Hands On: AWS VPC Essentials

(Part - 1)

Here I am explaining the AWS VPC core components with Hands-on experience based on an Architectural diagram.

**In part-1, focused on below elements;**

* **IP CIDR Ranges for VPC and Subnets**
* **AWS VPC**
* **Subnets**
* **Internet Gateway**
* **Route Table**
* **Security Groups and NACL**
* **Bastion Host**
* **NAT Gateway and NAT Instance**

**IP CIDR Blocks for VPC and Subnets**

**CIDR – Classless Inter Domain Routing**

* Classless Inter-Domain Routing (CIDR) is an IP address allocation method that improves data routing efficiency on the internet.
* With CIDR, your organization has more flexibility in assigning IP addresses and routing data between devices.
* CIDR is based on the idea that IP addresses can be allocated and routed based on their network prefix rather than their class, which was the traditional way of IP address allocation.
* CIDR addresses are represented using a slash notation, which specifies the number of bits in the network prefix. For example, an IP address of 192.168.1.0 with a prefix length of 24 would be represented as 192.168.1.0/24. This notation indicates that the first 24 bits of the IP address are the network prefix and the remaining 8 bits are the host identifier.

**VPC CIDR Block**

When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses).

create a VPC, we recommend that you specify a CIDR block from the private IPv4 address ranges as specified;

10.0.0.0 - 10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

*(The "10" refers to the starting IP address of the range, which is 10.0.0.0.*

*The "/8" indicates that the first 8 bits of the IP address are fixed and represent the network part of the address and the remaining 24 bits can be used for host addresses within this network.)*

**Subnet CIDR Block**

The CIDR block of a subnet can be the same as the CIDR block for the VPC (to create a single subnet in the VPC).

If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap.

The allowed IPv4 CIDR block size for a subnet is between a /28 netmask and /16 netmask. The first four IP addresses and the last IP address in each subnet CIDR block are not available for your use, and they cannot be assigned to a resource.

For example, in a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:

* 10.0.0.0: Network address.
* 10.0.0.1: Reserved by AWS for the VPC router.
* 10.0.0.2: Reserved by AWS.
* 10.0.0.3: Reserved by AWS for future use
* 10.0.0.255: Network broadcast address.

**AWS VPC**

With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined.

**Use Cases**

* Launch a simple website or blog:

Improve your web application security posture by enforcing rules on inbound and outbound connections.

* Host multi-tier web applications:

Define network connectivity and restrictions between your web servers, application servers, and databases.

* Create Hybrid Connections:

Build and manage a compatible VPC network across your AWS services and on premises.

**Features**

* **VPC Flow logs:**

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs, Amazon S3

* **IP Addressing:**

IP addresses enable resources in your VPC to communicate with each other and with resources over the internet. Amazon VPC supports both the IPv4 and IPv6 addressing protocols.

* **Ingress Routing:**

With this feature, you can route all incoming and outgoing traffic flowing to/from an internet gateway or virtual private gateway to a specific Amazon EC2 instance’s elastic network interface. Configure your virtual private cloud to send all traffic to a gateway or an Amazon EC2 instance before it reaches your business workloads.

* **NACL – Network Access Control List:**

A network access control list (network ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.

***Stateless Nature****: NACLs are stateless, which means they do not keep track of connection state. For every request, both inbound and outbound rules must be explicitly defined. If an inbound request is allowed, the corresponding outbound response must also be explicitly allowed.*

* **Security Group:**

Create security groups to act as a firewall for associated Amazon EC2 instances, controlling inbound and outbound traffic at the instance level.

