

Creating Networking Resources in an Amazon Virtual Private Cloud (VPC)

Objectives

In this lab, you will:

- Summarize the customer scenario
- Create a VPC, Internet Gateway, Route Table, Security Group, Network Access List, and EC2 instance to create a routable network within the VPC
- Familiarize yourself with the console
- Develop a solution to the customers issue found within this lab.

The lab is complete once you can successfully utilize the command ping outside the VPC.

Duration

This lab total duration is 60 minutes.

Scenario

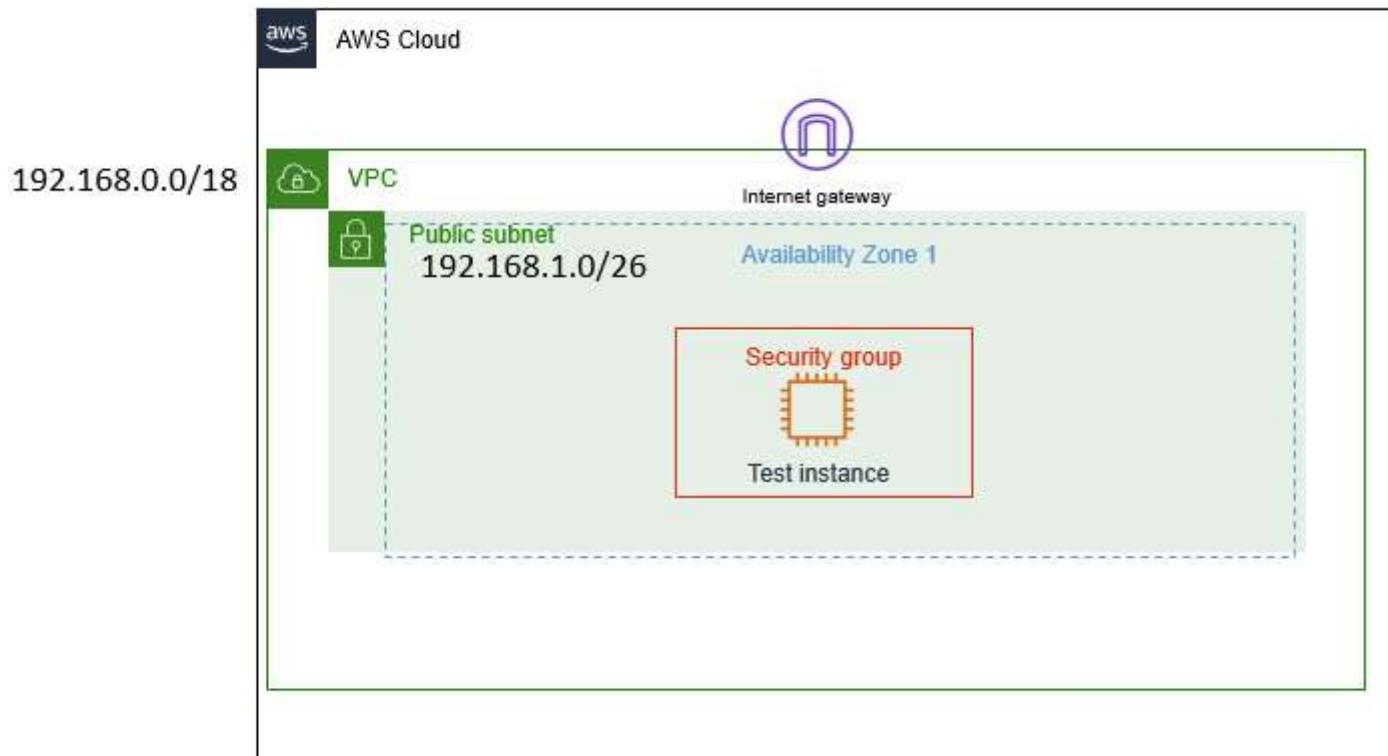
Your role is a Cloud Support Engineer at Amazon Web Services (AWS). During your shift, a customer from a startup company requests assistance regarding a networking issue within their AWS infrastructure. The email and an attachment of their architecture is below.

Email from the customer

Hello Cloud Support!

I previously reached out to you regarding help setting up my VPC. I thought I knew how to attach all the resources to make an internet connection, but I cannot even ping outside the VPC. All I need to do is ping! Can you please help me set up my VPC to where it has network connectivity and can ping? The architecture is below. Thanks!

Brock, startup owner



Customer VPC architecture

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab.

A **Start Lab** panel opens, and it displays the lab status.

