

LAB : VIRTUAL PRIVATE CLOUD(VPC)	2
TASK BREAKDOWN	2
TASK 1: CREATE A VPC WITH A PUBLIC SUBNET.....	2
TASK 2: CREATE A PRIVATE SUBNET.....	5
TASK 3: LAUNCH A WEB SERVER INTO THE PUBLIC SUBNET.....	10
TASK 4: LAUNCH A DATABASE SERVER INTO THE PRIVATE SUBNET.....	14
TASK 5: TEST OUT ACCESSIBILITY OF SUBNETS	17
TASK 6: LAUNCH A NAT INSTANCE.....	21
TASK 7: MODIFY THE PRIVATE ROUTE TABLE TO ROUTE TRAFFIC TO NAT	25
TASK 8: S3 ENDPOINT.....	27
TASK 9: VPC PEERING	31
TASK 10: SETUP TRANSIT GATEWAY.....	34
TASK 11: VPC PRIVATE LINKS	37
TASK 12: DOWNLOAD SECURITY CREDENTIALS	45
TASK 13: SETUP THE AWS CLI.....	49
TASK 14: MANAGE VPC THROUGH CLI	49

Lab : Virtual Private Cloud(VPC)

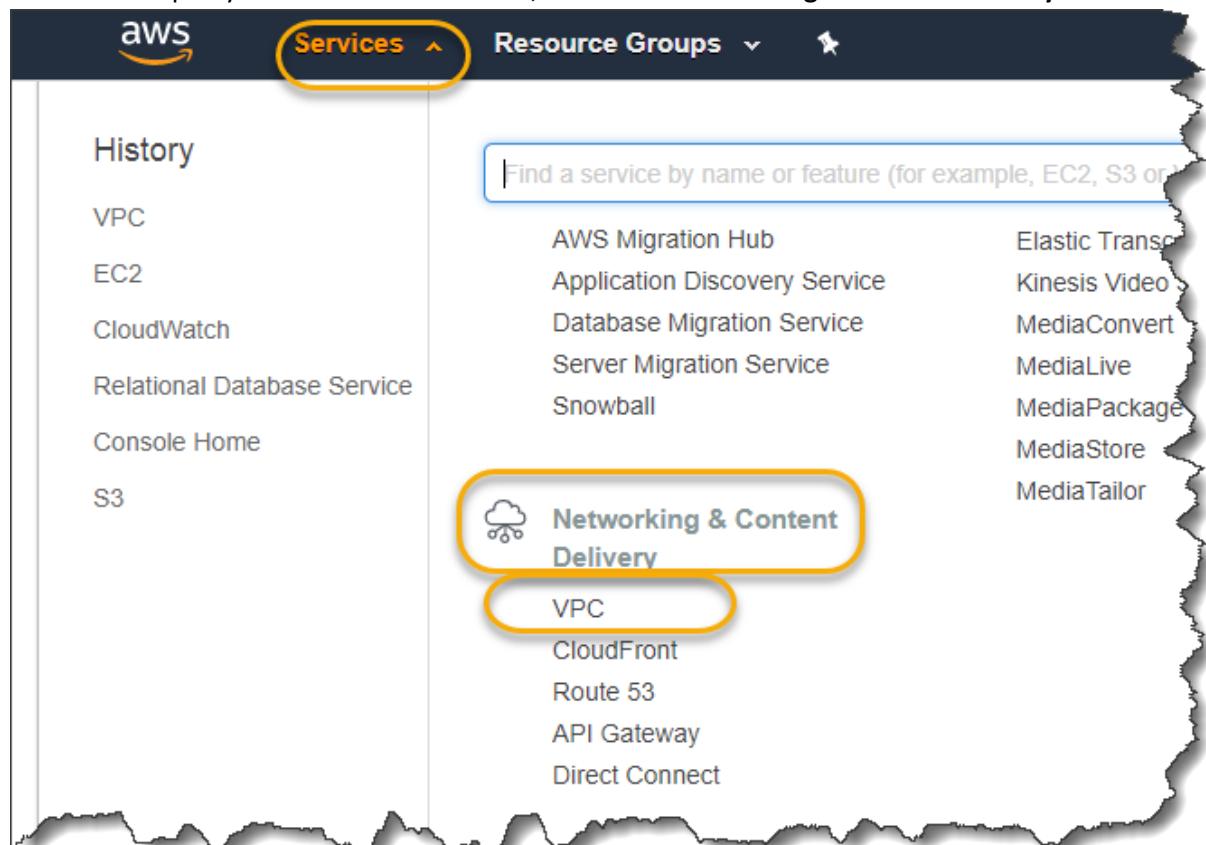
In this lab we will create a VPC, create subnets, configure public and private access, configure network address translation and then perform tests to demonstrate the working of our configuration.

Task Breakdown

- Task 1: Create a VPC with a Public subnet
- Task 2: Create a Private subnet
- Task 3: Launch a web server into the Public subnet
- Task 4: Launch a database server into the Private subnet
- Task 5: Test out accessibility of subnets
- Task 6: Launch a NAT instance.
- Task 7: Modify the private route table to Route traffic to NAT
- Task 8: Download Security Credentials
- Task 9: Setup the AWS CLI
- Task 10: Manage VPC through CLI

Task 1: Create a VPC with a Public subnet

1. On the top of your screen Click **Services**, and under **Networking & Content Delivery** Click **VPC**



2. On the Main **VPC Dashboard** click **Start VPC Wizard**

The screenshot shows the AWS VPC Dashboard. At the top left is a search bar labeled "Select a VPC". To its right are two buttons: "Start VPC Wizard" and "Launch EC2 Instances", with "Start VPC Wizard" being highlighted with a yellow oval. Below these buttons is a note: "Note: Your Instances will launch in the Asia Pacific (Mumbai) region." To the left, there's a sidebar with links for "Virtual Private Cloud", "Your VPCs", "Subnets", "Route Tables", "Internet Gateways", "Egress Only Internet Gateways", and "Options Set". The main area displays a summary of resources: 1 VPC, 0 Egress-only Internet Gateways, 1 Route Table, 1 Elastic IP, 0 Endpoints, 4 Security Groups, 0 VPN Connections, 0 Customer Gateways, 1 Internet Gateway, 2 Subnets, 1 Network ACL, 0 VPC Peering Connections, 0 Nat Gateways, 1 Running Instance, 0 Virtual Private Gateways, and 1 DHCP Options Set.

3. Select the configuration includes a **VPC with a Single Public Subnet**

The screenshot shows the "Step 1: Select a VPC Configuration" screen. On the left, there are three options: "VPC with a Single Public Subnet" (highlighted with a yellow oval), "VPC with Public and Private Subnets", and "VPC with Public and Private Subnets and Hardware VPN Access". To the right, there is descriptive text: "Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances." Below this is a "Creates:" section: "A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet." To the right of this text is a "Select" button (highlighted with a yellow oval) and a diagram. The diagram shows a cloud icon labeled "Internet, S3, DynamoDB, SNS, SQS, etc." connected to a square icon labeled "Public Subnet" with the text "Amazon Virtual Private Cloud" underneath.

4. Enter **192.168.0.0/16** as the **IPv4 CIDR block** of the VPC and enter **HybridSkill-VPC** as the **VPC name**. Enter **192.168.1.0/24** as the **IPv4 CIDR block** of the subnet, select **ap-south-1a (any zone in your region)** as the **Availability Zone** and enter **Public Subnet** as the **Subnet name**. Finally click **Create VPC**

Step 2: VPC with a Single Public Subnet

IPv4 CIDR block: (65531 IP addresses available)

IPv6 CIDR block: No IPv6 CIDR Block Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

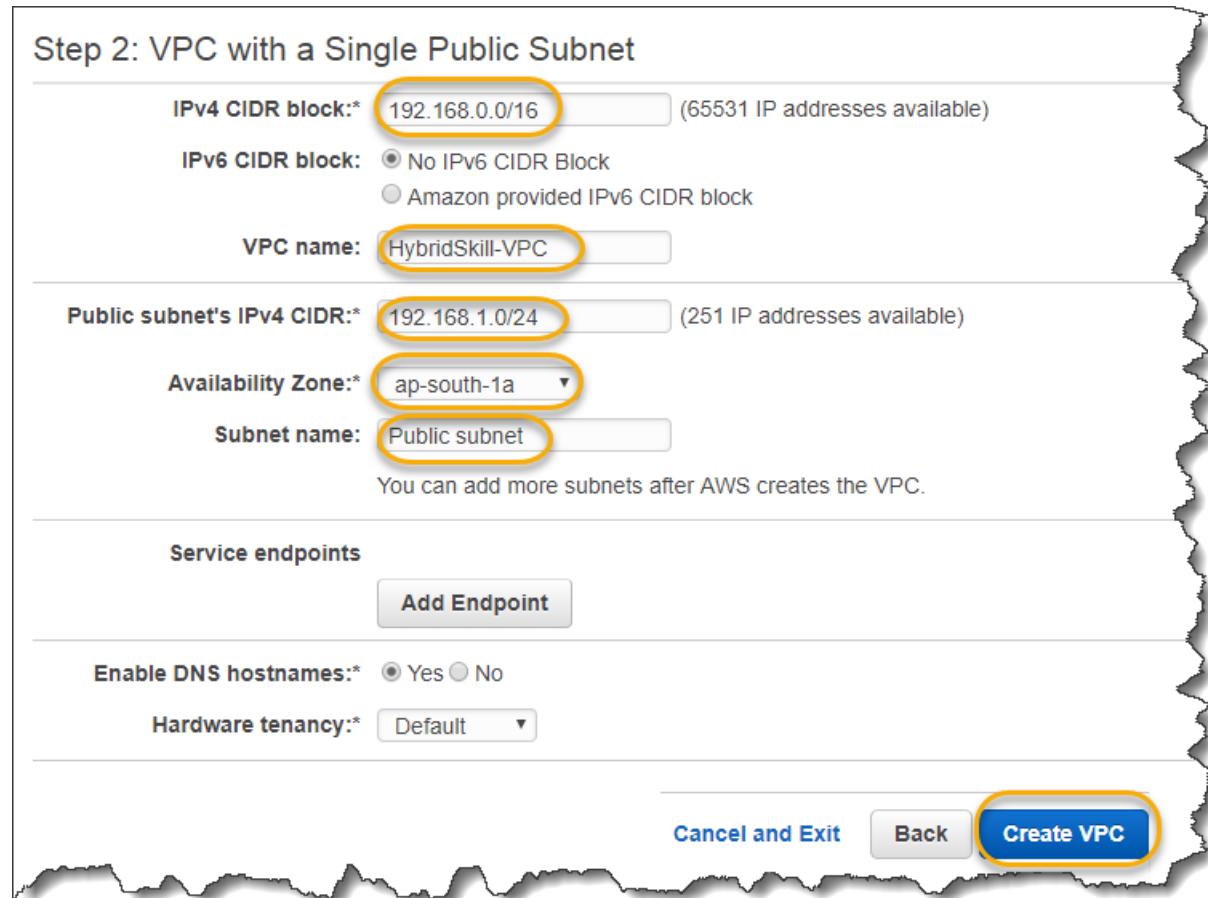
Subnet name:

You can add more subnets after AWS creates the VPC.

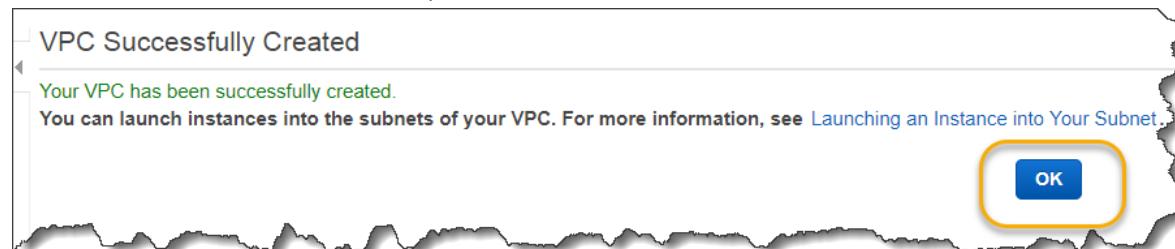
Service endpoints

Enable DNS hostnames: Yes No

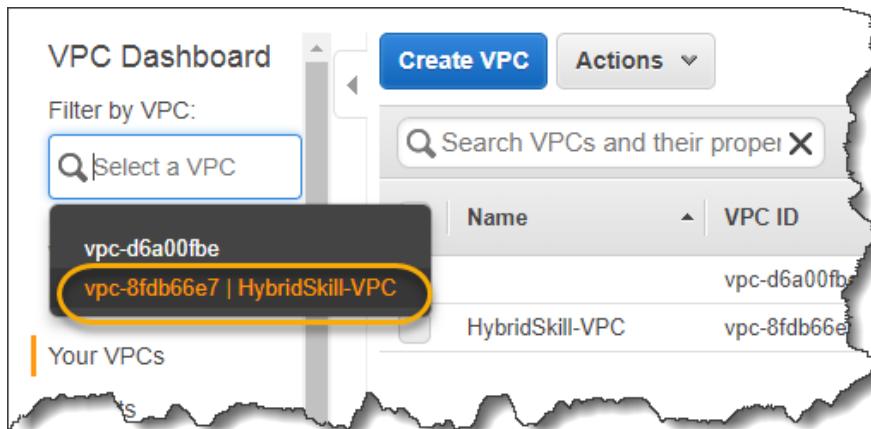
Hardware tenancy:



5. Your VPC is created successfully. Click the **OK** button

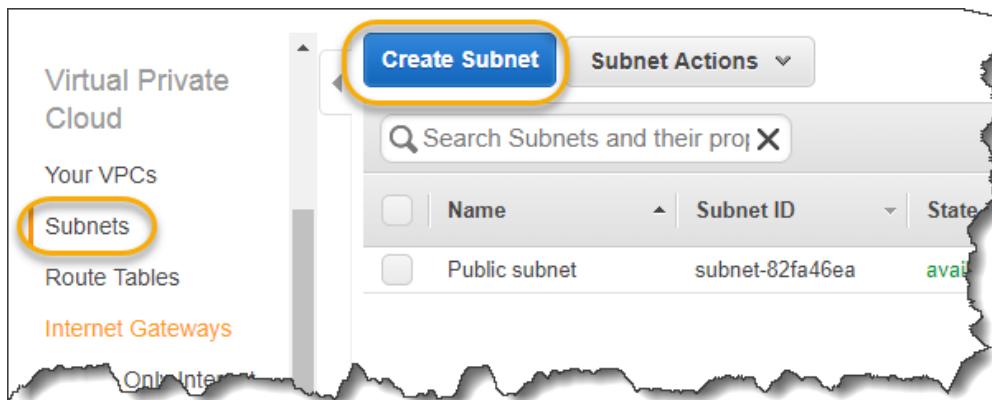


6. In order to prevent confusion between the Default VPC and the one you just created use **the Filter By VPC** option at the top left of your screen. Select Your VPC from the dropdown here.

