ASSESSMENT ON VPC

Q1. When to use Elastic IP over Public IP

ANS: Public IP addresses are dynamic - i.e. if you stop/start your instance you get reassigned a new public IP.

Elastic IPs get allocated to your account, and stay the same - it's up to you to attach them to an instance or not. You could say they are static public IP addresses.

Elastic IP address is a public **static IPv4 address** which is reachable from the Internet. Basically Elastic IP addresses are used by AWS to manage its dynamic cloud computing services. Within the AWS infrastructure, customers have virtual private clouds (VPC), within the VPCs, users have instances. So when you launch an EC2 instance, you receive a Public IP address by which that instance is reachable from internet. Once you stop that instance and restart the instance you get a new Public IP for the same instance. So it's basically a problem to connect your instance from internet for not having a static IP. To overcome this problem, *we attach an Elastic IP to an Instance which doesn't change after you stop / start the instance*.

Q2. Valid IP Ranges for LAN, Implication of using Public IP ranges for Private Network.

ANS:

* Class A : 10.0.0.0 – 10.0.255.255

Class B : 172.16.0.0 – 172.13.255.255

Class C : 192.168.0.0 – 192.168.255.255

If you ever expect to connect these systems to an Internet-facing router, though, then you could experience the following issues if you don't stick with private IP ranges:

- Traffic destined for another host may leak out on to the Internet.
- You might want to get to the IANA-assigned host on that IP and may not be able to do it if it's an internal host.

- If you aren't the only one mantaining this network, you could horribly confuse someone who is doing troubleshooting.

Q3. List down the things to keep in mind while VPC peering.

ANS:

To create a VPC peering connection with another VPC, be aware of the following limitations and rules:

- You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks. Amazon always assigns your VPC a unique IPv6 CIDR block. If your IPv6 CIDR blocks are unique but your IPv4 blocks are not, you cannot create the peering connection.
- You have a quota on the number of active and pending VPC peering connections that you can have per VPC. For more information, see Amazon VPC Quotas in the *Amazon VPC User Guide*
- VPC peering does not support transitive peering relationships. In a VPC peering connection, your VPC does not have access to any other VPCs with which the peer VPC may be peered. This includes VPC peering connections that are established entirely within your own AWS account. For more information about unsupported peering relationships, see Unsupported VPC Peering Configurations. For examples of supported peering relationships, see VPC Peering Scenarios.
- You cannot have more than one VPC peering connection between the same two VPCs at the same time.
- Unicast reverse path forwarding in VPC peering connections is not supported. For more information, see Routing for Response Traffic.
- Any tags that you create for your VPC peering connection are only applied in the account or region in which you create them.

- If the IPv4 CIDR block of a VPC in a VPC peering connection falls outside of the private IPv4 address ranges specified by RFC 1918, private DNS hostnames for that VPC cannot be resolved to private IP addresses. To resolve private DNS hostnames to private IP addresses, you can enable DNS resolution support for the VPC peering connection. For more information, see Enabling DNS Resolution Support for a VPC Peering Connection.
- You cannot connect to or query the Amazon DNS server in a peer VPC.

An inter-region VPC peering connection has additional limitations:

- You cannot create a security group rule that references a peer VPC security group.
- You cannot enable support for an EC2-Classic instance that's linked to a VPC via ClassicLink to communicate with the peer VPC.
- The Maximum Transmission Unit (MTU) across the VPC peering connection is 1500 bytes (jumbo frames are not supported).
- You must enable DNS resolution support for the VPC peering connection to resolve private DNS hostnames of the peered VPC to private IP addresses, even if the IPv4 CIDR for the VPC falls into the private IPv4 address ranges specified by RFC 1918.
- Inter-region peering in China is only allowed between the China (Beijing) Region, operated by SINNET and the China (Ningxia) Region, operated by NWCD.

Q4. CIDR of a VPC is 10.0.0.0/16, if the subnet mask is /20 calculate the number of subnets that could be created from the VPC. Also find the number of IP in subnet.

ANS:

* 10.0.0.0/16: 00001010.00000000.00000000.00000000 (In /16, first 2 octets are fixed).
00001010.00000000.00000000.00000000 (In /20, extra 4 bits are borrowed from hosts)
* These extra 4 bits are subnetting bits.

So, total number of subnets = 2^4 (16)

And, total IP'S in each subnet = 2^12 (4096)

Q5. Differentiate between NACL and Security Groups.

ANS :

Security Group	NACL (Network Access Control List)
It supports only allow rules, and by default, all the rules are denied. You cannot deny the rule for establishing a connection.	**It supports both allow and deny rules, and by default, all the rules are denied. You need to add the rule which you can either allow or deny it.**
It is a stateful means that any changes made in the inbound rule will be automatically reflected in the outbound rule. For example, If you are allowing an incoming port 80, then you also have to add the outbound rule explicitly.	**It is a stateless means that any changes made in the inbound rule will not reflect the outbound rule, i.e., you need to add the outbound rule separately. For example, if you add an inbound rule port number 80, then you also have to explicitly add the outbound rule.**
It is associated with an EC2 instance.	**It is associated with a subnet.**

All the rules are evaluated before deciding whether to allow the traffic.	**Rules are evaluated in order, starting from the lowest number.**
Security Group is applied to an instance only when you specify a security group while launching an instance.	**NACL has applied automatically to all the instances which are associated with an instance.**
It is the first layer of defense.	**It is the second layer of defense.**

Q6. Implement a 2-tier vpc with following requirements:

