

ASSESSMENT ON VPC



Q1. When to use Elastic IP over Public IP

ANS: Public IP addresses are dynamic - i.e. if you stop/start your instance you get reassigned a new public IP.

Elastic IPs get allocated to your account, and stay the same - it's up to you to attach them to an instance or not. You could say they are static public IP addresses.

Elastic IP address is a public **static IPv4 address** which is reachable from the Internet.

Basically Elastic IP addresses are used by AWS to manage its dynamic cloud computing services. Within the AWS infrastructure, customers have virtual private clouds (VPC), within the VPCs, users have instances. So when you launch an EC2 instance, you receive a Public IP address by which that instance is reachable from internet. Once you stop that instance and restart the instance you get a new Public IP for the same instance. So it's basically a problem to connect your instance from internet for not having a static IP. To overcome this problem, *we attach an Elastic IP to an Instance which doesn't change after you stop / start the instance.*

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

ANS:

* Class A : 10.0.0.0 – 10.0.255.255

Class B : 172.16.0.0 – 172.13.255.255

Class C : 192.168.0.0 – 192.168.255.255

If you ever expect to connect these systems to an Internet-facing router, though, then you could experience the following issues if you don't stick with private IP ranges:

- Traffic destined for another host may leak out on to the Internet.
- You might want to get to the IANA-assigned host on that IP and may not be able to do it if it's an internal host.

- If you aren't the only one maintaining this network, you could horribly confuse someone who is doing troubleshooting.

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

ANS:

To create a VPC peering connection with another VPC, be aware of the following limitations and rules:

- You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks. Amazon always assigns your VPC a unique IPv6 CIDR block. If your IPv6 CIDR blocks are unique but your IPv4 blocks are not, you cannot create the peering connection.
- You have a quota on the number of active and pending VPC peering connections that you can have per VPC. For more information, see Amazon VPC Quotas in the *Amazon VPC User Guide*
- VPC peering does not support transitive peering relationships. In a VPC peering connection, your VPC does not have access to any other VPCs with which the peer VPC may be peered. This includes VPC peering connections that are established entirely within your own AWS account. For more information about unsupported peering relationships, see *Unsupported VPC Peering Configurations*. For examples of supported peering relationships, see *VPC Peering Scenarios*.
- You cannot have more than one VPC peering connection between the same two VPCs at the same time.
- Unicast reverse path forwarding in VPC peering connections is not supported. For more information, see *Routing for Response Traffic*.
- Any tags that you create for your VPC peering connection are only applied in the account or region in which you create them.

- If the IPv4 CIDR block of a VPC in a VPC peering connection falls outside of the private IPv4 address ranges specified by RFC 1918, private DNS hostnames for that VPC cannot be resolved to private IP addresses. To resolve private DNS hostnames to private IP addresses, you can enable DNS resolution support for the VPC peering connection. For more information, see [Enabling DNS Resolution Support for a VPC Peering Connection](#).
- You cannot connect to or query the Amazon DNS server in a peer VPC.

An inter-region VPC peering connection has additional limitations:

- You cannot create a security group rule that references a peer VPC security group.
- You cannot enable support for an EC2-Classic instance that's linked to a VPC via ClassicLink to communicate with the peer VPC.
- The Maximum Transmission Unit (MTU) across the VPC peering connection is 1500 bytes (jumbo frames are not supported).
- You must enable DNS resolution support for the VPC peering connection to resolve private DNS hostnames of the peered VPC to private IP addresses, even if the IPv4 CIDR for the VPC falls into the private IPv4 address ranges specified by RFC 1918.
- Inter-region peering in China is only allowed between the China (Beijing) Region, operated by SINNET and the China (Ningxia) Region, operated by NWCD.

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.

ANS:

* 10.0.0.0/16: 00001010.00000000.00000000.00000000 (In /16, first 2 octets are fixed).
00001010.00000000.00000000.00000000 (In /20, extra 4 bits are borrowed from hosts)

* These extra 4 bits are subnetting bits.

So, total number of subnets = 2^4 (16)

And, total IP'S in each subnet = 2^{12} (4096)

Q5. Differentiate between NACL and Security Groups.

ANS :

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.

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.

STEP 1: Creating VPC

[VPCs](#) > Create VPC

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 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 with the VPC.

Name tag ⓘ

IPv4 CIDR block* ⓘ

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

Tenancy ⓘ

* Required

STEP 2: Creating subnet (Private)

[Subnets](#) > Create 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 CIDR block.

Name tag ⓘ

VPC* ⓘ

Availability Zone ⓘ

VPC CIDRs	CIDR	Status
	10.0.0.0/16	associated

IPv4 CIDR block* ⓘ

* Required

STEP 3: Creating subnet (Public)

[Subnets](#) > Create 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 a /28 CIDR block.

Name tag ⓘ

VPC* ⓘ

Availability Zone ⓘ

VPC CIDRs	CIDR	Status
	10.0.0.0/16	associated

IPv4 CIDR block* ⓘ

* Required

STEP 4: Creating 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

[Internet gateways](#) > Create internet gateway

Create internet gateway

✓ The following internet gateway was created:

Internet gateway ID [igw-0d9354efa9dd18f7c](#)

Close

STEP 5: Attaching this Internet Gateway to the VPC.