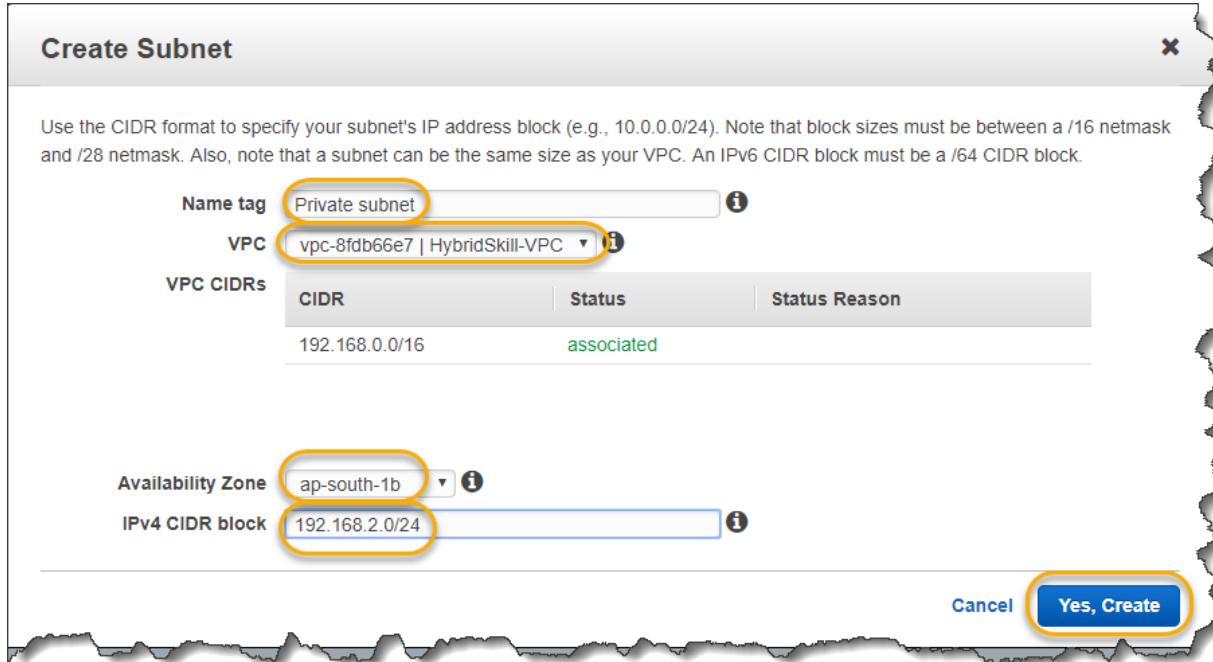


Task 2: Create a Private subnet

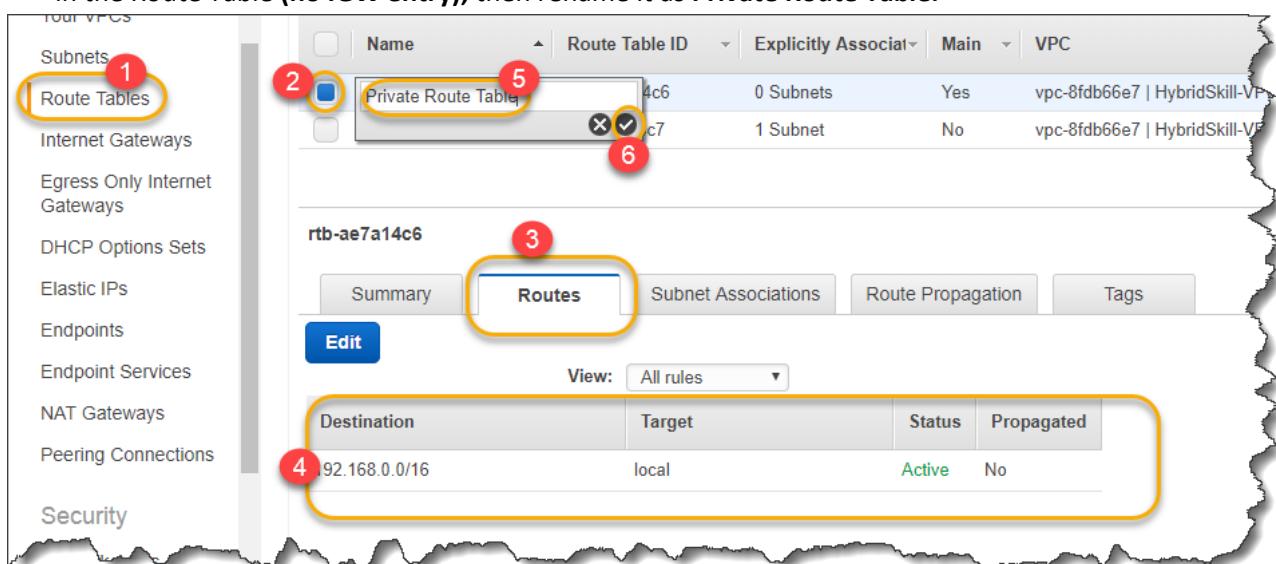
1. Click **Subnets** and then click **Create Subnet**.



2. Enter **Private subnet** as the **Name** of the subnet, select your **VPC(HybridSkill-VPC)** from the dropdown. Select **ap-south-1b (any AZ from your region)** as the **Availability Zone**. Enter **192.168.2.0/24** as the **IPv4 CIDR block** of the subnet and click **Yes, Create**.



3. Next let's review our Route Tables. Rename them as public and Private to prevent confusion. Select **Route Tables**. Select a route table. On the bottom click **Routes** and if see only **one entry** in the Route Table (**no IGW entry**), then rename it as **Private Route Table**.



4. Similarly, if you see **2 routes** in the **Route Table** (**a route to the IGW**) then rename it to **Public Route Table**

The screenshot shows the AWS VPC console. On the left sidebar, under 'Your VPCs', 'Route Tables' is selected and highlighted with a red circle labeled 1. In the main content area, a list of route tables is shown. One route table, 'rtb-af7a14c7', is selected and highlighted with a red circle labeled 2. A modal window is open over the table, showing the current name 'rtb-af7a14c7' and a new name 'Public Route Table'. The new name is highlighted with a red circle labeled 5. A red circle labeled 6 is at the bottom right of the modal, next to a checkmark icon.

Name	Route Table ID	Explicitly Associated	Main	VPC
Private Route Table	rtb-ae7a14c6	0 Subnets	Yes	vpc-8fdb66e7
Public Route Table	rtb-af7a14c7	1 Subnet	No	vpc-8fdb66e7

Below the table, the details for 'rtb-af7a14c7' are shown. The 'Routes' tab is selected and highlighted with a red circle labeled 3. The table lists two routes:

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	igw-fe526c97	Active	No

5. Select **Subnets**, select **Public subnet**, click **Route table** tab at the bottom and make sure it is using the **Public Route Table**.

The screenshot shows the AWS VPC console. On the left sidebar, 'Subnets' is selected and highlighted with a red circle labeled 1. In the main content area, a list of subnets is shown. One subnet, 'Public subnet', is selected and highlighted with a red circle labeled 2. Below the list, the details for 'subnet-82fa46ea | Public subnet' are shown. The 'Route Table' tab is selected and highlighted with a red circle labeled 3. The 'Route Table' dropdown shows 'rtb-af7a14c7 | Public Route Table', which is highlighted with a red box.

Name	Subnet ID	State	VPC
Public subnet	subnet-82fa46ea	available	vpc-8fdb66e7 HybridSkill-VPC
Private subnet	subnet-8bf519c7	available	vpc-8fdb66e7 HybridSkill-VPC

Below the subnet details, the route table configuration is shown:

Destination	Target
192.168.0.0/16	local
0.0.0.0/0	igw-fe526c97

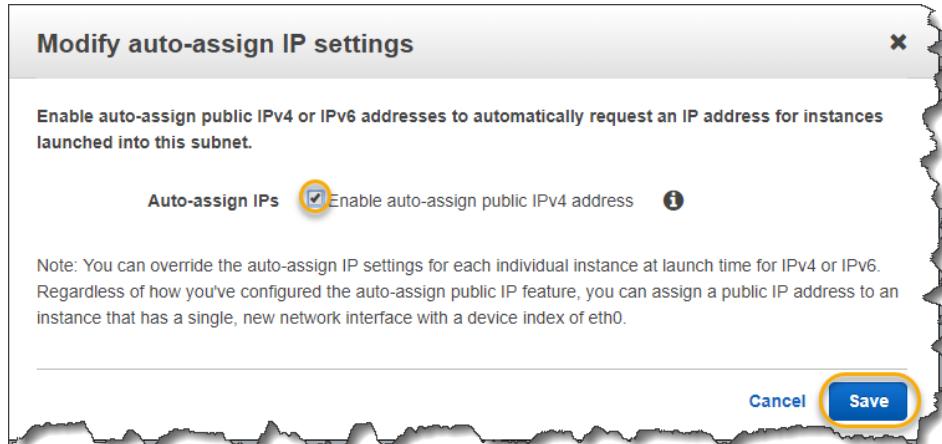
6. Select **Subnets**, select **Private subnet**, click **Route table** tab at the bottom and make sure it is using the **Private Route Table**.

The screenshot shows the AWS VPC Subnets page. On the left sidebar, 'Subnets' is highlighted with a yellow circle. In the main content area, a table lists two subnets: 'Public subnet' and 'Private subnet'. The 'Private subnet' row is selected, indicated by a blue selection bar. Below the table, a modal window for 'subnet-8bf519c7 | Private subnet' is open. The 'Route Table' tab is selected and highlighted with a yellow circle. Inside the modal, the 'Route Table' field shows 'rtb-ae7a14c6 | Private Route Table', which is also highlighted with a yellow circle.

7. Select your **Public Subnet**, click **Subnet Actions** and click **Modify auto-assign IP settings**

The screenshot shows the AWS VPC Subnets page. On the left sidebar, 'Subnets' is highlighted with a yellow circle. In the main content area, a table lists two subnets: 'Public subnet' and 'Private subnet'. The 'Public subnet' row is selected, indicated by a blue selection bar. A context menu is open over the 'Public subnet' row, listing options: 'Delete Subnet', 'Edit IPv6 CIDRs', 'Create Flow Log', and 'Modify auto-assign IP settings'. The 'Modify auto-assign IP settings' option is highlighted with a yellow circle.

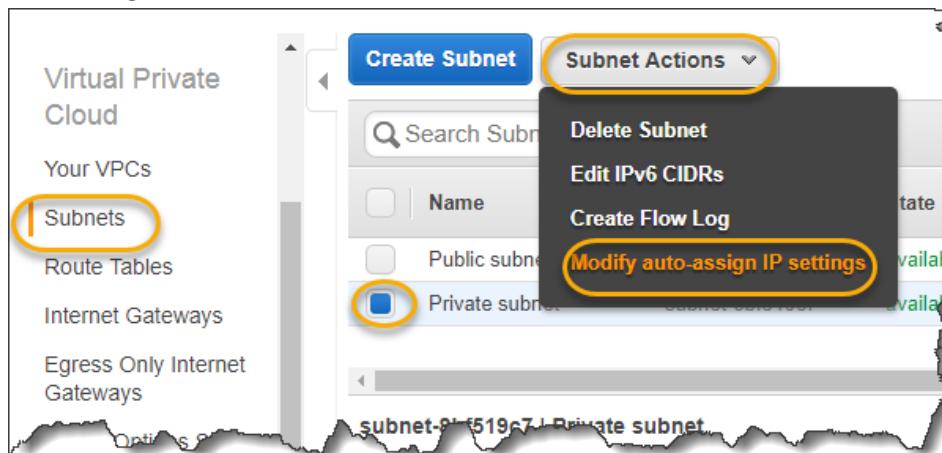
8. Make sure the option of **Enable auto assign public IPv4 address** is checked and click **Save**



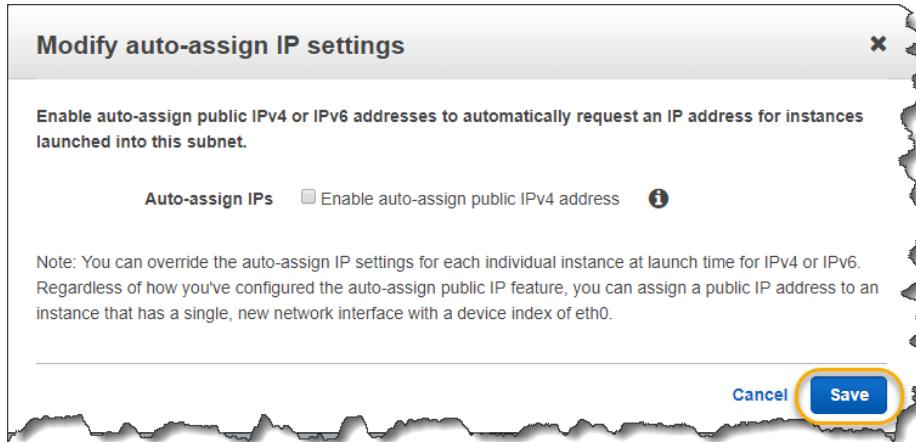
9. Click **Close**



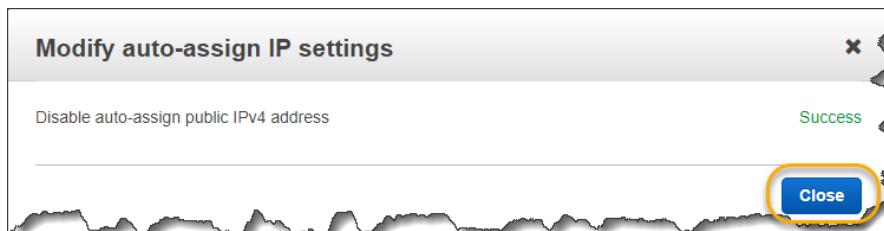
10. Similarly select your **Private Subnet**, click **Subnet Actions** and click **Modify auto-assign IP settings**



11. Make sure the option of **Enable auto assign public IPv4 address** is unchecked and click **Save**

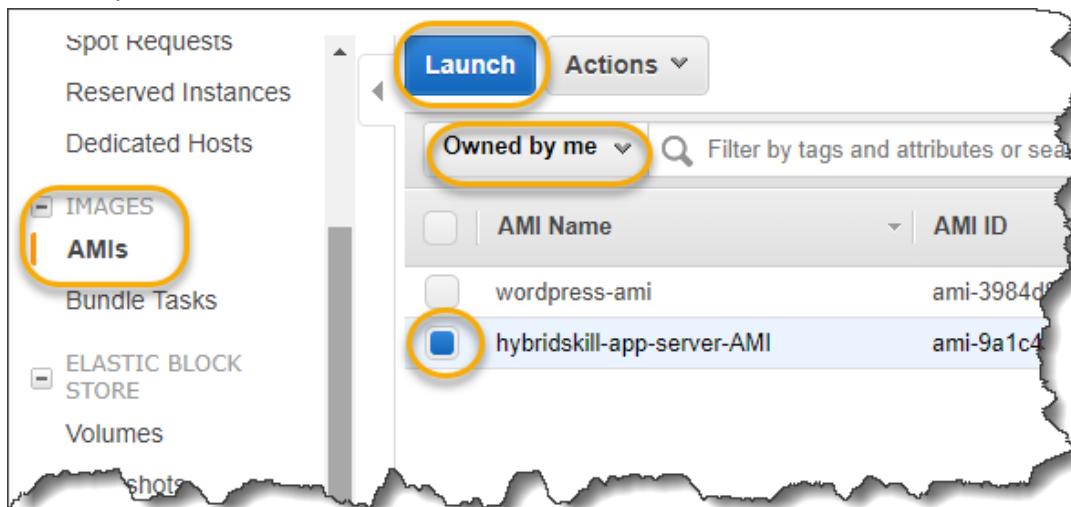


12. Click **Save**



Task 3: Launch a web server into the Public subnet

1. Go the **EC2 Dashboard**. Click **AMIs**, select **Owned by me**, select your **Hybrid-skill-app-server-AMI** you created earlier and click **Launch**.



2. Select the instance type as **t2.micro** and click **Next: Configure Instance Details**

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run your applications, provide compute capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more about instance types](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only

Buttons: Cancel | Previous | **Review and Launch** | **Next: Configure Instance Details**

3. For Network select your **HybridSkill-VPC**, select Public Subnet you created earlier from the dropdown. Enter **192.168.1.4** as the Primary IP of the server and click **Next: Add Storage**

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower price per hour, or use reserved instances, and more.

Number of instances	<input type="text" value="1"/> Launch into Auto Scaling Group				
Purchasing option	<input type="checkbox"/> Request Spot instances				
Network	vpc-8fdbb6e7 HybridSkill-VPC				
Subnet	subnet-82fa46ea Public subnet ap-south-1a 254 IP Addresses available				
Auto-assign Public IP	Use subnet setting (Enable)				
IAM role	None				
Shutdown behavior	Stop				
Enable termination protection	<input type="checkbox"/> Protect against accidental termination				
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>				
Tenancy	Shared - Run a shared hardware instance <small>Additional charges will apply for dedicated tenancy.</small>				
T2 Unlimited	<input type="checkbox"/> Enable <small>Additional charges may apply</small>				
Network interfaces					
Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-82fa46ea	192.168.1.4	Add IP	

Buttons: Add Device | Cancel | Previous | **Review and Launch** | **Next: Add Storage**

4. Click Next: Add Tags

The screenshot shows the 'Step 4: Add Storage' page of the AWS EC2 instance creation wizard. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage (which is highlighted in orange), Add Tags, Configure Security Group, Review and Launch, and Revise. The main content area is titled 'Step 4: Add Storage' with the sub-instruction 'Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes after launching an instance, but not instance store volumes or storage options in Amazon EC2.' Below this, there's a table for defining storage volumes. A single row is shown for the 'Root' volume, which is mounted at '/dev/sda1'. The 'Size (GiB)' is set to 10, and the 'Volume Type' is set to 'General Purpose SSD (GP2)'. The 'IO Performance' dropdown shows '100 / ?'. There is also an 'Add New Volume' button. A note below states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier usage restrictions.' At the bottom, there are 'Cancel', 'Previous', 'Review and Launch' (in blue), and 'Next: Add Tags' (which is highlighted with a yellow box).

5. Enter web-server as the Name tag of the server

The screenshot shows the 'Step 5: Add Tags' page of the AWS EC2 wizard. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags (which is highlighted in orange), and Configure Security Group. The main content area is titled 'Step 5: Add Tags' with the sub-instruction 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.' Below this, there is a table for adding tags. One tag is listed: 'Name' with a value of 'web-server'. There is also a button 'Add another tag' and a note '(Up to 50 tags maximum)'. At the bottom, there are 'Cancel', 'Previous', 'Review and Launch' (in blue), and 'Next: Configure Security Group' (which is highlighted with a yellow box).

