Build, Secure and Monitor Networks on AWS

**SPL-TF-100-NWNIDL-1 - Version 1.0.5**

© 2023 Amazon Web Services, Inc. or 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. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab Overview**

In this lab, you will be helping AnyCompany to build, monitor, and secure their new AWS network environment. The environment they have asked you to create contains three VPCs (A, B, and C) which are peered. Building the three peered VPCs demonstrates how configuration and routing complexity grows as you add more VPCs and associate gateways and resources. You will help AnyCompany to re-configure the network into a hub-spoke model, using AWS Transit Gateway. You will also help AnyCompany to configure network monitoring, and security controls to protect the organization’s resources.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Setup routing between VPCs.
* Select the appropriate connectivity options for an environment.
* Capture network traffic information (metadata) with VPC flow logs.
* Configure monitoring for networking statistics and metrics.
* Filter network traffic with Network Access Lists (NACLs), Security Groups (SG).

TECHNICAL KNOWLEDGE PREREQUISITES

Learners should have knowledge about basic networking concepts (such as IP Addressing, CIDR notation, and routing), an understanding with basic cloud operations, and familiarity with navigating in the AWS Management Console.

PREREQUISITES

This lab requires:

* Access to a computer with Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat).
* A modern internet browser such as Chrome or Firefox.

DURATION

This lab requires approximately 120 minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to certain aspects of the guide. The following list explains the purpose for each one:

* **Command:** A command that you must run.
* **Additional information:** Information of special interest or importance.
* **Note:** The note icon specifies important hints, tips, guidance, or advice.
* **Caution:** Calls attention to information of special interest or importance. Failure to read the note does not result in breaking the service or losing any data, but could result in the need to repeat certain steps.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Conclusion:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

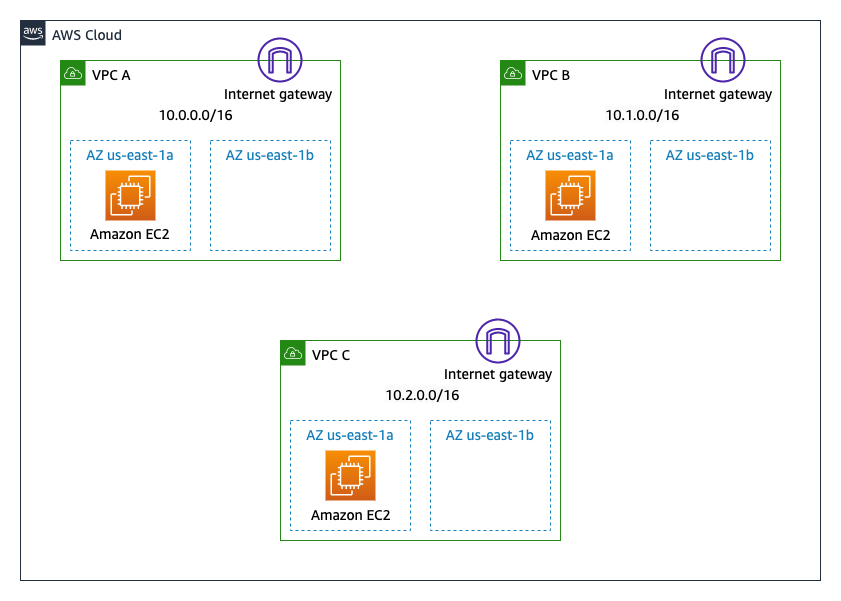
* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Build a Multi-VPC Architecture**

AnyCompany needs you to provision three logically isolated sections of the AWS Cloud into VPCs. In addition to using multiple VPCs, you will span the environment across multiple Availability zones (AZ) within a Region. After creating these VPCs, you will launch Amazon Elastic Compute Cloud (Amazon EC2) instances into the virtual networks that you define.

Each of the VPCs you create require an internet gateway (IGW). This is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

You will finish this task by observing that by default, EC2 instances in different VPCs are not able to communicate with each other using private IP addresses.



TASK 1.1: CREATE THREE VPCS WITH PRIVATE SUBNETS

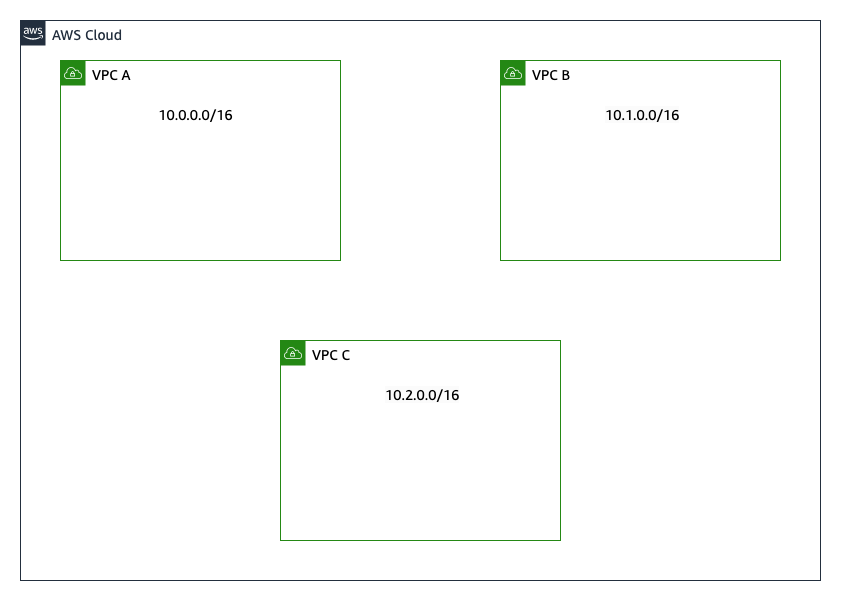
In this task, you will be creating three VPCs with private subnets. Each VPC will have subnets in two Availability Zones within the Region.

**Table 1.** Below table shows IPv4 CIDR allocations for VPCs and AZs

| **VPC Name** | **VPC CIDR Block** | **Availability Zone** | **Availability Zone CIDR Block** |
| --- | --- | --- | --- |
| VPC A | 10.0.0.0/16 | us-west-2a | 10.0.0.0/24 |
|  |  | us-west-2b | 10.0.1.0/24 |
| VPC B | 10.1.0.0/16 | us-west-2a | 10.1.0.0/24 |
|  |  | us-west-2b | 10.1.1.0/24 |
| VPC C | 10.2.0.0/16 | us-west-2a | 10.2.0.0/24 |
|  |  | us-west-2b | 10.2.1.0/24 |

**Task 1.1.1: Create VPCs**

Our first step is to create VPCs with non-overlapping CIDR blocks.



1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Your VPCs**.

 In every region, a default VPC with CIDR 172.31.0.0/16 has already been created for you. So, even if you haven’t created anything in your account yet, you will see some pre-existing VPC resources already there.

1. Choose **Create VPC** and configure the following:

* **Resources to create:** Select  VPC only
* **Name tag:**

VPC A

* **IPv4 CIDR block:** Select  IPv4 CIDR manual input
* **IPv4 CIDR:**

10.0.0.0/16

* **IPv6 CIDR block:** Select  No IPv6 CIDR block
* **Tenancy:** Select **Default**
* **Tags:** Accept proposed Tags

1. Choose **Create VPC** .
2. Follow the same steps to create **VPC B** with

10.1.0.0/16

 IPv4 CIDR and **VPC C** with

10.2.0.0/16

 IPv4 CIDR; refer to [Table 1.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%201.) for the details.

 After completing these steps, you should have three new VPCs and default listed under **Your VPCs**.

**Task 1.1.2: Create Subnets**

For each VPC, we will create two subnets - one per availability zone (AZs name could differ - depending on your chosen region).



1. On the left navigation pane, under  **Virtual private cloud**, choose **Subnets**.
2. Choose **Create subnet** to create **Subnet 1 of 1** into AZ1 with name

VPC A - AZ1

 and configure the following:

* **VPC ID:** **VPC A**
* **Subnet name:** Create subnets with names that reflect VPC and AZ placement, such as

VPC A - AZ1

* **Availability Zone:** Select the **first** Availability Zone in the list. (Do **not** choose *No Preference*)
* **IPv4 CIDR block:**

10.0.0.0/24

* **Tags:** Accept proposed Tags

1. Choose **Add new subnet** to add **Subnet 2 of 2** into AZ2 with name

VPC A - AZ2

 and configure the following:

* **Subnet name:** Create subnets with names that reflect VPC and AZ placement, such as

VPC A - AZ2

* **Availability Zone:** Select the **second** Availability Zone in the list. (Do **not** choose *No Preference*)
* **IPv4 CIDR block:**

10.0.1.0/24

* **Tags:** Accept proposed Tags

1. Choose **Create subnet** .
2. Repeat the steps above to create subnets for **VPC B** and **VPC C**; refer to the below table for copying subnet name and subnet CIDR allocations.

| **VPC Name** | **VPC CIDR Block** | **Subnet Name** | **Availability Zone** | **Subnet CIDR Block** |
| --- | --- | --- | --- | --- |
| VPC A | 10.0.0.0/16 | VPC A - AZ1 | us-west-2a | 10.0.0.0/24 |
|  |  | VPC A - AZ2 | us-west-2b | 10.0.1.0/24 |
| VPC B | 10.1.0.0/16 | VPC B - AZ1 | us-west-2a | 10.1.0.0/24 |
|  |  | VPC B - AZ2 | us-west-2b | 10.1.1.0/24 |
| VPC C | 10.2.0.0/16 | VPC C - AZ1 | us-west-2a | 10.2.0.0/24 |
|  |  | VPC C - AZ2 | us-west-2b | 10.2.1.0/24 |

 After you finish the task, six new subnets should be available.

