AWS-VPC

Ques 1. When to use Elastic IP over Public IP

Ans 1.

- Use case:

Elastic IP is used when you are working on a long time project and configuration of IP sometimes consumes more time.

Public IP is used when you are working on small projects and running 2-3 servers. Here in this situation you make use of IP for a short time.

- Do remember one thing if you have elastic IP in your account and it's not in use,then you will be charged for it.
- Elastic IP addresses are used by AWS to manage its dynamic cloud computing services. Within the AWS infrastructure, customers have virtual Private cloudWithin the VPCs, users have instances. The Elastic IP address is what is used to advertise the data within the instance to the public internet.

Ques 2. Valid IP Ranges for LAN, Implication of using Public IP ranges for Private Network.

Ans 2.

192.168.0.0 - 192.168.255.255 (65,536 IP addresses)

172.16.0.0 - 172.31.255.255 (1,048,576 IP addresses)

10.0.0.0 - 10.255.255.255 (16,777,216 IP addresses)

Ques 3. List down the things to keep in mind while VPC peering.

Ans 3.

1. Choosing the proper VPC configuration for your organization's needs

2. Choosing a CIDR block for your VPC implementation

3. Isolating your VPC environments

4. Best practices for securing your AWS VPC implementation

5. Creating your disaster recovery plan

6. Traffic control and security

7. Keep your data close

8 .Determining the NAT instance type

9. ELB on Amazon VPC

Ques 5. Differentiate between NACL and Security Groups.

Ans 5.

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.

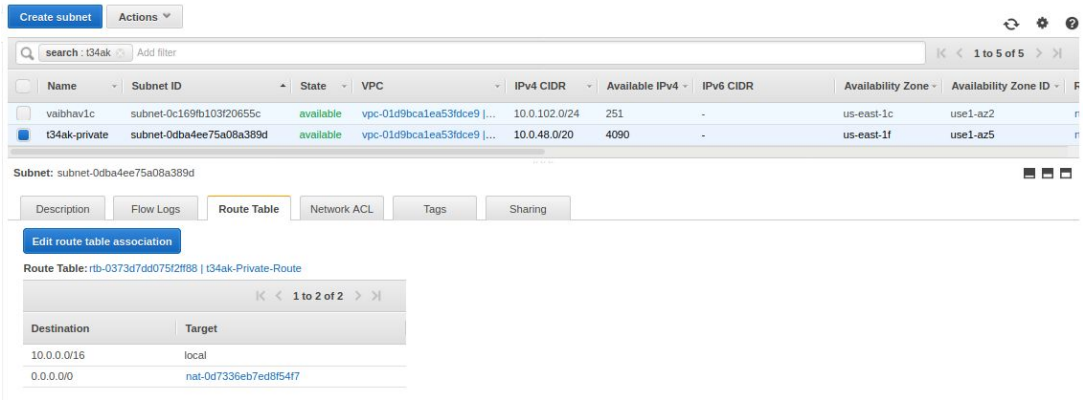
Ques 6.Implement a 2-tier vpc with following requirements:

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

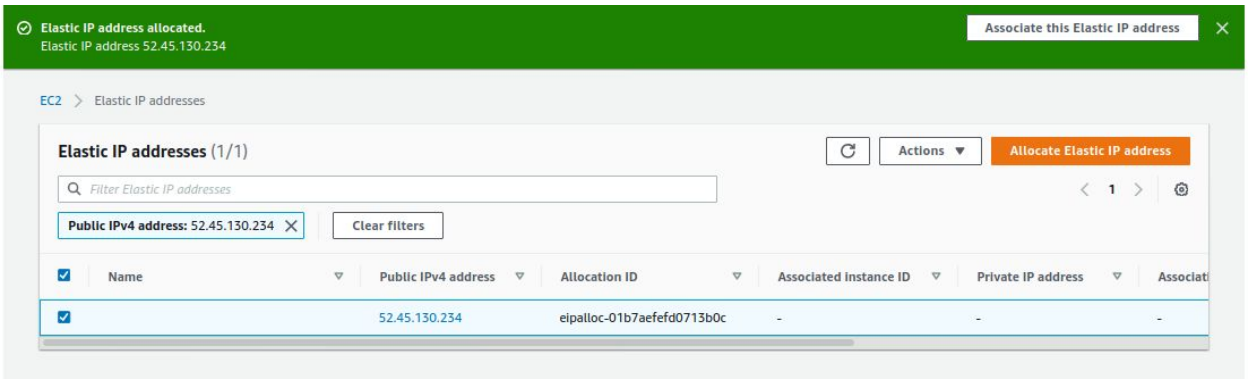
After Implementing this on AWS, create an architecture diagram for this use case.

