

Amazon Virtual Private Cloud (Amazon VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables us to launch AWS resources into a virtual network that we've defined.

- Amazon VPC is the networking layer for Amazon EC2 Instances.
- For Fresh account Amazon have default VPC Network.
- Best Practice to always create our own VPC with our own Settings and Configuration.

Few Key Points and Terminology:

- Virtual private cloud (VPC): A virtual network dedicated to our AWS account.
- Subnet: A range of IP addresses in our VPC.
- Route table: A set of rules, called routes, that are used to determine where network traffic is directed.
- Internet gateway: A gateway that we attach to our VPC to enable communication between resources in our VPC and the internet.
- VPC endpoint: Enables us to privately connect our VPC to supported AWS services and VPC endpoint services powered by Private Link without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

What user can do with VPC:

- Launch instances in a subnet of our choice. we can choose our own subnet addressing.
- we can assign custom IP address ranges in each subnet.
- we can configure route tables between subnets.
- we can create an internet gateway and attach it to our VPC.
- It provides much better security control over our AWS resources.
- we can assign security groups to individual instances.

Amazon Virtual Private Cloud (Amazon VPC)

- we also have subnet network access control lists (ACLs).
- For Small or Medium Setup One VPC will be enough.
- An Instance Launched in one VPC can never communicate to Instance Launched in another VPC via Private IP.
- Public IP is must to setup the inter VPC communication But Two VPCs can be linked via PVC Peering.

VPC Peering:

- VPC Peering is a networking connection that allows us to connect one VPC with another VPC through a direct network route using private IP addresses. ➤ Instances behave as if they were on the same private network.
- we can peer VPC's with other AWS accounts as well as other VPCs in the same account.
- Peering is in a star configuration, i.e., 1 VPC peers other 4 VPCs.

we can peer between regions. Suppose we have one VPC in one region and other VPC in another region, then we can peer the VPCs between different regions. One VPC can be connected via 1 or more VPCs.

Subnet in VPC:

Virtual private cloud (VPC) is a virtual network dedicated to our AWS account. It is logically isolated from other virtual networks in the AWS Cloud.

- When we create a VPC, we must specify a range of IPv4 addresses for the VPC in the form of a Classless InterDomain Routing (CIDR) block 10.0.0.0/16 is the primary CIDR block for your VPC.
- VPC spans all of the Availability Zones in the Region. After creating a VPC, we can add one or more subnets in each Availability Zone.
- If a subnet's traffic is routed to an internet gateway, the subnet is known as a public subnet.

Amazon Virtual Private Cloud (Amazon VPC)

➤ If a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet.

➤ VPC and subnet sizing for IPv4: 10.0.0.0 – 10.255.255.255 (10/8 prefix) – User VPC must be /16 or smaller,

for example, 10.0.0.0/16.

172.16.0.0 – 172.31.255.255 (172.16/12 prefix)–User VPC must be /16 or smaller, for example, 172.31.0.0/16.

➤ 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)–User VPC can be smaller, for example 192.168.0.0/20.

To add a CIDR block to our VPC, the following rules apply:

➤ The allowed block size is between a /28 netmask and /16 netmask.

➤ CIDR block must not overlap with any existing CIDR block that's associated with the VPC. User cannot increase or decrease the size of an existing CIDR block.

Security Group in AWS

➤ Security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

➤ Upto 5 SGs can be assigned to Instance in AWS.

➤ SGs are Instance Level not Subnet Level.

Basics of Security Group in AWS

➤ User can specify allow rules, but not deny rules.

➤ User can specify separate rules for inbound and outbound traffic.

➤ Security group rules enable us to filter traffic based on protocols and port numbers.

➤ By default, a Security Group don't have any Inbound Rule.

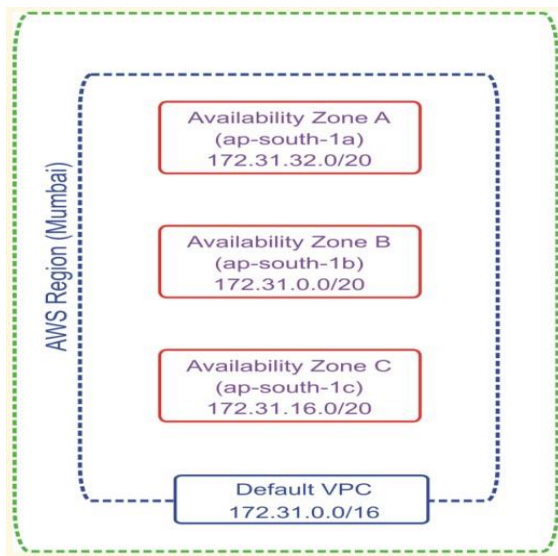
Amazon Virtual Private Cloud (Amazon VPC)

➤ By default, a security group includes an outbound rule that allows all outbound traffic.

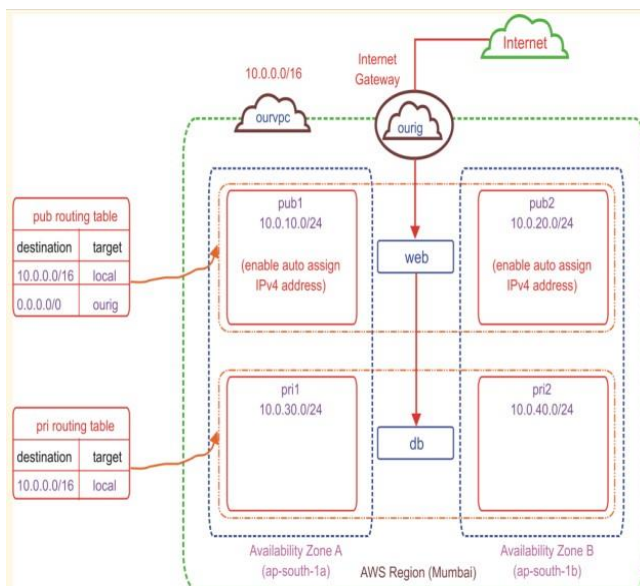
There are quotas on the number of security groups that we can create per VPC, the number of rules that we can add to each security group, and the number of security groups that we can associate with a network interface.

Lab Guide:

Lab-1: Understanding the Default VPC



Lab-2: Creating the Custom VPC with private and public subnets

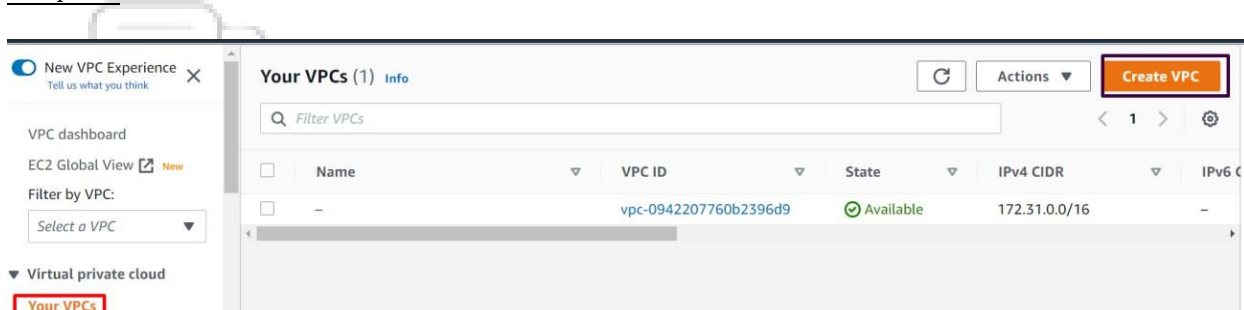


Amazon Virtual Private Cloud (Amazon VPC)

1. Create VPC with 10.0.0.0/16 CIDR
2. Create Public subnet-1 within the custom VPC with CIDR 10.0.10.0/24 in the region ap-south-1a
3. Create public subnet-2 within the custom VPC with CIDR 10.0.20.0/24 in the region ap-south-1b
4. Create private subnet-1 within the custom VPC with CIDR 10.0.30.0/24 in the region ap-south-1a
5. Create private subnet-3 within the custom VPC with CIDR 10.0.40.0/24 in the region ap-south-1b
6. Create Internetgateway with in the custom VPC
7. Create Route Tables for public subnets and private subnets and attach it to VPC
8. Edit the route table of the public subnet and attach the Internetgateway.
9. Create an EC2 machine in public subnet and install nginx in it, we should be able to connect to the machine and access the nginx application.
10. Create an ec2 instance in private subnet we should not be able to connect to it.

Creating the VPC

Step-1:Click on Create VPC button in the VPC dashboard as shown below:



Step-2: when we click on Create VPC button as shown above, we will get the below screen to create the VPC.

Amazon Virtual Private Cloud (Amazon VPC)

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional Info
Creates a tag with a key of 'Name' and a value that you specify.

my-vpc-01

IPv4 CIDR block Info
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/24

IPv6 CIDR block Info
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

Tenancy Info
Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

You can add 50 more tags.