**Task 1.1.3: Deploy Internet Gateways**

In this section, you will deploy one Internet Gateway (IGW) per VPC. You need an Internet Gateway in order to establish outside connectivity to EC2 instances in VPCs.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Internet gateways**.
2. Choose **Create internet gateway** and configure the following:

* **Name tag:**

VPC A - IGW

* **Tags:** Accept proposed Tags

1. Choose **Create internet gateway** .
2. Select **Attach to a VPC** .
3. For **Available VPCs**, choose **VPC A**.
4. Choose **Attach internet gateway** .
5. Repeat the above steps to create and attach IGWs

VPC B - IGW

 to **VPC B** and

VPC C - IGW

 to **VPC C**.

 You should now have an IGW for the default VPC and three newly created IGWs available.

**Task 1.1.4: Update Routing Tables**

In order to utilize newly created Internet Gateways, you need to update VPC routing tables to point the default routes to these IGWs.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Route tables**.

 Please make sure that you ***DO NOT*** create a new Route table. Please follow the next steps to rename the existing Main Route table.

1. Select  any of the unnamed Route tables and navigate to the **Details** tab to check which VPC the Route table belongs to.
2. Assign the following names to the Route Tables by identifying what VPC a given Route Table belongs to:

* **VPC A:**

VPC A Route Table

* **VPC B:**

VPC B Route Table

* **VPC C:**

VPC C Route Table

1. Select  **VPC A Route Table**.
2. Choose the **Routes** tab in the lower half of the page.

 There is one route in your route table that allows traffic within the 10.0.0.0/16 network to flow within the network, but it does not route traffic outside of the network.

1. Choose **Edit routes** .
2. Choose **Add route** and then configure the following:

* **Destination:**

0.0.0.0/0

* **Target:** Choose **Internet Gateway** from the drop-down list and then choose the displayed internet gateway ID.

1. Choose **Save changes** .
2. Repeat the above steps to add the default route 0.0.0.0/0 to the Route tables

VPC B Route Table

 for **VPC B** and

VPC C Route Table

 for **VPC C**.

TASK 1.2: DEPLOY EC2 INSTANCES IN VPC’S

You will deploy one EC2 instance per VPC and demonstrate that, by default, VPCs provide network isolation. The EC2 instances should not be able to reach each other using ping (a common diagnostic tool) before the next task is completed.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances**.
2. Choose **Launch instances** and configure the following:

* **Name and tags:** Under Name, enter

EC2 VPC A - AZ1

* **Application and OS Images (Amazon Machine Image):**
  + **Quick Start:** Select **Amazon Linux**
  + **Amazon Machine Image (AMI):**

Amazon Linux 2 AMI (HVM), SSD Volume Type

* + **Architecture:** **64-bit (x86)**
* **Instance type:** **t2.micro**
* **Key pair (login):** **Proceed without a key pair (Not recommended)**

 You will use SSM session Manager to access the shell running on the EC2, so a key pair is not needed in the lab.

1. Scroll down to  **Network settings**, choose **Edit** and configure the following:

* **VPC - required:** **VPC A**
* **Subnet:** **VPC A - AZ1**
* **Auto-assign public IP:** **Enable**
* **Firewall (security groups):**
  + Choose **Create security group**
  + **Security group name - required:**

VPC A EC2 Security Group

* + **Description - required:**

Allow ICMP Traffic

* **Inbound security groups rules:**
  + **Type:** Select **All ICMP - IPv4**from the dropdown instead of **SSH**
  + **Source type:** **Custom**
  + **Source:** **10.0.0.0/8**

 It is not a best practice to have wide open Security groups that allow anyone/everyone such as 0.0.0.0/0, limit access to only what is required.

1. Scroll down to **Advanced details** and configure the following:

* **IAM instance profile:** From the drop-down list, select the instance profile with

Ec2SsmInstanceProfile

 in the name

1. Choose **Launch instance** .
2. Choose **View all instances** to display all the instances launched.
3. Occasionally choose the console refresh button  and wait for **Public Instance** to display the **Instance state** as  **Running** and wait for Status check to pass  **2/2 checks passed**.

 The Amazon EC2 instance named Public Instance is initially in a *Pending* state. The instance state then changes to  **Running** indicating that the instance has finished booting.

1. Launch 2 more EC2 instances and assign them names accordingly:

**i.** Second EC2 instance in **VPC B** with the following configuration

* **Name and tags:**

EC2 VPC B - AZ1

* **Network settings:**
  + **VPC - required:** **VPC B**
  + **Subnet:** **VPC B - AZ1**
* **Firewall (security groups):**
  + **Security group name - required:**

VPC B EC2 Security Group

* + **Description - required:**

Allow ICMP Traffic

 Except for the above details, keep all other configurations same as first EC2 instance.

**ii.** Third EC2 instance in **VPC C** with the following configuration

* **Name and tags:**

EC2 VPC C - AZ1

* **Network settings:**
  + **VPC - required:** **VPC C**
  + **Subnet:** **VPC C - AZ1**
* **Firewall (security groups):**
  + **Security group name - required:**

VPC C EC2 Security Group

* + **Description - required:**

Allow ICMP Traffic

 Except for the above details, keep all other configurations same as first EC2 instance.

 For all EC2 instances, update the Security Group rules under the Security tab to allow **ICMP** traffic. There is no need to have a rule for SSH since SSM Session Manager will be used to connect to the terminal.

 After few minutes, you should now have 3 EC2 instances in the  **Running** state.

1. Copy the  **Private IPv4 addresses** assigned to EC2 instances by choosing an Instance and navigating to the **Details** tab.
2. Copy the following table with IP information to your favorite notepad tool, and populate the private IP addresses that you copied in the previous step.

**Table 2.** Populate the following table with EC2 instances private IP Addresses:

| **VPC** | **EC2 instance Private IP Address** |
| --- | --- |
| VPC A | 10.0.0.[…] |
| VPC B | 10.1.0.[…] |
| VPC C | 10.2.0.[…] |

TASK 1.3: TEST INTER-VPC COMMUNICATION BETWEEN EC2 INSTANCES

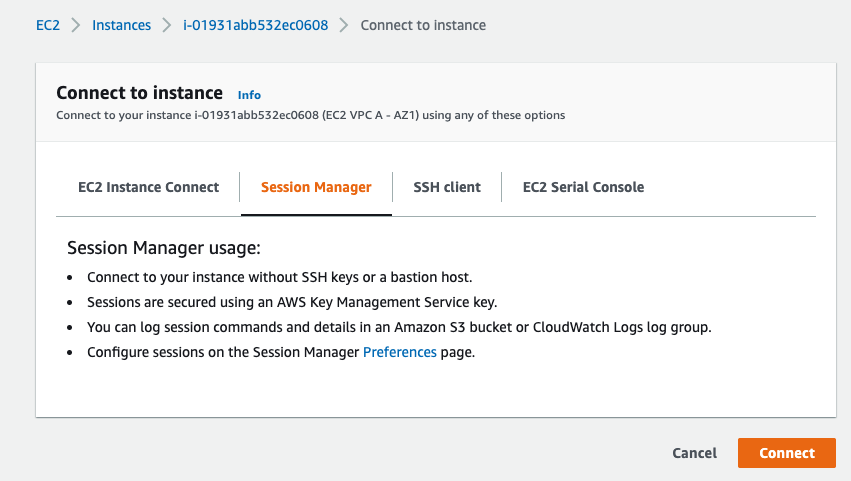
Now that the EC2 instances are available in each VPC, you will use ping (a common diagnostic tool) to verify that the instances cannot communicate demonstrating how VPCs provide isolation.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC A - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .



 Terminal session should open in a new browser tab.

1. From the

EC2 VPC A - AZ1

 instance in **VPC A**, try pinging the private IP addresses of

EC2 VPC B - AZ1

 instance in **VPC B** and

EC2 VPC C - AZ1

 instance in **VPC C**. Check if those addresses are pingable.

