

# Lab 1: Securing Amazon S3 VPC Endpoint Communications

## v3.0.2

Monday, March 1, 2021 9:43 AM

\*\*\*Challenge is to add a policy to the S3 endpoint.

# Advanced Architecting on AWS – Lab 1: Securing Amazon S3 VPC Endpoint Communications



© 2021 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. All trademarks are the property of their owners.

Corrections, feedback, or other questions? Contact us at [AWS Training and Certification](#).

## Lab Overview

Data security is crucial and should always be job one. AWS offers several solutions

## Lab Overview

Data security is crucial and should always be job one. AWS offers several solutions and best practices to help secure your data. Understanding how to make the best decisions to secure your data can be challenging. Running applications in private subnets, which have no route to the internet, provides data security by limiting the attack surface to only internal traffic. This is a great security measure; however, it can cause problems when your application needs to access data from services such as Amazon Simple Storage Service (Amazon S3).

To solve this problem, AWS provides Amazon Virtual Private Cloud (Amazon VPC) endpoints. With a VPC endpoint, you can privately connect your VPC to supported AWS services without requiring an internet gateway, NAT gateway, VPN connection, or AWS Direct Connect connection. Communication through the VPC endpoint does not require resources in your VPC to have public IP addresses. This enables traffic through your VPC endpoint to stay within the Amazon network.

In this lab, you create VPC endpoints and use them to access Amazon S3 from an Amazon Elastic Compute Cloud (Amazon EC2) instance located in a private subnet. To further improve data security, you create a VPC endpoint policy to restrict usage of the endpoint to specific resources.

## Objectives

After completing this lab, you will be able to:

- Understand private and public subnets and why they can or cannot communicate with Amazon S3
- Configure VPC endpoints using the AWS Management Console and AWS Command Line Interface (AWS CLI)
- Interact with Amazon S3 through a VPC endpoint in a private subnet
- Create a VPC endpoint policy to restrict resource access

## Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS, or Linux (Ubuntu, SuSE, or Red Hat)
- An internet browser such as Chrome, Firefox, or Microsoft Edge
- A plaintext editor

## Duration

This lab requires approximately **60** minutes to complete.

## AWS Services Not Used in This Lab

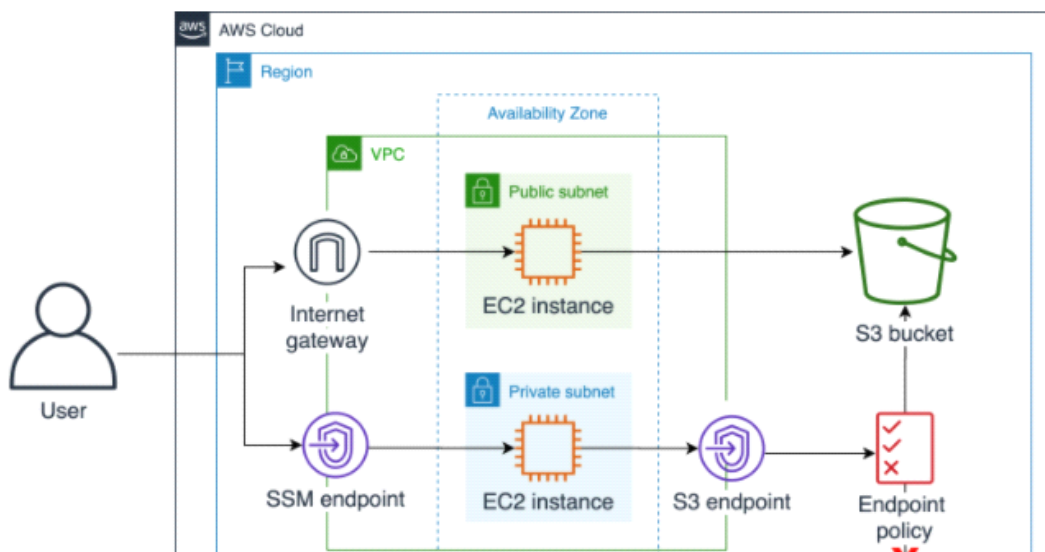
AWS services that are not used in this lab are disabled in the lab environment. In addition, the capabilities of the services used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

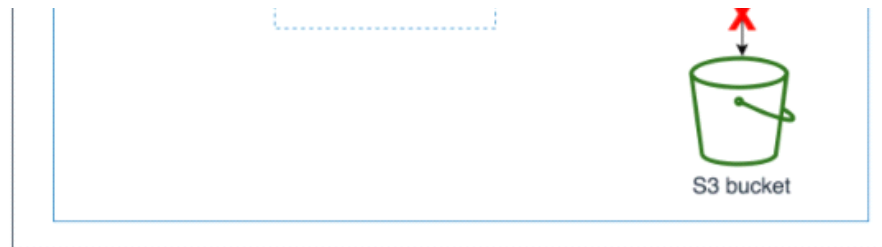
## Lab Environment

As part of the lab, a VPC with both public and private subnets has been created. The private subnet route table does *not* include a NAT gateway or internet gateway. This means that resources launched into a private subnet cannot communicate with the public internet or any AWS services that transfer data over the public internet.