Create internet gateway Actions ^

Filter by tags and attributes

	Name	ID		VPC	Owner
<input type="checkbox"/>	sarthak-kohli	igw-04a7d2d8e6c...	attached	vpc-0dc75714b5d...	187632318301
<input checked="" type="checkbox"/>	diksha_inter...	igw-06855f33858d...	detached	-	187632318301
<input type="checkbox"/>	Test-IGW	igw-0713d68cdb0...	attached	vpc-066a708be04...	187632318301
<input type="checkbox"/>	practice	igw-0867bfbd7c25...	attached	vpc-d38d68b7 d...	187632318301
<input type="checkbox"/>		igw-09402f52541f...	attached	vpc-07c3975194a...	187632318301

Actions menu:

- Delete internet gateway
- Attach to VPC
- Detach from VPC
- Add/Edit Tags

[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*

► AWS Command Line Interface command

* Required

STEP 6: Creating route table for public subnet.

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

[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

STEP 7: Editing the routes (for making publically accessible)

[Route Tables](#) > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="igw-0d9354efa9dd18f7d"/>		No
<input type="button" value="Add route"/>			

* Required

STEP 8: Associate public subnet with the public route table.

[Route Tables](#) > Edit subnet associations

Edit subnet associations

Route table rtb-0c2b08656f6a8a813 (Public_subnetDiksha)

Associated subnets subnet-0ca0a882801f676b6

Filter by attributes or search by keyword				1 to 2 of 2	
<input type="checkbox"/>	Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table	
<input checked="" type="checkbox"/>	subnet-0ca0a882801f676b6 PublicSub...	10.0.2.0/24	-	Main	
<input type="checkbox"/>	subnet-0afc948beae26f4e4 PrivateSub...	10.0.1.0/24	-	Main	

STEP 9: Launching an instance in private and other in public subnets.

Launch Instance

Connect

Actions

owner : Diksha

Add filter

	Name	Purpose	Instance ID	Instance Type	Availability Zone	Instance State	S
<input type="checkbox"/>	PrivateInstance	Launching a...	i-014121a59f1bc9813	t2.nano	us-east-1a	terminated	
<input checked="" type="checkbox"/>	privateInstan...	Launching in...	i-0851fb31e900519da	t2.nano	us-east-1a	running	
<input type="checkbox"/>		Launching in...	i-0955adf837bc40f79	t2.nano	us-east-1b	terminated	
<input checked="" type="checkbox"/>	PublicInstan...	Launching in...	i-0d7b60efe4b829974	t2.nano	us-east-1b	running	

STEP 10: Now making NAT gateway

[NAT Gateways](#) > Create NAT Gateway

Create NAT Gateway

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

Subnet* subnet-0afc948beae26f4e4  

Elastic IP Allocation ID*  Filter by attributes

Subnet ID	Subnet Name	VPC ID	VPC Name
subnet-08e82a11fd6e9190b	bijoy-public-subnet-2	vpc-07c3975194af4d40f	bijoy-vpc
subnet-0c206ef481fdf6d53	Test_A	vpc-066a708be046c7c92	Test
subnet-0ca0a882801f676b6	PublicSubnet	vpc-05b90a8310a6687f1	diksha_vpc
subnet-08e22df1ad019c6db	Test_B	vpc-066a708be046c7c92	Test
subnet-0afc948beae26f4e4	PrivateSubnet	vpc-05b90a8310a6687f1	diksha_vpc


* Required

[NAT Gateways](#) > Create NAT Gateway

Create NAT Gateway

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

Subnet* subnet-0387965841402c2ff  

Elastic IP Allocation ID* eipalloc-059d31fb2a1aef288   **Allocate Elastic IP address** 

Elastic IP address (18.211.34.214) allocated.


* Required Cancel Create a NAT Gateway

STEP 11: Now edit routes in my private subnet's route table.

[Route Tables](#) > Edit routes

Edit routes

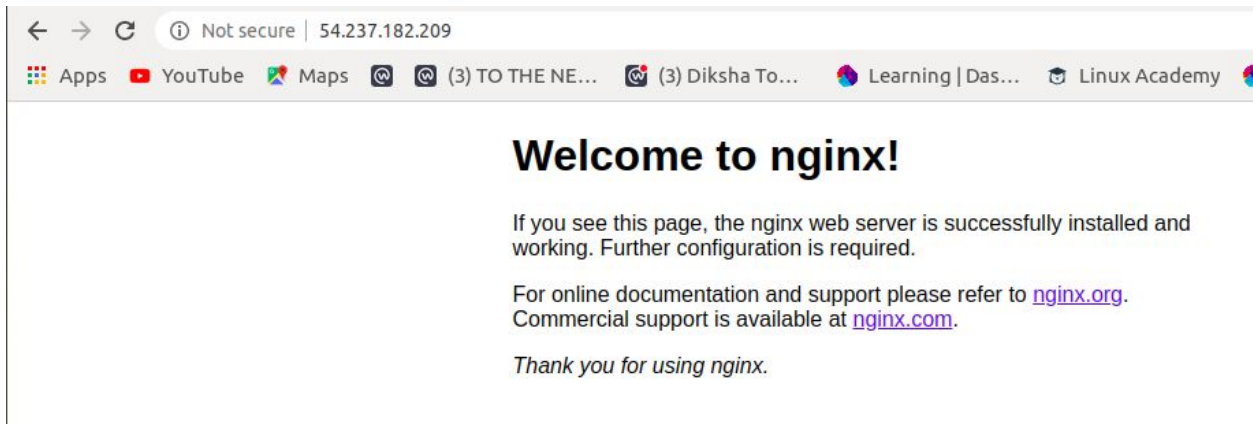
Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-089ed0ceb0bf62619		No

