

Name :- Rushikesh Anil Mashidkar

Email Id :- rishikeshmashidkar@gmail.com

Note :- The answer is in bullets (➤)

Assignment

Ques 1:

- Create your own new custom VPC
 - ➔ Created new VPC name as **assignment-8** and give CIDR range **10.0.0.0/16**. Then I select vpc and right click and enabled the DNS hostnames and save. It is not enabled by default so.

The screenshot shows the AWS VPC console interface. At the top, there's a search bar with 'vpc-0064a91d1a7126738' entered. Below the search bar, a table lists VPCs. The first entry is 'assignment-8' with VPC ID 'vpc-0064a91d1a7126738', State 'Available', IPv4 CIDR '10.0.0.0/16', and IPv6 CIDR '-'. Below the table, the 'Details' tab is selected for the VPC. The details are organized into four columns: VPC ID (vpc-0064a91d1a7126738), State (Available), DNS hostnames (Enabled), and DNS resolution (Enabled). Other details include Tenancy (Default), Default VPC (No), Network Address Usage metrics (Disabled), DHCP option set (dopt-01d98571aeec23eba), Main route table (rtb-01f2b648b618c0303), IPv6 pool (-), Main network ACL (acl-06ab638ee15fe4150), IPv6 CIDR (Network border group) (-), and Owner ID (162477790861).

- and then create 3 subnets of that VPC
 - ➔ Created 3 subnet as names **subnet-1**, **subnet-2** & **subnet-3** and give CIDR range of **10.0.0.0/24**, **10.0.1.0/24**, **10.0.2.0/24** respectively and add each of them in different AZ.
 - After that modified auto assign IP settings to Enabled for auto assigning Ip to subnet one by one.

The screenshot shows the AWS VPC console interface for subnets. On the left, there's a sidebar with navigation options. The main area shows a search bar with 'vpc-0064a91d1a7126738' entered. Below the search bar, a table lists subnets. The first entry is 'subnet-1' with Subnet ID 'subnet-040f537938df90fe', State 'Available', VPC 'vpc-0064a91d1a7126738 | assignment-8', IPv4 CIDR '10.0.0.0/24', IPv6 CIDR '-', and Available IP addresses '250'. Below the table, the 'Details' tab is selected for 'subnet-040f537938df90fe / subnet-1'. The details are organized into four columns: Subnet ID (subnet-040f537938df90fe), Subnet ARN (arn:aws:ec2:us-east-1:162477790861:subnet/subnet-040f537938df90fe), State (Available), and IPv4 CIDR (10.0.0.0/24). Other details include Availability Zone (us-east-1a), Route table (rtb-011b3fea3a9749d93 | MyCustomRT), Network ACL (acl-06ab638ee15fe4150), Auto-assign IPv6 address (Auto-assign IPv6 address), and Auto-assign customer-owned IPv4 address (Auto-assign customer-owned IPv4 address).

- Attach an internet gateway to your custom VPC
- ➔ Create internet gateway named as **My-IGW**. And attached to VPC (**assignment-8**).
- You didn't mention about Route table we have to create route table for internate access. So I created Route table name as **MyCustomRT** and after that associated my vpc subnets to Internet gateway.
- Also I edit the route to **0.0.0.0/0** and assign to **My-IGW** for internet from anywhere.

Internet gateways (1/1) Info

Filter internet gateways

search: vpc-0064a91d1a7126738 X Clear filters

<input checked="" type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Owner
<input checked="" type="checkbox"/>	My-IGW	igw-0a423f994df6ee4a0	Attached	vpc-0064a91d1a7126738 assignment-8	162477790861

igw-0a423f994df6ee4a0 / My-IGW

Details Tags

Details

Internet gateway ID
igw-0a423f994df6ee4a0

State
Attached

VPC ID
vpc-0064a91d1a7126738 | assignment-8

Owner
162477790861

Route tables (1/2) Info

Filter route tables

search: vpc-0064a91d1a7126738 X Clear filters

<input checked="" type="checkbox"/>	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
<input checked="" type="checkbox"/>	MyCustomRT	rtb-011b3fea3a9749d93	3 subnets	-	No	vpc-0064a91d1a7126738 ass...	162477790861
<input type="checkbox"/>	-	rtb-01f2b648b618c0303	-	-	Yes	vpc-0064a91d1a7126738 ass...	162477790861

rtb-011b3fea3a9749d93 / MyCustomRT

Details Routes Subnet associations Edge associations Route propagation Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Details

Route table ID
rtb-011b3fea3a9749d93

Main
No

Explicit subnet associations
3 subnets

Edge associations
-

VPC
vpc-0064a91d1a7126738 | assignment-8

Owner ID
162477790861

Route tables (1/2) Info

Filter route tables

search: vpc-0064a91d1a7126738 X Clear filters

<input checked="" type="checkbox"/>	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
<input checked="" type="checkbox"/>	MyCustomRT	rtb-011b3fea3a9749d93	3 subnets	-	No	vpc-0064a91d1a7126738 ass...	162477790861

rtb-011b3fea3a9749d93 / MyCustomRT

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes Both

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0a423f994df6ee4a0	Active	No
10.0.0.0/16	local	Active	No

- And configure your EC2 linux instance inside your custom VPC
- ➔ Launch instance with name **VPC-Server** inside my VPC **assignment-8** and also connect through SSH and **ping to google.com**. The output is given below.

The screenshot shows the AWS Management Console interface. At the top, there's a search bar and navigation tabs. The main content area displays the 'Instances (1/1) Info' page. A table lists the instance 'VPC-Server' with ID 'i-090f4d0c6b2fb0256', state 'Running', type 't2.micro', and public IP '52.91.87.212'. Below the table, the 'Instance: i-090f4d0c6b2fb0256 (VPC-Server)' details are shown. The 'Details' tab is active, displaying a summary of the instance's configuration, including its ID, IP addresses, hostname, and VPC ID.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address
VPC-Server	i-090f4d0c6b2fb0256	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-52-91-87-212.com...	52.91.87.212

Instance: i-090f4d0c6b2fb0256 (VPC-Server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID i-090f4d0c6b2fb0256 (VPC-Server)	Public IPv4 address 52.91.87.212 open address	Private IPv4 addresses 10.0.0.98
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-91-87-212.compute-1.amazonaws.com open address
Hostname type IP name: ip-10-0-0-98.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-0-98.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A) Auto-assigned IP address 52.91.87.212 [Public IP]	Instance type t2.micro VPC ID vpc-0064a91d1a7126738 (assignment-8) open address	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more

```
ubuntu@ip-10-0-0-98: ~  
ubuntu@ip-10-0-0-98:~$ ping google.com  
PING google.com (142.251.163.101) 56(84) bytes of data:  
64 bytes from wv-in-f101.1e100.net (142.251.163.101): icmp_seq=1 ttl=96 time=2.08 ms  
64 bytes from wv-in-f101.1e100.net (142.251.163.101): icmp_seq=2 ttl=96 time=2.13 ms  
64 bytes from wv-in-f101.1e100.net (142.251.163.101): icmp_seq=3 ttl=96 time=2.16 ms  
52.91.87.21264 bytes from wv-in-f101.1e100.net (142.251.163.101): icmp_seq=4 ttl=96 time=2.58 ms  
64 bytes from wv-in-f101.1e100.net (142.251.163.101): icmp_seq=5 ttl=96 time=2.13 ms  
^C  
--- google.com ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4006ms  
rtt min/avg/max/mdev = 2.080/2.215/2.576/0.182 ms  
ubuntu@ip-10-0-0-98:~$
```