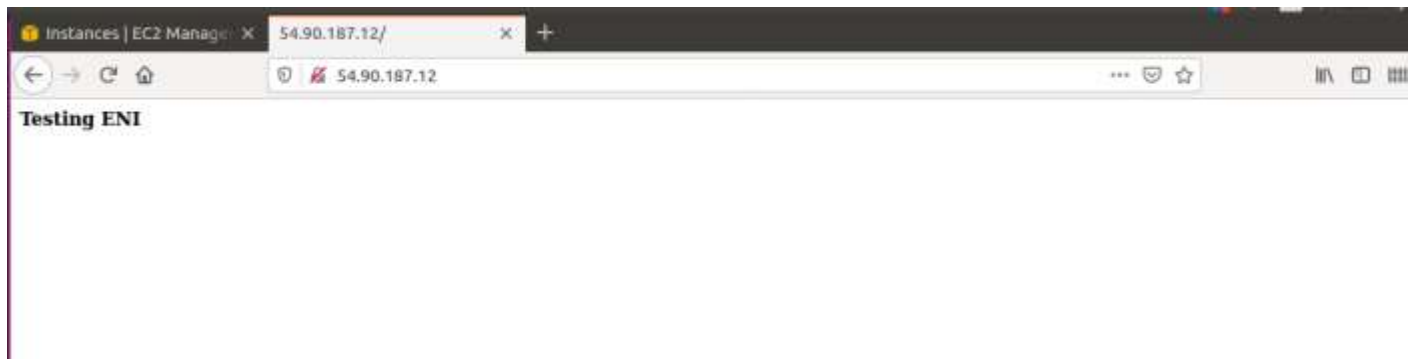


Figure 6 — Running an application via primary/ default ENI

Now, lets try to create another ENI and attach an Elastic IP with a new Security Group attached to it. This allows us to navigate the same instance via a different network address (new Elastic IP).

### **Task 5: Create the additional ENI (second ENI)**

As a prerequisite, create another Security Group (my-sg-2) without allowing any ports to it.

Now, create the additional ENI.

Go to *Network Interfaces* → Click *Create Network Interface* → Select the same subnet that you created the EC2 instance → Select *Auto Assign* for the IPv4 Private IP → Select the Security Group (my-sg-2) → Create ENI (See Figure 7).

## Create Network Interface

Description  ⓘ

Subnet\*  ⓘ

IPv4 Private IP ☒ Auto-assign ⓘ  
☐ Custom

Elastic Fabric Adapter ☐ ⓘ

Security groups\*  ⓘ

Filter by attributes or search by keyword

Group ID	Group name	Description
sg-04a0eacd5...	default	default VPC security group
sg-0b4054db8...	my-sg	my-sg
sg-0e63aabda...	my-sg-2	my-sg-2

Figure 7 — Creating an additional ENI

## Task 6: Create another Elastic IP

Create Network Interface ⓘ ⓘ ⓘ ⓘ Actions ▾

Filter by tags and attributes or search by keyword

Name	Network interface ID	Subnet ID	VPC ID	Zone	Security group	Description	Instance ID	Status
	eni-04122349503675c6d	subnet-04b56...	vpc-0676cc3b...	us-east-1a	my-sg-2	second-eni		Available
	eni-0ff17d2a4d2f5c1af	subnet-04b56...	vpc-0676cc3b...	us-east-1a	my-sg	Primary network I...	i-0f2af30c5f93c194b	In Use

Select a network interface above

Figure 8 — Creating an Elastic IP for the additional ENI

## Task 7: Associate the Elastic IP (created under step 6) to the additional ENI

## Associate Elastic IP address


Choose the instance or network interface to associate to this Elastic IP address (52.202.108.130)

### Elastic IP address: 52.202.108.130

#### Resource type

Choose the type of resource with which to associate the Elastic IP address.

- ☐ Instance
- ☒ Network interface

 If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)

#### Network interface

eni-04122349503675c6d

#### Private IP address

The private IP address with which to associate the Elastic IP address.

Choose a private IP address

#### Reassociation

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

- ☐ Allow this Elastic IP address to be reassociated

Cancel

Associate

Figure 9 — Associate Elastic IP to the additional ENI

### Task 8: Attach the additional ENI to the EC2 instance.

However, in order to attach this additional ENI to the EC2 instance, you are required to do the following configuration.

