**Secure Hybrid Access to S3 using VPC Endpoint**

[Access S3 From On-premises]

**Introduction**

VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between your compute resources and AWS services without imposing availability risks.

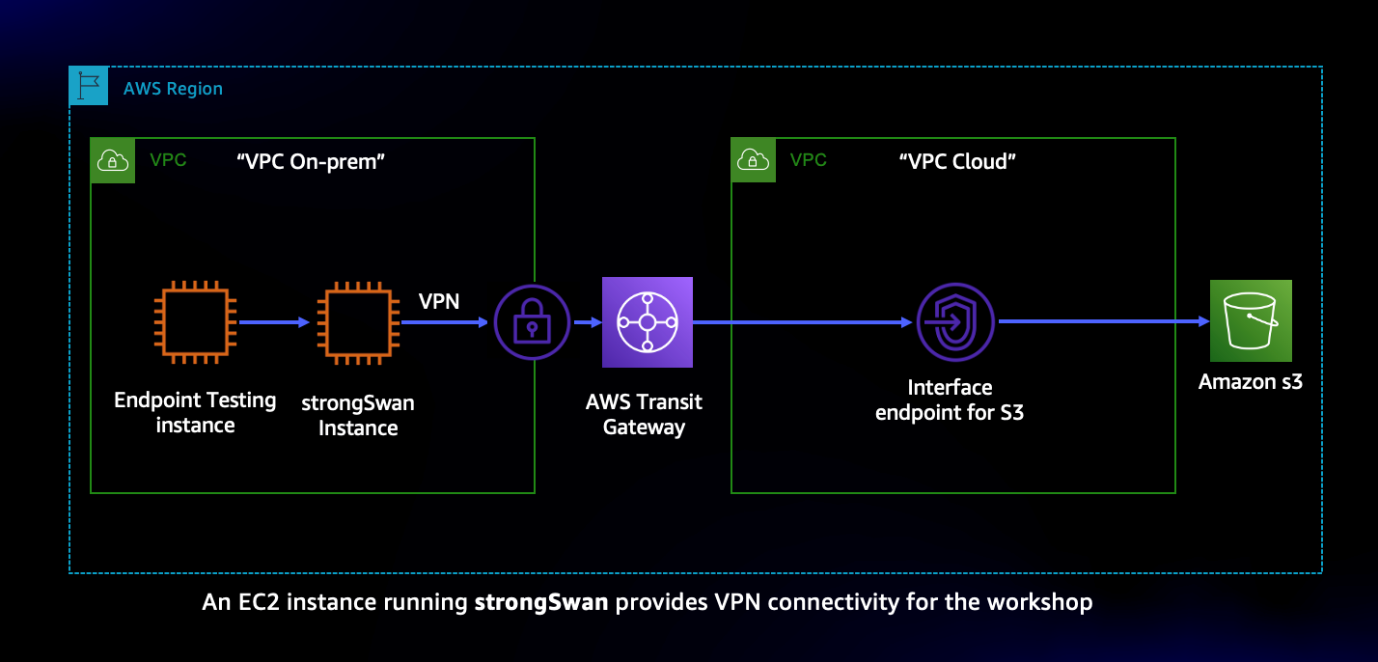
**Project Description – Scenario**

In this project, "VPC On-Prem" simulates an on-premises environment. An EC2 instance running strongSwan VPN software has been deployed in "VPC On-prem" and automatically configured to establish a Site-to-Site VPN tunnel with AWS Transit Gateway. This VPN simulates connectivity from an on-premises location to the AWS cloud.

The Intention of this project is to demonstrate, how you can create, configure, and test VPC endpoints that enable your workloads to reach AWS services without traversing the Public Internet. I will create an Interface endpoint to access Amazon S3 from a simulated on-premises environment. The Interface endpoint will allow you to route to Amazon S3 over a VPN connection from your simulated on-premises environment.

I will also create a VPC endpoint policy, which controls access to specific S3 resources through the VPC endpoint.

