# Assessment - 10

## VPC

TO
THE
NEW

**Q1.When to use Elastic IP over Public IP**

**ANS :** When the public endpoint is a web server.

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

**ANS :** 10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

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

**ANS :**

1. You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks.
2. VPC peering does not support transitive peering relationships.
3. You cannot have more than one VPC peering connection between the same two VPCs at the same time.
4. Any tags that you create for your VPC peering connection are only applied in the account or region in which you create them.
5. You cannot connect to or query the Amazon DNS server in a peer VPC.

**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 IPs in the subnet.**

**ANS :** /20 being the subnet mask means that 4 extra bits are borrowed from the host so

No. of subnets = 2^4 = 16 subnets.

Total number of IPs in each subnet = 2^12(32-20)

**Q5. Differentiate between NACL and Security Groups.**

**ANS :**

| Security Group | NACL |
|---|---|
| It supports only allow rules | It supports both allow and deny rules |
| It is stateful | It is stateless |
| 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 is applied automatically to all the instances which are associated with a subnet. |
| It is the first layer of defense. | It is the second layer of defense. |

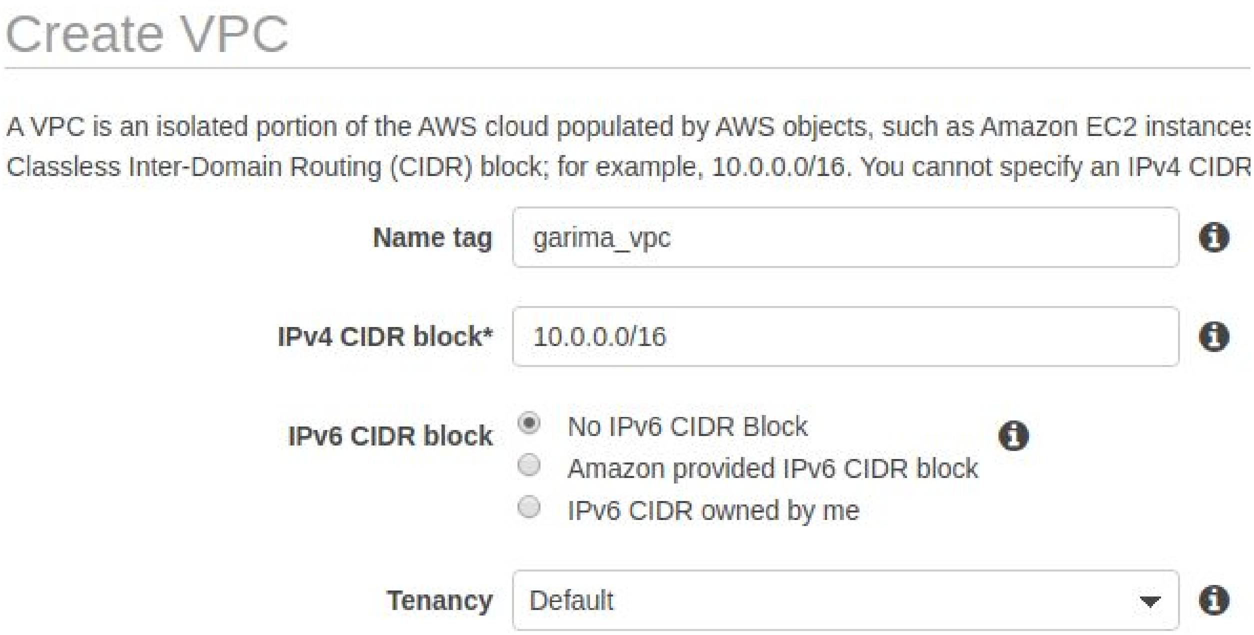
**Q6. Implement a 2-tier vpc with following requirements:**

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.**

**ANS:**



## Create VPC

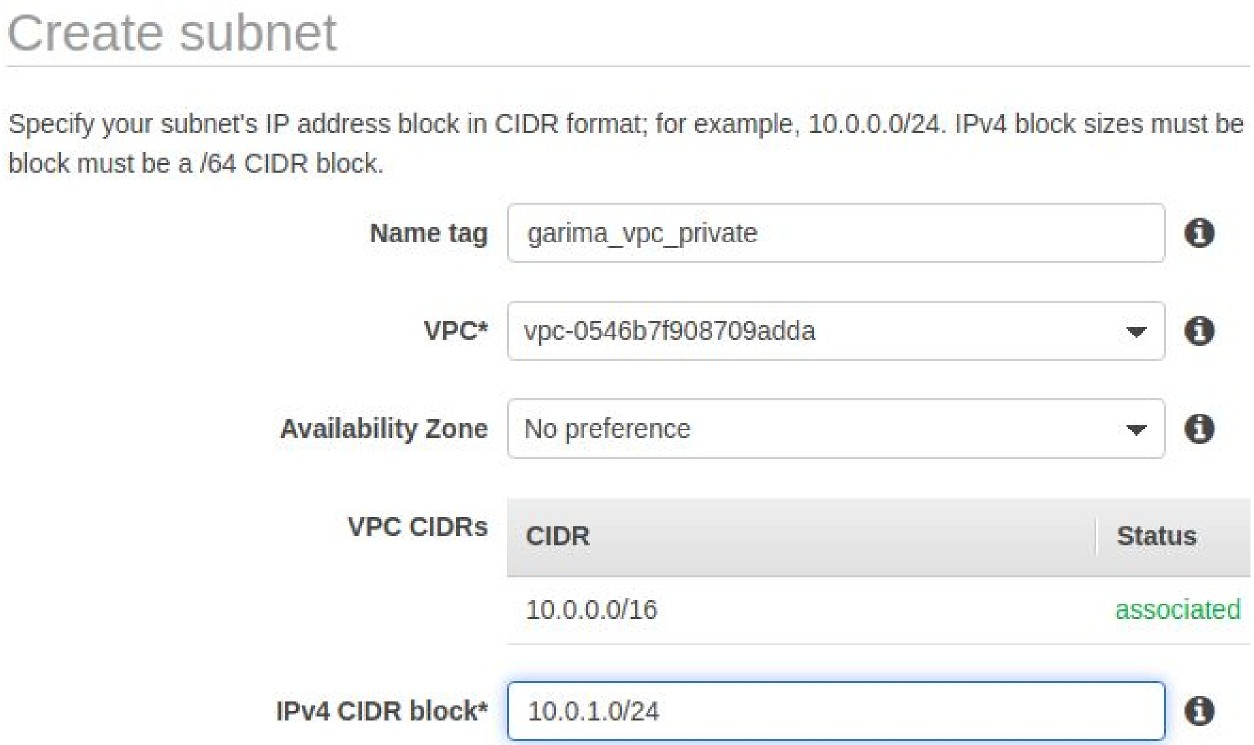
A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instance
Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR

**Name tag** garima_vpc

**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



## Create subnet

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

**Name tag** garima_vpc_private

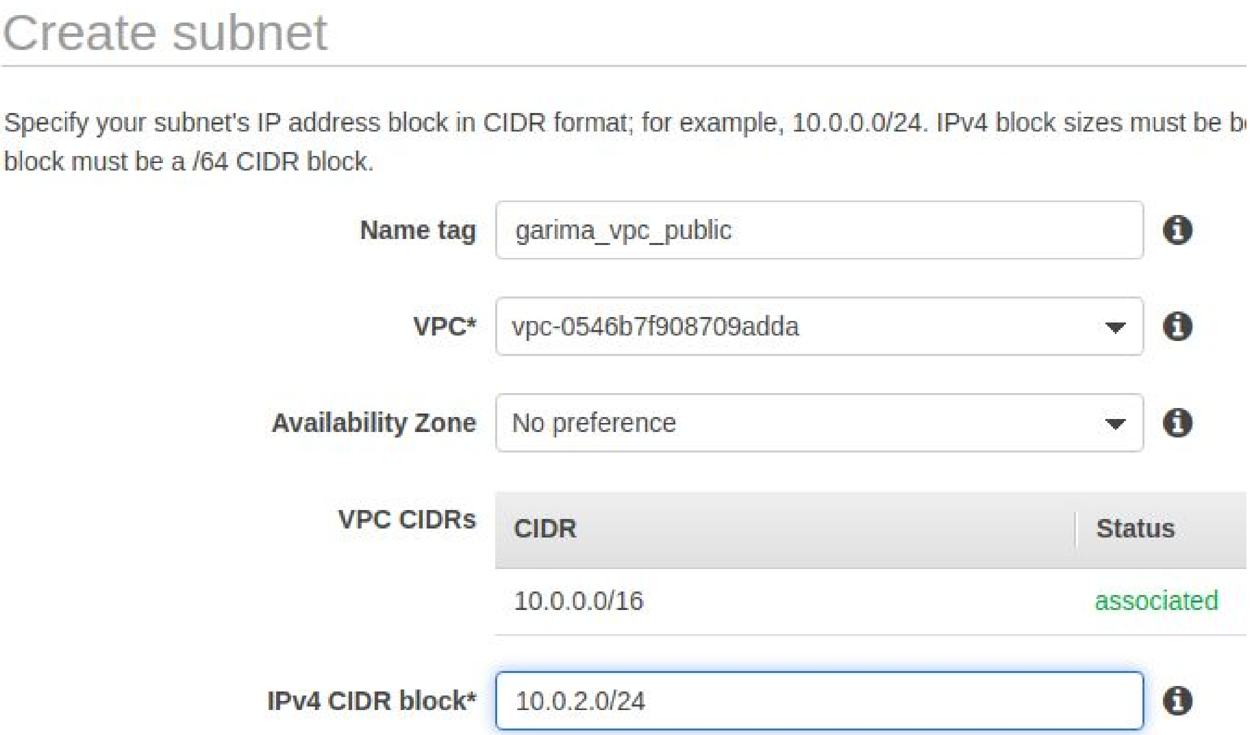
**VPC*** vpc-0546b7f908709adda

**Availability Zone** No preference

**VPC CIDRs**

| CIDR | Status |
|---|---|
| 10.0.0.0/16 | associated |

**IPv4 CIDR block*** 10.0.1.0/24



# Create subnet

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

**Name tag** garima_vpc_public

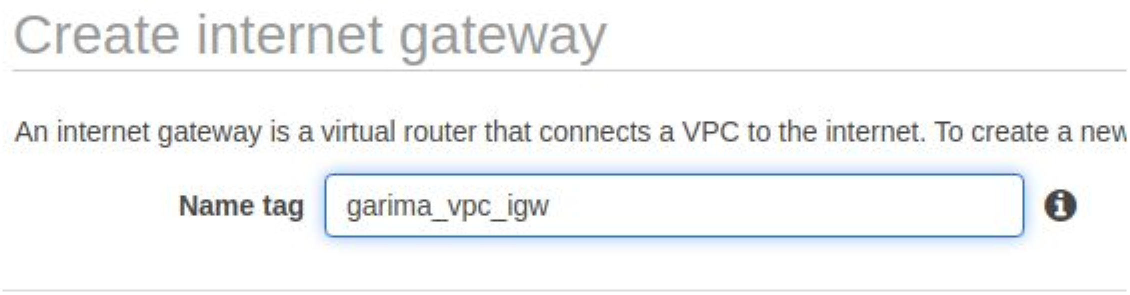
**VPC*** vpc-0546b7f908709adda

**Availability Zone** No preference

**VPC CIDRs**

| CIDR | Status |
|---|---|
| 10.0.0.0/16 | associated |

