Q1. When to use Elastic IP over Public IP

We use elastic IP over public ip when we want that IP doesn’t change when we restart the instance.

For eg: If we are hosting a website on an Ec2 instance we will give that instance an elastic ip so that it doesn’t get changed.

We use elastic Ip in the NAT server in a VPC.

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

Class A Network - 10.0.0.0 - 10.255.255.255

Class B Network - 172.16.0.0 - 172.31.255.255

Class C Network - 192.168.0.0 - 192.168.255.255

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

1. The owner of the requester VPC sends a request to the owner of the accepter VPC to create the VPC peering connection. The accepter VPC can be owned by you, or another AWS account, and cannot have a CIDR block that overlaps with the requestor VPC's CIDR block.

2. The owner of the accepter VPC accepts the VPC peering connection request to activate the VPC peering connection. 3. To enable the flow of traffic between the VPCs using private IP addresses, the owner of each VPC in the VPC peering connection must manually add a route to one or

more of their VPC route tables that points to the IP address range of the other VPC (the peer VPC). 4. If required, update the security group rules that are associated with your instance to ensure that traffic to and from the peer VPC is not restricted. 5. Both the VPC should have different CIDR range otherwise there will be an ambiguity in the route table and Network access control lists.

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.

No. of subnets = 20 - 16 = 4 so 2^4 =16 subnets

Ip in each subnet= 32 - 20 = 12 so 2^{12} = 4096 Ip in a particular subnet.

Q5. Differentiate between NACL and Security Groups.

NACL has Rule number priority whereas Security Group doesn’t have.

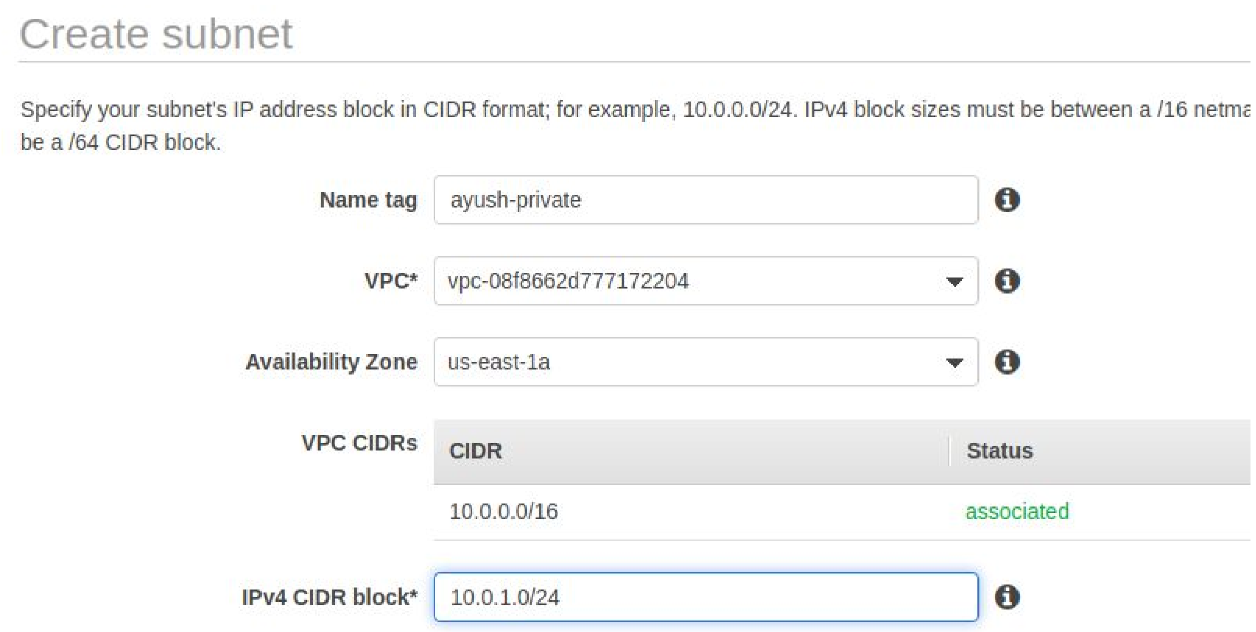
In NACL we can allow and deny both and in Security group we can only allow.

NACL acts as a firewall for a particular subnet whereas Security group acts as a firewall for a particular EC2 instance.