6. Enter **vpc-app-server-sg** as Security group name, enter a description. Click on add rule and from the dropdown add **HTTP**, **ICMP-IPv4** and **SSH** as rules. For **SSH** select the source as **My IP** from the dropdown. For **HTTP** leave the source as **Anywhere** and for **ICMP-IPv4** select the source as **My IP**. Finally click on **Review and Launch**.

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. You can also 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 an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: vpc-app-server-sg

Description: vpc-app-server-sg

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP 106.201.55.137/32
HTTP	TCP	80	Anywhere 0.0.0.0/:/0
All ICMP - IPv4	ICMP	0 - 65535	My IP 106.201.55.137/32

Add Rule

Review and Launch

7. Review your settings and click **Launch**

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance.

AMI Details

hybridskill-app-server-AMI - ami-9a1c40f5

Root Device Type: ebs Virtualization type: hvm

Instance Type

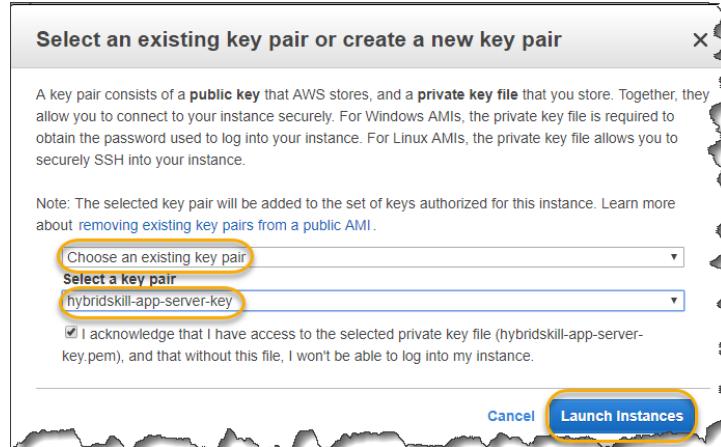
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS Optimized
t2.micro	Variable	1	1	EBS only	

Security Groups

Security group name	Description
vpc-app-server-sg	vpc-app-server-sg

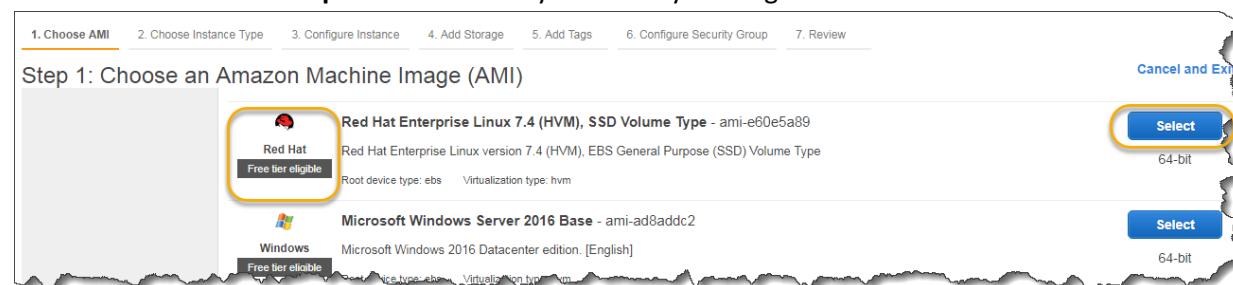
Launch

8. Select **Choose an existing key-pair** select the key you created earlier and Click **Launch Instances**

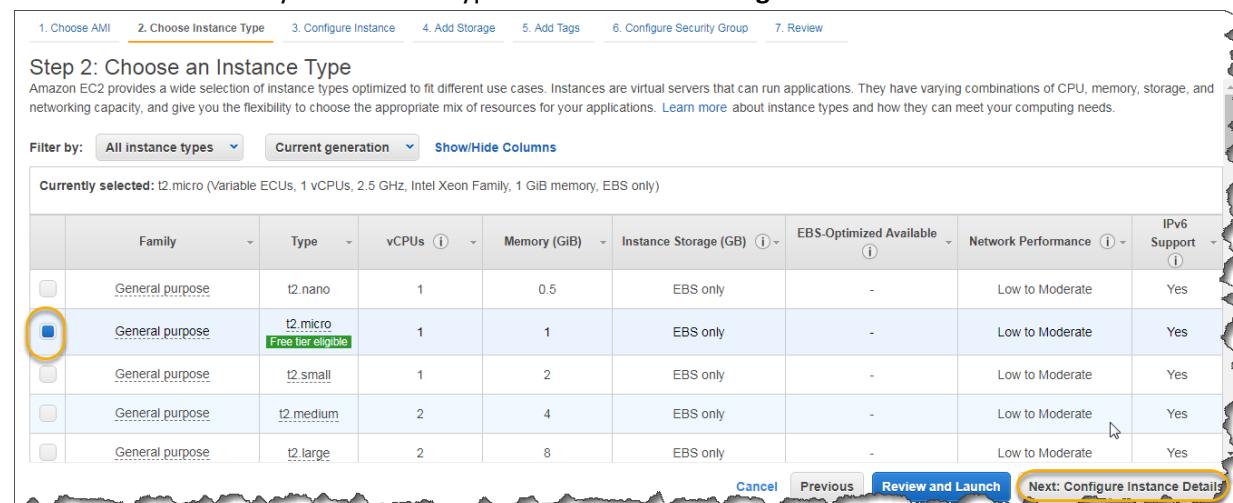


Task 4: Launch a database server into the Private subnet

1. Choose Red Hat Enterprise Linux 7.4 as your AMI by clicking the **Select** button.



2. Select **t2.micro** as your Instance Type and click **Next: Configure Instance Details**



3. For **Network** select your **HybridSkill-VPC**, this time select the **Private Subnet** you created earlier from the dropdown. Enter **192.168.2.4** as the **Primary IP** of the server and click **Next: Add Storage**

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to save costs, and more.

Number of instances	1	Launch into Auto Scaling Group		
Purchasing option	<input type="checkbox"/> Request Spot Instances			
Network	vpc-8fdb66e7 HybridSkill-VPC	Create new VPC		
Subnet	subnet-8bf519c7 Private subnet ap-south-1b	Create new subnet		
Auto-assign Public IP	Use subnet setting (Disable)			
IAM role	None	Create new IAM Role		
Shutdown behavior	Stop			
Enable termination protection	<input type="checkbox"/> Protect against accidental termination			
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>			
Tenancy	Shared - Run a shared hardware instance	<small>Additional charges will apply for dedicated tenancy.</small>		
T2 Unlimited	<input type="checkbox"/> Enable <small>Additional charges may apply</small>			
Network interfaces				
Device	Network Interface	Subnet	Primary IP	Secondary IP addresses
eth0	New network interface	subnet-8bf519c7	192.168.2.4	Add IP
<input type="button" value="Add Device"/> <input type="button" value="Cancel"/> <input type="button" value="Previous"/> <input type="button" value="Review and Launch"/> <input type="button" value="Next: Add Storage"/>				
<input type="button" value="Advanced Details"/>				

4. Leave all the options as default and click on **Next: Add Tags**

Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2](#).

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-08cf78b93268c67e0	10	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
<input type="button" value="Add New Volume"/>								
<small>Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.</small>								
<input type="button" value="Cancel"/> <input type="button" value="Previous"/> <input type="button" value="Review and Launch"/> <input type="button" value="Next: Add Tags"/>								

5. Click **Create** name tag and enter **database-server** as the name of your EC2 instance click **Next: Configure Security Group**

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(127 characters maximum)	Value	(255 characters maximum)
Name		database-server	

Add another tag (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

6. Enter **vpc-db-server-sg** as Security group name, enter a description. Click on add rule and from the dropdown add **ICMP-IPv4** and **SSH** as rules. For **both the rules**, select the source as **Custom** and enter **192.168.1.0/24** as the IP range to allow traffic from. (*You could alternatively set the Source as the security group of the web-server*). Finally click on **Review and Launch**.

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. If you want to 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 an existing Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: vpc-db-server-sg

Description: vpc-db-server-sg

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 192.168.1.0/24
All ICMP - IPv4	ICMP	0 - 65535	Custom 192.168.1.0/24

[Add Rule](#) [Cancel](#) [Previous](#) [Review and Launch](#)

7. Review your settings and click **Launch**

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Free tier eligible	Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-e60e5a89	Edit AMI
	Red Hat Enterprise Linux version 7.4 (HVM), EBS General Purpose (SSD) Volume Type	
	Root Device Type: ebs Virtualization type: hvm	

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name	hybridskill-app-server-sg
Description	created 2018-02-18T22:49:12.538+05:30

[Edit security groups](#)

[Cancel](#) [Previous](#) **Launch**

8. Select **Choose an existing key-pair** select the key you created earlier and Click **Launch Instances**

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

hybridskill-app-server-key

I acknowledge that I have access to the selected private key file (hybridskill-app-server-key.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) **Launch Instances**

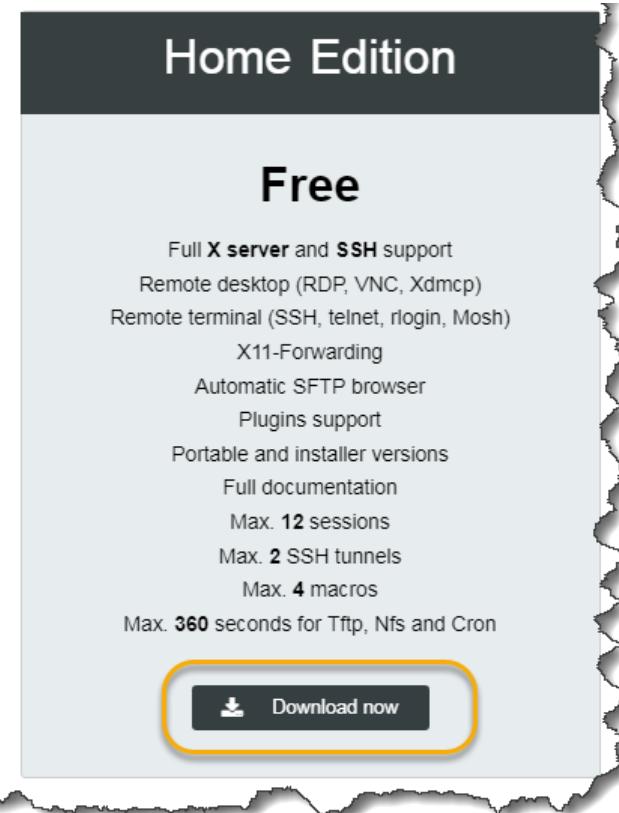
Task 5: Test out accessibility of subnets

1. Open a command prompt and ping the public IP address of the webserver. You should get a response.

1. ping 13.127.205.144
- 2.
3. Pinging 13.127.205.144 **with** 32 bytes of data:
4. Reply from 13.127.205.144: bytes=32 time=52ms TTL=53
5. Reply from 13.127.205.144: bytes=32 time=53ms TTL=53
6. Reply from 13.127.205.144: bytes=32 time=52ms TTL=53
7. Reply from 13.127.205.144: bytes=32 time=53ms TTL=53
- 8.
9. Ping statistics **for** 13.127.205.144:
10. Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
11. Approximate round trip times **in** milli-seconds:
12. Minimum = 52ms, Maximum = 53ms, Average = 52ms

2. Visit the public IP of the web server on browser. You should see the wordpress home page displayed
 3. Log in to the webserver using SSH.
 - . For Mac and Linux users run the following command to connect your instances. Make sure to use your public **IPv4** and your **Key pair** in the command.
- ```
1. chmod 500 hybridskill-app-server-key.pem
2. ssh -i hybridskill-app-server-key.pem ec2-user@13.126.115.88
```

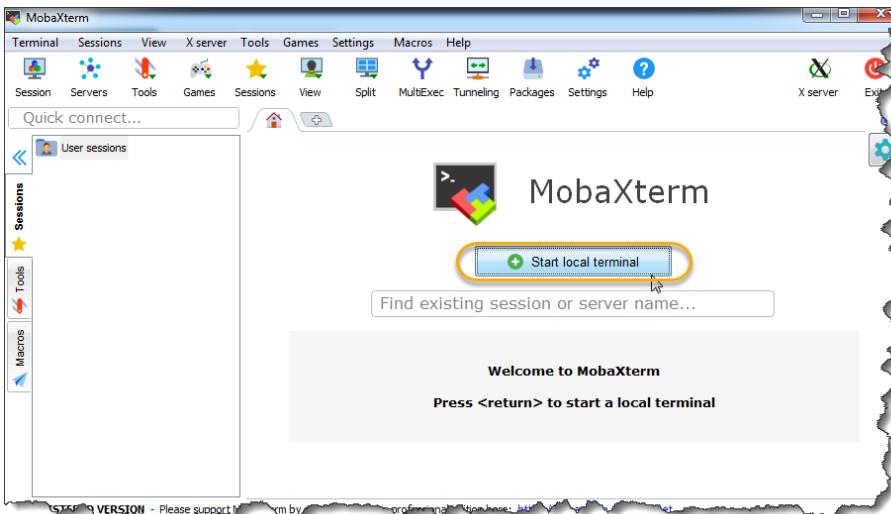
For Windows users we are going to use a SSH client call MobaXterm. Visit this link  
<https://mobaxterm.mobatek.net/download.html> Download the home edition.



Download and install the portable edition.

The screenshot shows the MobaXterm website with a search bar at the top containing "MobaXterm Home Edition". Below the search bar are two download buttons: a blue one labeled "MobaXterm Home Edition v10.5 (Portable edition)" and a green one labeled "MobaXterm Home Edition v10.5 (Installer edition)". Below these buttons, there is a link to "Download previous stable version: MobaXterm Portable v10.4 MobaXterm Installer v10.4". A note below states "By downloading MobaXterm software, you accept MobaXterm terms and conditions". At the bottom, it says "You can download MobaXterm and plugins sources [here](#)".

Click on Start local terminal



Copy your key pair to your desktop. Run the following command to connect your instance.

Make sure to use your public **IPv4** and your **Key pair** in the command

1. cd Desktop
2. chmod 500 hybridskill-app-server-key.pem
3. ssh -i hybridskill-app-server-key.pem ec2-user@13.126.115.88
  
4. Ping google.com. You should get a response.
  1. [ec2-user@ip-192-168-1-4 ~]\$ ping google.com
  2. PING google.com (209.85.202.101) 56(84) bytes of data.
  3. 64 bytes from dg-**inf**101.1e100.net (209.85.202.101): icmp\_seq=1 ttl=36 time=239 ms
  4. 64 bytes from dg-**inf**101.1e100.net (209.85.202.101): icmp\_seq=2 ttl=36 time=239 ms
  5. 64 bytes from dg-**inf**101.1e100.net (209.85.202.101): icmp\_seq=3 ttl=36 time=239 ms
  6. 64 bytes from dg-**inf**101.1e100.net (209.85.202.101): icmp\_seq=4 ttl=36 time=239 ms
  7. 64 bytes from dg-**inf**101.1e100.net (209.85.202.101): icmp\_seq=5 ttl=36 time=239 ms
  8. ^C
  9. --- google.com ping statistics ---
  10. 6 packets transmitted, 5 received, 16% packet loss, time 5006ms
  11. rtt min/avg/max/mdev = 239.837/239.856/239.882/0.438 ms

5. Next try and ping the database server. You should get response too.

1. [ec2-user@ip-192-168-1-4 ~]\$ ping 192.168.2.4
2. PING 192.168.2.4 (192.168.2.4) 56(84) bytes of data.
3. 64 bytes from 192.168.2.4: icmp\_seq=1 ttl=64 time=0.781 ms
4. 64 bytes from 192.168.2.4: icmp\_seq=2 ttl=64 time=0.967 ms
5. 64 bytes from 192.168.2.4: icmp\_seq=3 ttl=64 time=0.893 ms
6. 64 bytes from 192.168.2.4: icmp\_seq=4 ttl=64 time=0.908 ms
7. ^C
8. --- 192.168.2.4 ping statistics ---
9. 4 packets transmitted, 4 received, 0% packet loss, time 3002ms
10. rtt min/avg/max/mdev = 0.781/0.887/0.967/0.070 ms

6. Next log into the database server from your webserver. You will need the key of the database server placed on the webserver for this to happen.

- a. For Windows users, **MobaXterm** has a provision for this. Open a new terminal and run following command use IP of webserver.

```
scp -i hybridskill-key.pem hybridskill-key.pem ec2-user@13.126.115.88:
```

- b. For MAC users you need to use the built in **SCP** command. Replace the IP with the Public IP of your webserver, and hybridskill-key.pem with the path to your key on your system.