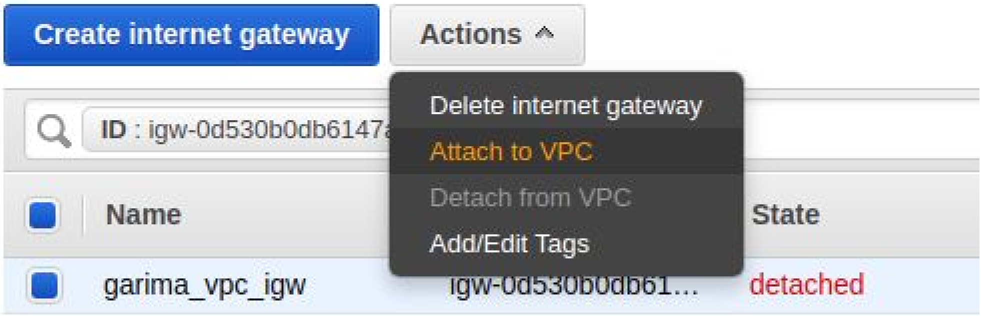
**IPv4 CIDR block*** 10.0.2.0/24



# Create internet gateway

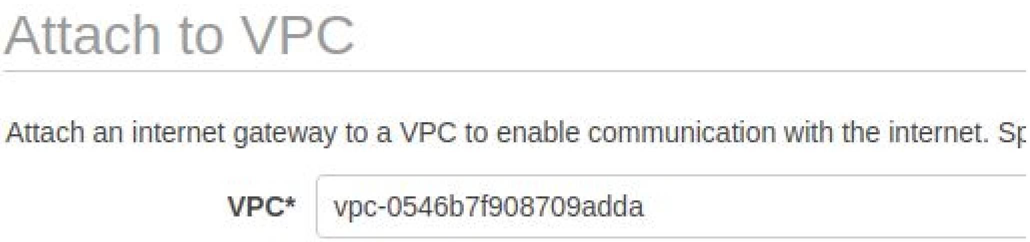
An internet gateway is a virtual router that connects a VPC to the internet. To create a new

**Name tag** garima_vpc_igw



ID : igw-0d530b0db6147a

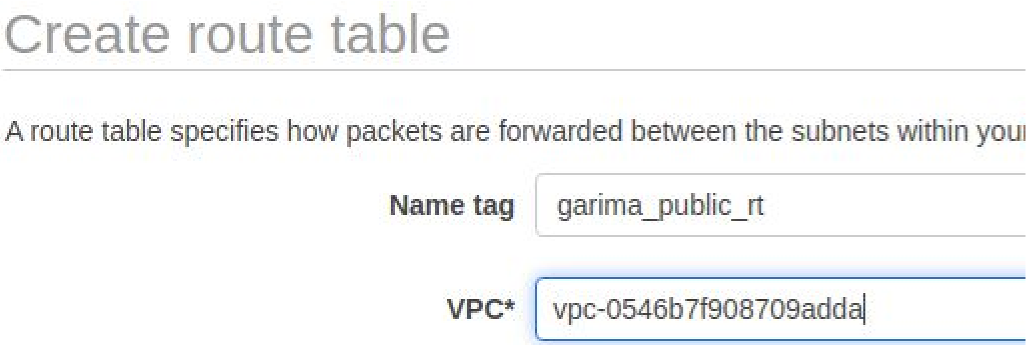
| Name | | State |
|---|---|---|
| garima_vpc_igw | igw-0d530b0db61… | detached |



## Attach to VPC

Attach an internet gateway to a VPC to enable communication with the internet. Sp

**VPC*** vpc-0546b7f908709adda

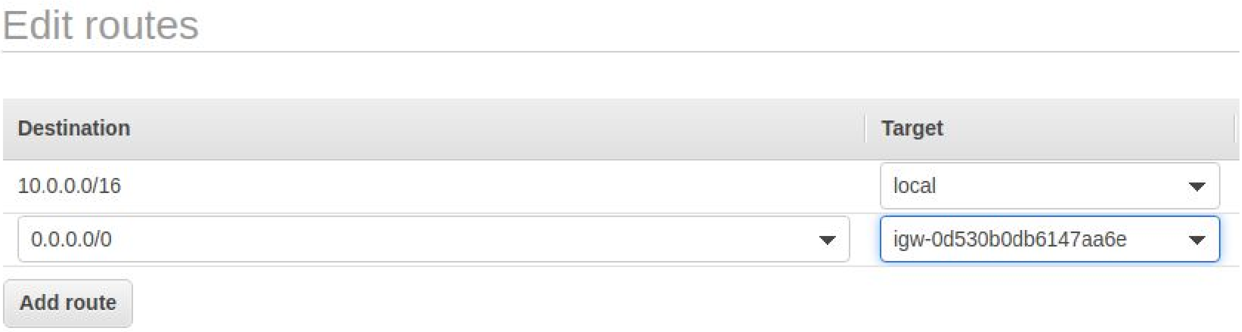


## Create route table

A route table specifies how packets are forwarded between the subnets within your

