**Design and Implement Gateway VPC Endpoints**

Created by[Zeal Vora](https://deloittedevelopment.udemy.com/user/cybercorp/)

**AWS Workspace**

**60-120 minutes**

In this lab, you will be a consultant at an organization that provides Payment Gateway services to its customers. They intend to store sensitive customer information like card information and various others. One of the prerequisites is to be compliant with regional security compliance. As part of the initial audit, the auditors have recommended that subnets hosting EC2 instances that are responsible for storing sensitive data be blocked from accessing the Internet (both inbound and outbound connections). These EC2 instances regularly communicate with Amazon S3 to perform backup-related operations of its data. The architecture group thinks that using the VPC Gateway Endpoints will allow the communication to S3 service even if the internet connectivity is removed at a subnet level. Your boss wants you to build a proof-of-concept for this solution.

The Engineering team has raised an issue. The application that will be hosting the sensitive data is in a private subnet and regularly communicates with the AWS S3 service for backup-related operations. If the NAT Gateway is removed from the subnet, all the necessary communication to S3 will be blocked, which can hamper the BCP/DR processes. I want you to create a POC using Gateway Endpoints, allowing an EC2 instance to backup data to S3 bucket even if the Internet connectivity (inbound and outbound) is blocked.

The Gateway VPC Endpoint should only allow the backup-related (upload) operations to a specific S3 bucket. No other operations should be allowed. The S3 bucket should also have an appropriate policy that will only allow connections to it from the Gateway Endpoint. No other communications should be allowed. For the POC, you can use the default VPC subnet to implement the new architecture.

### Tasks

1. Modify VPC subnet for private connectivity
2. Create gateway VPC endpoint
3. Create S3 Bucket
4. Create a custom Gateway endpoint policy
5. Create S3 Bucket with bucket policy
6. Create temporary PEM rivate key for ec2
7. Launche ec2 to test endpoint connectivity
8. Verify connectivity between ec2 and s3 via endpoind
9. Clean Up

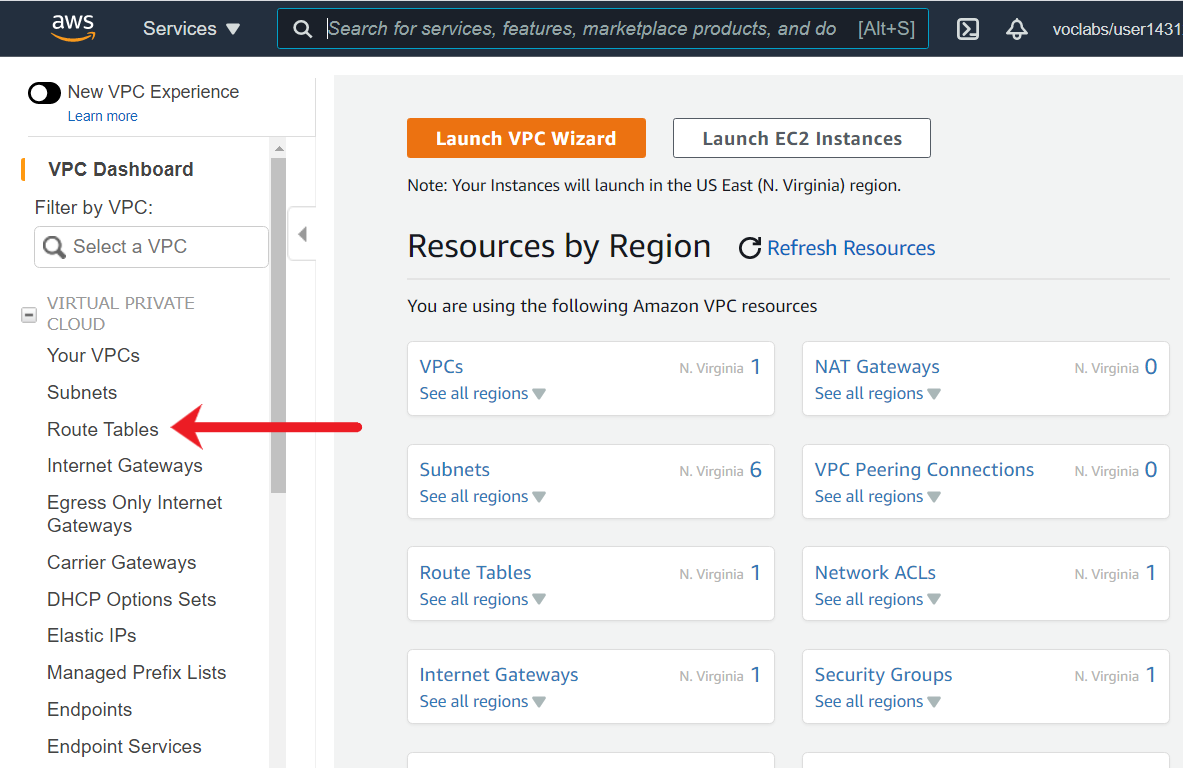
Modify VPC subnet for private connectivity

You need to modify one of the subnets in the default VPC to make it suitable for private-only connections. This subnet should not have any inbound or outbound internet connectivity.