Tip: If you need more time to complete the lab, choose the Start Lab button again to restart the timer for the environment.

2. Wait until you see the message *Lab status: ready*, then close the **Start Lab** panel by choosing the **X**.

3. At the top of these instructions, choose **AWS**.

This opens the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with a message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and then choose **Allow pop ups**.

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps more easily.

Task 1: Investigate the customer's needs

► Recall

For task 1, you will investigate the customer's request and build a VPC that has network connectivity. You will complete this lab when you can successfully ping from your EC2 instance to the internet showing that the VPC has network connectivity.

In the scenario, Brock, the customer requesting assistance, has requested help in creating resources for his VPC to be routable to the internet. Keep the VPC CIDR at 192.168.0.0/18 and public subnet CIDR of 192.168.1.0/26.

VPC Dashboard

EC2 Global View [New](#)

Filter by VPC:

 Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

Carrier Gateways

DHCP Options Sets

Elastic IPs

Managed Prefix Lists

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

SECURITY

Network ACLs

Security Groups

Figure: A great guide to building a VPC is to follow the left hand navigation pane, starting from "Your VPCs" and working your way down.

Before you start, let's review VPC and its components to make it network compatible.

- A **Virtual Private Cloud (VPC)** is like a data center but in the cloud. Its logically isolated from other virtual networks from which you can spin up and launch your AWS resources within minutes.

- **Private Internet Protocol (IP)** addresses are how resources within the VPC communicate with each other. An instance needs a public IP address for it to communicate outside the VPC. The VPC will need networking resources such as an Internet Gateway (IGW) and a route table in order for the instance to reach the internet.
- An **Internet Gateway (IGW)** is what makes it possible for the VPC to have internet connectivity. It has two jobs: perform network address translation (NAT) and be the target to route traffic to the internet for the VPC. An IGW's route on a route table is always 0.0.0.0/0.
- A **subnet** is a range of IP addresses within your VPC.
- A **route table** contains routes for your subnet and directs traffic using the rules defined within the route table. You associate the route table to a subnet. If an IGW was on a route table, the destination would be 0.0.0.0/0 and the target would be IGW.
- **Security groups** and **Network Access Control Lists (NACLs)** work as the firewall within your VPC. Security groups work at the instance level and are stateful, which means they block everything by default. NACLs work at the subnet level and are stateless, which means they do not block everything by default.

Steps

5. Select the **AWS** button located in the top right of the Vocareum home environment. This will open the AWS console in a new tab.
6. Once in the AWS console, click **VPC** under **Recently visited services**. If it is not there, navigate to the top left corner, and select **VPC** under **Networking and Content Delivery** in the **Services** navigation pane.

