

1. When to use Elastic IP over Public IP

Public ips are allotted by AWS to all the “auto-assign ip” enabled instances , but whenever the instances are stopped and started, a new ip is given. However, if we wish to make our instance permanently available on a single ip (for long term projects) we should use elastic ips. Elastic ips are specially allocated ips by aws which is given to you only and it won't be taken away even if the instance is stopped and started again.

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

There are three ranges of addresses that can be used in a private network:

- 10.0. 0.0 – 10.255. 255.255.
- 172.16. 0.0 – 172.31. 255.255.
- 192.168. 0.0 – 192.168. 255.255.

If you use public addresses on your private network, t

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

1. Transitive VPC peering is not allowed. If VPC A is peered VPC B and VPC B is peered with VPC C then it doesn't mean that VPC A is peered with VPC C, you need to explicitly peer them.
2. VPC's with overlapping IP's can't be peered.
3. It is not possible to create a VPC peering connection between VPCs present in different regions.
4. Only one VPC peering connection is possible between two VPCs at a time
5. In case of following VPC peering connections, it is not allowed to extend the peering

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.

Subnet mask of /20 means first 20 bits of the ip address are network bits and the rest are host bits

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

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

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

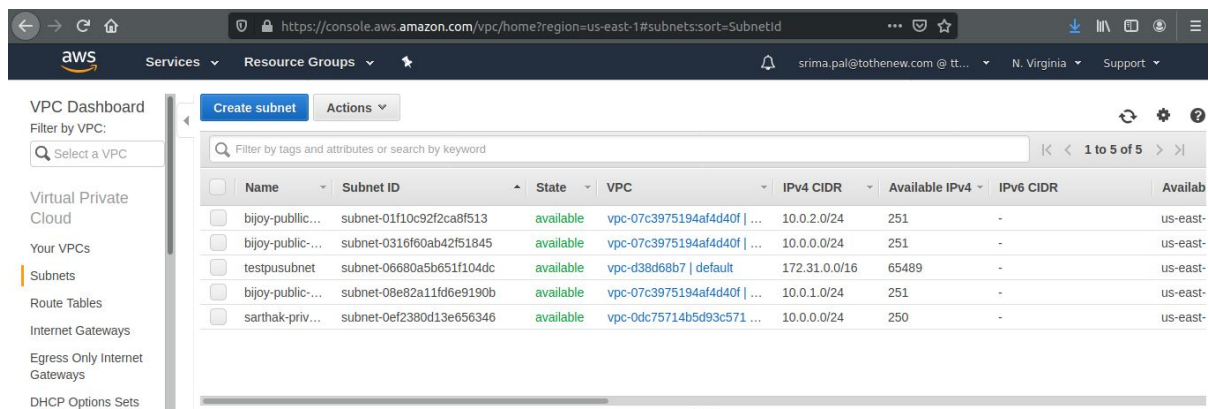
5. Differentiate between NACL and Security Groups.

Sno	Security Groups	Network ACL's
1.	Supports Allow rules only (implicit deny) You cannot deny a certain IP address from establishing a connection.	You can both allow and deny rules.
2.	Stateful: This means any changes applied to an incoming rule will be automatically applied to the outgoing rule.	Stateless: This means any changes applied to an incoming rule will not be applied to the outgoing rule.
3.	Security groups are tied to an instance.	Network ACL are tied to the subnet.
4.	All rules in a security group are applied.	Rules are applied in their order (the rule with the lower number gets processed first)