***Stateful Nature****: Security groups are stateful, which means that if you allow an inbound request from a specific IP and port, the response traffic is automatically allowed regardless of outbound rules*

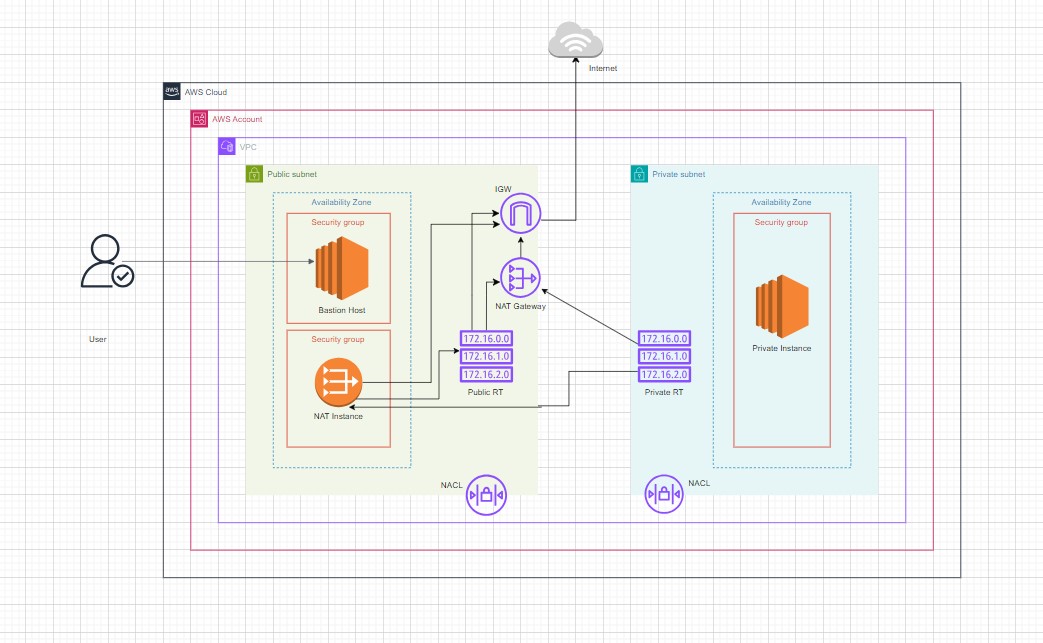
* **Traffic Mirroring:**

Traffic Mirroring is an Amazon VPC feature that you can use to copy network traffic from an elastic network interface of type interface. You can then send the traffic to out-of-band security and monitoring appliances for:

* Content inspection
* Threat monitoring
* Troubleshooting

The security and monitoring appliances can be deployed as individual instances, or as a fleet of instances behind either a Network Load Balancer or a Gateway Load Balancer with a UDP listener.

**Configuration of VPC Architectural Diagrams**



Login to AWS Account with as IAM User with AdministratorFullAccess

**Create VPC**

* VPC Dashboard. Select VPCs. Click Create VPC.
* Resource to create — VPC only
* Name tag — “MyCloud”
* IPv4 CIDR manual input
* IPv4 CIDR “10.0.0.0/16”
* Create VPC
* Filter by VPC to see this VPC and its resource only.

**Create Subnets**

A subnet is a range of IP addresses in your VPC. You can create AWS resources, such as EC2 instances, in specific subnets’ IP addresses in your VPC. You can create AWS resources, such as EC2 instances, in specific subnets.

**Public subnet** – The subnet has a direct route to an internet gateway. Resources in a public subnet can access the public internet.

**Private subnet** – The subnet does not have a direct route to an internet gateway. Resources in a private subnet require a NAT device to access the public internet.

**Create Public Subnet:**

* VPC Dashboard. Select Subnets. (Filter VPC by Filter by VPC and select MyCloud)
* Click Create Subnet (AZ1)
* VPC — MyCloud
* Subnet name — public-subnet-1a

(Availability Zone — US East (N. Virginia) / us-east-1a)

IPv4 CIDR block — 10.0.1.0/24

**Create Private Subnet:**

* Click Create Subnet
* VPC — MyCloud
* Subnet name — private-Subnet-1a

Availability Zone — US East (N. Virginia) / us-east-1a

IPv4 CIDR block — 10.0.2.0/24

Click Create Subnet

*We can create public and private subnets in two AZ’s for better availability*.

**Internet Gateway**

An internet gateway is a **horizontally scaled**, redundant, and highly available VPC component that allows communication between your VPC and the internet.

Only one VPC can be attached to an Internet gateway and vice versa.

