VirtualPrivateCloud

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 clouds (VPCs). Within 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)

Q 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

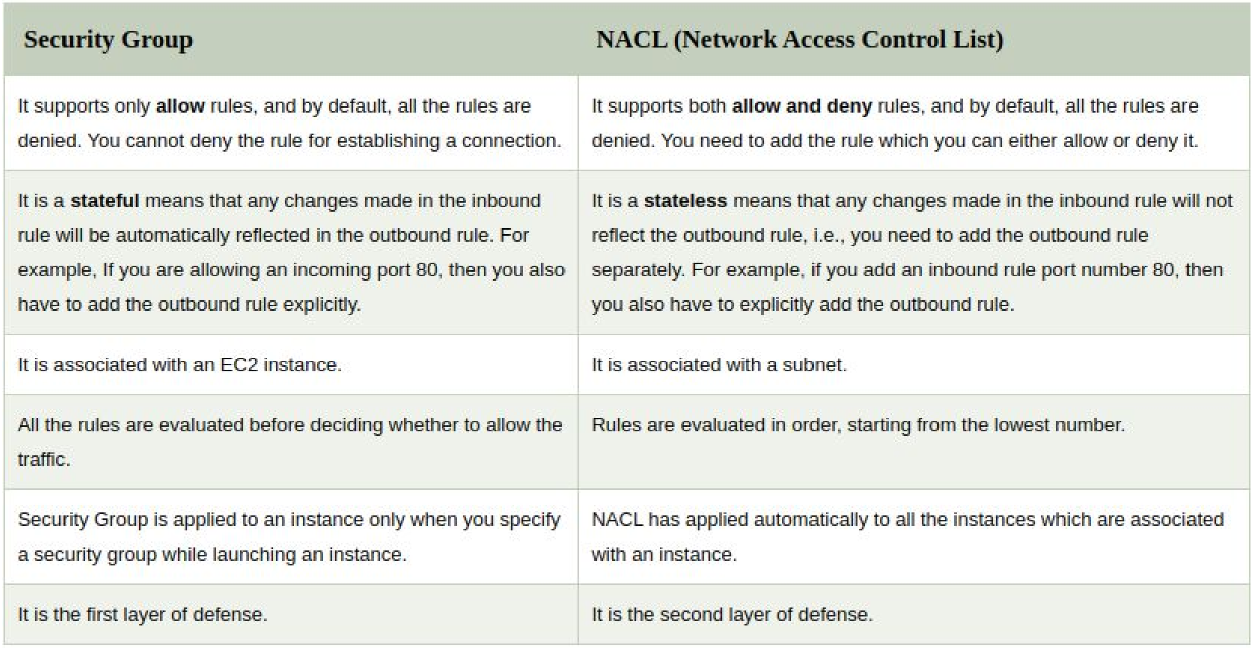
Ques 4. CIDR of a VPC is 10.0.0.0/16, if the subnet mask is /20 calculate the number of subnets that could be created from the VPC. Also find the number of IP in subnet.

Ans 4.

CIDR of a VPC is 10.0.0.0/20 THEN,
NETMASK = 255.255.240.0
NO. OF SUBNETS WILL BE = 16
NO. OF IP ADDRESS WILL BE = 4096

Ques 5. Differentiate between NACL and Security Groups.

Ans 5.