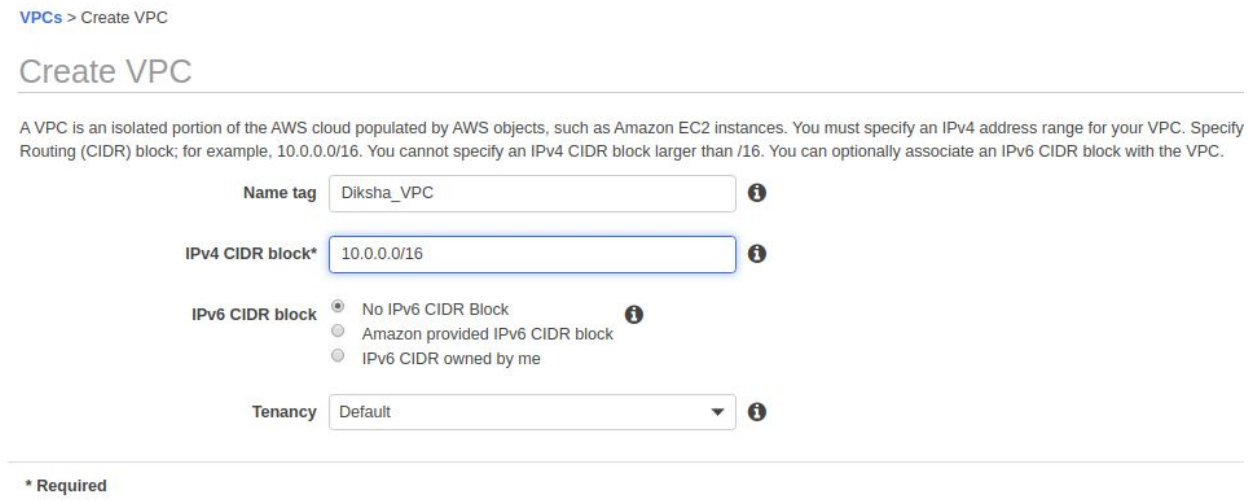
1. Create a private subnet, attach NAT, and host an application server(Tomcat)

2. Create a public subnet, and host a web server(Nginx), also proxypass to Tomcat from Nginx

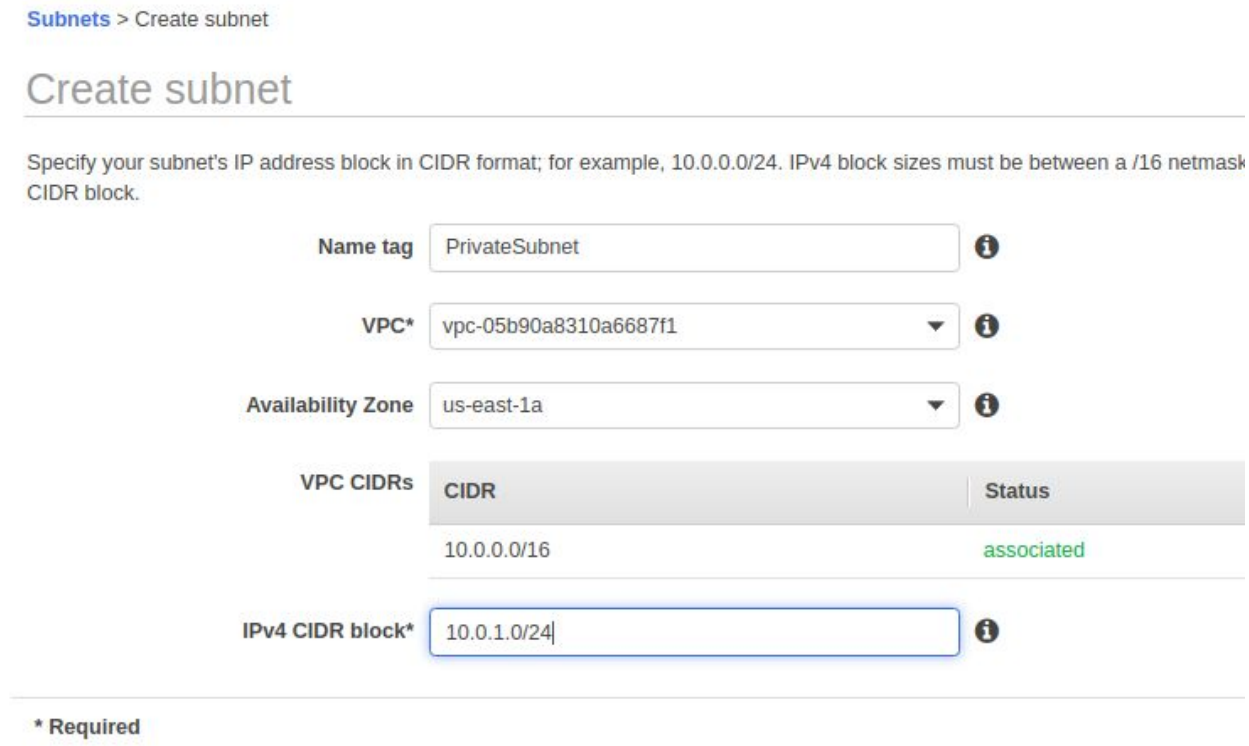
After Implementing this on AWS, create an architecture diagram for this use case.

Note: For hosting Nginx in public subnet, use Elastic IP.

