AWS High Availability Set up

Kiran

May 13, 2021

Contents

1 AWS HA Set up

This document lists out the steps to create a *Highly Available* AWS infrastructure in *nondefault* VPC.

1.1 High Availability

High Availability is the characteristic of a system which aims to provide an agreed level of system uptime.

Important principles of High Availability

1. Elimination of Single Point of Failure
2. Reliable fail-over system.
3. Failure detection

1.2 AWS VPC nondefault setup

In every aws account, there is a *default* VPC in every region. In the *default* VPC there is one subnet per each Availability Zone. Any instances created in this subnet do have inbound and outbound connectivity - meaning - the instances can access any data on the internet and anyone can access the ec2 instances (provided the Security Groups are configured).

While, it's easy to manage *default* VPC, IT organizations create and manage their own VPCs called as *nondefault* VPCs.

nondefault VPCs are different from different from *default* VPCs. *nondefault* VPCs don't have inbound or outbound connectivity to the internet by default.

To be able to establish the connectivity, we need to use Gateway services.

Every *nondefault* VPC should have an Internet Gateway for the public subnets and a NAT Gateway for Private subnets.

Public subnet is the one where the instances can send outbound traffic to internet and can be accessed from the internet subject to the Security Group rules. Generally used for Public facing application, Bastion hosts and by Classic Load balancers.

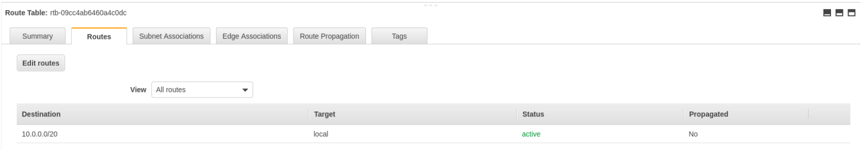
Private Subnets are the ones where the instances cannot be accessed directly from the internet. The outbound traffic is routed through NAT Gateway.

1.3 Routing

1.3.1 Route Table

A Route table is table that defines the paths for subnets inside a VPC. For instance, the Route table for Public subnets contain the routes for the instances.

By default a main route table is created for every VPC. The Main route table has one route

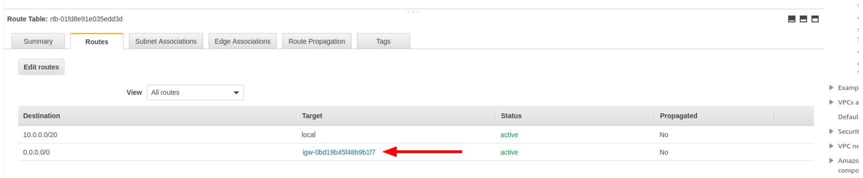


Route Table: rtb-09cc4ab6460a4c0dc

Destination	Target	Status	Propagated
10.0.0.0/20	local	active	No

The default entry in the table enables the instances in the VPC to communicate with each other.

Let's take a look at Public Route table



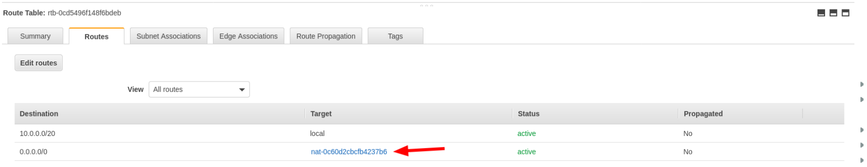
Route Table: rtb-01fd8e91e035edd3d

Destination	Target	Status	Propagated
10.0.0.0/20	local	active	No
0.0.0.0/0	igw-0bd19b45f48b9b1f7	active	No

In the above table, there are 2 entries. The First entry is the same as the one in the main route table.

Whereas, the second route routes all the subnet traffic to the Interet over the internet gateway.

Note: Before this route table entry is created, Internet Gateway should already be created.



Route Table: rtb-0cd5496f148f6bdeb

View: All routes

Destination	Target	Status	Propagated
10.0.0.0/20	local	active	No
0.0.0.0/0	nat-0c60d2cbcfb4237b6	active	No

In the above table, the first entry is the default route, the second one sends all the subnet traffic bound to internet to the NAT gateway.

Note: Before this route table entry is created, NAT Gateway should already be created.

More information on Routing can be found here.

1.4 Activity - Build a Web Application in HA Set up.

To set up Highly Available (HA) Web server, it's recommended to create redundant ec2 instances in different AZs. For this excercise, let's create 2 ec2 instances in 2 Availability Zones front-ended by a Classic Load Balancer.

Based on the traffic/load, you may use more than 2 ec2 instances for one tier of the servers.

We are going to put the web servers in the private subnets and use a Bastion host to access the web servers.

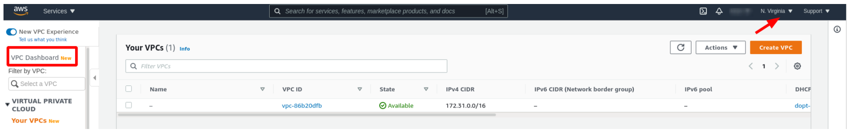
The following is the summary of resources we are going to create:

I. VPC II. Public Subnets - 2 in 2 Availability Zones III. Private Subnets - 2 in 2 Availability Zones. IV. Internet Gateway - To be associated with the Public Route Table V. NAT Gateway - To be associated with the Private Route Table VI. Public Route Table - Assoicated to the Public Subnets VII. Private Route Table - Associated to teh Private Subnets VII. ec2 instnaces: a. 1 bastion - To access other instances in private subnets. This will be created in one of the public subnets b. 2 Web - This will serve the web pages VIII. Load balancer - To Load balance the traffic from internet to ec2 instances. IX. Security Groups a. Web NSG - For ec2 instances running Web server b. LB NSG - For the Classic Load balancer. c. Bastion NSG - For the bastion ec2 instance through which we'll connect to other ec2 instances

Let's create the resources in order.

