Professional Skills

Agile Fundamentals

Jira

Git

Databases Introduction

Java Beginner

Maven

Testing (Foundation)

Java Intermediate

HTML

CSS

Javascript

Spring Boot

Selenium

Sonarqube

Advanced Testing (Theory)

Cucumber

MongoDB

Express

NodeJS

React

Express-Testing

Networking

Security

Cloud Fundamentals

AWS Foundations

AWS Intermediate

○ Virtual Private Cloud (VPC)

○ EC2 VPC Security Groups

○ EC2 VPC Subnets

AWS Network Address Translation (NAT) Gateway

Contents

Overview

You can use a **N**etwork **A**ddress **T**ranslation (NAT) Gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

You will be charged for creating and using a NAT Gateway hourly. More about the cost can be found here

NAT Gateway Basics

To use a NAT Gateway, you would need to specify a public subnet for it to reside. You must also define an **Elastic IP** when creating your NAT Gateway. The Elastic IP address cannot be changed once you associate it with the NAT Gateway. After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point Internet-bound traffic to the NAT gateway.
This enables instances in your private subnets to communicate with the internet.

NAT Gateway Rules and Limitations

- A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps. If you require more, you can distribute the workload by splitting your resources into multiple subnets, and creating a NAT gateway in each subnet.

- You can associate exactly one Elastic IP address with a NAT gateway. You cannot disassociate an Elastic IP address from a NAT gateway after it's created. To use a different Elastic IP address for your NAT gateway, you must create a new NAT gateway with the required address, update your route tables, and then delete the existing NAT gateway if it's no longer required.

- A NAT gateway supports the following protocols:
  - TCP
  - UDP
  - ICMP
- A NAT gateway can support up to 55,000 simultaneous connections to each unique destination. This limit also applies if you create approximately 900 connections per second to a single destination (about 55,000 connections per minute). If the destination IP address, the destination port, or the protocol (TCP/UDP/ICMP) changes, you can create

- EC2 VPC Internet Gateways
- AWS Route Tables
- AWS Network Address Translation (NAT) Gateway
- AWS Network Access Control Lists (NACLs) CLI
- AWS Java SDK
- AWS DynamoDB
- AWS Lambda Functions
- AWS API Gateway
- SQS Introduction
- AWS Serverless CRUD Solution
- AWS Serverless Solution with DynamoDB
- CloudWatch CLI
- CloudTrail

Linux

DevOps

Jenkins Introduction

Jenkins Pipeline

Markdown

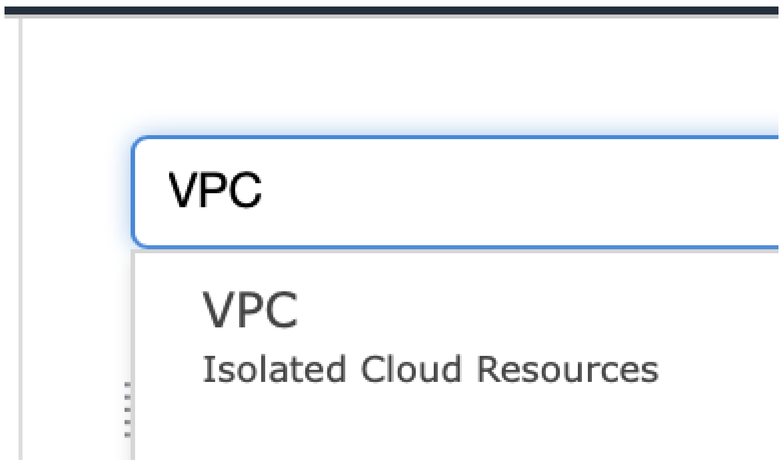
IDE Cheatsheet

an additional 55,000 connections. For more than 55,000 connections, there is an increased chance of connection errors due to port allocation errors.