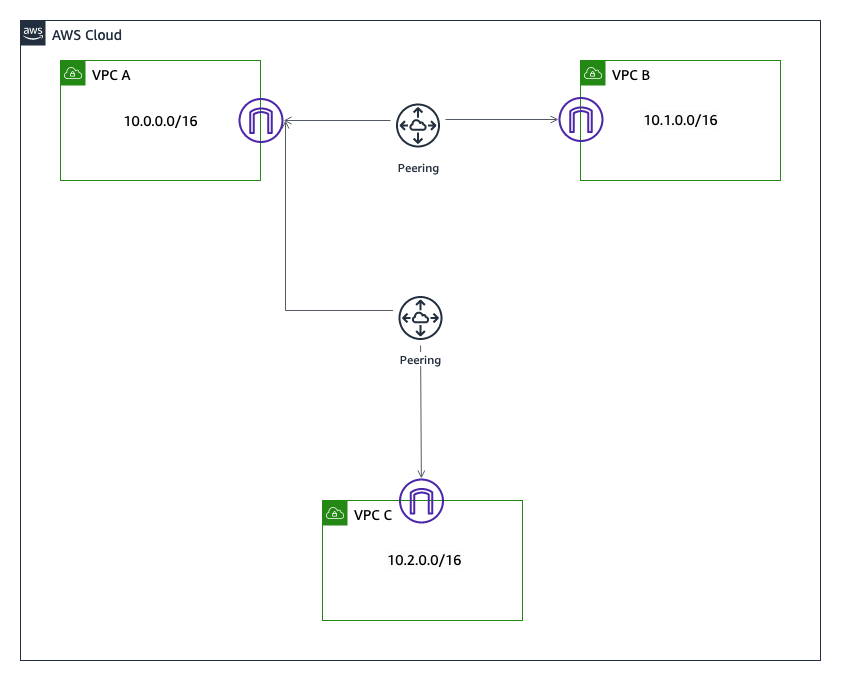
ping 10.1.0.x

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

TASK 1.4: SETUP VPC PEERING

The end of the previous task demonstrated that EC2 instances in different VPCs cannot reach each other on their private IP addresses. Anycompany wants to ensure that their AWS environment keeps all the traffic between VPCs on the AWS Backbone, and not traversing the internet. You will help them achieve this using VPC peering. VPC Peering is a connection between two VPCs that enables you to route your traffic between them.

In this task, you will establish VPC peering links between **VPC A** and **VPC B**, as well as **VPC A** and **VPC C**. Note that all three VPCs have non-overlapping CIDRS. You cannot create a VPC peering connection between VPCs with matching or overlapping IPv4 CIDR blocks.



1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Peering connections**.
2. Choose **Create peering connection** and configure the following:

* **Name**:

VPC A <-> VPC B

* **VPC ID (Requester):** **VPC A**
* **Account**:  My account
* **Region**:  This Region (xx-xxxx-x)

 Make sure the region labeled **AWSRegion** in the left side of the instructions matches with the region where you are creating the VPC Peering connection.

* **VPC ID (Accepter)**: **VPC B**
* **Tags:** Accept proposed Tags

1. Choose **Create peering connection** .

 The status transitions through *Initiating Request* to *Pending Acceptance*.

1. From the **Actions** button, choose **Accept request**.
2. Choose **Accept request** .
3. Repeat the above steps to create **VPC A <-> VPC C** peering connection with the following configuration:

* **Name**:

VPC A <-> VPC C

* **VPC ID (Requester):** **VPC A**
* **Account**:  My account
* **Region**:  This Region (xx-xxxx-x)
* **VPC ID (Accepter)**: **VPC C**
* **Tags:** Accept proposed Tags

 You should now have two active peering connections.

**Task 1.4.1: Update Route Tables for VPCs**

1. On the left navigation pane, under  **Virtual private cloud**, choose **Route Tables**.

**A. Update Route Table for VPC A**

1. Select  **VPC A Route Table**.
2. Choose **Routes** tab.
3. Choose **Edit routes** .
4. Choose **Add route** to add **VPC B** CIDR in **Destination** to enable *VPC A* to reach *VPC B* through the VPC peering connection:

* **Destination:**

10.1.0.0/16

* **Target:** Choose **Peering Connection** from drop-down list
* Select **pcx-xxxxxxxxxx (VPC A <-> VPC B)** from the drop-down list

1. Choose **Add route** to add **VPC C** CIDR in **Destination** to enable *VPC A* to reach *VPC C* through the VPC peering connection:

* **Destination:**

10.2.0.0/16

* **Target:** Choose **Peering Connection** from drop-down list
* Select **pcx-yyyyyyyyyy (VPC A <-> VPC C)** from the drop-down list

1. Choose **Save changes** .

**B. Update Route Tables for VPC B**

1. On the left navigation pane, under  **Virtual private cloud**, choose **Route tables**.
2. Select  **VPC B Route Table**.
3. Choose **Routes** tab.
4. Choose **Edit routes** .
5. Choose **Add route** to add **VPC A** CIDR in **Destination** to enable *VPC B* to reach *VPC A* through the VPC peering connection:

* **Destination:**

10.0.0.0/16

* **Target:** Choose **Peering Connection** from drop-down list
* Select **pcx-xxxxxxxxxx (VPC A <-> VPC B)** from the drop-down list

1. Choose **Save changes** .

**C. Update Route Table for VPC C**

1. On the left navigation pane, under  **Virtual private cloud**, choose **Route tables**.
2. Select  **VPC C Route Table**.
3. Choose **Routes** tab.
4. Choose **Edit routes** .
5. Choose **Add route** to add **VPC A** CIDR in **Destination** to enable *VPC C* to reach *VPC A* through the VPC peering connection:

* **Destination:**

10.0.0.0/16

* **Target:** Choose **Peering Connection** from drop-down list
* Select **pcx-yyyyyyyyyy (VPC A <-> VPC C)** from the drop-down list

1. Choose **Save changes** .

**D. Check EC2 connectivity in VPC A**

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC A - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC A - AZ1

 instance in **VPC A**, try pinging the private IP addresses of

EC2 VPC B - AZ1

 instance in **VPC B** and

EC2 VPC C - AZ1

 instance in **VPC C**.

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 If peering and routing are configured correctly, you should be able to ping both instances.

1. Select the  **EC2 VPC B - AZ1** instance.
2. Choose **Connect** from the navigation bar.
3. With Session Manager tab selected, choose **Connect** .
4. From the

EC2 VPC B - AZ1

 instance in **VPC B**, try pinging the private IP addresses of

EC2 VPC C - AZ1

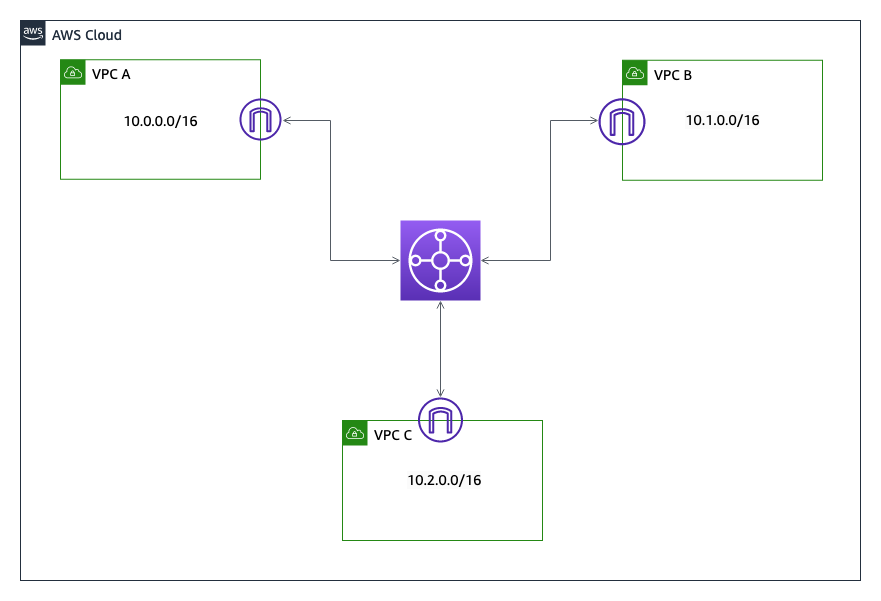
 instance in **VPC C**. Can you ping the EC2 instance in **VPC C**?

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

TASK 1.5: SIMPLIFY ROUTING USING AWS TRANSIT GATEWAY

In the previous section, you created VPC Peering Links in order to facilitate connectivity between the VPCs, without sending that traffic over the public internet. While this approach can be used to interconnect many VPCs, managing many point-to-point connections can be cumbersome as the number of VPCs you connect grows. A more scalable approach is to utilize AWS Transit Gateway.

In this task, you will remove point-to-point peering connections between **VPC A** and **VPC B** ; **VPC A** and **VPC C**. You will setup Transit Gateway (TGW) and use it to interconnect **VPC A**, **VPC B**, and **VPC C**.



**Task 1.5.1: Delete VPC Peering Connections**

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Peering connections**.
2. Select  **VPC A <-> VPC B** peering connection.
3. From the **Actions** button, choose **Delete peering connection**.
4. Select the option to  **Delete related route table entries** to avoid traffic blackholing scenario.
5. To confirm deletion, type *delete* in the field: **delete** .
6. Choose **Delete** .
7. Repeat the above steps to delete **VPC A <-> VPC C** peering connection.

**Task 1.5.2: Setup Transit Gateway**

**A. Create Transit Gateway**

1. On the left navigation pane, under  **Transit gateways**, choose **Transit gateways**.
2. Choose **Create transit gateway** and configure the following:

* **Name tag:**

TGW

* **Description:**

Lab Transit Gateway TGW

* **Amazon side Autonomous System Number (ASN):**

64512

 The Amazon side ASN or Multicast support cannot be changed after the transit gateway is created.