1.4.1 Create a VPC:

To create a VPC selec VPC from the services and make sure you're in the desired region and click on create VPC



Your VPCs (1) Info

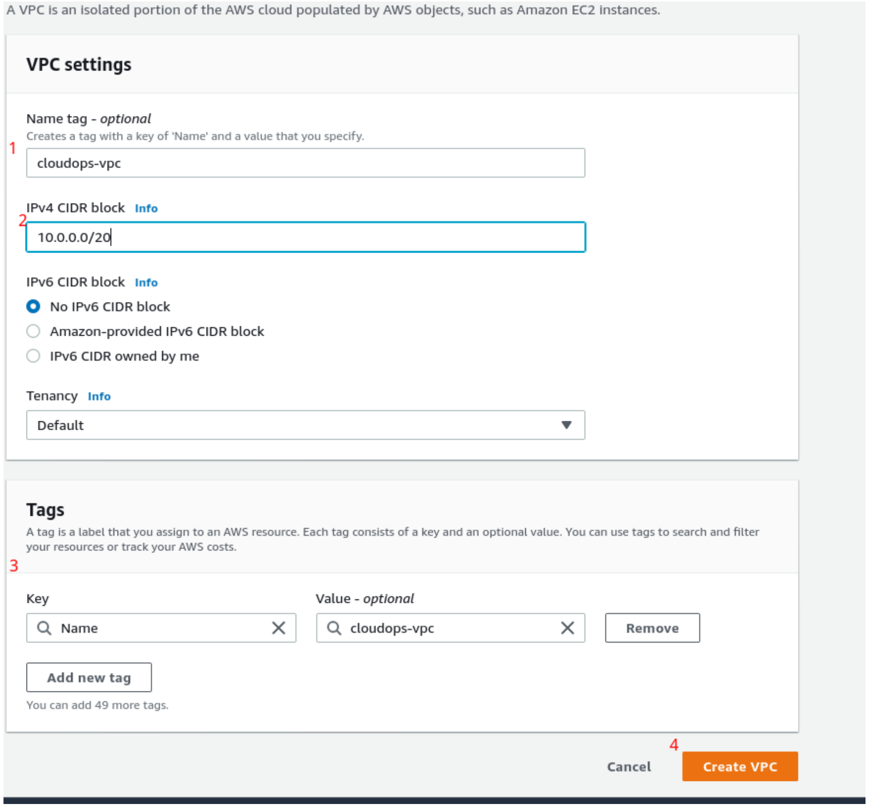
□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)	IPv6 pool	DHCF
□	–	vpc-86b20dfb	✓ Available	172.31.0.0/16	–	–	dopt-

1. Enter the Name of the VPC.

2. Choose a CIDR Range.

3. Enter Tags (optional)

4. Click on Create.


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

VPC settings

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

1 cloudops-vpc

IPv4 CIDR block Info

2 10.0.0.0/20

IPv6 CIDR block Info

● No IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info

Default ▼

Tags

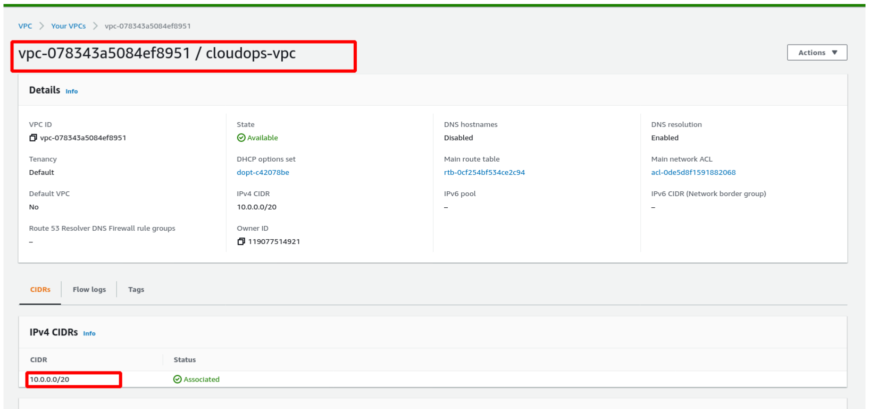
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

3

Key	Value - *optional*
Name	cloudops-vpc

You can add 49 more tags.

Upon succesful creation of VPC, you should see the details as below.



vpc-078343a5084ef8951 / cloudops-vpc

Details Info

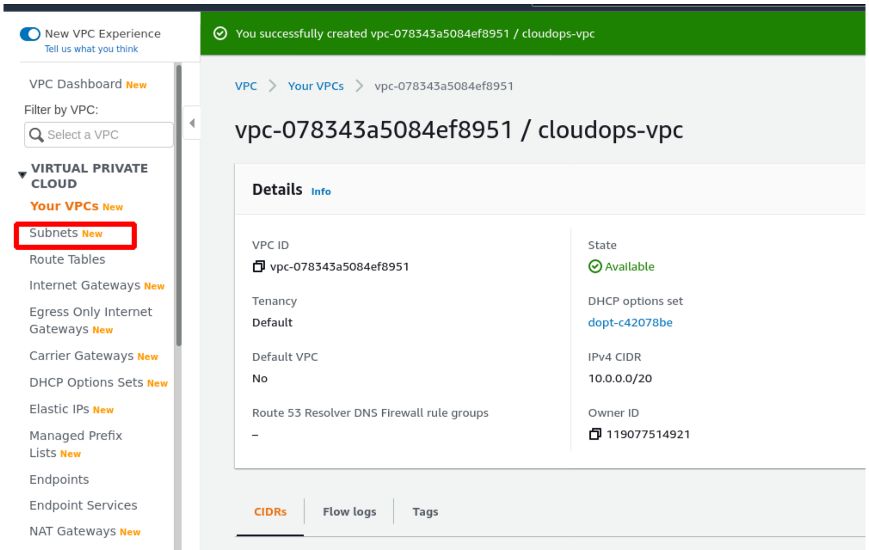
VPC ID vpc-078343a5084ef8951	State ✓ Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-c42078be	Main route table rtb-0cf254bf534ce2c94	Main network ACL acl-0de5d8f1591882068
Default VPC No	IPv4 CIDR 10.0.0.0/20	IPv6 pool –	IPv6 CIDR (Network border group) –
Route 53 Resolver DNS Firewall rule groups –	Owner ID 119077514921		

IPv4 CIDRs Info

CIDR	Status
10.0.0.0/20	✓ Associated

Let's create subnets in the VPC.

To create subnets, click on `subnets` in the left pane of the VPC dashboard.