NACL are stateless which means whenever we allow a particular port in inbound it will not be automatically applied in outbound. Whereas Security group is stateful

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

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



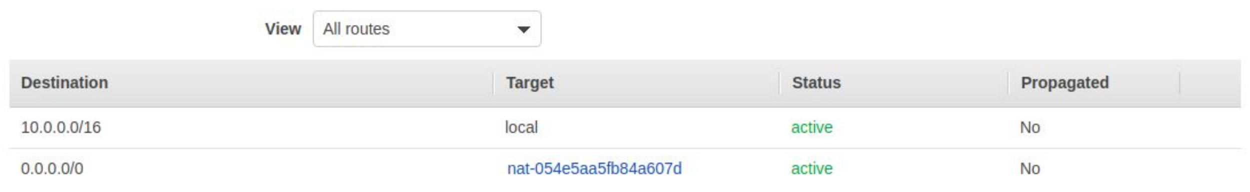
Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netma be a /64 CIDR block.

Name tag	ayush-private
VPC*	vpc-08f8662d777172204
Availability Zone	us-east-1a
IPv4 CIDR block*	10.0.1.0/24

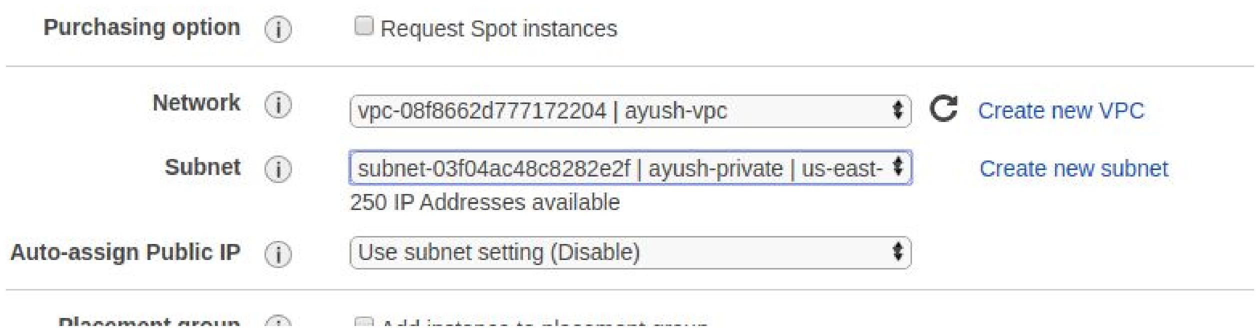
VPC CIDRs

CIDR	Status
10.0.0.0/16	associated

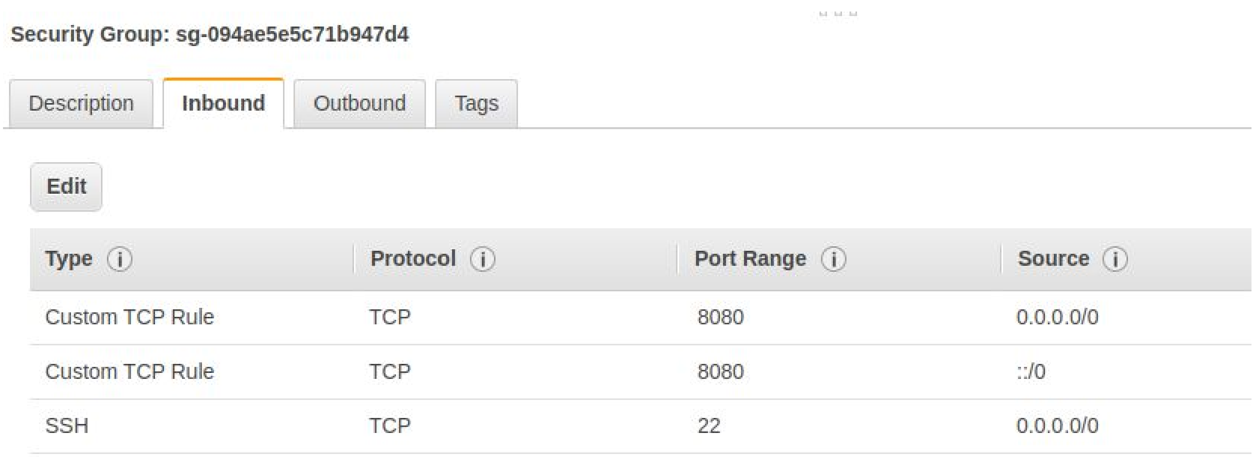


View: All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-054e5aa5fb84a607d	active	No