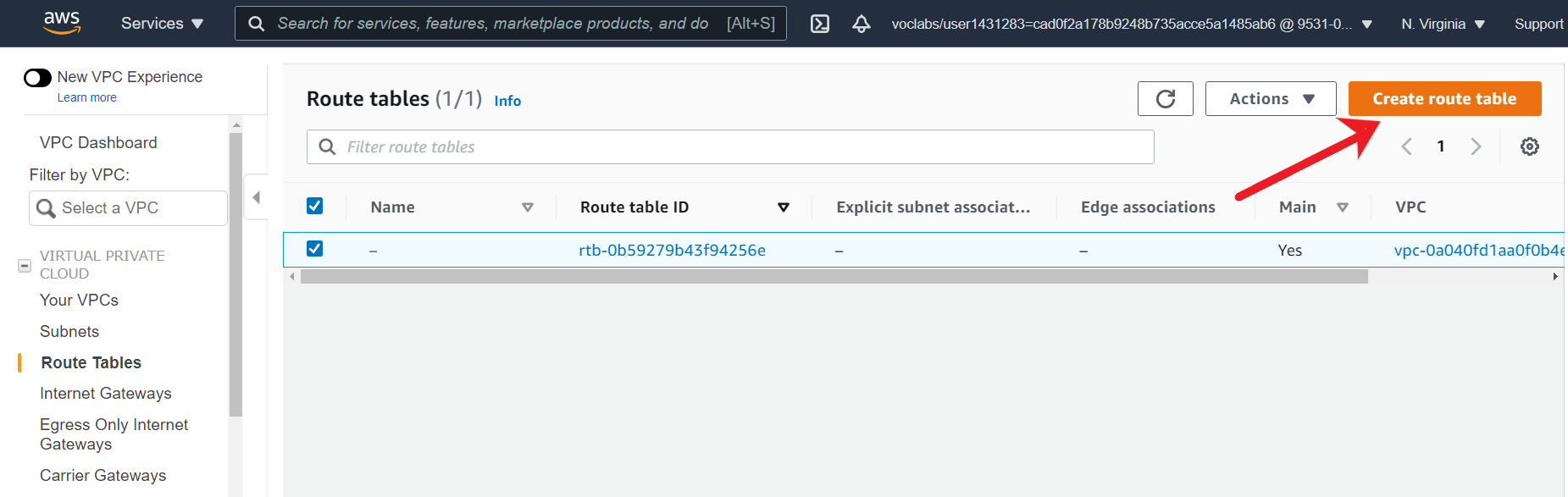
1. Open the VPC console by searching for VPC.



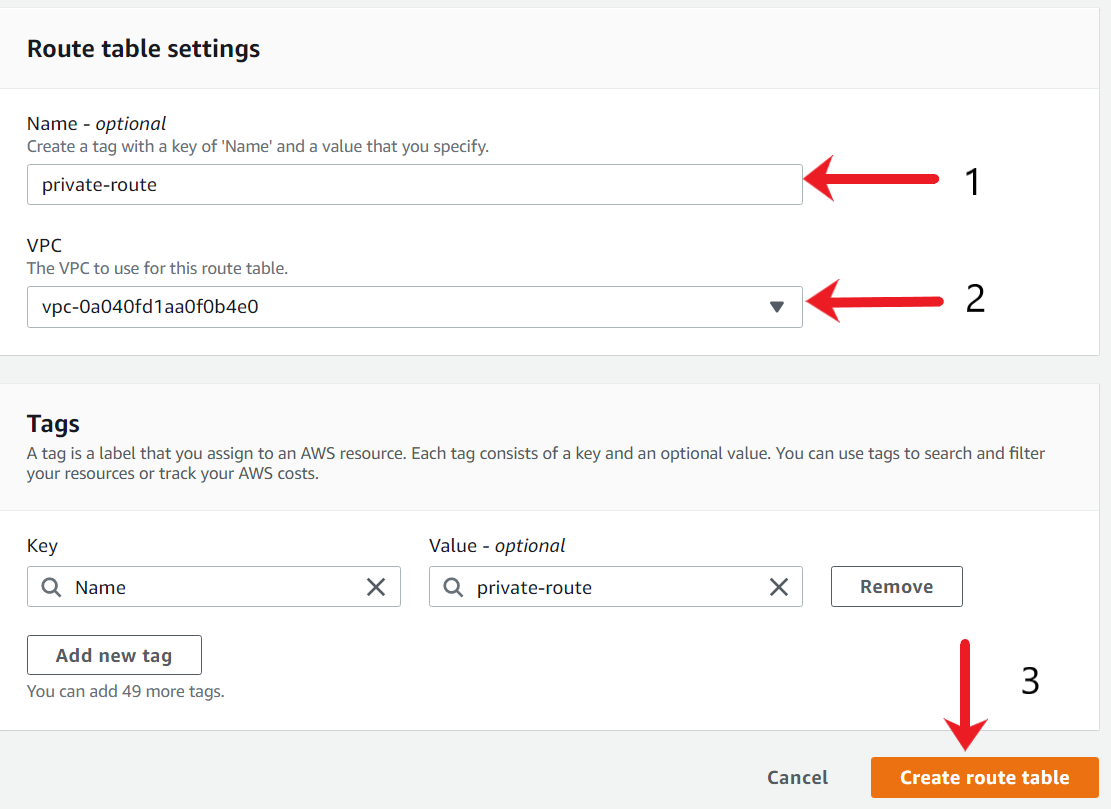
1. In the navigation pane, choose “Route Tables”.