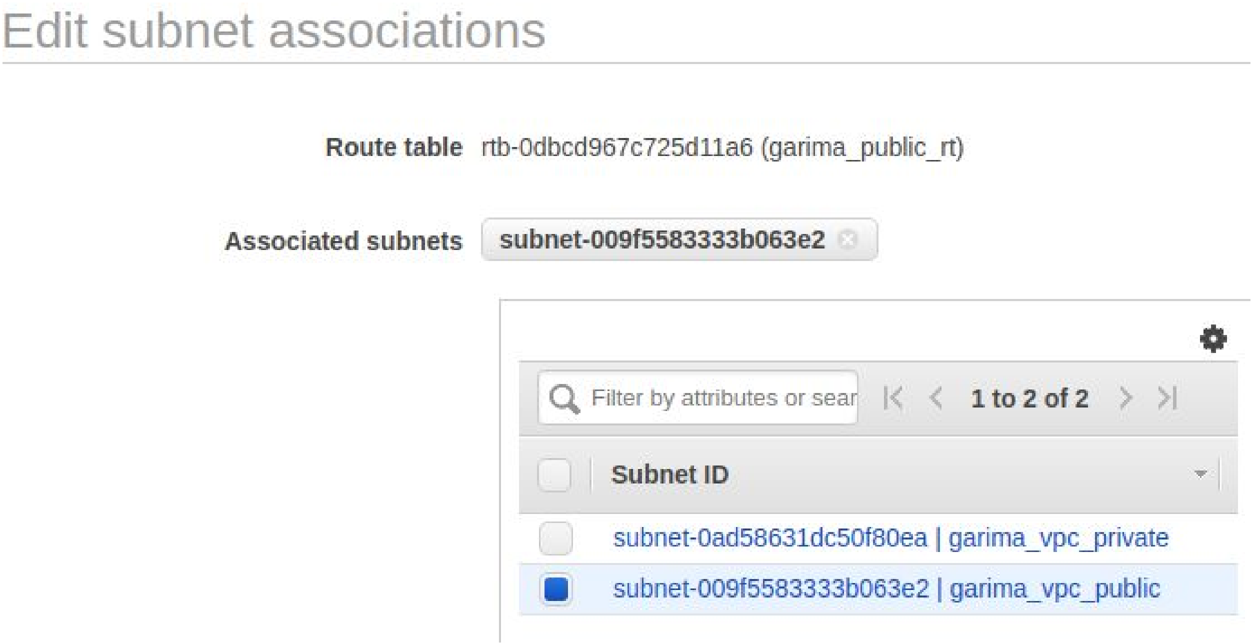
**Name tag** garima_public_rt

**VPC*** vpc-0546b7f908709adda



## Edit routes

| Destination | Target |
|---|---|
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | igw-0d530b0db6147aa6e |

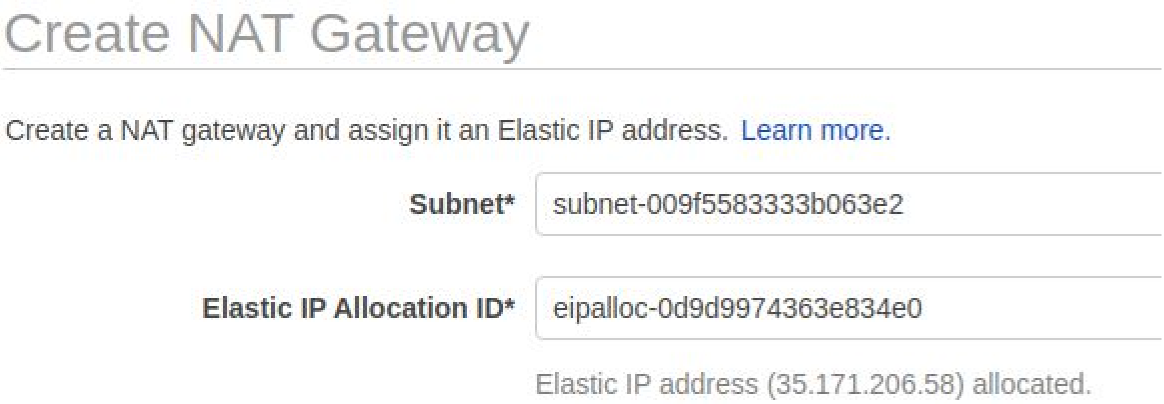


## Edit subnet associations

**Route table** rtb-0dbcd967c725d11a6 (garima_public_rt)

**Associated subnets** subnet-009f5583333b063e2

| | Subnet ID |
|---|---|
| □ | subnet-0ad58631dc50f80ea \| garima_vpc_private |
| ☑ | subnet-009f5583333b063e2 \| garima_vpc_public |



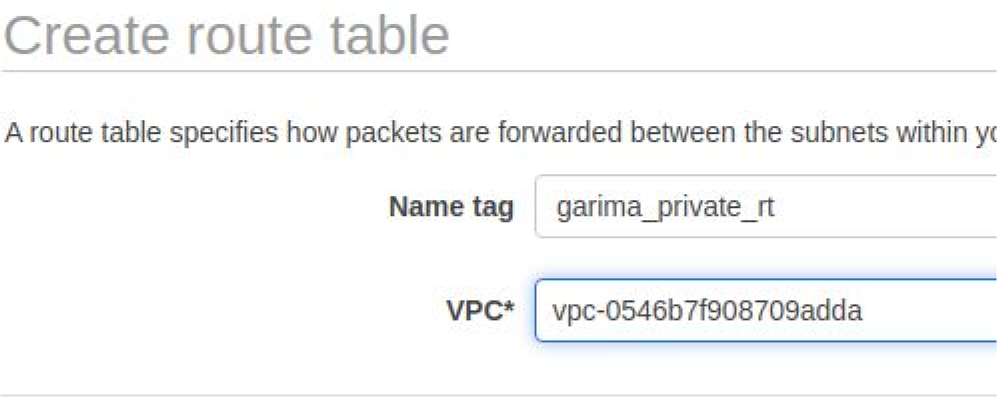
## Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. Learn more.

**Subnet*** subnet-009f5583333b063e2

**Elastic IP Allocation ID*** eipalloc-0d9d9974363e834e0

Elastic IP address (35.171.206.58) allocated.