You need to

1. scp -i hybridskill-key.pem hybridskill-key.pem ec2-user@13.127.205.144:
2. Warning: Permanently added '13.127.205.144' (RSA) to the list of known hosts.
3. hybridskill--key.pem

6. After the copy is done run the following commands to log into your database instance.

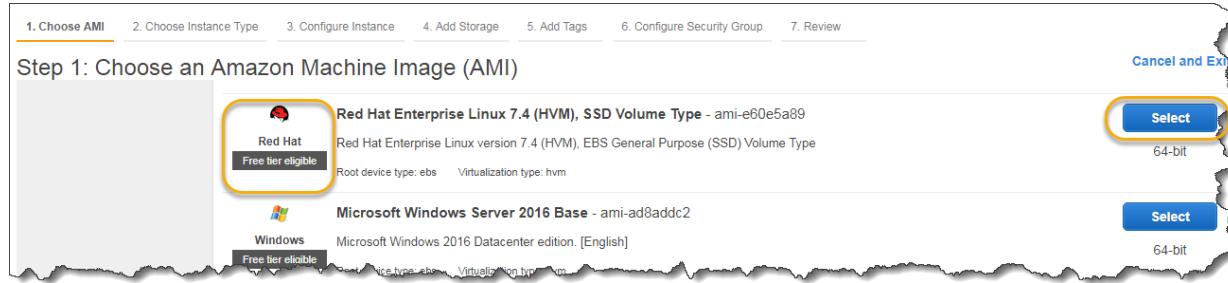
1. [ec2-user@ip-192-168-1-4 ~]\$ sudo chmod 500 hybridskill-app-server-key.pem
2. [ec2-user@ip-192-168-1-4 ~]\$ ssh -i hybridskill-app-server-key.pem ec2-user@192.168.2.4
3. The authenticity of host '192.168.2.4 (192.168.2.4)' can't be established.
4. ECDSA key fingerprint is SHA256:xOjxcpTh0X01v0IBG30BCLnQbCy2O1aFUQppliNEwg4.
5. ECDSA key fingerprint is MD5:10:a3:d7:37:af:de:56:41:40:ba:9d:7d:20:11:b3:d6.
6. Are you sure you want to **continue** connecting (yes/no)? yes
7. Warning: Permanently added '192.168.2.4' (ECDSA) to the list of known hosts.
8. [ec2-user@ip-192-168-2-4 ~]\$

7. Once you are logged into the database server try and ping google.com. You should not be able to get a response.

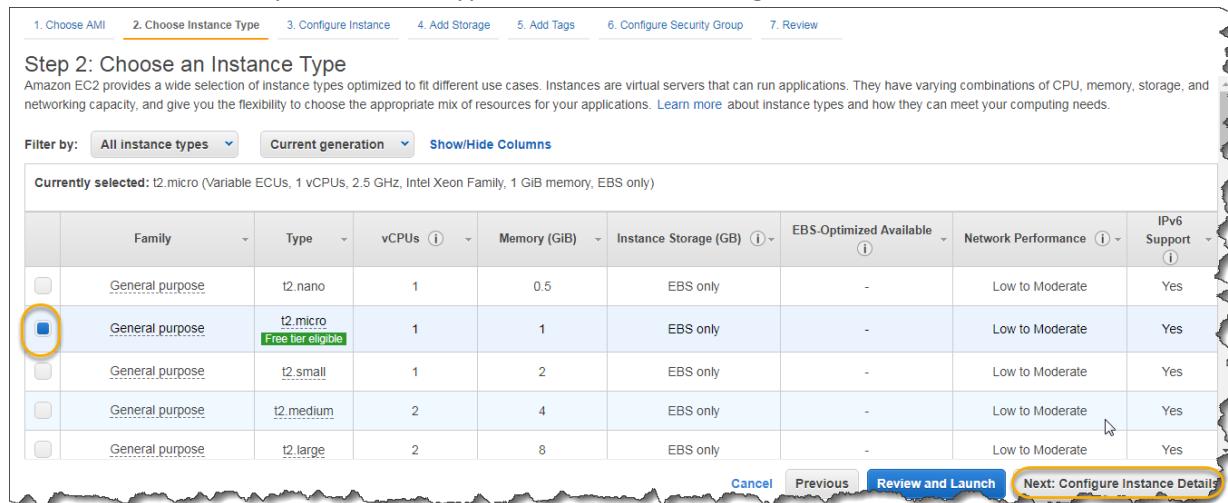
1. [ec2-user@ip-192-168-2-4 ~]\$ ping google.com
2. PING google.com (172.217.27.206) 56(84) bytes of data.
3. ^C
4. --- google.com ping statistics ---
5. 40 packets transmitted, 0 received, 100% packet loss, time 38999ms

## Task 6: Launch a NAT instance.

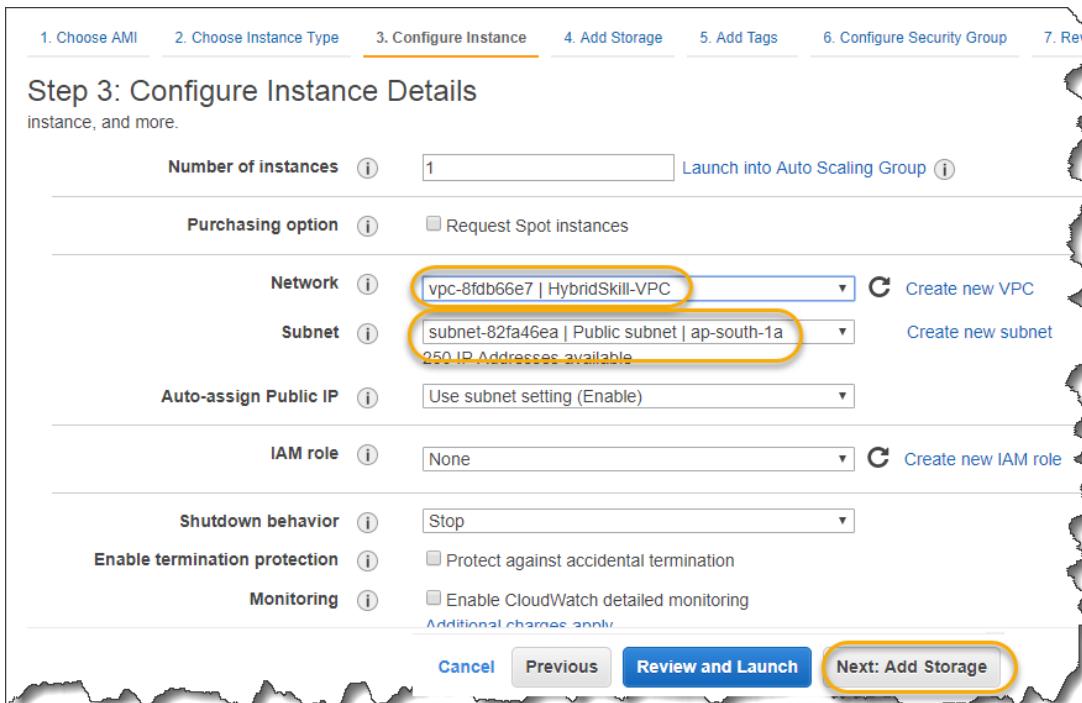
1. Choose Red Hat Enterprise Linux 7.4 as your AMI by clicking the Select button.



2. Select t2.micro as your Instance Type and click Next: Configure Instance Details



3. For Network select your HybridSkill-VPC, select Public Subnet you created earlier from the dropdown. Leave all other options as default and click Next: Add Storage



4. Leave all the options as default and click on **Next: Add Tags**

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

| Volume Type | Device    | Snapshot               | Size (GiB) | Volume Type               | IOPS       | Throughput (MB/s) | Delete on Termination               | Encrypted     |
|-------------|-----------|------------------------|------------|---------------------------|------------|-------------------|-------------------------------------|---------------|
| Root        | /dev/sda1 | snap-08cf78b93268c67e0 | 10         | General Purpose SSD (GP2) | 100 / 3000 | N/A               | <input checked="" type="checkbox"/> | Not Encrypted |

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch **Next: Add Tags**

5. Click **Create** name tag and enter **NAT** as the name of your EC2 instance click **Next: Configure Security Group**

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Web. A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

| Key  | (127 characters maximum) | Value | (255 characters maximum) |
|------|--------------------------|-------|--------------------------|
| Name |                          | NAT   |                          |

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch **Next: Configure Security Group**

6. Enter **NAT-sg** as Security group name, enter a description. Click on add rule and from the dropdown add **ICMP-IPv4** and **SSH** as rules. For **SSH** select the source as **My IP** from the dropdown. For **ICMP-IPv4** select the source as **Custom** and enter **192.168.2.0/24**(IP range of **Private Subnet**) as the IP range to allow traffic from. (*You could alternatively set the Source as the security group of the database-server*). Finally click on **Review and Launch**.

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, 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 an existing Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: **NAT-sg**

Description: **NAT-sg**

| Type            | Protocol | Port Range | Source                     |
|-----------------|----------|------------|----------------------------|
| SSH             | TCP      | 22         | My IP<br>106.200.199.87/32 |
| All ICMP - IPv4 | ICMP     | 0 - 65535  | Custom<br>192.168.2.0/24   |

Add Rule

Cancel Previous Review and Launch

7. Review your settings and click **Launch**

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-e60e5a89

Free tier eligible

Instance Type

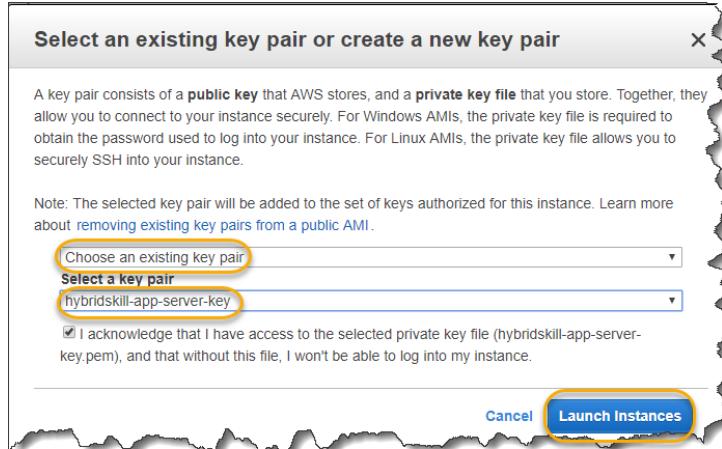
| Instance Type | ECUs     | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|----------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro      | Variable | 1     | 1            | EBS only              | -                       | Low to Moderate     |

Security Groups

hybridskill-app-server-sg  
created 2018-02-18T22:49:12.538+05:30

Cancel Previous Launch

8. Select **Choose an existing key-pair** select the key you created earlier and Click **Launch Instances**



9. Log into the NAT instance and run the following commands to setup NAT

1. sudo sysctl -w net.ipv4.ip\_forward=1
2. sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

## Task 7: Modify the private route table to Route traffic to NAT

1. Go to VPC -> Route Tables, select Private Route Table. Next on the bottom of the screen, Select the Routes Tab and click Edit.

VPC Dashboard

Filter by VPC: vpc-8fdb6...

Virtual Private Cloud

Your VPCs

Subnets

Route Tables (highlighted)

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and their...

| Name                | Route Table ID | Explicitly Associated | Main | VPC                        |
|---------------------|----------------|-----------------------|------|----------------------------|
| Private Route Table | rtb-ae7a14c6   | 0 Subnets             | Yes  | vpc-8fdb66e7   HybridSkill |
| Public Route Table  | rtb-af7a14c7   | 1 Subnet              | No   | vpc-8fdb66e7   HybridSkill |

rtb-ae7a14c6 | Private Route Table

Summary Routes Subnet Associations Route Propagation Tags

Edit (highlighted)

View: All rules

| Destination    | Target | Status | Propagated |
|----------------|--------|--------|------------|
| 192.168.0.0/16 | local  | Active | No         |

2. Click Add another Route. Enter 0.0.0.0/0 as the Destination. For the Target search and select your NAT instance you created earlier. Finally Click Save.

rtb-ae7a14c6 | Private Route Table

Summary Routes Subnet Associations Route Propagation Tags

Cancel Save (highlighted)

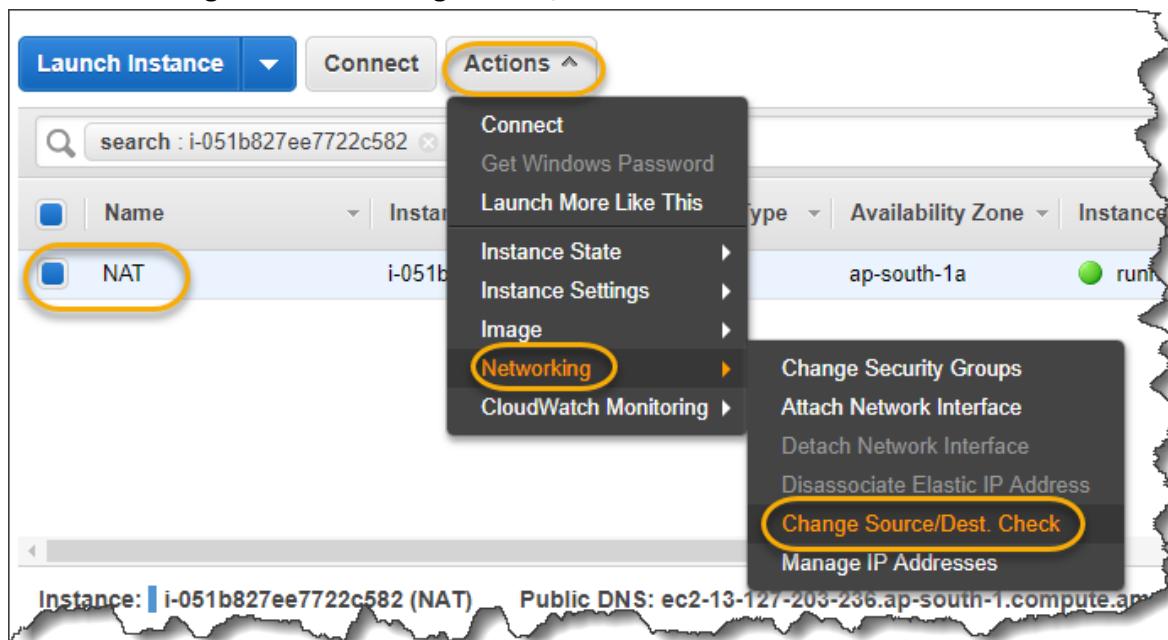
View: All rules

| Destination    | Target            | Status | Propagated | Remove |
|----------------|-------------------|--------|------------|--------|
| 192.168.0.0/16 | local             | Active | No         |        |
| 0.0.0.0/0      | NAT (highlighted) | No     |            |        |

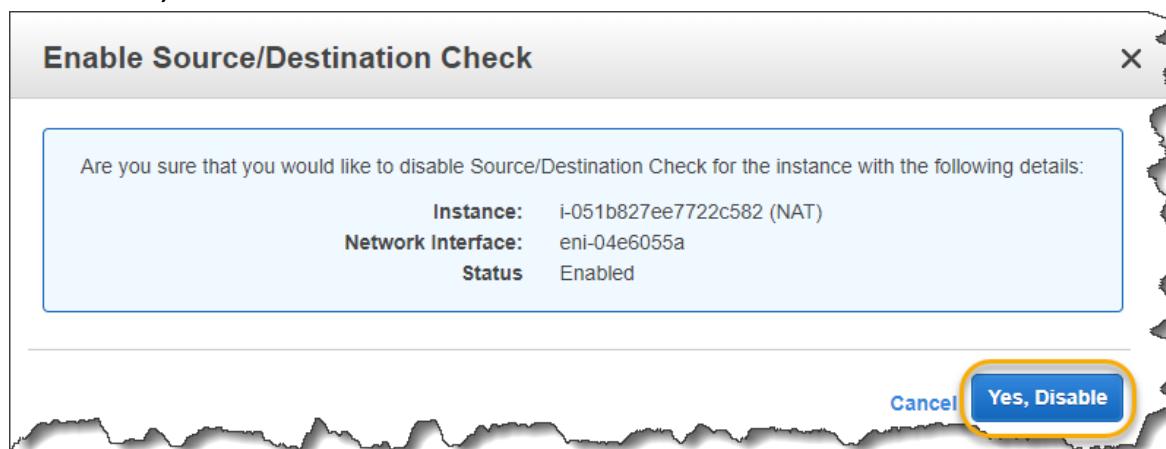
Add another route (highlighted)

i-051b827ee7722c582 | NAT (highlighted)

3. There is one final setting to be done. select your **NAT instance** click **Actions**, scroll down to **Networking** and click on **Change Source/Dest Check**.