Step-3: enter the details as below to create the VPC

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional Info
Creates a tag with a key of 'Name' and a value that you specify.

weshopify-platform-vpc

IPv4 CIDR block Info
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

IPv6 CIDR block Info
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

Tenancy Info
Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
Name	weshopify-platform-vpc	<input type="button" value="Remove"/>

You can add 49 more tags.

Here IPV4CIDR choose as 10.0.0.0/16 to generate the $2^{16}=65536$ ip addresses)

Step-4: As soon as we click on Create VPC button above the VPC will be created with the default route table as main as shown below:

Amazon Virtual Private Cloud (Amazon VPC)

The screenshot shows the 'Details' tab for a VPC. The VPC ID is vpc-032fdf526c5aa9a79. The state is 'Available'. The main route table is highlighted with a red box and contains the entry 'rtb-0f6dc39175a5da811'. The main network ACL is 'acl-0f6940c39da4f8bc1'. The IPv4 CIDR is '10.0.0.0/16'. The IPv6 CIDR is '-'. The owner ID is '708699230411'.

Address type	CIDR	Pool	Status
IPv4	10.0.0.0/16	-	Associated

Note: here we must have to change the default route table to the one we gonna create in next further steps.

Creating the Subnets

Step-1: Click on Subnets and click on create subnet button as shown below:

The screenshot shows the 'Subnets (3)' page in the Amazon VPC console. The 'Create subnet' button is highlighted with a red box. The table lists three subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available
-	subnet-0c849719b21364a67	Available	vpc-0942207760b2396d9	172.31.0.0/20	-	4091
-	subnet-03f83fa005411722a	Available	vpc-0942207760b2396d9	172.31.32.0/20	-	4091
-	subnet-079b6dca4998d41c1	Available	vpc-0942207760b2396d9	172.31.16.0/20	-	4091

Step-2: As soon as we click on Create Subnet button we will see the below screen to create the customized subnets.

Amazon Virtual Private Cloud (Amazon VPC)

VPC > Subnets > Create subnet

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.

Select a VPC

Q

vpc-0942207760b2396d9 (default)
172.31.0.0/16

vpc-032fd526c5aa9a79 (weshopify-platform-vpc)
10.0.0.0/16

Select a VPC first to create new subnets.

Add new subnet

Cancel Create subnet

Step-3: Choose the Custom VPC I.e. weshopify-platform-vpc from the drop down as shown above and the following screen will be opened for us to create the subnets.

1. Creating the public subnet-1

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

weshopify-public-cloud-01
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Singapore) / ap-southeast-1a

IPv4 CIDR block [Info](#)
Q 10.0.10.0/24

Tags - optional

Key Value - optional

Q Name X Q weshopify-public-cloud-01 X Remove

Add new tag
You can add 49 more tags.

Remove

Add new subnet

2. Creating the public subnet-2

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

weshopify-public-cloud-02
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Singapore) / ap-southeast-1b

IPv4 CIDR block [Info](#)
Q 10.0.20.0/24

Tags - optional

Key Value - optional

Q Name X Q weshopify-public-cloud-02 X Remove

Add new tag
You can add 49 more tags.

Remove

Add new subnet

3. Creating the private subnet-01

Amazon Virtual Private Cloud (Amazon VPC)

Subnet 3 of 3

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

weshopify-private-cloud-01

The name can be up to 256 characters long.

Availability Zone

[Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Singapore) / ap-southeast-1a

IPv4 CIDR block

[Info](#)

10.0.30.0/24

Tags - optional

Key

Name

Value - optional

weshopify-private-cloud-01

Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

4. Creating the private subnet-02

Subnet 4 of 4

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

weshopify-private-cloud-02

The name can be up to 256 characters long.

Availability Zone

[Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Singapore) / ap-southeast-1b

IPv4 CIDR block

[Info](#)

10.0.40.0/24

Tags - optional

Key

Name

Value - optional

weshopify-private-cloud-02

Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel

Create subnet

As soon as we click on create subnet button as shown above, we will be able to see the list of subnets created as shown below:

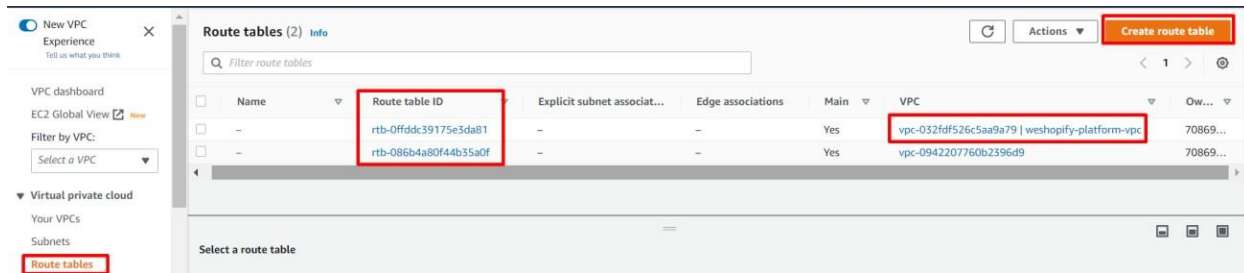
<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	weshopify-private-cloud-01	subnet-0d59a51d44c960fd3	Available	vpc-032fdf526c5aa9a79 weshopify-platform-vpc	10.0.30.0/24
<input type="checkbox"/>	weshopify-private-cloud-02	subnet-07b9cd718da393690	Available	vpc-032fdf526c5aa9a79 weshopify-platform-vpc	10.0.40.0/24
<input type="checkbox"/>	weshopify-public-cloud-01	subnet-06f119134bc2f9879	Available	vpc-032fdf526c5aa9a79 weshopify-platform-vpc	10.0.10.0/24
<input type="checkbox"/>	weshopify-public-cloud-02	subnet-03a2cc986f76e7a14	Available	vpc-032fdf526c5aa9a79 weshopify-platform-vpc	10.0.20.0/24

Creating the Route Tables

Step-1:

Create the 2 route tables later we can assign one for public cloud and another for private cloud . To Create the route tables click on the create route table button as shown below:

Amazon Virtual Private Cloud (Amazon VPC)



Note: here in the above screen shot we can see a route table created when our weshopify-platform-vpc created. We need to replace this default route table with the one we are now going to create.

Step-2: Create Route Table for private cloud

1. When we click on the create route table button as shown in the above screen shot we will get the below screen to enter the route names and then to create it

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
weshopify-private-cloud-route-table

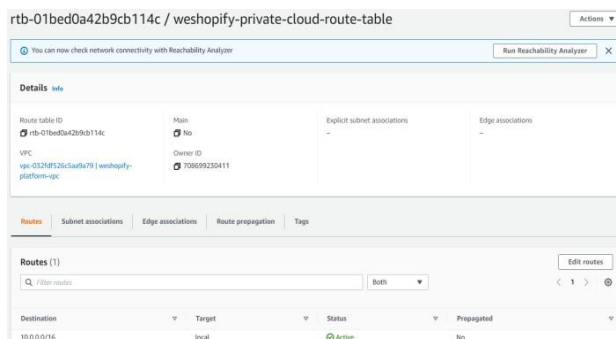
VPC
The VPC to use for this route table.
vpc-032fdf526c5aa9a79 (weshopify-platform-vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key: Name, Value: weshopify-private-cloud-route-1

Buttons: Add new tag, Cancel, Create route table

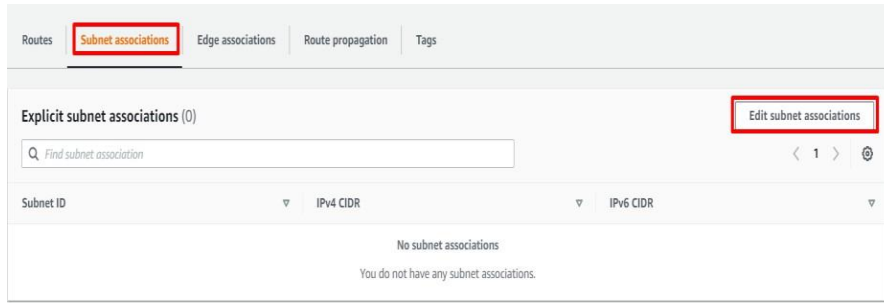
As soon as we click on create route table the route table will be created as shown below:



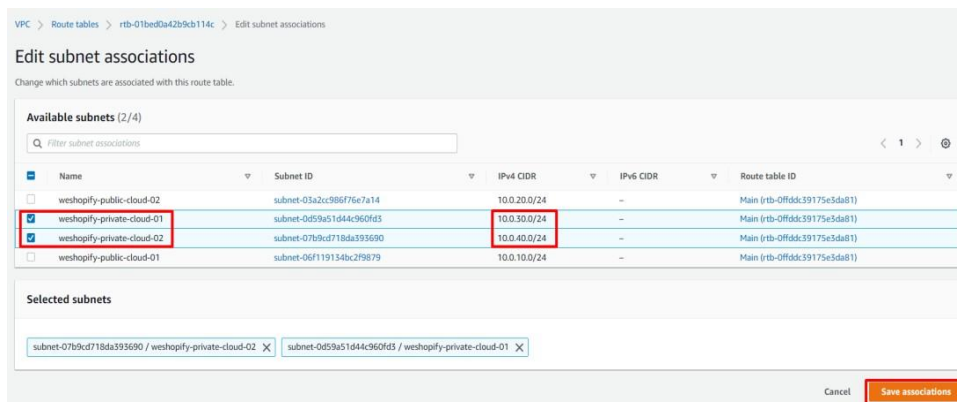
2. adding the subnet associations in the private cloud route as shown below:

Click on Subnet associations and edit subnet associations as shown below:

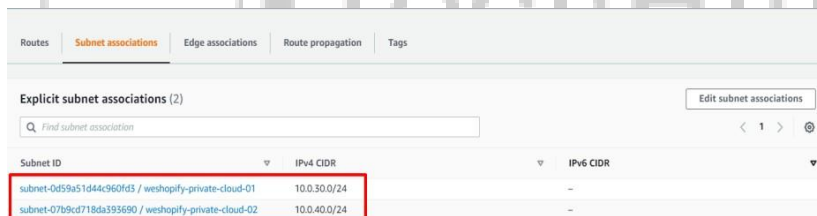
Amazon Virtual Private Cloud (Amazon VPC)



3. as soon as we click on edit subnet associations the below screen will be opened for us to choose the subnets.



4. as soon as we click on save associations we can see the private subnets mapped to the private route table as shown below:



Step-3: Create Route Table for public cloud

1. Create route table for public cloud as showb below

Amazon Virtual Private Cloud (Amazon VPC)

Create route table [info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional

Create a tag with a key of 'Name' and a value that you specify.

weshopify-public-cloud-route-table

VPC

The VPC to use for this route table.

vpc-032fd526c5aa9a79 (weshopify-platform-vpc)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Q Name X

Value - optional

Q weshopify-public-cloud-route-t X

Remove

Add new tag

You can add 49 more tags.

Cancel

Create route table

2. Associate public subnets using the public route table

VPC > Route tables > rtb-096568214d9619d5 [Edit subnet associations](#)

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

Filter subnet associations

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> weshopify-public-cloud-02	subnet-03a2cc986f75e7a14	10.0.20.0/24	-	Main (rtb-096568214d9619d5)
<input type="checkbox"/> weshopify-private-cloud-01	subnet-0d59a5164496043	10.0.30.0/24	-	rtb-01bed3a229b5114c / weshopify-private-cloud-route-table
<input type="checkbox"/> weshopify-private-cloud-02	subnet-0769d718da393600	10.0.40.0/24	-	rtb-01bed3a229b5114c / weshopify-private-cloud-route-table
<input checked="" type="checkbox"/> weshopify-public-cloud-01	subnet-06f119134bc2f9879	10.0.10.0/24	-	Main (rtb-096568214d9619d5)

Selected subnets

subnet-06f119134bc2f9879 / weshopify-public-cloud-01 X subnet-03a2cc986f75e7a14 / weshopify-public-cloud-02 X

Cancel [Save associations](#)

3. As soon as we click on the save associations we can see the public cloud subnets associated with the public cloud route table as shown below:

Routes **Subnet associations** Edge associations Route propagation Tags

Explicit subnet associations (2)

Find subnet association

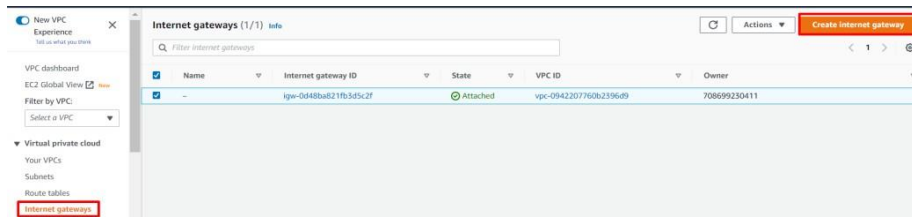
Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-03a2cc986f75e7a14 / weshopify-public-cloud-02	10.0.20.0/24	-
subnet-06f119134bc2f9879 / weshopify-public-cloud-01	10.0.10.0/24	-

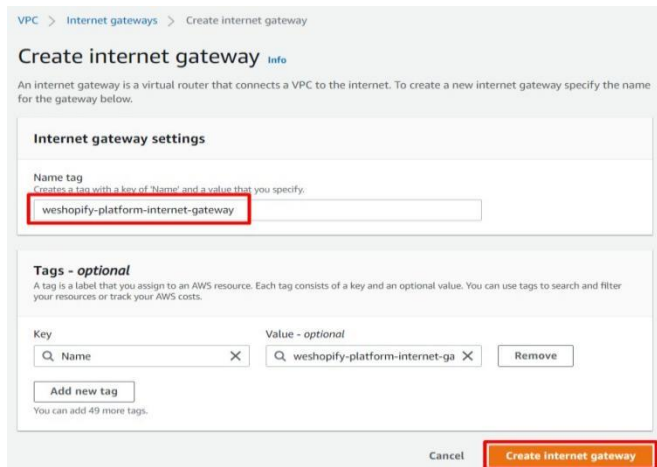
Creating the Internet Gateway

Step-1: To Create internet gateway, click on the Create Internet Gateway button as shown below:

Amazon Virtual Private Cloud (Amazon VPC)



Step-2: as soon as we click on Create Internet Gateway we will get the below screen to enter the name of the gateway



Step-3: once internet gateway created, attach it with the VPC as shown below:



As soon as we click on the Attach to VPC, the below screen is shown to attach it to the available VPC's of our choice.



Step-4: Once the Internet gateway is attached, then we can see the internet gateway as shown below:

Amazon Virtual Private Cloud (Amazon VPC)

VPC > Internet gateways > igw-06d39e64f17f2b76a

igw-06d39e64f17f2b76a / weshopify-platform-internet-gateway

Actions

Details info

Internet gateway ID
igw-06d39e64f17f2b76a

State
Attached

VPC ID
vpc-032df526c5aa9a79 | weshopify-platform-vpc

Owner
708699230411

Tags

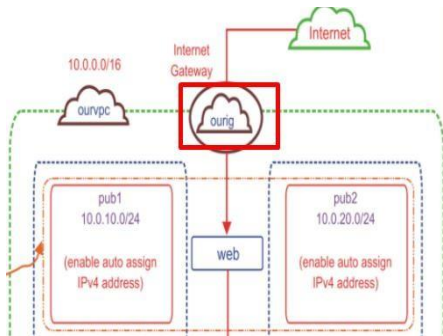
Search tags

Key	Value
Name	weshopify-platform-internet-gateway

Manage tags

Adding the Internet Gateway to the public Route

We have to attach the internet gateway to the public cloud subnets only to access the resources those were deployed on the public cloud.



Goto Route tables and choose the public route table created as shown below:

New VPC Experience

VPC dashboard
EC2 Global View
Filter by VPC
Select a VPC

Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP Option Sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways

Route tables (1/4)

Filter route tables

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
weshopify-private-...	rtb-01bed0a42b9db114c	2 subnets	-	No	vpc-032df526c5aa9a79 wes...	708699230411
-	rtb-0f5d839175e3da81	-	-	Yes	vpc-032df526c5aa9a79 wes...	708699230411
-	rtb-08b4a0f44b35a0f	-	-	Yes	vpc-094220776b2396d9 wes...	708699230411
weshopify-public-d...	rtb-09d56d8214d9619d5	2 subnets	-	No	vpc-032df526c5aa9a79 wes...	708699230411

rtb-09d56d8214d9619d5 / weshopify-public-cloud-route-table

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Filter routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Edit routes

Click on Edit Route tables and attach the internet gateway

VPC > Route tables > rtb-09d56d8214d9619d5 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
	Core Network	-	No
	Egress Only Internet Gateway	-	No
	Gateway Load Balancer Endpoint	-	No
	Instance	-	No
	Internet Gateway	-	No
	local	-	No

Add route

Cancel Preview Save changes

Amazon Virtual Private Cloud (Amazon VPC)

VPC > Route tables > rtb-09d56d8214d9619d5 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
<input type="text"/>	<input type="text" value="igw-06d39e64f17f2b76a (weshopify-platform-internet-gateway)"/>	-	No

Configure from anywhere over the internet establish for our internet gateway

VPC > Route tables > rtb-09d56d8214d9619d5 > 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-06d39e64f17f2b76a"/>	-	No

Now once we click on the save changes, we can see the below screen where it allows only the public subnets over the internet through the internet gateway.