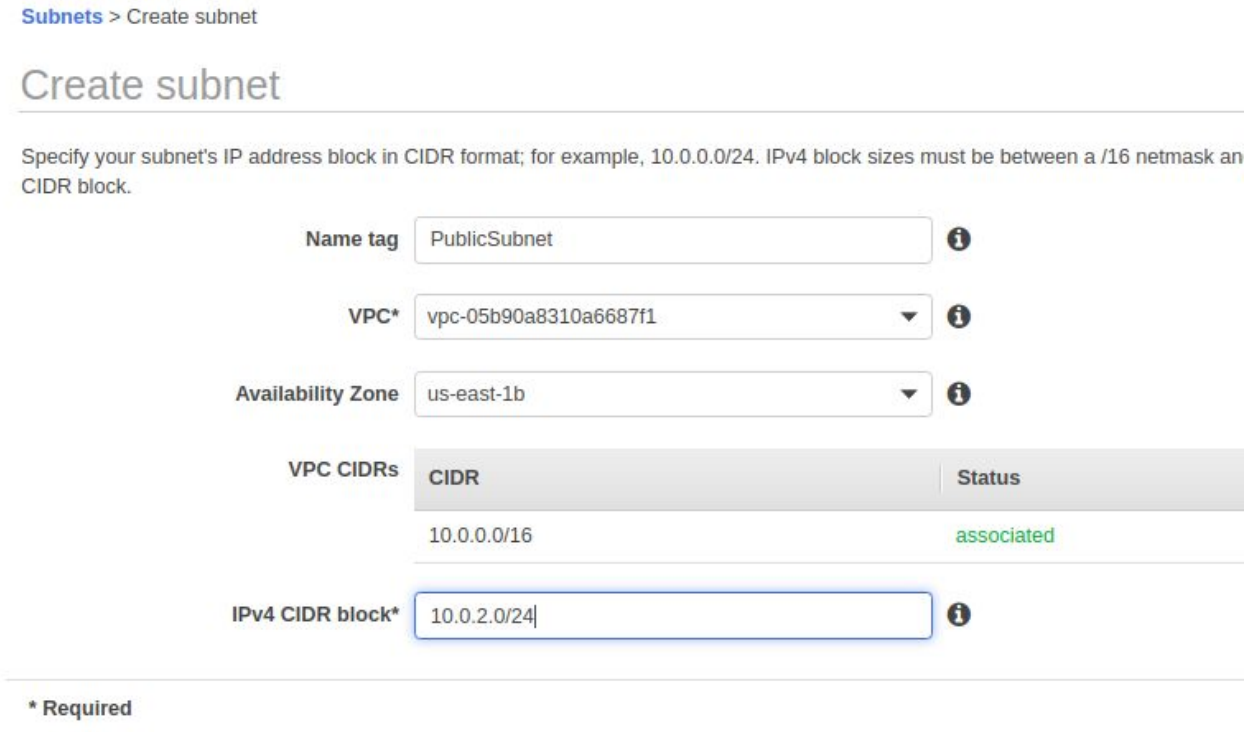
STEP 1: Creating VPC



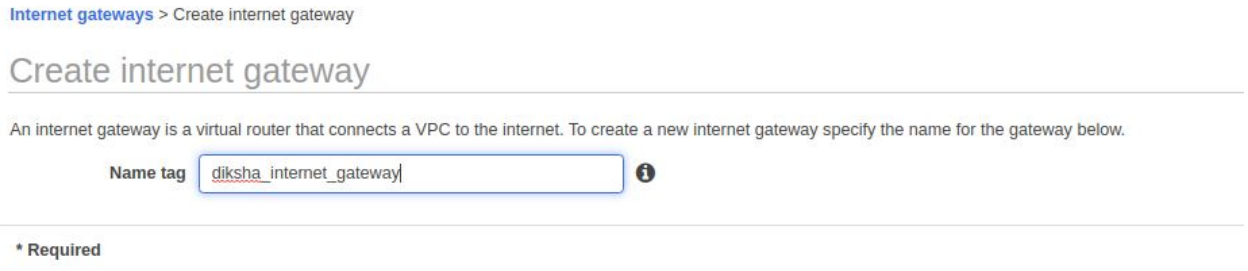
STEP 2: Creating subnet (Private)

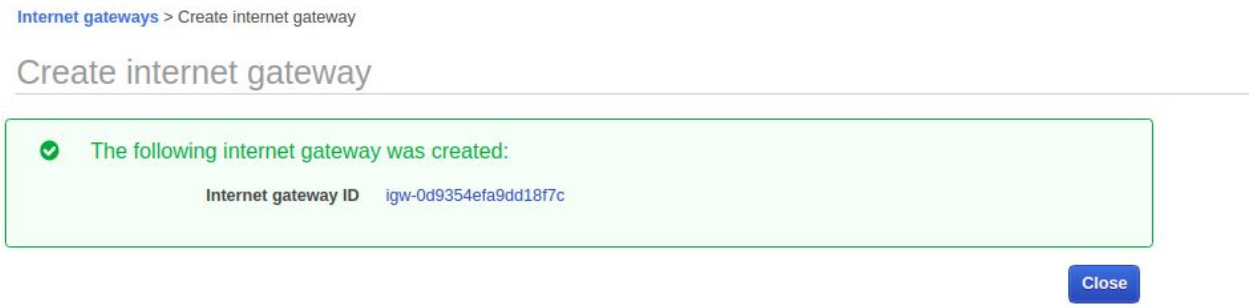


STEP 3: Creating subnet (Public)