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

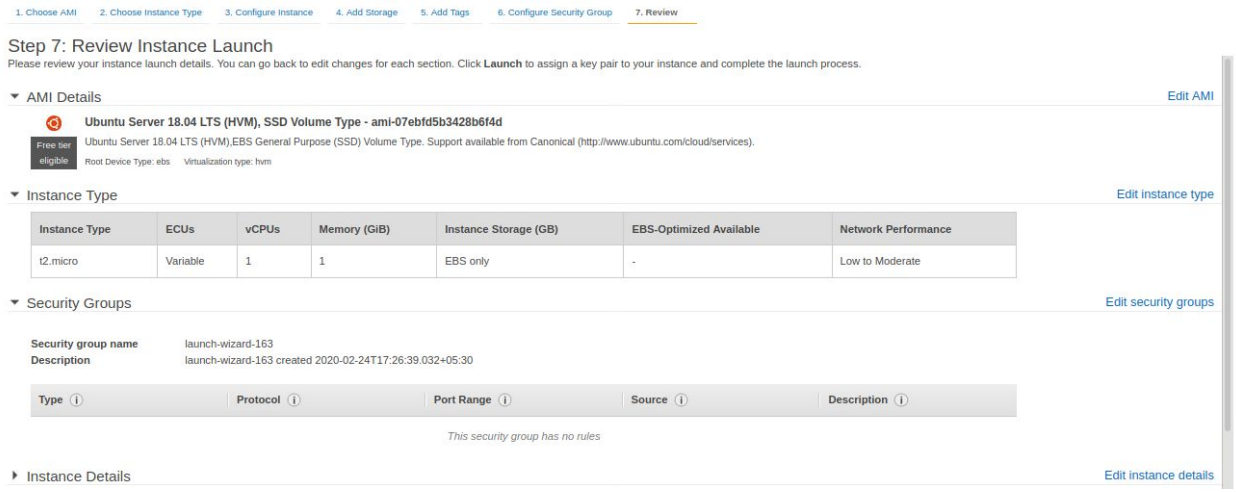
Ans 6. First we have to make a VPC then create two subnets one is for private and other is for public
When ever we are creating the NAT we need the Elsastic IP