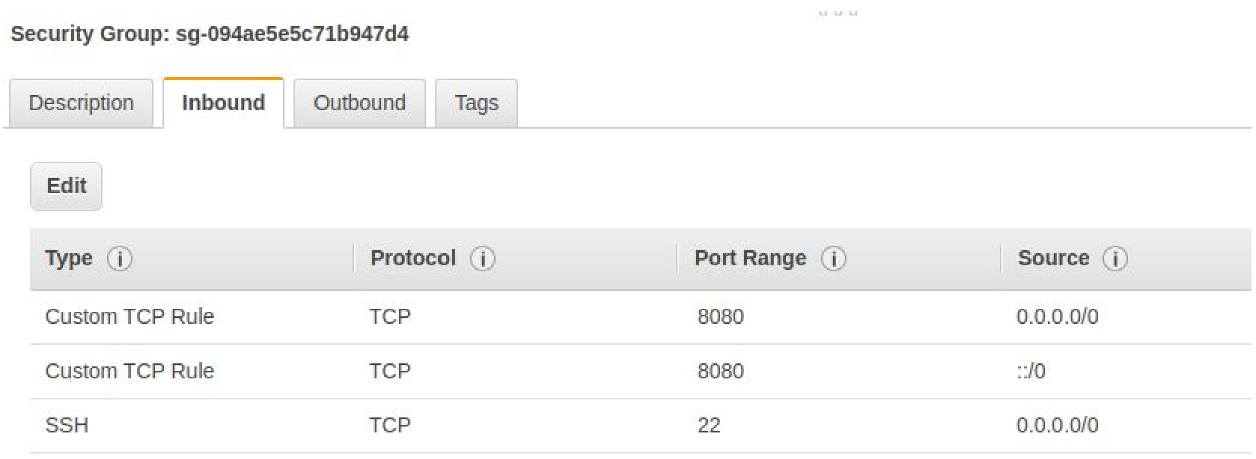


Purchasing option	☐ Request Spot instances
Network	vpc-08f8662d777172204 | ayush-vpc
Subnet	subnet-03f04ac48c8282e2f | ayush-private | us-east- 250 IP Addresses available
Auto-assign Public IP	Use subnet setting (Disable)



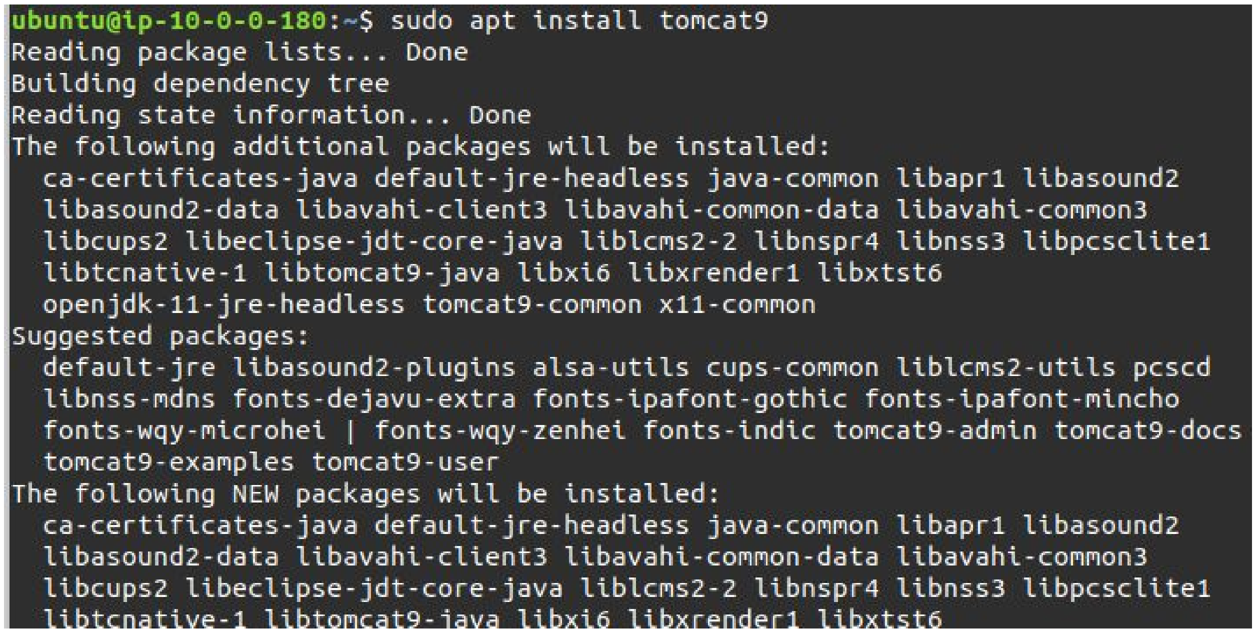
Security Group: sg-094ae5e5c71b947d4

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	8080	::/0
SSH	TCP	22	0.0.0.0/0



