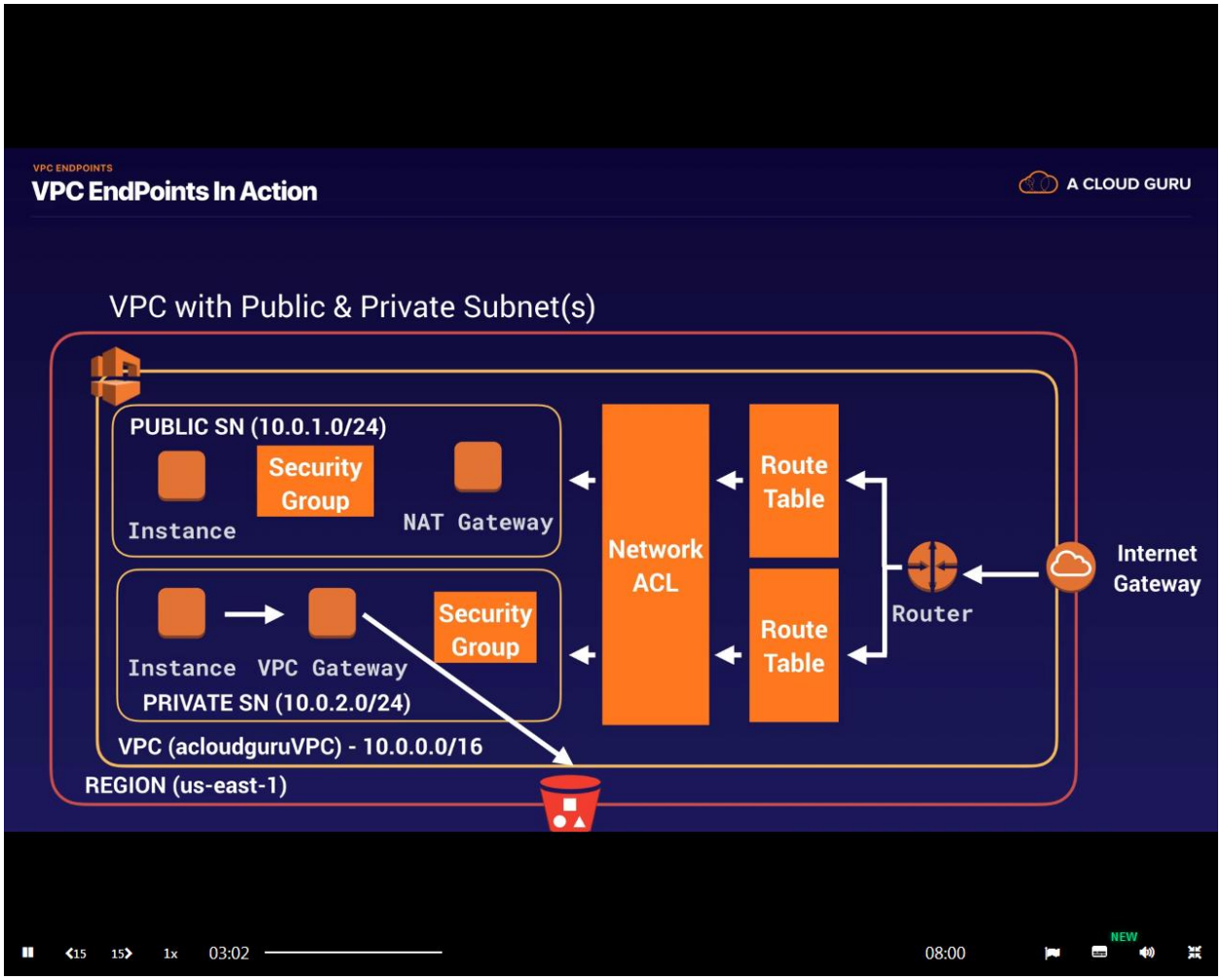
VPC Endpoints

What is a VPC endpoint?

A VPC endpoint allows you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect Connection. Instances inside your VPC do not require public IP addresses to communicate with resources in the service. **★Traffic between your VPC and the other service does not leave the amazon network.**

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your Network traffic.



