VPC

Assessment

Name – Maithely Sharma
College – University of Petroleum and Energy Studies
EmployeeID – 4057

Q1. When to use Elastic IP over Public IP

It is assigned to your AWS account.Elastic IP do not change and they remain same even if you terminate the instance and later again restart the same instance.

WHEN TO USE:

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

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

RFC1918 name	IP address range	Number of addresses	Largest CIDR block (subnet mask)	Host ID size	Mask bits	*Classful* description
24-bit block	10.0.0.0 – 10.255.255.255	16777216	10.0.0.0/8 (255.0.0.0)	24 bits	8 bits	single class A network
20-bit block	172.16.0.0 – 172.31.255.255	1048576	172.16.0.0/12 (255.240.0.0)	20 bits	12 bits	16 contiguous class B networks
16-bit block	192.168.0.0 – 192.168.255.255	65536	192.168.0.0/16 (255.255.0.0)	16 bits	16 bits	256 contiguous class C networks

A public IP is assigned to a range or block of addresses. The Internet Assigned Numbers Authority (IANA) controls ownership of these IP ranges and assigns each block to organizations such as Internet Service Providers (ISPs) who in turn allocate individual IP addresses to customers.

ISPs shouldn't let **private-IP ranges** out onto the **public** Internet. This convention is why people usually **use** them when indicated. If the two computers are only connected

to each other, then you have no need - or ability - to **use public IP** addresses. A **public IP** is assigned to a **range** or block of **addresses**.

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

- VPC Peering is allowed for the connection of two VPC's such that the instances in the VPC can communicate with each other. The VPC's can be part of multiple accounts, ut must be in the same region.
- When you enable VPC peering between two VPCs, those VPCs must exist within the same region
- VPC peering is that the instances within a VPC communicate with instances in a peered VPC using either the IPv4 or the IPv6 protocol.
- VPCs that have been peered together cannot contain duplicate IP addresses or overlapping IP address scopes.
- AWS only allows you to create a single peering relationship between two VPCs. Of course this limitation is common sense, because there is no real advantage to creating multiple peer links between the same set of VPCs.
- No support for transitive peering.

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.

CIDR of VPC is 10.0.0.0/16, Total number of IP's in VPC = 2^16

If one subnet is of /20 cidr then total Subnet= 2^16/2^12 = 16 Subnet

Total number of IP in one sunet = 2^12-5 = 4091

Q5. Differentiate between NACL and Security Groups.

Security Group	**Network ACL**

Supports Allow rules only { by default all rules are denied **}** You **cannot deny** a certain IP address from establishing a connection	**Supports Allow and Deny rules**
Stateful: This means any changes applied to an incoming rule will be automatically applied to the outgoing rule. Example: If you allow an incoming port 80, the outgoing port 80 will be automatically opened.	**Stateless**: This means any changes applied to an incoming rule will not be applied to the outgoing rule. Example: If you allow an incoming port 80, you would also need to apply the rule for outgoing traffic.
Security groups are tied to an instance.	**Network ACL are tied to the subnet.**

All rules in a security group are applied.	**Rules are applied in their order (the rule with the lower number gets processed first)**
First Layer of Defence	**Second Layer of the defence**
Is the Firewall of EC2 Instances	**Is the Firewall of the Subnet**
Security groups are used for many cases, for example restricting inbound traffic of an EC2 instance to be from Load balancer only.	The same thing applies for Network ACL Used in running a production server

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.

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Spec
Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 C

Name tag maithely

IPv4 CIDR block* 10.0.0.0/16

IPv6 CIDR block
- No IPv6 CIDR Block
- Amazon provided IPv6 CIDR block
- IPv6 CIDR owned by me

Tenancy Default

*** Required**

VPC ID : vpc-0c297a0d16cde7bbd | Add filter

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set
maithely	vpc-0c297a0d16cde7bbd	available	10.0.0.0/16	-	dopt-519d6f34

