

## VPC:

Q1. When to use Elastic IP over Public IP

Public IP	Elastic IP
It is assigned to your launched instance.	It is assigned to your AWS account.
when an instance is terminated the public IP attached to it gets released and further when you relaunch the same instance new IP address is assigned.	Elastic IP does not change and they remain the same even if you terminate the instance and later again restart the same instance.

- **Use case:**

1. Elastic IP is used when you are working on a long time project and configuration of IP sometimes consumes more time.
2. 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 short time.

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

### Valid IP ranges of LAN:

- The first three octets of an IP address should be the same for all computers in the LAN. For example, if a total of 128 hosts exist in a single LAN, the IP addresses could be assigned starting with 192.168.1. x, where x represents a

number in the range of 1 to 128. You could create consecutive LANs within the same company in a similar manner consisting of up to another 128 computers.

- There are different classes of networks that determine the size and total possible unique IP addresses of any given LAN. For example, a class A LAN can have over 16 million unique IP addresses. A class B LAN can have over 65,000 unique IP addresses. The size of your LAN depends on which reserved address range you use and the subnet mask (explained later in the article) associated with that range

Address range	Subnet mask	Provides	Addresses per LAN
10.0.0.0 – 10.255.255.255	255.0.0.0	1 class A LAN	16,777,216
172.16.0.0 – 172.31.255.255	255.255.0.0	16 class B LANs	65,536
192.168.0.0 – 192.168.255.255	25.255.255.0	256 class C LANs	256

#### Implications:

We can use every IP-Address-Range we want in our private network. There is nothing against this. But we have to take precautions to avoid routing-trouble when a machine with an IP-Address that actually belongs to a public range wants to access the internet. Here you have to have a Router or Firewall that is able to NAT your internal address bidirectionally.

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

- **Cannot create a VPC peering connection between VPCs in different regions. This may change in future releases.**
- **VPC soft limit of 50 peering connection applies.**
- **VPC peering connections created between VPCs that have overlapping subnet CIDR blocks may not take effect.**
- **You cannot have more than one VPC peering connection between the same two VPCs at the same time.**
- **You cannot create a VPC peering connection between VPCs in different regions.**
- **After a VPC peering connection is established, the local and peer tenants must add routes in the local and peer VPCs to enable communication between the two VPCs.**

- You cannot delete a VPC for which VPC peering connection routes have been configured.

Q4. Differentiate between NACL and Security Groups.

**In terms of application:**

Security groups are tied to an instance whereas Network ACLs are tied to the subnet. i.e. Network Access control lists are applicable at the subnet level, so any instance in the subnet with an associated NACL will follow rules of NACL. That's not the case with security groups, security groups have to be assigned explicitly to the instance. This means any instances within the subnet group get the rule applied.

**In terms of state:**

Security groups are stateful. Network ACLs are stateless: This means any changes applied to an incoming rule will not be applied to the outgoing rule.

**In terms of rules:**

All rules in a security group are applied whereas rules are applied in their order (the rule with the lower number gets processed first) in Network ACL.

Security group first layer of defense, whereas Network ACL is second layer of the defense.

Q5. 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.

/20 means first 20 bits of the IP address are network bits and the rest are host bits

Number of hosts bits =  $32 - 20 = 12$

Number of ip's possible =  $2^{\text{host bits} - 2} = 2^{12} - 2$

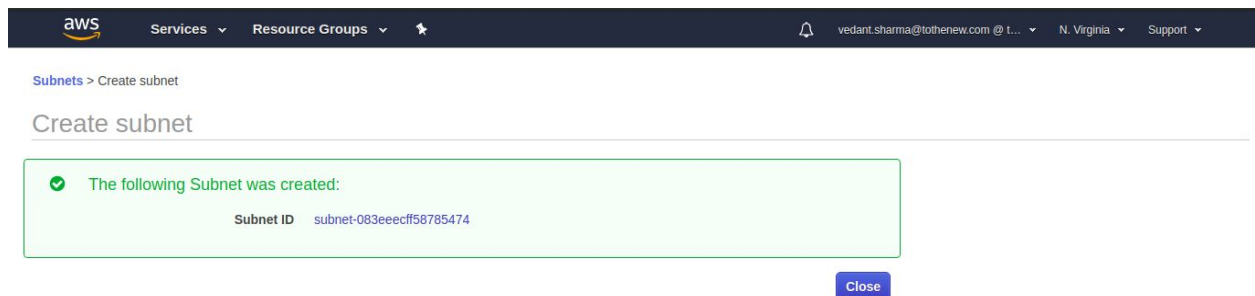
Number of subnets possible =  $2^4 = 16$