(Notice that the VPC Gateway is located in the private subnet)

There are two types of VPC endpoints

Interface endpoints - Supported for many services

Gateway endpoints - supported for just S3 & DynamoDB

Let's create an S3 Gateway Endpoint.

(So instead of traversing from the instance in the private subnet to a NAT Gateway and then outside the AWS network to S3, we simply use an S3 Gateway Endpoint and never have to leave the AWS network.)

★ You need at least one bucket in S3 before starting this lesson. If you don't have one, create one before proceeding.

First we need to create a role that allows our EC2 instances to access S3.

Click **Services** click **IAM (Identity and Access Management)** located under Security, Identity, & Compliance

Click **Roles** in the left margin

Click **Create role**

Choose a use case: ✓EC2 ← Select EC2

Click Next: Permission

Type s3 into the filter

✓AmazonS3FullAccess ← select this policy

Then click **Next: Tags** ←No need for tags

Then click **Next: Review**

Role name*: S3AdministratorAccess

Role description: Allows EC2 instances to call AWS services on your behalf)

Click **Create role**

Click **Services** and click **EC2** click **running instances**

✓DBServerinPrivateSubnet ← select our instance that is in our private subnet

Click **actions** and hover over **instance settings** and click **attach/replace IAM role**

IAM role*: S3AdministratorAccess ← select the role we just created

Click **apply**

Click **Services** and click **VPC**

To make our lives easier for this lesson let's associate both of our subnets to the default NACL that was created when we created our VPC. As you recall the default NACL allows all inbound and outbound traffic by default

Click Network ACLs in the left margin

✓Select the default NACL that was created when we created our VPC (**Not** our PublicSubnetNACL)

Click the **subnet associations** tab

Click **Edit subnet associations**

✓10.0.1.0-ap-southeast-1a-PublicSubnet ← select our public subnet

Click **edit**

So for simplicity both of our subnets are now associated with our default NACL

Let's SSH into our instance located in our private subnet and to do so we first need to SSH into our instance located in our public subnet

Click **Services** and click **EC2** click **Running instances**

✓WebServerInPublicSubnet ←select our instance in our public subnet

Click the **connect** button

✓EC2 Instance Connect (browser-based SSH connection) ←select this

Click **connect**

Now that we are SSh'd in type the following:

sudo su (and hit enter)

clear (and hit enter)

type ls (and hit enter) ← (that is a lowercase L not a capital i)(this will show us the file called SingaporeKeyPair.pem that we created in the NAT instance/gateway lesson. Refer back if necessary)

type the following:

ssh ec2-user@10.0.2.118 -i SingaporeKeyPair.pem (and hit enter)(The 10.0.2.118 number is the private IP address of our instance located in our private subnet. In the EC2 dashboard just select DBServerinPrivateSubnet and scroll down to the description tab to find the private IP address…yours will be different than 10.0.2.118 when you do the exercise)

We are SSH'd into our private instance

Type the following:

sudo su (and hit enter)

clear (and hit enter)

aws s3 ls (and hit enter) (this will list all of the s3 buckets that we have)

echo "test" > test.txt (and hit enter)

ls (and hit enter) ← (that is a lower case L not a capital i) (we can now see the test.txt file has been created)

aws s3 cp test.txt s3://thisisatestbucket987654321 (and hit enter) (that has now copied the text.txt file over to our bucket in S3) (You can check that it is there by going to the AWS console and clicking **services** and then **S3** and open your bucket)

Now minimize the command line and go to the AWS console to the VPC dashboard

Click **route tables** in the left margin

✓Select our main route table (the one that is NOT OurPublicRoute)

Click the **Routes** tab

Click Edit Routes

And you will see the following:

Destination	Target	Status	Propagated
Leave as is → 10.0.0.0/16	local	active	No
Leave as is → 2406:da18:233:9500::/56	local	active	No
Delete this → 0.0.0.0/0	NAT_Gateway	active	No

Just delete the route that has the target as our NAT_Gateway

Then click **Save Routes**

Now return back to our terminal where we are SSH'd into our Private instance

Type clear (and hit enter)