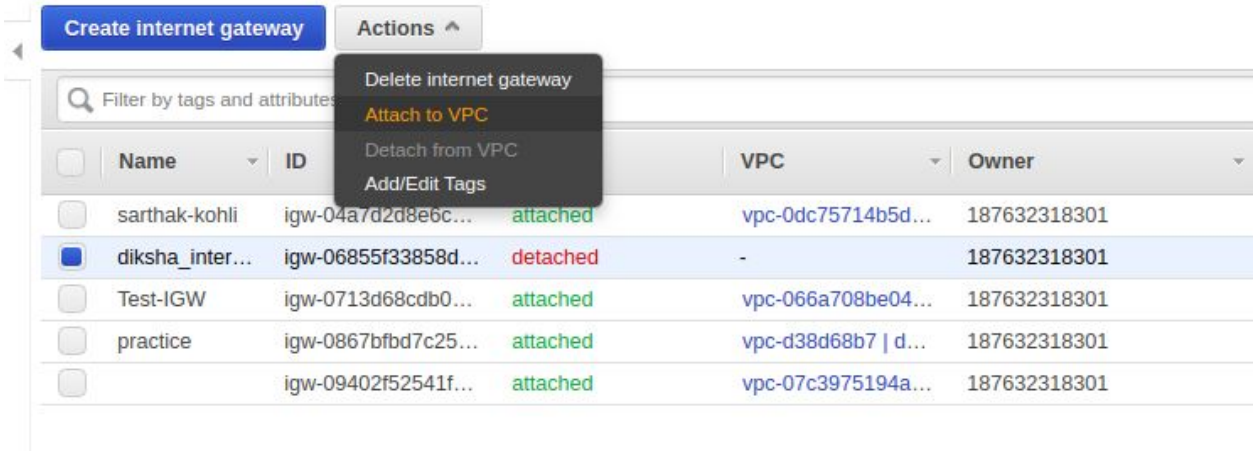


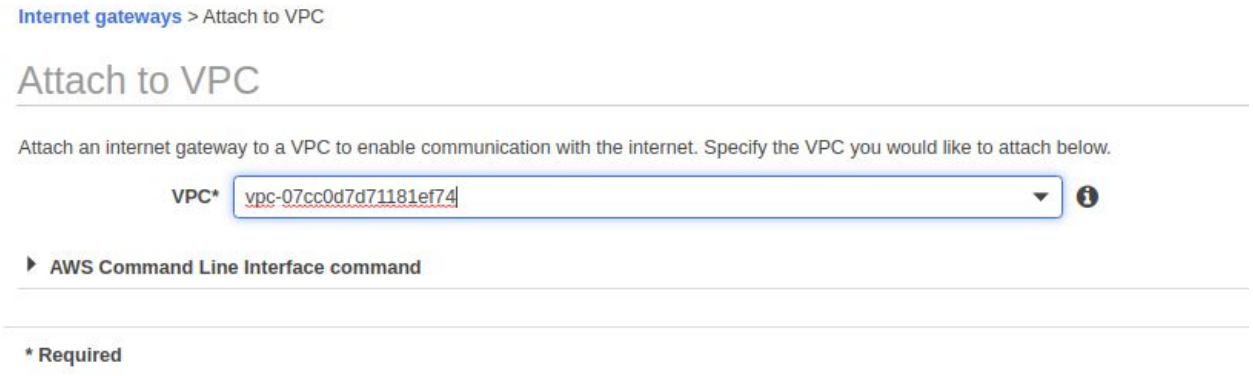
STEP 4: Creating Internet Gateway



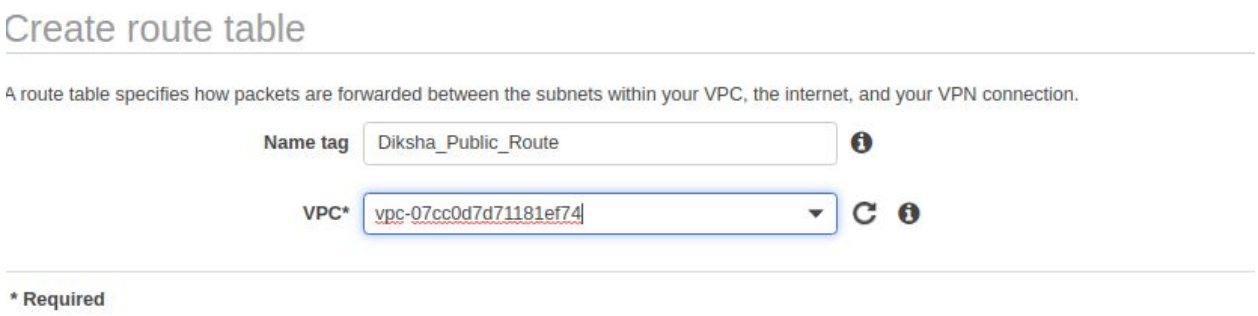


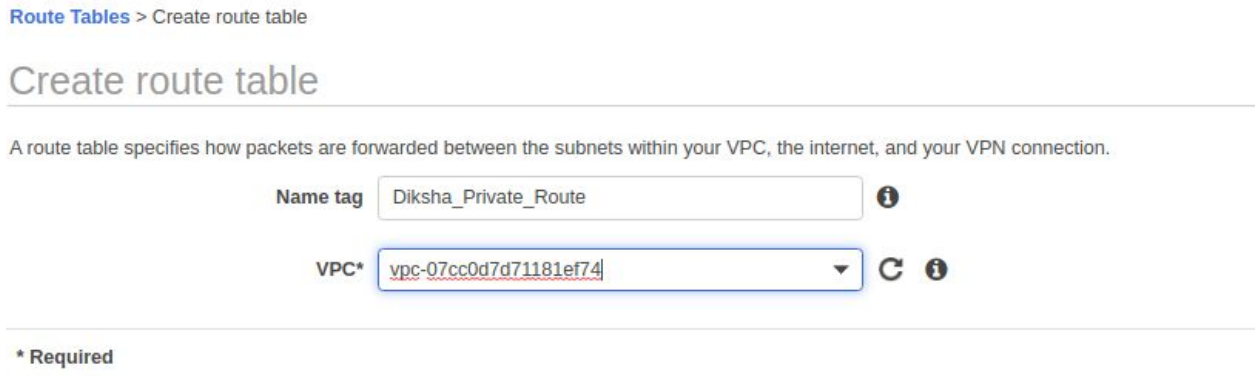
STEP 5: Attaching this Internet Gateway to the VPC.