You successfully created vpc-078343a5084ef8951 / cloudops-vpc

vpc-078343a5084ef8951 / cloudops-vpc

Details Info

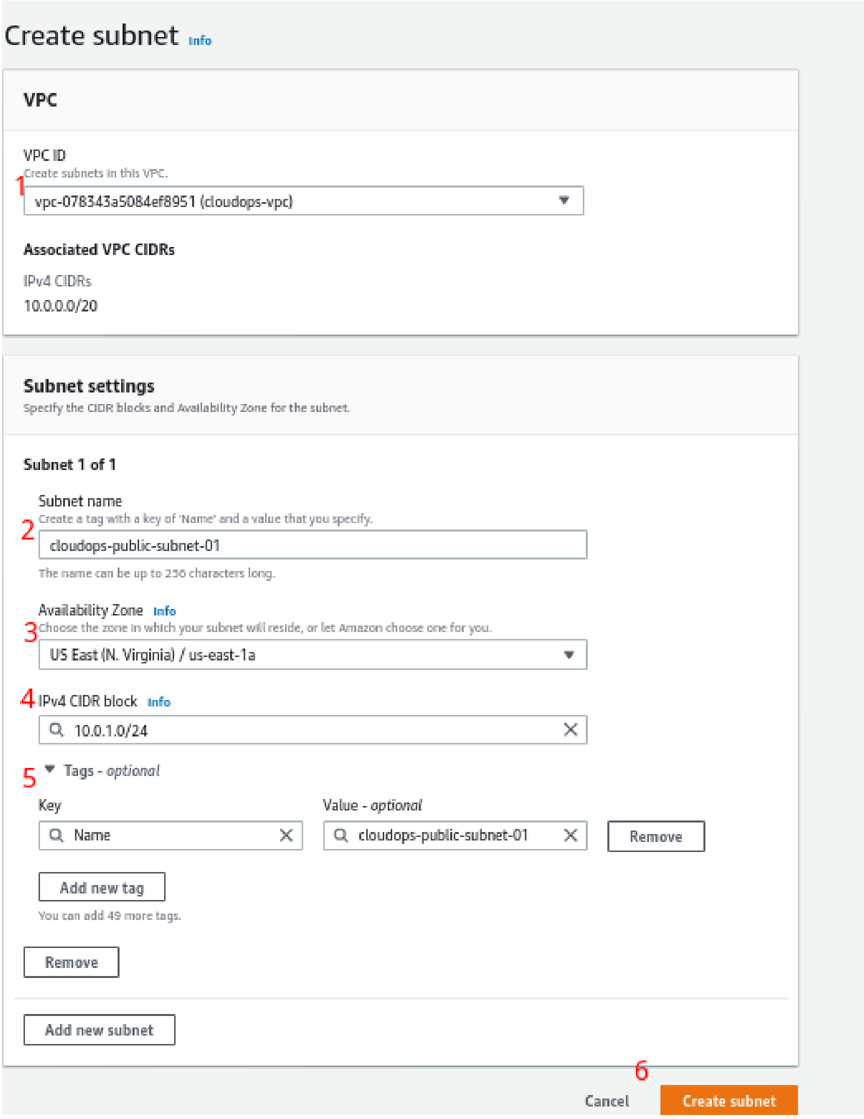
VPC ID vpc-078343a5084ef8951	State ✓ Available
Tenancy Default	DHCP options set dopt-c42078be
Default VPC No	IPv4 CIDR 10.0.0.0/20
Route 53 Resolver DNS Firewall rule groups –	Owner ID 119077514921

Then click on Create subnet and fill in the details as below

1. Select the VPC
2. Enter the Name of the subnet.
3. Pick an Availability Zone.
4. Select CIDR Range for the subnet.

5. Enter Tags (optional)

6. Click on Create



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-078343a5084ef8951 (cloudops-vpc)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/20

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.
cloudops-public-subnet-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.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info
10.0.1.0/24

Tags - *optional*

Key	Value - *optional*
Name	cloudops-public-subnet-01

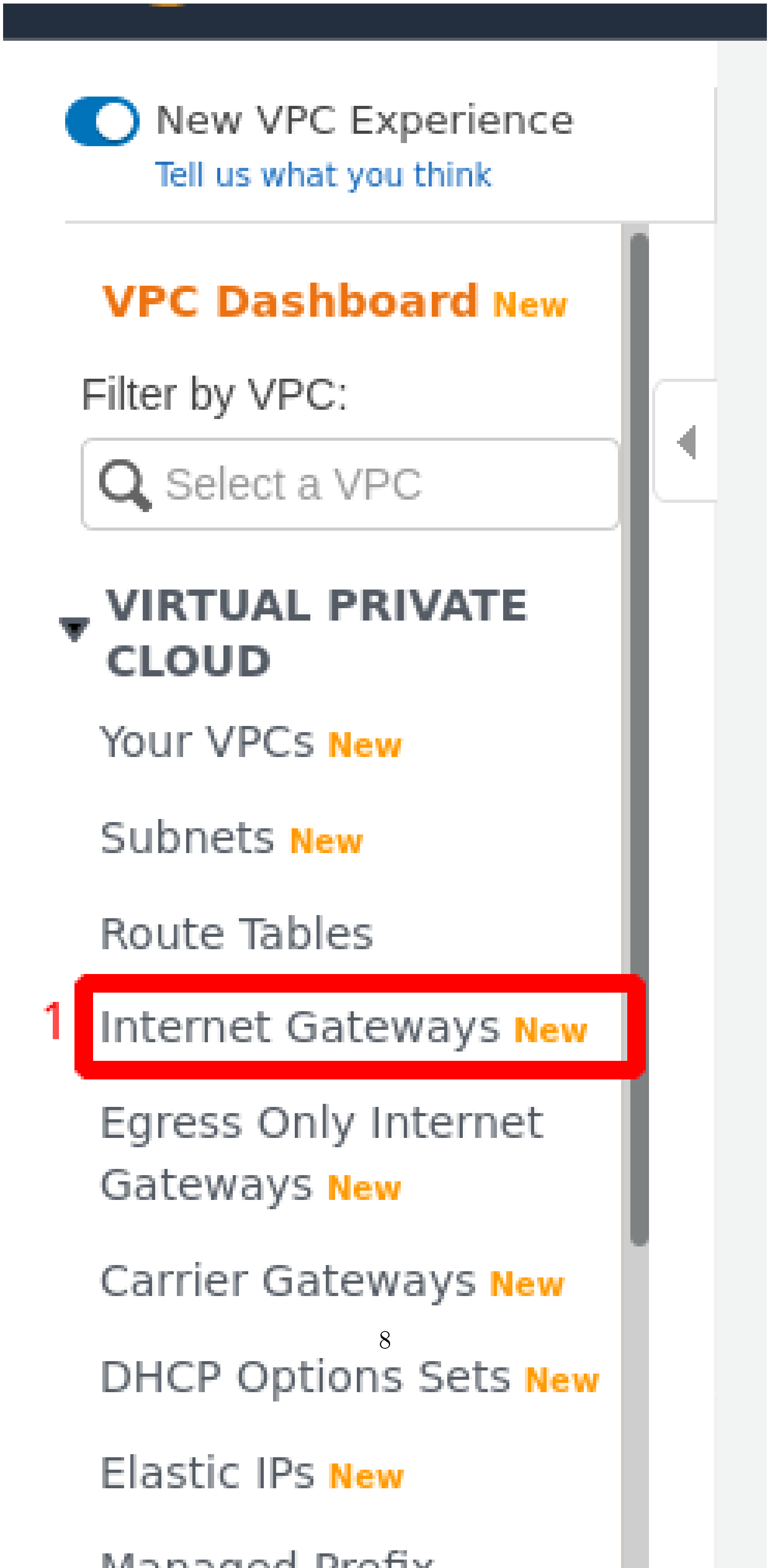
You can add 49 more tags.