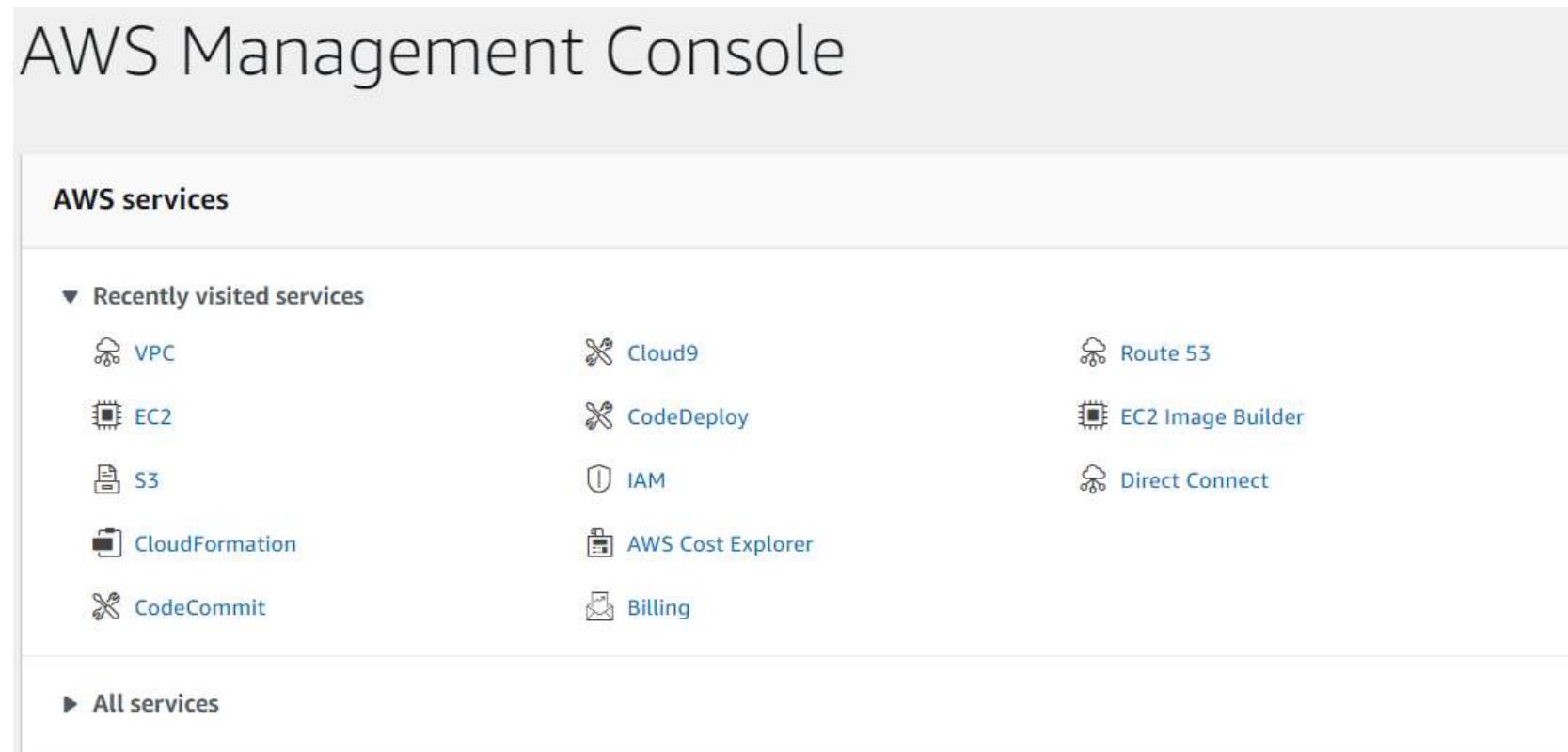


Figure: Recently visited services in the AWS console

The screenshot shows the AWS Services navigation drop-down menu. At the top left is the AWS logo and a 'Services ▾' button. To its right is a search bar with placeholder text 'Search for services, features, marketplace products, and docs' and a keyboard shortcut '[Alt+S]'. The main area is divided into two columns: 'Favorites' on the left and 'All services' on the right.

Favorites

- Resource Groups & Tag Editor

Recently visited

- Console Home
- VPC
- EC2
- S3
- CloudFormation
- CodeCommit
- Cloud9
- CodeDeploy
- IAM
- AWS Cost Explorer
- Billing
- Route 53
- EC2 Image Builder
- Direct Connect

All services

The 'All services' section lists various AWS services under several categories:

- Server Migration Service**
- AWS Transfer Family**
- AWS Snow Family**
- DataSync**
- Networking & Content Delivery**
- VPC
- CloudFront
- Route 53
- API Gateway
- Direct Connect
- AWS App Mesh
- AWS Cloud Map
- Global Accelerator
- Resource Groups & Tag Editor**
- Amazon Grafana
- Amazon Prometheus
- AWS Proton
- Incident Manager
- Media Services**
- Kinesis Video Streams
- MediaConnect
- MediaConvert
- MediaLive
- MediaPackage
- MediaStore
- MediaTailor
- Elemental Appliances & Software
- Amazon Interactive Video Service
- Elastic Transcoder
- Nimble Studio

Figure: Services navigation drop down

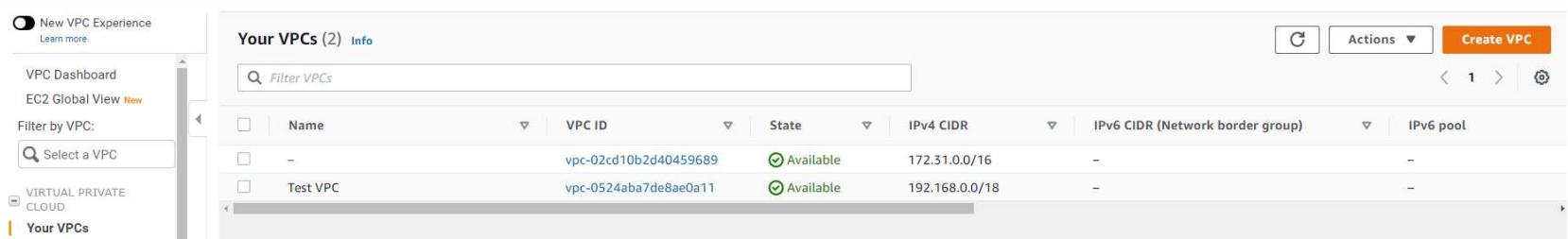
Creating the VPC

► Recall

7. Start at the top of the left navigation pane at **Your VPCs** and work your way down. Select **Your VPCs**, navigate to the top right corner, and select **Create VPC**.