Create IGW

Internet gateways > Create internet gateway

Create internet gateway

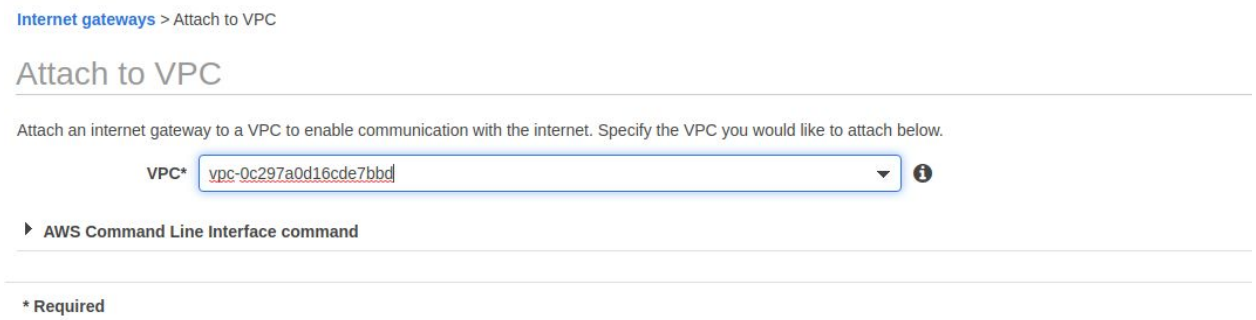
An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Name tag maithely-igw