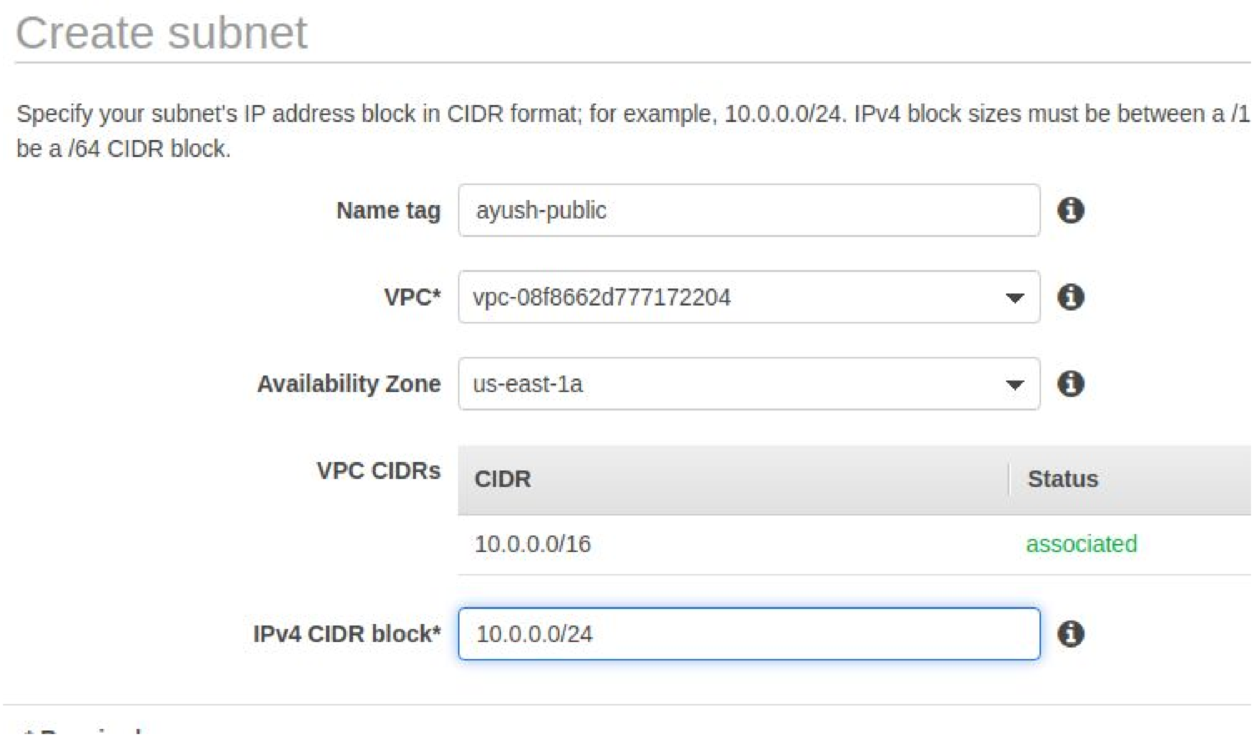
Security Group: sg-094ae5e5c71b947d4

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	8080	::/0
SSH	TCP	22	0.0.0.0/0



```
ubuntu@ip-10-0-0-180:~$ sudo apt install tomcat9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless java-common libapr1 libasound2
  libasound2-data libavahi-client3 libavahi-common-data libavahi-common3
  libcups2 libeclipse-jdt-core-java liblcms2-2 libnspr4 libnss3 libpcsclite1
  libtcnative-1 libtomcat9-java libxi6 libxrender1 libxtst6
  openjdk-11-jre-headless tomcat9-common x11-common
Suggested packages:
  default-jre libasound2-plugins alsa-utils cups-common liblcms2-utils pcscd
  libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic tomcat9-admin tomcat9-docs
  tomcat9-examples tomcat9-user
The following NEW packages will be installed:
  ca-certificates-java default-jre-headless java-common libapr1 libasound2
  libasound2-data libavahi-client3 libavahi-common-data libavahi-common3
  libcups2 libeclipse-jdt-core-java liblcms2-2 libnspr4 libnss3 libpcsclite1
  libtcnative-1 libtomcat9-java libxi6 libxrender1 libxtst6
```