Note

Note, you will be using a top-down theory with the top being the VPC.



Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)	IPv6 pool
-	vpc-02cd10b2d40459689	Available	172.31.0.0/16	-	-
Test VPC	vpc-0524aba7de8ae0a11	Available	192.168.0.0/18	-	-

Figure: Navigate to "Your VPCs" and select Create VPC.

8. Name the VPC: **Test VPC**

IPv4 CIDR block: **192.168.0.0/18**

9. Leave everything else as default, and select **Create VPC**.

VPC Successfully Created

Your VPC has been successfully created.

You can launch instances into the subnets of your VPC. For more information, see [Launching an Instance into Your Subnet](#).

Figure: VPC settings configuration

Creating Subnets

► Recall

10. Now that the VPC is complete, look at the left navigation pane and select **Subnets**. In the top right corner, select **Create subnet**.

Note

Please note: Although almost anything can be created in any order, it is easier to have an approach. Having a flow or an approach will assist you in troubleshooting issues and ensure that you do not forget a resource.

Subnets (4) Info										
	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 a			
<input type="checkbox"/>	-	subnet-038ed9e7b38319620	Available	vpc-02cd10b2d40459689	172.31.16.0/20	-	4091			
<input type="checkbox"/>	-	subnet-0c364b868f7ef6f7c	Available	vpc-02cd10b2d40459689	172.31.48.0/20	-	4091			
<input type="checkbox"/>	-	subnet-03a499f278b913925	Available	vpc-02cd10b2d40459689	172.31.0.0/20	-	4091			
<input type="checkbox"/>	-	subnet-0c8cc63299097dec1	Available	vpc-02cd10b2d40459689	172.31.32.0/20	-	4091			

Figure: Select Create subnet

11. Configure like the following picture:

VPC

VPC ID

Create subnets in this VPC.

▼
vpc-0524aba7de8ae0a11 (Test VPC)

Associated VPC CIDRs

IPv4 CIDRs

192.168.0.0/18

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.

Public subnet

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.

No preference

IPv4 CIDR block [Info](#)

Q 192.168.1.0/28

X

▼ Tags - optional

Key

Value - optional



Figure: Subnet configuration

Create Route Table

► Recall

12. Navigate to the left navigation pane, and select **Route Tables**. In the top right corner select **Create route table**.

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
-	rtb-0f4c9ea27f2b50085	-	-	Yes	vpc-0524aba7de8ae0a11 Test VPC	300476415442
-	rtb-002a7022b83f1641c	-	-	Yes	vpc-02cd10b2d40459689	300476415442

Figure: Select Create route table.

13. Configure like the following picture:

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.

Public route table

VPC

The VPC to use for this route table.

vpc-0524aba7de8ae0a11 (Test 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 - optional

Public route table



Remove

Add new tag

You can add 49 more tags.

Cancel

Create route table

Figure: Route table configuration

Create Internet Gateway and attach Internet Gateway

► In this lab

14. From the left navigation pane, select **Internet Gateways**. Create an Internet Gateway (IGW) by selecting **Create internet gateway** at the top right corner.

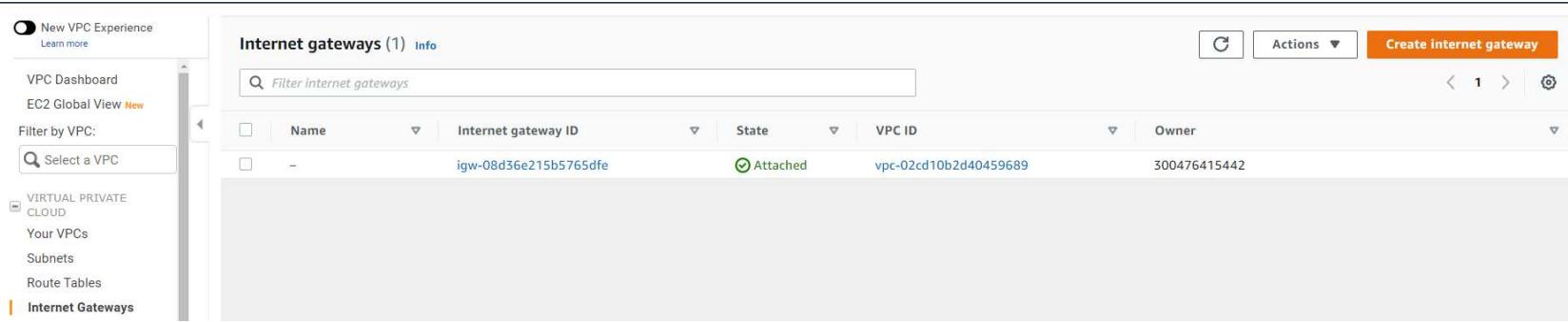


Figure: Select *Create internet gateway*

15. Configure like the following picture:

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.