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

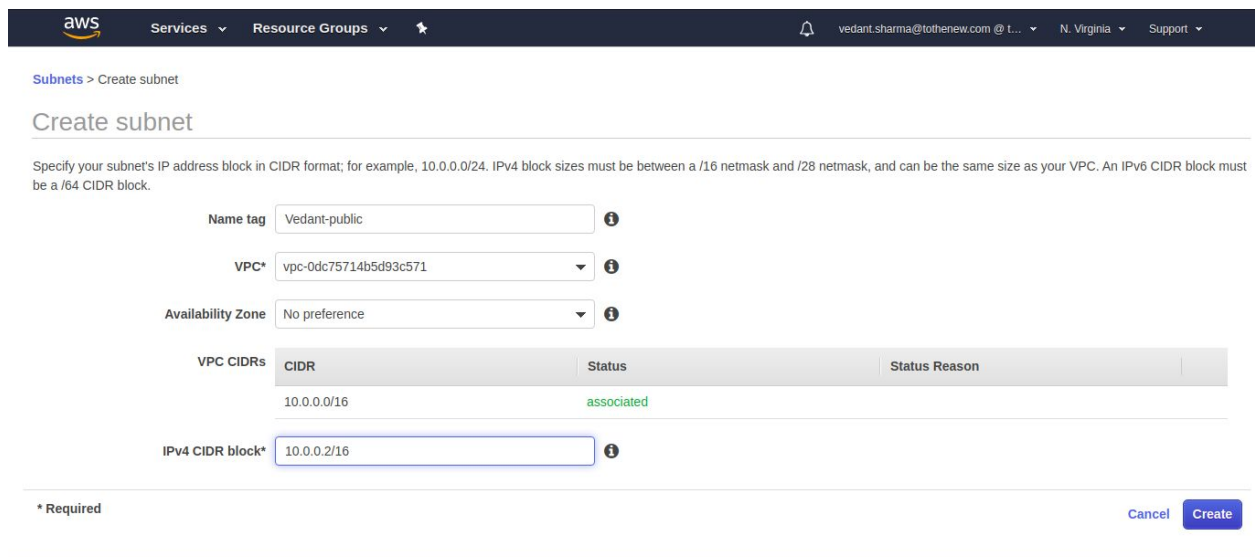
1. Create a private subnet, attach NAT, and host an application server(Tomcat)
2. Create a public subnet, and host a web server(Nginx), also proxypass to Tomcat from Nginx

Creating a subnet:

Private subnet



The screenshot shows the AWS console 'Create subnet' page. A green confirmation box states: 'The following Subnet was created: Subnet ID subnet-083eeecff58785474'. A 'Close' button is located at the bottom right of the confirmation box.

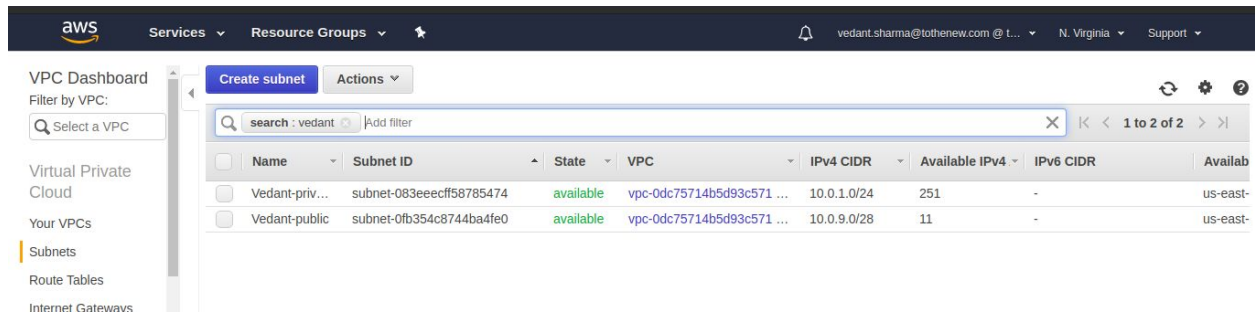


The screenshot shows the AWS console 'Create subnet' form. The fields are filled as follows:

- Name tag: Vedant-public
- VPC\*: vpc-0dc75714b5d93c571
- Availability Zone: No preference
- VPC CIDRs: A table with 3 columns: CIDR, Status, and Status Reason. The first row shows CIDR 10.0.0.0/16 with Status 'associated'.
- IPv4 CIDR block\*: 10.0.0.2/16

At the bottom, there is a '\* Required' label and 'Cancel' and 'Create' buttons.