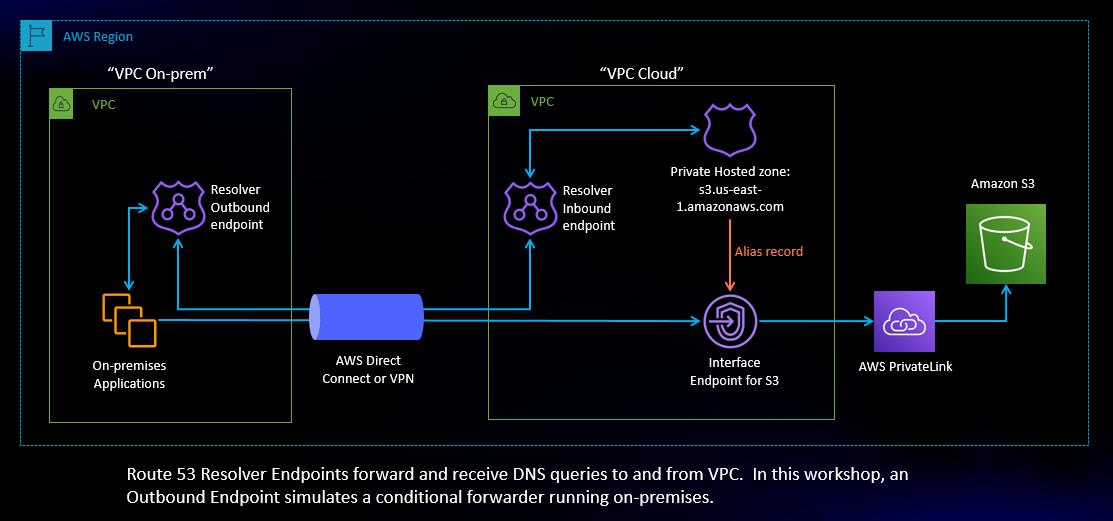
**Architecture – Overview Deploy CloudFormation Stack | Main CF**



**When to use what**

* **Gateway endpoints** only work with resources running in the VPC where they are created.
* **Interface endpoints** work with resources running in VPC and resources running in on-premises environments, connectivity from your on-premises environment to the cloud can be provided by AWS Site-to-Site VPN or AWS Direct Connect.
* **Interface endpoints** allow you to connect to services that’s powered by AWS Private Link.

**Deploy CloudFormation Stack | additional service | Depiction**



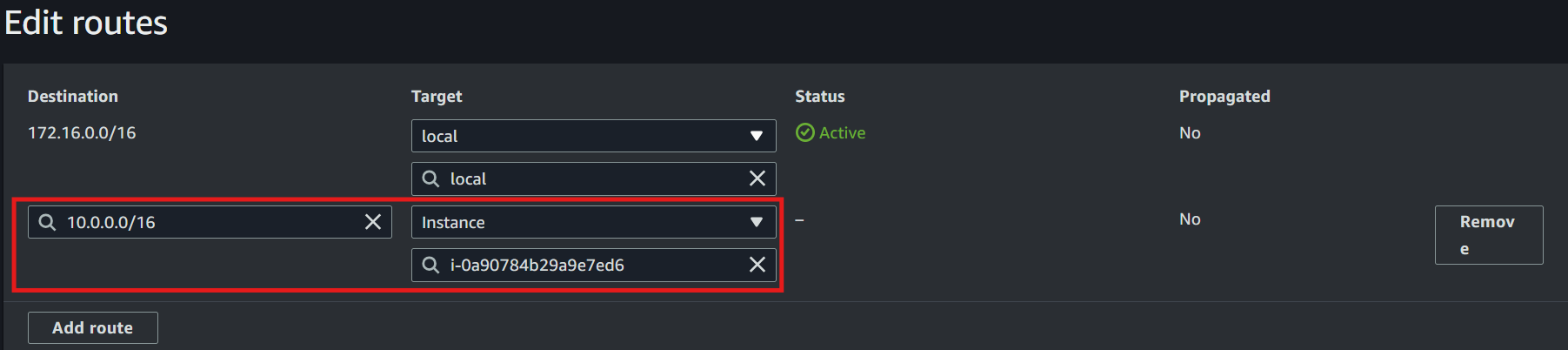
**Creation**

* One Route 53 Private Hosted Zone that hosts Alias records for the PrivateLink S3 endpoint
* One Route 53 Inbound Resolver endpoint that enables **"VPC Cloud"** to resolve inbound DNS resolution requests to the Private Hosted Zone
* One Route 53 Outbound Resolver endpoint that enables **"VPC On-prem"** to forward DNS requests *fo*r **S3** *to* **"VPC Cloud"**