IGW test VPC

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

Name

Value - optional

IGW test VPC



Remove

Add new tag

You can add 49 more tags.

Cancel

Create internet gateway

Figure: Internet gateway configuration

- Once created, attach the **Internet Gateway** to the VPC by selecting **Actions** at the top right corner and clicking **Attach to VPC**.

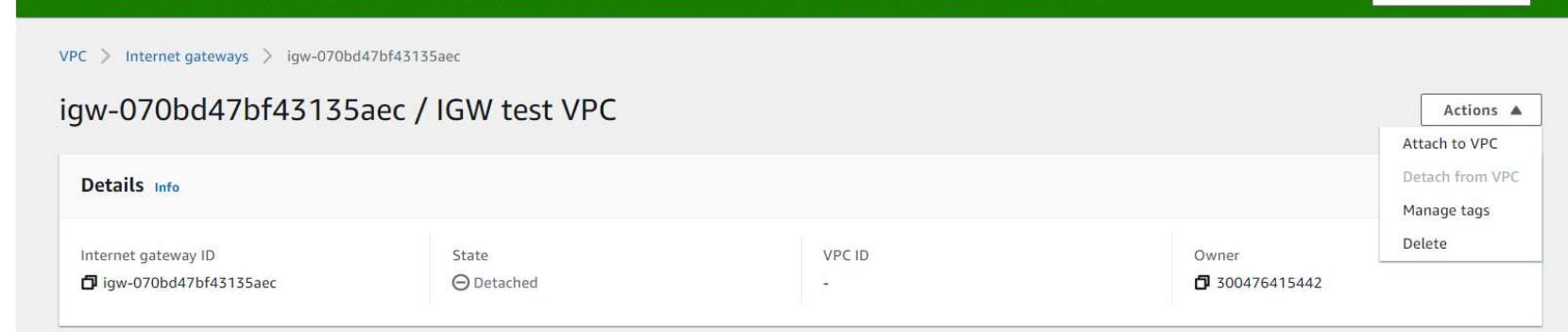


Figure: Attaching the IGW that was just created.

Now your IGW is attached! You now need to add its route to the route table and associate the subnet you created to the route table.

Add route to route table and associate subnet to route table

17. Navigate to the **Route Table** section on the left navigation pane. Select **Public Route Table**, and scroll to the bottom and select the **Routes** tab. Select the Edit routes button located in the routes box.

On the Edit routes page, the first IP address is the local route and cannot be changed.

Select **Add route**.

- In the **Destination** section, type **0.0.0.0/0** in the search box. This is the route to the IGW. You are telling the route table that any traffic that needs internet connection will use 0.0.0.0/0 to reach the IGW so that it can reach the internet.
- Click in the **Target** section and select **Internet Gateway** since you are targeting any traffic that needs to go to the internet to the IGW. Once you select the IGW, you will see your **TEST VPC IGW** appear. Select that IGW, navigate to the bottom right, and select **Save changes**.