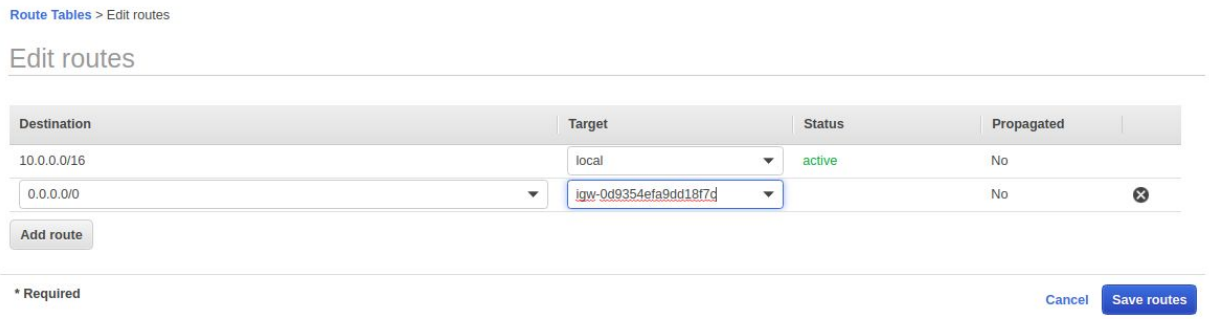


STEP 6: Creating route table for public subnet.

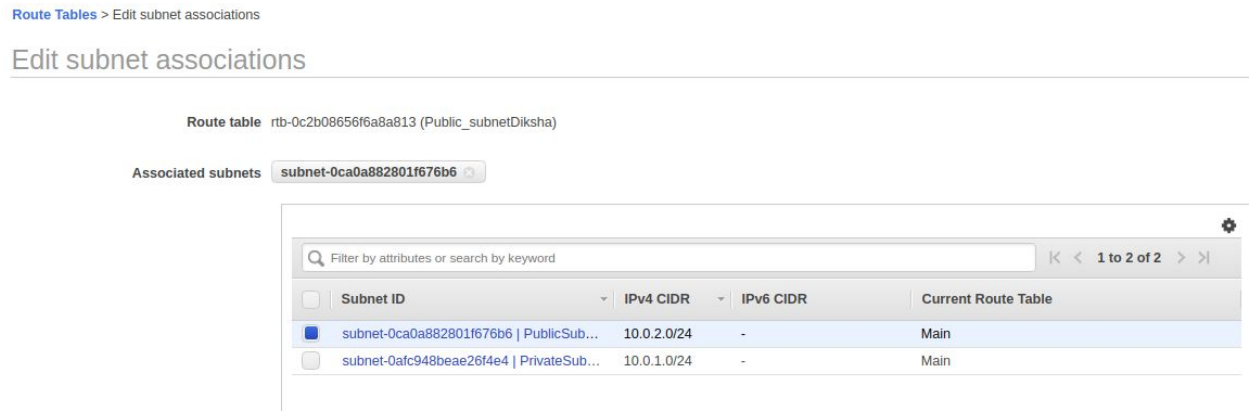




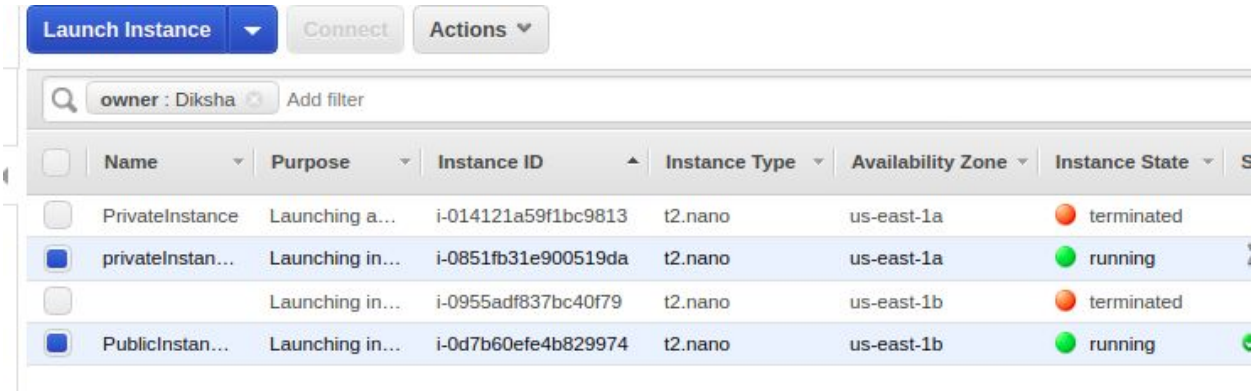
STEP 7: Editing the routes (for making publically accessible)