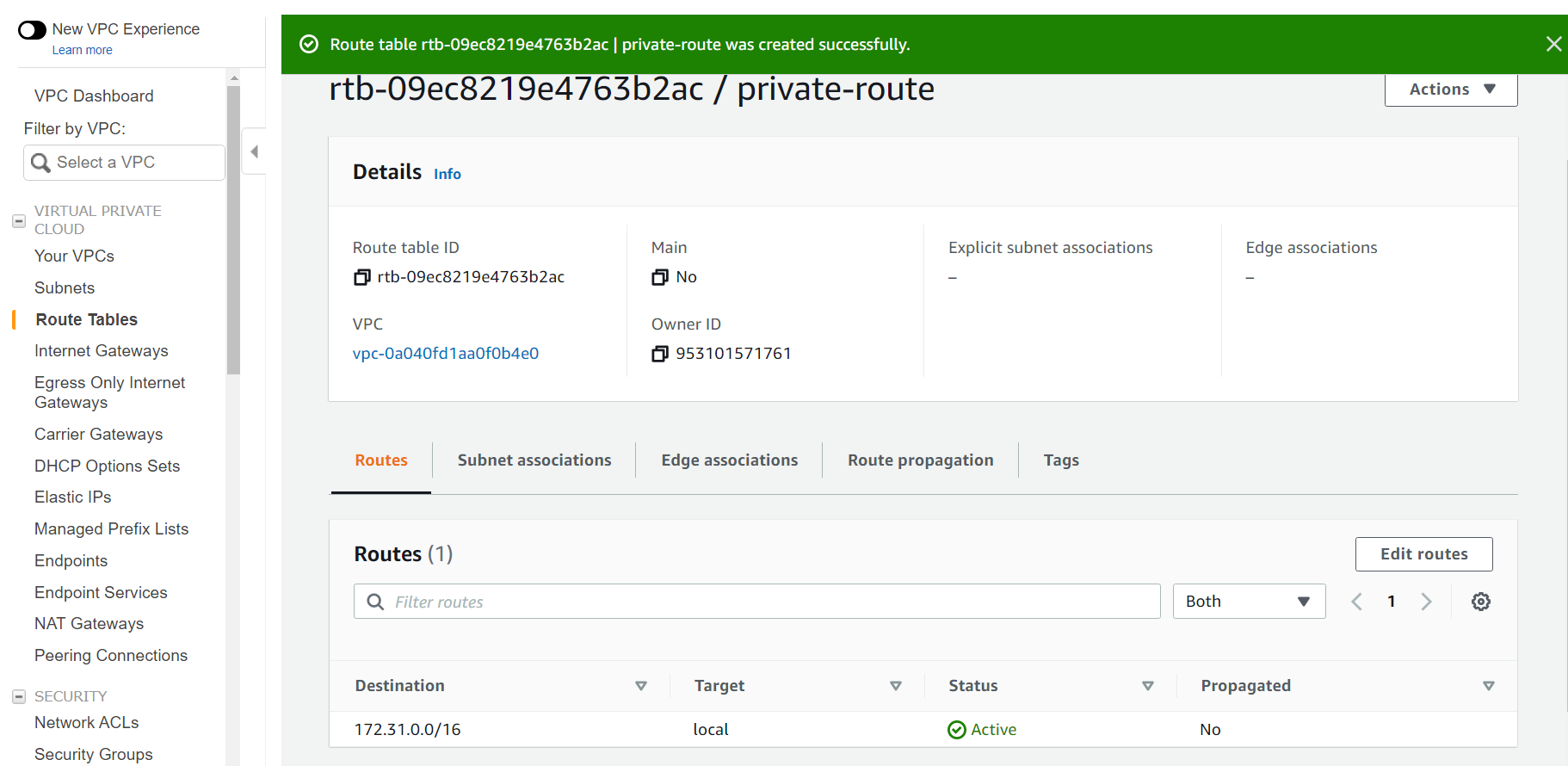
1. Choose **Create route table**.