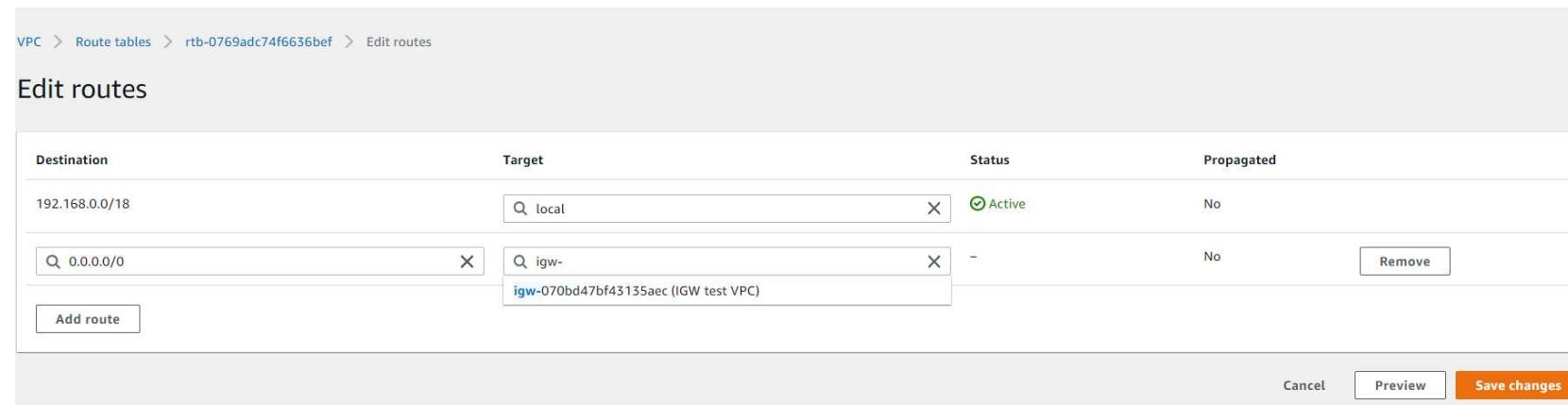


Figure: Adding the IGW in the route table (0.0.0.0/0 as the destination and IGW as the target).

Now your traffic has a route to the internet via the IGW.

18. From the Public route table dashboard, select the **Subnet associations** tab. Select the **Edit subnet associations** button.

The screenshot shows the 'Edit subnet associations' interface. At the top, there's a breadcrumb navigation: VPC > Route tables > rtb-0769adc74f6636bef > Edit subnet associations. Below the header, it says 'Change which subnets are associated with this route table.' The main area has two sections: 'Available subnets (1/1)' and 'Selected subnets'. The 'Available subnets' table has columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route table ID. One row is shown: 'Public subnet' with Subnet ID 'subnet-09a3b15dff7bdb6e5', IPv4 CIDR '192.168.1.0/28', and Route table ID 'Main (rtb-0f4c9ea27f2b30085)'. The 'Selected subnets' section contains a single item: 'subnet-09a3b15dff7bdb6e5 / Public subnet'. At the bottom right are 'Cancel' and 'Save associations' buttons.

Figure: Associate the Public subnet and select save association.

19. Select **Save association**.

Note: Every route table needs to be associated to a subnet. You are now associating this route table to this subnet. As you probably noticed, the naming convention is kept the same (public route table, public subnet, etc) in order to associate the same resources together. Keep this in mind when your network and resources grow. You can have multiples of the same resources and it can get confusing to which belongs where.

Creating a Network ACL

► Recall

20. From the left navigation pane, select **Network ACLs**. Navigate to the top right corner and select **Create network ACL** to create a Network Access Control Lists (NACLs).

This picture shows the Network ACLs home page when "Network ACLs" is selected. This is where you will navigate to the top right button and select Create network ACL.

Figure: Select Create network ACL

21. On the **Create network ACL**, configure the following:

- **Name:** `Public Subnet NACL`
- **VPC:** Choose `Test VPC` from dropdown
- Choose **Create network ACL**

22. On the **Network ACLs** option, from the list of ACLs select **Public Subnet ACL**

23. From the tabs below, select **Inbound rules** and then choose **Edit inbound rules**

24. On the **Edit inbound rules**, choose **Add new rule** and configure:

- Rule number: Enter **100**
- Type: Choose **All traffic** from dropdown

25. Choose **Save changes**

26. Back on the **Network ACLs** option, ensure that **Public Subnet ACL** is selected

27. Choose **Outbound rules** and then choose ***Edit outbound rules**

28. On the **Edit outbound rules**, choose **Add new rule** and configure:

- Rule number: Enter **100**
- Type: Choose **All traffic** from dropdown

29. Choose **Save changes**

Inbound After creating the NACL, it will look like the following. This indicates there is only one rule number, which is 100, that states that all traffic, all protocols, all port ranges, from any source (0.0.0.0/0) are allowed to enter (inbound) the subnet. The asterisk * indicates that anything else that does not match this rule is denied.