* VPC Dashboard - Internet Gateway > Create Internet Gateway.
* Name tag – MyCloud-IGW
* Click Create Internet Gateway.
* Attach to a VPC, or Actions. Attach to a VPC.

Available VPCs. Select - MyCloud

**Route Table**

A Route Table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed.

key concepts for route tables.

* **Main route table**—The route table that automatically comes with your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table.
* **Custom route table**—A route table that you create for your VPC.
* **Destination**—The range of IP addresses where you want traffic to go (destination CIDR). For example, an external corporate network with the CIDR 172.16.0.0/12.
* **Target**—The gateway, network interface, or connection through which to send the destination traffic; for example, an internet gateway.
* **Route table association**—The association between a route table and a subnet, internet gateway, or virtual private gateway.
* **Subnet route table**—A route table that's associated with a subnet.
* **Local route**—A default route for communication within the VPC.

**Create a Public route table**

* + PC Dashboard - Left Panel. Select Route Tables.

*(Note. There is a route table created automatically when VPC was created. This is called Main route table. It is private by default.)*

* + Click Create route table
  + Name — “MyCloud-public-RT”
  + VPC — MyCloud
  + Click Create route table.

**Edit Subnet Association for Public RT**

* + In PRT Details, select Subnet associations tab.
  + Click Edit subnet associations.
  + In Available subnets, Select the public subnet MyCloud-public-1a- Save associations.

**Add the public routes**

* In PRT Details, select Routes tab > Edit routes > Add route
* Destination — 0.0.0.0/0(internet)
* Target — MyCloud-IGW
* Click Save Changes
* Check the new route in Routes tab of PRT details page

**Create a private Route Table**

**Create private route table** using the same steps below, change the RT name to

“MyCloud-private-RT”

Follow the same steps mentioned above for **subnet association**, but here we should choose the private subnet MyCloud-private-1a.

For time being we skipped the step for edit routes now, will edit after create NAT Gateway.

**Launch EC2 Instances**

**Bastion Host (Public-Instance)**

* Navigate to the EC2 Dashboard
* Click on create Instance
* Give Instance Name – MyCloud-Bastion-ec2.
* Select AMI and Instance type as

AMI: Amazon Linux 2023

Type: t3. micro (both are free tier eligible)

* Create keypair with name **-** MyCloud-public-ec2-key
* **Configure Network**: Under VPC, select our VPC – MyCloud
* Select subnet – MyCloud-public-1a
* Enable Auto-assign public IP
* Click on create security Group

Security Group name – “MyCloud-public-SG”

Edit Inbound rules:

Source: 0.0.0.0 – port :80(http) and 443(https)

Source: Custom IP – port: 22(SSH)

Outbound rules keep default as allow all traffic.

* Edit Userdata with below bash script;

#!/bin/bash

# Use this for your user data (script from top to bottom)

# install httpd (Linux 2 version)

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<h1>Hello World from $(hostname -f)</h1>" > /var/www/html/index.html

**Private Instance**

* Click on launch Instance
* Follow the same step for MyCloud-private-ec2
* Choose Subnet as - MyCloud-private-1a
* Select security group as - MyCloud-private-SG

Edit Inbound rules:

Source: Custom IP (10.0.1.0/24 – from public subnet of MyCloud VPC) – port: 22(SSH)

Outbound rules keep default as allow all traffic.

* Didn’t edit userdata.
* Click on Launch Instance.

**Security Group and NACL**

**Security Group:**

A security group controls the In-bound and Out-bound traffic to the resources that it is associated with.

Security Groups operates at resource level and supports only allow rules

**Stateful: Return traffic is automatically allowed, regardless of any rules**

Here:

MyCloud-public-SG

Edit Inbound rules:

Source: 0.0.0.0 – port :80(http) and 443(https)

Source: Custom IP – port: 22(SSH)

MyCloud-private-SG

Edit Inbound rules:

Source: Custom IP (10.0.1.0/24 – from public subnet of MyCloud VPC) – port: 22(SSH)

**NACL – Network Access Control List**

A network access control list (ACL) allows or denies specific inbound or outbound traffic at the subnet level.

There is no additional charge for using network ACLs.