* **DNS support:**  enable
* **VPN ECMP support:**  enable
* **Default route table association:**  enable
* **Default route table propagation:**  enable
* **Multicast support:**  enable
* Keep the default settings for the rest of the parameters

1. Choose **Create transit gateway** .

 A few moments later, TGW will transition from *Pending* to  **Available** state.

**B. Create Transit Gateway attachments subnets**

 According to best practices it is recommended to use a separate small /28 subnet for each transit gateway VPC attachment.

**Table 3.** Below table shows IPv4 CIDR allocations for all Subnets in all the VPCs.

| **Subnet Name** | **Subnet CIDR block** | **Availability Zone** |
| --- | --- | --- |
| VPC A - AZ1 TGW | 10.0.2.0/28 | us-west-2a |
| VPC A - AZ2 TGW | 10.0.3.0/28 | us-west-2b |
| VPC B - AZ1 TGW | 10.1.2.0/28 | us-west-2a |
| VPC B - AZ2 TGW | 10.1.3.0/28 | us-west-2b |
| VPC C - AZ1 TGW | 10.2.2.0/28 | us-west-2a |
| VPC C - AZ2 TGW | 10.2.3.0/28 | us-west-2b |

 The availability zones in the above table is shown as an example. Please use the availability zones from the region where your lab is deployed.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Subnets**.
2. Choose **Create subnet** and configure the following:

* **VPC ID:** **VPC A**
* **Subnet name:** Create subnets with names that reflect VPC and AZ placement, such as

VPC A - AZ1 TGW

* **Availability Zone:** Select the **first** Availability Zone in the list. (Do **not** choose *No Preference*)
* **IPv4 CIDR block:**

10.0.2.0/28

* **Tags:** Accept proposed Tags

1. Choose **Add new subnet** and configure the following:

* **Subnet name:** Create subnets with names that reflect VPC and AZ placement, such as

VPC A - AZ2 TGW

* **Availability Zone:** Select the **second** Availability Zone in the list. (Do **not** choose *No Preference*)
* **IPv4 CIDR block:**

10.0.3.0/28

* **Tags:** Accept proposed Tags

1. Choose **Create subnet** .
2. Repeat the steps above to create subnets for **VPC B** and **VPC C**; refer to [Table 3.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%203.) above for CIDR allocations.

 After you finish the task, six new Transit Gateway attachments subnets should be available.

**C. Create Transit Gateway attachments**

1. On the left navigation pane, under  **Transit gateways**, choose **Transit gateway attachments**.
2. Choose **Create transit gateway attachment** and configure the following:

* **Name tag:**

VPC A Attachment

* **Transit Gateway ID:** **TGW**
* **Attachment type:** **VPC**
* **VPC ID:** **VPC A**
* **Subnet IDs:** Choose both subnets from the drop-down list:
  + VPC A - AZ1 TGW
  + VPC A - AZ2 TGW

1. Choose **Create transit gateway attachment** .
2. Repeat these steps to create attachments

VPC B Attachment

 for **VPC B** and

VPC C Attachment

 for **VPC C**.

 Upon completion, you should see three Transit Gateway attachments.

**Task 1.5.3: Check Transit Gateway route table**

1. On the left navigation pane, under  **Transit gateways**, choose **Transit gateway route tables**.
2. Select the route table you see with route table ID starting with  **tgw-rtb-xxxxxxxxxxxxxx**.
3. Choose the **Routes** tab in the lower half of the page.

 Your routing table should be populated with **VPC A**, **VPC B**, **VPC C** routes.

**Task 1.5.4: Update Route Tables of VPCs**

1. On the left navigation pane, under  **Virtual private cloud**, choose **Route tables**.
2. Select  **VPC A Route Table**.
3. Choose **Routes** tab.
4. Choose **Edit routes** .
5. Choose **Add route** and configure the following:

* **Destination:** Enter

10.0.0.0/8

* **Target:** Choose **Transit Gateway** from the drop-down list and then choose the displayed Transit Gateway ID.

 You can do this because the existing local route for **VPC A** (10.0.0.0/16) is more specific and therefore, any traffic for 10.0.0.0/16 will traverse the more specific local route to the VPC. Anything other traffic for 10.0.0.0/8 will traverse the less specific route (10.0.0.0/8) to the transit gateway.

1. Choose **Save changes** .
2. Repeat these steps to create

10.0.0.0/8

 route pointing to the

TGW

 Transit Gateway in **VPC B Route Table** and **VPC C Route Table** Routing tables.

TASK 1.6: CHECK CONNECTIVITY BETWEEN VPCS USING THE TGW

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC A - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC A - AZ1

 instance in **VPC A**, try pinging the private IP addresses of

EC2 VPC B - AZ1

 instance in **VPC B** and

EC2 VPC C - AZ1

 instance in **VPC C**.

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 If Transit gateway and routing are configured correctly, you should be able to ping both instances.

1. Select the  **EC2 VPC B - AZ1** instance.
2. Choose **Connect** from the navigation bar.
3. With **Session Manager** tab selected, choose **Connect** .
4. From the

EC2 VPC B - AZ1

 instance in **VPC B**, try pinging the private IP addresses of

EC2 VPC A - AZ1

 instance in **VPC A** and

EC2 VPC C - AZ1

 instance in **VPC C**.

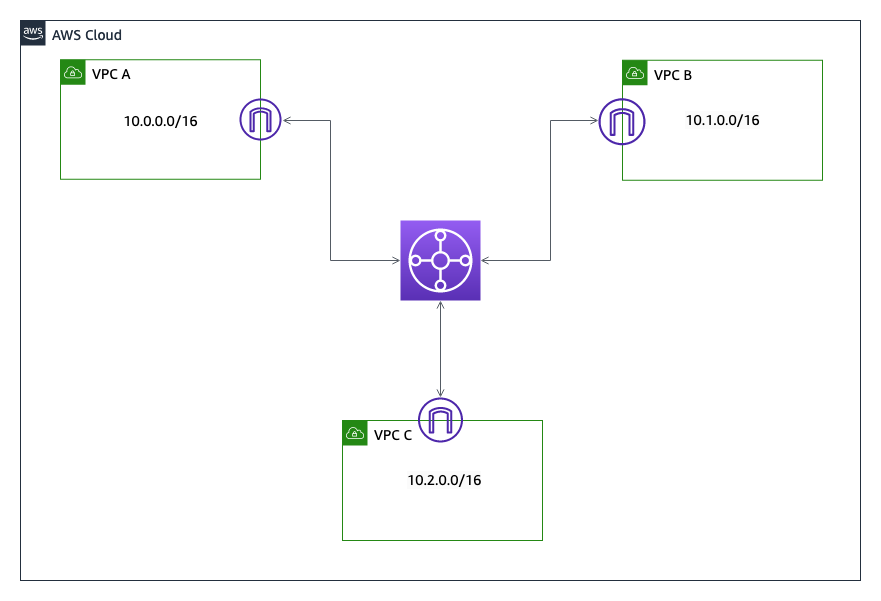
 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 If Transit gateway and routing are configured correctly, you should be able to ping both instances.

**Task complete:** You have successfully validated that EC2 instances in all three VPCs can reach each other using Transit Gateway.

**Task 2: Configure Network Monitoring**

In this task, you will be set up a way to log network traffic using VPC Flow logs. You will also use Amazon CloudWatch to monitor and alarm based on predetermined conditions. Finally, you will explore dashboards where you can customize your experience.



You will utilize the three VPC’s with Internet Gateways, Transit Gateway, and EC2 instances that were created in task 1. You will set up VPC Flow logs for VPC A, generate some traffic, and then view the logs in CloudWatch.

TASK 2.1: CREATE VPC FLOW LOGS

VPC Flow Logs is a feature that enables you to capture information (metadata) about the IP traffic going to and from network interfaces in your VPC. For example, if you have a content delivery platform, flow logs can profile, analyze, and predict customer patterns of the content access, and track down top talkers and malicious calls.

In this task you will create a flow log for all traffic in VPC A and save it to the destination log group **VPCFlowLog**.

**Task 2.1.1: Create CloudWatch Log group**

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**CloudWatch**

 and then choose **CloudWatch** from the list.

1. On the left navigation pane, under  **Logs**, choose **Log groups**.
2. Choose **Create log group** and configure the following:

* **Log group name:**

VPCFlowLog

* **Retention setting:** **Never expire**

1. Choose **Create** .

**Task 2.1.2: Create VPC Flow log**

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Your VPCs** and select  **VPC A**.
2. From **Actions** and choose **Create flow log**.
3. On the **Create flow log** page, under **Flow log settings**, configure the following:

* **Filter:**  All
* **Maximum aggregation interval:**  1 minute
* **Destination:**  Send to CloudWatch Logs
* **Destination log group:**

VPCFlowLog

* **IAM role:**

VPCFlowLogRole

* **Log record format:**  AWS default format

1. Choose **Create flow log** .

TASK 2.2: GENERATE NETWORK TRAFFIC BETWEEN VPCS

IP traffic going to and from network interfaces in VPC A is now being collected in through VPC Flow Logs and stored using a Amazon CloudWatch Log Group. You need to verify that your monitoring is setup properly. To accomplish this, you will generate some traffic between the Amazon EC2 instance in

VPC A

**(EC2 VPC A – AZ1)**, and instance in

VPC B

**(EC2 VPC B – AZ1)** using

iperf

 (a widely used tool for network performance measurement and tuning).