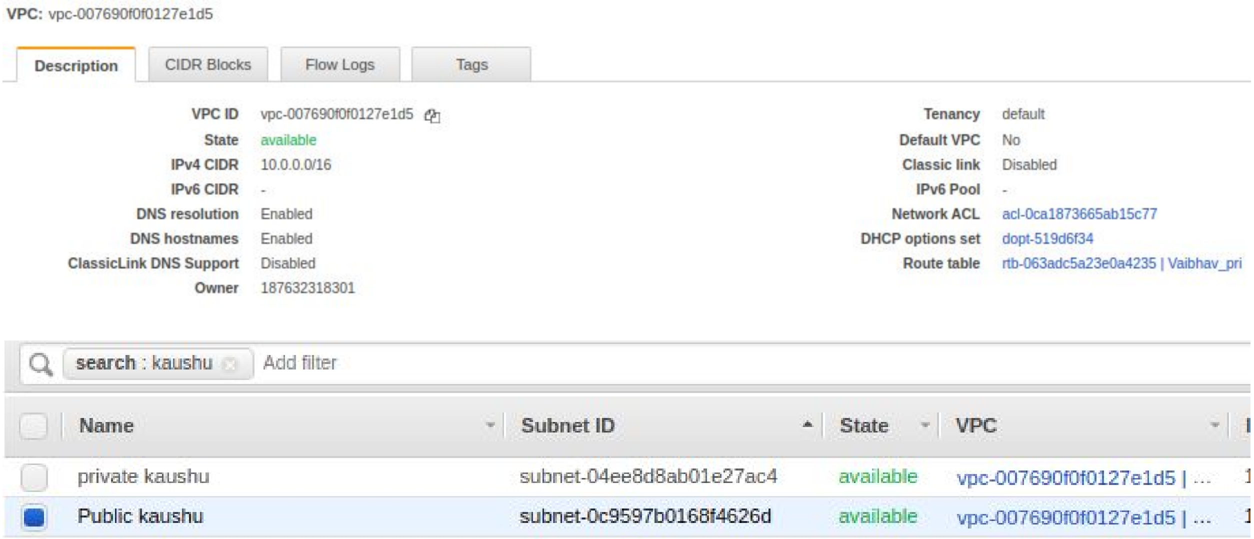
Security Group	NACL (Network Access Control List)
It supports only **allow** rules, and by default, all the rules are denied. You cannot deny the rule for establishing a connection.	It supports both **allow and deny** rules, and by default, all the rules are denied. You need to add the rule which you 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. Created a private vpc..



VPC: vpc-007690f0f0127e1d5

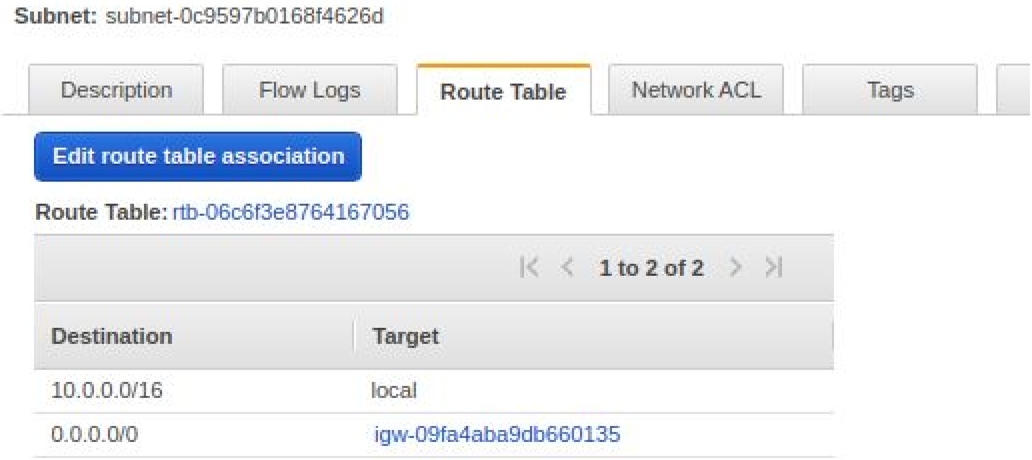
VPC ID	vpc-007690f0f0127e1d5
State	available
IPv4 CIDR	10.0.0.0/16
IPv6 CIDR	-
DNS resolution	Enabled
DNS hostnames	Enabled
ClassicLink DNS Support	Disabled
Owner	187632318301

Tenancy	default
Default VPC	No
Classic link	Disabled
IPv6 Pool	-
Network ACL	acl-0ca1873665ab15c77
DHCP options set	dopt-519d6f34
Route table	rtb-063adc5a23e0a4235 | Vaibhav_pri

search : kaushu

	Name	Subnet ID	State	VPC
□	private kaushu	subnet-04ee8d8ab01e27ac4	available	vpc-007690f0f0127e1d5 | ...
■	Public kaushu	subnet-0c9597b0168f4626d	available	vpc-007690f0f0127e1d5 | ...