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



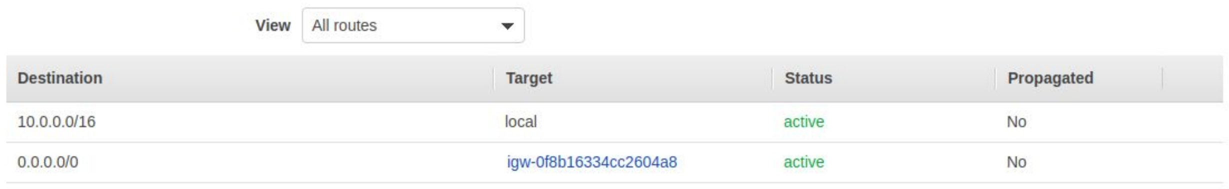
Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /1
be a /64 CIDR block.

Name tag	ayush-public
VPC*	vpc-08f8662d777172204
Availability Zone	us-east-1a
IPv4 CIDR block*	10.0.0.0/24

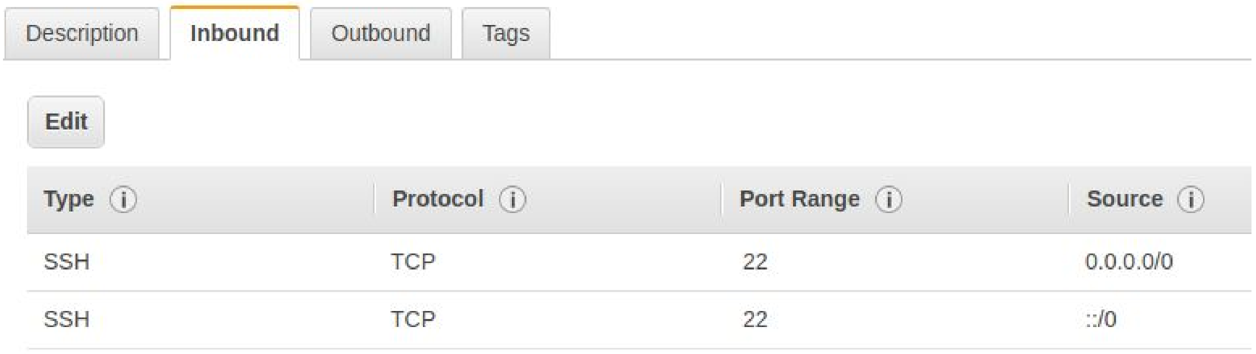
VPC CIDRs

CIDR	Status
10.0.0.0/16	associated

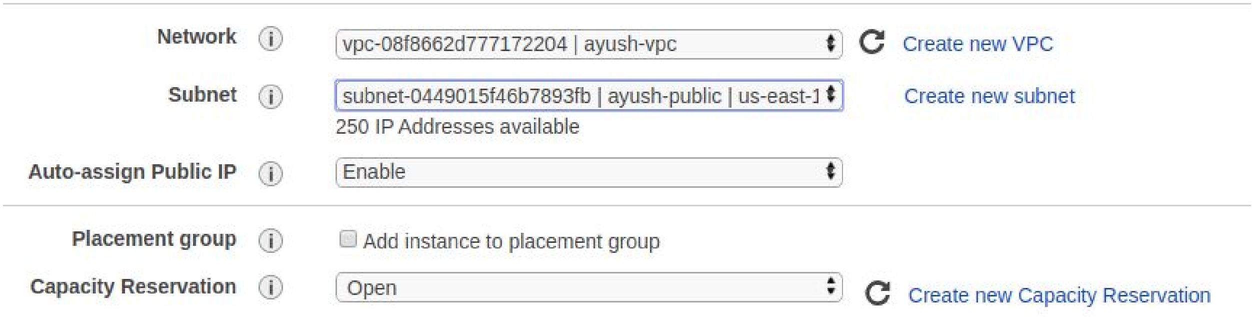


View: All routes

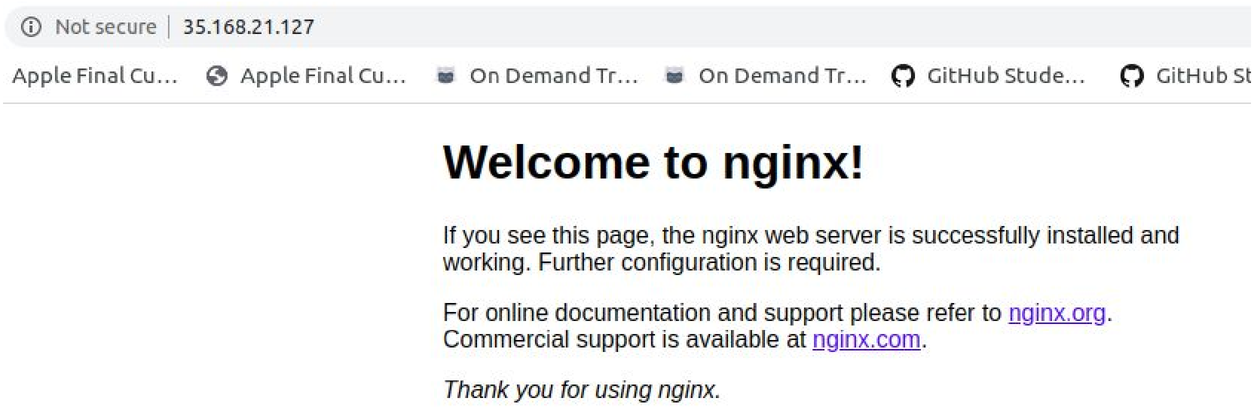
Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0f8b16334cc2604a8	active	No



Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
SSH	TCP	22	::/0



Network	vpc-08f8662d777172204 | ayush-vpc
Subnet	subnet-0449015f46b7893fb | ayush-public | us-east-1 250 IP Addresses available
Auto-assign Public IP	Enable
Placement group	☐ Add instance to placement group
Capacity Reservation	Open

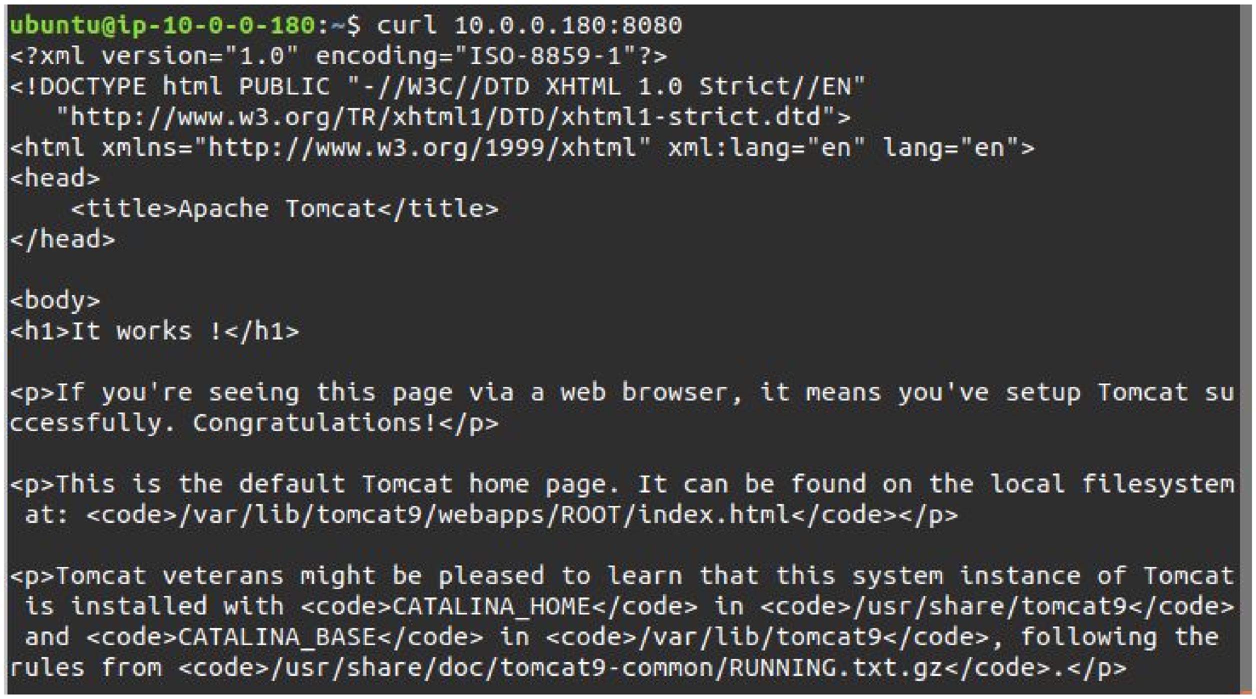


Welcome to nginx!