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Both

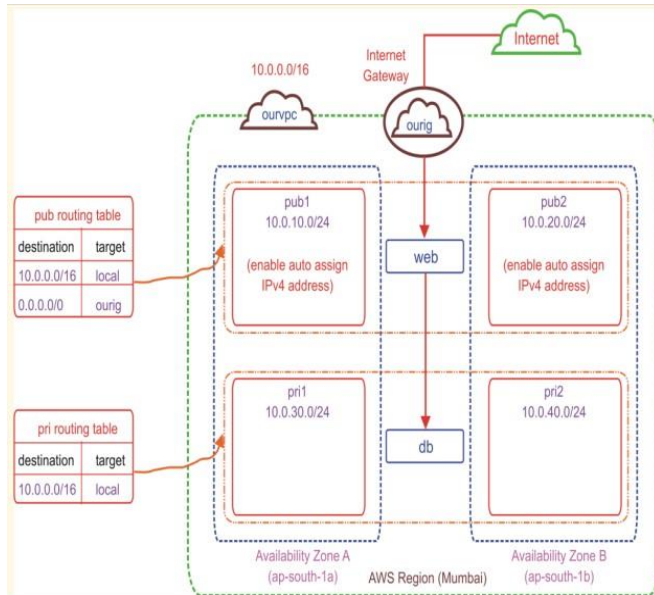
Destination	Target	Status	Propagated
0.0.0.0/0	igw-06d39e64f17f2b76a	Active	No
10.0.0.0/16	local	Active	No

With these configurations I.e.

- ✓ VPC
- ✓ Subnets(private and public)
- ✓ Route table
- ✓ Internetgateway

We are now able to achieve the below VPC architecture.

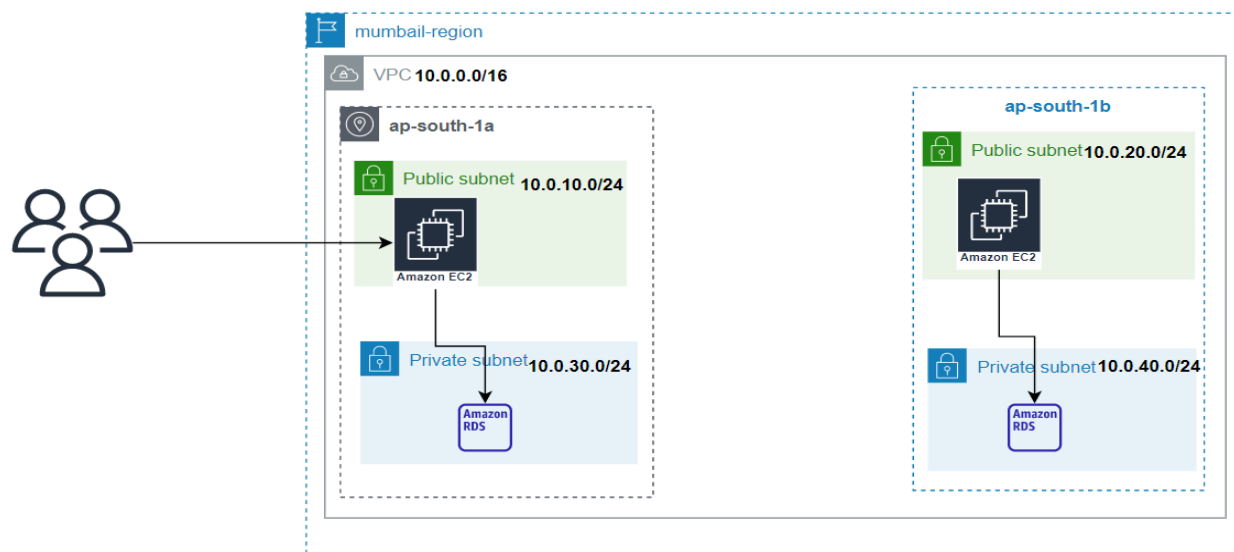
Amazon Virtual Private Cloud (Amazon VPC)



Application

Use Case:

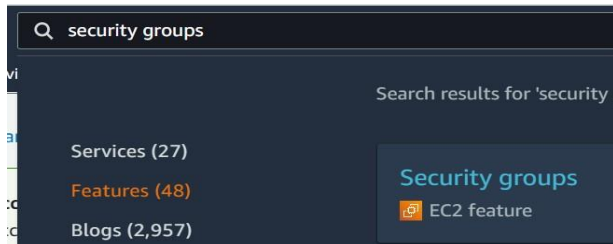
- ✓ Take one spring boot java application which requires a database connection. Take the ec2-machine in the public cloud and deploy the spring boot application in public cloud.
- ✓ Create Mysql database using **RDS** in private subnets
- ✓ Configure the mysql details in the spring boot application
- ✓ Now spring boot application should be able to connect to the database and the application can be accessible over the internet but not the database.



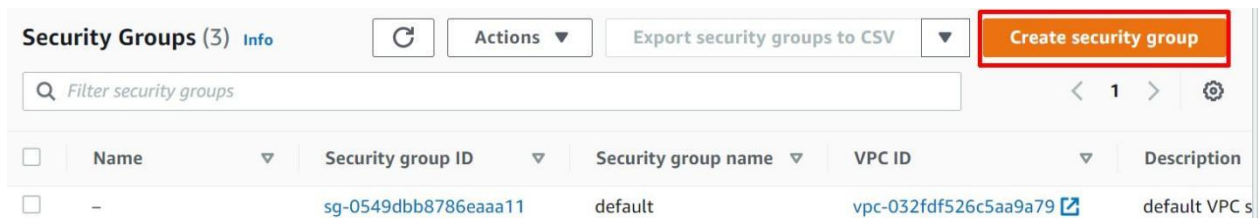
Amazon Virtual Private Cloud (Amazon VPC)

Create Security Group in our Custom VPC

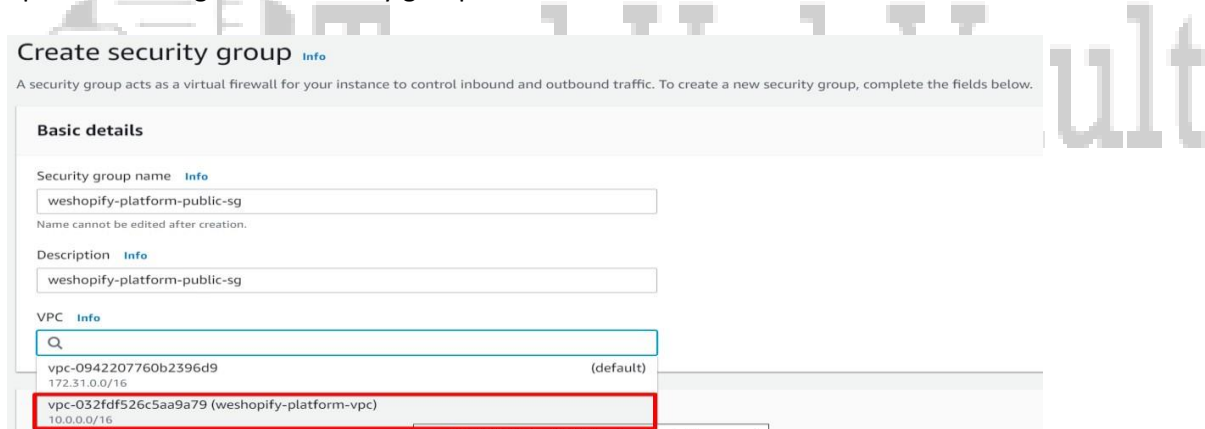
Step-1: To Create the security group search for security groups in search bar as shown below:



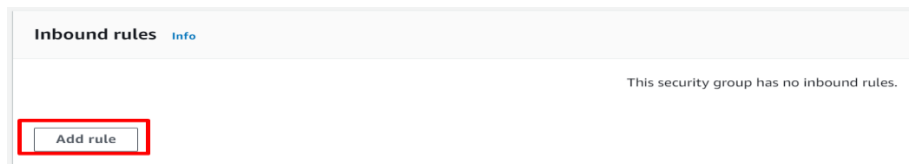
Step-2: as soon as we click on the security groups option as shown above we will get the following screen to create the security group



Step-3: as soon as we click on Create security group button as shown above the following screen will be opened to configure the security group in our VPC



Step-4: Configure the inbound rules, by clicking on the add rule button as shown below:



Step-5: Configure the inbound rules for SSH and for Mysql DB access as shown below:

Amazon Virtual Private Cloud (Amazon VPC)

The screenshot shows the 'Inbound rules' section of the Amazon VPC console. It displays two existing rules:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywh... 0.0.0.0/0	
MySQL/Aurora	TCP	3306	Anywh... 0.0.0.0/0	

Buttons for 'Delete' and 'Add rule' are visible.