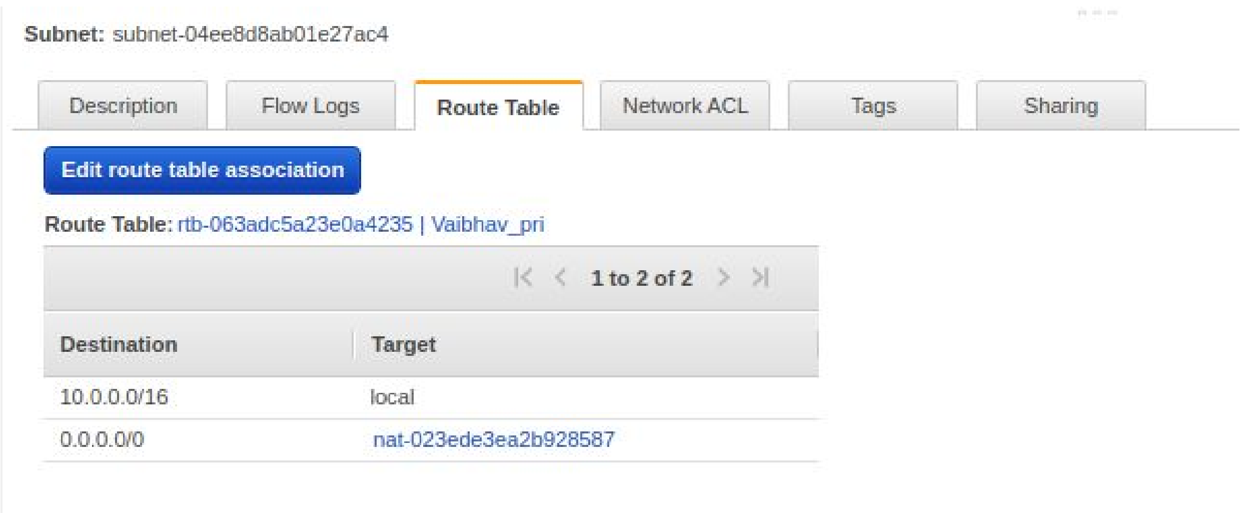
Then created a 2 subnet..
1 private
2 public