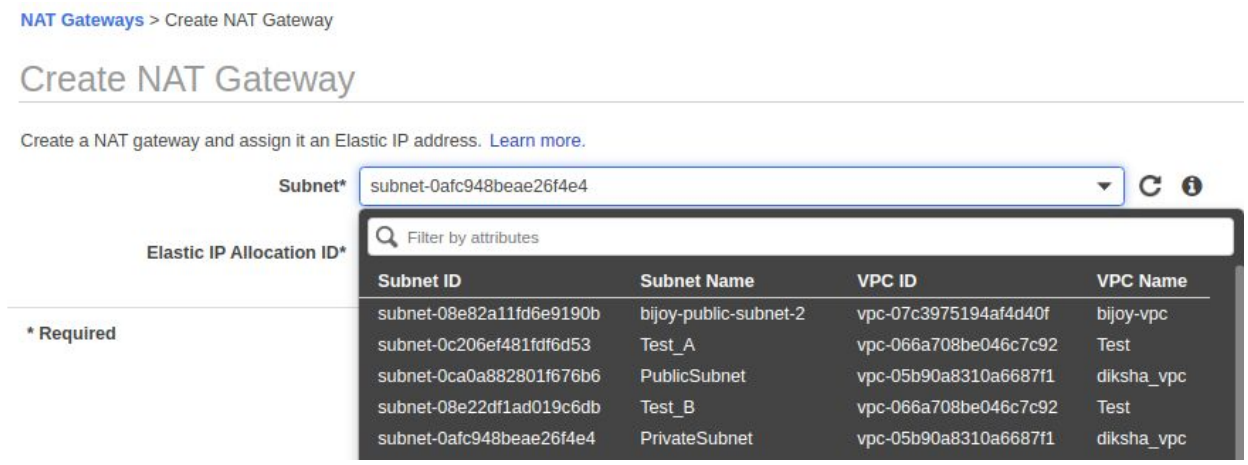
STEP 8: Associate public subnet with the public route table.



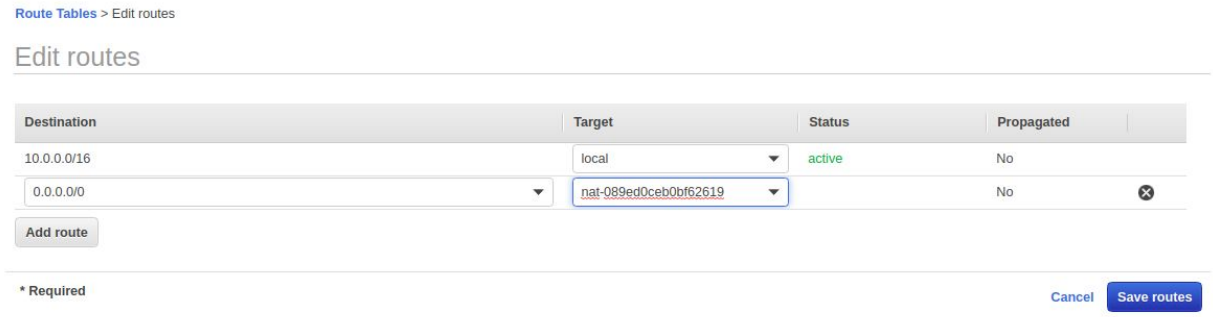
STEP 9: Launching an instance in private and other in public subnets.