Your VPC automatically comes with a **Default network ACL, it allows all inbound and outbound** IPv4 traffic and, if applicable, IPv6 traffic.

You can associate a network ACL with multiple subnets. However, a subnet can be associated with only one network ACL at a time

Network ACLs can't block DNS requests to or from the Route 53.

**NACLs are stateless, which means that information about previously sent or received traffic is not saved. If, for example, you create a NACL rule to allow specific inbound traffic to a subnet, responses to that traffic are not automatically allowed.**

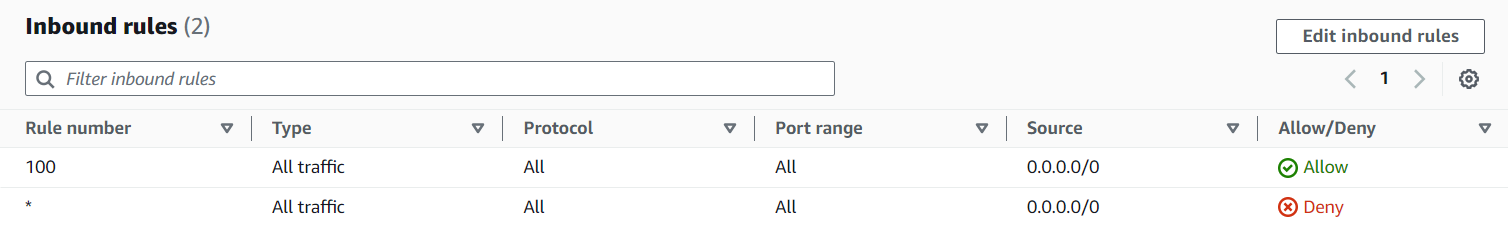
**A network ACL has inbound rules and outbound rules. Each rule can either allow or deny traffic. Each rule has a number from 1 to 32766. We evaluate the rules in order, starting with the lowest numbered rule, when deciding whether allow or deny traffic. If the traffic matches a rule, the rule is applied and we do not evaluate any additional rules.**

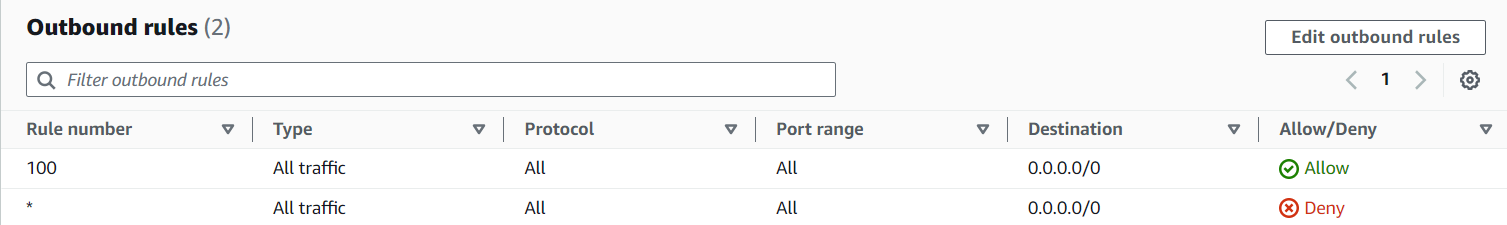
**NACL Configuration:**

* Navigate to VPC dashboard
* Left side – select NACL
* Here we can see a default NACL is already created for our VPC