Step-6: click on create security group button once after done the above configurations



Step-7: we can see the security group that we created in our custom VPC as shown below:

The screenshot shows the 'Security Groups (4)' page in the Amazon VPC console. A table lists the security groups, with the 'weshopify-platform-sg' and 'weshopify-platform-public-sg' rows highlighted by a red box.

Security group ID	Security group name	VPC ID	Description	Owner
sg-0c945f31ea7028d7a	weshopify-platform-sg	vpc-032fdf526c5aa9a79	weshopify-platform-sg	708699230411
sg-0e8da0631df9411b	weshopify-platform-public-sg	vpc-032fdf526c5aa9a79	weshopify-platform-p...	708699230411

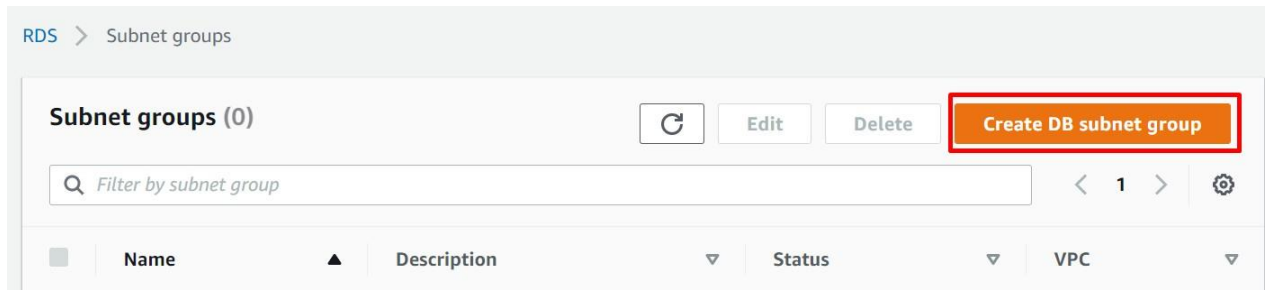
Create Mysql Database using RDS Service in Private Subnets

Step-1: Create database subnet groups, to configure the database in the private subnets as shown below:



Step-2: Once we click on subnet groups, the following window will be opened to create the "DB Subnet Group" as shown below:

Amazon Virtual Private Cloud (Amazon VPC)

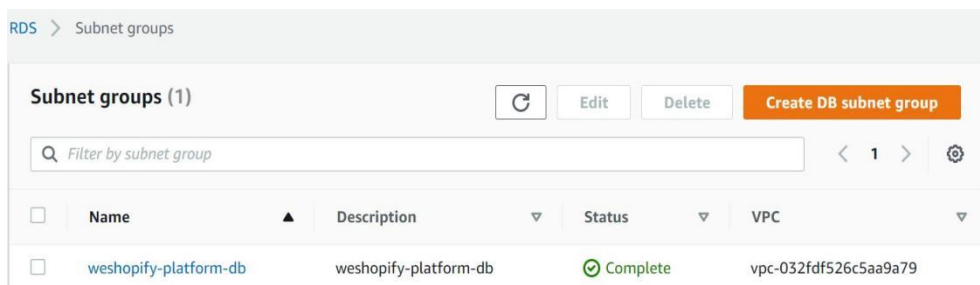


Step-3: Once we click on the Create DB subnet group button, the following screen will be opened where we can enter the name for the db subnet group, and choose the VPC, and region in which the subnets were created and finally choose the private subnets.

The screenshot shows the 'Subnet group details' and 'Add subnets' form. The 'Subnet group details' section includes fields for Name (weshopify-platform-db), Description (weshopify-platform-db), and VPC (weshopify-platform-vpc). The 'Add subnets' section includes a dropdown for Availability Zones (ap-southeast-1a, ap-southeast-1b) and a dropdown for Subnets (subnet-07b9cd718da393690, subnet-0d59a51d44c960fd3). The 'Subnets selected (2)' table shows the selected subnets.

Availability zone	Subnet ID	CIDR block
ap-southeast-1b	subnet-07b9cd718da393690	10.0.40.0/24
ap-southeast-1a	subnet-0d59a51d44c960fd3	10.0.30.0/24

Step-4: Once we click on create button then we can see the db subnet group as shown below:



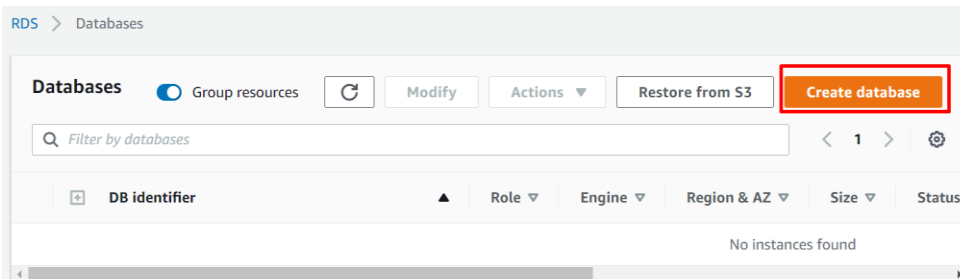
Amazon Virtual Private Cloud (Amazon VPC)

Step-5: Once the DB subnet group is created as shown above, lets create the DB now by choosing the databases option as shown below:

Amazon RDS

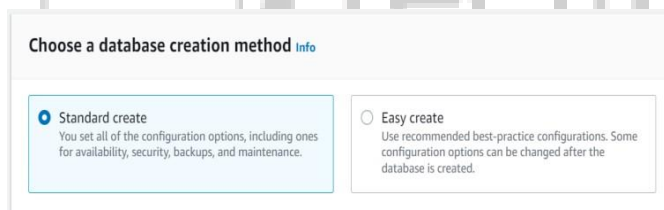
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Automated backups
Reserved instances
Proxies

Step-6: once we click on the Databases options as shown above, we will get the below screen to create the Database.

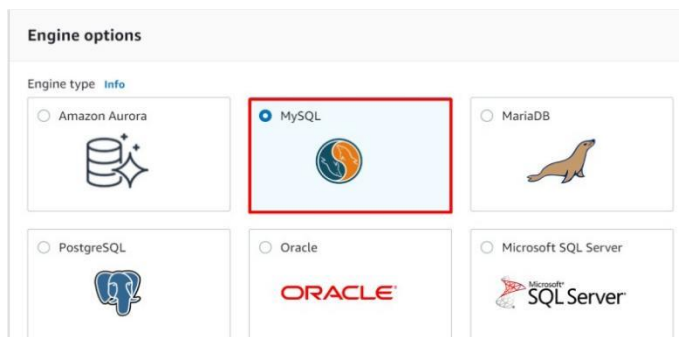


Step-7: Once we click on Create Database button as shown above, we will get the below screen:

1. Choose a database



2. Select mysql



Amazon Virtual Private Cloud (Amazon VPC)

3. Choose ever latest mysql version from the list of available db versions

Edition

☒ MySQL Community



Known issues/limitations

Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

Version

MySQL 8.0.28

4. Choose the Free tier mysql instance

Templates

Choose a sample template to meet your use case.



Production

Use defaults for high availability and fast, consistent performance.



Dev/Test

This instance is intended for development use outside of a production environment.



Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

5. Configure the db settings like db name, user name, password as shown below:

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

weshopify-platform-db

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

root

1 to 16 alphanumeric characters. First character must be a letter.

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm password [Info](#)

Here I have choosen the password as : **Adance123\$**

6. Choose the our VPC in the connectivity window:

Amazon Virtual Private Cloud (Amazon VPC)

Connectivity [Info](#)

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☒ Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

☐ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) [Info](#)

weshopify-platform-vpc (vpc-032fdf526c5aa9a79)

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB Subnet group [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

weshopify-platform-db

Public access [Info](#)

☐ Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

☒ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

☒ Choose existing
Choose existing VPC security groups

☐ Create new
Create new VPC security group

Existing VPC security groups

Choose one or more options

default

weshopify-platform-sg

Availability Zone [Info](#)

No preference

7. Additional Configurations

Additional configuration

Database options, encryption turned off, backup turned off, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)

weshopify_platform

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

default.mysql8.0

Option group [Info](#)

default:mysql-8-0

☐ Enable automated backups
Creates a point-in-time snapshot of your database

Encryption

☐ Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

Log exports

Select the log types to publish to Amazon CloudWatch Logs

☐ Audit log

☐ Error log

☐ General log

☐ Slow query log

IAM role

The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Amazon Virtual Private Cloud (Amazon VPC)

ⓘ Ensure that general, slow query, and audit logs are turned on. Error logs are enabled by default. [Learn more](#)

Maintenance

Auto minor version upgrade [Info](#)

☐ Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

☐ Choose a window

☒ No preference

Deletion protection

☐ Enable deletion protection

Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Note: here in the above screen, the **"Initial database name"** will represents the name of the database schema, in which the tables will be created

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page.](#)

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel [Create database](#)

Finally click on Create Database button.