## Sub-nets



The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with navigation links: VPC Dashboard, Filter by VPC, Virtual Private Cloud, Your VPCs, Subnets (highlighted), Route Tables, and Internet Gateways. The main area has a 'Create subnet' button and an 'Actions' dropdown. Below these is a search bar with 'search : vedant' and an 'Add filter' button. A table lists subnets with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, and Availability Zone. Two subnets are listed: 'Vedant-priv...' and 'Vedant-public', both in 'available' state.

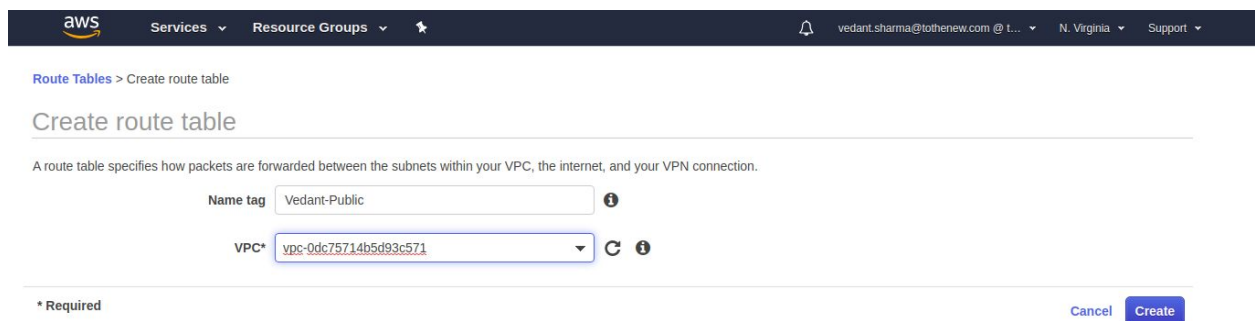
Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
Vedant-priv...	subnet-083eeecff58785474	available	vpc-0dc75714b5d93c571 ...	10.0.1.0/24	251	-	us-east-
Vedant-public	subnet-0fb354c8744ba4fe0	available	vpc-0dc75714b5d93c571 ...	10.0.9.0/28	11	-	us-east-

## Creating a internet gateway



The screenshot shows the 'Create internet gateway' page in the AWS console. It includes a breadcrumb 'Internet gateways > Create internet gateway' and a title 'Create internet gateway'. A description states: '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.' There is a 'Name tag' input field with the value 'vedant-IGW'. At the bottom, there's a '\* Required' label and two buttons: 'Cancel' and 'Create'.

## Creating route tables:



The screenshot shows the 'Create route table' page in the AWS console. It includes a breadcrumb 'Route Tables > Create route table' and a title 'Create route table'. A description states: 'A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.' There are two input fields: 'Name tag' with the value 'Vedant-Public' and 'VPC\*' with a dropdown menu showing 'vpc-0dc75714b5d93c571'. At the bottom, there's a '\* Required' label and two buttons: 'Cancel' and 'Create'.

**aws** Services Resource Groups

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet

Create route table Actions

search: vedant Add filter

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Own
Vedant-priv...	rtb-00e18f2419ea96f58	-	-	No	vpc-0dc75714b5d93c571 ...	1876
Vedant-Public	rtb-0515a84a0bb4dbf8e	-	-	No	vpc-0dc75714b5d93c571 ...	1876

Adding internet gateway to public route table:

**aws** Services Resource Groups

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-04a7d2d8e6cc8744d		No

Add route

\* Required

Cancel Save routes

Subnet association:

**aws** Services Resource Groups

Route table rtb-0515a84a0bb4dbf8e (Vedant-Public)

Associated subnets subnet-0fb354c8744ba4fe0

Filter by attributes or search by keyword

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0c642f7a4e8504d42   sarthak-pu...	10.0.4.0/28	-	rtb-0e054e348e944e9e8
subnet-083eeecff58785474   Vedant-priv...	10.0.1.0/24	-	Main
subnet-0ef2380d13e656346   sarthak-pri...	10.0.0.0/24	-	rtb-06166de8de244b793
subnet-022cabd877f3e421b   Srma-pub	10.0.6.0/28	-	rtb-0c1b96f67e23f6940
subnet-0fb354c8744ba4fe0   Vedant-pub...	10.0.9.0/28	-	Main
subnet-04f1674698b6e63f7   Srma-priv	10.0.5.0/28	-	rtb-041f2ca32602129cb