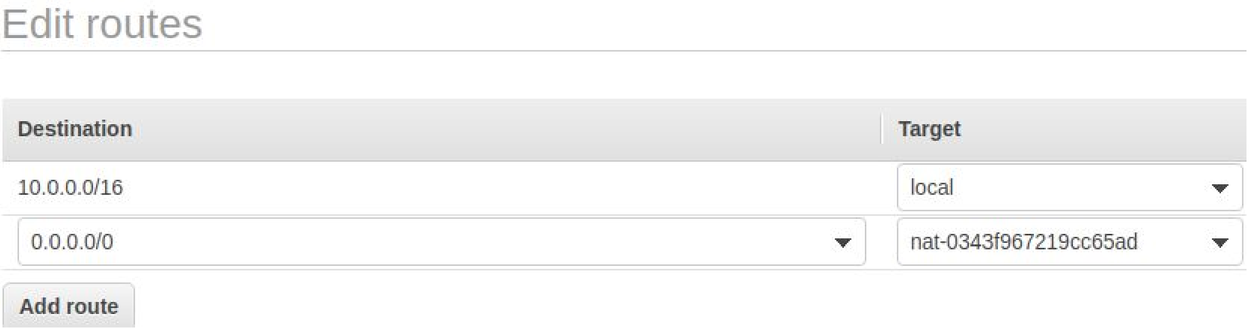


## Create route table

A route table specifies how packets are forwarded between the subnets within yo

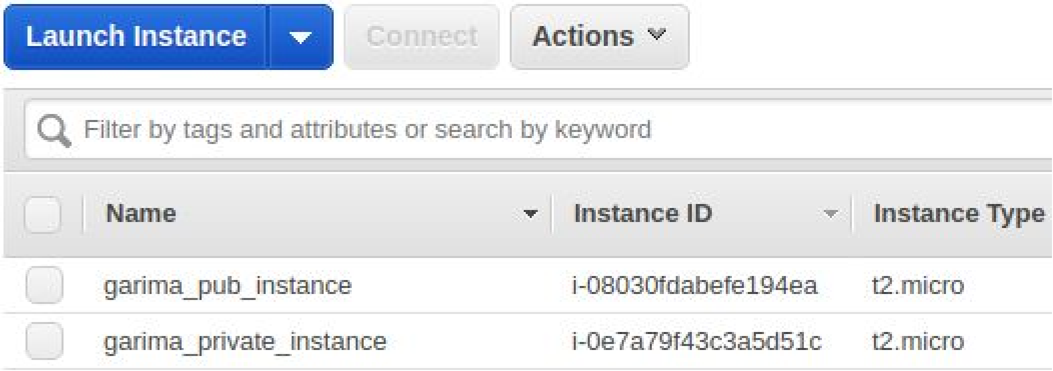
**Name tag** garima_private_rt

**VPC*** vpc-0546b7f908709adda

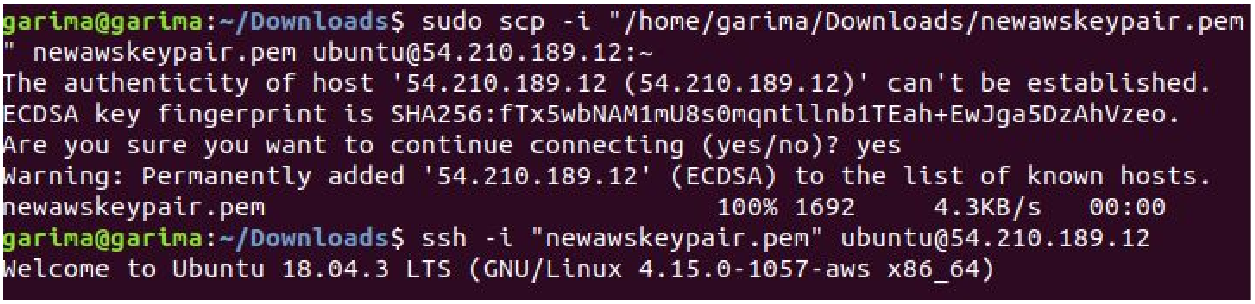


## Edit routes

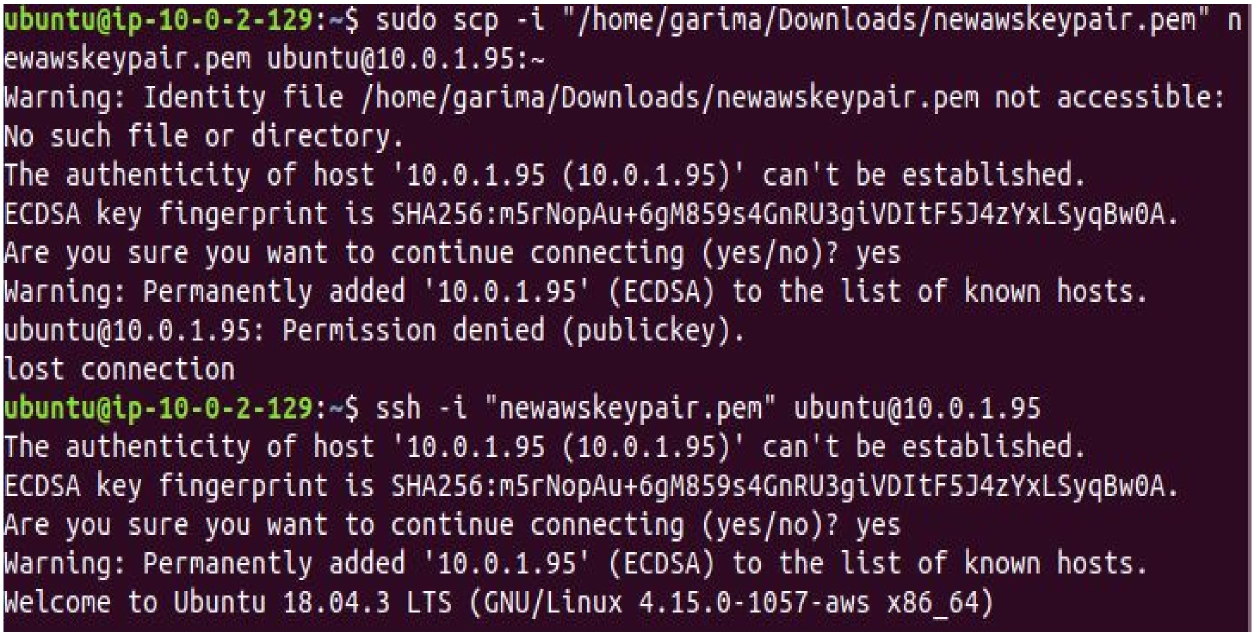
| Destination | Target |
|---|---|
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | nat-0343f967219cc65ad |