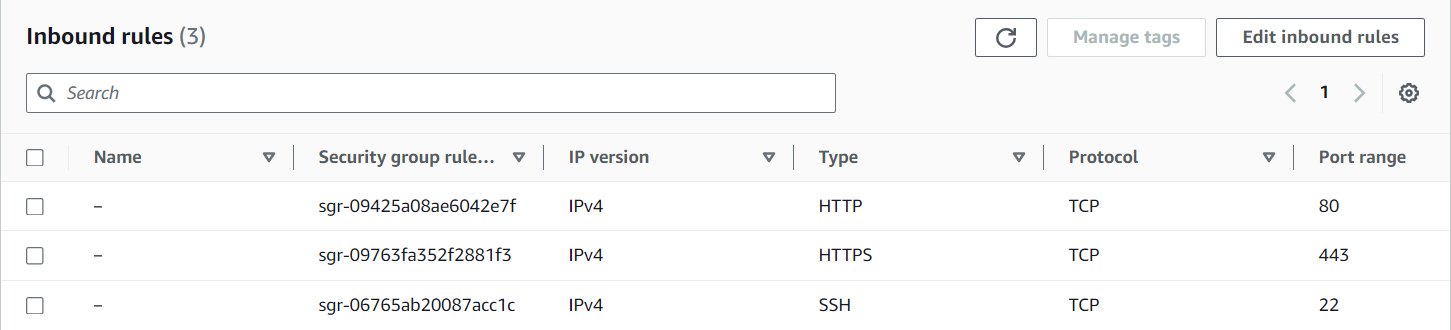
And associated with our two subnets.

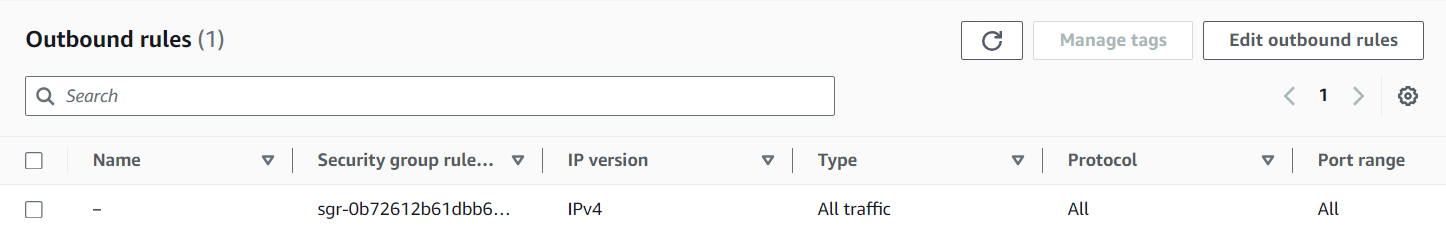
* Edit its name as **MyCloud-Default-NACL** for further identification.
* In default NACL all the Inboud and Outbound traffic are allowed.



****

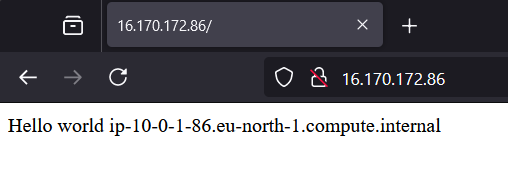
And the MyCloud-public-SG configured as below;





Then for practice NACL functionality NACL rules precedence, perform below steps;

* Run the public IP of the MyCloud-Bastion-EC2 instance on a web-server, it displays a web as below;

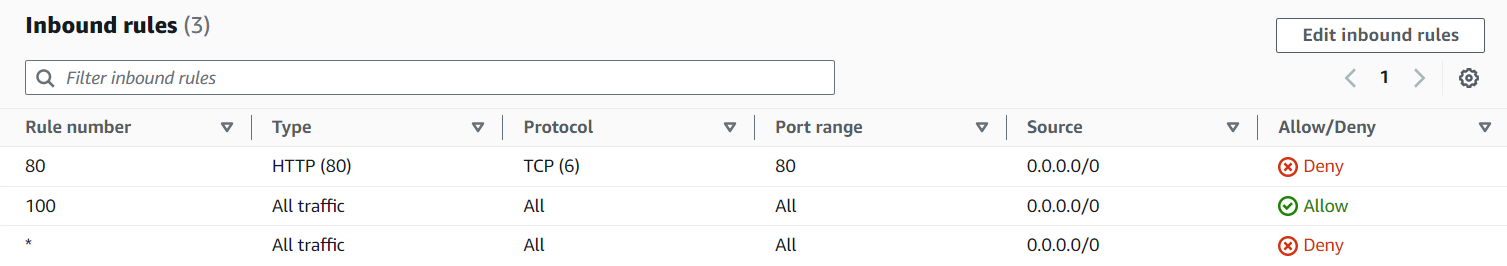


* Then we edit the NACL In-bound traffic;