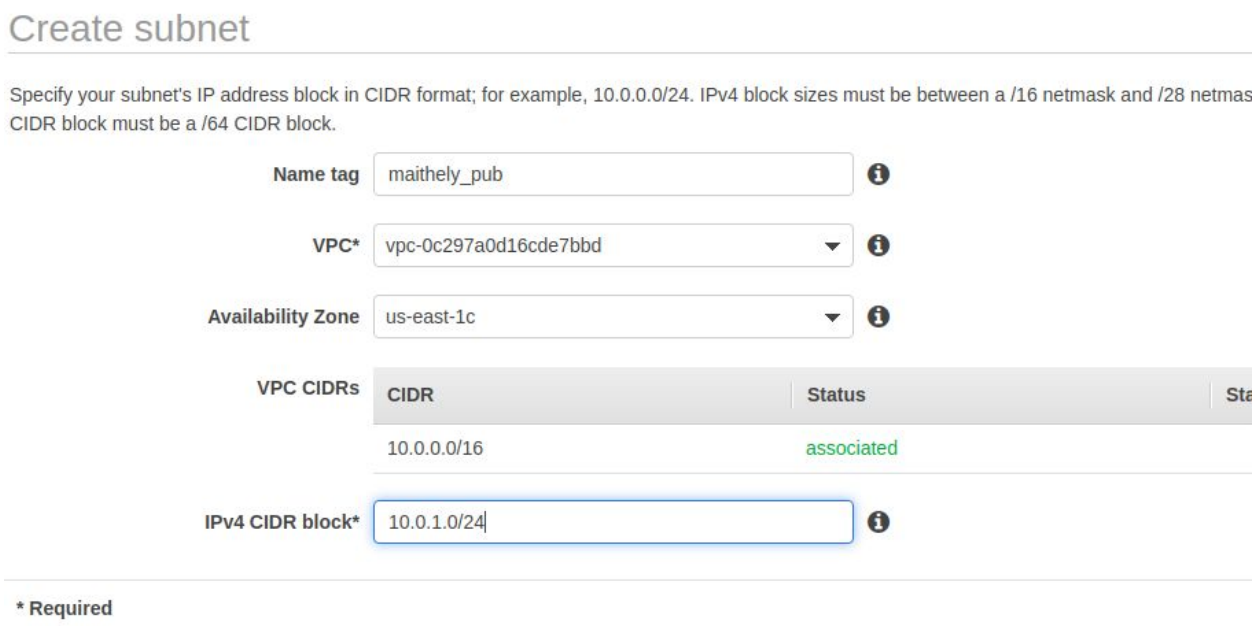
*** Required**

Cancel

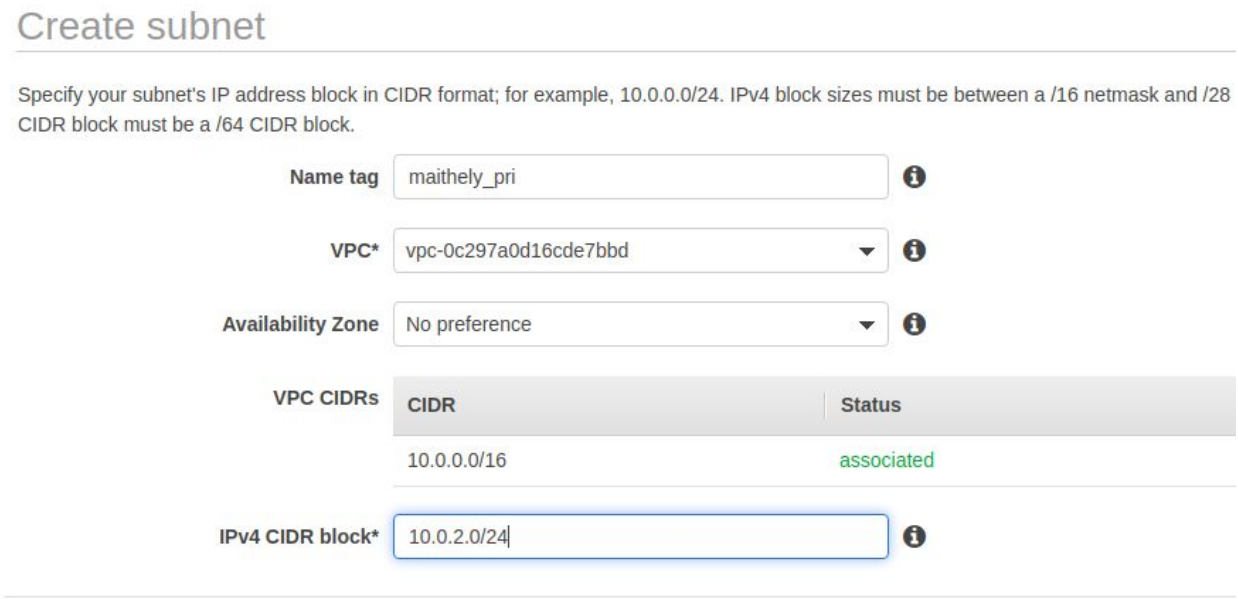
Attach igw to vpc



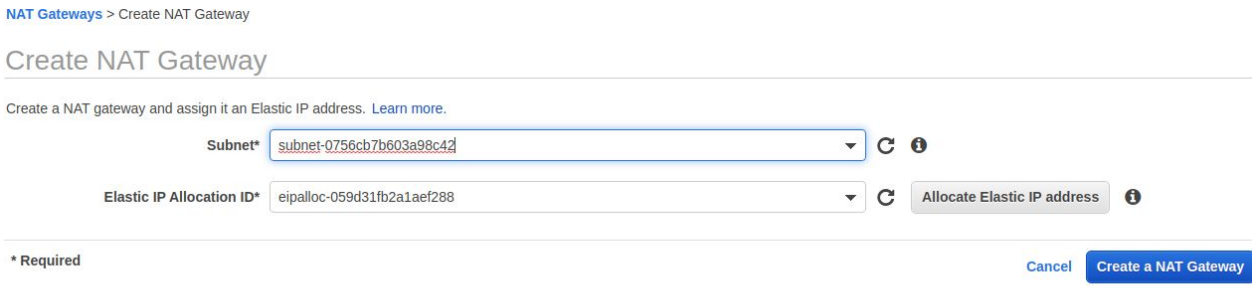
Public subnet



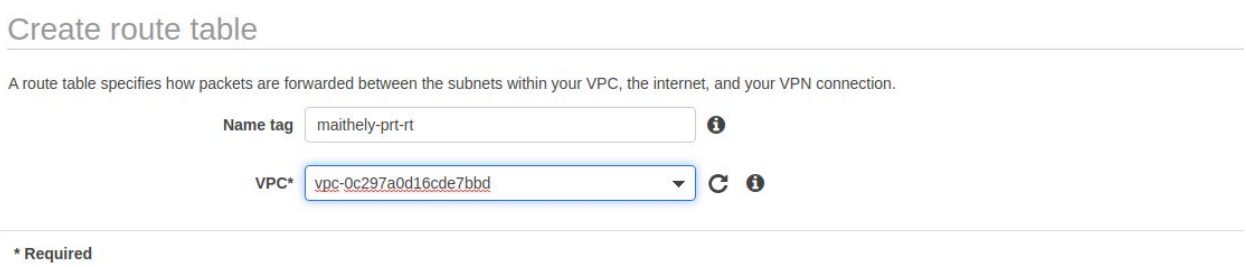
Private subnet



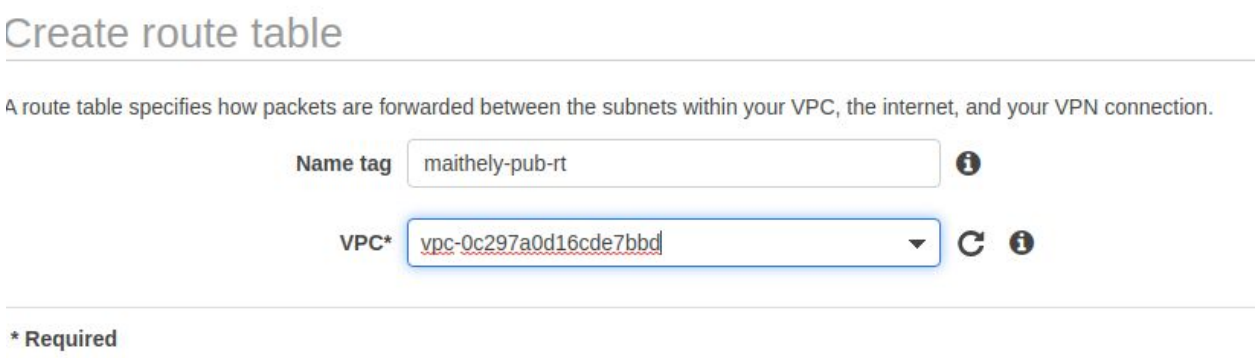
Now create NAT in public subnet



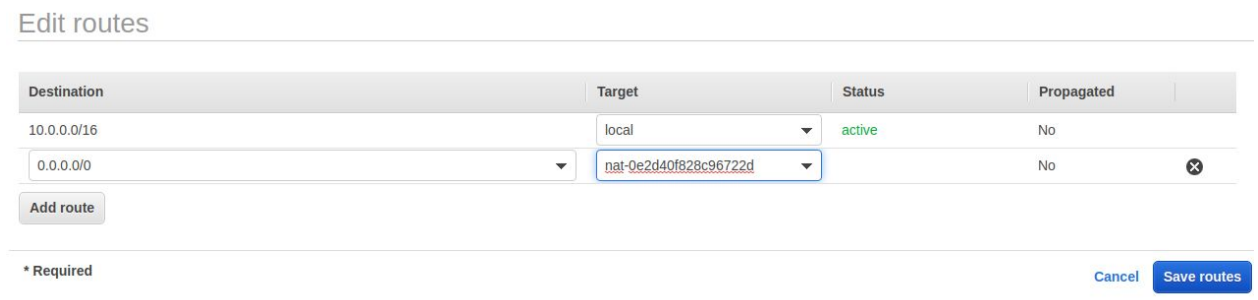