**Updating the On-Premises private route table**

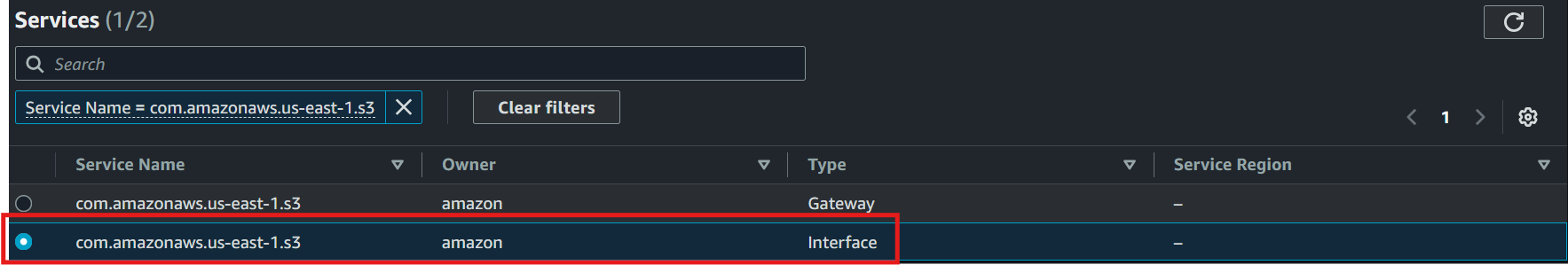
This project uses a strongSwan VPN running on an EC2 instance to simulate connectivity between an on-premises datacentre and the AWS cloud.

 I will update the VPN configuration and modify the **"VPC On-prem"** routing table to direct traffic destined for the cloud to the strongSwan VPN instance.



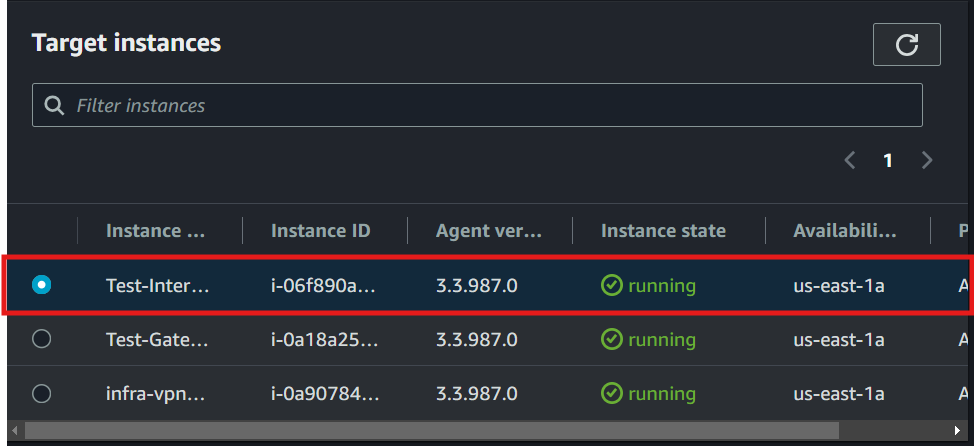
**Create an S3 Interface endpoint**

I will create and test an S3 interface endpoint using the simulated on-premises environment deployed.



**Testing the Interface Endpoint**

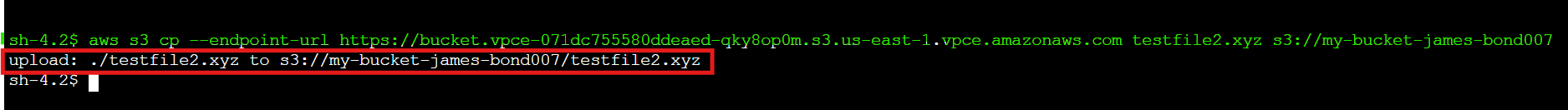
This EC2 instance named Test-Interface-Endpoint. This EC2 instance is running in "VPC On-prem" and will be used to test connectivity to Amazon S3 through the Interface endpoint

****

I will copy an object to my S3 bucket using my endpoint URL parameter because I need to use the endpoint-specific DNS name to access S3 using an Interface endpoint.