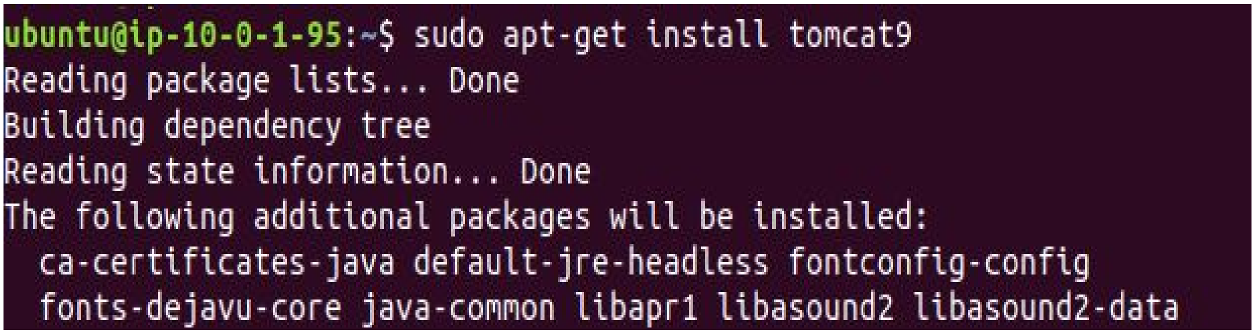
| Name | Instance ID | Instance Type |
|---|---|---|
| garima_pub_instance | i-08030fdabefe194ea | t2.micro |
| garima_private_instance | i-0e7a79f43c3a5d51c | t2.micro |



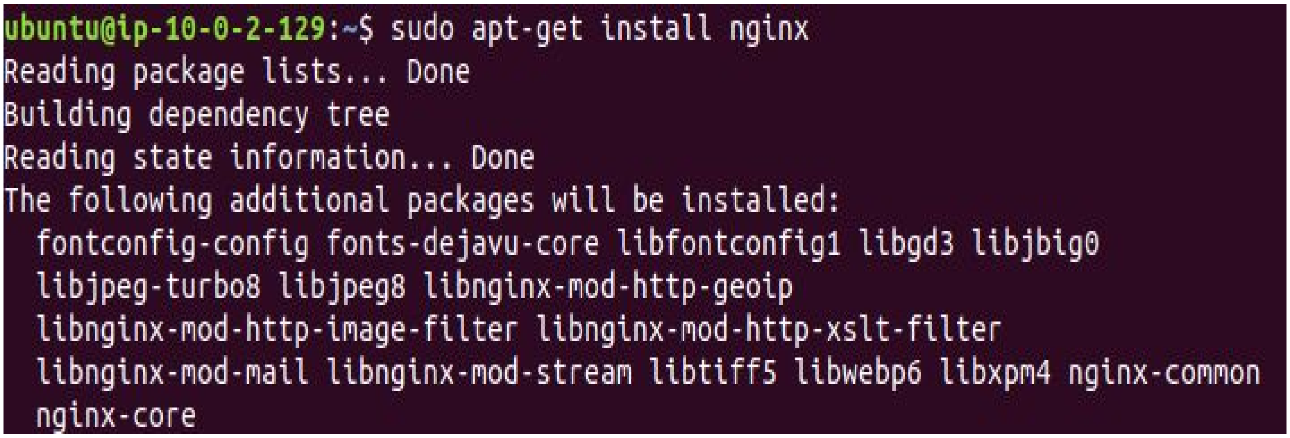
```
garima@garima:~/Downloads$ sudo scp -i "/home/garima/Downloads/newawskeypair.pem
" newawskeypair.pem ubuntu@54.210.189.12:~
The authenticity of host '54.210.189.12 (54.210.189.12)' can't be established.
ECDSA key fingerprint is SHA256:fTx5wbNAM1mU8s0mqntllnb1TEah+EwJga5DzAhVzeo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.210.189.12' (ECDSA) to the list of known hosts.
newawskeypair.pem                             100% 1692     4.3KB/s   00:00
garima@garima:~/Downloads$ ssh -i "newawskeypair.pem" ubuntu@54.210.189.12
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)
```