STEP 10: Now making NAT gateway

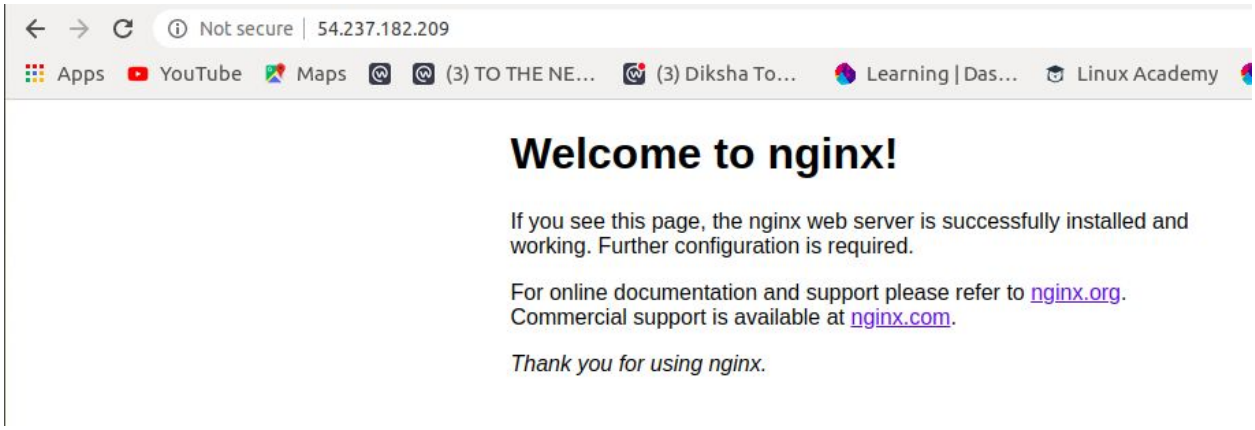


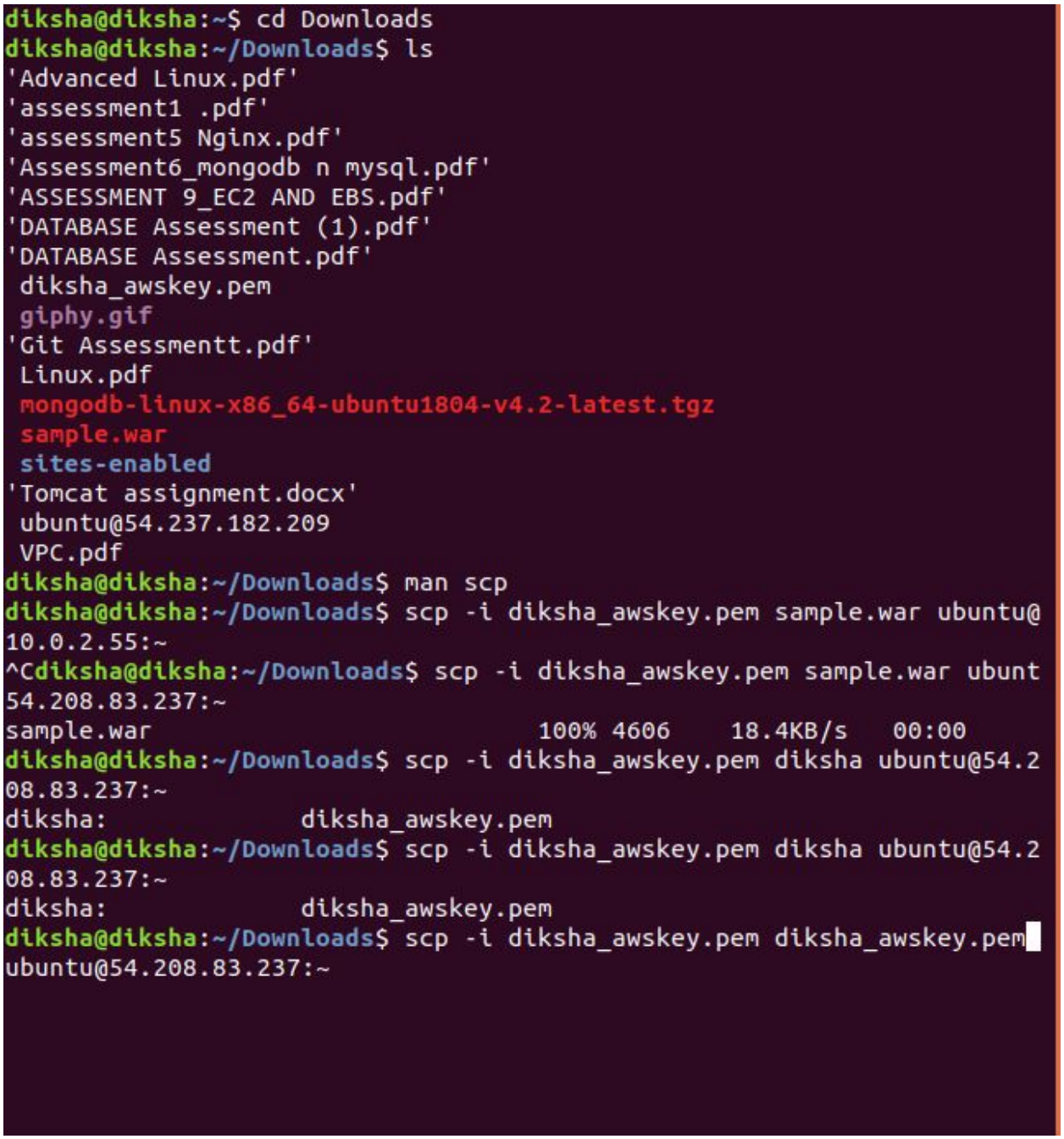
STEP 11: Now edit routes in my private subnet's route table.



STEP 12: Installing TOMCAT in private subnet's instance.

STEP 13: Installing Nginx in public subnet’s instance





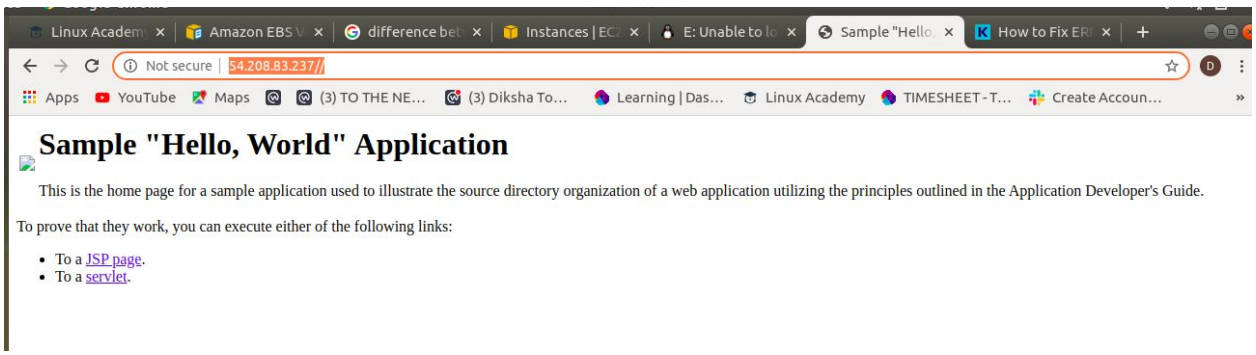
STEP 14: Now proxypassing tomcat through nginx.