Go to EC2 instances → Select the EC2 instance → Select *Actions* → Select *Networking* → Select *Attach Network Interface* and select second ENI you have created (See Figure 10).

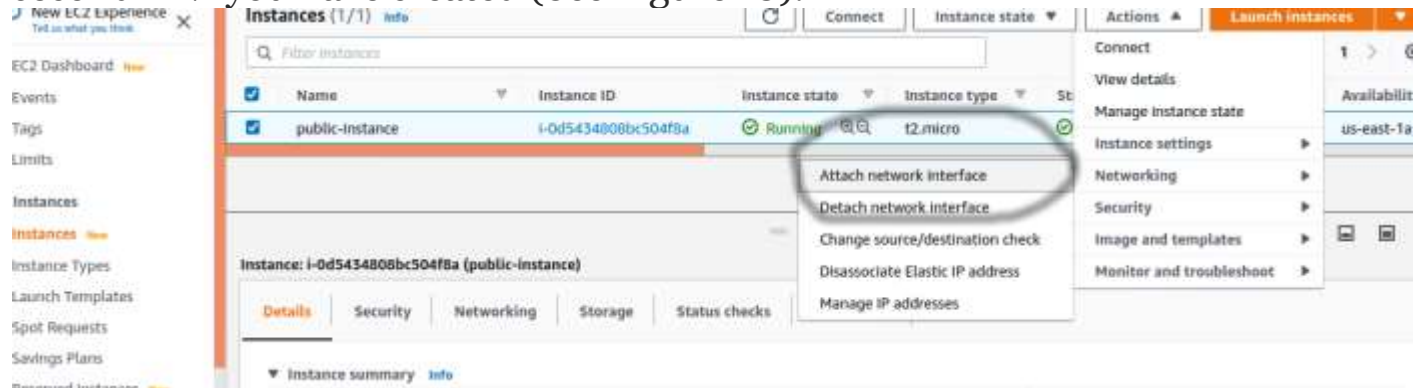


Figure 10

Finally, as a result of the above exercise, the EC2 instance is now attached with two ENIs.

1. Default / Primary ENI (eth0)
2. Additional ENI (eth1)

Now you would see something similar to the following.

