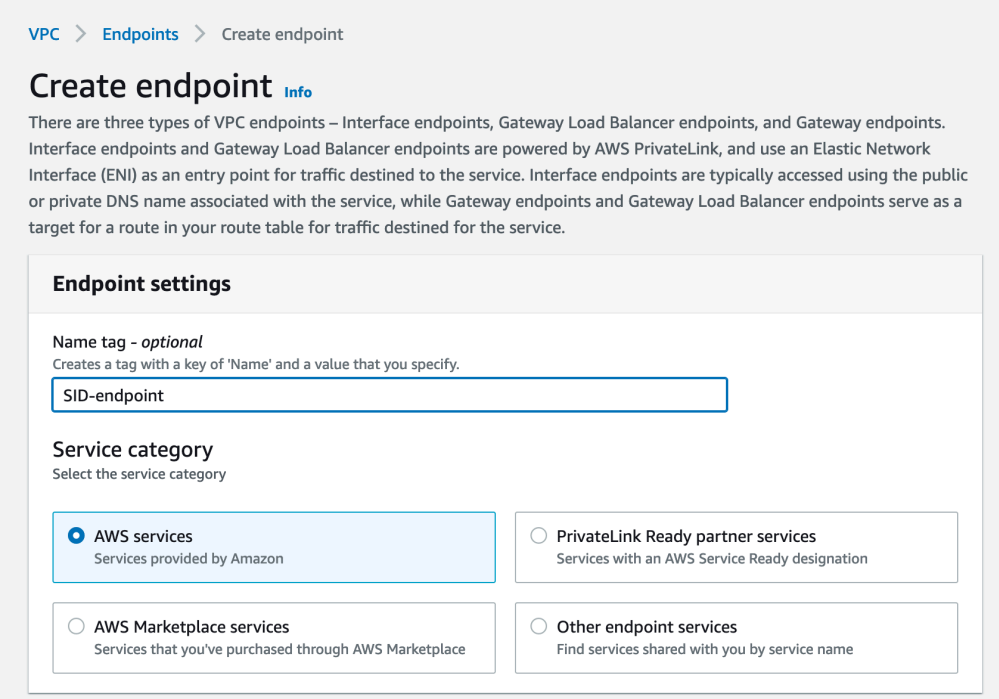
Restrict Access to an S3 VPC Endpoint

You can simplify access to S3 resources from within a VPC by using a VPC Endpoint. These endpoints are easy to configure, highly reliable, and provide a secure connection to S3 that does not require a gateway or NAT instance.

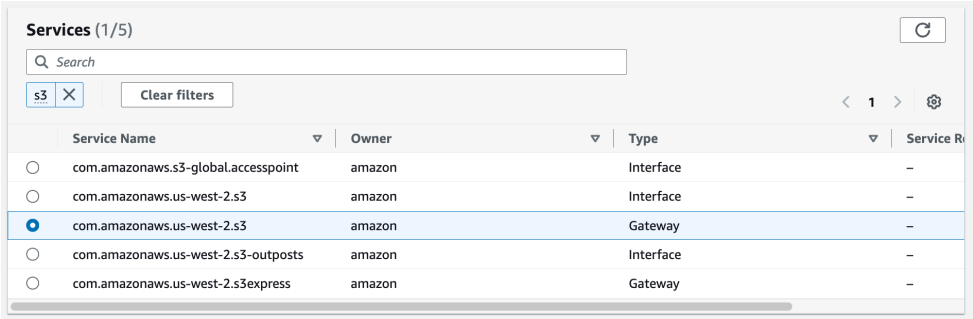
In this exercise, we will configure a S3 VPC Endpoint and a bucket policy to limit access to only requests that pass through the VPC Endpoint. This is an easy way to limit access to only clients in your VPC.

From the AWS console in the top search bar, search and select `VPC`.

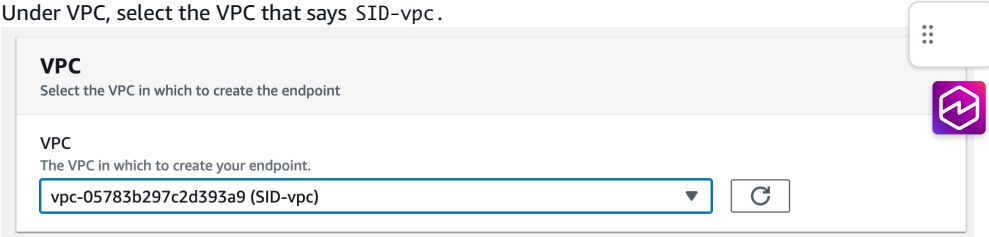
- Click `Endpoints` on the column to the left.
- Click `Create Endpoint`.
- Name your endpoint `SID-endpoint` .