**Task 2.2.1: Update Security Group for EC2 instance in VPC B**

First, you will have to update the Security Group for the Amazon EC2 instance in **VPC B** to allow the iperf server to receive incoming traffic.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances** and select  **EC2 VPC B - AZ1**.
2. Navigate to the **Security** tab below and choose on the Security Group with ID starting with **sg- (VPC B EC2 Security Group)** in the lower pane to view and edit its rules.
3. On **VPC B EC2 Security Group** Security Group page, at the bottom of the page, choose the **Inbound rules** tab.
4. At the right side of the **Inbound rules** section, choose **Edit inbound rules** .
5. On the **Edit inbound rules** page, choose **Add rule** and configure the following:

* **Type:** **Custom TCP**
* **Port range:**

5201

* **Source type:** **Custom**
* **Source:**

10.0.0.0/16

 You added a rule to allow traffic on TCP port

5201

 from the Amazon EC2 instance in VPC A by allowing the VPC CIDR

10.0.0.0/16

 in source.

1. Choose **Save rules** .

**Task 2.2.2: Install and run iperf3 server on EC2 instance in VPC B**

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

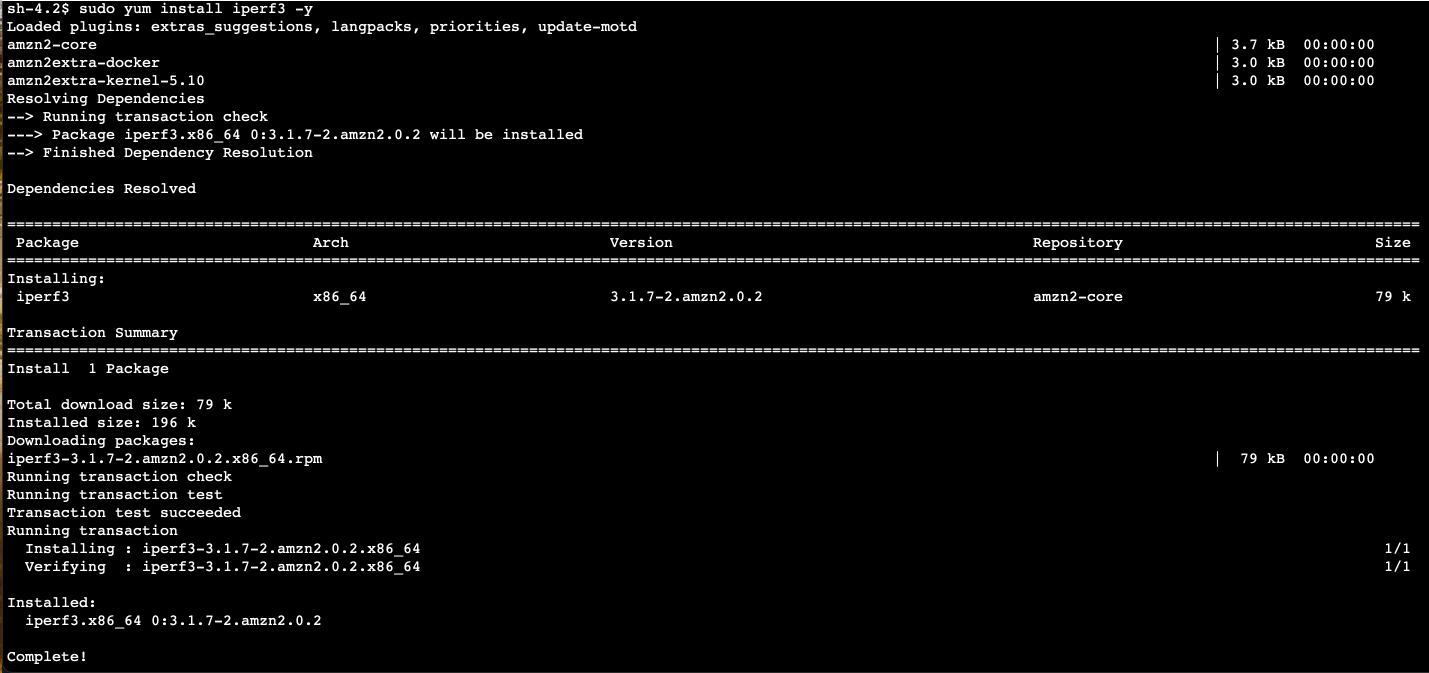
1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC B - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. Enter the following command to install iperf:

sudo yum install iperf3 -y

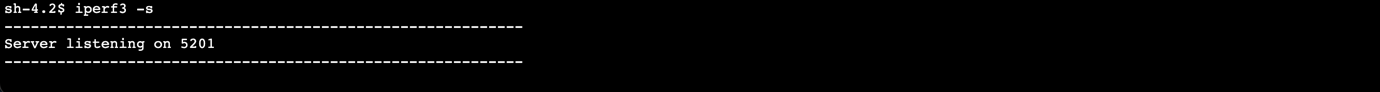
**Sample output:**



1. Enter the following command to start the iperf server on the EC2 instance in VPC B:

iperf3 -s

**Sample output:**



**Task 2.2.3: Perform the following on the EC2 instance in VPC A**

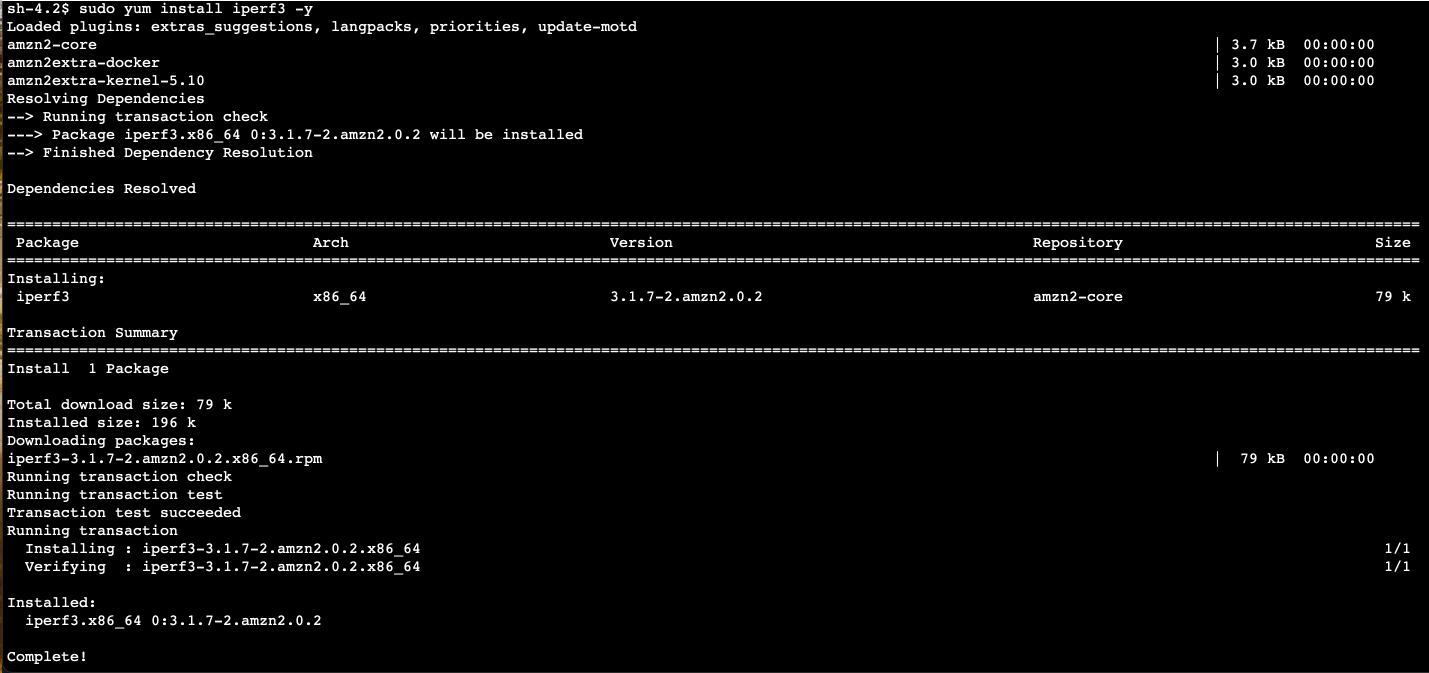
1. Navigate to the browser tab with **EC2 Instances**.
2. On the left navigation pane, choose **Instances**.
3. Select the  **EC2 VPC A - AZ1** instance.
4. Choose **Connect** from the navigation bar.
5. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. Enter the following command to install iperf:

sudo yum install iperf3 -y

**Sample output:**



1. Enter the following command to set up a TCP transfer with 2 parallel streams for 30 seconds to the EC2 instance in VPC B. Make sure to replace