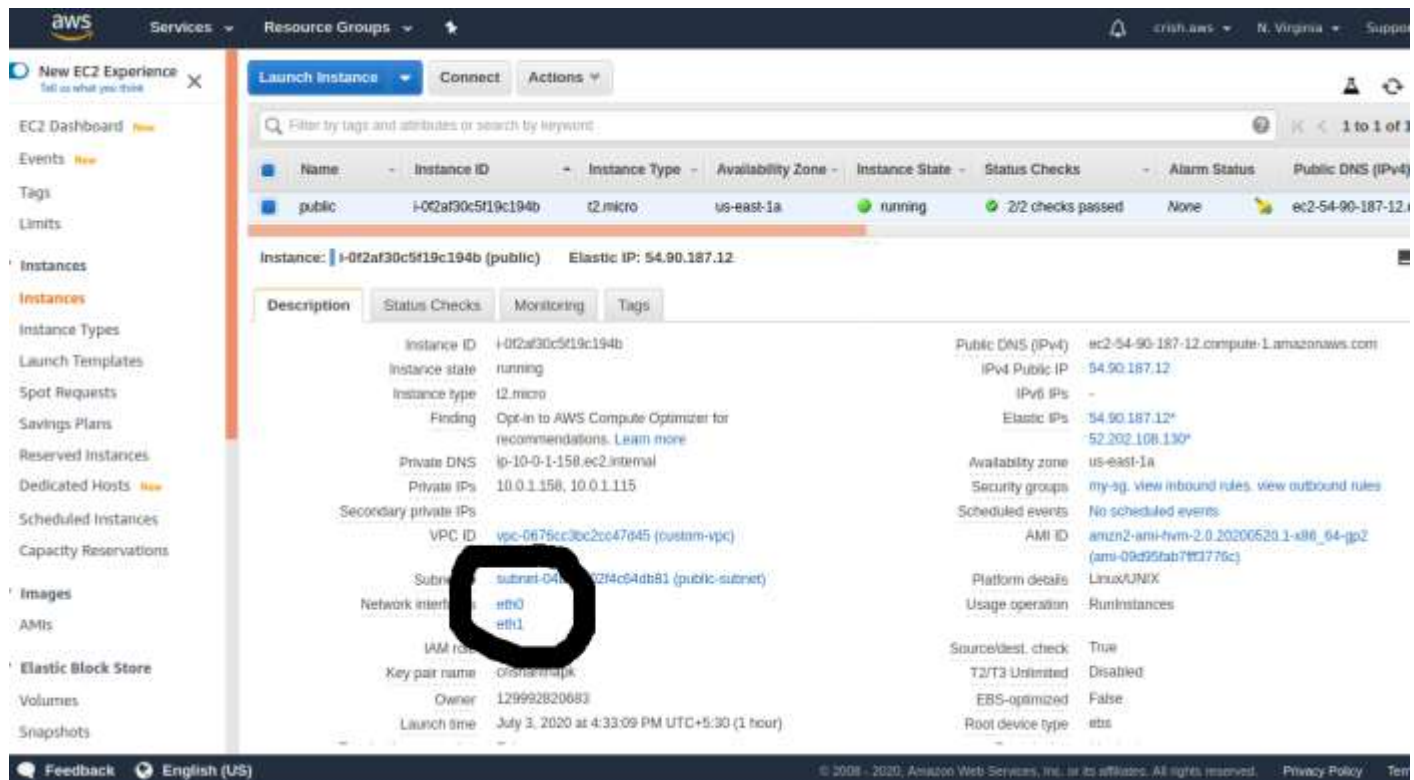


Figure 10 — The EC2 instance with both eth0 and eth1

# Secure AWS VPC Design and Configuration

## AWS Virtual Private Cloud (VPC)

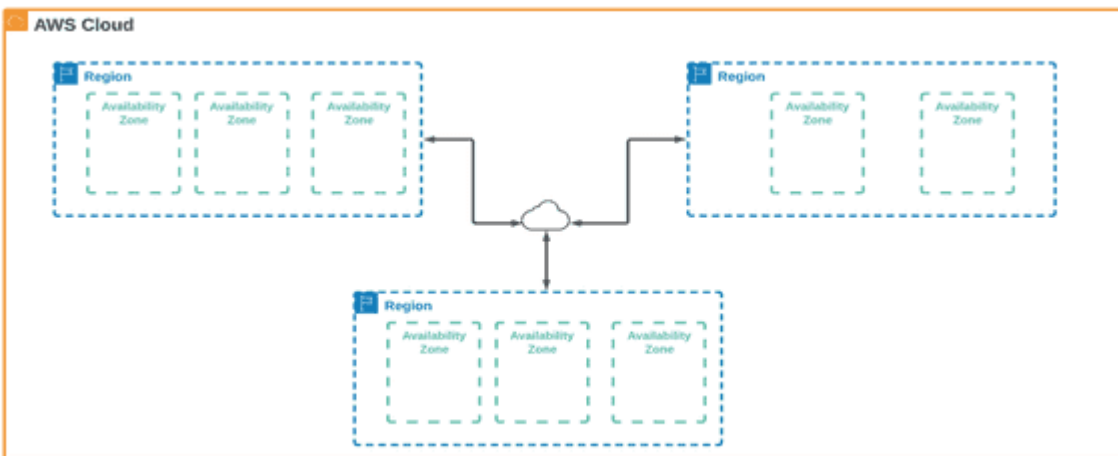
An AWS Virtual Private Cloud (VPC) is the keystone of AWS networking. VPC is an isolated virtual network that you define in your AWS infrastructure. Organizations use VPCs for different usage scenarios such as:

- Hosting Applications
- EC2 Management
- Back-up and Disaster Recovery
- Hybrid Cloud Solutions

- Cloud Migration and Data Center Operations
- Secure data and AWS Resources

VPC configuration and management could be difficult and confusing for most customers. It could be done with AWS Management Console, AWS CLI, AWS CloudFormation, or Terraform templates. On the other hand, AWS VPC security is one of the issues that should be considered.