Now create route table :private



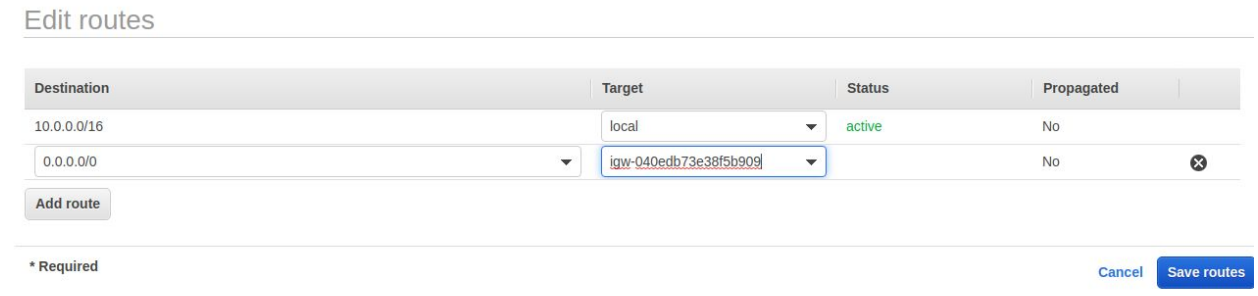
Create public rt



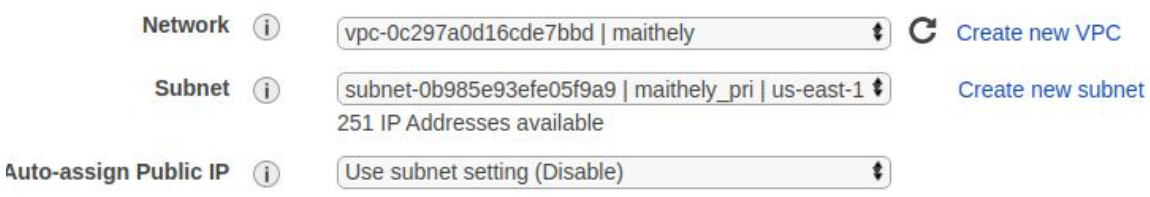
Now add nat in route table of private

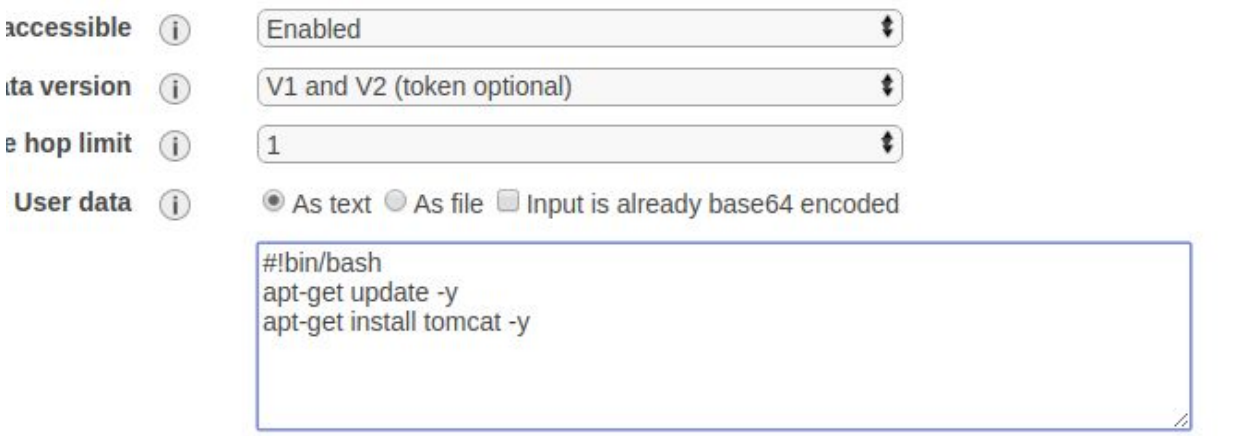


Now add igw to public rt

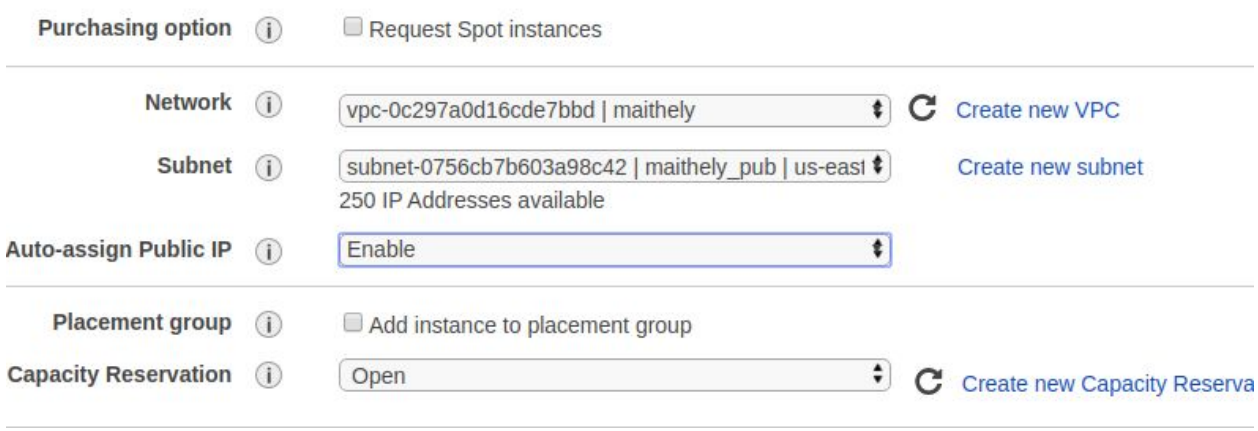


Create an instance in private subnet





Now create one public instance



Public inbound