<PRIVATE IP OF "EC2 VPC B - AZ1" INSTANCE 10.1.0.X>

 in the following command:

iperf3 -c <PRIVATE IP OF "EC2 VPC B - AZ1" INSTANCE 10.1.0.X> -P 2 -t 30

**Sample output:**

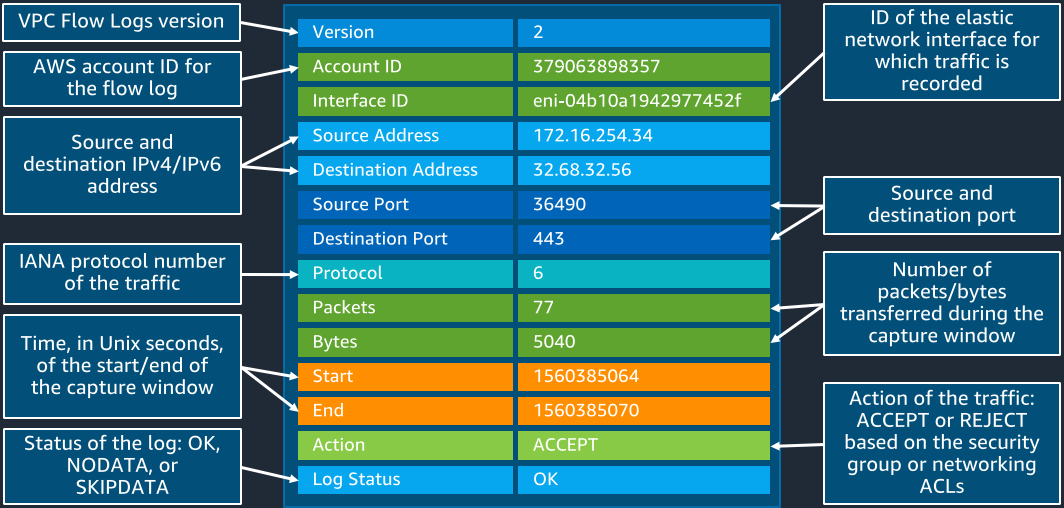


1. Shutdown the iperf3 server running on **EC2 VPC B - AZ1** instance in **VPC B** by switching back to Session Manager and by pressing *CTRL+C*.

TASK 2.3: VIEW FLOW LOGS IN CLOUDWATCH

VPC Flow logs can be sent to either an Amazon S3 bucket or CloudWatch. In this lab, you configured the flow logs from VPC A to be sent to CloudWatch. Navigate to CloudWatch to view the logs.

Anatomy of a flow log:



1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Virtual private cloud**, choose **Your VPCs** and select  **VPC A**.
2. In the **Details** pane at the bottom of the page, choose the **Flow logs** tab.
3. Under **Destination name**, choose VPCFlowLog  to navigate to CloudWatch.

 When publishing to CloudWatch, flow log data is published to a log group, and each network interface has a unique log stream in the log group. Log streams contain flow log records. You can create multiple flow logs that publish data to the same log group.

1. Select one of the network interface (ENI) log streams to see the flow records for that interface.

 To find appropriate log, make sure to use ENI from **EC2 VPC A - AZ1** (on EC2 Instance - Networking Tab scroll to see Networking interfaces list).

TASK 2.4: QUERY FLOW LOG FOR INSIGHTS

CloudWatch Logs Insights enables you to interactively search and analyze log data in CloudWatch Logs, including VPC flow logs. You can perform queries to help you more efficiently and effectively respond to operational issues.

Run a query to show the top 10 talkers based on bytes transferred.

1. Navigate to the browser tab with **CloudWatch** service.
2. On the left navigation pane, under  **Logs**, choose **Logs Insights**.
3. From the drop-down list, select  VPCFlowLog.
4. On the right navigation pane, choose  **Queries**.
5. Under **Sample queries**, expand  **VPC Flow Logs**.
6. From the list, choose  **Top 10 byte transfers by source and destination IP addresses**.
7. Choose Apply .
8. Choose **Run query** .

 Review the query results. Do you recognize the top two IP addresses?

*Hint: Take a look at the Primary private IPv4 address and Description columns in the*[*EC2 - Network Interfaces*](https://console.aws.amazon.com/ec2/v2/home?#NIC:)*and see what they are for.*

TASK 2.5: USING AMAZON CLOUDWATCH TO SET AN ALARM

Amazon CloudWatch is a metrics repository. Amazon VPC publishes data points to Amazon CloudWatch for your transit gateways and transit gateway attachments. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time series data, known as metrics. In this section, you will create a static route, set up an alarm on a threshold breach, and set up a dashboard to view a set of metrics on one page.

**Task 2.5.1: Create a static route**

You can create a static route for a VPC, VPN, or transit gateway peering attachment, or you can create a blackhole route. The blackhole state indicates that the route’s target isn’t available. For more details on static route and blackhole route, check [Additional Resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#resources).

In this section you will create a blackhole route in your transit gateway route table that drops traffic that matches the route. This task is necessary to view the CloudWatch Metrics in the later task.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Transit gateways**, choose **Transit gateway route tables**.
2. Select the route table you see with route table ID starting with  **tgw-rtb-xxxxxxxxxxxxxx**.
3. Choose the **Routes** tab in the lower half of the page.
4. Choose **Create static route** .
5. Under **Details**, configure the following:

* **CIDR:** Enter

10.3.0.0/16

* **Type:** Choose  Blackhole

1. Choose **Create static route** .

**Task 2.5.2: Create Alarm**

You can create a CloudWatch alarm that monitors CloudWatch metrics for a given AWS service. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify.

Create an alarm that monitors the number of packets dropped because they matched a blackhole route of a Transit gateway. If the packets are dropped, an email notification is sent to the configured email address.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**CloudWatch**

 and then choose **CloudWatch** from the list.

1. On the left navigation pane, under CloudWatch  **Alarms**, choose **All alarms**.
2. Select **Create alarm** .
3. Choose **Select metric** .
4. On **Select metric** page, under the **Metrics** section, select the **TransitGateway** card.
5. Select the **Per-TransitGateway Metrics** card.
6. Search for **PacketDropCountBlackhole** .
7. Select  **tgw-xxxxxxxxxxxxxx** Transit Gateway.
8. Choose **Select metric** .
9. On the **Metric** page, change the **Statistic** and **Period** parameters to following:

| **Key** | **Value** |
| --- | --- |
| Statistic | **Maximum** |
| Period | **1 Minute** |

1. On the **Conditions** section, configure the following:

* **Threshold type:** **Static**
* **Whenever PacketDropCountBlackhole is…** **Greater/Equal**.
* Set the **threshold value** to

1

.

1. Choose **Next** .
2. On the **Notification** page, configure the following:

* **Alarm state trigger:** **In alarm**
* **Send a notification to the following SNS topic:**  Create new topic
* **Create a new topic…**

PacketDropCountBlackhole-Alarm

* **Email endpoints that will receive the notification…** Enter your email address where you want to receive the alarm notification.
* Choose **Create topic**

1. Choose **Next** .
2. For **Alarm name**, enter

PacketDropCountBlackhole-Alarm

 .

1. Choose **Next** .
2. Review the settings on the next page and choose **Create alarm** .

 The **State** of the alarm may show **Insufficient data**. This will happen until enough data points are received by the alarm.

 Under **Actions** you may see Pending confirmation which means that you have not confirmed the subscription yet.

 Amazon SNS will send a confirmation response to your email address. Be aware that this can take a few moments for it to be delivered.

1. Go into your email, open the new email from **AWS Notifications** and choose the Confirm subscription link.

 A new browser page opens indicating the subscription status has changed to **Subscription Confirmed!**. This will be required to ensure you receive the notification.

**Task 2.5.3: Trigger Alarm**

In this section you will get the opportunity to trigger the alarm and test the notification that you configured.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC A - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC A - AZ1

 instance in **VPC A** try pinging any private IP address belonging in

10.3.0.0/16

 CIDR range for example

10.3.0.0

 .

 If the number of packets dropped is greater or equal to 1 because they matched a blackhole route of a Transit gateway, then it will activate the CloudWatch Alarm you created.

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**CloudWatch**

 and then choose **CloudWatch** from the list.

1. On the left navigation pane, under CloudWatch  **Alarms**, choose **All alarms**.
2. Confirm that the **State** of the **PacketDropCountBlackhole-Alarm** alarm is in  In alarm.

 Please note that it may take a few minutes before you see the state change to In alarm.

 The number of packets dropped crossed the set threshold. This will invoke the CloudWatch Alarm you created, which will send a message to the Amazon SNS topic. Amazon SNS will then send you an email message.

1. Check your email to confirm that you received a notification with subject **ALARM: “PacketDropCountBlackhole-Alarm”** alerting you that your Amazon CloudWatch Alarm has entered the ALARM state.

**Task 2.5.4: Create dashboard**

Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those resources that are spread across different Regions. You can use CloudWatch dashboards to create customized views of the metrics and alarms for your AWS resources.

In this section, you will create a CloudWatch Dashboard to view **PacketDropCountBlackhole** metric for the Transit Gateway.

1. On the left navigation pane, choose **Dashboards**.
2. Select **Create dashboard** .
3. Enter

PacketDropCountBlackhole-Dashboard

 for Dashboard name.