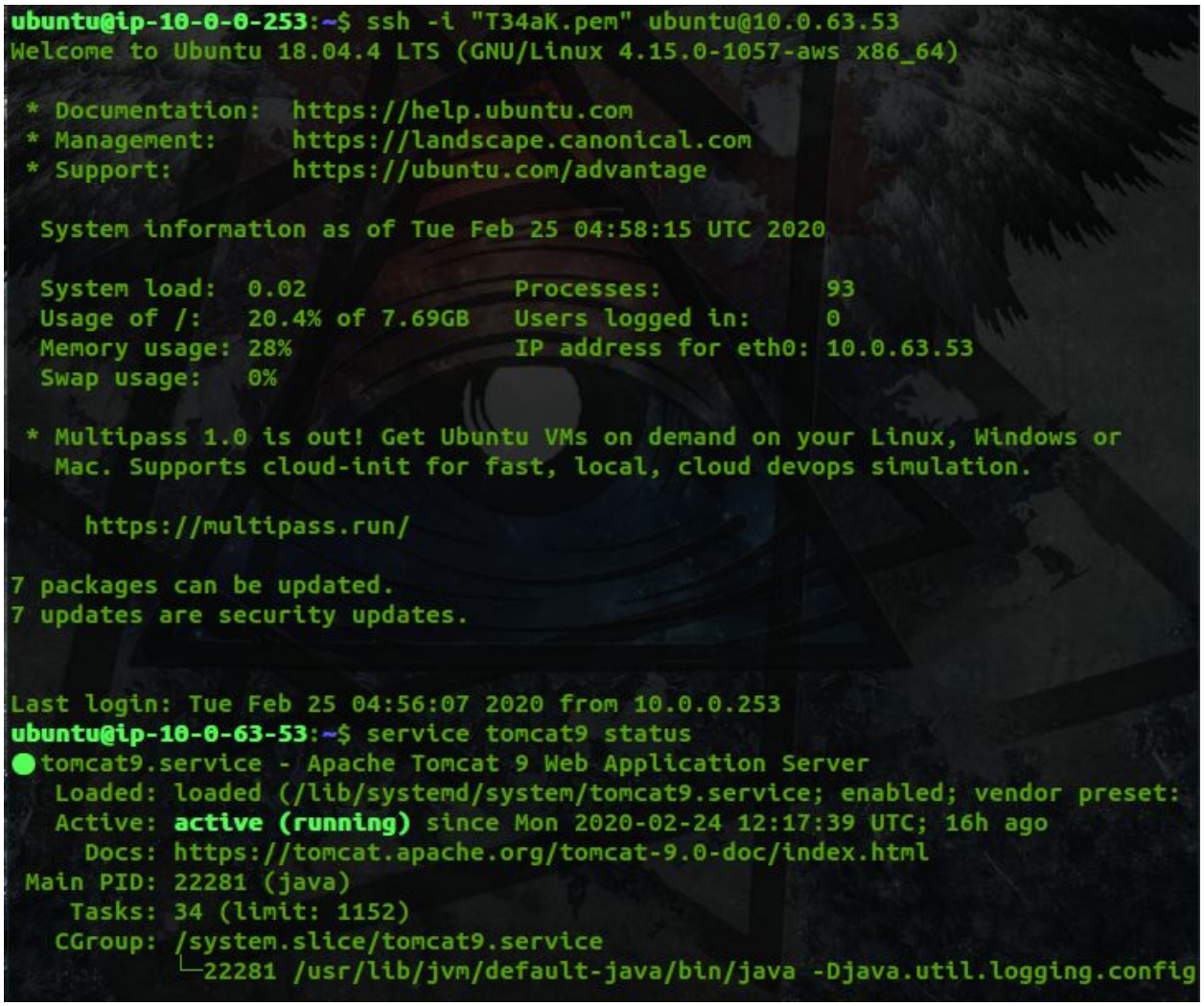
Creating NAT with Elastic IP



Now we have to launch the instance

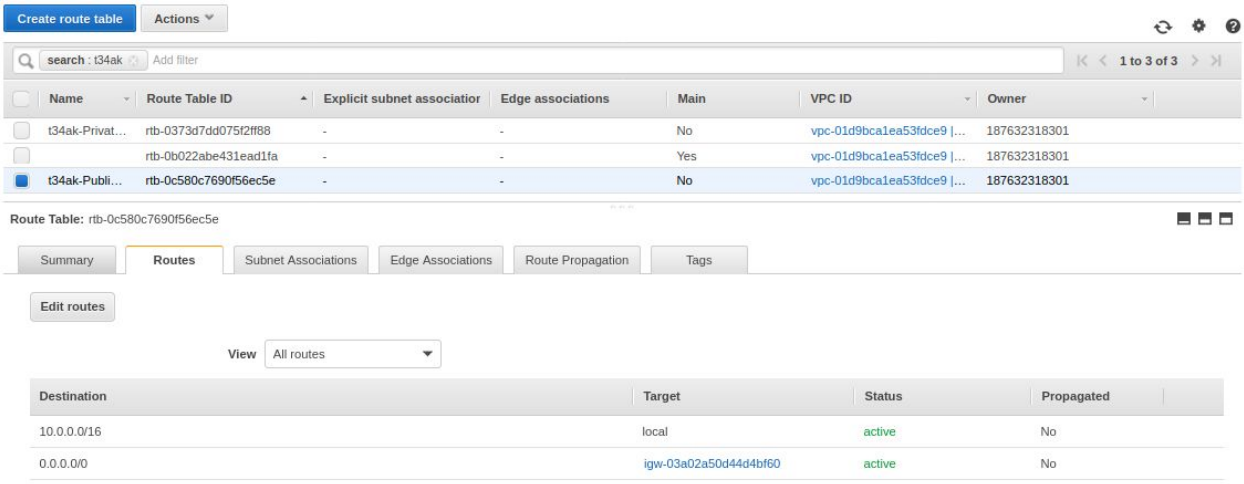


To access the instance with the private IP we have to access it with the Instance with which it is attach with public subnet