4. Click Yes, Disable

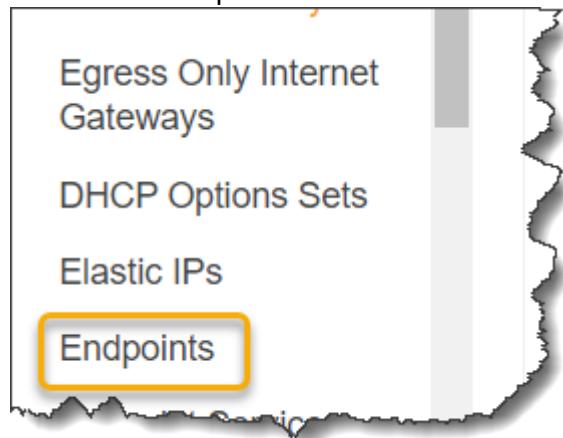


5. Now try and ping google.com from your database server. You should get successful ping.

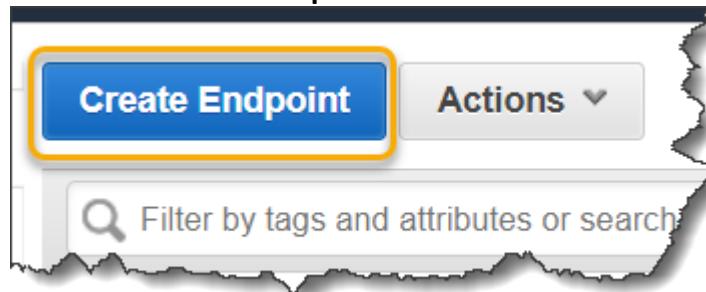
1. ping google.com
2. PING google.com (172.217.27.206) 56(84) bytes of data.
3. 64 bytes from bom07s15-in-f14.1e100.net (172.217.27.206): icmp\_seq=6 ttl=52 time=2.69 ms
4. 64 bytes from bom07s15-in-f14.1e100.net (172.217.27.206): icmp\_seq=7 ttl=52 time=2.63 ms
5. 64 bytes from bom07s15-in-f14.1e100.net (172.217.27.206): icmp\_seq=8 ttl=52 time=2.70 ms
6. 64 bytes from bom07s15-in-f14.1e100.net (172.217.27.206): icmp\_seq=9 ttl=52 time=2.73 ms
7. ^C
8. --- google.com ping statistics ---
9. 9 packets transmitted, 4 received, 55% packet loss, time 8005ms
10. rtt min/avg/max/mdev = 2.636/2.692/2.735/0.036 ms

### Task 8: S3 Endpoint

1. Click on Endpoints on left side of VPC dashboard



2. Click on Create Endpoint



3. Select AWS services, and scroll down and select **com.amazonaws.us-east-1.s3**

Service category  AWS services  
 Find service by name  
 Your AWS Marketplace services

Service Name com.amazonaws.us-east-1.s3

s3

Service Name

com.amazonaws.us-east-1.monitoring

com.amazonaws.us-east-1.perfgamma.kin...

com.amazonaws.us-east-1.qldb.session

com.amazonaws.us-east-1.rekognition-fips

com.amazonaws.us-east-1.s3

4. From drop down select your VPC and check the both route tables.

vpc-0721b88908987d250

A rule with destination **pl-63a5400a** (**com.amazonaws.us-east-1.s3**) and a target with this endpoints' ID (e.g. **vpce-12345678**) will be added to tables you select below.

Subnets associated with selected route tables will be able to access this endpoint.

rtb-0b1d0c0b3ebd85780 rtb-0eb580b95e3d59626

| Route Table ID                                            | Main | Associated With                           |
|-----------------------------------------------------------|------|-------------------------------------------|
| <input checked="" type="checkbox"/> rtb-0b1d0c0b3ebd85780 | Yes  | subnet-0ba61eccb490f1acd   Private subnet |
| <input checked="" type="checkbox"/> rtb-0eb580b95e3d59626 | No   | subnet-0d1d04f26d7824c22   Public subnet  |

## 6. Select Full Access and click on Create endpoint.

The screenshot shows the AWS VPC Endpoint creation wizard. In the top left, there are two radio button options: 'Full Access' (selected) and 'Custom'. A tooltip for 'Full Access' explains: 'Allow access by any user or service within the VPC using credentials from any AWS accounts to any resources in this AWS service. All policies — IAM user policies, VPC endpoint policies, and AWS service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must grant the necessary permissions for access to succeed.' Below this, a note says 'Use the [policy creation tool](#) to generate a policy, then paste the generated policy below.' A code editor window displays a JSON policy template:

```
{
 "Statement": [
 {
 "Action": "*",
 "Effect": "Allow",
 "Resource": "*"
 }
]
}
```

In the bottom right corner of the wizard, there are 'Cancel' and 'Create endpoint' buttons, with 'Create endpoint' being highlighted with a yellow box.

## 7. Now install AWS-cli on this machine by running following commands

```
curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
sudo yum install zip unzip
unzip awscli-bundle.zip
cd awscli-bundle
sudo ./install -i /usr/local/aws -b /usr/local/bin/aws
aws --version
```

## 8. Now configure aws and provide your access key and secret key and your region.

```
aws configure
```

9. Now go to your **Private Route table** click on **Routes** and then on **Edit routes** and remove **Nat instance route rule**, you can also notice that **s3 endpoint route** is also added to your route table now. It will look like below

| Name          | Route Table ID        | Explicit subnet association | Main | VPC ID        |
|---------------|-----------------------|-----------------------------|------|---------------|
| Private Route | rtb-0b1d0c0b3ebd85780 | -                           | Yes  | vpc-0721b8890 |
| Public Route  | rtb-0eb580b95e3d59626 | subnet-0d1d04f26d7824c22    | No   | vpc-0721b8890 |
|               | rtb-7fd53a03          | -                           | Yes  | vpc-774a8c0c  |

Route Table: rtb-0b1d0c0b3ebd85780

Summary    Routes    Subnet Associations    Route Propagation    Tags

Edit routes

View All routes

| Destination                                                            | Target                 |
|------------------------------------------------------------------------|------------------------|
| 10.0.0.0/16                                                            | local                  |
| pl-63a5400a (com.amazonaws.us-east-1.s3, 54.231.0.0/17, 52.216.0.0/15) | vpce-0be900026752927fc |

10. Now login to your DB instance in private subnet and ping google.com as expected it will fail so it is clear that there is no internet connection.

```
[ec2-user@ip-10-0-1-22 ~]$ ping google.com
PING google.com (172.217.164.174) 56(84) bytes of data.
^C
--- google.com ping statistics ---
37 packets transmitted, 0 received, 100% packet loss, time 35999ms
```

11. Now try to list s3 buckets

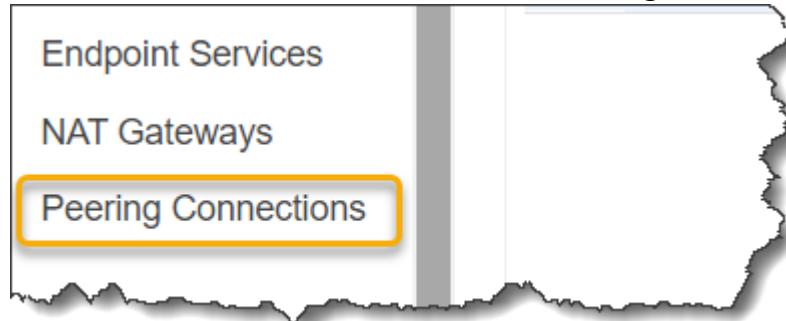
```
aws s3 ls
```

12. As you can see even though there is no internet route even then we are able to list s3 buckets due to endpoint.

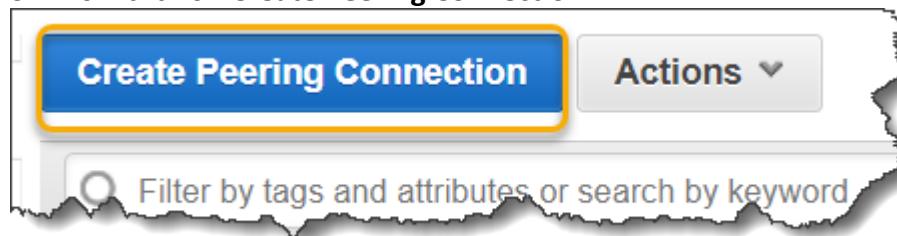
```
[ec2-user@ip-10-0-1-22 ~]$ aws s3 ls
2019-09-04 09:35:35 anthony-hybridskill-cg-01
2018-04-24 08:28:26 anthony-hybridskill-training
2018-02-15 19:42:18 hybridskill
2019-08-14 16:54:57 hybridskill-artifacts
```

### Task 9: VPC Peering

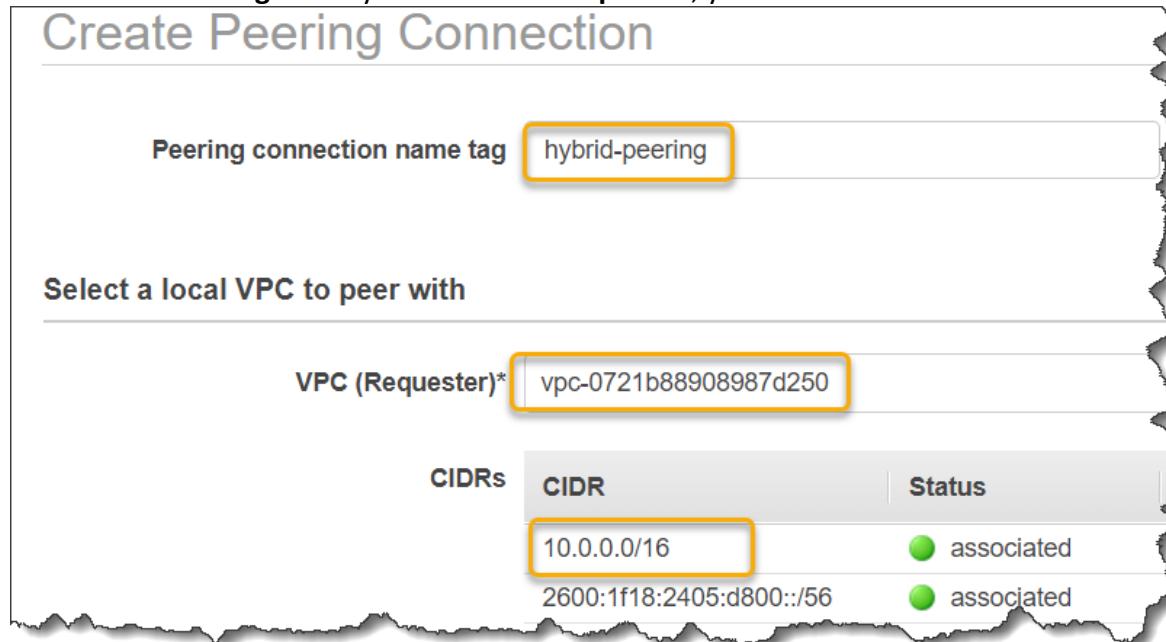
1. Create **another VPC** if you do not have two VPCs or ask your friend for his/her **VPC id**. But make sure that range of both **VPC CIDRs** do not coincide for example if one VPC has **10.0.0.0/16** CIDR other should have different CIDR like **192.168.0.0/16** etc.
2. On left side of VPC dashboard click on **Peering Connections**.



3. Now click on **Create Peering Connection**.



4. Give a **name tag** select your **VPC id** as **Requester**, your **VPC CIDR** can be seen.



5. Choose **Account** in this case My account. Choose **Region** choose **Acceptor VPC** and you can see its **CIDR is different** from Requester CIDR now click on **Create Peering Connection**

VPC to peer with

Account  My account  Another account

Region  This region (us-east-1)  Another Region

VPC (Acceptor)\* vpc-0f52a9a3c7fd12fbc

| CIDRs | CIDR           | Status                                          | Status Reason |
|-------|----------------|-------------------------------------------------|---------------|
|       | 192.168.0.0/16 | <span style="color: green;">●</span> associated |               |

**Create Peering Connection**

6. If accepter VPC was in diff Region/Account visit there and click on **Peering connections** select your **Connection** and click on **Action** and then on **Accept Request**.

Endpoints

Endpoint Services

NAT Gateways

**Peering Connections**

Security

Network ACLs

**Create Peering Connection**

**Actions**

- Accept Request**
- Reject Request
- Delete VPC Peering
- Edit ClassicLink Settings
- Edit DNS Settings
- Add/Edit Tags

7. Click on **Yes Accept**.

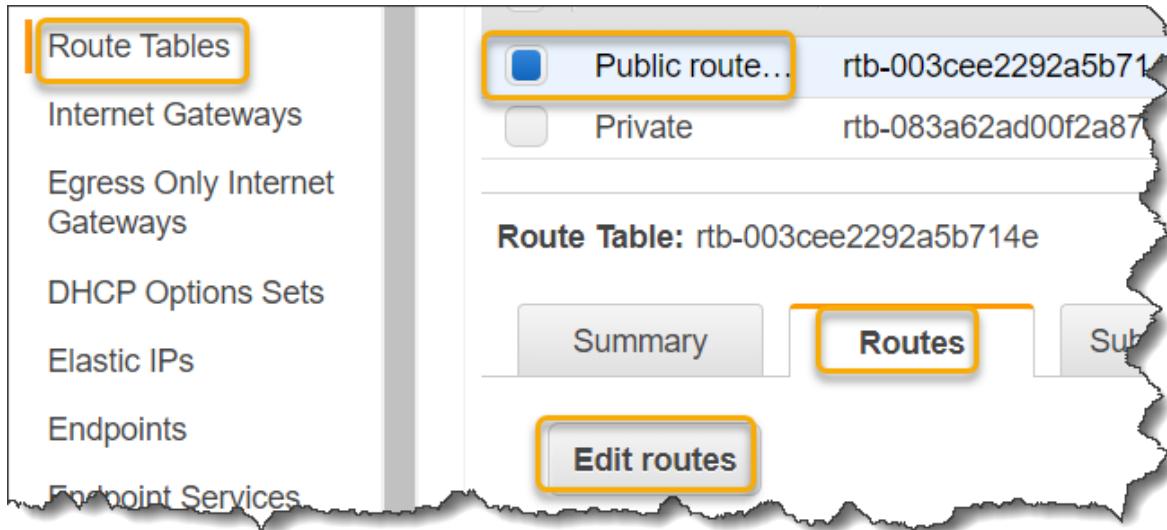
**Accept VPC Peering Connection Request**

Are you sure you want to accept this VPC peering connection request (pcx-0f5a7a2508b37fb59)?

|                      |                             |                     |                             |
|----------------------|-----------------------------|---------------------|-----------------------------|
| Requester Account ID | 111194279774 (This account) | Acceptor Account ID | 111194279774 (This account) |
| Requester VPC ID     | vpc-0721b88908987d250       | Acceptor VPC ID     | vpc-0f52a9a3c7fd12fbc       |
| Requester VPC Region | us-east-1                   | Acceptor VPC Region | us-east-1                   |
| Requester VPC CIDR   | 2 CIDRs                     | Acceptor VPC CIDR   | -                           |

**Cancel** **Yes, Accept**

8. Now as final step we have to add routes to the Route tables for both VPCs click on **Route Tables** on left hand side of VPC dashboard, click one **Public route table**, then click on **Routes** and then on **Edit routes**



9. Click on **Add route** and provide **CIDR of second VPC** in Destination and select **Peering Connection** in target and click on **Save routes**.

| Destination      | Target                | Status |
|------------------|-----------------------|--------|
| 192.168.0.0/16   | local                 | active |
| 0.0.0.0/0        | igw-06d7a7adfe7f83375 | active |
| 10.0.0.0/16      | pcx-0f5a7a2508b37fb59 |        |
| <b>Add route</b> |                       |        |

Buttons at the bottom right: Cancel and Save routes (highlighted with a yellow box).

10. Repeat for all Route Tables.

11. Now do the same for other VPC Route tables also. In destination provide CIDR of first VPC.

| Destination                                                            | Target                 | Status |
|------------------------------------------------------------------------|------------------------|--------|
| 10.0.0.0/16                                                            | local                  | active |
| 2600:1f18:2405:d800::/56                                               | local                  | active |
| pl-63a5400a (com.amazonaws.us-east-1.s3, 54.231.0.0/17, 52.216.0.0/15) | vpce-0be900026752927fc | active |
| 0.0.0.0/0                                                              | igw-0997b1a367c0d3f5e  | active |
| 192.168.0.0/16                                                         | pcx-0f5a7a2508b37fb59  |        |
| <b>Add route</b>                                                       |                        |        |

Buttons at the bottom right: Cancel and Save routes (highlighted with a yellow box). A note at the bottom left says: \* Required

13. Now try to ping one of instance's **private IP** from an instance in other VPC ping should be successful.

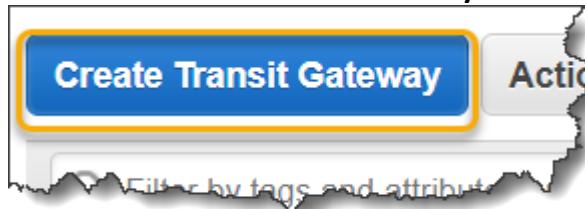
## Task 10: Setup Transit Gateway

To demonstrate a simple example of using a transit gateway, create three VPCs one Dev, one prod and one staging in the same Region. The VPCs cannot have overlapping CIDRs. Launch one EC2 instance in each VPC. Through transit gateway we will make such arrangement that Dev and prod can talk to Staging VPC and vice-versa, but prod and dev will not talk to each other.