 Add route

* Required Cancel Save routes

STEP 12: Installing TOMCAT in private subnet's instance.

STEP 13: Installing Nginx in public subnet's instance



```

diksha@diksha:~$ cd Downloads
diksha@diksha:~/Downloads$ ls
'Advanced Linux.pdf'
'assessment1 .pdf'
'assessment5 Nginx.pdf'
'Assessment6_mongodb n mysql.pdf'
'ASSESSMENT 9_EC2 AND EBS.pdf'
'DATABASE Assessment (1).pdf'
'DATABASE Assessment.pdf'
diksha_awskey.pem
giphy.gif
'Git Assessmentt.pdf'
Linux.pdf
mongodb-linux-x86_64-ubuntu1804-v4.2-latest.tgz
sample.war
sites-enabled
'Tomcat assignment.docx'
ubuntu@54.237.182.209
VPC.pdf
diksha@diksha:~/Downloads$ man scp
diksha@diksha:~/Downloads$ scp -i diksha_awskey.pem sample.war ubuntu@
10.0.2.55:~
^Cdiksha@diksha:~/Downloads$ scp -i diksha_awskey.pem sample.war ubuntu
54.208.83.237:~
sample.war                                100% 4606    18.4KB/s   00:00
diksha@diksha:~/Downloads$ scp -i diksha_awskey.pem diksha ubuntu@54.2
08.83.237:~
diksha:                                    diksha_awskey.pem
diksha@diksha:~/Downloads$ scp -i diksha_awskey.pem diksha ubuntu@54.2
08.83.237:~
diksha:                                    diksha_awskey.pem
diksha@diksha:~/Downloads$ scp -i diksha_awskey.pem diksha_awskey.pem
ubuntu@54.208.83.237:~

```

STEP 14: Now proxy passing tomcat through nginx.


```
# 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://10.0.2.205:8080/sample/;
    try_files $uri $uri/ =404;
}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
```

STEP 15: Checking the status using curl

```
Processing triggers for ureadahead (0.100.0-21) ...
root@ip-10-0-2-205:~# curl localhost
curl: (7) Failed to connect to localhost port 80: Connection refused
root@ip-10-0-2-205:~# curl localhost: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>

<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>

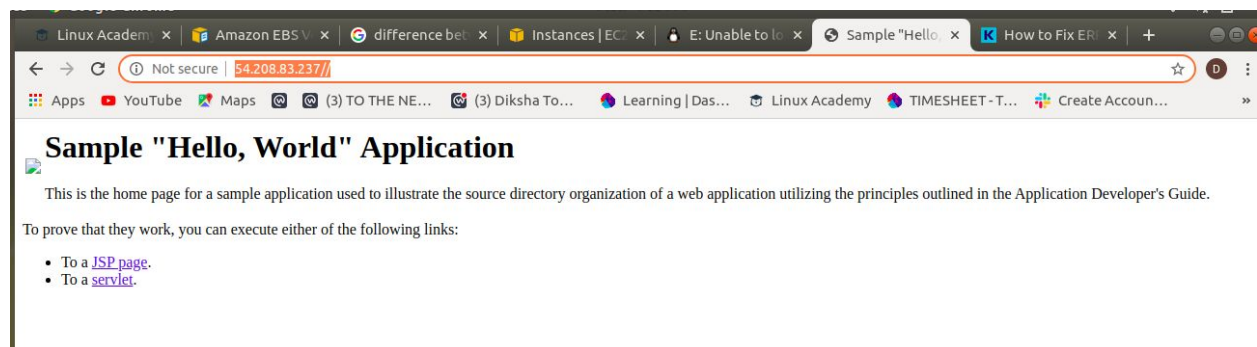
<p><b>tomcat9-docs</b>: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking <a href="docs/">here</a>.</p>

<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>
```

STEP 16: In Browser.



The screenshot shows a web browser window with multiple tabs. The active tab is titled "Sample 'Hello, World' Application". The address bar shows the URL "54.208.83.237/". The page content includes a heading "Sample 'Hello, World' Application" and a paragraph stating: "This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide." Below this, there is a section "To prove that they work, you can execute either of the following links:" followed by a bulleted list:

- To a [JSP page](#).
- To a [servlet](#).

AWS