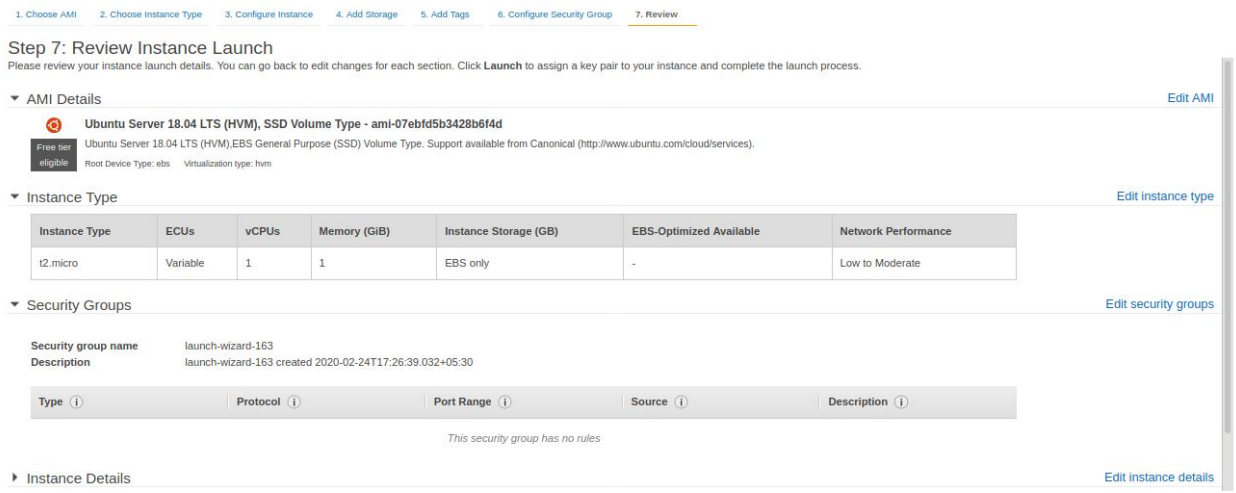


2. Create a public subnet, and host a web server(Nginx), also proxypass to Tomcat from Nginx

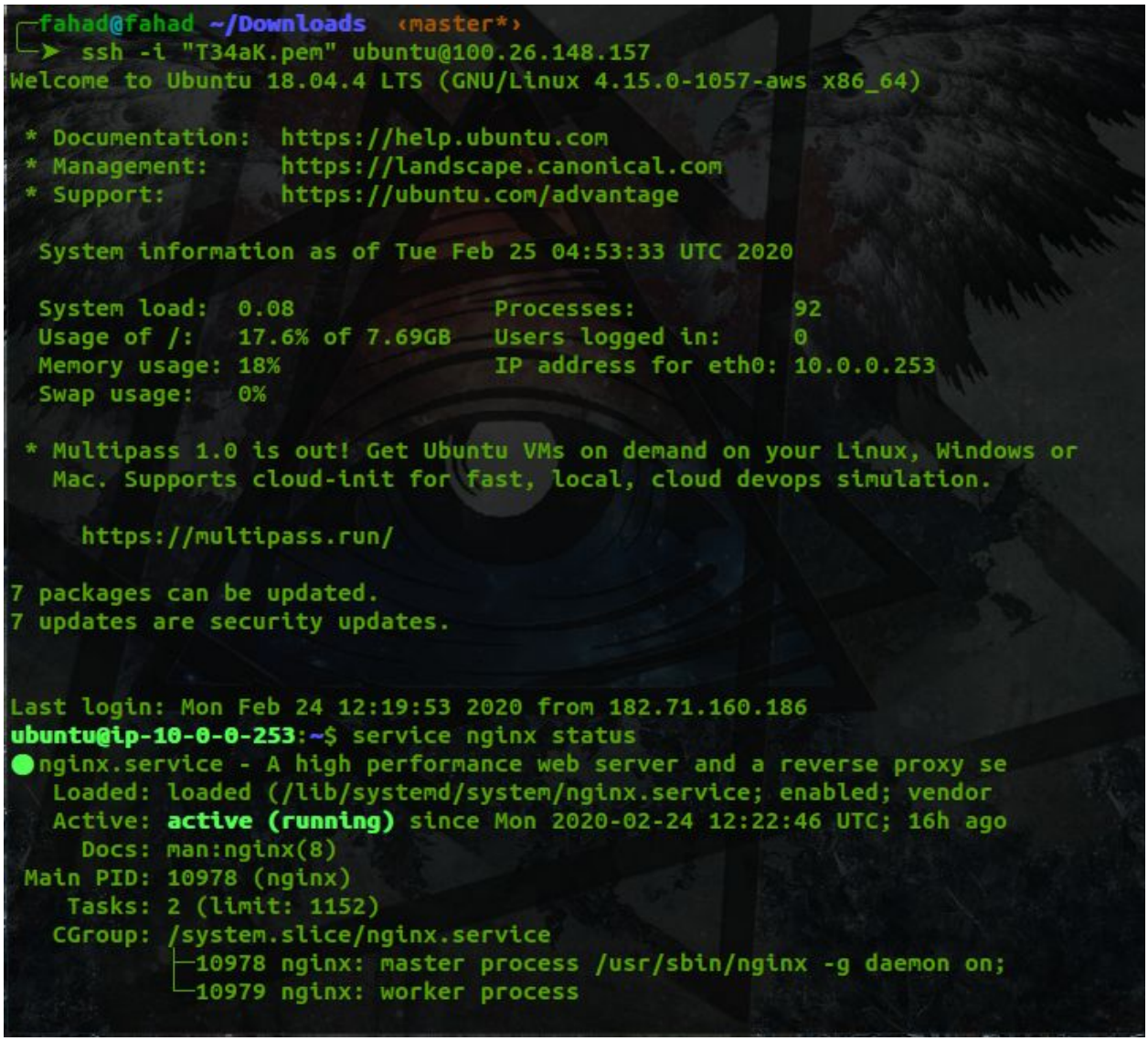
-Now we have to create a public subnet and also have to attach the IGW(internet Gateway) to that public subnet.



Now we have to launch Instance with the public subnet



