

VPC -Scenario-1

1. Create VPC for “10.20.0.0/20” ☐ for 300 IP’s per Network and Consecutive 6 networks. (6 Subnets)
2. Create Internet Gateway and attach it to YOUR VPC.
3. Add below Routing entries in the Routing Table.
 - a. “0.0.0.0/0” ☐ “I.G that has been created”.
 - b. “:::/0” ☐ “I.G that has been created”.
4. Create EC2 instance and ENABLE both “ Public IPV4” and “Public IP v6”.
5. Create an “New” key for this instance.
6. Access the EC2 Instance via internet (SSH , as it would be an linux machine)

Solution: On AWS CLI Command Sequence

Note:-- Please make a note of , VPC-ID, Subnet-ID, Routing-table-ID, Internet-Gateway-ID,

1. `aws ec2 create-vpc --cidr-block 10.20.0.0/20 --amazon-provided-ipv6-cidr-block`

Output:

CreateVpc	
Vpc	
CidrBlock	10.20.0.0/20
DhcpOptionsId	dopt-9d6c83e4
InstanceTenancy	default
IsDefault	False
State	pending
VpcId	
CidrBlockAssociationSet	
AssociationId	vpc-cidr-assoc-08516b932075641b4
CidrBlock	10.20.0.0/20
CidrBlockState	
State	associated
Ipv6CidrBlockAssociationSet	
AssociationId	vpc-cidr-assoc-038654ef9b2b0d78f
Ipv6CidrBlock	
Ipv6CidrBlockState	
State	associating

`aws ec2 describe-vpcs --vpc-ids vpc-XXXXXXXXXXXXXX` → will give the details of the IPv6 that is assigned.

2. `aws ec2 create-subnet --availability-zone us-east-1a --cidr-block 10.20.0.0/23 --ipv6-cidr-block 2600:XXXX:XXXX:XX00::/64 --vpc-ids vpc-XXXXXXXXXXXXXX`

'Output:

CreateSubnet	
Subnet	
AssignIpv6AddressOnCreation	False
AvailabilityZone	us-east-1a
AvailableIpAddressCount	507
CidrBlock	10.20.0.0/23
DefaultForAz	False
MapPublicIpOnLaunch	False
State	pending
SubnetId	subnet-02a[REDACTED]
VpcId	vpc-0a40fd[REDACTED]
Ipv6CidrBlockAssociationSet	
AssociationId	subnet-cidr-assoc-00591da9a1f4ca9b6
Ipv6CidrBlock	2600:1f18:448f:6b00::/64
Ipv6CidrBlockState	
State	associating

Similarly Create 5 more subnets

- 2.2 `aws ec2 create-subnet --availability-zone us-east-1b --cidr-block 10.20.2.0/23 --ipv6-cidr-block 2600:XXXX:XXXX:XX01::/64 --vpc-ids vpc-XXXXXXXXXXXXXXXXXX`
- 2.3 `aws ec2 create-subnet --availability-zone us-east-1c --cidr-block 10.20.4.0/23 --ipv6-cidr-block 2600:XXXX:XXXX:XX02::/64 --vpc-ids vpc-XXXXXXXXXXXXXXXXXX`
- 2.4 `aws ec2 create-subnet --availability-zone us-east-1d --cidr-block 10.20.6.0/23 --ipv6-cidr-block 2600:XXXX:XXXX:XX03::/64 --vpc-ids vpc-XXXXXXXXXXXXXXXXXX`
- 2.5 `aws ec2 create-subnet --availability-zone us-east-1e --cidr-block 10.20.8.0/23 --ipv6-cidr-block 2600:XXXX:XXXX:XX04::/64 --vpc-ids vpc-XXXXXXXXXXXXXXXXXX`
- 2.6 `aws ec2 create-subnet --availability-zone us-east-1f --cidr-block 10.20.10.0/23 --ipv6-cidr-block 2600:XXXX:XXXX:XX05::/64 --vpc-ids vpc-XXXXXXXXXXXXXXXXXX`