## AWS Global Infrastructure

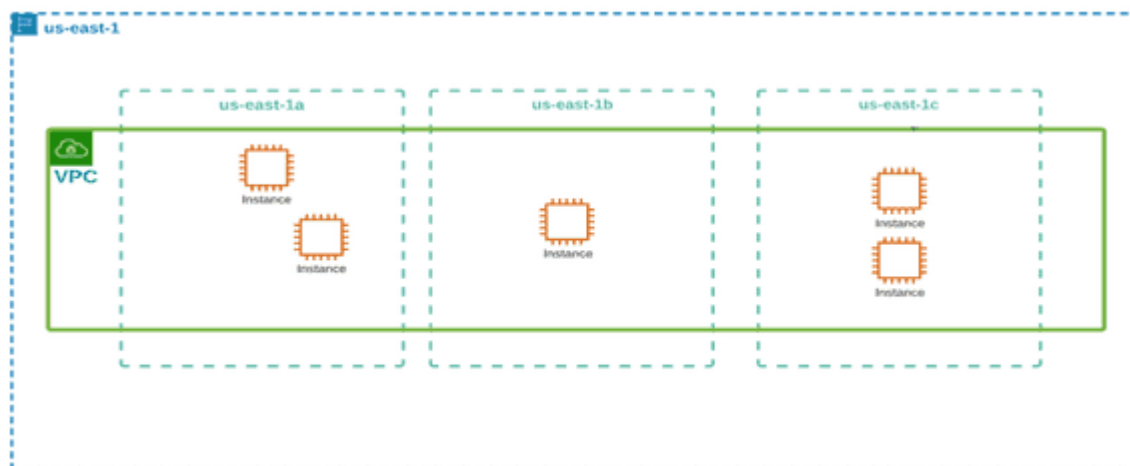


AWS Global Infrastructure has different concepts. First of all, a region is defined as an isolated geographical area. A region has different availability zones which are one or more data centers that have redundant power, networking between them. Every region has at least two availability zones. With different availability zones, AWS customers ensure high availability for customer solutions. For now, AWS announced [25 launched regions and 80 availability zones](#). New regions and availability zones continue to be launched day by day.

# Public and Private Subnets on VPC

The first thing to consider when starting a custom VPC design is choosing a CIDR block for your custom VPC. For example, if you want to use different VPCs together, VPC CIDR blocks should not overlap. You cannot alter or modify your VPC's CIDR block after creating it, so you should pick the CIDR block carefully.

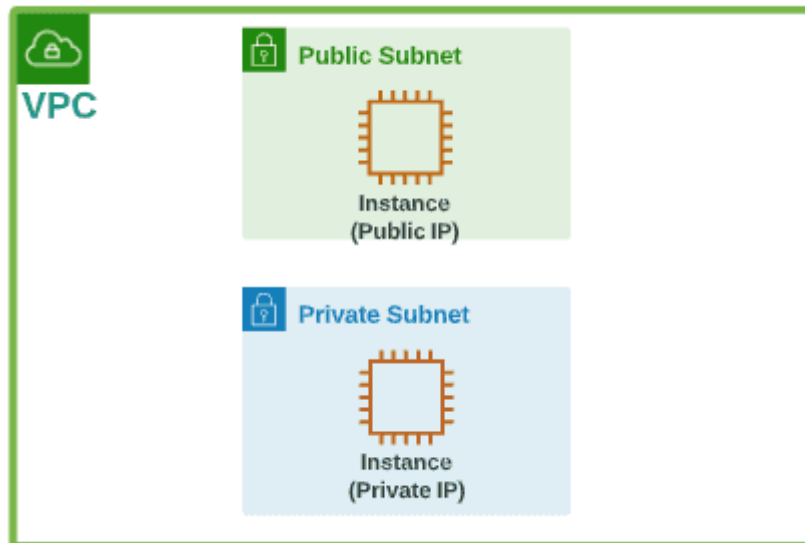
Once you create a custom VPC, you should create subnets within this VPC. Subnets are very typical of what you would configure in your data center. When you launch a subnet, it's within an availability zone. We ensure high availability by using different availability zones. While a public subnet has Internet access, a private subnet should not have public access directly.



Note: On AWS Environment, a default VPC and public subnet are automatically created for each region in an AWS account. Default VPC should be removed if you want to build simple and secure data storage in your environment. Also, the number of VPCs per region in an account is limited to five and this is a soft limit. If you want all your VPC designed by your team, you should delete the default VPC.