Following the above process create the following subnets in the cloudops-vpc

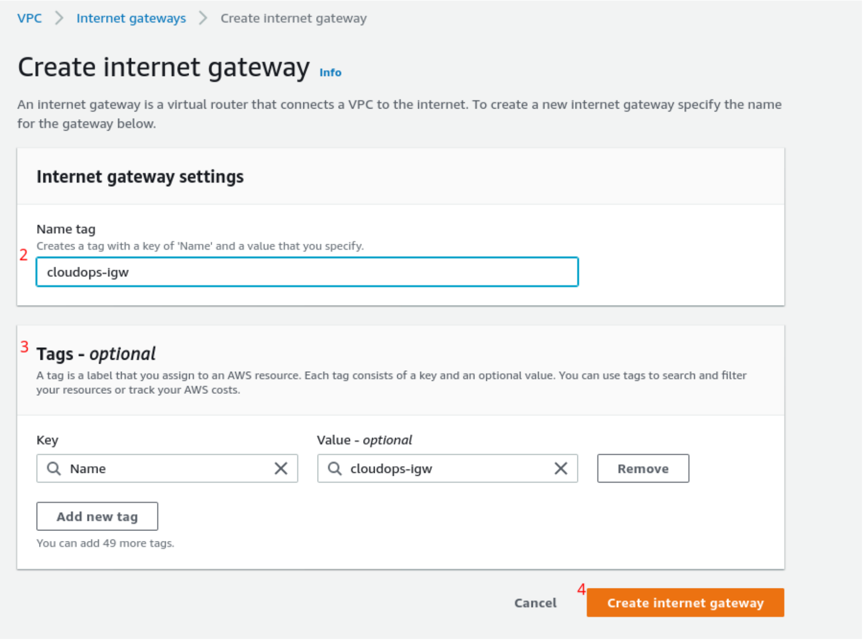
Subnet Name	Availability Zone	CIDR Range	Private/Public
cloudops-public-subnet-01	us-east-1a	10.0.1.0/24	Public
cloudops-public-subnet-02	us-east-1b	10.0.2.0/24	Private
cloudops-web-subnet-01	us-east-1a	10.0.3.0/24	Private
cloudops-web-subnet-02	us-east-1b	10.0.4.0/24	Private

1.4.2 Create Internet Gateway.

1. Click on the Internet Gateways in the left menu of the VPC dashboard


New VPC Experience
Tell us what you think
VPC Dashboard New
Filter by VPC:
Select a VPC
VIRTUAL PRIVATE CLOUD
Your VPCs New
Subnets New
Route Tables
1
Internet Gateways New
Egress Only Internet Gateways New
Carrier Gateways New
DHCP Options Sets New
Elastic IPs New
Managed Prefix

1. Click on `Create Internet Gateway`
2. Enter Name of the IGW
3. Enter tags (optional)



VPC > Internet gateways > Create internet gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

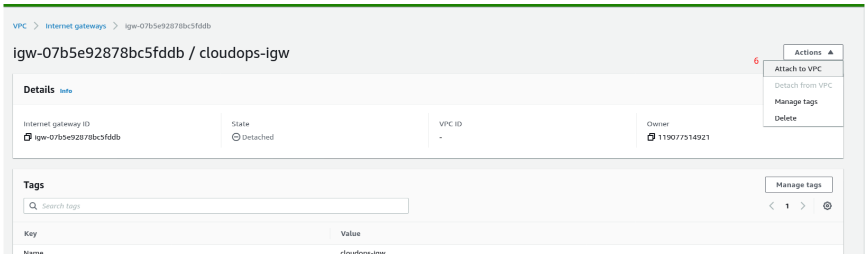
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
2 cloudops-igw

3 Tags - *optional*
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	cloudops-igw

You can add 49 more tags.

1. Once the IGW is created, attach it to the cloudops-vpc



VPC > Internet gateways > igw-07b5e92878bc5fddb

igw-07b5e92878bc5fddb / cloudops-igw

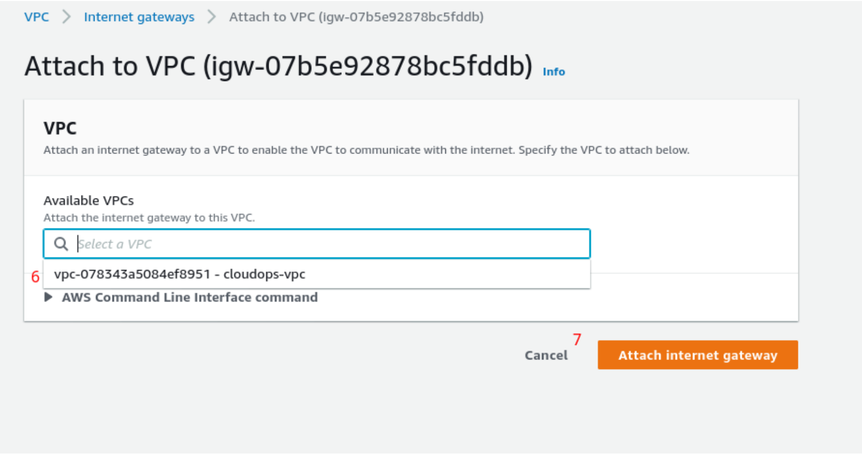
Details Info

Internet gateway ID	State	VPC ID	Owner
igw-07b5e92878bc5fddb	Detached	-	119077514921