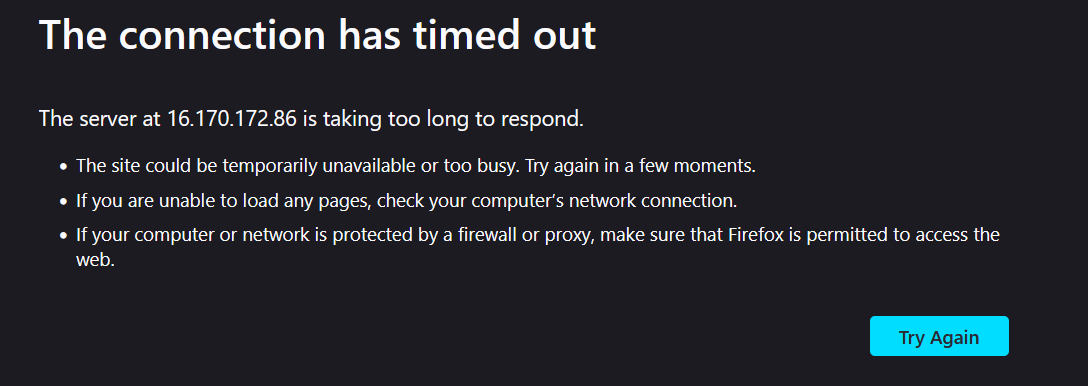
Rule number: 80

Port: HTTP – 80

Deny



Here the rule number 80 has higher precedence, so it denies the http request and display a timed-out request screen.



**If the Out-bound allows all-traffic, the NACL block the incoming http request shows the Stateless behaviour of NACL.**

**Also, the security group associated with the Bastion instance explicitly allows the inbound traffic of the http request, the NACL block the same from the subnet level.**

* Then we remove the http deny rule from NACL, and now the web page is accessible.
* For observing the Security Group Stateful behaviour, we have explicitly removed the all out-bound traffic and try to access the web page again – It will work.
* Because of the stateful behaviour the inbound allow http rule allows the corresponding outbound traffic.

**Bastion Host**

A bastion host, sometimes called a jump box, is a server that provides a single point of access from an external network to the resources located in a private network. A server exposed to an external public network, such as the internet, poses a potential security risk for unauthorized access. It’s important to secure and control access to these servers.

* Here we have already created an EC2 instance in public subnet and considered as Bastion Host – MyCloud-Bastion-EC2.
* As Its primary purpose is to provide a controlled entry point for accessing private instances located in private subnets.
* So here we connecting to our MyCloud-private-EC2 in private subnet from our MyCloud-Bastion-EC2.

Steps:

* Connect to MyCloud-Bastion-EC2 from our local windows terminal using below ssh command.

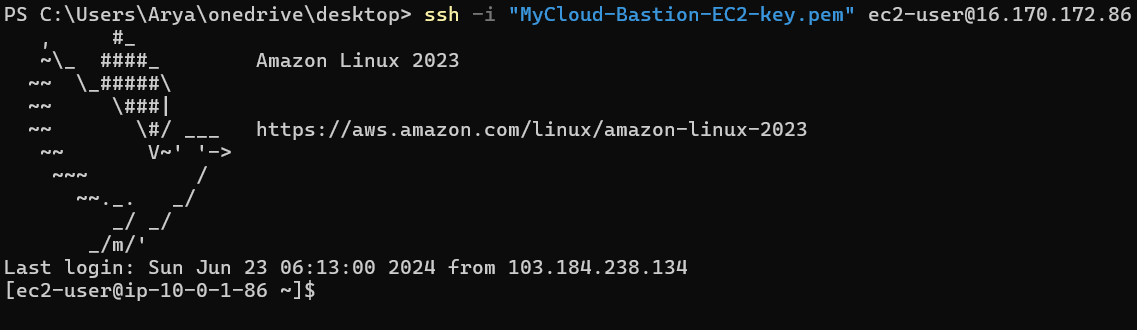
Run the below mentioned command from the correct directory which contains the key file.