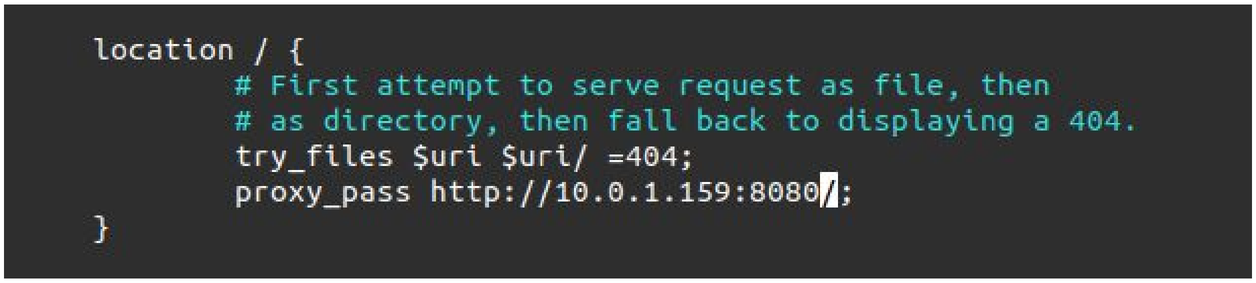
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

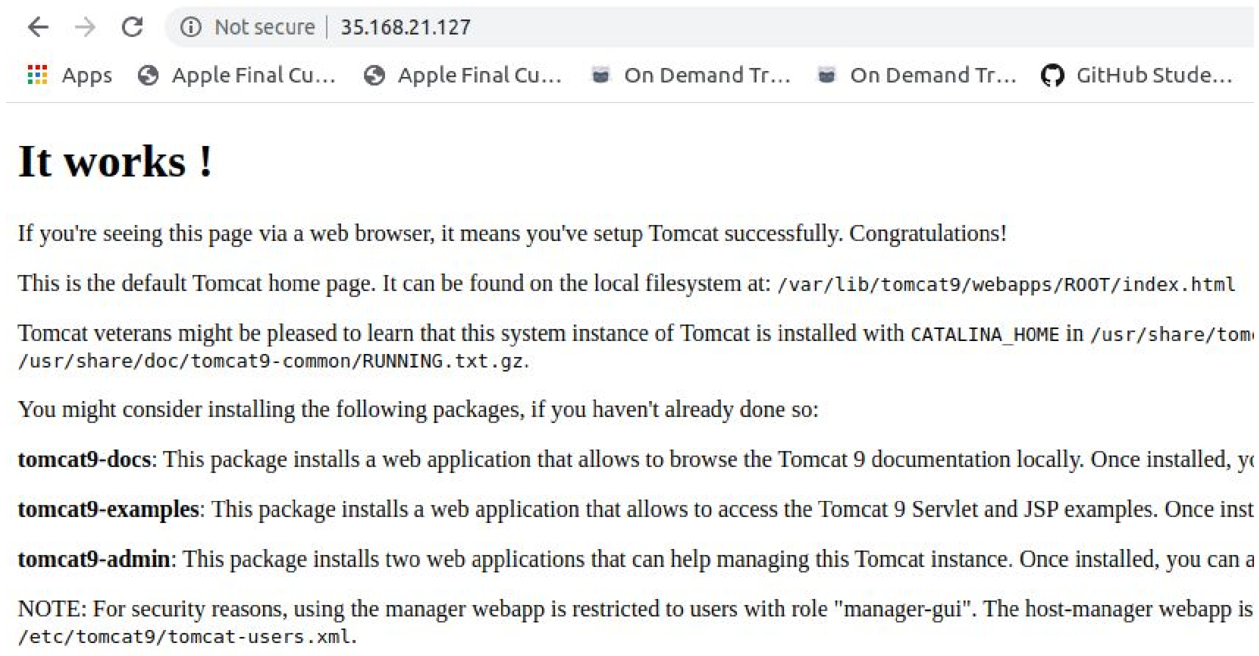
Thank you for using nginx.



```
ubuntu@ip-10-0-0-180:~$ curl 10.0.0.180:8080
<?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 su
ccessfully. 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/share/tomcat9</code>
 and <code>CATALINA_BASE</code> in <code>/var/lib/tomcat9</code>, following the
rules from <code>/usr/share/doc/tomcat9-common/RUNNING.txt.gz</code>.</p>
```



```
    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://10.0.1.159:8080/;
    }
```



It works !