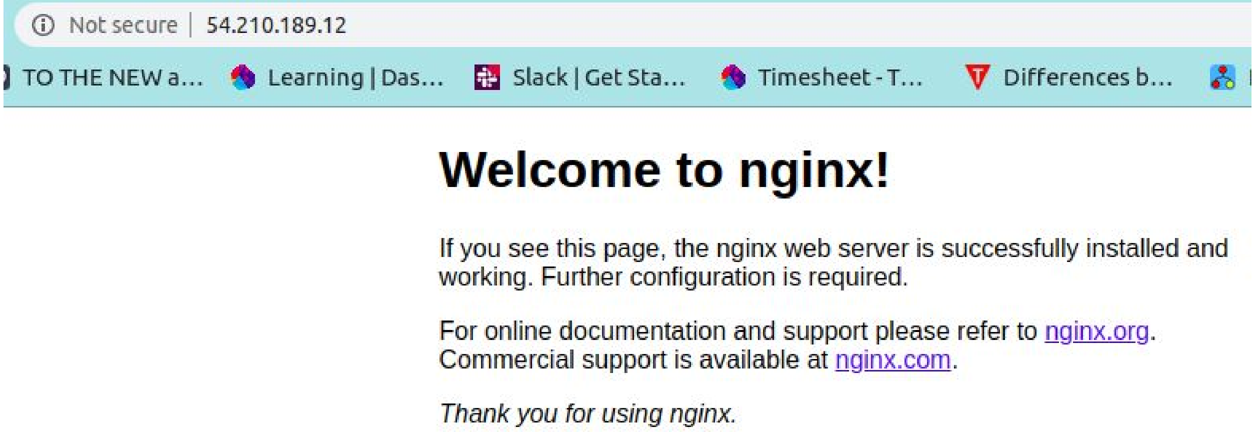
```
ubuntu@ip-10-0-2-129:~$ sudo scp -i "/home/garima/Downloads/newawskeypair.pem" n
ewawskeypair.pem ubuntu@10.0.1.95:~
Warning: Identity file /home/garima/Downloads/newawskeypair.pem not accessible:
No such file or directory.
The authenticity of host '10.0.1.95 (10.0.1.95)' can't be established.
ECDSA key fingerprint is SHA256:m5rNopAu+6gM859s4GnRU3giVDItF5J4zYxLSyqBw0A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.95' (ECDSA) to the list of known hosts.
ubuntu@10.0.1.95: Permission denied (publickey).
lost connection
ubuntu@ip-10-0-2-129:~$ ssh -i "newawskeypair.pem" ubuntu@10.0.1.95
The authenticity of host '10.0.1.95 (10.0.1.95)' can't be established.
ECDSA key fingerprint is SHA256:m5rNopAu+6gM859s4GnRU3giVDItF5J4zYxLSyqBw0A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.95' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)
```



```
ubuntu@ip-10-0-1-95:~$ sudo apt-get 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 fontconfig-config
  fonts-dejavu-core java-common libapr1 libasound2 libasound2-data
```



```
ubuntu@ip-10-0-2-129:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common
  nginx-core
```

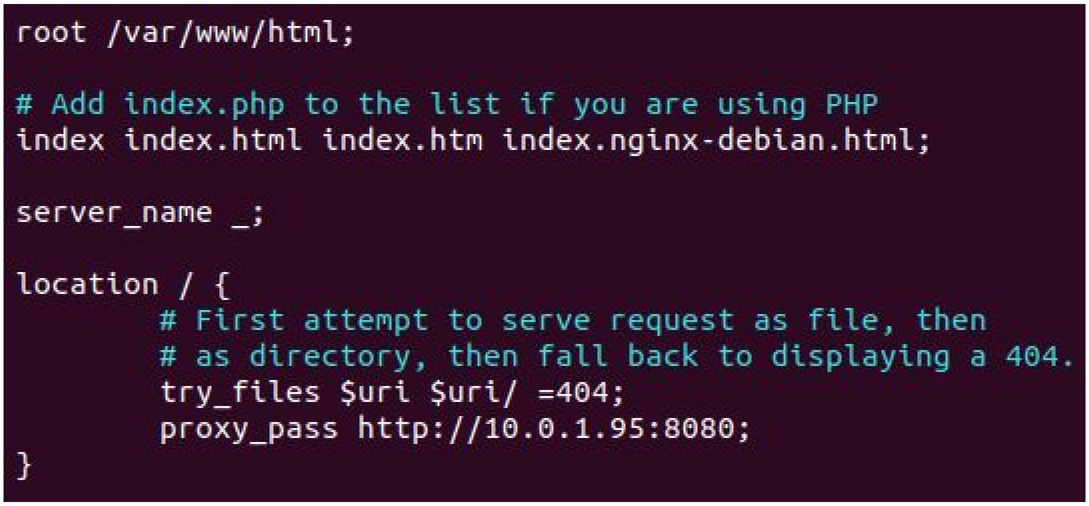


# Welcome to nginx!

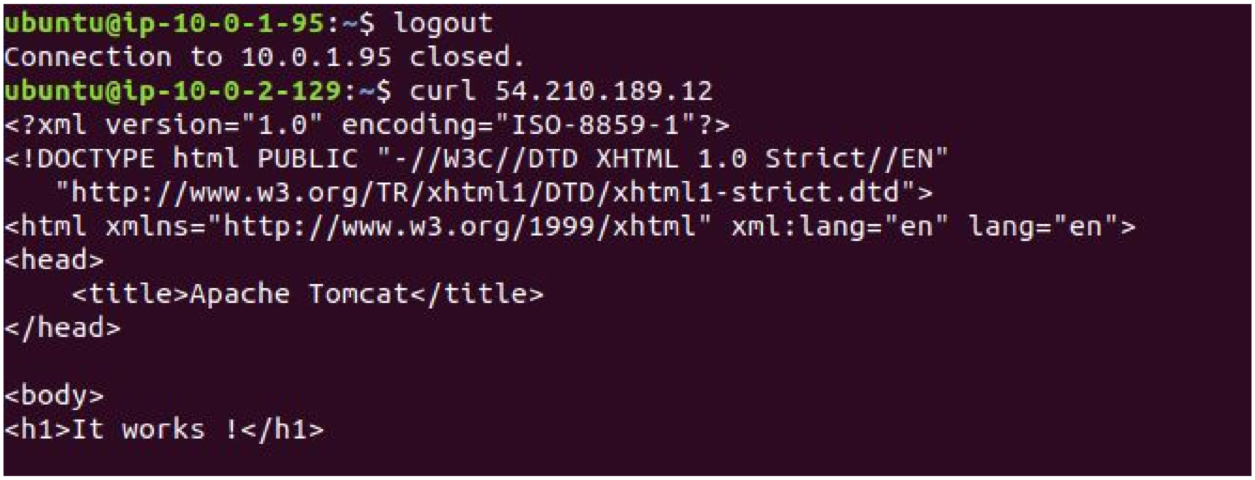
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