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

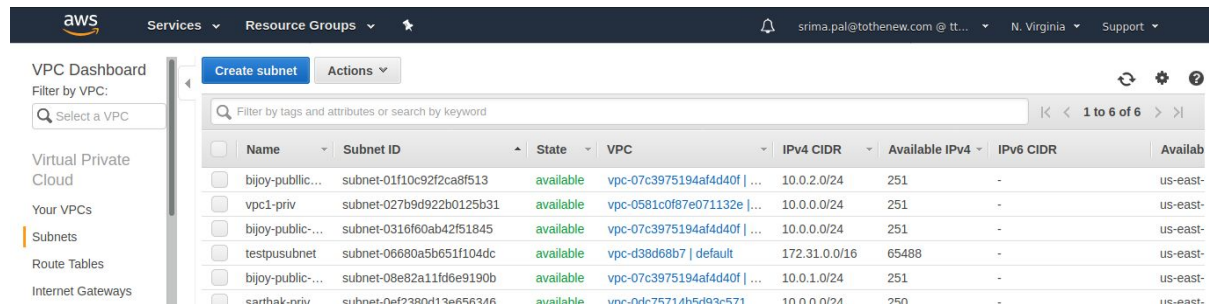
Step 1. Go to VPC, Subnets and then create subnets.



The screenshot shows the AWS Management Console VPC Dashboard. The left sidebar contains navigation links for VPC Dashboard, Filter by VPC, Virtual Private Cloud, Your VPCs, Subnets (highlighted), Route Tables, Internet Gateways, Egress Only Internet Gateways, and DHCP Options Sets. The main content area displays a table of subnets with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, and Availability Zone. The table lists five subnets, all in an 'available' state, associated with different VPCs. The first three subnets are public (bijoy-public-...), and the last two are private (sarthak-priv-...).

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
bijoy-public-...	subnet-01f10c92f2ca8f513	available	vpc-07c3975194af4d40f ...	10.0.2.0/24	251	-	us-east-
bijoy-public-...	subnet-0316f60ab42f51845	available	vpc-07c3975194af4d40f ...	10.0.0.0/24	251	-	us-east-
testpusubnet	subnet-06680a5b651f104dc	available	vpc-d38d68b7 default	172.31.0.0/16	65489	-	us-east-
bijoy-public-...	subnet-08e82a11fd6e9190b	available	vpc-07c3975194af4d40f ...	10.0.1.0/24	251	-	us-east-
sarthak-priv-...	subnet-0ef2380d13e656346	available	vpc-0dc75714b5d93c571 ...	10.0.0.0/24	250	-	us-east-

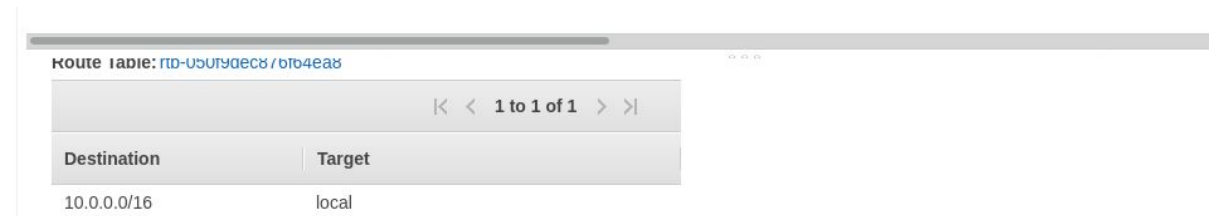
Step 2. Private Subnet Created by giving the subnet range to be 10.0.0.0/24



The screenshot shows the AWS VPC Dashboard with a list of subnets. The 'Subnets' link is highlighted in the left-hand navigation menu. The table below lists the subnets, including their names, IDs, states, associated VPCs, and IP address ranges.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
bijoy-public...	subnet-01f10c92f2ca8f513	available	vpc-07c3975194af4d40f ...	10.0.2.0/24	251	-	us-east-
vpc1-priv	subnet-027b9d922b0125b31	available	vpc-0581c0f87e071132e ...	10.0.0.0/24	251	-	us-east-
bijoy-public...	subnet-0316f60ab42f51845	available	vpc-07c3975194af4d40f ...	10.0.0.0/24	251	-	us-east-
testpusubnet	subnet-06680a5b651f104dc	available	vpc-d38d68b7 default	172.31.0.0/16	65488	-	us-east-
bijoy-public...	subnet-08e82a11fd6e9190b	available	vpc-07c3975194af4d40f ...	10.0.1.0/24	251	-	us-east-
santhak-nriv	subnet-0ef2380d113e656346	available	vpc-04fe75714b5d93e571	10.0.0.0/24	250	-	us-east-