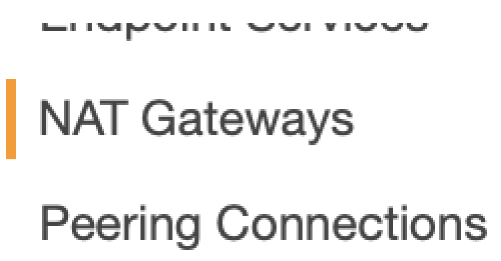
Note: For this tutorial, you will need to create an EC2 Instance in a private subnet, so do not assign this EC2 Instance an IP Address on creation.
You will also need to create a route table which is associated with the subnet your EC2 instance is in.

Tutorial

1. Navigate to the AWS Console and sign in here
2. Search for VPC under the services drop-down menu, and click on VPC

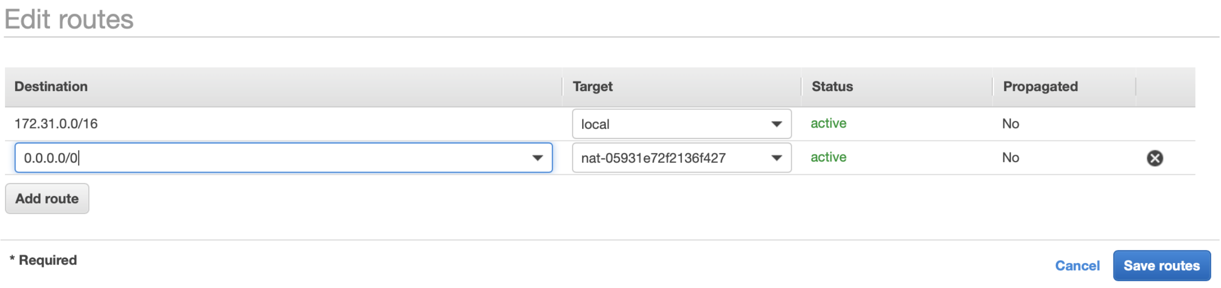


3. On the left pane, under the Virtual Private Cloud section, click **NAT Gateways**



4. Click on the **Create NAT Gateway** button and you will need to define a subnet and an Elastic IP address for your NAT Gateway as discussed earlier.
   Select any of the options for each part and then click **Create NAT Gateway**
5. If this was created successfully, then you would see a green banner and prompting you to **Edit Route Table**.
   Click on **Edit Route Tables** button.
6. You will see the homepage for all your **Route Tables**, select the route table which is associated with the subnet your EC2 Instance resides.
7. Go to the **Routes** tabs and click on **Edit routes**. You then need to add the route which your subnet is allowed to connect to.
8. The configuration should be as follows:

- Destination: 0.0.0.0/0
- Target: nat-****, where you select the NAT Gateway and its ID


Edit routes

Destination	Target	Status	Propagated
172.31.0.0/16	local	active	No
0.0.0.0/0	nat-05931e72f2136f427	active	No

9. Click **Save** and the NAT Gateway has been added to your private Subnet

Test

Because your EC2 Instance is in private subnet and does not have a public IP Address, you would need to create another EC2 Instance within the same VPC and put it in a public subnet or at least give it a public IP address.
From there you would need to SSH into the EC2 Instance in the public subnet and then through that EC2 Instance SSH into the EC2 Instance in the private subnet.

To SSH into your EC2 Instance inside the private subnet, you would need to send over your private key to your EC2 Instance in your public subnet. So then you can use the key to SSH into your EC2 Instance in your private subnet.

Command:

```

scp -i <pem key> <pem key> <username>@<ip address>:/home/<username>
```

This will copy and paste your private pem key to the EC2 Instance in the public subnet. This will then allow you to SSH into the EC2 Instance in your private subnet.

You should be able to run the following command:

```
sudo apt update -y
```

Removing the NAT Gateway will prevent you from running the above command as you will lose access to the internet.

Resources

The follwing resources were used:

- EC2 Instance
- VPC
- Route Table
- NAT Gateway

Exercises

There are no exercises for this module.