Type aws s3 ls (and hit enter)

(Nothing happens because it has no route out to the internet since we just removed that route from our main route table)

So let's now create a VPC Endpoint

Minimize the terminal and return to the AWS console

Click Services and click VPC

Click Endpoints in the left margin

Click Create endpoint

Service Category: ✓AWS services ← select this

Type gateway into the filter

Service Name	Owner	Type
✓com.amazonaws.ap-southeast-1.s3	amazon	Gateway

VPC*: ✓OurNewCustomVPC ← select OurNewCustomVPC

Then ✓Select the Main Route Table Associated with 10.0.2.0-ap-southeast-1b-PrivateSubnet

(because the subnet associated with the route table we selected will be able to access the endpoint we are creating)

Policy*: ✓Full Access ←select this

Click Create Endpoint (This has created our VPC endpoint)

Click Route Tables in the left margin

✓Select our main route table and click the routes tab

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
2406:da18:233:9500::/56	local	active	No
Pl-6fa54006 (com.amazonaws.ap-southeast-1.s3,…	vpce-014b47e5c08129164	active	No

(In a few minutes or less you will notice that a new route has appeared with the target as our VPC endpoint)

Return to the terminal where we are SSHd into our private instance and type the following:

aws s3 ls (and hit enter)← that is a lowercase L not an uppercase I

(Nothing appears to be happening)

Hit Ctrl + C to stop the operation

Now type aws s3 ls --region ap-southeast-1 (and hit enter)

That will list our buckets (for some reason there is a requirement that you include your region)

So we are viewing our buckets in S3 and we are not leaving the AWS network to do so. We are doing it through our VPC endpoint.

Mission Complete. Below are important concepts and practice questions pertaining to VPC Endpoints.

★**VPC Endpoint Cheat Sheet**

-VPC Endpoints help keep traffic between AWS services **within the AWS Network**

-There are two kinds of VPC Endpoints. Interface Endpoints and Gateway Endpoints

-Interface Endpoints **cost money,** Gateway Endpoints **are free.**

-Interface Endpoints uses an Elastic network Interface (ENI) with a Private IP address (powered by AWS PrivateLink)

-a Gateway Endpoint is a target for a specific route in your route table

-Interface Endpoints support many AWS services

-Gateway Endpoints only support DynamoDB and S3

Question
A shipping company brokers transportation arrangements for a large freight carrier. The shipping company currently uses an online form to make reservations with the freight carrier, but they'd like to automate the ordering process. The freight carrier runs its logistics system on AWS. The shipping company also runs its IT infrastructure on AWS. Which architecture should the shipping company put in place to provide the best security and operational efficiency for their transactions with the freight carrier?

A) Have the freight carrier create an Endpoint Service and use an Interface VPC Endpoint to connect
B) Create an IPSec VPN tunnel to the freight carrier's network thorough a Virtual Private GatewaySelected
C) Implement VPC Peering to the freight carrier's VPC
D) Establish a Direct Connect circuit to the freight company

Explanation:
AWS PrivateLink provides private connectivity between VPCs and AWS services securely on the AWS network without exposure to the public Internet. AWS PrivateLink is implemented by a service provider creating an Endpoint Service and a service consumer connecting via an Interface VPC Endpoint. Direct Connect is established between a customer and AWS, not between two AWS customers. VPC Peering and VPN connections will work, but will require more operational overhead than PrivateLink. Answer is A.

Question

Your application is using S3 to store some customer generated data, however your security team has mandated that this data not traverse the internet. What would be your advice as an AWS Solutions Architect?

A) This is not possible - S3 is a service that must be accessed across the internet
B) Create an S3 Direct Connect for the applicationSelected
C) Create an S3 VPN Endpoint in the VPC the application resides in
D) Create a VPC Endpoint for S3 inside the VPC that the application resides in

---

Explanation:
This scenario is best addressed by use of a VPC endpoint, which creates a private route for the VPC to communicate with S3 without having to traverse the internet. S3 VPN Endpoint and S3 Direct Connect are not actual services that can be used. Answer is D.