1. Choose **Create dashboard** .
2. Select **Number** on the **Add widget** page.
3. On **Add metric graph** page, under the **Metrics** section, select the **TransitGateway** card.
4. Select the **Per-TransitGateway Metrics** card.
5. Search for **PacketDropCountBlackhole** .
6. Select  **tgw-xxxxxxxxxxxxxx** Transit Gateway.
7. Choose **Graphed metrics (1)** tab.
8. Change the **Statistic** and **Period** parameters:

| **Key** | **Value** |
| --- | --- |
| Statistic | **Maximum** |
| Period | **1 Minute** |

1. Scroll to the top of the **Add metric graph** page. Near the top of the screen, by **Untitled graph**, choose the **Edit**  icon.
2. Enter a name for the graph, such as

PacketDropCountBlackhole-Graph

 .

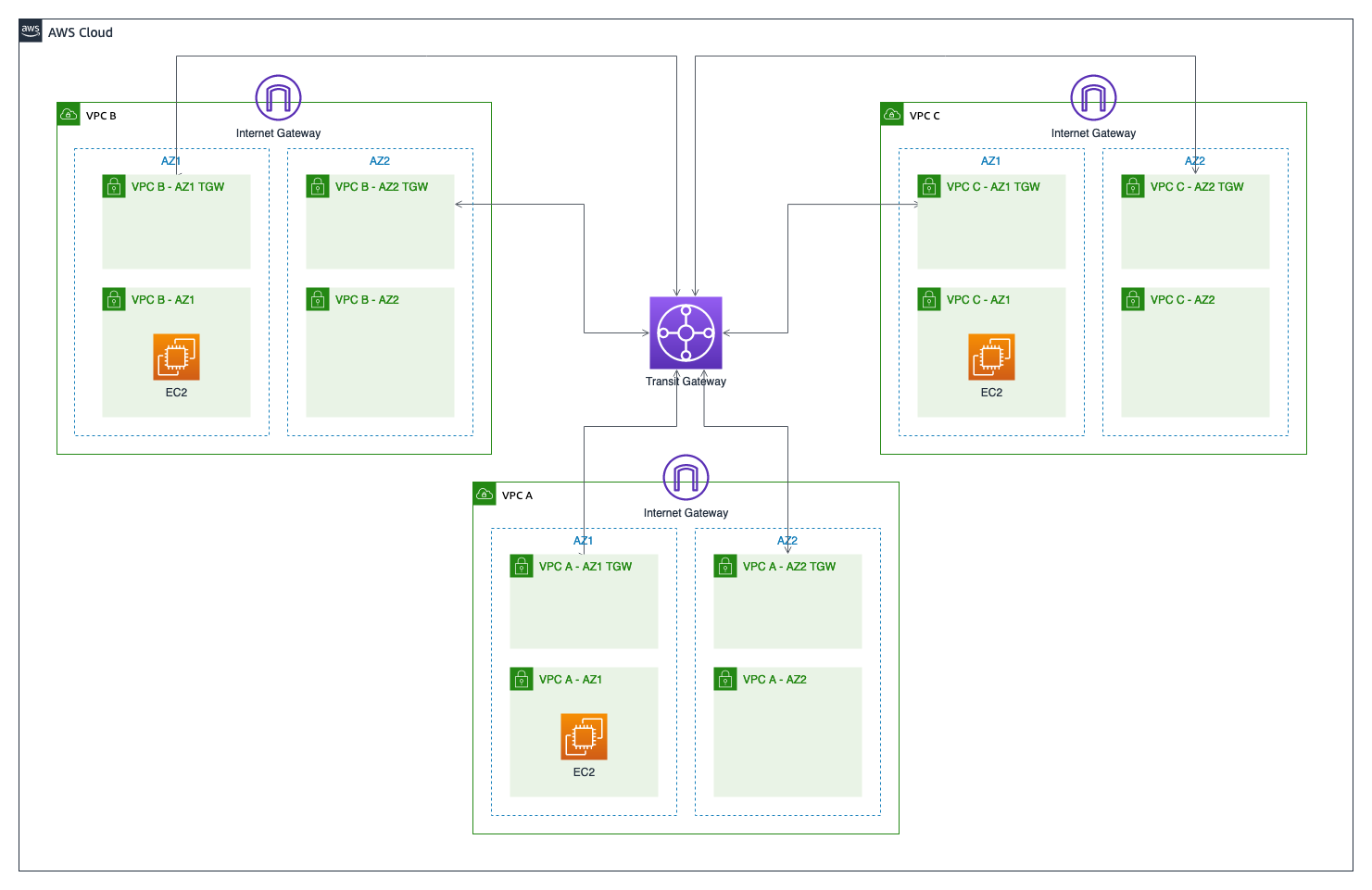
1. Choose **Apply** .
2. Choose **Create widget** at the bottom of the page.
3. At the top of the dashboard, choose **Save** .

 You can resize the widget by dragging the lower right corner of the graph. Change the data time span to **1h** for better visualization. After making these modifications, always remember to choose **Save** button to save your Dashboard changes.

**Task complete:** You have successfully generated traffic between instances, viewed the flow log in CloudWatch, configured a CloudWatch Alarm, and a CloudWatch Dashboard.

**Task 3: Security Controls**

Recall that at the end of [Task 1](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Task%201:%20Multi-VPC%20Account%20Architecture), you had provisioned 3 AWS VPCs and EC2 instances in each VPC. The VPCs were interconnected using Transit Gateway. The environment is depicted as below.



You will now be adding security to the AnyCompany environment you have built. You will use Network Access Lists (NACLs) and Security Groups, which are basic filtering mecahnisms.

TASK 3.1: MODIFYING DEFAULT NETWORK ACLS TO BLOCK, ALLOW SUBNETS

Network ACLs are stateless access controls you configure at a subnet level, to allow or block a CIDR block on a particular port or range of ports. Network ACL rules are numbered list and evaluated top down, with a DENY ALL at the end. If a rule is matched, subsequent rules are not evaluated.

Both inbound and outbound traffic can be controlled with these rules. By default when you have created the above subnets, the default Network ACL attached to them will have an ALLOW ALL rule for both inbound and outbound traffic.

In this task, you will modify Network ACL on **VPC A - AZ1** to allow only **VPC B**’s CIDR; and test connectivity using ping to send ICMP traffic from **VPC B** to **VPC A**, and from **VPC C** to **VPC A** as well.

 According to best practices it is recommended to use a separate subnet for each transit gateway VPC attachment in case of using NACLs with Transit Gateways. We keep the inbound and outbound NACL that is associated with the transit gateway subnets open, and will apply NACLs with filters to your workload subnets.

**Task 3.1.1: TGW Subnets & NACL configuration**

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**VPC**

 and then choose **VPC** from the list.

1. On the left navigation pane, under  **Security**, choose **Network ACLs**.
2. At the upper-right corner of the **Network ACLs** page, choose **Create network ACL** .
3. On the **Create network ACL** page, under **Network ACL settings**, configure the following:

* **Name:**

nacl-tgw-vpca

* **VPC:** **VPC A**
* **Tags:** Accept proposed Tags

1. Choose **Create network ACL** .
2. Repeat these steps to create network ACLs

nacl-tgw-vpcb

 for **VPC B** and

nacl-tgw-vpcc

 for **VPC C**.

1. On the **Network ACLs** page, select  **nacl-tgw-vpca**.
2. Choose the **Subnet associations** tab.
3. Choose **Edit subnet associations** .
4. From the **Edit subnet associations** page, select the following:

* **VPC A - AZ1 TGW**
* **VPC A - AZ2 TGW**

1. Choose **Save changes** .
2. Repeat these steps for **VPC B** and **VPC C** by choosing corresponding subnets with TGW in Subnet name.
3. On the **Network ACLs** page, select  **nacl-tgw-vpca**.
4. On **nacl-tgw-vpca** Network ACLs page, at the bottom of the page, choose the **Inbound rules** tab.

 Notice there is only one inbound rule that denies all incoming traffic.

1. At the right side of the **Inbound rules** section, choose **Edit inbound rules** .
2. On the **Edit inbound rules** page, choose **Add new rule** and configure the following:

* **Rule number:**

100

* **Type:** **All traffic**
* **Source:**

0.0.0.0/0

* **Allow/Deny:** **Allow**

1. Choose **Save changes** .
2. On **nacl-tgw-vpca** Network ACLs page, at the bottom of the page, choose the **Outbound rules** tab.

 Notice there is only one outbound rule that denies all incoming traffic.

1. At the right side of the **Outbound rules** section, choose **Edit outbound rules** .
2. On the **Edit outbound rules** page, choose **Add new rule** and configure the following:

* **Rule number:**

100

* **Type:** **All traffic**
* **Destination:**

0.0.0.0/0

* **Allow/Deny:** **Allow**

1. Choose **Save changes** .
2. Repeat the above steps to add same Inbound and Outbound Rules in **nacl-tgw-vpcb** and **nacl-tgw-vpcc** Network ACLs.

Now we’re ready to proceed to the next section and configure NACLs for EC2 instances to allow the ICMP traffic, TGW NACLs that allow everything will not introduce any blocking for us.