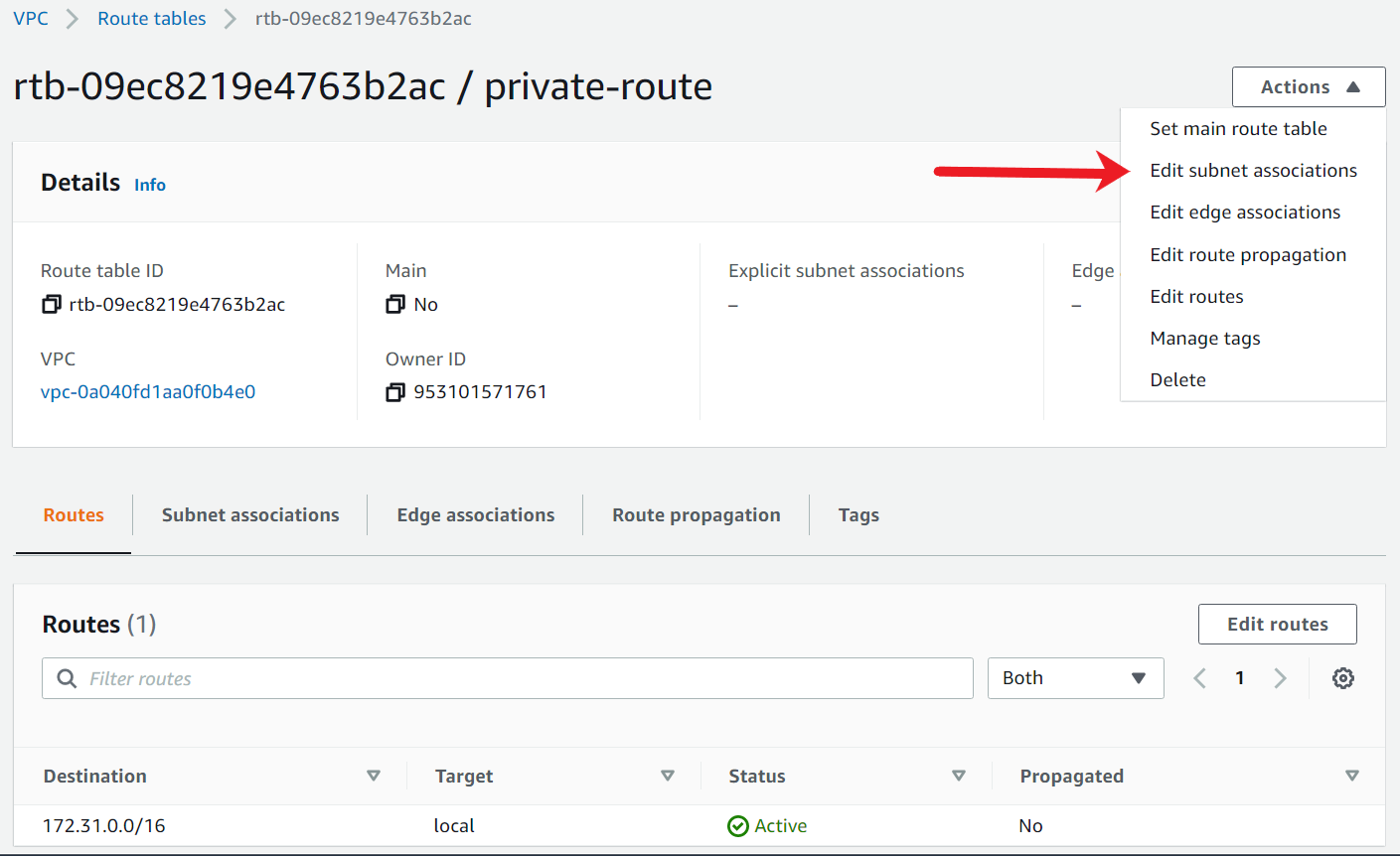
1. For Name tag, enter a name for your route table. For VPC, choose the default VPC of the region. Choose Create.



After the route table is created, you will see a success message. By default, there will only be a local route.



1. To associate the route table to one of the subnets in order to make the subnet completely private without internet connectivity, click on **Actions**, and choose **Edit subnet associations**.



1. Select the check box for the subnet to associate with the route table, and then choose Save associations. For this example, I have chosen a subnet associated with the us-east-1b availability zone.



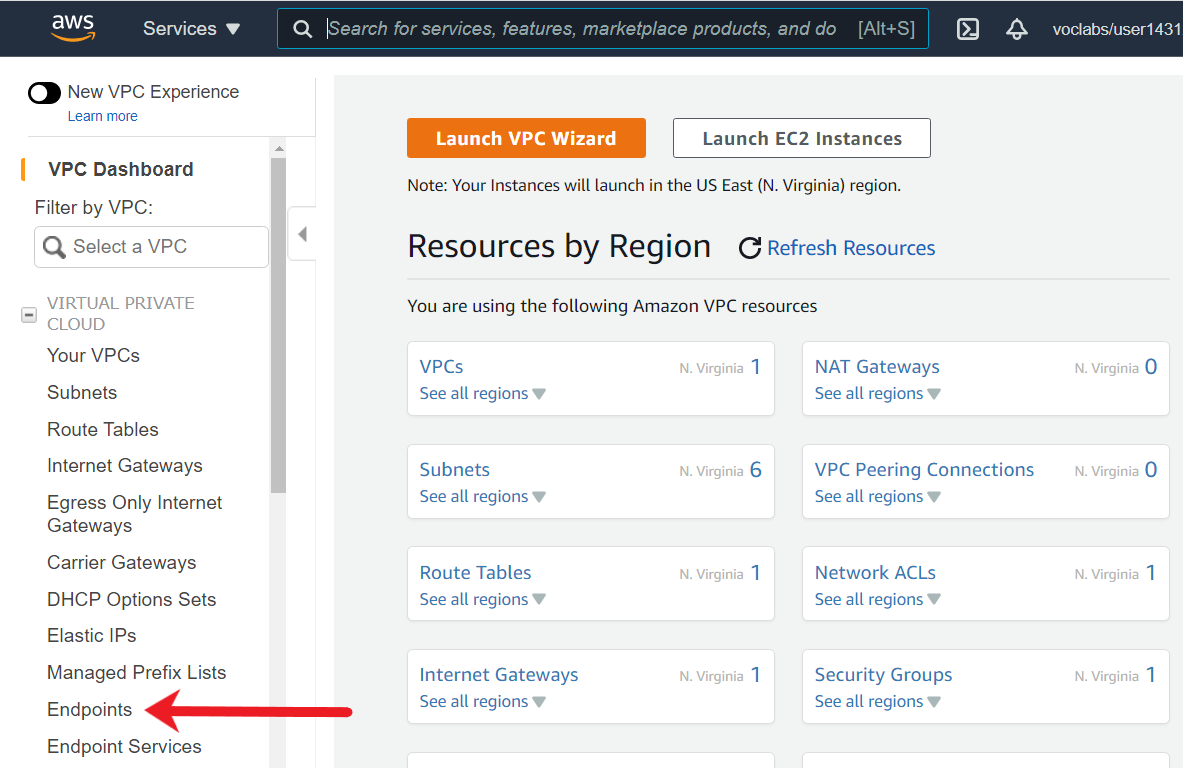
1. Verify if the route table is created from the main console. This route table should only have one local route and no connectivity towards the internet.