8. As soon as we click on Create Database button, we can see the data base creating as shown below.

RDS > Databases

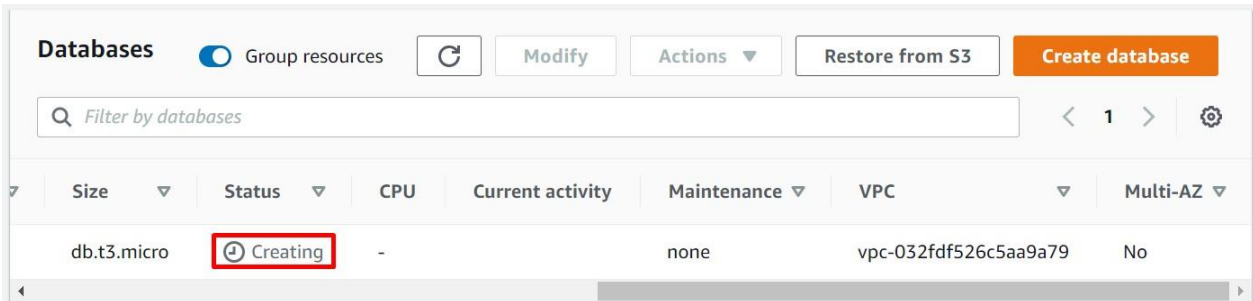
Databases

☒ Group resources

[Refresh](#) [Modify](#) [Actions](#) [Restore from S3](#) [Create database](#)

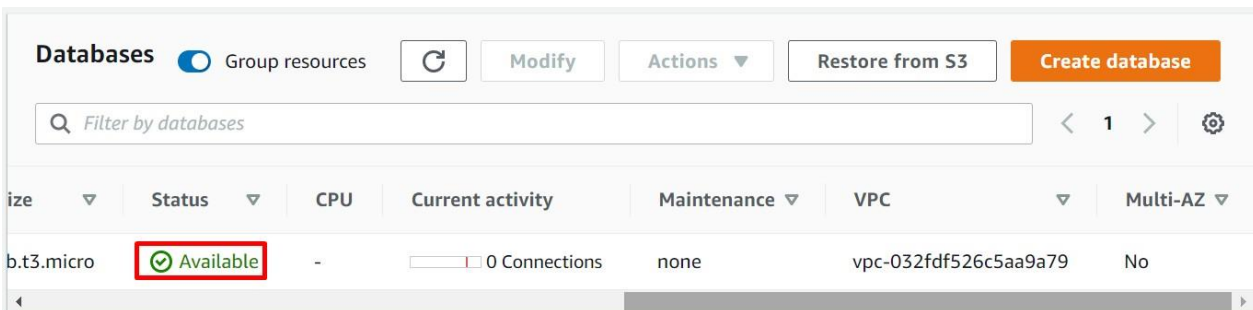
<input type="checkbox"/>	DB identifier	Role	Engine	Region & AZ	Size
<input type="radio"/>	weshopify-platform-sg	Instance	MySQL Community	-	db.t3.mi

Amazon Virtual Private Cloud (Amazon VPC)



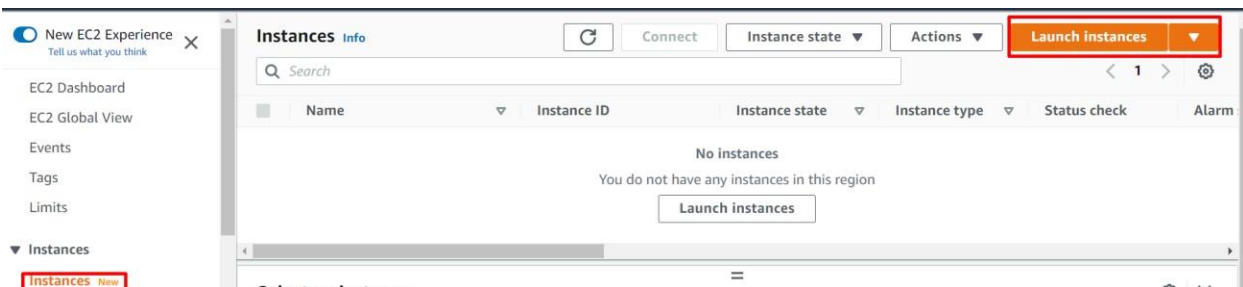
This will take little longer time, but once came to created status, we can connect to it from the AWS resource like EC2 using the command line utility but we cant access it over internet as we have configured it in the private subnets.

Once the database is available after around 10-15 mins of time we can see its status as available as shown below:



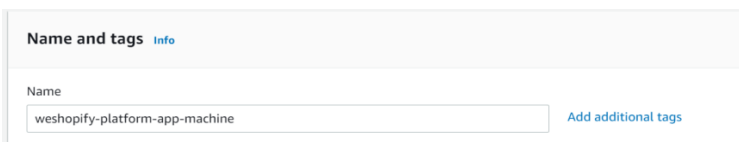
Create Ec2 instance in Public Subnets

Step-1: To Launch EC2 instances, click on Instances link as shown below



Step-2: Click on Launch Instances button as shown above, to create the instance

1. Name and Tags



Amazon Virtual Private Cloud (Amazon VPC)

2. OS Choice

Recents Quick Start

Amazon Linux macOS **Ubuntu** Windows Red Hat S

aws Mac ubuntu Microsoft Red Hat

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-04ff9e9b51c1f62ca (64-bit (x86)) / ami-0b405a114cc7c331d (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 20.04 LTS, amd64 focal image build on 2022-06-10

Architecture AMI ID

64-bit (x86) ami-04ff9e9b51c1f62ca **Verified provider**

3. Instance Type

▼ Instance type Info

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0146 USD per Hour

On-Demand Windows pricing: 0.0192 USD per Hour

Compare instance types

4. Choose the key pair if already created other wise click on create key pair

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

weshopify-platform-app-machine-key

Create new key pair

5. Edit Network settings as shown below to configure this ec2 machine in our vpc as shown below

▼ Network settings Info

Edit

Amazon Virtual Private Cloud (Amazon VPC)

Network settings [Info](#)

VPC - required [Info](#)

vpc-032fdf526c5aa9a79 (weshopify-platform-vpc)
10.0.0.0/16

Subnet [Info](#)

subnet-03a2cc986f76e7a14 **weshopify-public-cloud-02** [Create new subnet](#)
VPC: vpc-032fdf526c5aa9a79 Owner: 708699230411
Availability Zone: ap-southeast-1b IP addresses available: 251 CIDR: 10.0.20.0/24

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ **Select existing security group**

Common security groups [Info](#)

Select security groups

weshopify-platform-sg sg-0c945f31ea7028d7a X
VPC: vpc-032fdf526c5aa9a79

Security groups that you add or remove here will be added to or removed from all your network interfaces.

[Advanced network configuration](#)

Click on launch instance as shown below:

[Cancel](#) [Launch instance](#)

Step-2: Connect to RDS from the ec2 instance that we created

1. To Connect to the DB from the ec2 instance, goto the ec2 instances list and select the instance what we created and click on connect button as shown below:

Instances (1) [Info](#) [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[Clear filters](#)

Instance ID = i-0f61ada589f2afcb3 X

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	weshopify-platform-app-ma...	i-0f61ada589f2afcb3	Running	t2.micro	Initializing	No alarms	ap-southeast-1b	-

[EC2](#) > [Instances](#) > [i-0f61ada589f2afcb3](#)

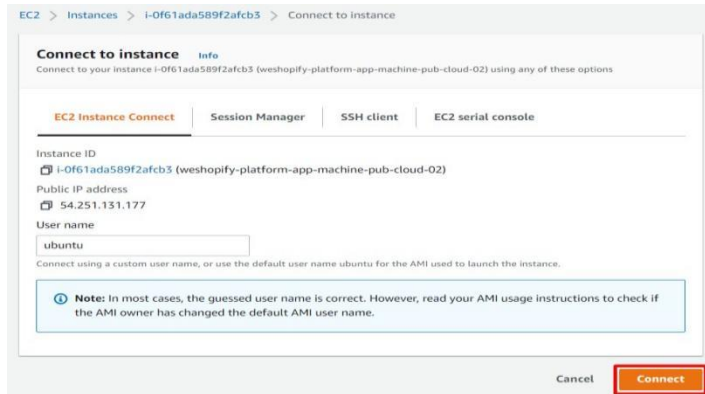
Instance summary for i-0f61ada589f2afcb3 (weshopify-platform-app-machine-pub-cloud-02) [Info](#)