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

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

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tom`
`/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

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

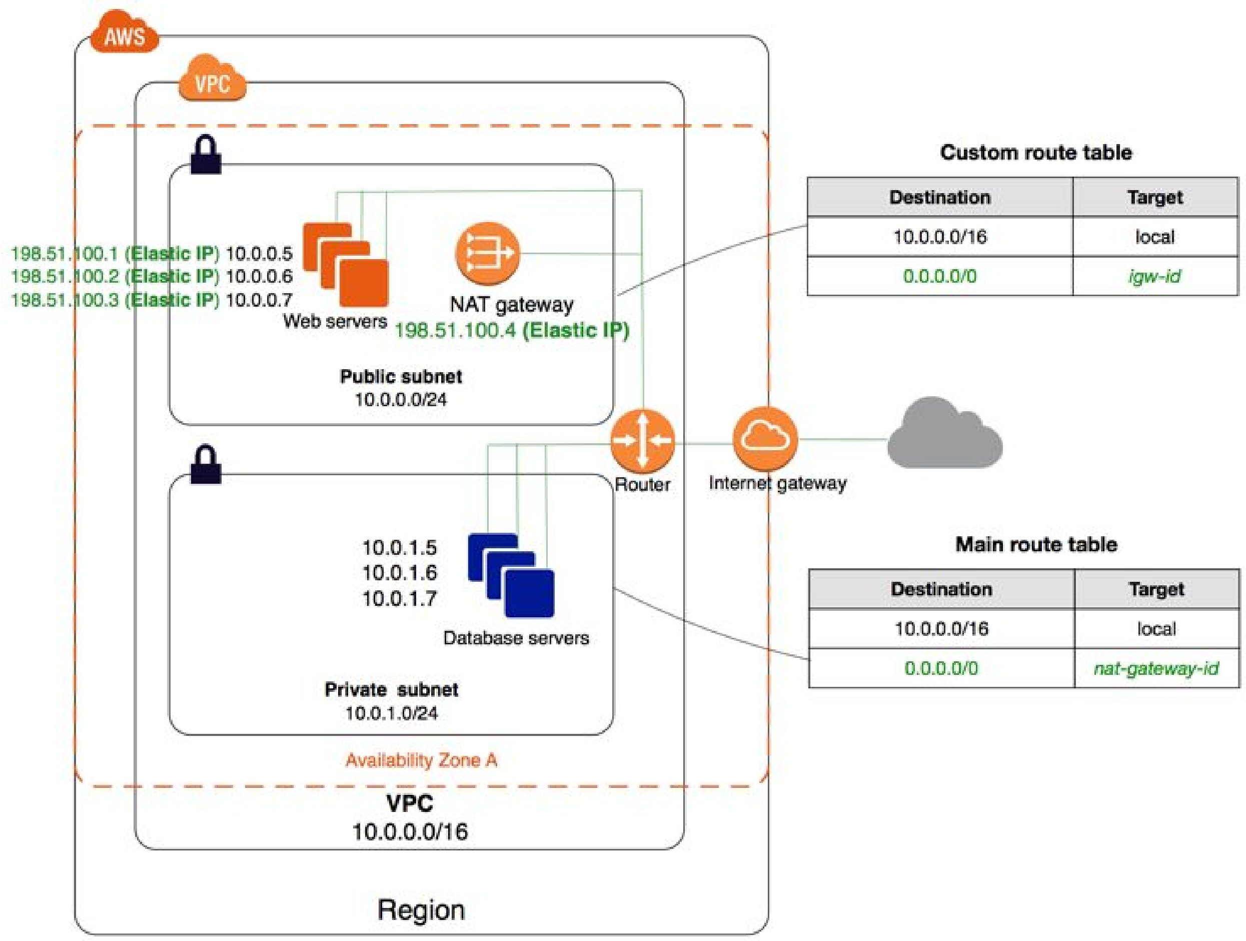
tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, yo

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once inst

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can a

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is
`/etc/tomcat9/tomcat-users.xml`.

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



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-gateway-id*