<https://docs.aws.amazon.com/vpc/latest/userguide/route-table-options.html>

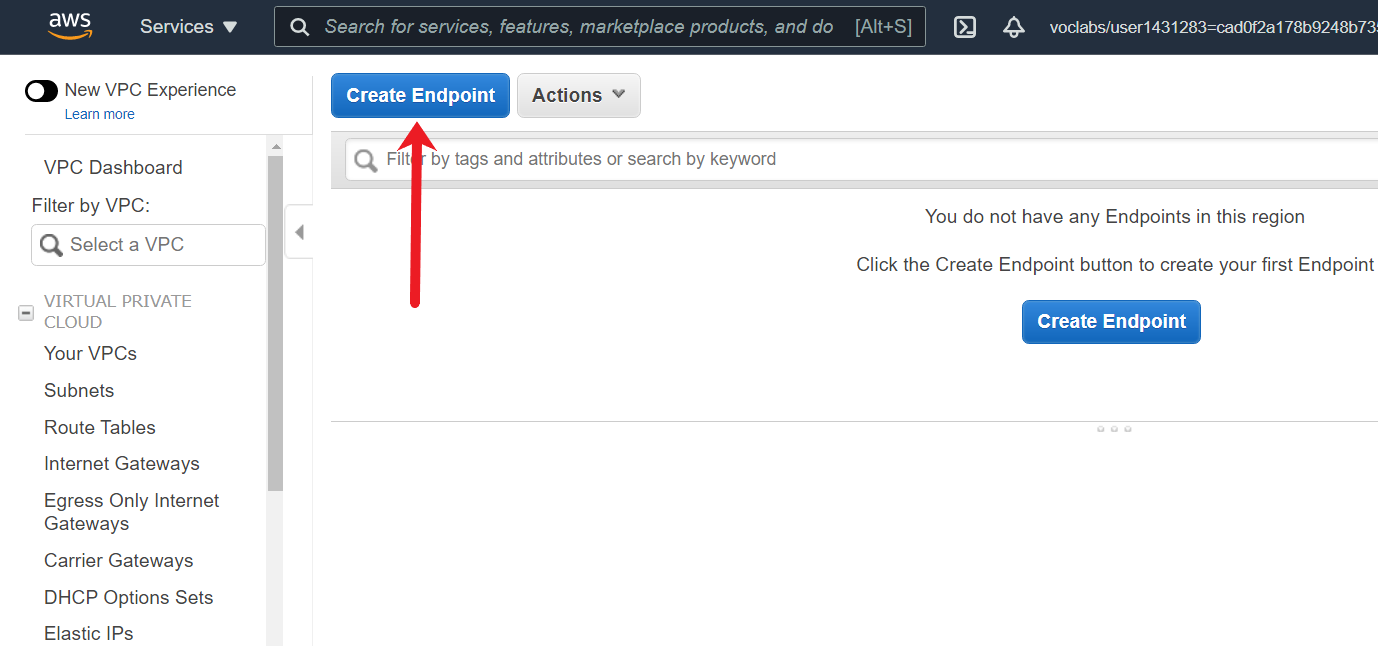
Create gateway VPC endpoint

You need to create a Gateway VPC Endpoint that can connect to the Amazon S3 service.