Refreshing instance data

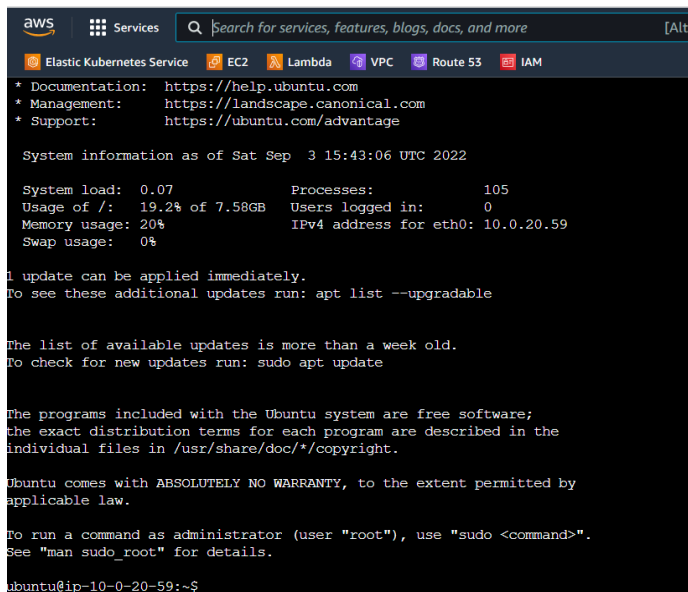
[Refresh](#) [Connect](#) [Instance state](#) [Actions](#)

2. Once we click on connect the following window will be open click on connect.

Amazon Virtual Private Cloud (Amazon VPC)



3. As soon as we click on connect the ec2 machine will be opened in the browser as shown below:



```
aws Services Search for services, features, blogs, docs, and more [Alt+]
Elastic Kubernetes Service EC2 Lambda VPC Route 53 IAM
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sat Sep 3 15:43:06 UTC 2022

System load: 0.07 Processes: 105
Usage of /: 19.2% of 7.58GB Users logged in: 0
Memory usage: 20% IPv4 address for eth0: 10.0.20.59
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-20-59:~$
```

4. Switch to root user and update the machine

```
ubuntu@ip-10-0-20-59:~$ sudo -i
root@ip-10-0-20-59:~# apt-get update
```

5. Install the mysql client as shown below

Amazon Virtual Private Cloud (Amazon VPC)

```
root@ip-10-0-20-59:~# mysql

Command 'mysql' not found, but can be installed with:

apt install mysql-client-core-8.0 # version 8.0.30-0ubuntu0.20.04.2, or
apt install mariadb-client-core-10.3 # version 1:10.3.34-0ubuntu0.20.04.1

root@ip-10-0-20-59:~# apt install mysql-client-core-8.0
```

6. Once the mysql client will install we can connect to the RDS from the ec2 instance using the below command:

```
root@ip-10-0-20-59:~# mysql -h weshopify-platform-sg.cci53w9t6upc.ap-southeast-1.rds.amazonaws.com -P 3306 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Command is: `mysql -h <RDS_MYSQL_ENDPOINT> -P 3306 -u <user_name> -p`

For example: `mysql -h weshopify-platform-sg.cci53w9t6upc.ap-southeast-1.rds.amazonaws.com -P 3306 -u root -p`

Here the RDS_MYSQL_ENDPOINT can be collected from the below screen:

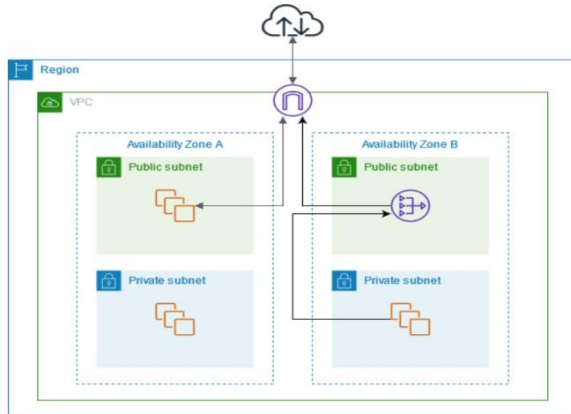
Connectivity & security		
Endpoint & port	Networking	Security
Endpoint weshopify-platform-sg.cci53w9t6upc.ap-southeast-1.rds.amazonaws.com	Availability Zone ap-southeast-1b	VPC security groups weshopify-platform-sg (sg-0c945f31ea7028d7a) Active
Port 3306	VPC weshopify-platform-vpc (vpc-032fd1f526c5aa9a79)	default (sg-0549dbb8786eaaa11) Active
	Subnet group	

NAT Gateway

Lets run the application inside the ec2 machine which we will set up in private subnet. But the application which is now in private subnet cant be accessed by the users over the internet. So the private network needs to be translate to the public network. To do this we will use the NAT Gateway.

The NAT gateway sends the traffic to the internet gateway, using its Elastic IP address as the source IP address.

Amazon Virtual Private Cloud (Amazon VPC)



Here



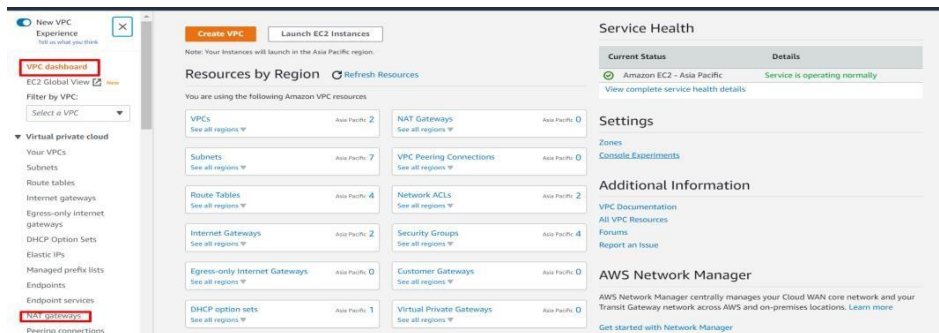
means NAT Gateway



means Internet gateway

NAT Gateway Setup

Step-1: To setup the NAT Gateway, click on the NAT Gateway option as shown below:



Step-2:

Once we click on the Nat gateways, then the screen will be opened like as below:



Step-3: when we click on Create NAT Gateway, then the following screen will be opened where we can enter the name of the nat gateway and choose the subnet. Note that always **NAT gateway should be set up in public subnet only.**

Amazon Virtual Private Cloud (Amazon VPC)

Elastic IP address 54.251.153.104 (ipalloc-09a6620375e6a5083) allocated.

VPC > NAT gateways > Create NAT gateway

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the Internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
weshopify-platform-nat-gateway
The name can be up to 256 characters long.

Subnet
Select a subnet on which to create the NAT gateway.
subnet-06f119134bc2f9879 (weshopify-public-cloud-01)

Connectivity type
Select a connectivity type for the NAT gateway.
☒ Public
☐ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
ipalloc-09a6620375e6a5083 Allocate Elastic IP

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Q Name	Q weshopify-platform-nat-gatew. X Remove

Add new tag
You can add 48 more tags.

Cancel Create NAT gateway

Step-4: Once we click on Create NAT gateway, then we will see the NAT Gateway created as shown below:

NAT gateway nat-08c24b83f6c2ba714 | weshopify-platform-nat-gateway was created successfully.

VPC > NAT gateways > nat-08c24b83f6c2ba714

nat-08c24b83f6c2ba714 / weshopify-platform-nat-gateway Delete

Details Info

NAT gateway ID nat-08c24b83f6c2ba714	Connectivity type Public	State Pending	State message <small>Info</small> -
NAT gateway ARN arn:aws:ec2:ap-southeast-1:708699230411:natgateway/nat-08c24b83f6c2ba714	Elastic IP address -	Private IP address -	Network interface ID -
VPC vpc-032fd526c5a9a79 / weshopify-platform-vpc	Subnet subnet-06f119134bc2f9879 / weshopify-public-cloud-01	Created Wednesday, September 7, 2022 at 20:26:48 GMT+5:30	Deleted -

NAT gateways (1/1) Info

Filter NAT gateways

Name	NAT gateway ID	Connectivity...	State	State message	Elastic IP address	Private IP address	Network interface ID	VPC
weshopify-platfor...	nat-08c24b83f6c2ba714	Public	Available	-	54.251.153.104	10.0.10.63	eni-0bd99b33e117f9be8	vpc-

Attach NAT Gateway to the private subnet

Step-1: go to route tables as shown below and choose the private route table in which we would like to attach the NAT Gateway.

Amazon Virtual Private Cloud (Amazon VPC)

The screenshot shows the Amazon VPC console interface. On the left, there's a sidebar with navigation options like 'VPC dashboard', 'EC2 Global View', and 'Virtual private cloud'. The main area displays 'Route tables (4)'. A table lists route tables with columns: Name, Route table ID, Explicit subnet associations, and Edge associations. The row for 'weshopify-private-cloud-route-table' is highlighted, showing it is associated with '2 subnets'.

Step-2: Once we click on private cloud-01 route table, select edit routes

The screenshot shows the 'Routes (1)' page for the selected route table. It has tabs for 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. The 'Routes' tab is active, showing a table with columns: Destination, Target, Status, and Propagated. The first row shows a route for destination '10.0.0.0/16' with target 'local' and status 'Active'. The 'Edit routes' button is highlighted with a red box.

Step-3: once we click on Edit routes, the following screen will be opened where we can add the route for the NAT Gateway as shown below:

The screenshot shows the 'Edit routes' dialog box. It has a table with columns: Destination, Target, Status, and Propagated. The first row shows a route for destination '10.0.0.0/16' with target 'local' and status 'Active'. The second row shows a route for destination '0.0.0.0/0' with target 'NAT Gateway' and status 'No'. The 'Add route' button is highlighted with a red box.

Click on save changes.

Amazon Virtual Private Cloud (Amazon VPC)

Testing NAT Gateway Connection

Step-1: Create an EC2 machine in the private subnet

To Test NAT Gateway connection, we have to create an ec2 machine in private subnet as shown below, as we have attached the NAT Gateway to the private subnet's route table.

EC2 > Instances > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

weshopify-platform-private-cloud-01-instance

Add additional tags

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

5

Browse more AMIs

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type

Free tier eligible

Description

Canonical, Ubuntu, 20.04 LTS, amd64 focal image build on 2022-06-10

Architecture

64-bit (x86)

AMI ID

ami-04ff9e9b51c1f62ca

Verified provider

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

weshopify-platform-app-machine-key

Create new key pair

Amazon Virtual Private Cloud (Amazon VPC)

▼ Network settings Info

VPC - required Info
vpc-032fdf526c5aa9a79 (weshopify-platform-vpc)
10.0.0.0/16

Subnet - required Info
subnet-0d59a51d44c960fd3 weshopify-private-cloud-01
VPC: vpc-032fdf526c5aa9a79 Owner: 708699230411
Availability Zone: ap-southeast-1a IP addresses available: 250 CIDR: 10.0.30.0/24

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups Info
Select security groups
weshopify-platform-public-sg sg-0e8da0631df9f411b X
VPC: vpc-032fdf526c5aa9a79

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

Note

Please note a small correction in the above screen shot i.e. Auto-assign public IP chosen as Enable but it must be as "Disable". because we should not connect to the ec2 machine in the private cloud from an external over the internet.

Click on launch instances

Step-2: Create an EC2 machine in the public subnet

Repeat the same steps as above but while editing the networking settings, choose the public subnet.

Step-3: Connect the private cloud ec2 machine from the public cloud ec2 machine

First connect to public machine using the moba xterm or directly AWS Connect button over the web.

Session settings

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL

Basic SSH settings

Remote host 18.136.207.64 Specify username ubuntu Port 22

Advanced SSH settings Terminal settings Network settings Bookmark settings

☒ X11-Forwarding ☒ Compression Remote environment: Interactive shell

Execute command: Do not exit after command ends

SSH-browser type: SFTP protocol Compress data transferred through the SSH channel (mental)

☒ Use private key D:\personals\trainings\devops_w... ☐ Adapt locales on remote server

Execute macro at session start: <none>

OK Cancel

When we have created the ec2 machine in private cloud , we must have downloaded the key to connect to the machine as shown below:

Amazon Virtual Private Cloud (Amazon VPC)

> personals > trainings > devops_with_aws > ec2_key_pairs >

Name	Date modified	Type	Size
region_virginia	4/30/2022 11:06 AM	File folder	
apache-kafka-machines-key.pem	4/30/2022 11:05 AM	PEM File	2 KB
aws-admin_user_credentials.csv	4/13/2022 8:40 AM	Comma Separated V...	1 KB
devops-7pm-ci-cd-key.pem	7/25/2022 7:46 PM	PEM File	2 KB
docker-devops-8pm-machine.pem	5/23/2022 9:12 PM	PEM File	2 KB
linux_7pm_private_key.pem	6/8/2022 7:50 PM	PEM File	2 KB
linux_7pm_private_key.ppk	6/9/2022 7:50 PM	PPK File	2 KB
linux_practice_machine_keypair.pem	3/30/2022 8:37 PM	PEM File	2 KB
linux_practice_machine_keys_nvirginia.pem	4/6/2022 9:11 PM	PEM File	2 KB
medialab-jenkins-machines-private-key-nvirginia...	4/23/2022 8:41 PM	PEM File	2 KB
morning-docker-7am.pem	7/25/2022 7:18 AM	PEM File	2 KB
weshopify-platform-app-machine-key.pem	9/3/2022 8:52 PM	PEM File	2 KB
weshopify-platform-keys.pem	Type: PEM File	PEM File	2 KB
weshopify-platform-keys.ppk	Size: 1.63 KB	PPK File	2 KB

On public cloud machine, upload create a pem file with the content of the above file or otherwise upload this pem file as shown below:

```
root@ip-10-0-20-59:~# ls -l
total 8
-rw----- 1 root root 1679 Sep  8 15:24 privatemachine.pem
drwx----- 4 root root 4096 Sep  3 15:40 snap
root@ip-10-0-20-59:~#
```

Once after the file is created i.e. like above `privatemachine.pem` on the public machine connect to the private machine, change its permission to **`chmod 600 privatemachine.pem`**

This means providing the read and write permissions on the file to only the owner who created it.

Once after the file is uploaded/created, lets use the below command to connect to the private cloud machine from public cloud machine.

Remember as the both the machines are in same VPC i.e. our custom VPC, the machines able to communicate each other using the private ip address.

Command: `ssh -i <pem_file_of_private_cloud_machine> <user_name>@<private_ip>`

`ssh -i privatemachine.pem ubuntu@10.0.30.194`

```
root@ip-10-0-20-59:~# ls -l
total 8
-rw----- 1 root root 1679 Sep  8 15:24 privatemachine.pem
drwx----- 4 root root 4096 Sep  3 15:40 snap
root@ip-10-0-20-59:~# ssh -i privatemachine.pem ubuntu@10.0.30.194
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Sep 10 15:05:04 UTC 2022

System load:  0.0               Processes:    101
Usage of /:   27.6% of 7.57GB   Users logged in: 1
Memory usage: 20%              IPv4 address for eth0: 10.0.30.194
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
  https://ubuntu.com/aws/pro

24 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Sep 10 15:02:24 2022 from 10.0.20.59
ubuntu@ip-10-0-30-194:~$
```

Set up the application on the private machine as shown below:

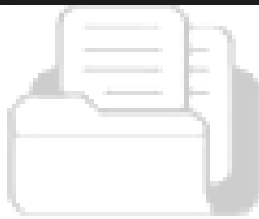
Amazon Virtual Private Cloud (Amazon VPC)

```
root@ip-10-0-30-194:/opt/weshopify-platform-monolith-app# ls -l
total 36
-rw-r--r-- 1 root root 33 Sep 10 15:15 README.md
-rw-r--r-- 1 root root 10284 Sep 10 15:15 mvnw
-rw-r--r-- 1 root root 6734 Sep 10 15:15 mvnw.cmd
-rw-r--r-- 1 root root 3493 Sep 10 15:15 pom.xml
drwxr-xr-x 4 root root 4096 Sep 10 15:15 src
drwxr-xr-x 9 root root 4096 Sep 10 15:20 target
root@ip-10-0-30-194:/opt/weshopify-platform-monolith-app# cd target/
root@ip-10-0-30-194:/opt/weshopify-platform-monolith-app/target# ls -l
total 88960
drwxr-xr-x 5 root root 4096 Sep 10 15:19 classes
drwxr-xr-x 3 root root 4096 Sep 10 15:19 generated-sources
drwxr-xr-x 3 root root 4096 Sep 10 15:19 generated-test-sources
drwxr-xr-x 2 root root 4096 Sep 10 15:19 maven-archiver
drwxr-xr-x 3 root root 4096 Sep 10 15:19 maven-status
drwxr-xr-x 3 root root 4096 Sep 10 15:19 test-classes
drwxr-xr-x 4 root root 4096 Sep 10 15:19 we-shopify-platform
-rw-r--r-- 1 root root 49497803 Sep 10 15:20 we-shopify-platform.war
-rw-r--r-- 1 root root 41562172 Sep 10 15:20 we-shopify-platform.war.original
root@ip-10-0-30-194:/opt/weshopify-platform-monolith-app/target# java -version
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
root@ip-10-0-30-194:/opt/weshopify-platform-monolith-app/target# java -jar we-shopify-platform.war
```

Now login back to public machine in next window of the mobaxterm and do a curl to the private machine where the application is running as shown below then we should be able to access the application:

```
root@ip-10-0-20-59:~# curl 10.0.30.194:5001
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Login</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="..assets/vendor/bootstrap/css/bootstrap.min.css">
  <link href="..assets/vendor/fonts/circular-std/style.css" rel="stylesheet">
  <link rel="stylesheet" href="..assets/libs/css/style.css">
  <link rel="stylesheet" href="..assets/vendor/fonts/fontawesome/css/fontawesome-all.css">
  <style>
    html,
    body {
      height: 100%;
    }
  </style>
</head>
```



TechHubVault