***ssh -i MyCloud-Bastion-ec2-key ec2-user@ 16.170.172.86***

MyCloud-Bastion-ec2-key.pem: ssh private key for Bastion instance

16.170.172.86: public IP for Bastion Instance

Change the specific permissions (**remove the all-inherited permissions for the key file and provide full access for the logged in user**) for *MyCloud-Bastion-ec2-key for accessing the bastion host.*



Verify the private IP address of the Bastion instance is from public subnet.

* For connecting to the private instance;

First, we copy the private key from our windows local host to the MyCloud-Bastion-EC2 with a file name “MyCloud-private-ec2-key-pem”.

* Change the permission of the private key using below command

***chmod 400 MyCloud-private-ec2-key.pem***

the above command gives the read only access for the user.

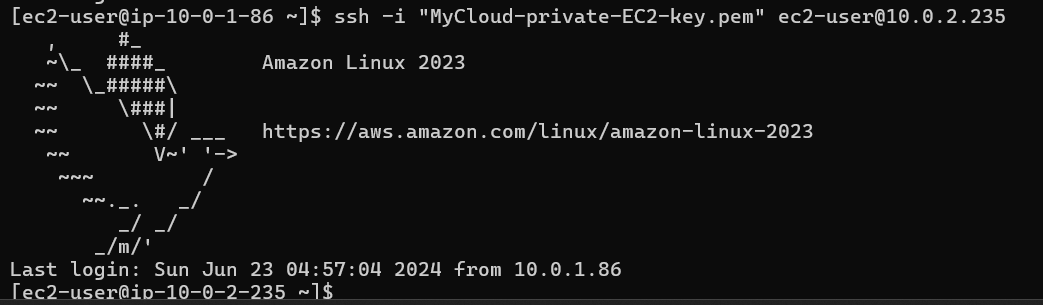
* Then we connect the private instance from Bastion host using below command.

***ssh -i MyCloud-private-ec2-key.pem ec2-user@ 10.0.2.235***

MyCloud-private-ec2-key.pem: ssh private key for private instance

10.0.2.235: private IP for private Instance.

Verify the private IP address of the Bastion instance is from public subnet.



**NAT Instance and NAT Gateway**

**Network Address Translation**

The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required. Network Address Translation (NAT) is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts.

**NAT Instance (Outdated)**

A NAT instance provides network address translation (NAT). You can use a NAT instance to allow resources in a private subnet to communicate with destinations outside the virtual private cloud (VPC), such as the internet or an on-premises network. The resources in the private subnet can initiate outbound IPv4 traffic to the internet, but they can't receive inbound traffic initiated on the internet.

Note:

* Launch NAT instance in public subnet
* **Must disable EC2 setting: Source/Destination Check**
* **Must attach an Elastic IP**
* Route table must be configured to route the traffic from