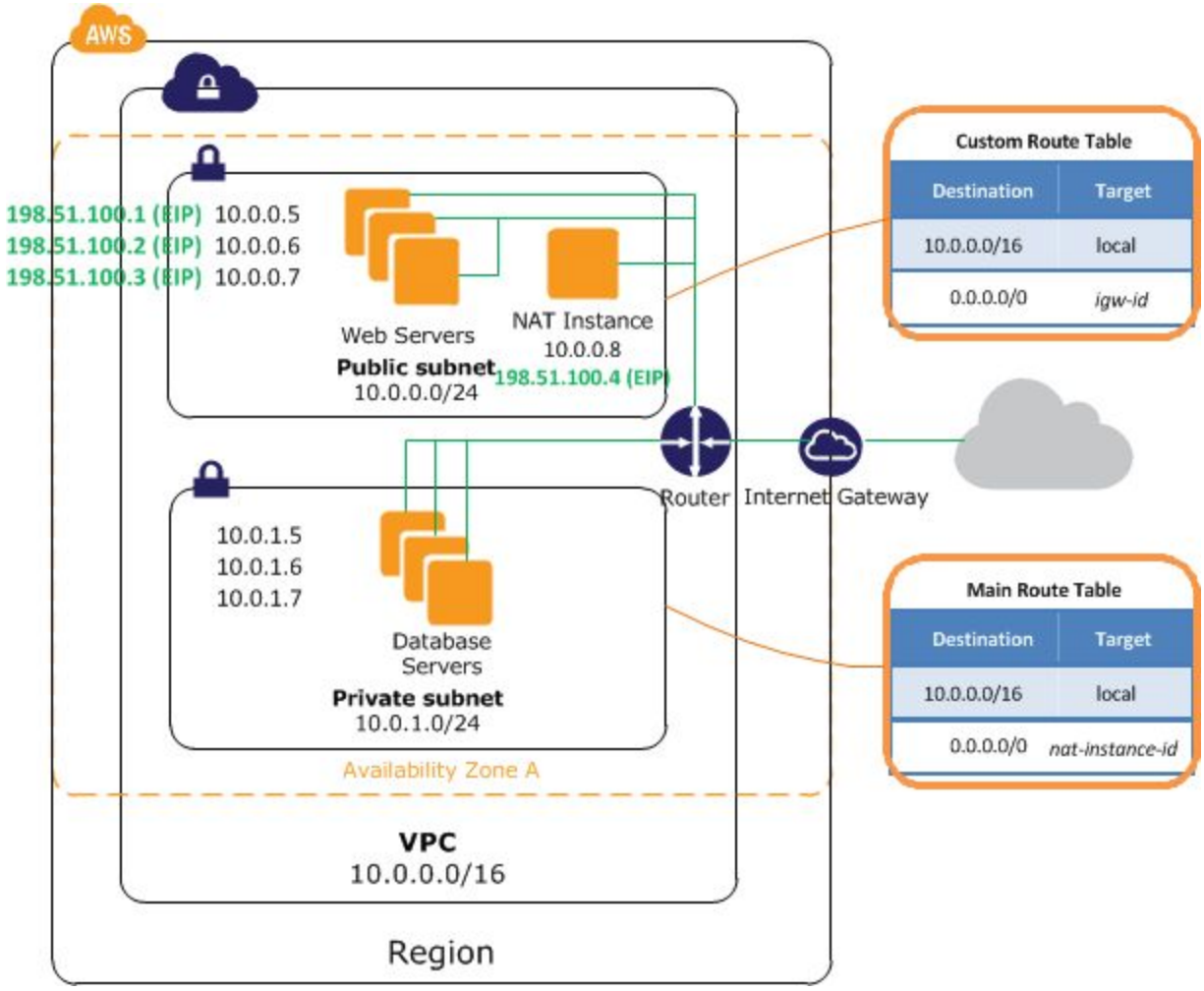
```
# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        proxy_pass http://10.0.2.205:8080/sample/;
        try_files $uri $uri/ =404;
}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
```

STEP 15: Checking the status using curl

```
Processing triggers for ureadahead (0.100.0-21) ...
root@ip-10-0-2-205:~# curl localhost
curl: (7) Failed to connect to localhost port 80: Connection refused
root@ip-10-0-2-205:~# curl localhost:8080
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Apache Tomcat</title>
</head>

<body>
<h1>It works !</h1>

<p>If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!</p>

<p>This is the default Tomcat home page. It can be found on the local filesystem at: <code>/var/lib/tomcat9/webapps/ROOT/index.html</code></p>

<p>Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with <code>CATALINA_HOME</code> in <code>/usr/sh
are/tomcat9</code> and <code>CATALINA_BASE</code> in <code>/var/lib/tomcat9</code>, following the rules from <code>/usr/share/doc/tomcat9-comm
on/RUNNING.txt.gz</code>.</p>

<p>You might consider installing the following packages, if you haven't already done so:</p>

<p><b>tomcat9-docs</b>: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you
can access it by clicking <a href="docs/">here</a>.</p>

<p><b>tomcat9-examples</b>: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installe
d, you can access it by clicking <a href="examples/">here</a>.</p>

<p><b>tomcat9-admin</b>: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can acces
s the <a href="manager/html">manager webapp</a> and the <a href="host-manager/html">host-manager webapp</a>.</p>

<p>NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted
to users with role "admin-gui". Users are defined in <code>/etc/tomcat9/tomcat-users.xml</code>.</p>

</body>
```

STEP 16: In Browser.




AWS
198.51.100.1 (EIP) 10.0.0.5
198.51.100.2 (EIP) 10.0.0.6
198.51.100.3 (EIP) 10.0.0.7
Web Servers
NAT Instance
10.0.0.8
198.51.100.4 (EIP)
Public subnet
10.0.0.0/24
Router
Internet Gateway
10.0.1.5
10.0.1.6
10.0.1.7
Database Servers
Private subnet
10.0.1.0/24
Availability Zone A
VPC
10.0.0.0/16
Region
Custom Route Table
Destination
Target
10.0.0.0/16
local
0.0.0.0/0
igw-id
Main Route Table
Destination
Target
10.0.0.0/16
local
0.0.0.0/0
nat-instance-id