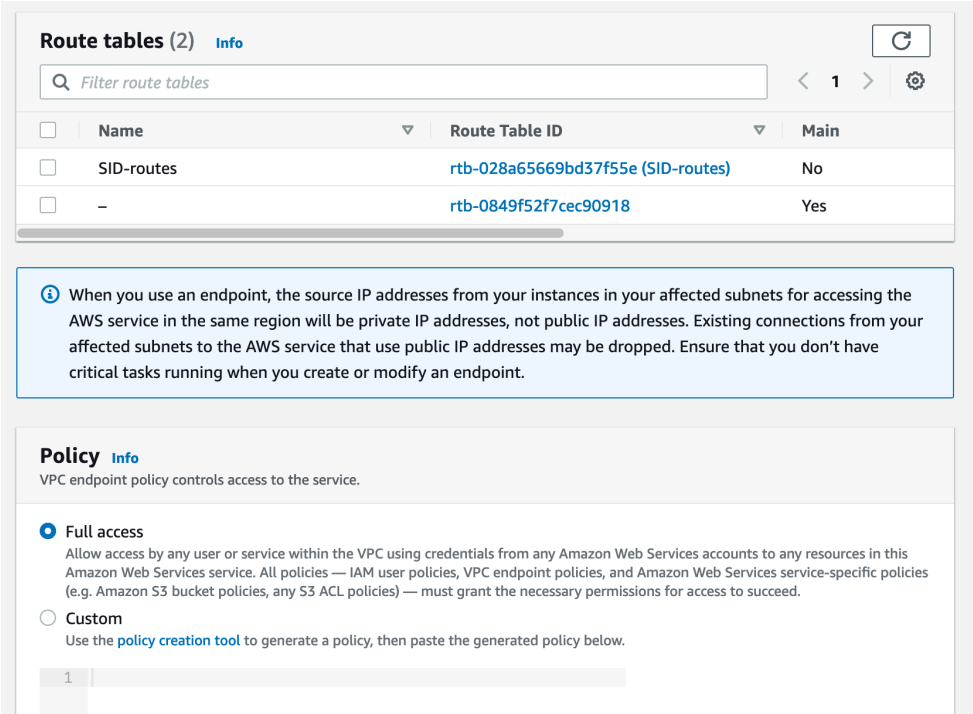
- Type `S3` in the search bar and press `enter`. This should filter to the S3 Endpoint. Select the `Gateway` type endpoint.



- Under VPC, select the VPC that says `SID-vpc`.

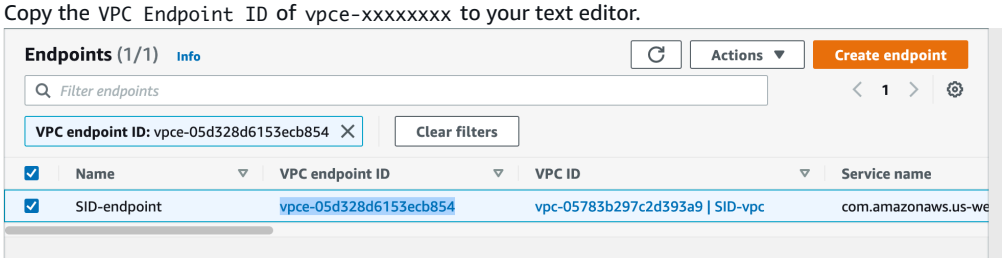


- Do not configure any route tables. Leave the Policy set to `Full Access`.



- Click `Create endpoint`.

Copy the `VPC Endpoint ID` of `vpce-xxxxxxxx` to your text editor.



From the AWS console in the top search bar, search and select `S3`.

- Click the bucket name starting with `sid-security-xxxxxxxx`.
- Click on the `Permissions` tab.
- Under `Bucket Policy` click `Edit`

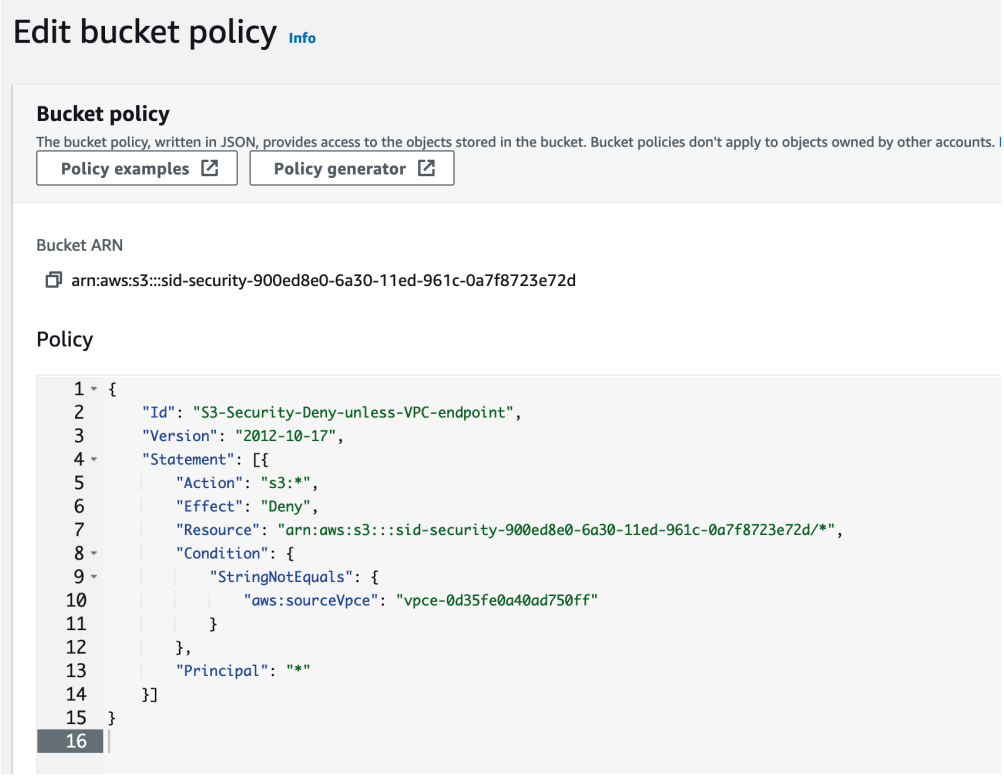
Delete the existing bucket policy. Copy the bucket policy below and paste into the Bucket Policy Editor.

```
{
    "Id": "S3-Security-Deny-unless-VPC-endpoint",
    "Version": "2012-10-17",
    "Statement": [{
        "Action": "s3:*",
        "Effect": "Deny",
        "Resource": "arn:aws:s3:::BUCKET_NAME/*",
        "Condition": {
            "StringNotEquals": {
                "aws:sourceVpce": "VPC_ENDPOINT_ID"
            }
        },
        "Principal": "*"
    }]
}
```