Private inbound

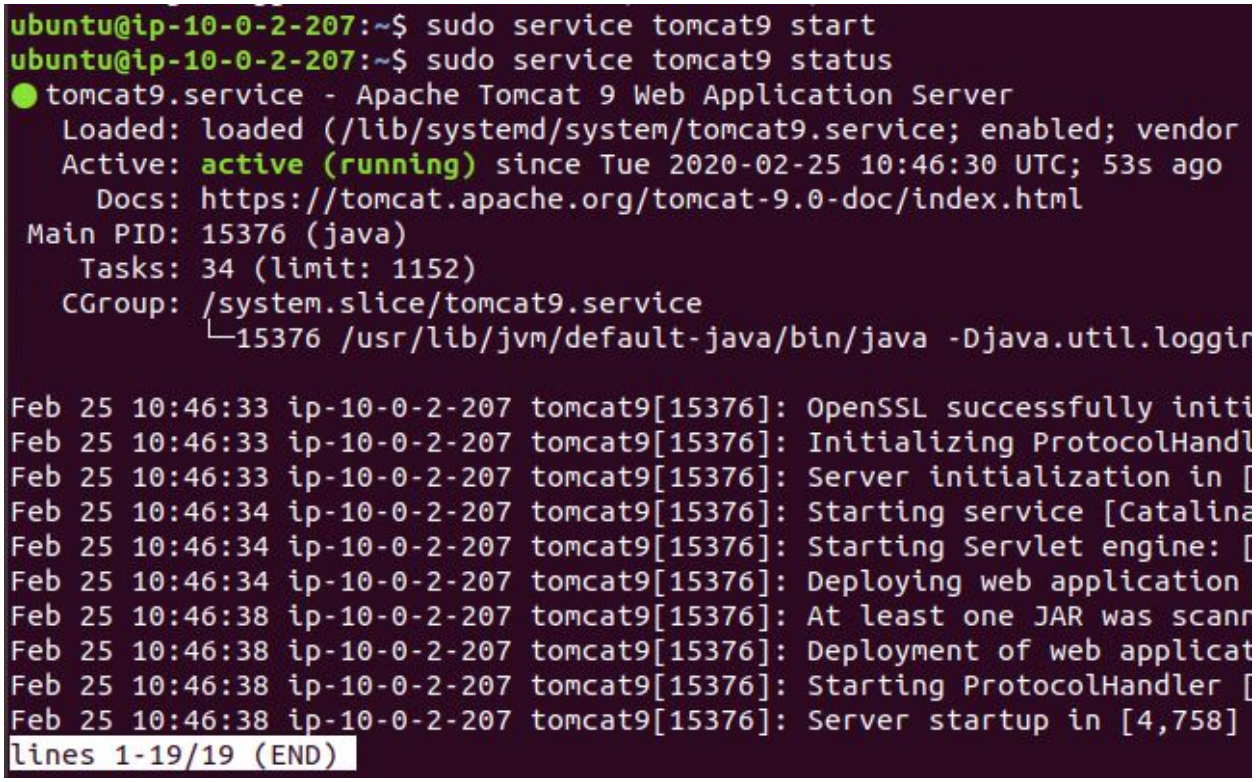
Now ssh in public

```
maithely@maithely:~/Downloads$ ssh -i "maithelykeypair.pem" ubuntu@3.92.209.206
The authenticity of host '3.92.209.206 (3.92.209.206)' can't be established.
ECDSA key fingerprint is SHA256:oIWwiM9xJ6lpmKD1EgAxpE80eLdOGPSkhLtI+GNzRKc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '3.92.209.206' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Feb 25 10:33:55 UTC 2020

  System load:  0.0               Processes:           86
  Usage of /:   13.8% of 7.69GB   Users logged in:     0
  Memory usage: 18%               IP address for eth0: 10.0.1.246
  Swap usage:   0%

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-246:~$ exit
```