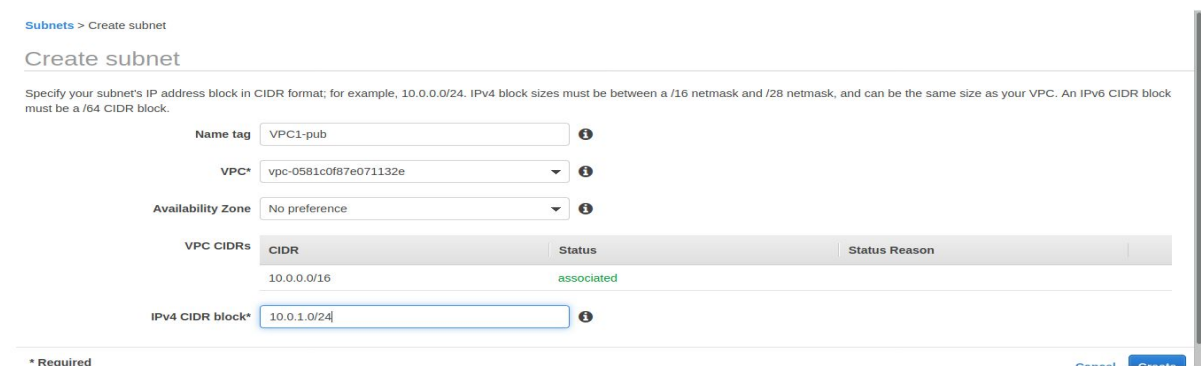
Step 3. We see that the route table(default) of the private subnet has no internet gateway attached



The screenshot shows the 'Route Table' configuration for the private subnet. The route table is named 'rtb-05079d6c8/b7b4eab8'. It contains a single route with the destination '10.0.0.0/16' and the target 'local', indicating that the subnet is not connected to the internet.

Destination	Target
10.0.0.0/16	local

Step 4. Create a public subnet



The screenshot shows the 'Create subnet' form in the AWS console. The form is titled 'Create subnet' and includes instructions on specifying the subnet's IP address block in CIDR format. The form fields are as follows:

- Name tag:** VPC1-pub
- VPC:** vpc-0581c0f87e071132e
- Availability Zone:** No preference
- VPC CIDRs:** A table showing the CIDR block '10.0.0.0/16' with a status of 'associated'.
- IPv4 CIDR block:** 10.0.1.0/24

* Required

Step 5. Make an internet gateway

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 Create

Internet gateways > Attach to VPC

Filter by tags and attributes or search by keyword

Name	ID	State	VPC	Owner
sarthak-kohli	igw-04a7d2d8e6c...	attached	vpc-0dc75714b5d...	187632318301
practice	igw-0867bfd7c25...	attached	vpc-d38d68b7 d...	187632318301
igw-09402f52541f...	igw-09402f52541f...	attached	vpc-07c3975194a...	187632318301
Srima	igw-0df86629c089...	detached	-	187632318301

Step 6. Attach internet gateway to VPC

Internet gateways > Attach 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*

Filter by attributes

VPC ID	Name
vpc-0581c0f87e071132e	VPC1

* Required Cancel Attach

Step 7. Create a route table

Route Tables > Create route table

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 Cancel Create

Step 8. Attach route table to the public subnet

The screenshot shows the AWS Management Console interface for editing subnet associations. The breadcrumb navigation is 'Route Tables > Edit subnet associations'. The page title is 'Edit subnet associations'. The route table selected is 'rtb-Ocea534897eea5511 (Srima-ttn)'. Under 'Associated subnets', 'subnet-0a09123785b6859ea' is selected. A table below shows the association details:

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0a09123785b6859ea VPC1-pub	10.0.1.0/24	-	Main
subnet-027b9d922b0125b31 vpc1-priv	10.0.0.0/24	-	Main

Step 7. Attach internet gateway to the route table of public subnet

The screenshot shows the AWS Management Console interface for editing routes. The breadcrumb navigation is 'Route Tables > Edit routes'. The page title is 'Edit routes'. The route table selected is 'rtb-Ocea534897eea5511 (Srima-ttn)'. A table below shows the existing routes:

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0df86629c0898c227		No

Below the table, there is an 'Add route' button. At the bottom, there are 'Cancel' and 'Save routes' buttons.