Replace `BUCKET_NAME` with the bucket name and `VPC_ENDPOINT_ID` with the Endpoint ID.

> Make sure you keep the /* at the end of the bucket name.

You policy should look similar to the below policy.



Click `Save Changes`.

From you SSH session run the following command.

```
aws s3api head-object --key app1/file1 --bucket ${bucket}
```

Curious, even though we've created our VPC endpoint and updated our S3 bucket policy to allow requests from the VPC endpoint `vpce-xxxxxxxx` the S3 head request fails.

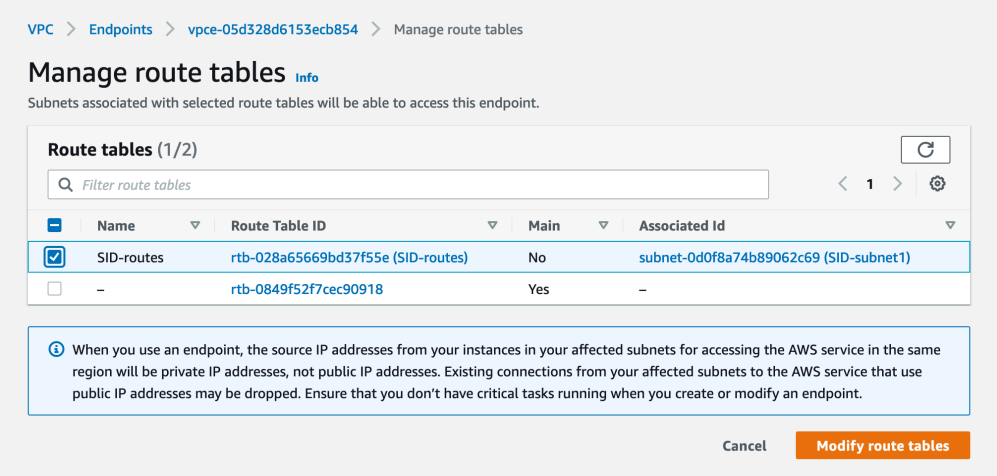
```
[ec2-user@storage-workshop ~]$
[ec2-user@storage-workshop ~]$ aws s3api head-object --key app1/file1 --bucket ${bucket}

An error occurred (403) when calling the HeadObject operation: Forbidden
[ec2-user@storage-workshop ~]$
```

Why is this so?

The reason for this is because our EC2 instance is accessing S3 via the internet gateway. This is because the VPC endpoint does not yet have a route table association. We will need to associate a route table to our newly created VPC endpoint so our EC2 instance can access S3 via the VPC endpoint instead of the internet gateway.

From the AWS console in the top search bar, search and select `VPC`.

- Click `Endpoints` on the column to the left.
- Select the VPC Endpoint `SID-endpoint` we created early. Click `Actions` and select `Manage Route Tables`.
- Select the Route Table ID named `SID-routes`.



Click `Modify Route Tables`.

From you SSH session run the following command.

```
aws s3api head-object --key app1/file1 --bucket ${bucket}
```

The request succeeds because the EC2 instance is able to find a route its S3 request via the VPC Endpoint and because our S3 bucket policy permits requests via the VPC endpoint.

```
[ec2-user@storage-workshop ~]$ aws s3api head-object --key app1/file1 --bucket ${bucket}
{
    "AcceptRanges": "bytes",
    "ContentType": "binary/octet-stream",
    "LastModified": "Sun, 08 Oct 2023 21:10:16 GMT",
    "ContentLength": 1048576,
    "ETag": "\"b6d81b360a5672d80c27430f39153e2c\"",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
[ec2-user@storage-workshop ~]$
```

Previous Next