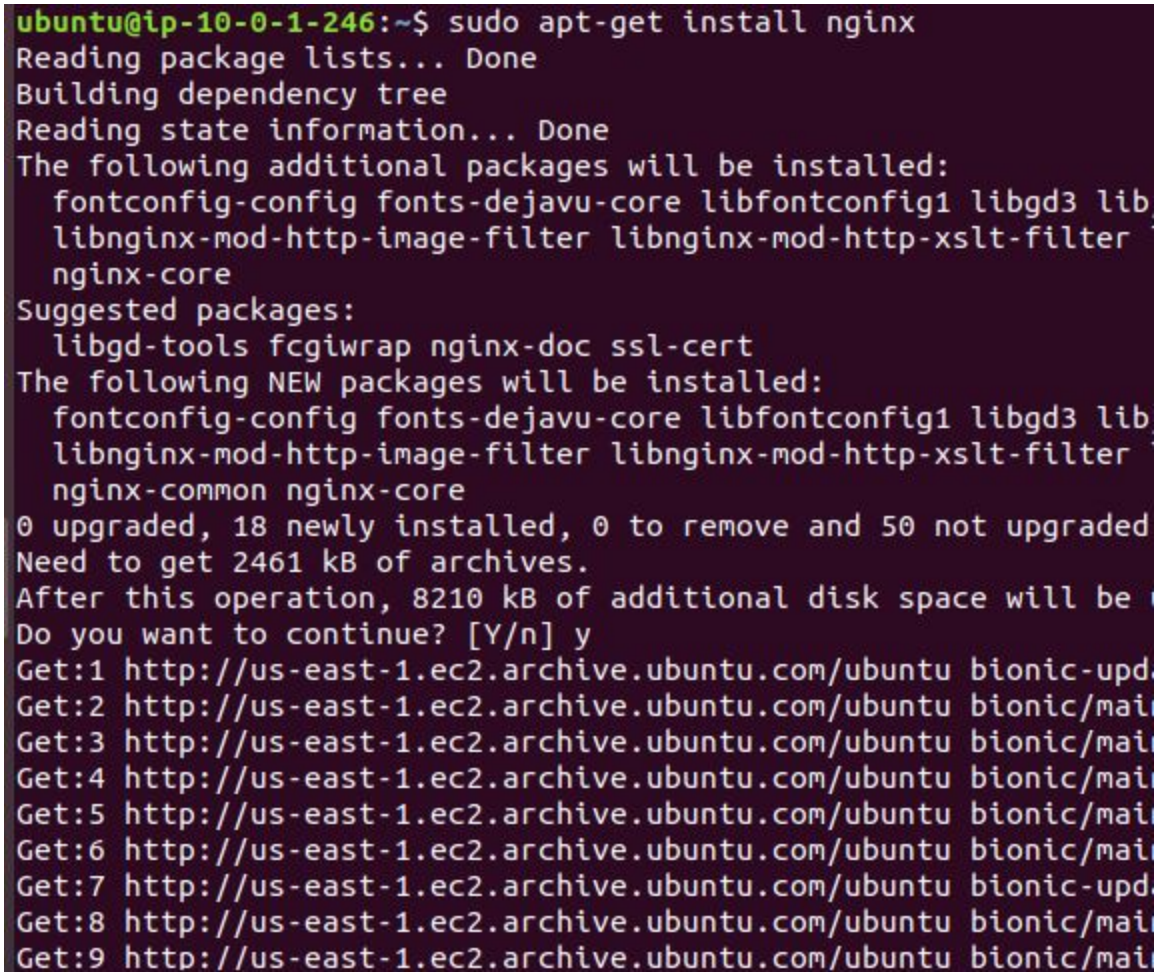
```
maithely@maithely:~/Downloads$ scp -i maithelykeypair.pem maithelykeypair.pem ubuntu@3.92.209.206:
maithelykeypair.pem                                           100% 1692     3.6KB/s   00:00
```

```
maithely@maithely:~/Downloads$ ssh -i "maithelykeypair.pem" ubuntu@3.92.209.206
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Feb 25 10:37:21 UTC 2020

  System load:  0.0               Processes:           88
  Usage of /:   13.8% of 7.69GB   Users logged in:     0
  Memory usage: 18%               IP address for eth0: 10.0.1.246
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.


Last login: Tue Feb 25 10:33:57 2020 from 61.12.91.218
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-246:~$ ls
maithelykeypair.pem
```

```
ubuntu@ip-10-0-1-246:~$ ssh -i maithelykeypair.pem ubuntu@10.0.2.207
The authenticity of host '10.0.2.207 (10.0.2.207)' can't be established.
ECDSA key fingerprint is SHA256:K+I2KdHRHYspx6AeFJRE6PvJGP1Bs41r7Ashx2GcVes.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.207' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Feb 25 10:42:02 UTC 2020

  System load:  0.08              Processes:           86
  Usage of /:   13.8% of 7.69GB   Users logged in:     0
  Memory usage: 18%               IP address for eth0: 10.0.2.207
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your I


Last login: Tue Feb 25 10:26:14 2020 from 10.0.1.107
ubuntu@ip-10-0-2-207:~$ cd /etc/t
terminfo/   tmpfiles.d/
ubuntu@ip-10-0-2-207:~$ sudo apt-get update
0% [Connecting to us-east-1.ec2.archive.ubuntu.com (34.229.150.131)] [Connecting
ubuntu@ip-10-0-2-207:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [8
```