Tags

Key	Value
Name	cloudops-igw

1. Select the cloud-ops VPC
2. Click on Attach VPC



VPC > Internet gateways > Attach to VPC (igw-07b5e92878bc5fddb)

Attach to VPC (igw-07b5e92878bc5fddb) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

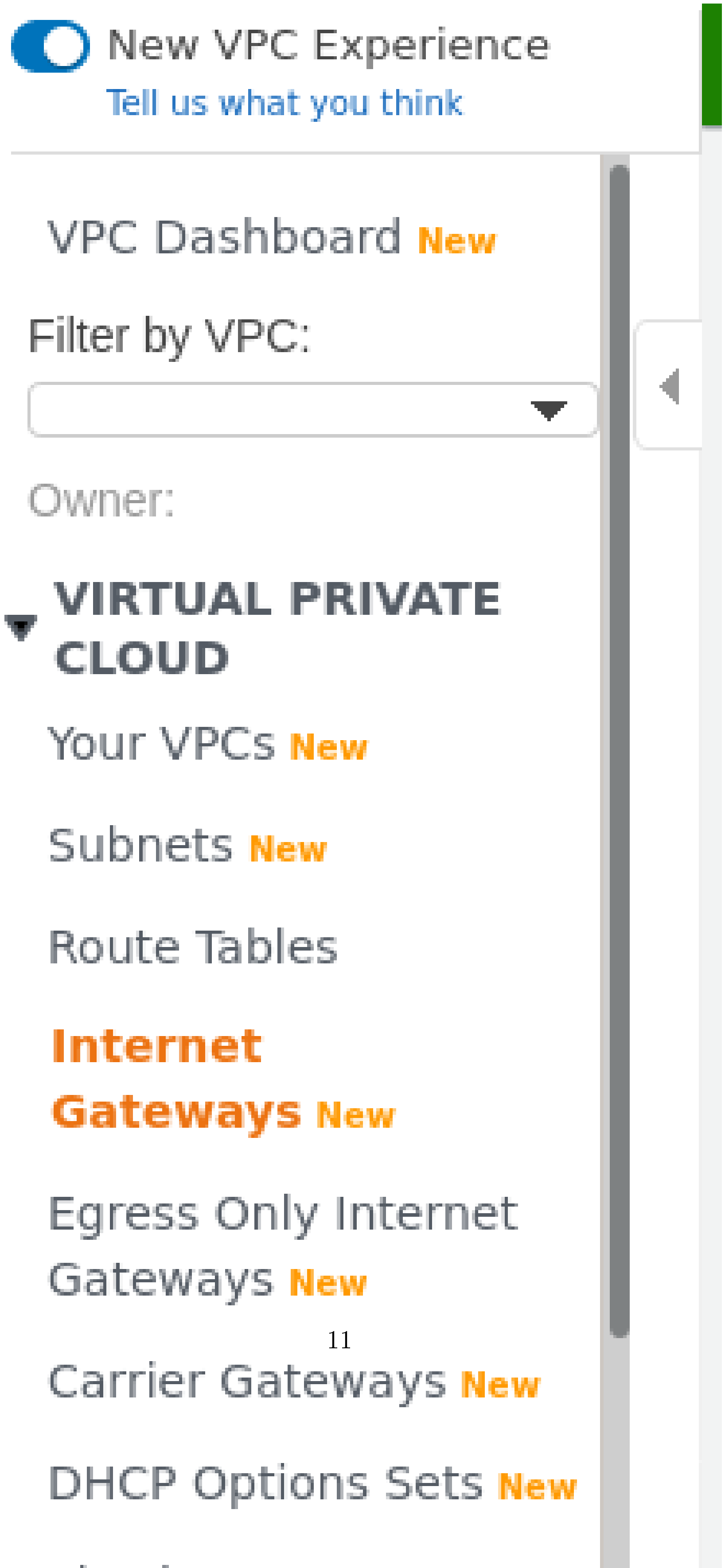
Select a VPC

6 vpc-078343a5084ef8951 - cloudops-vpc

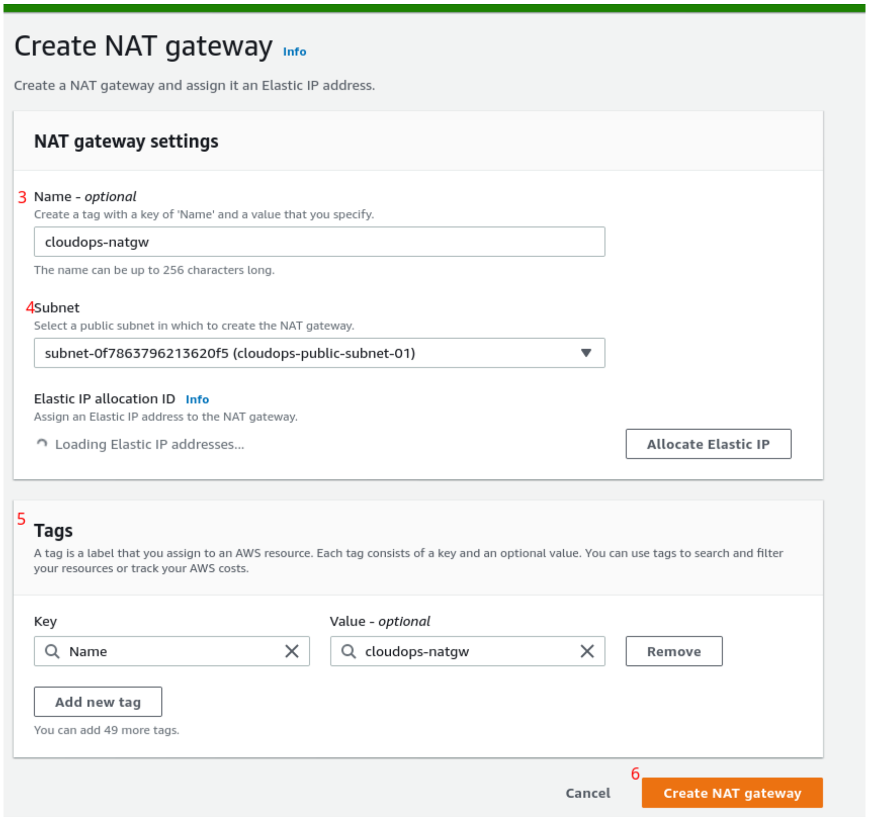
▶ AWS Command Line Interface command
7
Cancel
Attach internet gateway