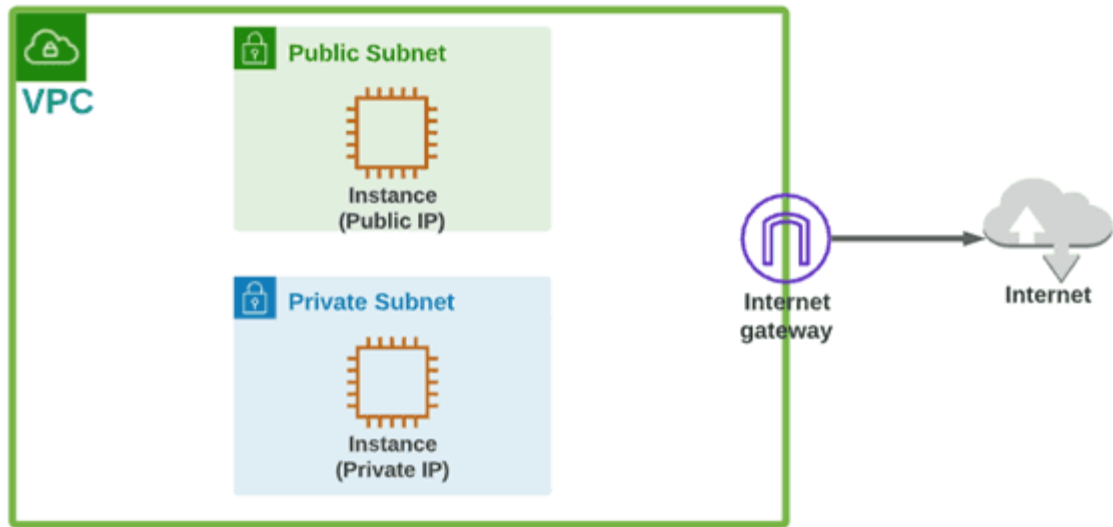
## 1. A Public IP address



When a public subnet resource wants Internet access, a Public IP address should be defined. On AWS environment, a Public IP address can be assigned in two ways:

Auto-assign Public IP address	Elastic IP address
Lost when the instance stopped	Static Public IP address
No cost	Charges apply when the Elastic IP Address is not attached to a running EC2 instance.
Cannot be moved between instances	Can be moved between instances

## 2. An Internet Gateway Attached to VPC

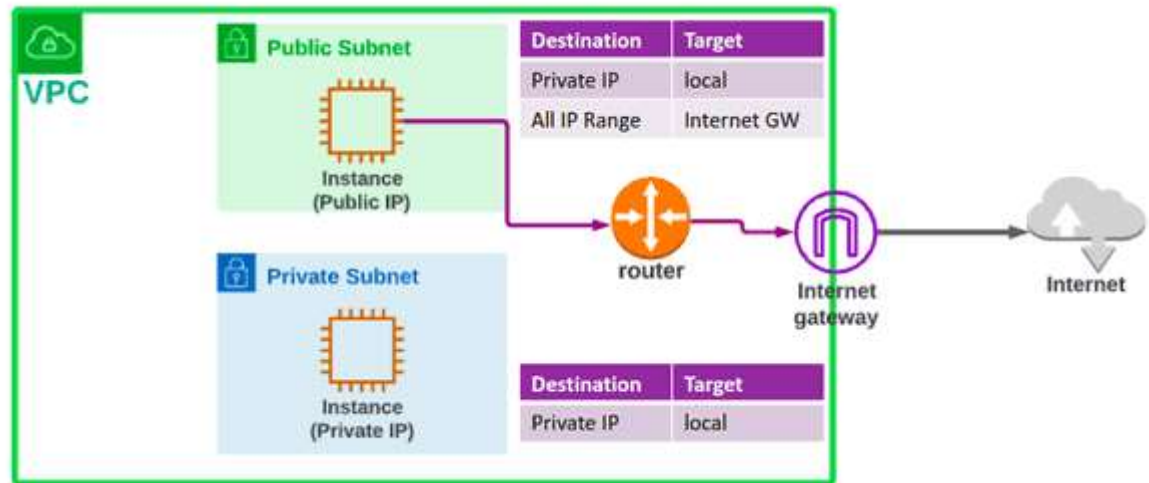


Internet Gateway is a highly scaled gateway managed by AWS. It performs 1:1 NAT and maps your private IP address to a public IP address.