1. Go to your VPC dashboard and from left hand side scroll down and select **Transit Gateways**.



2. Click on **Create Transit Gateway**

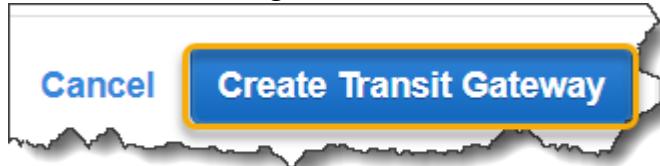


3. Provide a **Name tag** and **ASN** (btw 64512 to 65534)

The screenshot shows the 'Create Transit Gateway' configuration page. It includes fields for 'Name tag' (set to 'hybrid-tg') and 'Description' (set to 'hybrid-tg'). Below this, there's a section titled 'Configure the Transit Gateway' with a field for 'Amazon side ASN' (set to '64512') and an information icon (info icon).

|                 |           |
|-----------------|-----------|
| Name tag        | hybrid-tg |
| Description     | hybrid-tg |
| Amazon side ASN | 64512     |

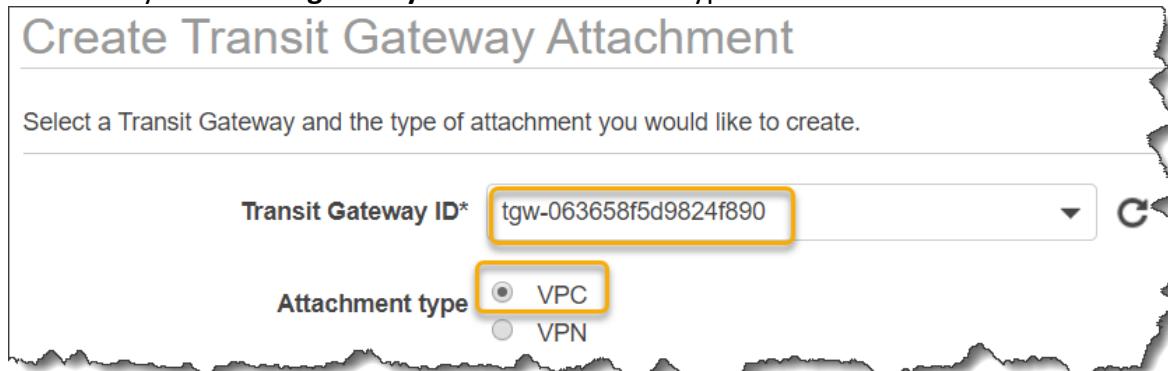
4. Leave other things as default and scroll down and click Create Transit Gateway



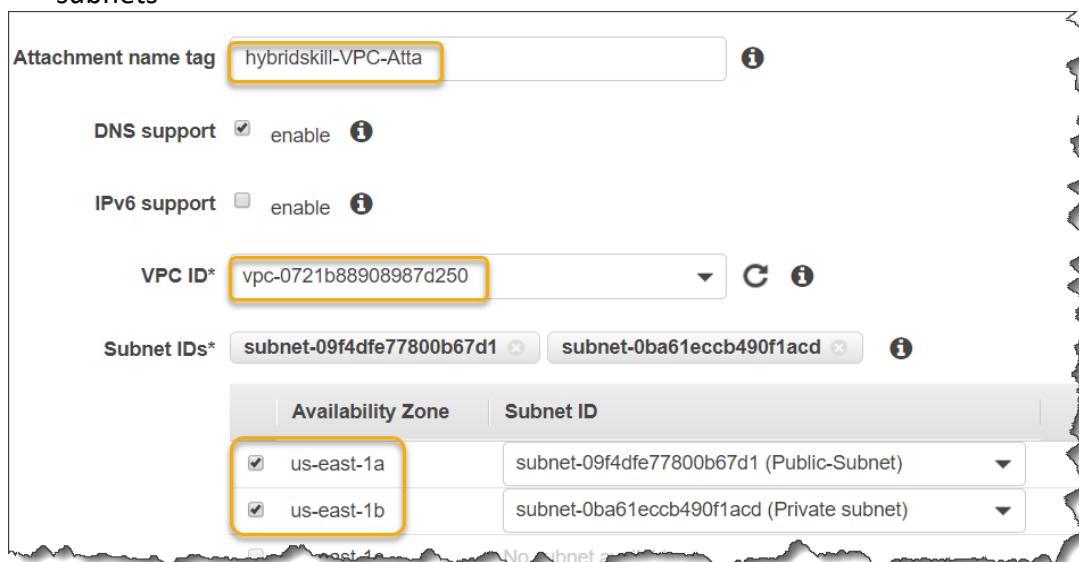
5. After you see the message **Create Transit Gateway request succeeded**, choose **Close**.  
The initial state of the transit gateway is **pending**. Let it become **available**.  
6. Now from left hand side scroll down and select **Transit Gateway Attachments**



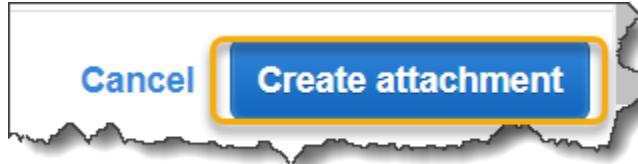
7. Select your **Transit gateway ID** and Attachment type as **VPC**.



8. Give a **name tag** select your **VPC ID** and select **Availability Zones** in which you have subnets



9. Leave other things as default scroll down and click on **Create attachment**.

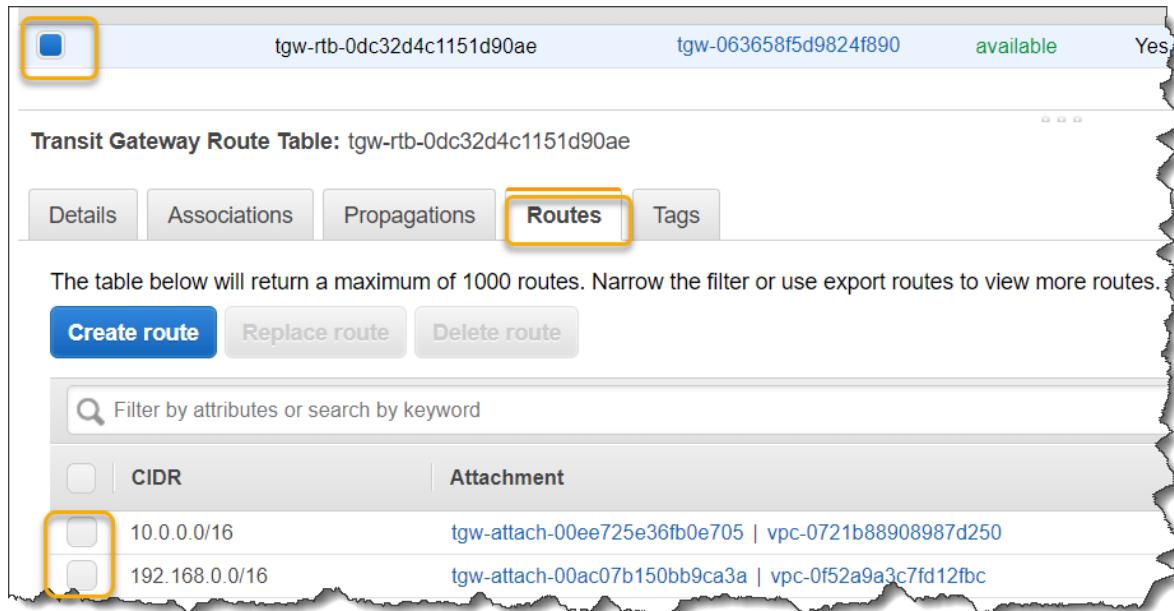


10. Make another attachment for the other VPC also.

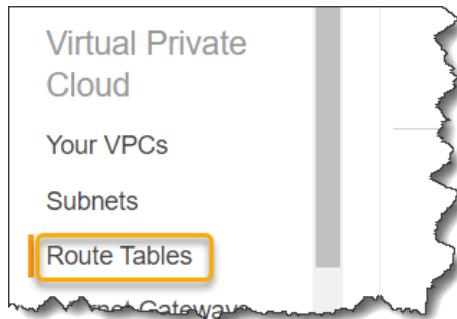
11. Now click on transit **Gateway Route Tables**.



12. Now select the **Route table** click on **Routes** we can see both **VPC CIDR** are added to the Routes.



14. Now we have to edit routes to Route Tables of VPCs, click **Route Tables** on left side of VPC Dashboard



15. So that you do not get confused from filter select your VPC and click on one of the **Route Tables**, then click on **Routes** and then on **Edit routes**

VPC Dashboard  
Filter by VPC:  
vpc-0721b...  
Virtual Private Cloud  
Your VPCs  
Subnets  
Route Tables  
Internet Gateways  
Egress Only Internet Gateways  
DHCP Options Sets

Create route table Actions ▾

| Name          | Route Table ID        | Explicit subnet |
|---------------|-----------------------|-----------------|
| Private Route | rtb-0b1d0c0b3ebd85780 | -               |
| Public Route  | rtb-0eb580b95e3d59626 | -               |

Route Table: rtb-0b1d0c0b3ebd85780

Summary Routes Subnet Associations Routes

Edit routes

16. Click on **Add route** provide other **VPC CIDR** select target as **Transit gateway** and click on **Save routes**, Repeat for all route tables in both VPCs.

- For prod- VPC route tables provide staging VPC CIDR
- For dev-VPC route tables provide staging VPC CIDR
- For staging-VPC route tables provide both Prod as well as dev. VPC CIDR

|                          |                       |        |
|--------------------------|-----------------------|--------|
| 10.0.0.0/16              | local                 | active |
| 2600:1f18:2405:d800::/56 | local                 | active |
| 192.168.0.0/16           | tgw-063658f5d9824f890 | active |

Add route \* Required Cancel Save routes

17. Now try to ping **Private IP** of one of instances in **Staging** VPC from instances in prod and dev VPC it will work and vice-versa.

18. But when u try ping **Dev VPC** instance from **Prod VPC** instance and vice-versa it will **fail**.

### Task 11: VPC Private Links

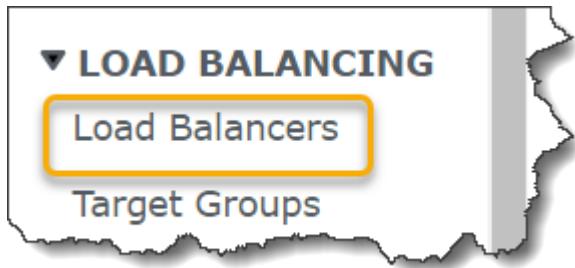
- Create two VPCs with a public subnet in a same availability zone name one as **Producer** and one as **Receiver**
- In **Producer** VPC create webserver in Public Subnet, In **Security Group** open TCP port 80 to receive inbound traffic from any IP and SSH port 22 to your IP.
- Now login into the VM and install HTTP webserver by running following commands and provide some **Text** in **index.html** like 'this is webserver'

```
sudo yum install -y httpd
sudo service httpd start
cd /var/www/html/
sudo vi index.html
```

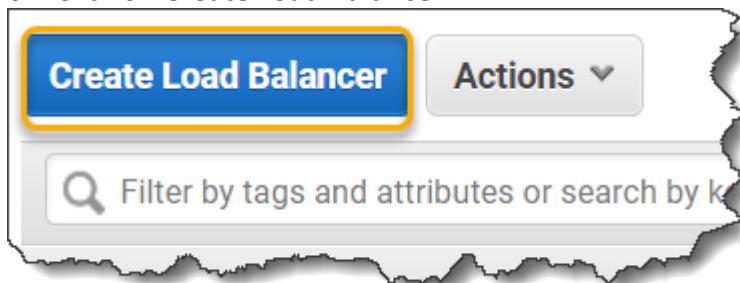
4. Now open Public IP of VM in a new browser tab you should be able to see your Text.



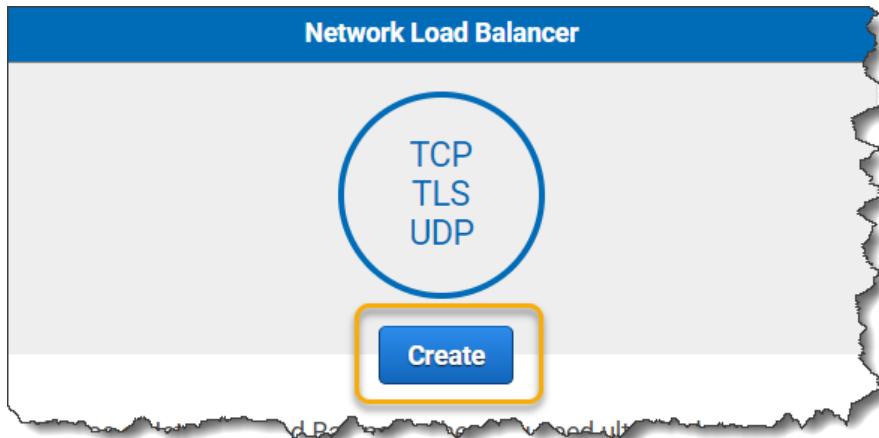
5. On left hand side of EC2 console click on **Load Balancers** under Load Balancing