This **command** uses the Regional DNS name for the endpoint.

aws s3 cp --endpoint-URL https://bucket.vpce-071dc755580ddeaed-qky8op0m.s3.us-east-1.vpce.amazonaws.com testfile2.xyz s3://**my-bucket-james-bond007**



A screenshot of a computer

Description automatically generated

**Note**: When you create an interface endpoint, PrivateLink generates two types of endpoint-specific DNS names: Regional and Zonal. Regional DNS names include a unique VPC endpoint ID, a service identifier, the AWS Region, and vpce.amazonaws.com in their DNS name. When querying a Regional DNS name, the results will include IP addresses in each Availability Zone where your endpoint is deployed. Zonal DNS names include the Availability Zone in their DNS name. You might use a Zonal DNS name if your architecture needs to isolate traffic to an endpoint deployed in a specific Availability Zone instead of any Availability Zone in the region**.**

**On-premises DNS Simulation**

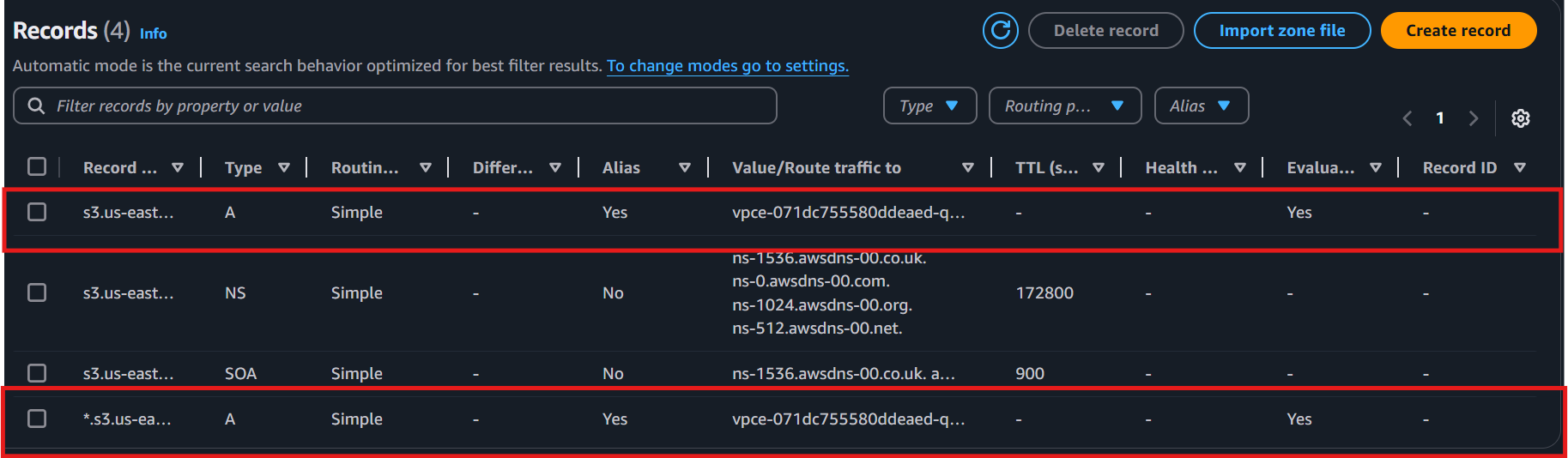
AWS PrivateLink endpoints have a fixed IP address in each Availability Zone where they are deployed,

These IP addresses are attached to Elastic Network Interfaces (ENIs).

**AWS recommends using DNS to resolve the IP addresses for endpoints so that downstream applications use the latest IP addresses when ENIs are added to new AZs or deleted over time.**

**DNS Alias Records for the Interface endpoint**

In this section I will create a forwarding rule to send DNS resolution requests from a simulated on-premises environment to a Route 53 Private Hosted Zone.



**Resolver Forwarding Rule**

Route 53 Resolver Forwarding Rules allow you to forward DNS queries from your VPC to other sources for name resolution**.**

I will create a forwarding rule that forwards DNS queries for Amazon S3 to a Private Hosted Zone running in "VPC Cloud" in-order to resolve the PrivateLink interface endpoint regional DNS name.

A screenshot of a computer

Description automatically generated

**Resolver IP addresses**

A screenshot of a computer

Description automatically generated

**Testing the on-premises DNS Simulation**

Testing DNS resolution. The dig command will return the IP addresses assigned to the **VPC Interface endpoint** running in VPC Cloud.