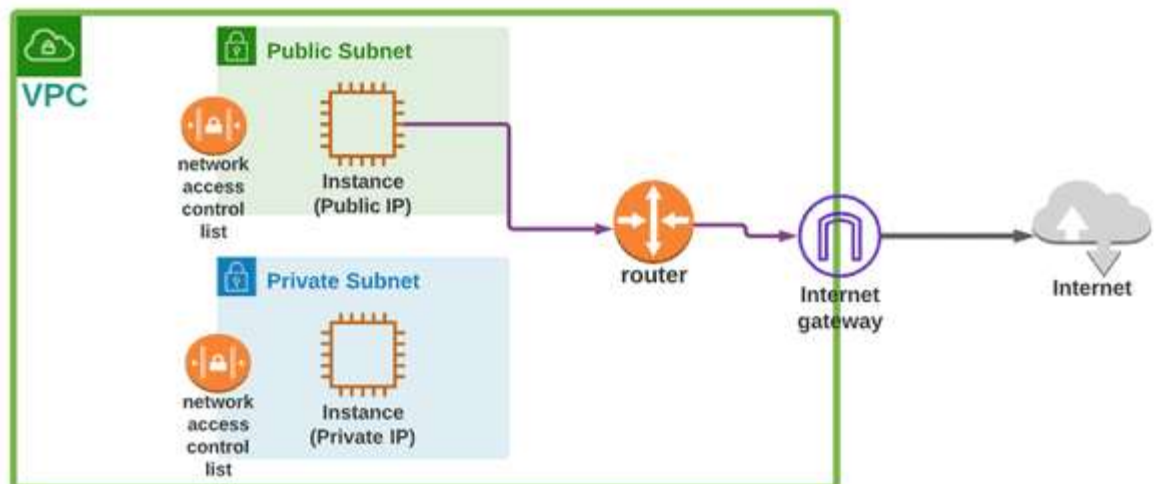
### 3. A Route Table to an Internet Gateway

As the name suggests, the routing table is used for routing your resources in VPC to your Internet Gateway. Public and private subnet have local targets for the private IP

range.

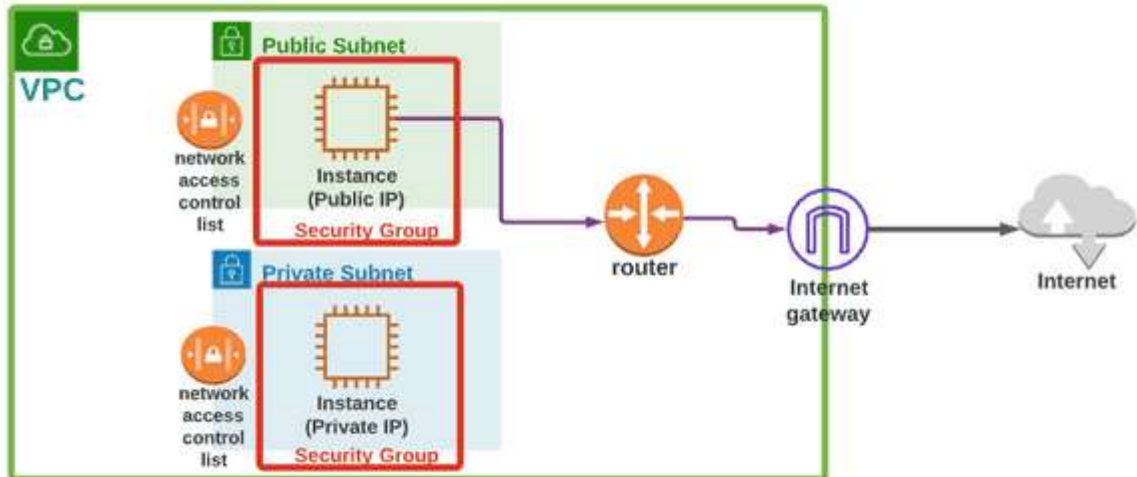


#### 4. Network Access Control List (NACL) for VPC



Network ACLs allow you to define the rules for the traffic flow. They are defined as subnet level. By default configurations, you will get default inbound and outbound traffic rules for the NACL. Default inbound and outbound rules allow all traffic.

## 6. Security Groups



## An Example Scenario: Hosting a Web Application

A client wants to host a web application on a Web server in an AWS environment. The client needs a database server for user data. The database server should connect with a web application. This web application is related to banking operations, so network and server security is the priority.

What is the easiest and most secure way to configure this network and server environment?

The database server does not need any traffic coming from outside the VPC. It only needs traffic from the web server subnet. For these types of client requirements, we should:

- Create a VPC
- Create a private and public subnet
- Configure an Internet Gateway
- Configure a Route Gateway
- Create EC2 for web and database

- Configure Security Group rules
- Configure Network ACL rules