The screenshot shows the AWS CloudFormation console interface for managing Network ACLs. At the top, a navigation bar includes 'Public Subnet NACL' (selected), 'acl-0c75d86131fc2b175', 'subnet-09a3b15dff7bdb6e5 / Public subnet', 'Yes', 'vpc-0524aba7de8ae0a11 / Test VPC', and '2 Inbound rules'. Below this, a main content area displays the 'acl-0c75d86131fc2b175 / Public Subnet NACL' details. A navigation tab bar at the top of the content area includes 'Details' (disabled), 'Inbound rules' (selected), 'Outbound rules', 'Subnet associations', and 'Tags'. A message box at the top right says 'You can now check network connectivity with Reachability Analyzer' with buttons for 'Run Reachability Analyzer' and 'X'. The 'Inbound rules (2)' section contains a table with two rows. The first row has a 'Rule number' of '100', 'Type' of 'All traffic', 'Protocol' of 'All', 'Port range' of 'All', 'Source' of '0.0.0.0/0', and an 'Allow/Deny' status of 'Allow' with a green checkmark. The second row has a 'Rule number' of '*', 'Type' of 'All traffic', 'Protocol' of 'All', 'Port range' of 'All', 'Source' of '0.0.0.0/0', and an 'Allow/Deny' status of 'Deny' with a red X. There are also buttons for 'Edit inbound rules', navigation arrows, and a settings gear icon.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input type="checkbox"/> Deny

Figure: Default inbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Outbound What do you think this rule says?

The screenshot shows the AWS CloudFormation console interface for managing Network ACLs (NACLs). The top navigation bar includes tabs for 'Details', 'Inbound rules', 'Outbound rules' (which is selected), 'Subnet associations', and 'Tags'. A status message at the top states, 'You can now check network connectivity with Reachability Analyzer' and includes a 'Run Reachability Analyzer' button. Below this, a section titled 'Outbound rules (2)' displays two entries in a table:

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny

Figure: Default outbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Creating a Security Group

► Recall

Recall that a security group is a virtual firewall at the instance level that controls inbound and outbound traffic. Just like a NACL, security groups control traffic; however, security groups cannot deny traffic. Security groups are stateful; you must allow traffic through the security group as it blocks everything by default, and it must be associated to an instance. A security group has the following parts for both inbound and outbound rules:

- Inbound Source: It can be an IP or a specific resource
- Outbound Destination: Can be an IP such as anywhere (0.0.0.0/0)
- Protocol: Example UDP or TCP
- Port range: All or specific range
- Description: You can input a description

30. From the left navigation pane, select **Security Groups**. Navigate to the top right corner and select **Create security group** to create a security group.



Figure: Select Create security group

Configure like the following image of the Basic details page:

Note: In the VPC portion, remove the current VPC, and select **Test VPC**.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

X

Figure: Configure the Basic details page

The completed security group is shown below. This indicates that for **Inbound rules** you are allowing SSH, HTTP, and HTTPS types of traffic, each of which has its own protocols and port range. The source from which this traffic reaches your instance can be originating from anywhere. For **Outbound rules**, you are allowing all traffic from outside your instance.

The screenshot shows the AWS Management Console interface for managing security groups. It is divided into two main sections: 'Inbound rules' and 'Outbound rules'.

Inbound rules:

- Type:** SSH, **Protocol:** TCP, **Port range:** 22, **Source:** Anywhere..., **Description - optional:** (empty), **Delete:** (button)
- Type:** HTTP, **Protocol:** TCP, **Port range:** 80, **Source:** Anywhere..., **Description - optional:** (empty), **Delete:** (button)
- Type:** HTTPS, **Protocol:** TCP, **Port range:** 443, **Source:** Anywhere..., **Description - optional:** (empty), **Delete:** (button)

Add rule button

Outbound rules:

- Type:** All traffic, **Protocol:** All, **Port range:** All, **Destination:** Custom, **Source:** 0.0.0.0/0, **Description - optional:** (empty), **Delete:** (button)

Add rule button

Figure: Configuration details for inbound and outbound rules for the security group

You now have a functional VPC. The next task is to launch an EC2 instance to ensure that everything works.

Task 2: Launch EC2 instance and SSH into instance

In task 2, you will launch an EC2 instance within your Public subnet and test connectivity by running the command **ping**. This will validate that your infrastructure is correct, such as security groups and network ACLs, to ensure that they are not blocking any traffic from your instance to the internet and vice versa. This will validate that you have a route to the IGW via the route table and that the IGW is attached.