To demonstrate how VPC endpoints work, an EC2 instance has been launched into a *public* subnet, and an identical EC2 instance has been launched into a *private* subnet. To verify that the VPC endpoint is allowing traffic from the *private* EC2 instance to access AWS services that require a public route, an Amazon S3 bucket has been created with a demo file in it.

The following diagram shows the resources provisioned for this lab and how they are connected at the end of the lab:





## Start Lab

1. At the top of your screen, launch your lab by choosing **Start Lab**

This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

**i** If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing **Open Console**

This opens an AWS Management Console sign-in page.

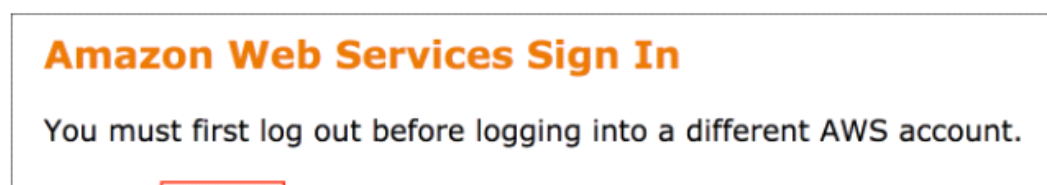
3. On the sign-in page, configure:

- **IAM user name:** `awsstudent`
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose **Sign In**

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

## Common Login Errors

**Error: You must first log out**



To logout, [click here](#)

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

- Choose **click here**
- Close your browser tab to return to your initial lab window
- Choose [Open Console](#) again

## Task 1: Explore the Lab Environment

In this task, you review the account resources that were provisioned for the lab. These resources include a VPC, subnets, an Amazon S3 bucket, and Amazon EC2 instances. You also create a VPC endpoint to allow AWS Systems Manager Session Manager access to your instances.

### Explore the Amazon VPC Resources

4. In the AWS Management Console, choose **Services** and select **VPC**.

During the lab setup, several VPC resources were created, including public and private subnets.

5. In the left navigation pane, choose the **Filter by VPC** box, and select **labVPC**.

6. In the left navigation pane, choose **Subnets**.

Notice that each Availability Zone has two subnets in the Region your lab was launched in. Of the two subnets in each Availability Zone, one is designated *public*, meaning it is associated with a route to the internet, and the other is designated *private*.

7. In the left navigation pane, choose **Route Tables**.

8. Select **PublicRouteTable**, and choose the **Routes** tab.

Notice the route for all traffic (*0.0.0.0/0*) to the internet gateway. This allows the associated subnets to be publicly accessible.

associated subnets to be publicly accessible.

9. Choose the **Subnet Associations** tab.

10. Expand the **Subnet ID** column so that you can read the full subnet IDs.

Notice that all of the *public* subnets are associated with this route table.

11. In the route tables list, select **PrivateRouteTable**.

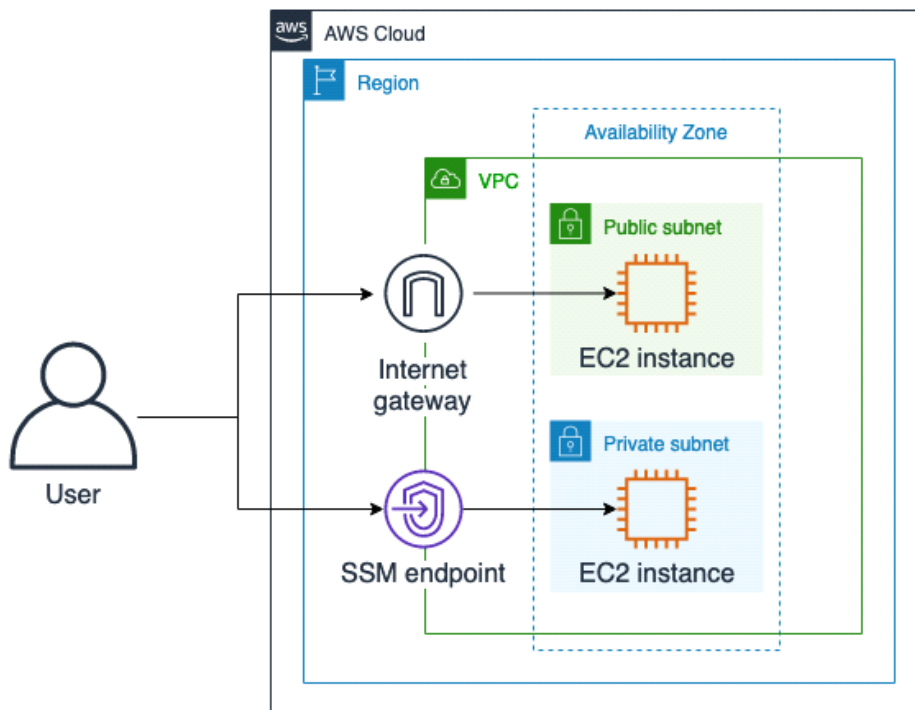
On the **Subnet Associations** tab, notice that all of the *private* subnets are associated with this route table.