In public instance install nginx



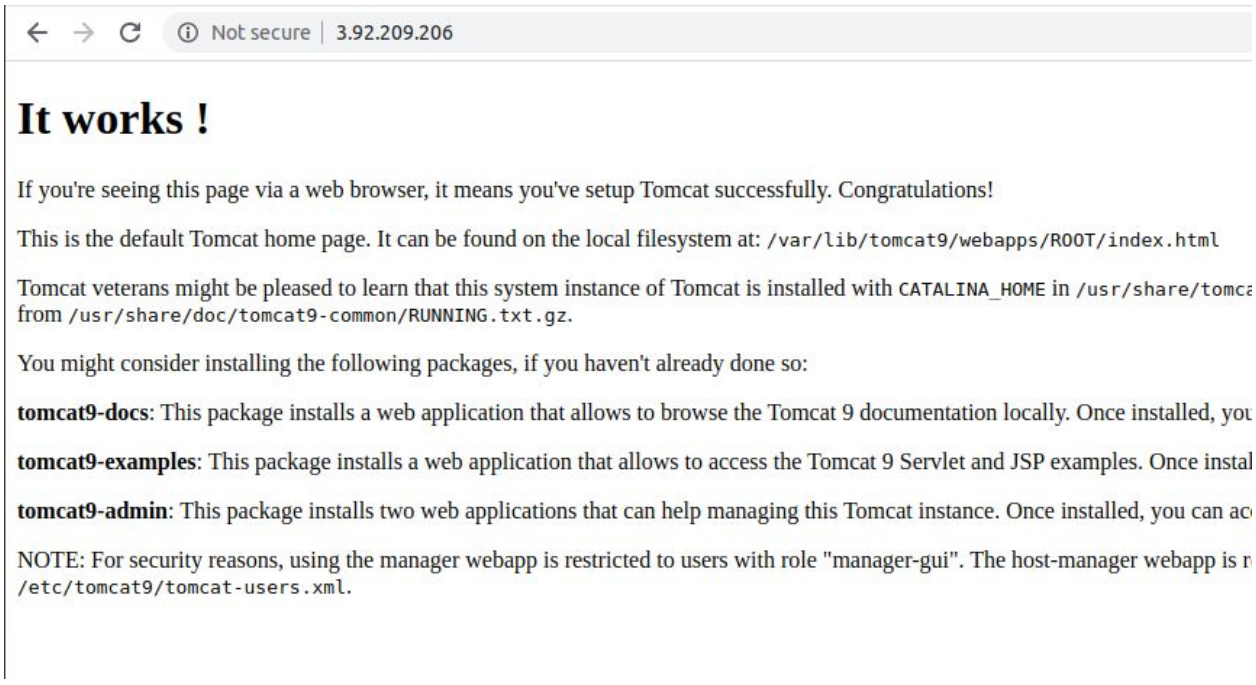
Now add location block in /etc/nginx/sites-enabled/default

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        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.
                try_files $uri $uri/ =404;
                proxy_pass http://10.0.2.207:8080/;
        }
```

```
ubuntu@ip-10-0-1-246:/etc/nginx/sites-available$ curl 10.0.2.207:8080
<?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 Tom

<p>This is the default Tomcat home page. It can be found on the local file

<p>Tomcat veterans might be pleased to learn that this system instance of
are/tomcat9</code> and <code>CATALINA_BASE</code> in <code>/var/lib/tomcat
```

```
ubuntu@ip-10-0-1-246:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-10-0-1-246:/etc/nginx/sites-available$ sudo nginx -s reload
```

Now write private ip in the browser



Architecture diagram for this use case