Subnet: subnet-0c9597b0168f4626d

Route Table: rtb-06c6f3e8764167056

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-09fa4aba9db660135

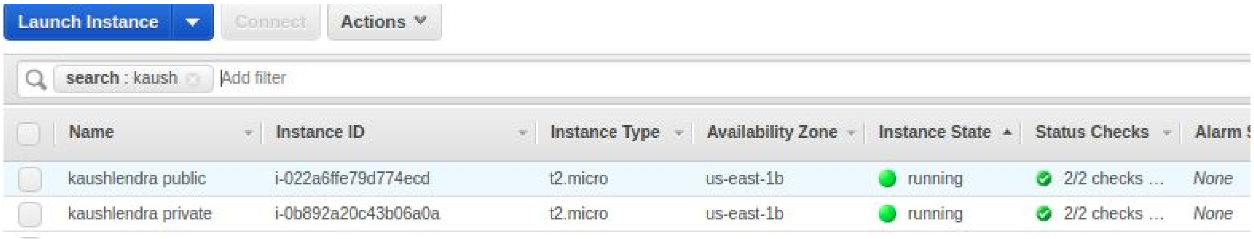


Subnet: subnet-04ee8d8ab01e27ac4

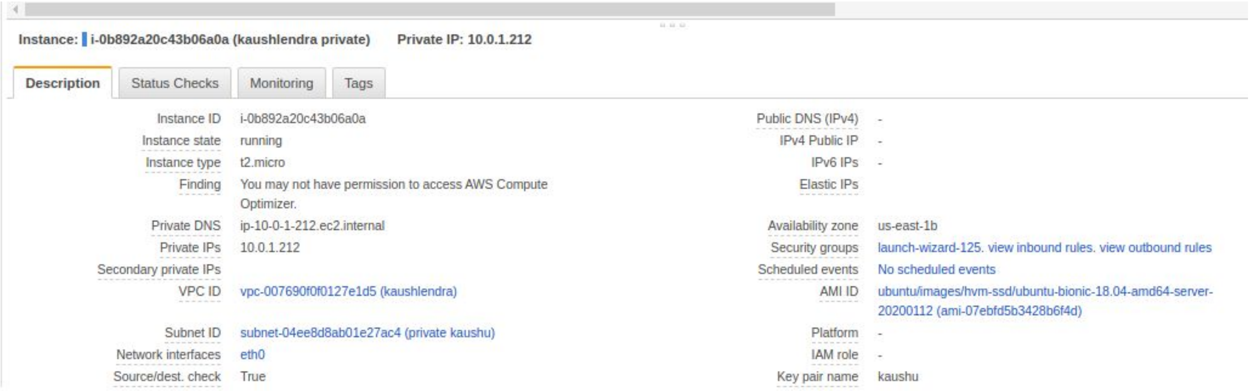
Route Table: rtb-063adc5a23e0a4235 | Vaibhav_pri

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-023ede3ea2b928587

Then launch the two instance ...

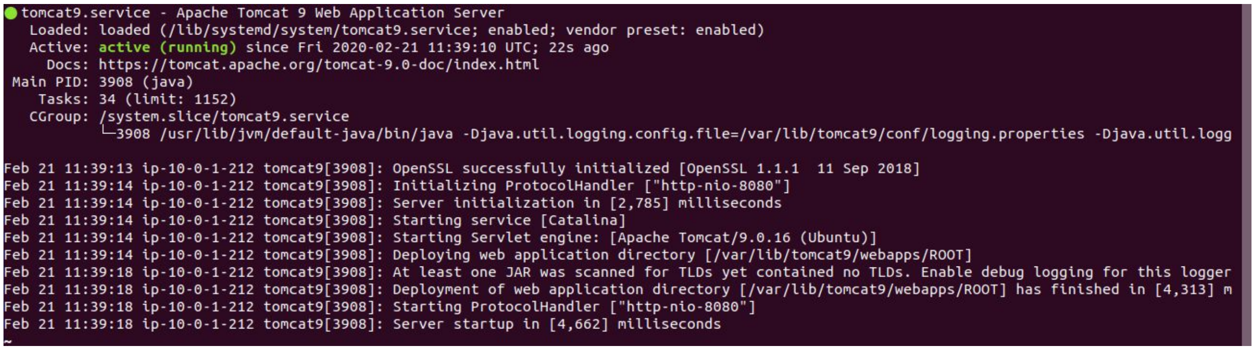


	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm S
□	kaushlendra public	i-022a6ffe79d774ecd	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None
□	kaushlendra private	i-0b892a20c43b06a0a	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None


Instance: i-0b892a20c43b06a0a (kaushlendra private)   Private IP: 10.0.1.212

Instance ID	i-0b892a20c43b06a0a	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	-
Instance type	t2.micro	IPv6 IPs	-
Finding	You may not have permission to access AWS Compute Optimizer.	Elastic IPs	
Private DNS	ip-10-0-1-212.ec2.internal	Availability zone	us-east-1b
Private IPs	10.0.1.212	Security groups	launch-wizard-125. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-007690f0f0127e1d5 (kaushlendra)	AMI ID	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112 (ami-07ebfd5b3428b6f4d)
Subnet ID	subnet-04ee8d8ab01e27ac4 (private kaushu)	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	kaushu

Installed the tomcat in private instance....