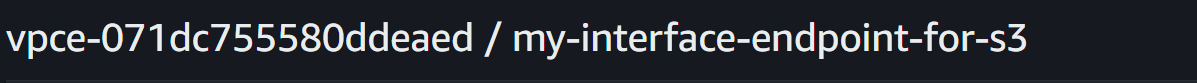
**command** dig +short s3.us-east-1.amazonaws.com

A black screen with white text

Description automatically generated

The IP address look similar because they are from the same VPC Cloud CIDR block.

 The S3 Interface endpoint



A screenshot of a computer

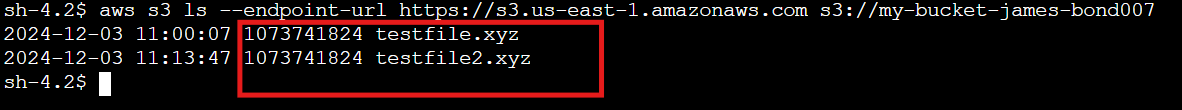
Description automatically generated

**Test listing your S3 buckets**

A computer screen shot of a black screen

Description automatically generated

**Test listing the contents of a specific S3 bucket**

****

**Summary**

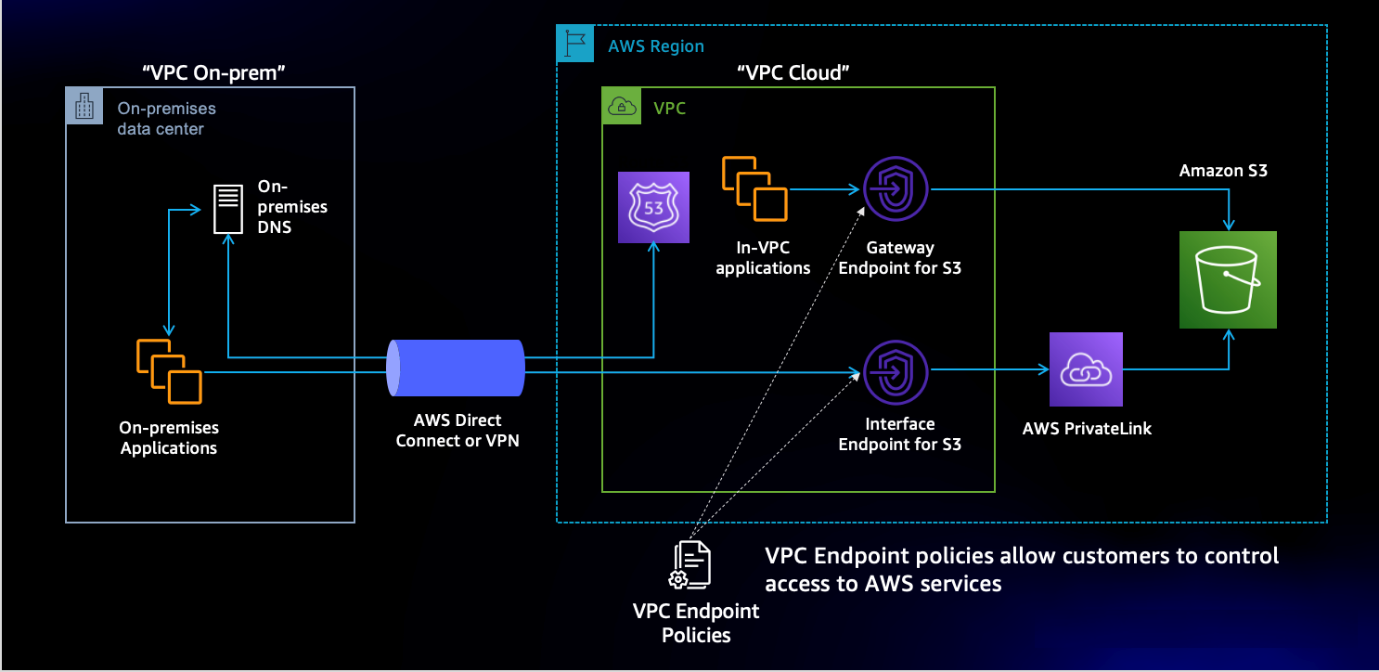
In this section I created an Interface endpoint for Amazon S3. This endpoint can be reached from on-premises through Site-to-Site VPN or AWS Direct Connect. Route 53 Resolver outbound endpoints simulated forwarding DNS requests from on-premises to a Private Hosted Zone running the cloud. Route 53 inbound Endpoints received the resolution request and returned a response containing the IP addresses of the VPC interface endpoint using DNS to resolve the endpoint IP addresses provides high availability in-case of an Availability Zone outage.