3. `aws ec2 create-internet-gateway`

Output:

CreateInternetGateway	
InternetGateway	
InternetGatewayId	igw-010[REDACTED]

- 3.2 `aws ec2 attach-internet-gateway --internet-gateway-id igw-0105e7d41e5XXXXXX --vpc-id vpc-0a40fd2844ffbXXXX`
- 4.1 `aws ec2 create-route --destination-cidr-block 0.0.0.0/0 --gateway-id igw-0105e7d41e58XXXXXX --route-table-id rtb-0613fa3d649aXXXXX`

```

-----
|   CreateRoute   |
+-----+-----+
|   Return   |   True   |
+-----+-----+

```

4.1 `aws ec2 create-route --destination-ipv6-cidr-block ::/0 --gateway-id igw-0105e7d41e58XXXXX --route-table-id rtb-0613fa3d649aXXXXX`

```

-----
|   CreateRoute   |
+-----+-----+
|   Return   |   True   |
+-----+-----+

```

5.0 Add subnet attribute to enable, Auto-assign IPv6 and IPv4, below are the commands respectively

a. `aws ec2 modify-subnet-attribute --subnet-id subnet-02a7734fecf9XXXXX --assign-ipv6-address-on-creation`

b. `aws ec2 modify-subnet-attribute --subnet-id subnet-02a7734fecf9XXXXX --map-public-ip-on-launch`

6.1 Create security group – on the VPC

`aws ec2 create-security-group --description SSH-port --group-name SSH-port --vpc-id vpc-0a40fd2844ffXXXXX`

output:

```

{
  "GroupId": "sg-02733e1b625ed1a67"
}

```

6.2 Assign rules to the security group

7.0 Create the EC2 instance

`aws ec2 run-instances --image-id ami-b70554c8 --count 1 --instance-type t2.micro --key-name b16-key1 --security-group-ids sg-XXXXXXXXXXXXX --subnet-id subnet-XXXXXXXXXXXXX`

Output:

DescribeInstances	
Reservations	
OwnerId	766476266373
ReservationId	r-0bba3a1a1abdc2a77
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-b70554c8
InstanceId	i-034d152 [REDACTED]
InstanceType	t2.micro
KeyName	b16-key1
LaunchTime	2018-08-10T05:31:28.000Z
PrivateDnsName	ip-10-20-0-179.ec2.internal
PrivateIpAddress	10.20.0.179
PublicDnsName	
PublicIpAddress	107.23. [REDACTED]
RootDeviceName	/dev/xvda
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-02a7734fecf9b4b4a
VirtualizationType	hvm
VpcId	vpc-0a40fd2 [REDACTED]
BlockDeviceMappings	
DeviceName	/dev/xvda

To delete the Complete Setup

1. delete the instance

aws ec2 terminate-instances --instance-ids i-03cef0d6afe8XXXXXX

2. Delete the security group

aws ec2 delete-security-group --group-id sg-02733e1b625XXXXXX

3. Remove the routing table entry for default routes

aws ec2 delete-route --route-table-id rtb-0613fa3d649XXXXXX --destination-cidr-block 0.0.0.0/0

aws ec2 delete-route --route-table-id rtb-0613fa3d649XXXXXX --destination-ipv6-cidr-block ::/0

4. Detach the VPC from internet-Gateway

aws ec2 detach-internet-gateway --internet-gateway-id igw-0105e7d41e5XXXXXX --vpc-id vpc-0a40fd2844fXXXXXX

5. Delete the Internet-Gateway

aws ec2 delete-internet-gateway --internet-gateway-id igw-0105e7d41e5XXXXXX

6. Delete the subnets

aws ec2 delete-subnet --subnet-id subnet-02a7734fecfXXXXXX – ALL The 6 subnets to be Deleted.

7. Delete the VPC

```
aws ec2 delete-vpc --vpc-id vpc-0a40fd2844fXXXXXX
```