**Task 3.1.2: NACL configuration for EC2 subnets in AZ1**

1. On the left navigation pane, under  **Security**, choose **Network ACLs**.
2. In **Network ACLs** page, in *Filter network ACLs* search bar, search for **VPC A** .
3. Select  the ACL marked as the Default with a value of **Yes**.

 Check the default inbound rules in the Network ACL can be seen above. All traffic get evaluated for Protocol, Port and Source IP match. In this default Network ACL, all traffic are allowed into **VPC A-AZ1** Subnet by the first rule. The second rule which is a DENY ALL is not evaluated.

We will now modify the first rule (100) to allow only ICMP traffic from **VPC B**’s CIDR.

1. On Network ACLs page, at the bottom of the page, choose the **Inbound rules** tab.
2. At the right side of the **Inbound rules** section, choose **Edit inbound rules** .
3. On the **Edit inbound rules** page, modify rule number 100 and configure the following:

* **Type:** **All ICMP - IPv4**
* **Source:**

10.1.0.0/16

* **Allow/Deny:** **Allow**

1. Choose **Save changes** .

We have now completed modifying Network ACL of **VPC A - AZ1** Subnet to allow ONLY ICMP traffic from **VPC B**’s CIDR and all other traffic will be denied by the catch-all DENY rule. Let us now test this from **VPC B - AZ1** Subnet for ALLOW, and **VPC C - AZ1** Subnet for DENY.

 Note that You have not modified outbound rules, and the default outbound rule allows ALL traffic to flow out of the subnet.

TASK 3.2: TESTING CONNECTIVITY THROUGH NACLS FROM VPC B TO VPC A

Here, you will login to **EC2 VPC B - AZ1** using EC2 instance connect, and verify reachability to **EC2 VPC A - AZ1** over ICMP (ping).

1. At the top of the AWS Management Console, to the right of **Services** menu, in the search bar, search for

**EC2**

 and then choose **EC2** from the list.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC B - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC B - AZ1

 instance in **VPC B**, try pinging the private IP addresses of

EC2 VPC A - AZ1

 instance in **VPC A**.

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 The ICMP traffic should flow through and show as being successfully returned. Enter ‘CTRL + C’ to stop the ping.

You have now verified that the Network ACL on **VPC A - AZ1** subnet is allowing ICMP traffic to flow in and out, from **EC2 VPC B - AZ1**.

TASK 3.3: TESTING CONNECTIVITY THROUGH NACLS FROM VPC C TO VPC A

Similarly, connect to **EC2 VPC C - AZ1** from EC2 console, and ping **EC2 VPC A - AZ1** as you did from **VPC B** above.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC C - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC C - AZ1

 instance in **VPC C**, try pinging the private IP addresses of

EC2 VPC A - AZ1

 instance in **VPC A**.

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 The ping command will freeze without progress because the Network ACL in **VPC A - AZ1** subnet is DENYING all traffic from VPC C. Enter ‘CTRL + C’ to stop the ping see review the results. They should indicate that no replies were received.

**Task complete:** You have successfully modified the default Network ACL in **VPC A** to allow ICMP traffic only from **VPC B**; the only other rule is a DENY ALL. You verified that ICMP traffic flows through from **EC2 VPC B - AZ1** to **EC2 VPC A - AZ1** and DID NOT flow through from **EC2 VPC C - AZ1**.

TASK 3.4: MODIFYING SECURITY GROUP TO ALLOW ONLY ICMP TRAFFIC FROM VPC C TO VPC A

Security Groups are virtual, stateful firewalls attached to an instance or network interface. Both inbound and outbound rules can be defined to allow specific protocols, ports and source/destination CIDR. A DENY is not possible with security groups.

With Security Groups, all rules are evaluated before a network packet is allowed or blocked, unlike Network ACLs where the rules are evaluated in order of rule number and once a rule matches subsequent rules are not evaluated.

In section, you will modify the security group attached to **EC2 VPC A - AZ1** to allow only ICMP traffic inbound from **VPC C**’s CIDR only. You will verify that **EC2 VPC C - AZ1** is able to ping **EC2 VPC A - AZ1**, and **EC2 VPC B - AZ1** is not able to ping **EC2 VPC A - AZ1**.

**Prerequisite:** Edit the Network ACL in **VPC A - AZ1** subnet to revert the change to Rule 100, and set it to allow ALL TRAFFIC from all sources (0.0.0.0/0), because you want all traffic to flow past the Network ACL to the instance in **VPC A** to test the Security Group at the instance level.

1. Navigate to the browser tab with **VPC** service.
2. On the left navigation pane, under  **Security**, choose **Network ACLs**.
3. In **Network ACLs** page, in *Filter network ACLs* search bar, search for **VPC A** .
4. Select  the ACL marked as the Default with a value of **Yes**.
5. On Network ACLs page, at the bottom of the page, choose the **Inbound rules** tab.
6. At the right side of the **Inbound rules** section, choose **Edit inbound rules** .
7. On the **Edit inbound rules** page, modify rule number 100 and configure the following:

* **Type:** **All traffic**
* **Source:**

0.0.0.0/0

* **Allow/Deny:** **Allow**

1. Choose **Save changes** .
2. Navigate to the browser tab with **EC2 Instances**.
3. On the left navigation pane, choose **Instances** and select  **EC2 VPC A - AZ1**.
4. Navigate to the **Security** tab below and choose on the Security Group with ID starting with **sg- (VPC A EC2 Security Group)** in the lower pane to view and edit its rules.
5. On **VPC A EC2 Security Group** Security Group page, at the bottom of the page, choose the **Inbound rules** tab.
6. At the right side of the **Inbound rules** section, choose **Edit inbound rules** .
7. On the **Edit inbound rules** page, in the rule that is currently allowing All ICMP from **10.0.0.0/8** source, change it to allow only from **VPC C**’s CIDR:

* **Type:** **All ICMP-IPv4**
* **Source type:** **Custom**
* **Source:**

10.2.0.0/16

1. Choose **Save rules** .

TASK 3.5: TESTING CONNECTIVITY FROM VPC B TO VPC A THROUGH SECURITY GROUPS

Now you have modified the Security Group on **EC2 VPC A - AZ1** to allow ICMP traffic (ping traffic) only from instances in **VPC C**. You will now verify that you are NOT able to ping this instance from **EC2 VPC B - AZ1**, and you are ABLE to ping from **EC2 VPC C - AZ1**.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC B - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC B - AZ1

 instance in **VPC B**, try pinging the private IP addresses of

EC2 VPC A - AZ1

 instance in **VPC A**.

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 It will freeze and make no progress. This is because the Security Group on **EC2 VPC A - AZ1** has only ICMP inbound rules to allow **VPC C**. There are no rules for allowing **VPC B**, and an implicit DENY occurs. Enter ‘CTRL + C’ to stop the ping.

TASK 3.6: TESTING CONNECTIVITY FROM VPC C TO VPC A THROUGH SECURITY GROUPS

Similarly connect to **EC2 VPC C - AZ1** and try to ping **EC2 VPC A - AZ1**.

1. On the left navigation pane, choose **Instances**.
2. Select the  **EC2 VPC C - AZ1** instance.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. From the

EC2 VPC C - AZ1

 instance in **VPC C**, try pinging the private IP addresses of

EC2 VPC A - AZ1

 instance in **VPC A**.

 Refer to [Table 2.](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-100-NWNIDL-1%3A1.0.5-cb222a2a/en-US#Table%202.) for private IP addresses of the instances.

 The ping will succeed and traffic will flow through. This is because the SG on **EC2 VPC A - AZ1** is allowing ICMP traffic from **VPC C**’s CIDR range. Enter ‘CTRL + C’ to stop the ping.

**Task complete:** You have successfully modified the Security Group on **EC2 VPC A - AZ1** to allow only ICMP traffic from **VPC C**. You tested and verified that you cannot ping this instance from **VPC B**, but You are able to ping it from **VPC C** confirming the behavior of the Security Group.

**Conclusion**

 Congratulations! You now have successfully:

* Demonstrate intra versus inter-VPC routing.
* Select the appropriate connectivity options for an environment.
* Capture network traffic information (metadata) with VPC flow logs.
* Configure monitoring for networking statistics and metrics.
* Filter network traffic with Network Access Lists (NACLs), Security Groups (SG).

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional Resources**

For more information about the topics covered in this lab, see:

* [Amazon Virtual Private Cloud](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html).
* [Amazon VPC Peering](https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html).
* [AWS Transit Gateway](https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html).
* [Route tables](https://docs.aws.amazon.com/vpc/latest/tgw/how-transit-gateways-work.html#tgw-route-tables-overview)
* [Create a static route](https://docs.aws.amazon.com/vpc/latest/tgw/tgw-route-tables.html#tgw-create-static-route)
* [VPC Flow Logs](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html).
* [Amazon CloudWatch](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html).
* [Amazon CloudWatch Metrics](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/working_with_metrics.html).
* [Amazon CloudWatch Alarms](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html).
* [Amazon CloudWatch Dashboards](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html).
* [AWS Network Access Control List](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html).
* [Amazon Security Group](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html).

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).