Step 8. Create another route table and attach it to the Private subnet

The screenshot shows the AWS Management Console interface for editing subnet associations. The breadcrumb navigation is 'Route Tables > Edit subnet associations'. The page title is 'Edit subnet associations'. The route table selected is 'rtb-Oe2ac7b2384639013 (Srima-ttn-priv)'. Under 'Associated subnets', 'subnet-027b9d922b0125b31' is selected. A table below shows the association details:

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0a09123785b6859ea VPC1-pub	10.0.1.0/24	-	rtb-Ocea534897eea5511
subnet-027b9d922b0125b31 vpc1-priv	10.0.0.0/24	-	Main

Step 9. Create a NAT Gateway

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-027b9d922b0125b31

Elastic IP Allocation ID* eipalloc-0034154adb4d2f37b

Elastic IP address 3.209.182.149 allocated.

* Required

Cancel Create a NAT Gateway

Step 10. Attach NAT Gateway to private subnet

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	nat-0a4f49a5043ca0123		No

Add route

* Required

Cancel Save routes

Step 11. Launch 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 3: Configure Instance Details

Purchasing option ☐ Request Spot instances

Network vpc-0dc75714b5d93c571 | sarthak-vpc

Subnet subnet-022cabd877f3e421b | Srma-pub | us-east-1c

Auto-assign Public IP Enable

Placement group ☐ Add instance to placement group

Capacity Reservation Open

4. Choose PM 5. Choose instance type 6. Configure instance 7. Auto storage 8. Auto tags 9. Configure security group 10. 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:

Description:

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

Step 12. Launch an instance in private subnet, disabling the auto assign ip feature and allowing custom tcp(port 8080) in security group .

Ssh into the public instance and access the private instance through the public instance

```

srlna@srlna: ~
ubuntu@ip-10-0-6-5: ~ 75x39
logout [root] EC2 Manag... TTN VPC Assignment... Learning | Dashboard...
srlna@srlna:~$ sudo ssh -i Downloads/Srlna-TTN-bootcamp.pem ubuntu@100.27.31.46
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 11:14:33 UTC 2020

System load:  0.0      Processes:      instance ID 87 - Instance Type
Usage of /:   13.8% of 7.69GB   Users logged in: 0
Memory usage: 15%      S IP address for eth0: 10.0.6.5
Swap usage:   0%

Limits
-----
SrlnaQ9          i-025bea4735a5e5032   i2.micro
Srlna(B)         i-0334eedf78ed55955   i2.micro
SrlnaPriv        i-040ab97e264d944...   i2.micro
Srlna(A)         i-067ce78f53dc7ab10   i2.micro

0 packages can be updated.
0 updates are security updates.

Last login: Mon Feb 24 11:11:36 2020 from 182.71.160.186
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-6-5:~$ sudo vim .ssh/authorized_keys
ubuntu@ip-10-0-6-5:~$ ls
Srlna-TTN-bootcamp.pem
ubuntu@ip-10-0-6-5:~$ ssh -i Srlna-TTN-bootcamp.pem ubuntu@10.0.5.13