1.4.3 Create NAT Gateway

1. Click on the NAT Gateways in the left menu of the VPC dashboard


New VPC Experience
Tell us what you think
VPC Dashboard New
Filter by VPC:
Owner:
VIRTUAL PRIVATE CLOUD
Your VPCs New
Subnets New
Route Tables
Internet Gateways New
Egress Only Internet Gateways New
Carrier Gateways New
DHCP Options Sets New

1. Click on `Create NAT Gateway`
2. Enter the name of the NAT gateway
3. Select one of the Private Subnets & Allocate an IP Address
4. Enter Tags (optional)
5. Click on `Create NAT Gateway`



Create NAT gateway Info

Create a NAT gateway and assign it an Elastic IP address.

NAT gateway settings

3 Name - *optional*

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

cloudops-natgw

The name can be up to 256 characters long.

4 Subnet

Select a public subnet in which to create the NAT gateway.

subnet-0f7863796213620f5 (cloudops-public-subnet-01) ▼

Elastic IP allocation ID Info

Assign an Elastic IP address to the NAT gateway.

Loading Elastic IP addresses...

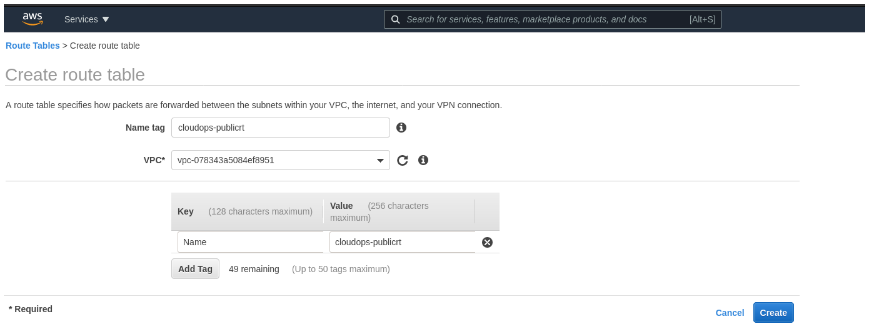
5 ### Tags

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	cloudops-natgw

You can add 49 more tags.

1.4.4 Create Route Tables



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]

Route Tables > Create route table

Create route table

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

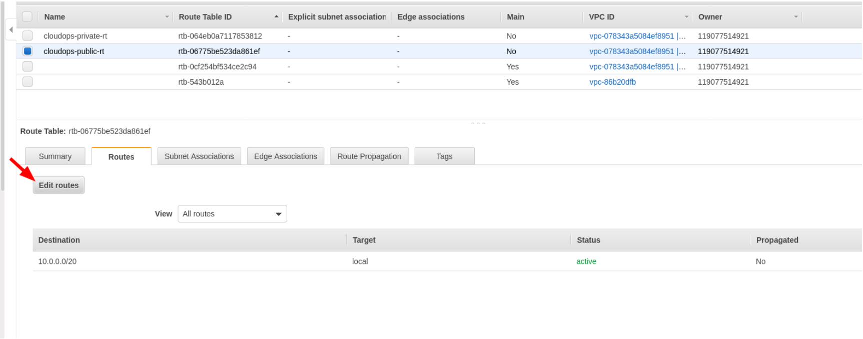
Name tag: cloudops-publicrt

VPC*: vpc-078343a5084ef8951

Key (128 characters maximum)	Value (256 characters maximum)
Name	cloudops-publicrt

Add Tag
49 remaining (Up to 50 tags maximum)
* Required
Cancel
Create

1. Create a Public Route Table with the following routes



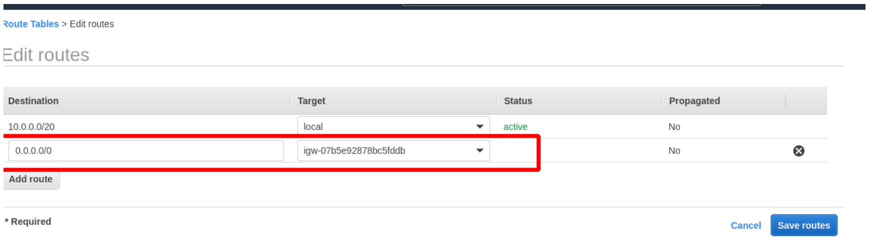
Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
cloudops-private-rt	rtb-064eb0a7117853812	-	-	No	vpc-078343a5084ef8951 |...	119077514921
cloudops-public-rt	rtb-06775be523da861ef	-	-	No	vpc-078343a5084ef8951 |...	119077514921
	rtb-0cf254bf534ce2c94	-	-	Yes	vpc-078343a5084ef8951 |...	119077514921
	rtb-543b012a	-	-	Yes	vpc-86b20dfb	119077514921

Route Table: rtb-06775be523da861ef

View: All routes

Destination	Target	Status	Propagated
10.0.0.0/20	local	active	No

1. Create a Private Route table with the following routes



Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/20	local	active	No
0.0.0.0/0	igw-07b5e92878bc5fddb		No