12. Choose the **Routes** tab.

Notice that only a local route is associated with this subnet. This means that traffic can only traverse the internal AWS network.

## Create an Amazon VPC Endpoint

Now, create a Session Manager endpoint so that you can connect to your private EC2 instances without an internet gateway or NAT gateway, as shown in the following diagram:




13. In the left navigation pane, choose **Endpoints**.


Several endpoints have been created on your behalf to support this activity. To use



Session Manager, you need to create an *interface endpoint*.

 **Learn more** For more information about interface and gateway endpoints refer to [VPC Endpoints](#).

14. Choose **Create Endpoint**.
15. In the **Service Name** filter box, enter `ssm` and press the ENTER key.
16. Select **com.amazonaws.REGION.ssm**, where **REGION** is the Region that your lab was launched in.
17. For **VPC**, select **labVPC** from the drop-down menu.
18. In the **Subnets** section:
  - Select only Availability Zone **a** and uncheck the other availability zone.

 **Note** For example, if you are in the us-west-2 Region, select only the **us-west-2a** Availability Zone.

  - For **Subnet ID**, select **PublicSubnetA**
19. In the **Security group** section:
  - Select the group named **xxxx-SSHSecurityGroup-xxxx**
  - Select the group named **xxxx-PrivateSSHSecurityGroup-xxxx**
  - Uncheck the group named **default**
20. Choose **Add Tag** and configure:
  - Key: `Name`
  - Value: `SSM Endpoint`
21. Choose **Create endpoint**.
22. Choose **Close**.

You have created an interface endpoint, which allows connection to your EC2 instances without an internet gateway or NAT gateway.

 **Caution** Wait for the status to be **available** before proceeding to the next step. To refresh the status, choose the refresh  icon at the top of the page.

## Explore the Amazon S3 Resources

Two Amazon S3 buckets have been created for you. One bucket contains a file that you will try to access, and the other bucket contains the S3 bucket access logs.

23. In the AWS Management Console, choose **Services** and select **S3**.
24. In the buckets list, choose the name of the bucket that matches the **LabBucket** value from the left side of the lab page.

This bucket contains a file that you will later try to access from EC2 instances located in the public and private subnets.

25. To return to the buckets list, choose **Amazon S3** in the breadcrumbs at the top of the page.
26. In the buckets list, choose the name of the bucket that matches the **LabLoggingBucket** value from the left side of the lab page.

This bucket is used to contain log files for S3 bucket access.

## Explore the Amazon EC2 Resources

27. In the AWS Management Console, choose **Services** and select **EC2**.
28. In the left navigation pane, choose **Security Groups**.

Two security groups have been created for you: **xxxx-SSHSecurityGroup-xxxx** and **xxxx-PrivateSSHSecurityGroup-xxxx**. Note that both groups have an inbound HTTPS rule that allows all VPC traffic (*10.0.0.0/16*) access to port 443. The security groups attached to the VPC endpoint must allow incoming connections on port 443, which allows you to use Session Manager to connect directly to the instances.

29. In the left navigation pane, choose **Instances**.

Two instances have been created for you: one in a *public* subnet and one in a *private* subnet. You connect to these instances to run AWS CLI commands and determine how Amazon S3 is being accessed.

30. Select the **PublicCommandHost** instance.

On the **Details** tab, make note of the **Subnet ID** and **Availability Zone**. While you were looking at the VPC resources, you noted that **PublicSubnetA** was configured to route



through an internet gateway. You should be able to freely connect to this instance via any ports opened in the security group.

31. With **PublicCommandHost** still selected, choose **Actions** at the top of the page, and select **Connect**.

32. Choose the **Session Manager** tab.

33. Choose **Connect**.

A new browser tab is opened. You are now connected to an instance in the Public subnet.

34. Close the browser tab with the terminal connection.

35. Select the **PrivateCommandHost** instance.

This instance is located in **PrivateSubnetA**. To connect to this instance, you either need to use the **PublicCommandHost** instance as a bastion (jump) server or connect directly using Session Manager. You previously set up a Session Manager VPC endpoint, which allows you to connect directly to your private instances without needing to set up a NAT gateway.