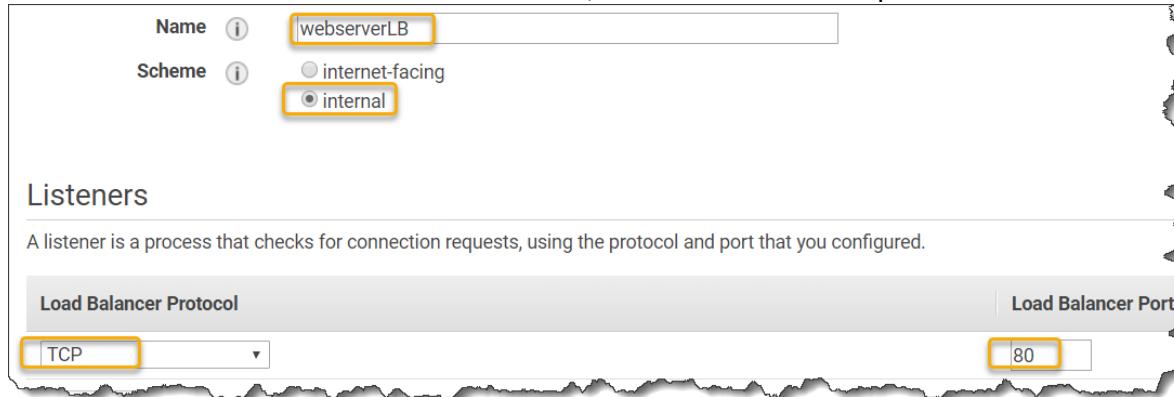
6. Click on **Create Load Balancer**



7. Under Network Load Balancer click on **Create**

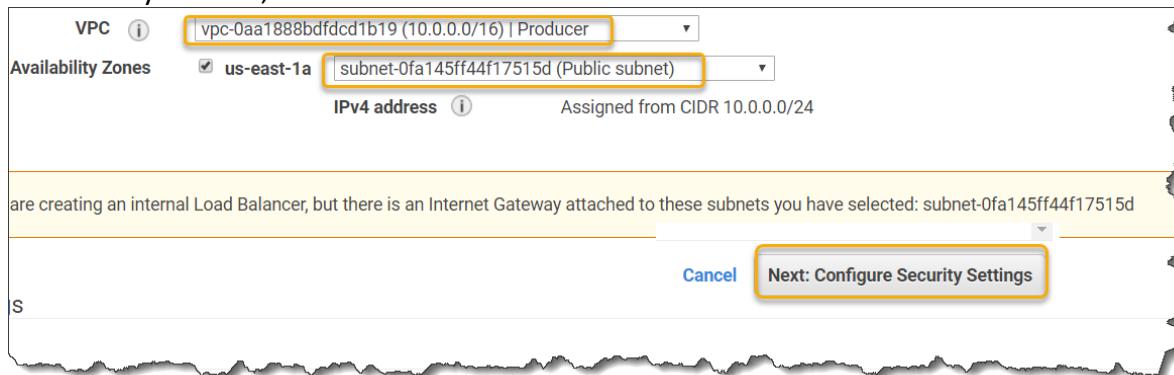


8. Give it a **Name** select Scheme as **internal**, add a **TCP** listener at port **80**



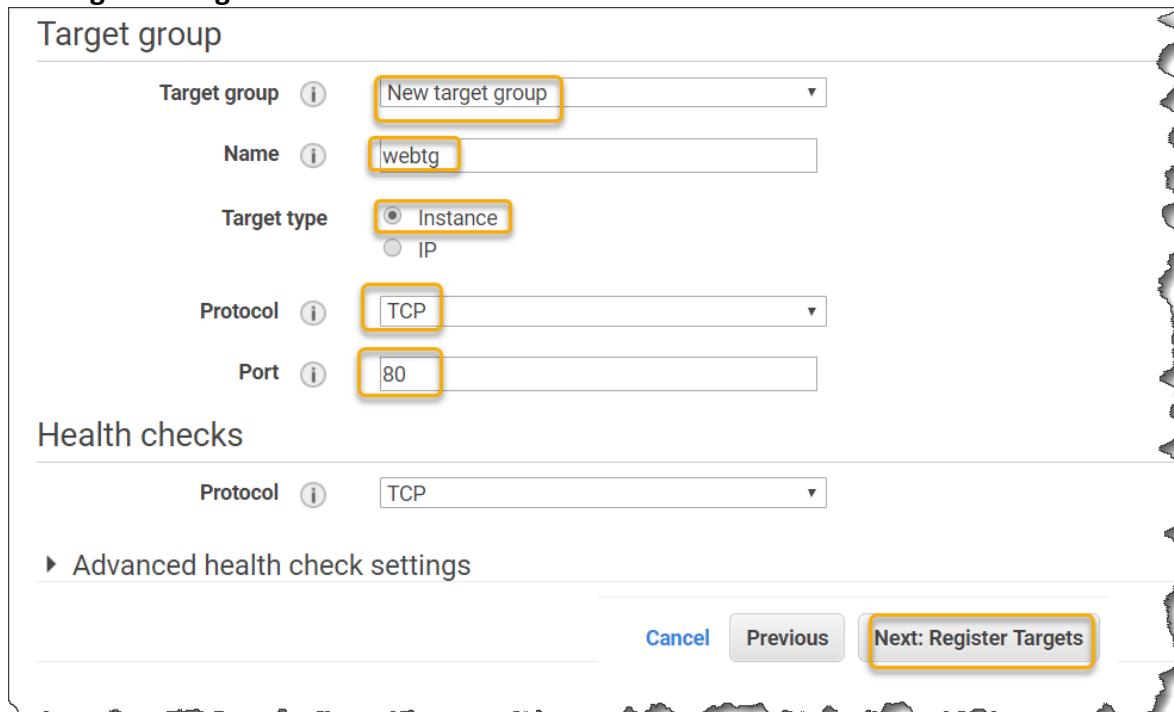
The screenshot shows the 'Listeners' section of the AWS Load Balancer configuration. It includes fields for 'Load Balancer Protocol' (set to 'TCP') and 'Load Balancer Port' (set to '80'). The 'Scheme' field is set to 'internal'. A note below states: 'A listener is a process that checks for connection requests, using the protocol and port that you configured.'

9. Select your **VPC, Subnet** and click on **Next:**



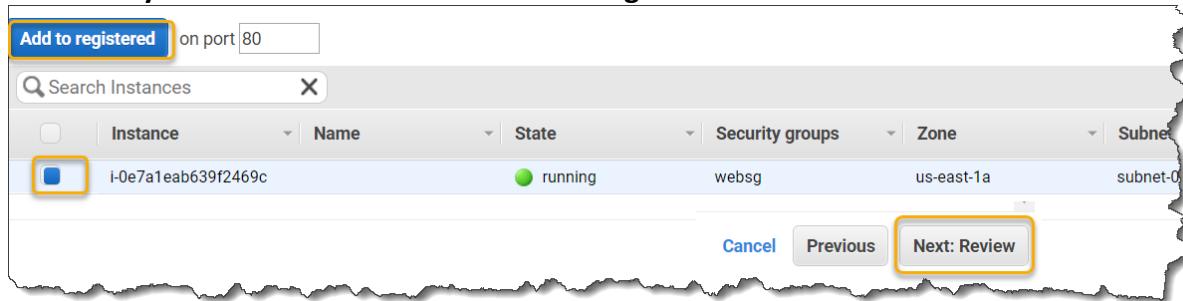
The screenshot shows the 'VPC' configuration step. It lists the VPC ('vpc-0aa1888bdfcd1b19 (10.0.0.0/16) | Producer'), Availability Zones ('us-east-1a'), and IPv4 address ('Assigned from CIDR 10.0.0.0/24'). A note at the bottom states: 'are creating an internal Load Balancer, but there is an Internet Gateway attached to these subnets you have selected: subnet-0fa145ff44f17515d'. The 'Next: Configure Security Settings' button is highlighted.

10. Click on **Next: Create Target Group** provide a **Name**, type as **instance** and click on **Next: Register Targets**

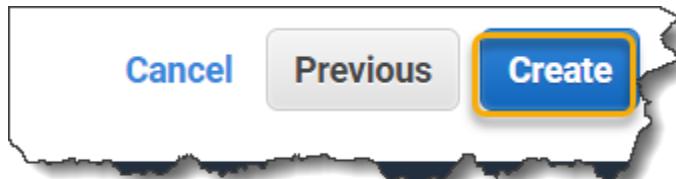


The screenshot shows the 'Target group' creation step. It includes fields for 'Target group' (set to 'New target group'), 'Name' (set to 'webtg'), 'Target type' (set to 'Instance'), 'Protocol' (set to 'TCP'), and 'Port' (set to '80'). Below this, the 'Health checks' section shows 'Protocol' (set to 'TCP'). The 'Next: Register Targets' button is highlighted.

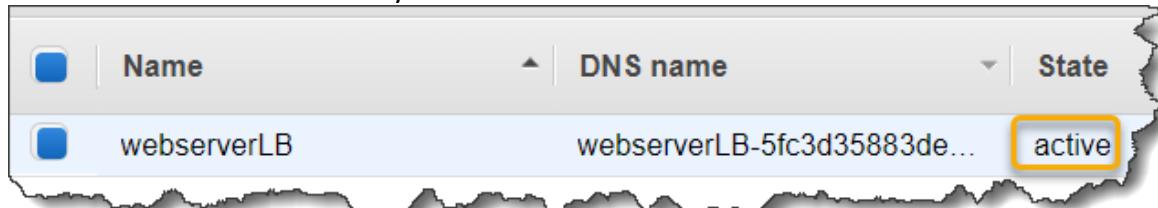
11. Select your instance then click on **Add to registered** and then on **Next: Review**



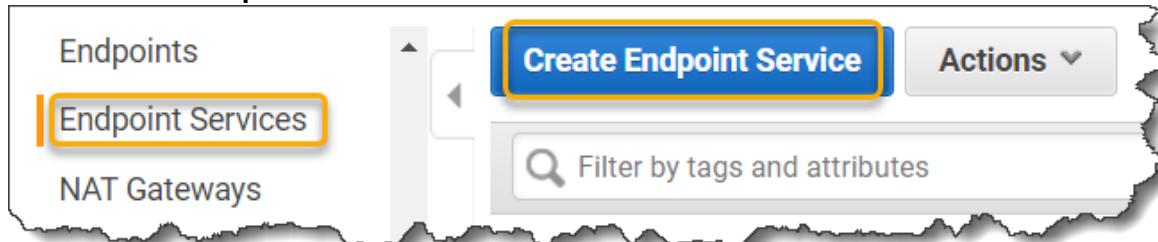
12. Review and scroll below and click on **Create**.



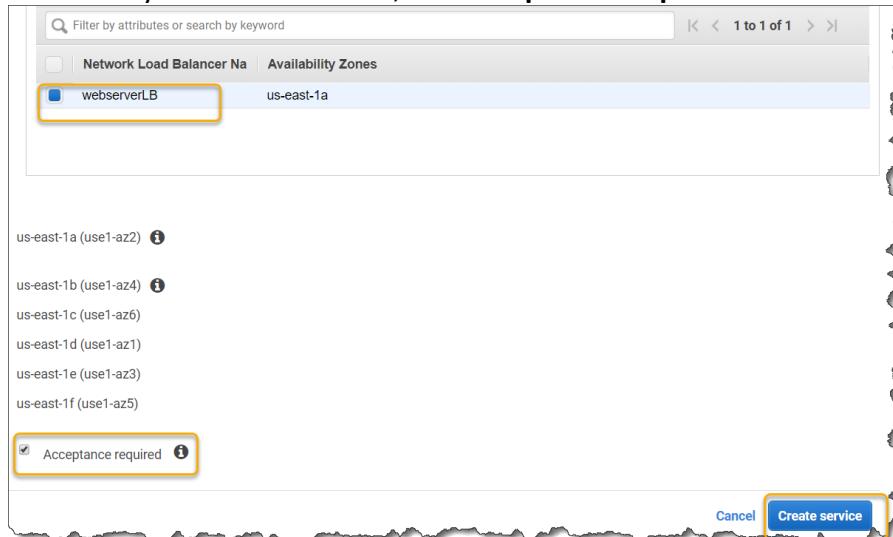
13. Wait till LB is **active**. It may take few minutes.



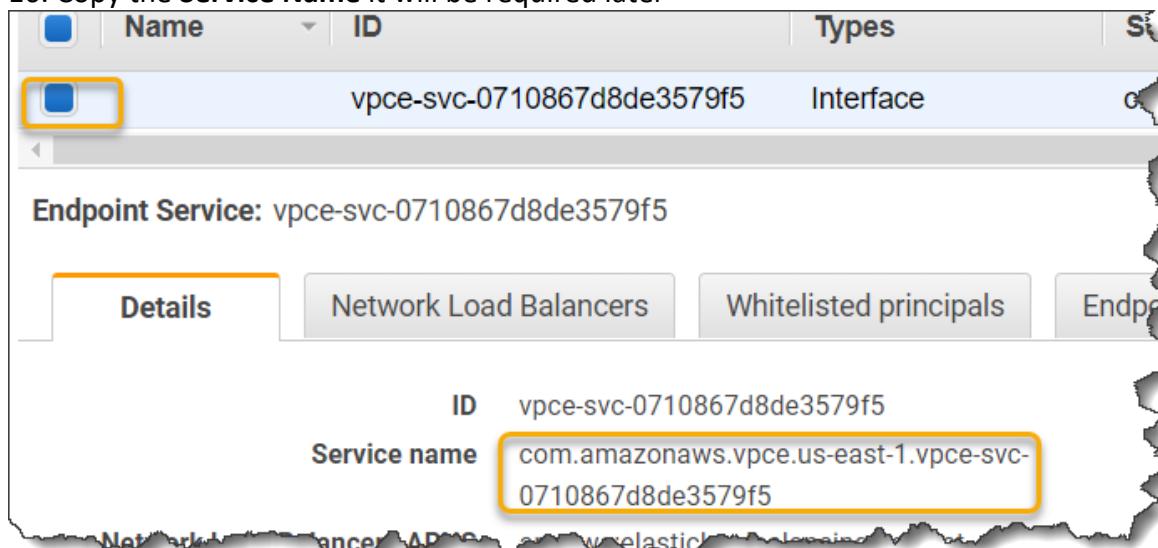
14. From left hand side of VPC console scroll down and click on **Endpoint Services** then click on **Create Endpoint Service**



15. Select your **Load balancer**, tick **Acceptance required** and click on **create service**.

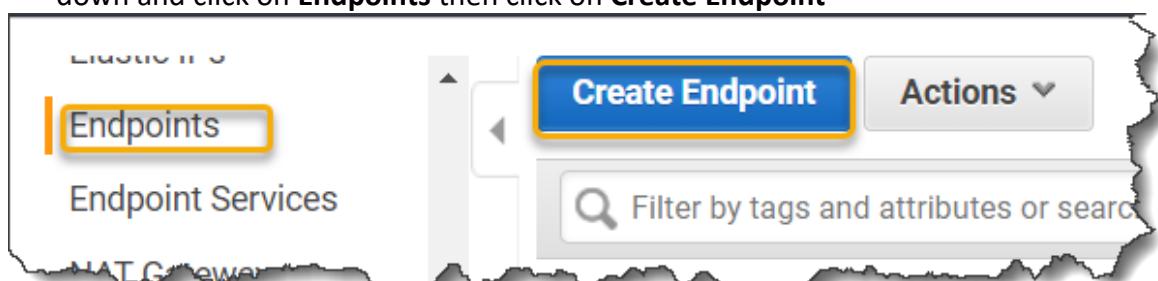


16. Copy the **Service Name** it will be required later



17. In consumer Launch **another instance**.

18. While instance is being launched go to VPC, from left hand side of VPC console scroll down and click on **Endpoints** then click on **Create Endpoint**



19. Select Find by service name and paste service name copied earlier and click on verify in VPC select Receiver VPC

Service category  AWS services  Find service by name  Your AWS Marketplace services

Service Name Enter private service name and verify. [i](#)

com.amazonaws.vpce.us-east-1.vpce-svc-0710867d

Service name found.

Verify

VPC\* vpc-09a239e7b828a4df4 [C](#) [i](#)

Subnets subnet-06753571d62506be4 [i](#)

20. Create a new Security group which allow TCP at port 80 from anywhere and add it to the Endpoint and click on Create endpoint

sg-02be3c2e603e03d03 [X](#) [Create a new security group](#) [i](#)

Select security groups [▲](#)

| Group ID                                           | Group Name      | VPC ID         | Description | Owner ID         |
|----------------------------------------------------|-----------------|----------------|-------------|------------------|
| <input checked="" type="checkbox"/> sg-02be3c2e... | newsg           | vpc-09a239e... | EC2-VPC     | newsg            |
| <input type="checkbox"/> sg-031ac77a...            | launch-wizard-2 | vpc-09a239e... | EC2-VPC     | launch-wizar...  |
| <input type="checkbox"/> sg-0974b92a...            | default         | vpc-09a239e... | EC2-VPC     | default VPC s... |

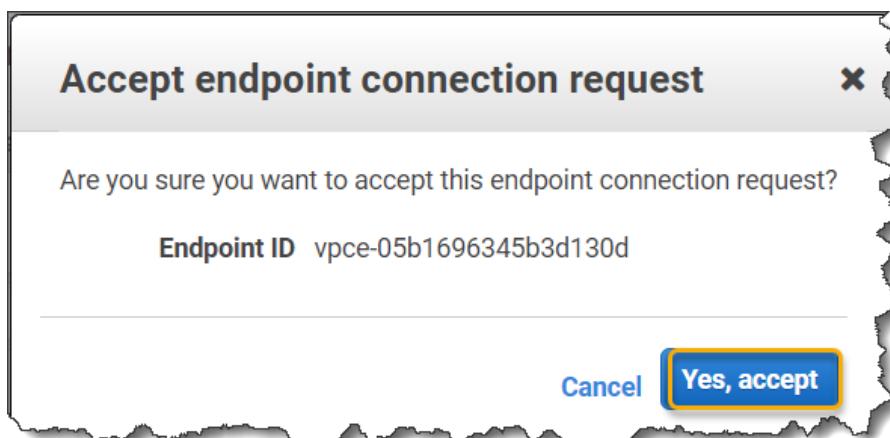
Filter by tags and attributes or search by keyword [K](#) [<](#) 1 to 3 of 3

[Cancel](#) [Create endpoint](#)

21. Now go again to Endpoint Services select **your service**, click on **Endpoint connections** under Action click on **Accept endpoint connection request**

The screenshot shows the AWS Lambda console with the 'Endpoint Service' set to 'vpce-svc-0710867d8de3579f5'. The 'Endpoint Connections' tab is selected. Under 'Actions', the 'Accept endpoint connection request' button is highlighted with a yellow box. A modal window is open, asking 'Are you sure you want to accept this endpoint connection request?' It shows the 'Endpoint ID' as 'vpce-05b1696345b3d130d'. The 'Yes, accept' button is also highlighted with a yellow box.

22. Click on **Yes accept**



23. Now go to the endpoints on VPC console and copy DNS name

The screenshot shows the AWS VPC Endpoints console. The left sidebar has 'Endpoints' selected. The main area shows a table with one endpoint listed. The 'DNS names' field for this endpoint is highlighted with a yellow box, showing two entries: 'vpce-05b1696345b3d130d-k0lx1lf5.vpc.amazonaws.com (Z7HUB22UULQXV)' and 'vpce-05b1696345b3d130d-k0lx1lf5-us-east-1a.vpc.amazonaws.com'.