1. Create an ec2 instance called `cloudops-bastion` in `cloudops-public-subnet-01` with a Public IP.

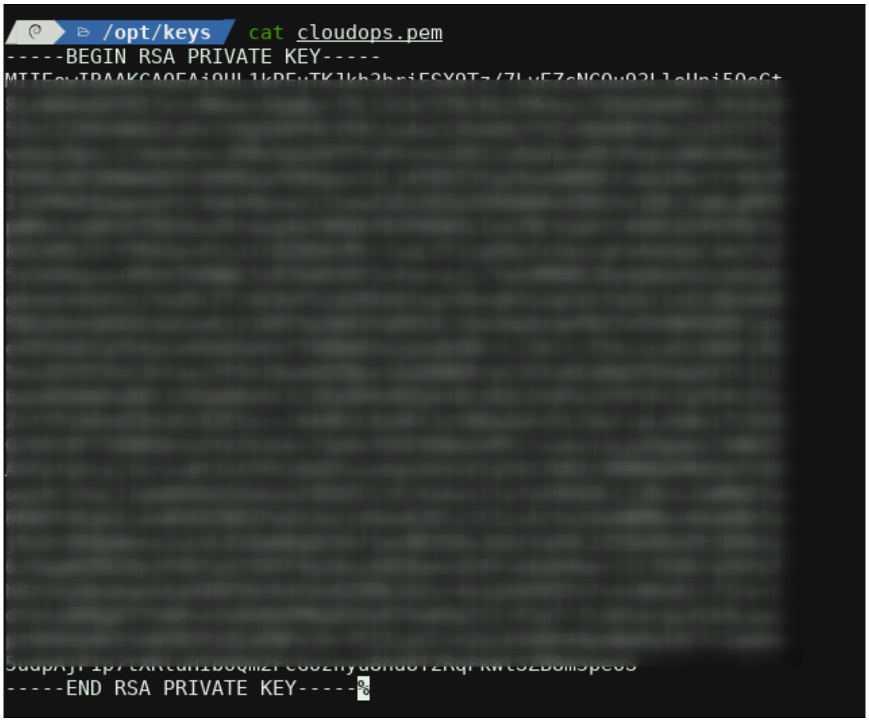
1. Create 2 ec2 instances `cloudops-web-01` in `cloudops-web-subnet01` `cloudops-web-02` in `cloudops-web-subnet02`

Note: Make sure you use the same pem key for the three instances.

1.4.5 Accessing the Web servers from the bastion

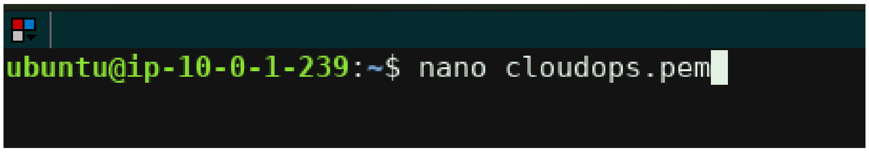
1. Connect to the bastion from your terminal or Git bash

2. Copy the contents of the pem key in your local machine (laptop).



```
/opt/keys  cat cloudops.pem
-----BEGIN RSA PRIVATE KEY-----
[redacted]
-----END RSA PRIVATE KEY-----%
```

1. In the bastion ec2, open a new file with the command `nano demokey.pem` (replace the `demokey.pem` with the name of your pem key)



```
ubuntu@ip-10-0-1-239:~$ nano cloudops.pem
```

or alternatively run the following command from the location where you stored the pem key on your machine

```
scp -i "cloudops.pem" cloudops.pem ubuntu@<publicIP>:~
```

Example:

```
scp -i "cloudops.pem" cloudops.pem ubuntu@3.80.235.140:~
```

1. Make sure you change the permissions of the pem key to 400

```
chmod 400 cloudops.pem
```

1.4.6 Install Apache2 on ec2 instances

From the bastion instance connect to web instance usign private IP

```
ssh -i cloudops.pem <privateIPofWebInstance>
```

Example:

```
ssh -i "cloudops.pem" ubuntu@10.0.3.139
```

Once inside the web instnace, run the following command to install apache web server.

```
sudo apt-get update && sudo apt-get install apache2 -y
```

Confirm if apache is installed by running the below command

```
curl http://localhost
```

This command outputs a long html file.
Replace index.html with your own homepage
Creat a file with the below contents using nano

```
nano index.html
```

Copy the below contents inside the `index.html` file

```
<html>
<title>CloudOps</title>
<body>
<h1> Welcome to the World of Cloud Ops</h1>
<h2> Automation is fun </h2>
<h3> Cloud is Great </h3>
<h4> This is webserver 1 </h4>

</body>
</html>
```

Now copy the `index.html` to `/var/www/html` by running the below command

```
sudo cp index.html /var/www/html/
```

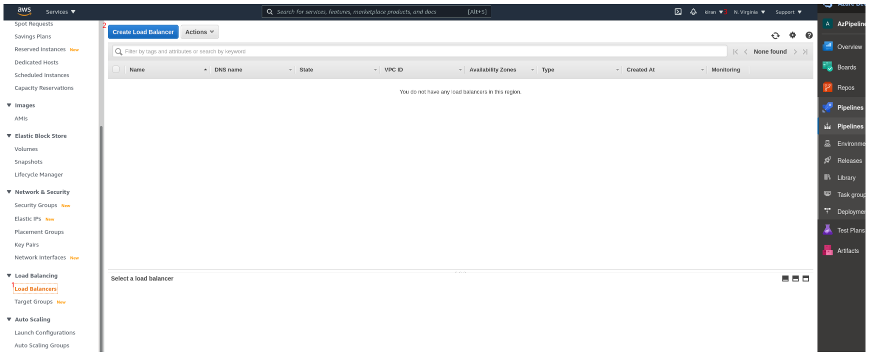
Confirm if the home page is updated by running the below command

```
curl http://localhost
```

You should see the same content that you copied to `index.html` file. Repeat the above steps in Web server 2.

1.4.7 Create Classic Load balancer

1. To create Classic Load balancer, go to ec2 dashboard and scroll down in the left pane and click on `Load balancers`.
2. In the Load balancers pane, click on `Create Load Balancer`
3. Make sure your in the same region as your ec2 instances

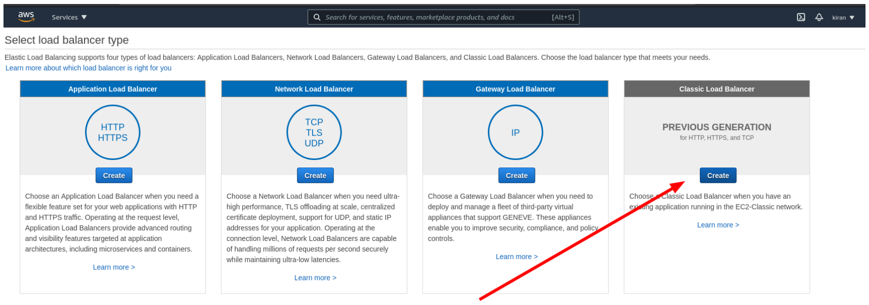


Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring

You do not have any load balancers in this region.