36. With **PrivateCommandHost** still selected, choose **Actions** at the top of the page, and select **Connect**.

37. Choose the **Session Manager** tab.

38. Choose **Connect**.

A new browser tab is opened. You are now connected directly to an instance in a private subnet via the VPC endpoint.

39. To verify the connection by listing all the files in the current directory, run the following command:

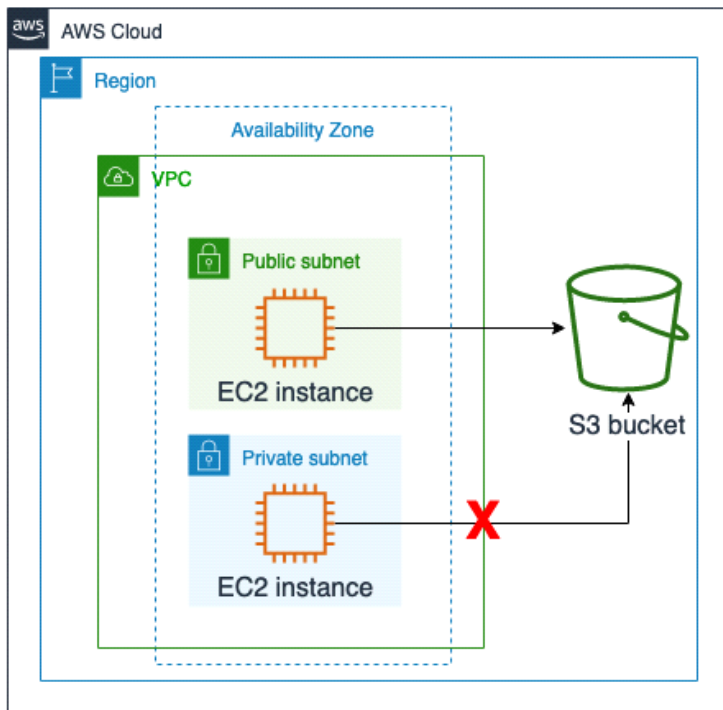
```
ls
```

40. Close the browser tab with the terminal connection.

## Task 2: Interact with Amazon S3 from Amazon EC2 Instances

Now that you have reviewed the AWS account resources that have been created on your behalf, explore VPC endpoints in greater depth. Connect to your public and private instances and attempt to access the file in your Amazon S3 bucket from each without using an Amazon S3 VPC endpoint.

As shown in the following diagram, you should be able to access the S3 bucket from the public instance but not from the private instance:



41. Copy the **PublicCommandHostURL** value from the left side of the lab page, paste it in a *new* browser tab, and press ENTER.

The command host terminal appears. You are now connecting into an SSH session on the **PublicCommandHost** server without having to use an SSH client.

42. To go to your home directory, run the following command:

```
cd ~
```

● **Suggestion** To help differentiate commands from output in the AWS CLI, run the following command. This adds a blank line before any output to the screen:

```
trap 'printf "\n"' DEBUG
```

You can also alter your command prompt to make output easier to read by exporting the PS1 variable. To do this, run the following command:


```
export PS1="\n[\u@\h \W] $ "
```

43. To configure the AWS CLI, run the following command:

```
aws configure
```

44. When prompted, configure:

- **AWS Access Key ID [None]:** Press ENTER
- **AWS Secret Access Key [None]:** Press ENTER
- **Default region name [None]:** Copy and paste the **Region** value from the left side of the lab page
- **Default output format [None]:** `json`

 **Note** If you get an error stating *"Partial credentials found in shared-credentials-file, missing: aws\_secret\_access\_key"*, run the following command to remove the credentials file and re-run **Step 43** otherwise proceed to **Step 45**.

```
rm ~/.aws/credentials
```

45. To list the S3 buckets you have access to in your account, run the following command:

```
aws s3 ls
```

Multiple bucket names are printed to the screen. Note that two of the buckets match the **LabBucket** and **LabLoggingBucket** names from the left side of the lab page.

46. To list all of the files in the **LabBucket**, run the following command. Replace *<LabBucket>* with the corresponding value from the left side of the lab page:

```
aws s3 ls s3://<labBucket>
```

```
aws s3 ls s3://<LabBucket>
```

This prints a list with the file that matches what you saw in the S3 bucket in the console earlier. Next, copy the file to the EC2 instance, edit it, and re-upload it to the server.

47. To copy the file from Amazon S3 to your local home directory, run the following command. Replace *<LabBucket>* with the corresponding value from the left side of the lab page:

```
aws s3 cp s3://<LabBucket>/demo.txt ~/
```

48. To print the contents of the file to the screen, run the following command:

```
cat demo.txt
```

You were able to download a file from Amazon S3 to the *public* instance. Next, modify the file and upload it back to the **LabBucket**.