24. Now SSH to your test instance and run following command

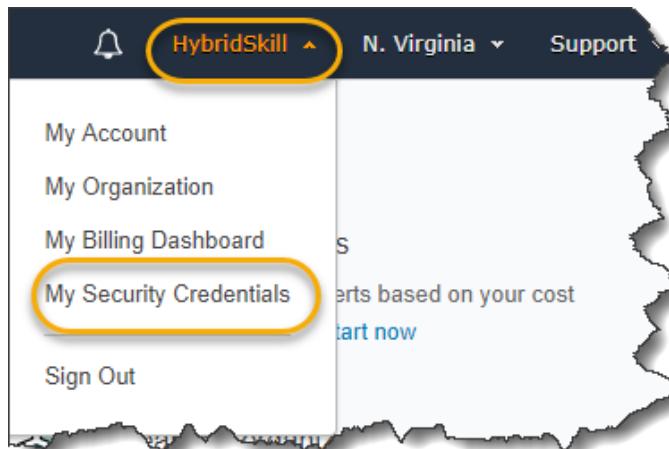
```
Curl vpce-05b1696345b3d130d-k0ix1lf5.vpce-svc-0710867d8de3579f5.us-east-1.vpce.amazonaws.com
```

## Task 12: Download Security Credentials

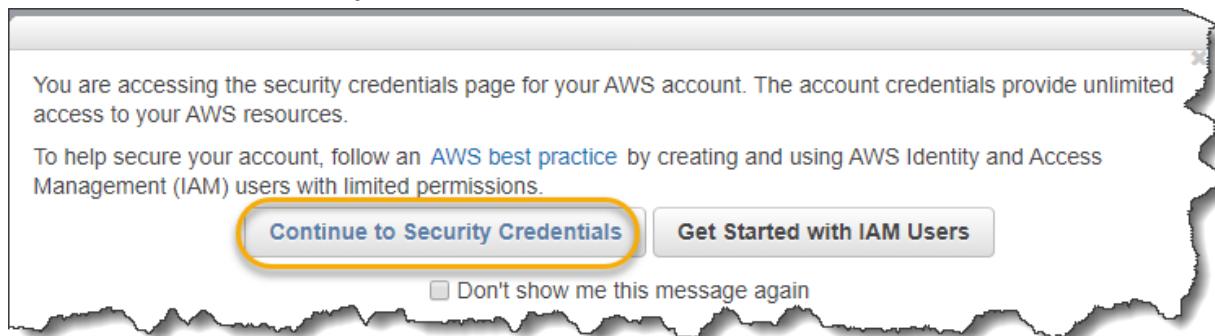
First we will download an Access key and Secret Key to sign our API calls. Depending on whether you have created your own account or are sub account under a corporate account follow either step A or B.

### A. For Root account users

1. Log into the console, at the top left click on your **login name** and from the dropdown click on **Security Credentials**.



2. Click **Continue to Security Credentials**



3. Expand Access keys (access key ID and secret access key) and click Create New Access Key

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM), see [AWS Identity and Access Management \(IAM\)](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#).

| Created | Deleted | Access Key ID | Last Used |
|---------|---------|---------------|-----------|
|         |         |               |           |

**Create New Access Key**

**Important Change - Managing Your AWS Secret Access Keys**

As described in a [previous announcement](#), you cannot retrieve the existing secret access key after it has been deleted. As a best practice, we recommend [creating an IAM user](#) and granting them the necessary permissions.

4. Click Download Key File

Create Access Key

**Your access key (access key ID and secret access key) has been created successfully.**

Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

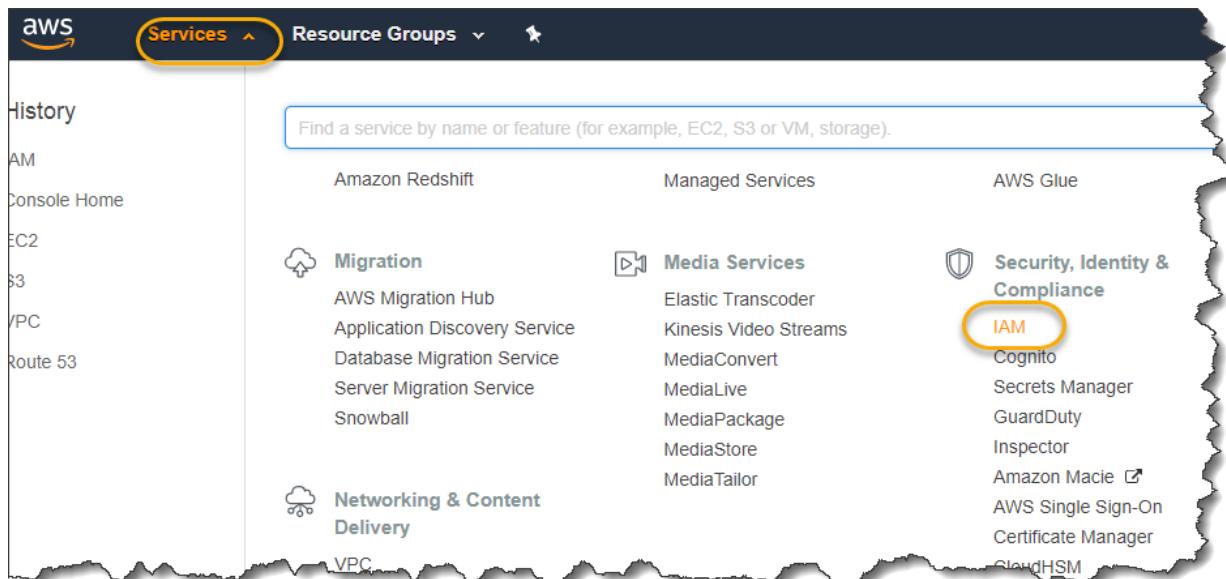
To help protect your security, store your secret access key securely and do not share it.

► Show Access Key

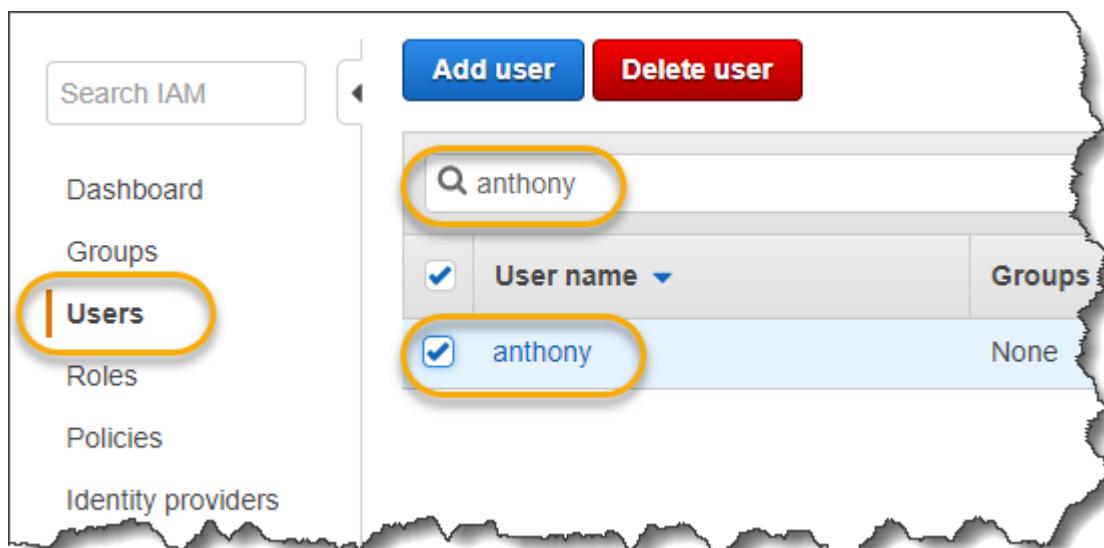
**Download Key File** Close

## B. For IAM subaccount users

1. Click Services and under Security Identity and Compliance click IAM



2. Click **Users** search for your **username** and click on it for more details



3. Click on the **Security credentials** tab and click on **Create access key**

The screenshot shows the AWS IAM User Summary page for a user named 'anthony'. At the top, there are tabs for 'Permissions', 'Groups (0)', 'Security credentials' (which is highlighted with a yellow box), and 'Access Advisor'. Below the tabs, there's a section for 'Sign-in credentials' and another for 'Access keys'. In the 'Access keys' section, a button labeled 'Create access key' is highlighted with a yellow box. The 'Access keys' table has columns for 'Access key ID', 'Created', and 'Last used'. There are no entries in the table.

Finally click on **Download .csv file**

The screenshot shows the 'Create access key' success page. It displays a green success message: 'Success: This is the only time that the secret access keys can be generated later. However, you can create new access keys at any time.' Below the message is a button labeled 'Download .csv file' with a yellow box around it. At the bottom, there are fields for 'Access key ID' (containing 'AKIAIHPV66FNFDP6K3JXQ') and 'Secret access key' (with a 'Show' link).

## Task 13: Setup the AWS CLI

In this task we are going to setup the AWS CLI and authorize it to perform command line API calls using the Access keys you downloaded earlier

1. For Windows, Users go to <http://aws.amazon.com/cli/> and download and run the 64-bit or 32-bit Windows Installer
2. For MAC users type the following commands

1. sudo curl https://bootstrap.pypa.io/ez\_setup.py -o - | sudo python
2. sudo easy\_install pip
3. sudo pip install awscli
4. **export LC\_ALL=en\_US.UTF-8**

Next open a Command prompt or Terminal and type the below command. Note that “**YOUR\_AWS\_ACCESS\_KEY**” should be replaced by your actual access key and “**YOUR\_AWS\_SECRET\_KEY**” should be replaced by an actual secret key that was downloaded.

```
aws configure

AWS Access Key ID [None]: YOUR_AWS_ACCESS_KEY

AWS Secret Access Key [None]: YOUR_AWS_SECRET_KEY

Default region name [None]: ap-southeast-1

Default output format [None]:
```

## Task 14: Manage VPC through CLI

Now that we have explored VPC through the console, let's do the same through the CLI. Run the following commands on the command line interface, you had setup earlier.

1. Create VPC and note down its vpc id:

```
:~$ aws ec2 create-vpc --cidr-block 10.10.0.0/16
```

```
{
 "Vpc": {
 "CidrBlock": "10.10.0.0/16",
 ...
 "VpcId": "vpc-602b9308",
 "CidrBlockAssociationSet": [{
 ...
 "CidrBlock": "10.10.0.0/16",
 ...
 }]
 }
}
```

2. Create Subnet and note down subnet id

```
:~$ aws ec2 create-subnet --vpc-id vpc-602b9308 --cidr-block 10.10.1.0/24

{
 "Subnet": {
 ..
 "SubnetId": "subnet-ecd56d84",
 ..
 }
}
```

3. Create gateway and note down the gateway ID:

```
:~$ aws ec2 create-internet-gateway
{
 "InternetGateway": {
 "Attachments": [],
 "InternetGatewayId
```

4. Attach Internet gateway to VPC

```
:~$ aws ec2 attach-internet-gateway --vpc-id vpc-602b9308 --internet-gateway-id igw-49576c20
```

5. Create Route Table

```
:~$ aws ec2 create-route-table --vpc-id vpc-602b9308
{
 "RouteTable": {
 ..
 "RouteTableId
```

6. Add a rule to route table for internet gateway

```
:~$ aws ec2 create-route --route-table-id rtb-4fbcc827 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-49576c20
{
 "Return": true
}
```

```
:~$ aws ec2 describe-route-tables --route-table-id rtb-4fbcc827
```

7. Associate to route table to subnet

```
:~$ aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-602b9308" --query 'Subnets[*].{ID:SubnetId,CIDR:CidrBlock}'
[
 {
 "SubnetId": "subnet-ecd56d84",
 "CidrBlock": "10.10.1.0/24"
 }
]
```

```
 "ID": "subnet-ecd56d84",
 "CIDR": "10.10.1.0/24"
 },
 {
 "ID": "subnet-0ed36b66",
 "CIDR": "10.10.2.0/24"
 }
]
```

8. This will make a subnet public:

```
:~$ aws ec2 associate-route-table --subnet-id subnet-ecd56d84 --route-table-id rtb-4fbcc827
{
 "AssociationId": "rtbassoc-adc92dc6"
}
```

9. Enable public IP addressing :

```
:~$ aws ec2 modify-subnet-attribute --subnet-id subnet-ecd56d84 --map-public-ip-on-launch
```

10. Launch wordpress server from AMI:

a. Create Security group

```
:~$ aws ec2 create-security-group --group-name webserver-nondefault --vpc-id vpc-602b9308 --
description "for non default vpc"
{
 "GroupId": "sg-4cea5827"
}
```

11. Add ingress rules

```
:~$ curl ipinfo.io/ip
106.200.201.58
```

```
:~$ aws ec2 authorize-security-group-ingress --group-id sg-4cea5827 --protocol tcp --port 22 --cidr
106.200.201.58/32
```

```
:~$ aws ec2 authorize-security-group-ingress --group-id sg-4cea5827 --protocol icmp --port all --cidr
106.200.201.58/32
```

```
:~$ aws ec2 authorize-security-group-ingress --group-id sg-4cea5827 --protocol tcp --port 80 --cidr
0.0.0.0/0
```

c. Launch box

```
:~$aws ec2 run-instances --image-id ami-0e356a61 --instance-type t2.micro --key-name hybridskill-
test --security-group-ids sg-4cea5827 --subnet-id subnet-ecd56d84
```

12. Launch db box in private subnet:

- a. The other subnet with default route table attached is the private subnet.
- b. Create security group for db

```
:~$ aws ec2 create-security-group --group-name db-nondefault --vpc-id vpc-602b9308 --description "for non default vpc"
```

13. Add ingress rules

```
:~$ aws ec2 authorize-security-group-ingress --group-id sg-f7dd6f9c --protocol tcp --port 22 --cidr 106.200.201.58/32
```

```
:~$ aws ec2 authorize-security-group-ingress --group-id sg-f7dd6f9c --protocol icmp --port all --cidr 10.10.0.0/16
```

14. Start db box:

```
:~$ aws ec2 run-instances --image-id ami-e60e5a89 --instance-type t2.micro --key-name hybridskill-test --security-group-ids sg-f7dd6f9c --subnet-id subnet-0ed36b66
```

15. Ping WebServer from local box:

```
:~$ ping 13.126.142.213
PING 13.126.142.213 (13.126.142.213): 56 data bytes
64 bytes from 13.126.142.213: icmp_seq=0 ttl=53 time=72.152 ms
64 bytes from 13.126.142.213: icmp_seq=1 ttl=53 time=72.130 ms
...
```

16. Login to Web Server:

```
:~$ ssh -i hybridskill-test.pem ec2-user@13.126.142.213
```

```
[ec2-user@ip-10-10-1-135 ~]$ ping google.com
PING google.com (216.58.211.174) 56(84) bytes of data.
64 bytes from dub08s01-in-f174.1e100.net (216.58.211.174): icmp_seq=1 ttl=43 time=243 ms
64 bytes from dub08s01-in-f174.1e100.net (216.58.211.174): icmp_seq=2 ttl=43 time=243 ms
^X64 bytes from dub08s01-in-f174.1e100.net (216.58.211.174): icmp_seq=3 ttl=43 time=243 ms
64 bytes from dub08s01-in-f174.1e100.net (216.58.211.174): icmp_seq=4 ttl=43 time=243 ms
```

17. Login to db server via Web Server:

```
[ec2-user@ip-10-10-1-135 ~]$ ssh ec2-user@10.10.2.110
Last login: Mon Mar 5 21:14:39 2018 from 10.10.1.135
[ec2-user@ip-10-10-2-110 ~]$ ping google.com
PING google.com (172.217.166.78) 56(84) bytes of data.
```

```
^C
--- google.com ping statistics ---
 74 packets transmitted, 0 received, 100% packet loss, time 72999ms
```

18. Launch NAT instance. First Create security group for NAT

```
:~$ aws ec2 create-security-group --group-name nat-nondefault --vpc-id vpc-602b9308 --description "for non default vpc"
```

```
{
 "GroupId": "sg-80cc7eeb"
}
```

19. Create security group Ingress rule

```
:$ aws ec2 authorize-security-group-ingress --group-id sg-80cc7eeb --protocol tcp --port 22 --cidr 106.200.201.58/32
```

```
:$ aws ec2 authorize-security-group-ingress --group-id sg-80cc7eeb --protocol icmp --port all --cidr 10.10.0.0/16
```

20. Launch box in public subnet using the Red Hat Enterprise Linux 7.4 AMI

```
:$ aws ec2 run-instances --image-id ami-e60e5a89 --instance-type t2.micro --key-name hybridskill-test --security-group-ids sg-80cc7eeb --subnet-id subnet-ecd56d84
```

21. Login into the NAT box:

```
:$ ssh -i hybridskill-test.pem ec2-user@13.126.142.213
:$ sysctl net.ipv4.ip_forward=1
:$ iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

22. Add rule in private route table for nat instance ID

```
:$ aws ec2 create-route --route-table-id rtb-e4b3c78c --destination-cidr-block 0.0.0.0/0 --instance-id i-0663e364fdb4dc43f
{
 "Return": true
}
```

23. Disable Source/Dest-check

```
:$ aws ec2 modify-instance-attribute --instance-id i-0663e364f5b48c67f --no-source-dest-check
```

Not login to db server and try to and ping the Internet.