Select a load balancer

1. Click on Classic Load balancer


aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
kiran
HTTP
HTTPS
Create
TCP
TLS
UDP
Create
IP
Create
Create

Select load balancer type

Elastic Load Balancing supports four types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. Choose the load balancer type that meets your needs.
Learn more about which load balancer is right for you

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Learn more >

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Learn more >

Gateway Load Balancer

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Learn more >

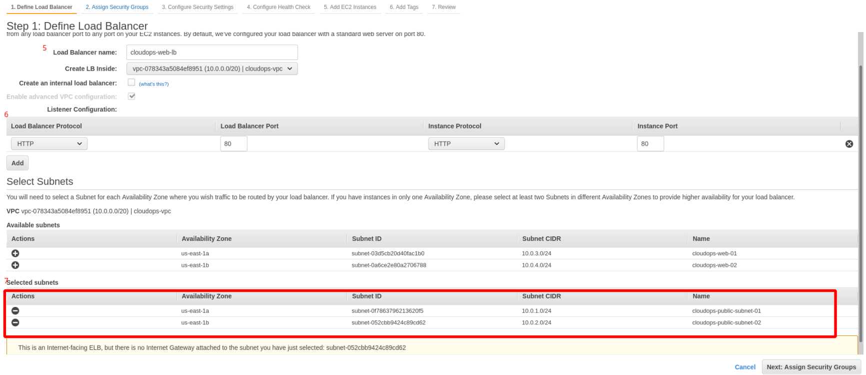
Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Learn more >

1. In the next screen, Enter the name of the load balancer.
2. Select the Right VPC
3. Select Load balancer protocol as 80
4. Select 2 Public Subnets in each AZ



1. Define Load Balancer | 2. Assign Security Groups | 3. Configure Security Settings | 4. Configure Health Check | 5. Add EC2 Instances | 6. Add Tags | 7. Review

Step 1: Define Load Balancer

from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

5 Load Balancer name: cloudops-web-lb

Create LB Inside: vpc-078343a5084ef8951 (10.0.0.0/20) | cloudops-vpc

Create an internal load balancer: □ (what's this?)

Enable advanced VPC configuration: ☑

Listener Configuration:

6

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Add

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-078343a5084ef8951 (10.0.0.0/20) | cloudops-vpc

Available subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊕	us-east-1a	subnet-03d5cb20d40fac1b0	10.0.3.0/24	cloudops-web-01
⊕	us-east-1b	subnet-0a6ce2e80a2706788	10.0.4.0/24	cloudops-web-02

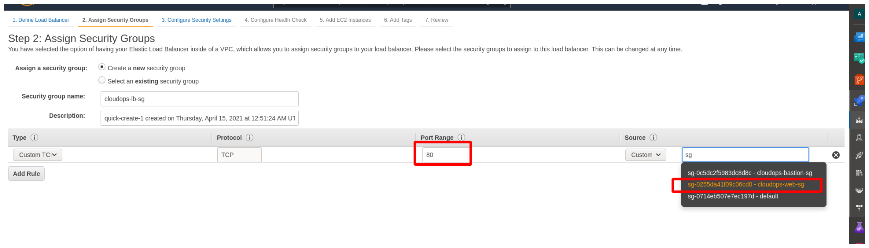
7 Selected subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊖	us-east-1a	subnet-0f7863796213620f5	10.0.1.0/24	cloudops-public-subnet-01
⊖	us-east-1b	subnet-052cbb9424c89cd62	10.0.2.0/24	cloudops-public-subnet-02

This is an Internet-facing ELB, but there is no Internet Gateway attached to the subnet you have just selected: subnet-052cbb9424c89cd62

Cancel | Next: Assign Security Groups

1. In the next screen create a new Security Group for Load balancer. Make sure you select port 80 in Port Range and in the source select the Security Group you created for web instances



1. Define Load Balancer | 2. Assign Security Groups | 3. Configure Security Settings | 4. Configure Health Check | 5. Add EC2 Instances | 6. Add Tags | 7. Review

Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: cloudops-lb-sg

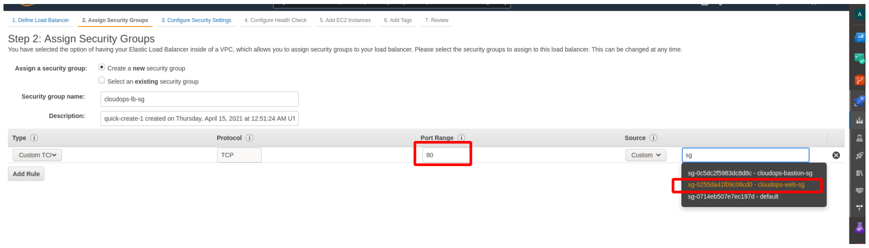
Description: quick-create-1 created on Thursday, April 15, 2021 at 12:51:24 AM UT

Type	Protocol	Port Range	Source	
Custom TC	TCP	80	Custom	sg

Add Rule

sg-0c5dc2f5983dc8d8c - cloudops-bastion-sg
sg-0255da41f09c06cd0 - cloudops-web-sg
sg-0714eb507e7ec197d - default

1. Configure Health check as below



1. Define Load Balancer | 2. Assign Security Groups | 3. Configure Security Settings | 4. Configure Health Check | 5. Add EC2 Instances | 6. Add Tags | 7. Review

Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: cloudops-lb-sg

Description: quick-create-1 created on Thursday, April 15, 2021 at 12:51:24 AM UT

Type: Custom TC | Protocol: TCP | Port Range: 80 | Source: Custom | sg

Add Rule

sg-0c5dc2f5983dc8d8c - cloudops-bastion-sg
sg-0255da41f09c06cd0 - cloudops-web-sg
sg-0714eb507e7ec197d - default

1. In the next page, select both the ec2 instances.
2. In hte next page, enter tags and click on Create.