**VPC Endpoint Policies**

When you create an interface or gateway endpoint, you can attach an endpoint policy to it that controls access to the service to which you are connecting. A VPC endpoint policy is an IAM resource policy that you attach to an endpoint. If you do not attach a policy when you create an endpoint, AWS attaches a default policy for you that allows full access to the service through the endpoint.

I will create a VPC endpoint policy that restricts access to the S3 bucket specified in the VPC endpoint policy.

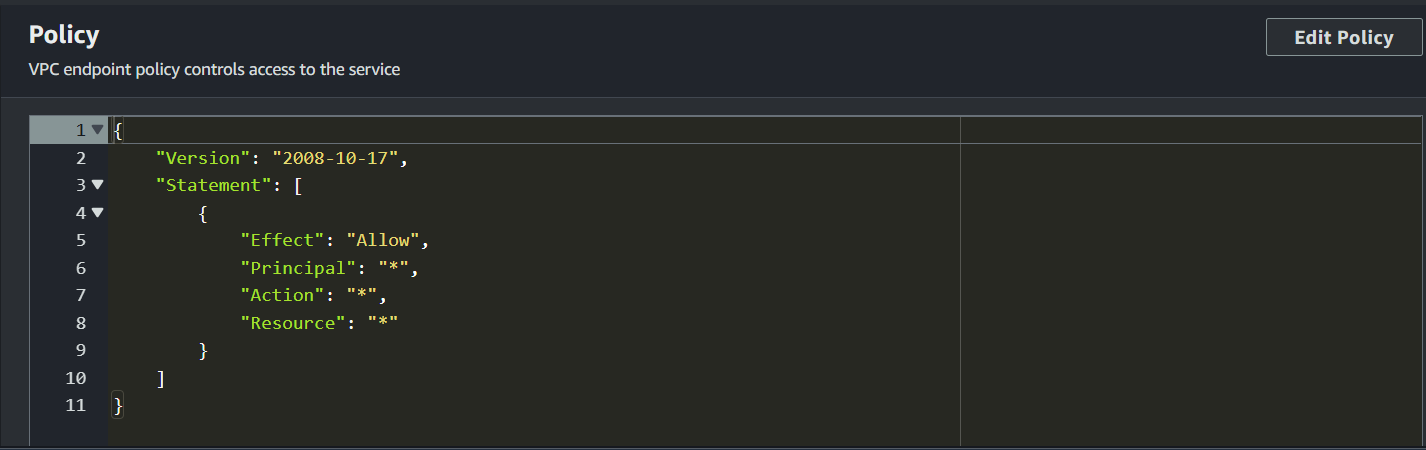
**Architecture – Overview - Depiction**



I created a new bucket called my-bucket-james-bond008

Note: The default policy allows access to all S3 Buckets through the VPC endpoint. I will edit the **Policy**, then change the Policy option from **Full Access** to **Custom**.

**Current Full Access**



**Edited Custom Access**

I replaced the bucket names with my newly created bucket

A screenshot of a computer program

Description automatically generated

**Test-Gateway-Endpoint** **instance**, test access to the S3 bucket

A screenshot of a computer

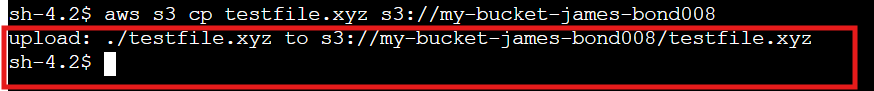
Description automatically generated

This command returns an error because access to **my-bucket-james-bond007** bucket is not permitted by my new VPC endpoint policy

A screen shot of a computer

Description automatically generated

I will upload data to my 2nd bucket, this operation will succeed because it is permitted by the VPC endpoint policy



**Summary**

In this section, I created a VPC endpoint policy for Amazon S3 and used the AWS CLI to test the policy. AWS CLI actions targeted to my original S3 bucket failed because I applied a policy that only allowed access to the second bucket you created. AWS CLI actions targeted for my second bucket succeeded because the policy allowed them. These policies can be useful in situations where you need to control access to resources through VPC endpoints.

**Done!**