49. To add text to the file and print the contents of the updated file to the screen, run the following command:

```
echo "  
This is some non-unique text that will be appended to your file." >>  
demo.txt  
cat demo.txt
```

50. To upload the updated file back to Amazon S3, run the following command. Replace *<LabBucket>* with the corresponding value from the left side of the lab page:

```
aws s3 cp demo.txt s3://<LabBucket>/
```

51. Switch to the browser tab with the AWS Management Console in it.
52. In the AWS Management Console, choose **Services** and select **S3**.
53. In the buckets list, choose the name of the bucket that matches the **LabBucket** value from the left side of the lab page.

54. Choose the name of the **demo.txt** file.

55. Choose **Object actions** and select **Open**. The file opens in a new browser tab.

Notice that you have successfully updated the file from the *public* instance.

**Note** If you do not find a file open in a browser tab, make sure that pop-ups are not being blocked.

Next, run the same commands on the instance located in the *private* subnet, which has no route to the internet. Amazon S3 is located outside of your VPC; without a route to the internet, you are not able to access the bucket. This also means that any requests from the *public* instance made to Amazon S3 need a route to the internet to access the internet routeable public Amazon S3 endpoint.

56. Copy the **PrivateCommandHostURL** value from the left side of the lab page, paste it in a *new* browser tab, and press ENTER.

The command host terminal appears. You are now connecting into an SSH session on the **PrivateCommandHost** server without having to use an SSH client.

57. To go to your home directory, run the following command:

```
cd ~
```

🗨 **Suggestion** To help differentiate commands from output in the AWS CLI, run the following command. This adds a blank line before any output to the screen:

```
trap 'printf "\n"' DEBUG
```

You can also alter your command prompt to make output easier to read by exporting the PS1 variable. To do this, run the following command:

```
export PS1="\n[\u@\h \W] $ "
```

58. To configure the AWS CLI, run the following command:

```
aws configure
```



59. When prompted, configure:

- **AWS Access Key ID [None]:** Press ENTER
- **AWS Secret Access Key [None]:** Press ENTER
- **Default region name [None]:** Copy and paste the **Region** value from the left side of the lab page
- **Default output format [None]:** `json`

**⚠ Note** If you get an error stating *"Partial credentials found in shared-credentials-file, missing: aws\_secret\_access\_key"*, run the following command to remove the credentials file and re-run **Step 58** otherwise proceed to **Step 60**.

```
rm ~/.aws/credentials
```

60. To list the S3 buckets you have access to in your account, run the following command:

```
aws s3 ls
```

After about 5 minutes, the command times out. The *private* instance does not have a route to Amazon S3 because there is no internet gateway, NAT gateway, or VPC endpoint.

## Task 3: Create the VPC Endpoint for Amazon S3

In this task, you create a VPC gateway endpoint to access Amazon S3 from your *private* instance. You use the AWS CLI to run commands and create the endpoint. For the AWS CLI to be able to communicate with resources outside the VPC, namely global AWS services, you need to run commands from the *public* instance.

61. Switch back to the tab that you connected to the *public* instance with.

62. To list the services that have VPC endpoints created for them, run the following command:



command:

```
aws ec2 describe-vpc-endpoints --query 'VpcEndpoints[*].ServiceName'
```

Notice that there is *no* Amazon S3 endpoint. The next steps guide you through the process of creating one. Absent the Amazon S3 endpoint, the instance in the private subnet does not have access to S3. Therefore, the Amazon S3 endpoint is created by the instance in the public subnet. Then, in **Task 4** the private instance will get access to S3.

63. To find the VPC ID for the **labVPC**, run the following commands:

```
VPC=$(aws ec2 describe-vpcs --query 'Vpcs[*].VpcId' --filters  
'Name=tag:Name, Values=labVPC' | jq -r '.[0]')  
echo $VPC
```


This command uses JQ to process the JSON output and put it in the correct format to be used later.

64. To find the route table ID for the *private* route table, run the following commands:

```
RTB=$(aws ec2 describe-route-tables --query 'RouteTables[*].RouteTableId'  
--filters 'Name=tag:Name, Values=PrivateRouteTable' | jq -r '.[0]')  
echo $RTB
```

65. To create an Amazon S3 endpoint, run the following commands:

```
export AWS_REGION=$(curl -s 169.254.169.254/latest/dynamic/instance-  
identity/document | jq -r '.region')  
echo $AWS_REGION  
aws ec2 create-vpc-endpoint \  
  --vpc-id $VPC \  
  --service-name com.amazonaws.$AWS_REGION.s3 \  
  --route-table-ids $RTB
```

 **Note** : Back slash are added to the above command in order to run as multi-line command.

The Amazon S3 VPC endpoint is now created.