31. On the AWS Management Console, in the **Search** bar, enter and choose **EC2** to go to the **EC2 Management Console**.
32. In the left navigation pane, choose **Instances**.
33. Choose **Launch instances** and configure the following options:
 - In the **Name and tags** section, leave the Name blank.
 - In the **Application and OS Images (Amazon Machine Image)** section, configure the following options:
 - **Quick Start:** Choose **Amazon Linux**.
 - **Amazon Machine Image (AMI):** Choose **Amazon Linux 2023 AMI**.
 - In the **Instance type** section, choose **t3.micro**.
 - In the **Key pair (login)** section, choose **vockey**.
34. In the **Network settings** section, choose **Edit** and configure the following options:

- **VPC - required:** Choose **Test VPC**.
- **Subnet:** Choose **Public Subnet**.
- **Auto-assign public IP:** Choose **Enable**.
- **Firewall (security groups):** Choose **Select existing security group**.
 - Choose **public security group**.

35. Choose **Launch instance**.

36. To display the launched instance, choose **View all instances**.

► The EC2 instance named **Bastion Server** is initially in a *Pending* state. The **Instance state** then changes to **Running** to indicate that the instance has finished booting.

Use SSH to connect to an Amazon Linux EC2 instance

► Ways to connect Amazon Linux EC2

The following instructions vary slightly depending on whether you are using Windows or Mac/Linux.

Windows Users: Using SSH to Connect

► These instructions are specifically for Windows users. If you are using macOS or Linux, [skip to the next section](#).

37. Select the **Details** drop-down menu above these instructions you are currently reading, and then select **Show**. A Credentials window will be presented.

38. Select the **Download PPK** button and save the **labsuser.ppk** file.

Typically your browser will save it to the Downloads directory.

39. Make a note of the **PublicIP** address.

40. Then exit the Details panel by selecting the **X**.

41. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, [download it here](#).

42. Open **putty.exe**

43. Configure your PuTTY session by following the directions in the following link: [Connect to your Linux instance using PuTTY](#)

44. Windows Users: [Select here to skip ahead to the next task](#).

macOS and Linux Users

These instructions are specifically for Mac/Linux users. If you are a Windows user, [skip ahead to the next task.](#)

45. Select the **Details** drop-down menu above these instructions you are currently reading, and then select **Show**. A Credentials window will be presented.
46. Select the **Download PEM** button and save the **labsuser.pem** file.
47. Make a note of **PublicIP**, the IPV4 server's address you have to connect to.
48. Then exit the Details panel by selecting the **X**.
49. Open a terminal window, and change directory `cd` to the directory where the `labsuser.pem` file was downloaded. For example, if the `labuser.pem` file was saved to your Downloads directory, run this command:

```
cd ~/Downloads
```

50. Change the permissions on the key to be read-only, by running this command:

```
chmod 400 labsuser.pem
```

51. Run the below command (*replace <public-ip> with the server's address you copied earlier*):

```
ssh -i labsuser.pem ec2-user@<public-ip>
```



A dark-themed terminal window is shown against a light gray background. The window has three colored circular icons (red, yellow, green) in the top-left corner. The main area contains the following text:
hostname ~]\$ ssh -i /path/my-key-pair.pem ec2-user@35.167.247.163

52. Type `yes` when prompted to allow the first connection to this remote SSH server.

Because you are using a key pair for authentication, you will not be prompted for a password.

Task 3: Use ping to test internet connectivity

53. Run the following command to test internet connectivity:

```
ping google.com
```

After a few seconds, exit ping by holding **CTRL+C** on Windows or **CMD+C** on Mac to exit. You should get the following result:

Successful ping:

```
[ec2-user@ip-192-168-1-8 ~]$ ping google.com
PING google.com (142.250.217.110) 56(84) bytes of data.
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=1 ttl=93 time=6.02 ms
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=2 ttl=93 time=5.96 ms
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=3 ttl=93 time=6.23 ms
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=4 ttl=93 time=6.01 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 5.969/6.060/6.230/0.126 ms
[ec2-user@ip-192-168-1-8 ~]$
```

Run ping to test connectivity. The above results are saying you have replies from google.com and have 0% packet loss.

If you are getting replies back, that means that you have connectivity.

Lab Complete

■ Congratulations! You have completed the lab.

54. Choose ■ **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel indicates that *You may close this message box now. Lab resources are terminating...*

55. Choose the X in the upper-right corner to close the **End Lab** panel.

Recap

► In this lab

Additional Resources

[What is Amazon VPC?](#)

[IP Addressing in your VPC](#)

[Route tables for your VPC](#)

[Internet Gateways](#)

[Network ACLs](#)

[Security Groups](#)

For more information about AWS Training and Certification, see [AWS Training and Certification Opens in new window](#)

Your feedback is welcome and appreciated.

If you would like to share any suggestions or corrections, please provide the details in our [AWS Training and Certification Contact Form Opens in new window](#)

© 2022 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.