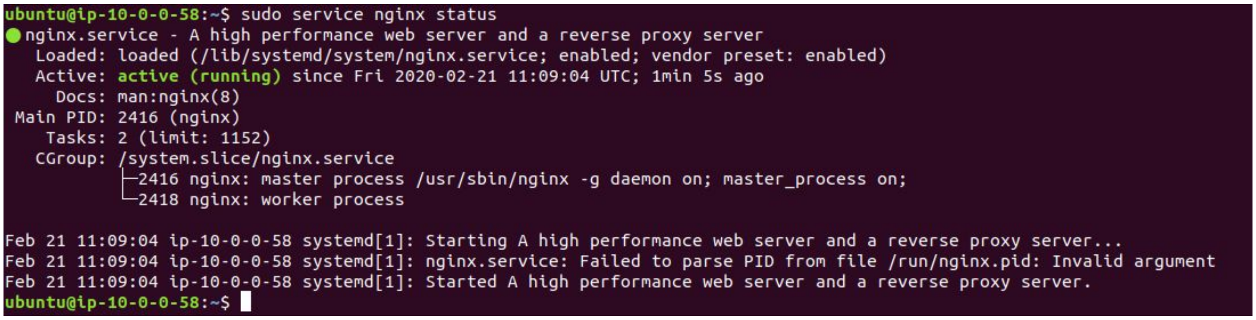

```
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-02-21 11:39:10 UTC; 22s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
 Main PID: 3908 (java)
    Tasks: 34 (limit: 1152)
   CGroup: /system.slice/tomcat9.service
           └─3908 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/conf/logging.properties -Djava.util.logg

Feb 21 11:39:13 ip-10-0-1-212 tomcat9[3908]: OpenSSL successfully initialized [OpenSSL 1.1.1  11 Sep 2018]
Feb 21 11:39:14 ip-10-0-1-212 tomcat9[3908]: Initializing ProtocolHandler ["http-nio-8080"]
Feb 21 11:39:14 ip-10-0-1-212 tomcat9[3908]: Server initialization in [2,785] milliseconds
Feb 21 11:39:14 ip-10-0-1-212 tomcat9[3908]: Starting service [Catalina]
Feb 21 11:39:14 ip-10-0-1-212 tomcat9[3908]: Starting Servlet engine: [Apache Tomcat/9.0.16 (Ubuntu)]
Feb 21 11:39:14 ip-10-0-1-212 tomcat9[3908]: Deploying web application directory [/var/lib/tomcat9/webapps/ROOT]
Feb 21 11:39:18 ip-10-0-1-212 tomcat9[3908]: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger
Feb 21 11:39:18 ip-10-0-1-212 tomcat9[3908]: Deployment of web application directory [/var/lib/tomcat9/webapps/ROOT] has finished in [4,313] m
Feb 21 11:39:18 ip-10-0-1-212 tomcat9[3908]: Starting ProtocolHandler ["http-nio-8080"]
Feb 21 11:39:18 ip-10-0-1-212 tomcat9[3908]: Server startup in [4,662] milliseconds
~
```

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.

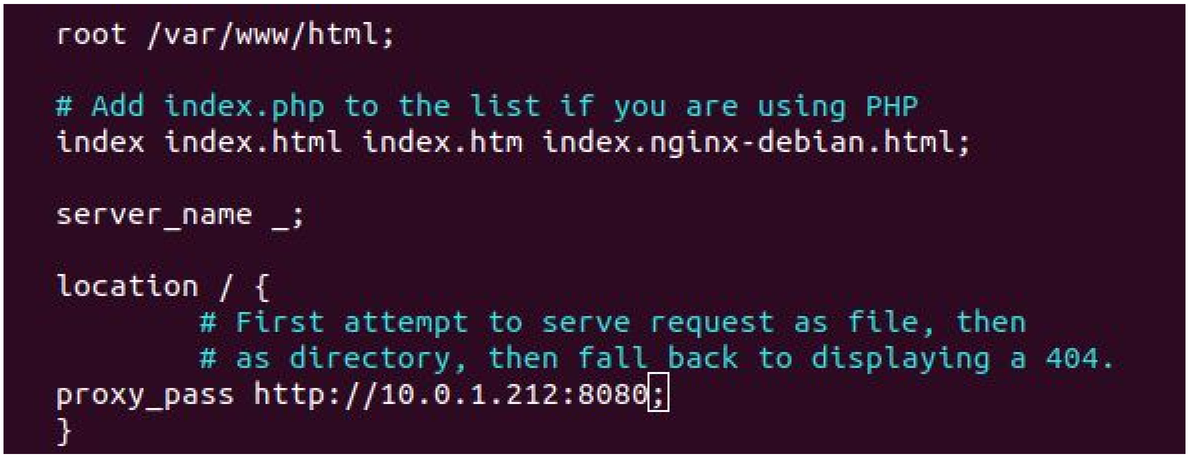
Ans.. Installed the nginx in public instance...



```
ubuntu@ip-10-0-0-58:~$ sudo service nginx status
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-02-21 11:09:04 UTC; 1min 5s ago
     Docs: man:nginx(8)
 Main PID: 2416 (nginx)
    Tasks: 2 (limit: 1152)
   CGroup: /system.slice/nginx.service
           ├─2416 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
           └─2418 nginx: worker process

Feb 21 11:09:04 ip-10-0-0-58 systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 21 11:09:04 ip-10-0-0-58 systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Feb 21 11:09:04 ip-10-0-0-58 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-10-0-0-58:~$ 
```