Now we have to ssh in to that public INSTANCE then we have to install the the nginx



For proxy passing we have to change the conf file

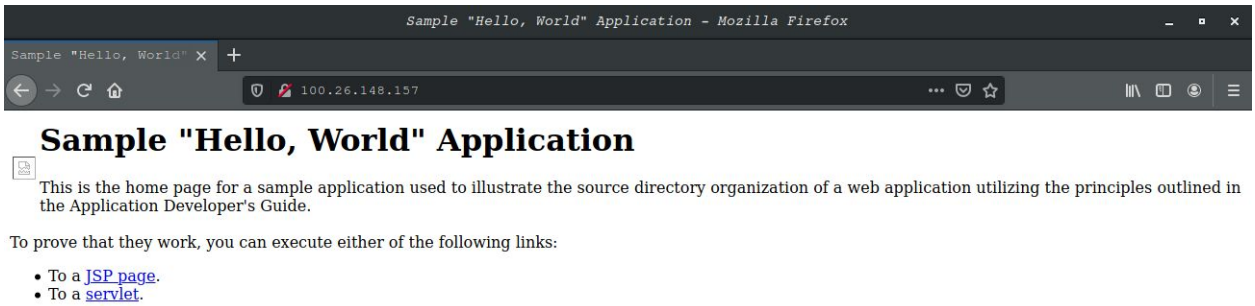
```
root /var/www/html;

# 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://10.0.63.53:8080/sample/;
        try_files $uri $uri/ =404;
}
```

After proxypassing it is redirecting to the tomcat page



Architecture diagram for this use case