```

```

srlna@srlna: ~ 75x30
srlna@srlna:~$ sudo scp Downloads/Srlna-TTN-bootcamp.pem ubuntu@100.27.31.46:/home/ubuntu
6:/home/ubuntu: Permission denied (publickey).
ubuntu@100.27.31.46: Permission denied (publickey).
lost connection
srlna@srlna:~$ scp Downloads/Srlna-TTN-bootcamp.pem ubuntu@100.27.31.46:/me/ubuntu
The authenticity of host '100.27.31.46 (100.27.31.46)' can't be established.
ECDSA key fingerprint is SHA256:Xun5NWLigUdnkNLVasBvUFGzWLD9BGJcRUavZ7LX...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '100.27.31.46' (ECDSA) to the list of known hosts.
ubuntu@100.27.31.46: Permission denied (publickey).
lost connection
srlna@srlna:~$ cd .ssh
srlna@srlna:~/.ssh$ sudo vim id_rsa
srlna@srlna:~/.ssh$ sudo vim id_rsa.pub
srlna@srlna:~/.ssh$ scp Downloads/Srlna-TTN-bootcamp.pem ubuntu@100.27.31.46:/home/ubuntu
Downloads/Srlna-TTN-bootcamp.pem: No such file or directory
srlna@srlna:~/.ssh$ cd
srlna@srlna:~$ scp Downloads/Srlna-TTN-bootcamp.pem ubuntu@100.27.31.46:/me/ubuntu
Srlna-TTN-bootcamp.pem
srlna@srlna:~$

```


Step 13. Install tomcat on private instance

```
ubuntu@ip-10-0-5-13:~$ sudo systemctl start tomcat9
ubuntu@ip-10-0-5-13:~$ sudo systemctl status tomcat9
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor pre
   Active: active (running) since Mon 2020-02-24 11:19:53 UTC; 1min 9s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Main PID: 3835 (java)
     Tasks: 34 (limit: 1152)
    CGroup: /system.slice/tomcat9.service
            └─3835 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.co

Feb 24 11:19:56 ip-10-0-5-13 tomcat9[3835]: OpenSSL successfully initialize
Feb 24 11:19:56 ip-10-0-5-13 tomcat9[3835]: Initializing ProtocolHandler ["
Feb 24 11:19:56 ip-10-0-5-13 tomcat9[3835]: Server initialization in [2,814
Feb 24 11:19:57 ip-10-0-5-13 tomcat9[3835]: Starting service [Catalina]
Feb 24 11:19:57 ip-10-0-5-13 tomcat9[3835]: Starting Servlet engine: [Apach
Feb 24 11:19:57 ip-10-0-5-13 tomcat9[3835]: Deploying web application direc
Feb 24 11:20:01 ip-10-0-5-13 tomcat9[3835]: At least one JAR was scanned fo
Feb 24 11:20:01 ip-10-0-5-13 tomcat9[3835]: Deployment of web application d
Feb 24 11:20:01 ip-10-0-5-13 tomcat9[3835]: Starting ProtocolHandler ["http
Feb 24 11:20:01 ip-10-0-5-13 tomcat9[3835]: Server startup in [4,669] milli
lines 1-19/19 (END)
```

Step 14. Install nginx on public instance

```
ubuntu@ip-10-0-6-5:~$ sudo systemctl start nginx
ubuntu@ip-10-0-6-5:~$ 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 prese
   Active: active (running) since Mon 2020-02-24 11:22:10 UTC; 53s ago
     Docs: man:nginx(8)
   Main PID: 2767 (nginx)
     Tasks: 2 (limit: 1152)
    CGroup: /system.slice/nginx.service
            └─2767 nginx: master process /usr/sbin/nginx -g daemon on; maste
              └─2769 nginx: worker process

Feb 24 11:22:10 ip-10-0-6-5 systemd[1]: Starting A high performance web ser
Feb 24 11:22:10 ip-10-0-6-5 systemd[1]: nginx.service: Failed to parse PID
Feb 24 11:22:10 ip-10-0-6-5 systemd[1]: Started A high performance web serv
lines 1-13/13 (END)
```


Step 15. Create file

```
ubuntu@ip-10-0-6-5: /var/www/html 75x39
<html>
<head>
<title>Nginx</title>
</head>
<body>
<p>NGINX</p>
</body>
</html>
```

Abc.com

```
ubuntu@ip-10-0-6-5: /etc/nginx/sites-available 75x39
server {
listen 80;
server_name abc.com;

root /var/www/html;
index index.html;
location / {
proxy_pass http://10.0.5.13:8080;
}
}
```

Step 16. Proxy Passed

```
ubuntu@ip-10-0-6-5:/etc/nginx/sites-available$ curl abc.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 file system 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>.
```