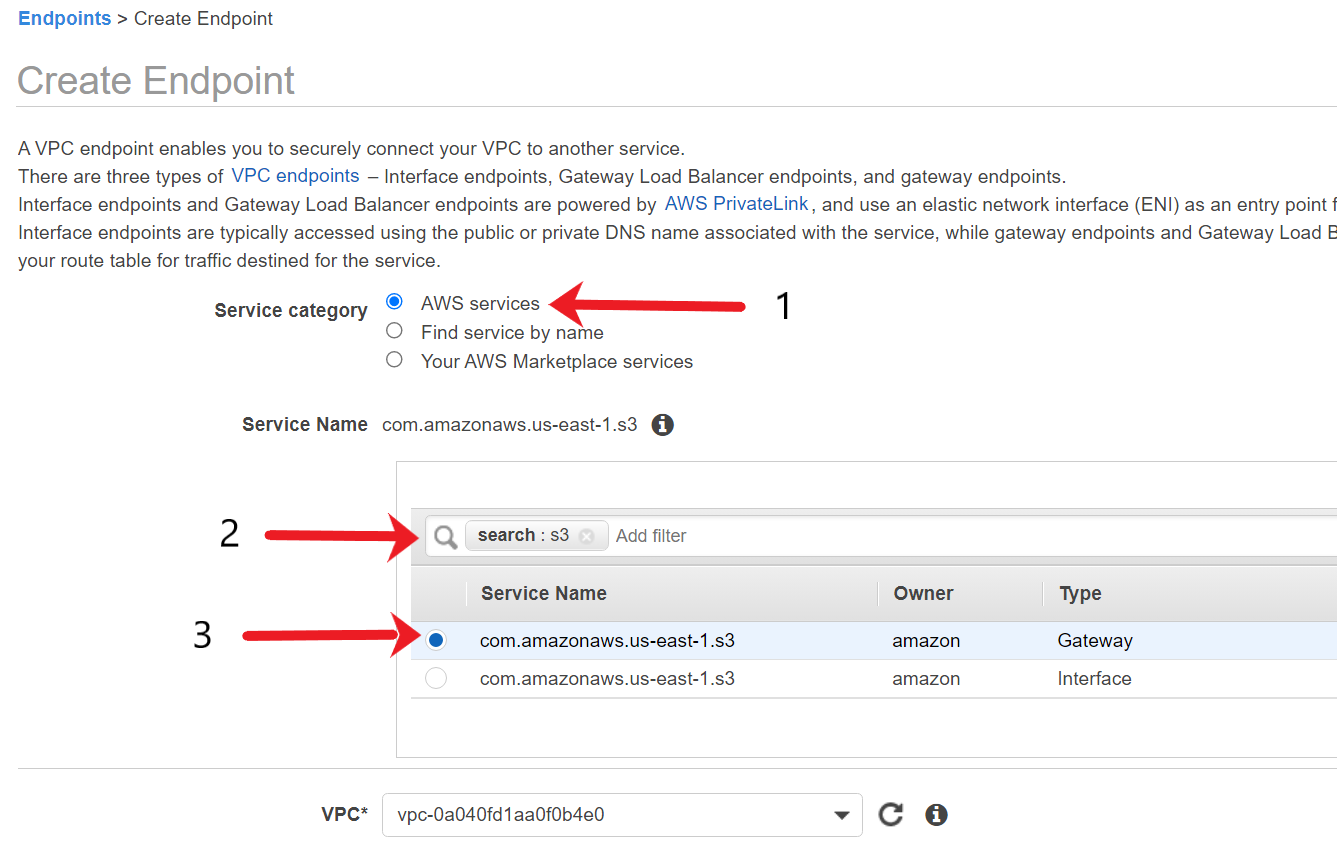
1. Under the VPC Console, In the navigation pane, choose Endpoints.



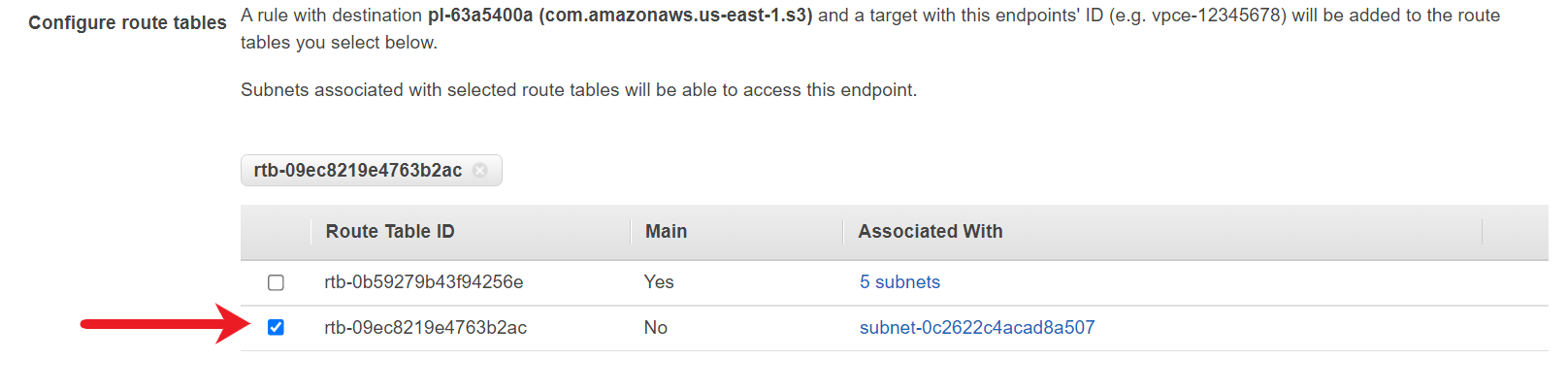
1. Choose **Create Endpoint**.