66. To verify that the Amazon S3 VPC endpoint has been created, run the following command:

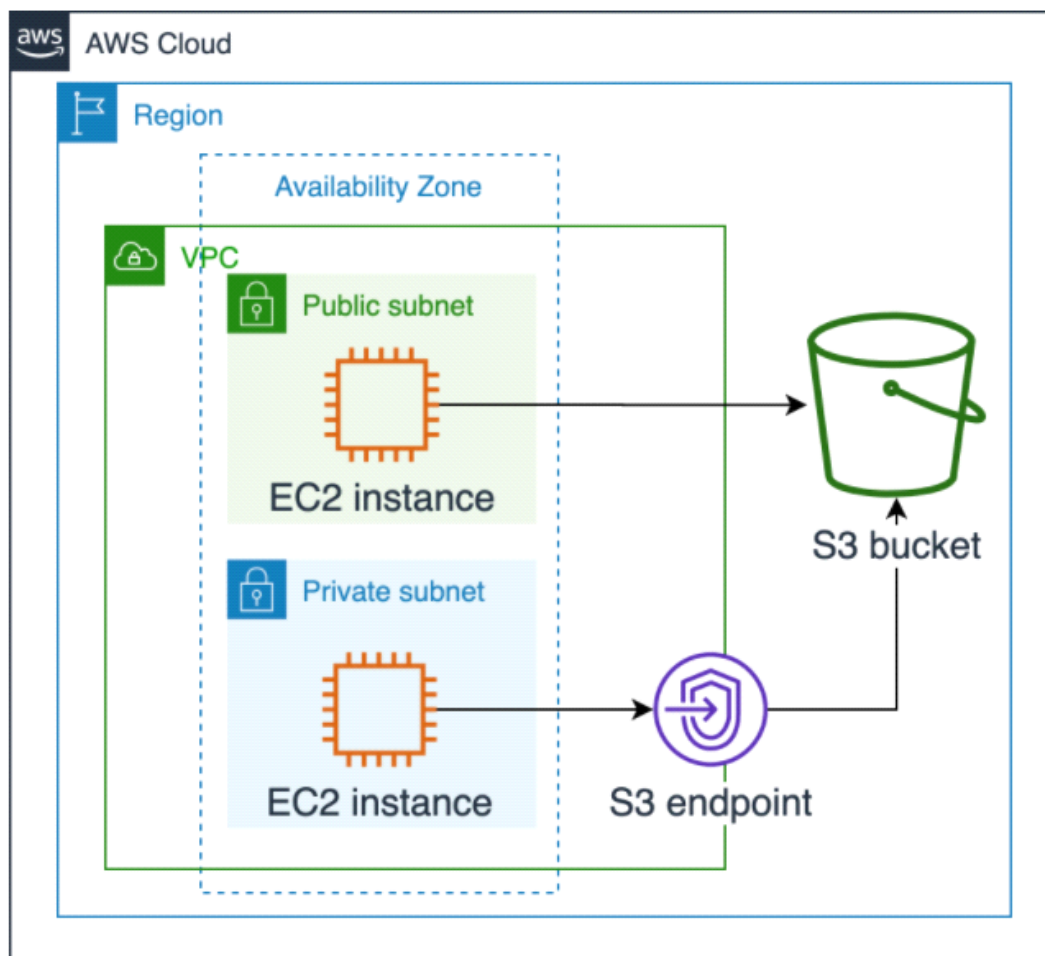
command.

```
aws ec2 describe-vpc-endpoints --query 'VpcEndpoints[*].ServiceName'
```

The Amazon S3 endpoint now appears in the list.


## Task 4: Interact with Amazon S3 via the Private Instance

In this task, you use the instance located in the *private* subnet, which was unable to access Amazon S3 earlier. Now that your Amazon S3 endpoint has been created and associated with the route table that your private subnet is associated with, you are able to access Amazon S3 directly, without using public resources, as shown in the following diagram:



67. Switch to the tab that you connected to the *private* instance with.
68. To list the S3 buckets you have access to in your account, run the following command:

```
aws s3 ls
```

 **Note** If this command does not work, you might need to wait a few minutes for the endpoint to finish creating. Wait a minute and try the command again.

69. To list all the files in the **LabBucket**, run the following command. Replace *<LabBucket>* with the corresponding value from the left side of the lab page:

```
aws s3 ls s3://<LabBucket>
```

This prints a list with the file that matches what you saw in the S3 bucket in the console earlier.

70. To display a list of files in the home directory of your instance, run the following commands:

```
cd ~  
ls -l
```

Notice that there are no local files because you downloaded the file from S3 to the *public* instance earlier and are now connected to the *private* instance.