\* Required

Cancel Save

aws Services Resource Groups

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Create route table Actions

search: vedant Add filter

1 to 2 of 2

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Own
Vedant-priv...	rtb-00e18f2419ea96f58	subnet-083eeecff58785474	-	No	vpc-0dc75714b5d93c571 ...	1876
Vedant-Public	rtb-0515a84a0bb4dbf8e	subnet-0fb354c8744ba4fe0	-	No	vpc-0dc75714b5d93c571 ...	1876

Route Table: rtb-00e18f2419ea96f58

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-083eeecff5878547...	10.0.1.0/24	-

## Creating a NAT gateway in public subnet

aws Services Resource Groups

NAT Gateways > Create NAT Gateway

### Create NAT Gateway

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

Subnet\* subnet-0fb354c8744ba4fe0

Elastic IP Allocation ID\* Enter an allocation ID or select an EIP

Allocate Elastic IP address

\* Required

Cancel Create a NAT Gateway

aws Services Resource Groups

NAT Gateways > Create NAT Gateway

### Create NAT Gateway

✓ Your NAT gateway has been created.

Note: In order to use your NAT gateway, ensure that you [edit your route tables](#) to include a route with the following NAT gateway. [Find out more.](#)

NAT Gateway ID nat-0a57e11f9833e710f

Edit route tables Close

## Associating NAT gateway to private route table

Services
Resource Groups

vedant.sharma@tothenew.com @ t...
N. Virginia
Support

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-06cb1052b630a9512		No

Add route

\* Required

Cancel
Save routes

## Launching an instance in public subnet

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

☒ Create a new security group
☐ Select an existing security group

Security group name: launch-wizard-157

Description: launch-wizard-157 created 2020-02-24T16:34:35.494+05:30

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	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Cancel
Previous
Review and Launch

## Installing tomcat using user data



aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

[Add the system](#) [Create new the system](#)

▼ Network interfaces ⓘ

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-083eeecf	Auto-assign	Add IP	Add IP

[Add Device](#)

▼ Advanced Details

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

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install tomcat9 -y
```

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

SSH into the public instance to install nginx

```
vedant@Vedant-Sharma:~/Desktop$ sudo ssh -i Vedant-Bootcamp.pem ubuntu@3.92.222.193
[sudo] password for vedant:
The authenticity of host '3.92.222.193 (3.92.222.193)' can't be established.
ECDSA key fingerprint is SHA256:Y9/81499bQw754LZtUBZS1XT+KnPKvf5rLaiWMJEB00.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '3.92.222.193' (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 Mon Feb 24 11:17:43 UTC 2020

System load:  0.0               Processes:    86
Usage of /:   13.6% of 7.69GB   Users logged in:  0
Memory usage: 14%              IP address for eth0: 10.0.9.10
Swap usage:   0%

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-10-0-9-10:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
ubuntu@ip-10-0-9-10:~$ 
ubuntu@ip-10-0-9-10:~$ 
ubuntu@ip-10-0-9-10:~$ sudo systemctl start nginx
ubuntu@ip-10-0-9-10:~$ sudo systemctl status nginx
● 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 Mon 2020-02-24 11:20:06 UTC; 1min 22s ago
     Docs: man:nginx(8)
  Main PID: 2326 (nginx)
    Tasks: 2 (limit: 1152)
   CGroup: /system.slice/nginx.service
           └─2326 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              └─2328 nginx: worker process

Feb 24 11:20:06 ip-10-0-9-10 systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 24 11:20:06 ip-10-0-9-10 systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Feb 24 11:20:06 ip-10-0-9-10 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-10-0-9-10:~$ 

```

Proxy pass:

```

ubuntu@ip-10-0-9-10: /etc/nginx/sites-available 150x38
server {
    listen 80;
    Server_name test.com

    root /var/www/html;
    index index.html;

    location / {
        proxy_pass http://10.0.1.129:8080
    }
}

```

Entry in /etc/hosts

```

ubuntu@ip-10-0-9-10: ~ 150x38
127.0.0.1 localhost test.com

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

```

Curl into test.com it will proxy pass to the apache server running on private subnet

```
ubuntu@ip-10-0-9-10:~$  
ubuntu@ip-10-0-9-10:~$ curl test.com  
<?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/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>  
  
<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 installed, 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 access 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>  
ubuntu@ip-10-0-9-10:~$
```