

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	<i>Classful</i> 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

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

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.	<p>The same thing applies for Network ACL</p> <p>Used in running a production server</p>

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. Specify a 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 CIDR block.

Name tag ⓘ

IPv4 CIDR block* ⓘ

IPv6 CIDR block ☒ No IPv6 CIDR Block ⓘ
☐ Amazon provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

Tenancy ⓘ

* Required

VPC ID : vpc-0c297a0d16cde7bbd ⓘ Add filter						
<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set
<input checked="" type="checkbox"/>	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 ⓘ

* Required

[Cancel](#)

Attach igw to vpc

Attach to VPC

Attach an internet gateway to a VPC to enable communication with the internet. Specify the VPC you would like to attach below.

VPC* ⓘ

▶ AWS Command Line Interface command

* Required

Public subnet

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 netmask and /28 netmas
CIDR block must be a /64 CIDR block.

Name tag ⓘ

VPC* ⓘ

Availability Zone ⓘ

VPC CIDRs	CIDR	Status	Sta
	10.0.0.0/16	associated	

IPv4 CIDR block* ⓘ

* Required

Private subnet

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 netmask and /28 CIDR block must be a /64 CIDR block.

Name tag	<input type="text" value="maithely_pri"/>					
VPC*	<input type="text" value="vpc-0c297a0d16cde7bbd"/>					
Availability Zone	<input type="text" value="No preference"/>					
VPC CIDRs	<table><thead><tr><th>CIDR</th><th>Status</th></tr></thead><tbody><tr><td>10.0.0.0/16</td><td>associated</td></tr></tbody></table>		CIDR	Status	10.0.0.0/16	associated
CIDR	Status					
10.0.0.0/16	associated					
IPv4 CIDR block*	<input type="text" value="10.0.2.0/24"/>					

Now create NAT in public subnet

[NAT Gateways](#) > Create NAT Gateway

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

Subnet*	<input type="text" value="subnet-0756cb7b603a98c42"/>		
Elastic IP Allocation ID*	<input type="text" value="eipalloc-059d31fb2a1aef288"/>		<input type="button" value="Allocate Elastic IP address"/>

* Required Cancel

Now create route table :private

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag	<input type="text" value="maithely-prt-rt"/>	
VPC*	<input type="text" value="vpc-0c297a0d16cde7bbd"/>	

* Required

Create public rt

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag ⓘ

VPC* ↕ ⓘ

* Required

Now add nat in route table of private

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	active	No	
0.0.0.0/0	nat-0e2d40f828c96722d		No	✕

Add route

* Required

Cancel Save routes

Now add igw to public rt

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	active	No	
0.0.0.0/0	igw-040edb73e38f5b909		No	✕

Add route

* Required

Cancel Save routes

Create an instance in private subnet

Network ⓘ ↕ [Create new VPC](#)

Subnet ⓘ ↕ [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP ⓘ ↕

accessible ⓘ Enabled ↕

Instance version ⓘ V1 and V2 (token optional) ↕

Maximum hop limit ⓘ 1 ↕

User data ⓘ ☒ As text ☐ As file ☐ Input is already base64 encoded

```
#!/bin/bash
apt-get update -y
apt-get install tomcat -y
```

Now create one public instance

Purchasing option ⓘ ☐ Request Spot instances

Network ⓘ vpc-0c297a0d16cde7bbd | maithely ↕ [Create new VPC](#)

Subnet ⓘ subnet-0756cb7b603a98c42 | maithely_pub | us-east ↕ [Create new subnet](#)
250 IP Addresses available

Auto-assign Public IP ⓘ Enable ↕

Placement group ⓘ ☐ Add instance to placement group

Capacity Reservation ⓘ Open ↕ [Create new Capacity Reservation](#)

Public inbound

Edit inbound rules
×

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
HTTP ▾	TCP	80	Custom ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Private inbound

Edit inbound rules

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

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

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 [84
```

```
ubuntu@ip-10-0-2-207:~$ sudo service tomcat9 start
ubuntu@ip-10-0-2-207:~$ sudo service tomcat9 status
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor
   Active: active (running) since Tue 2020-02-25 10:46:30 UTC; 53s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Main PID: 15376 (java)
     Tasks: 34 (limit: 1152)
    CGroup: /system.slice/tomcat9.service
            └─15376 /usr/lib/jvm/default-java/bin/java -Djava.util.logging

Feb 25 10:46:33 ip-10-0-2-207 tomcat9[15376]: OpenSSL successfully initi
Feb 25 10:46:33 ip-10-0-2-207 tomcat9[15376]: Initializing ProtocolHandl
Feb 25 10:46:33 ip-10-0-2-207 tomcat9[15376]: Server initialization in [
Feb 25 10:46:34 ip-10-0-2-207 tomcat9[15376]: Starting service [Catalina
Feb 25 10:46:34 ip-10-0-2-207 tomcat9[15376]: Starting Servlet engine: [
Feb 25 10:46:34 ip-10-0-2-207 tomcat9[15376]: Deploying web application
Feb 25 10:46:38 ip-10-0-2-207 tomcat9[15376]: At least one JAR was scanr
Feb 25 10:46:38 ip-10-0-2-207 tomcat9[15376]: Deployment of web applicat
Feb 25 10:46:38 ip-10-0-2-207 tomcat9[15376]: Starting ProtocolHandler [
Feb 25 10:46:38 ip-10-0-2-207 tomcat9[15376]: Server startup in [4,758]
lines 1-19/19 (END)
```

In public instance install nginx


```
ubuntu@ip-10-0-1-246:~$ 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 lib
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 lib
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  nginx-common nginx-core
0 upgraded, 18 newly installed, 0 to remove and 50 not upgraded
Need to get 2461 kB of archives.
After this operation, 8210 kB of additional disk space will be
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-upd
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-upd
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mai
```

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 Tomcat successfully. Congratulations!

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

<p>Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with CATALINA_HOME in /usr/share/tomcat9
from /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, you can access the documentation at http://localhost:8080/docs/.

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access the examples at http://localhost:8080/examples/.

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp at http://localhost:8080/manager/html/ and the host-manager webapp at http://localhost:8080/host-manager/html/.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is not restricted.

For more information on the Tomcat 9 installation, see the Tomcat 9 documentation at http://tomcat.apache.org/tomcat-9.0/faq.html.

</p>

```

```

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
nginx: signal 15 received

```

Now write private ip in the browser

←
→
↻
ⓘ Not secure | 3.92.209.206

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/tomcat9 from /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, you can access the documentation at http://localhost:8080/docs/.

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access the examples at http://localhost:8080/examples/.

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp at http://localhost:8080/manager/html/ and the host-manager webapp at http://localhost:8080/host-manager/html/.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is not restricted.

For more information on the Tomcat 9 installation, see the Tomcat 9 documentation at http://tomcat.apache.org/tomcat-9.0/faq.html.