71. To copy the file from Amazon S3 to your local home directory, run the following command. Replace *<LabBucket>* with the corresponding value from the left side of the lab page:

```
aws s3 cp s3://<LabBucket>/demo.txt ~/
```

72. To print the contents of the file to the screen, run the following command:

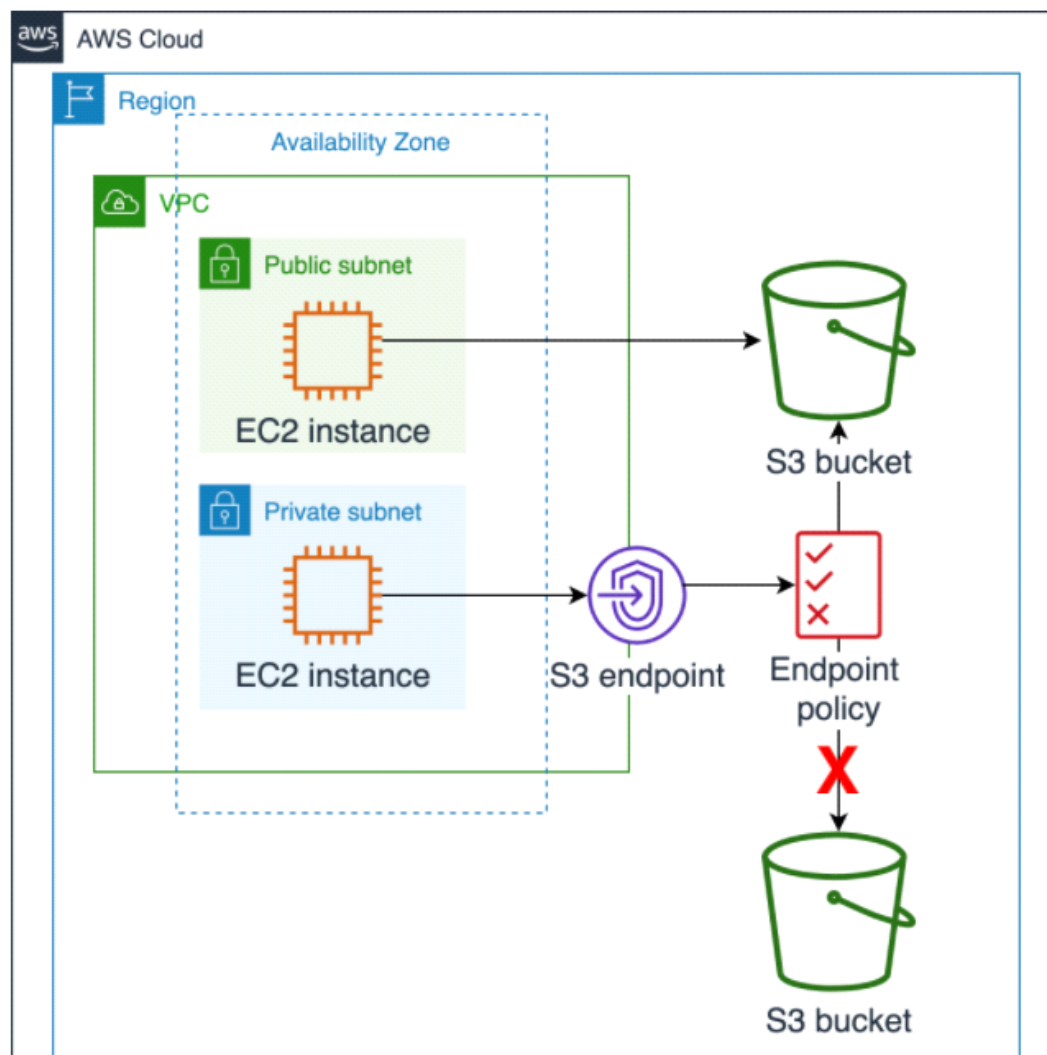
```
cat demo.txt
```

Now that you have the Amazon S3 VPC endpoint, you are able to download a file from Amazon S3 on the *private* instance.

## Challenge: Add a VPC Endpoint Policy

VPC gateway endpoints enable you to specify a policy to restrict access. Using an endpoint policy, you are able to specify exactly which Amazon S3 buckets an instance in your private subnet is allowed to access. This allows you to provide access to your data buckets but restrict access to your logging buckets, for example.