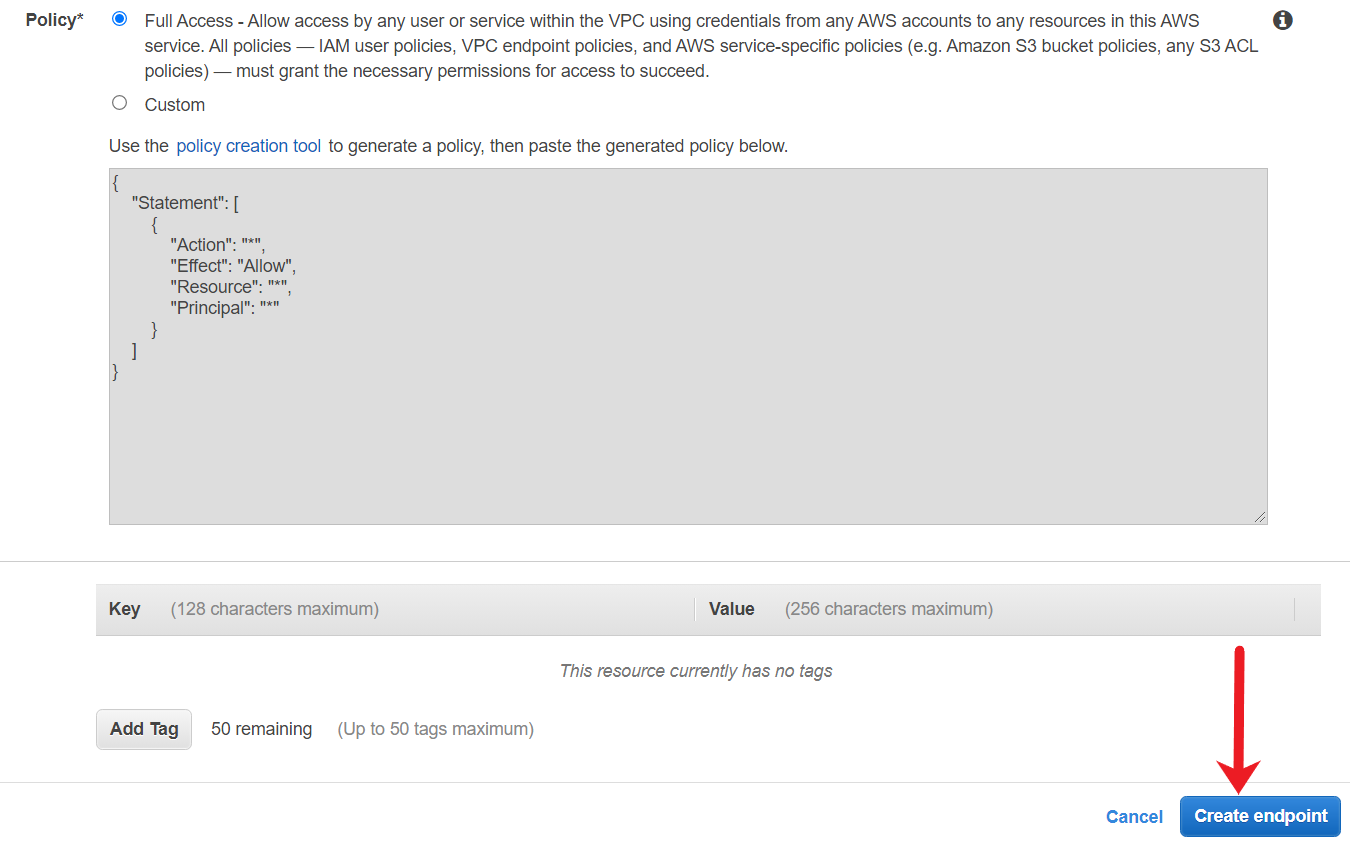
1. For Service Category, choose AWS services. In the filter, type S3 and select the service name of S3 associated with the Type of Gateway.



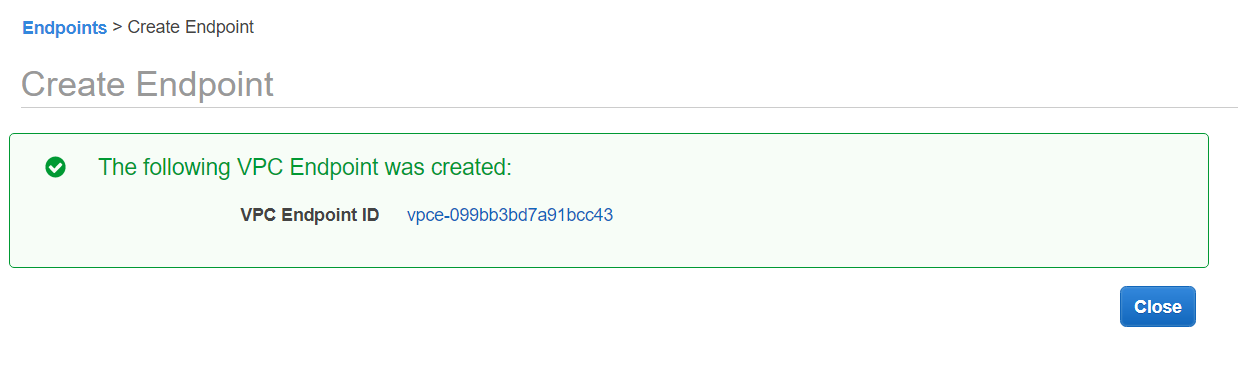
1. For the Choose Route tables, select the custom route table that you had created.