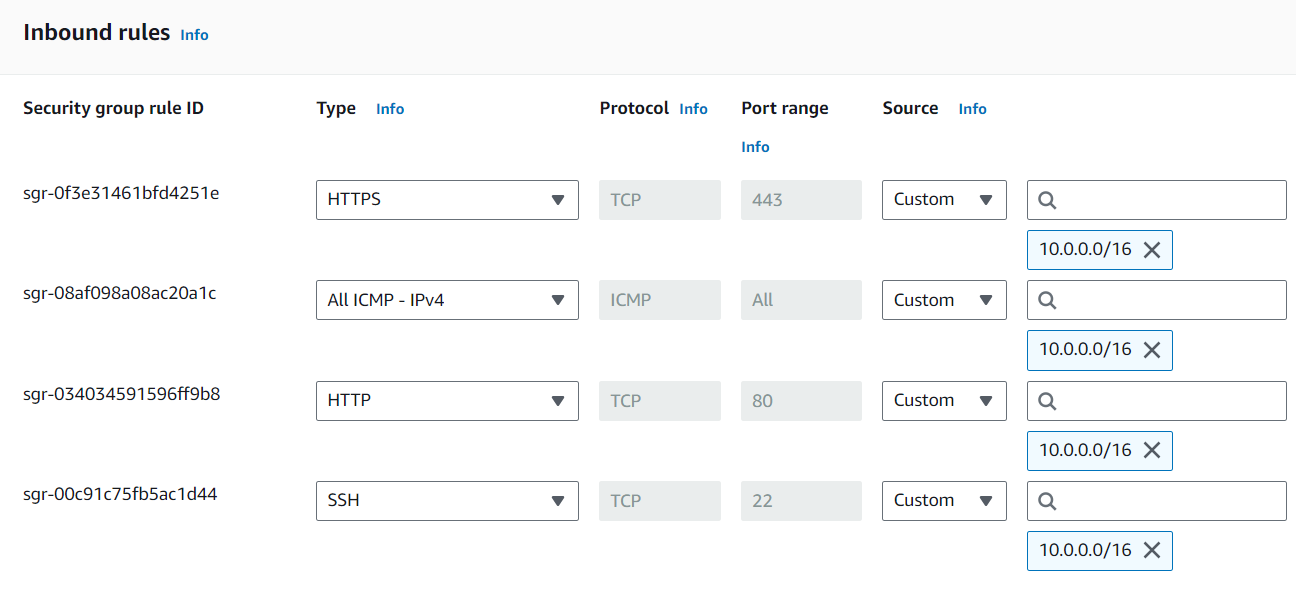
Private subnet to NAT Instance

**NAT Instance Configuration**

**Create a NAT Security Group**

Create a Security group “MyCloud-NAT-SG” with below mentioned In-bound

And allow on out-bound traffic.



**Launch a NAT Instance**

* Navigate to EC2 Dashboard
* choose Launch instance.
* For Name, enter – “MyCloud-NAT-Instance”
* For Application and OS Images, **select a NAT AMI** from Community AMI.
* Choose an instance type.
* Create a new key pair with name – “MyCloud-NAT-Instance-key”
* For network setting;

--choose VPC as – MyCloud

--select subnet – MyCloud-public-subnet-1a

--for firewall – select existing security group – “MyCloud-NAT-SG”

* Click on Launch Instance.

**Stop Source/Destination check**

* In the navigation pane, choose Instances.
* Select the NAT instance.
* Choose Actions, Networking, Change source/destination check.
* For Source/destination checking, select Stop.
* Choose Save.

**Edit private Route Table**

* Open the Amazon VPC Console
* In the navigation pane, choose Route tables.
* Select the MyCloud-private-RT
* On the Routes tab, choose Edit routes and then choose Add route.
* Enter 0.0.0.0/0 for Destination and the instance ID of the NAT instance (MyCloud-NAT-Instance) for Target.
* Choose Save changes.

Then we can connect from Bastion host to private Instance and ping to google.com using the ICMP port enabled in the NAT security group.

**NAT Gateway**

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances.

When you create a NAT gateway, you specify one of the following connectivity types:

* **Public** – (Default) Instances in private subnets can connect to the internet through a public NAT gateway. You create a public NAT gateway in a public subnet and must associate an elastic IP address with the NAT gateway at creation. You route traffic from the NAT gateway to the internet gateway for the VPC
* **Private** – Instances in private subnets can connect to other VPCs or you’re on-premises network through a private NAT gateway. You can route traffic from the NAT gateway through a transit gateway or a virtual private gateway. You cannot associate an elastic IP address with a private NAT gateway.

**NAT Gateway Configuration**

* Go to the VPC Dashboard
* In the left-hand navigation pane, click on NAT Gateways.
* Click on Create NAT Gateway.
* Subnet: MyCloud-public-subnet-1a
* Elastic IP Allocation ID: Allocate a new Elastic IP address or select an existing one.
* Click Create a NAT Gateway.
* Navigate to Route table VPC dashboard- Select MyCloud-private-RT
* Add a new route:
* Destination: 0.0.0.0/0
* Target: Select the NAT Gateway ID created earlier.
* Save the changes.

For test the NAT gateway; stop the NAT instance and remove the NAT instance route from the route table.

And then we can connect from Bastion host to private Instance and ping to google.com or install the httpd (Apache web server) in private instance.

--End--