Your challenge is to use either the console or AWS CLI to add a policy to your Amazon S3 VPC gateway endpoint that *allows* access to the **LabBucket** but *denies* access to the **LabLoggingBucket**, as shown in the following diagram:



Use the following policy template as a starting point:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:List*",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<LabBucket>",
        "arn:aws:s3:::<LabBucket>/*"
      ]
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<LabLoggingBucket>",
        "arn:aws:s3:::<LabLoggingBucket>/*"
      ]
    }
  ]
}
```

If you have trouble, refer to the [Challenge Solution](#) section at the end of the lab.

## Conclusion

👍 Congratulations! You now have successfully:

- Understood private and public subnets and why they can or cannot communicate with Amazon S3
- Configured VPC endpoints using the AWS Management Console and AWS CLI
- Interacted with Amazon S3 through a VPC endpoint in a private subnet

- Created a VPC endpoint policy that restricted resource access

## End Lab

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

73. Return to the AWS Management Console.

74. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

75. Choose  **End Lab**

76. Choose  **OK**

77. (Optional):

- Select the applicable number of stars ☆
- Type a comment
- Choose **Submit**
  - 1 star = Very dissatisfied
  - 2 stars = Dissatisfied
  - 3 stars = Neutral
  - 4 stars = Satisfied
  - 5 stars = Very satisfied

You may close the window if you don't want to provide feedback.

For more information about AWS Training and Certification, see <http://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](#).




# Appendix

## Challenge Solution

The following steps outline how to add a VPC gateway endpoint policy to your Amazon S3 gateway endpoint. Instructions are provided for doing this through the console as well as the AWS CLI.

### Console Steps

78. In the AWS Management Console, choose **Services ▾** and select **VPC**.
79. In the left navigation pane, choose **Endpoints**.
80. Select the **com.amazonaws.REGION.s3** endpoint, where **REGION** is the Region that your lab was launched in.
81. Choose the **Policy** tab.
82. Choose **Edit Policy**.
83. Choose **Custom**.
84. Copy and paste the following policy into the text box:

 **Note** Replace *<LabBucket>* and *<LabLoggingBucket>* with the corresponding values from the left side of the lab page.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:List*",
      "Resource": "arn:aws:s3:::*
```

```

    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<LabBucket>",
        "arn:aws:s3:::<LabBucket>/*"
      ]
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<LabLoggingBucket>",
        "arn:aws:s3:::<LabLoggingBucket>/*"
      ]
    }
  ]
}

```

85. Choose **Save**.

You have now added a policy to your Amazon S3 gateway endpoint. You can test this from your *private* instance by running the following commands:

**Note** Replace *<LabBucket>* and *<LabLoggingBucket>* with the corresponding values from the left side of the lab page.

```

aws s3 ls s3://<LabBucket>
aws s3 ls s3://<LabLoggingBucket>

```

Did the policy work? Was your access to the logging bucket restricted?

[Click here](#) to return to the lab conclusion.

## AWS CLI Steps

86. Go to the browser tab that you connected to the *public* instance with.

87. To create a JSON file of the policy document, run the following commands. Replace *<LabBucket>* and *<LabLoggingBucket>* with the corresponding values from the left side of the lab page:

```

cd ~
cat <<EOT >> policy.json

```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:List*",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<LabBucket>",
        "arn:aws:s3:::<LabBucket>/*"
      ]
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<LabLoggingBucket>",
        "arn:aws:s3:::<LabLoggingBucket>/*"
      ]
    }
  ]
}
EOT
```

88. To find the VPC endpoint IDs, run the following commands:

```
export vpcEndpointId=$(aws ec2 describe-vpc-endpoints --query
'VpcEndpoints[?contains(ServiceName, `s3`) == `true`].VpcEndpointId' --
output text)

echo ${vpcEndpointId}
```

89. To attach the policy to the VPC endpoint, run the following command:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id ${vpcEndpointId} --policy-
document file://policy.json
```

You have now added a policy to your Amazon S3 gateway endpoint. You can test this from your *private* instance by running the following commands:

**⚠ Note** Replace `<LabBucket>` and `<LabLoggingBucket>` with the corresponding values from the left side of the lab page.


```
aws s3 ls s3://<LabBucket>
aws s3 ls s3://<LabLoggingBucket>
```

Did the policy work? Was your access to the logging bucket restricted?


[Click here](#) to return to the lab conclusion.


## Connect to Your Linux EC2 Instance Using Systems Manager

Systems Manager allows administrators to grant AWS Identity and Access Management (IAM) users the ability to create browser-based terminal sessions. This allows authenticated users to have one-step access to approved instances. This helps improve access by removing the need to download/share PEM keys and offering access to the server to anyone who does not have a terminal program (such as PuTTY).

 **Note** You need to be logged in to the AWS Management Console before you can establish a session.

90. Copy the **PublicCommandHostURL** or **PrivateCommandHostURL** value from the left side of the lab page, and paste it in a new browser tab. The command host terminal appears.

 **Note** If you are having difficulty using Systems Manager, ask your instructor for help.

 **Suggestion** To help differentiate commands from output in the AWS CLI, run the following command. This adds a blank line before any output to the screen:

```
trap 'printf "\n"' DEBUG
```

You can also alter your command prompt to make command output easier to read by exporting the PS1 variable. To do this, run the following command:

```
export PS1="\n[username@hostname] $ "
```

```
export FST=$(nmap -u@ -w ) $
```

91. Once connected, continue to [Task 2](#).