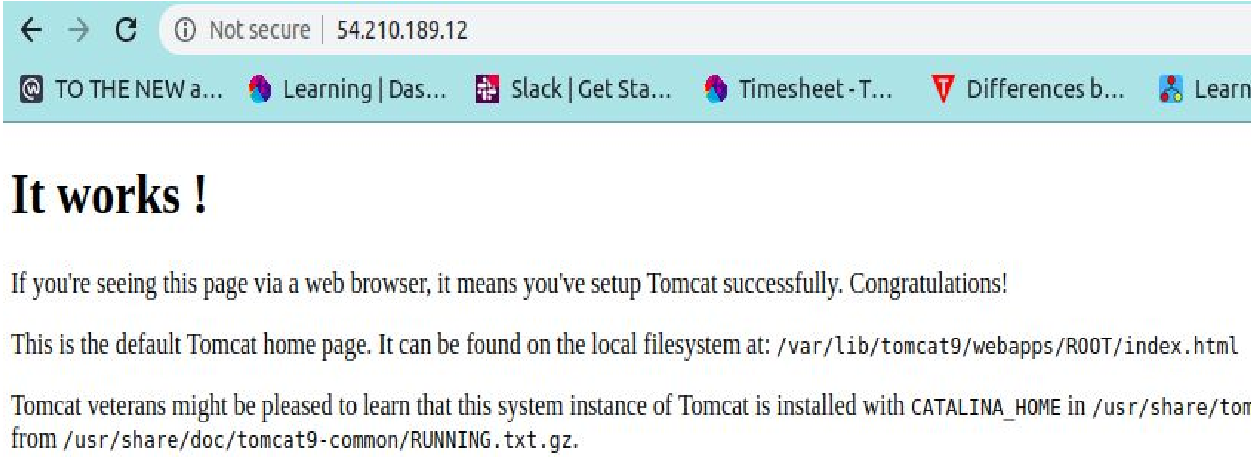
*Thank you for using nginx.*



```
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://10.0.1.95:8080;
}
```



```
ubuntu@ip-10-0-1-95:~$ logout
Connection to 10.0.1.95 closed.
ubuntu@ip-10-0-2-129:~$ curl 54.210.189.12
<?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>
```

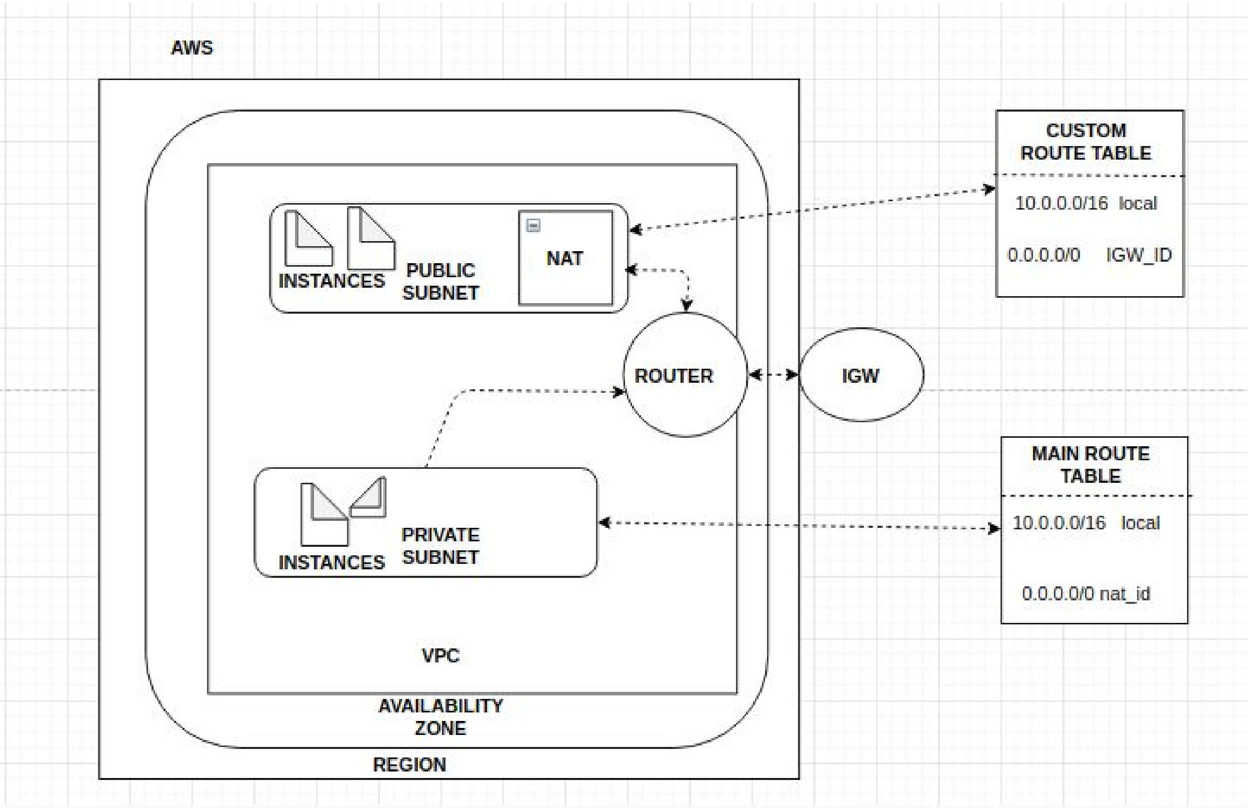


# It works !

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`
from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.



CUSTOM ROUTE TABLE

| | |
|---|---|
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | IGW_ID |

MAIN ROUTE TABLE

| | |
|---|---|
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | nat_id |