Proxy Pass ...



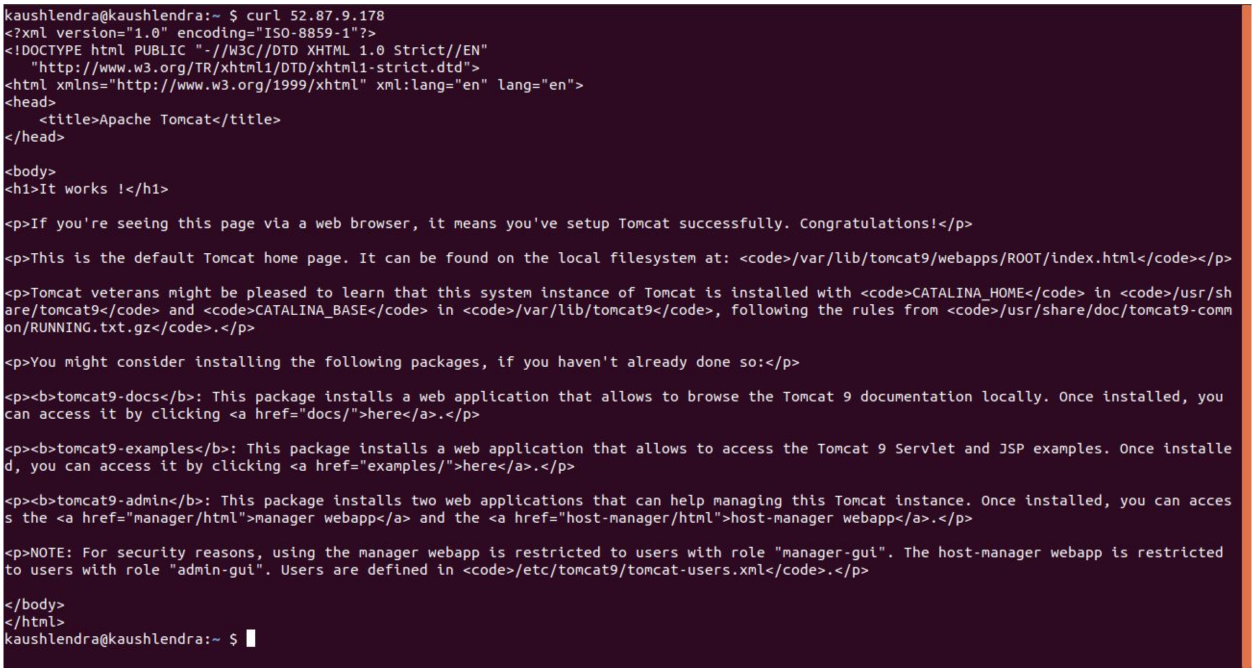
```
        root /var/www/html;

        # 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://10.0.1.212:8080;
        }
```

Proxy Pass from nginx to tomcat ...



```
kaushlendra@kaushlendra:~ $ curl 52.87.9.178
<?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>
</html>
kaushlendra@kaushlendra:~ $ 
```