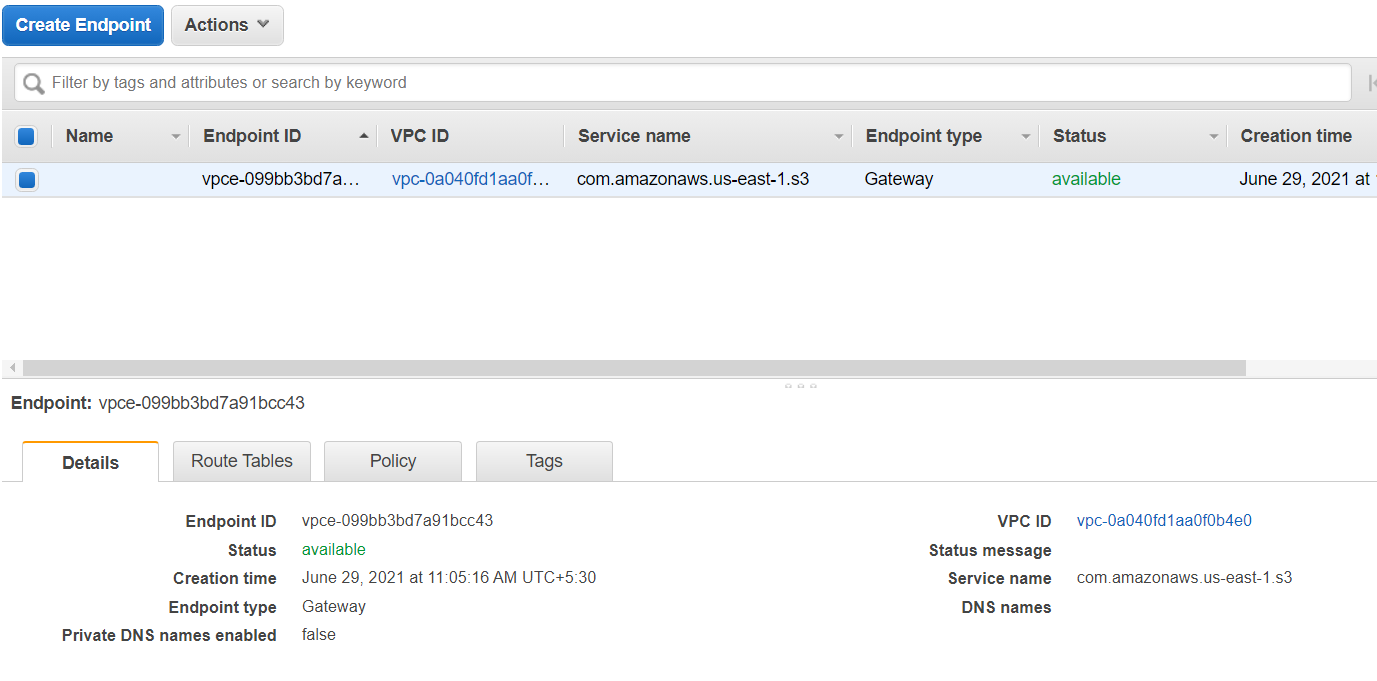


1. For the Policy, leave it as default for the time being and choose **Create endpoint.**



1. After the endpoint is created, you will get a success message. Click on Close and you should see your endpoint within the console.



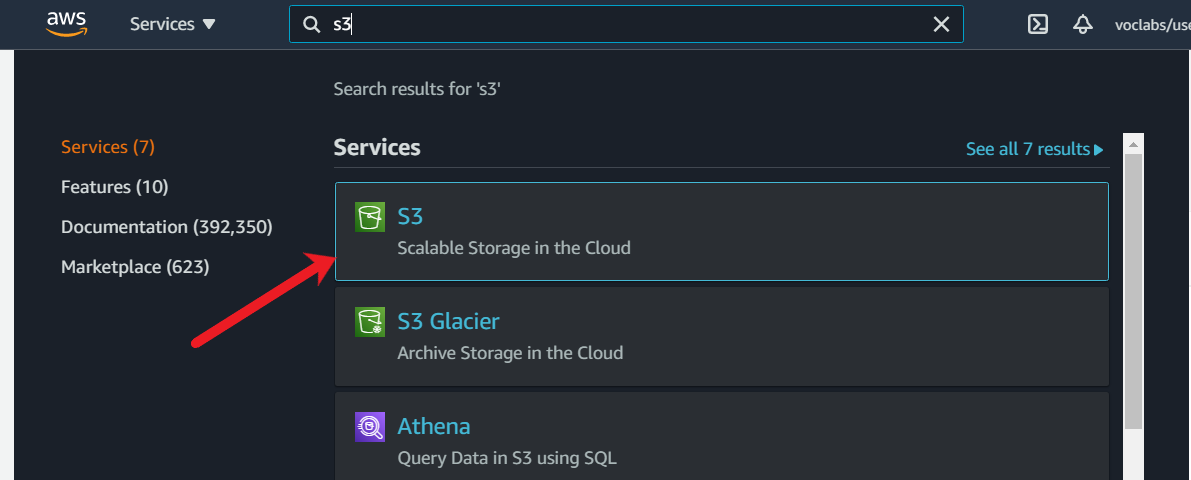


<https://docs.aws.amazon.com/vpc/latest/privatelink/gateway-endpoints.html>

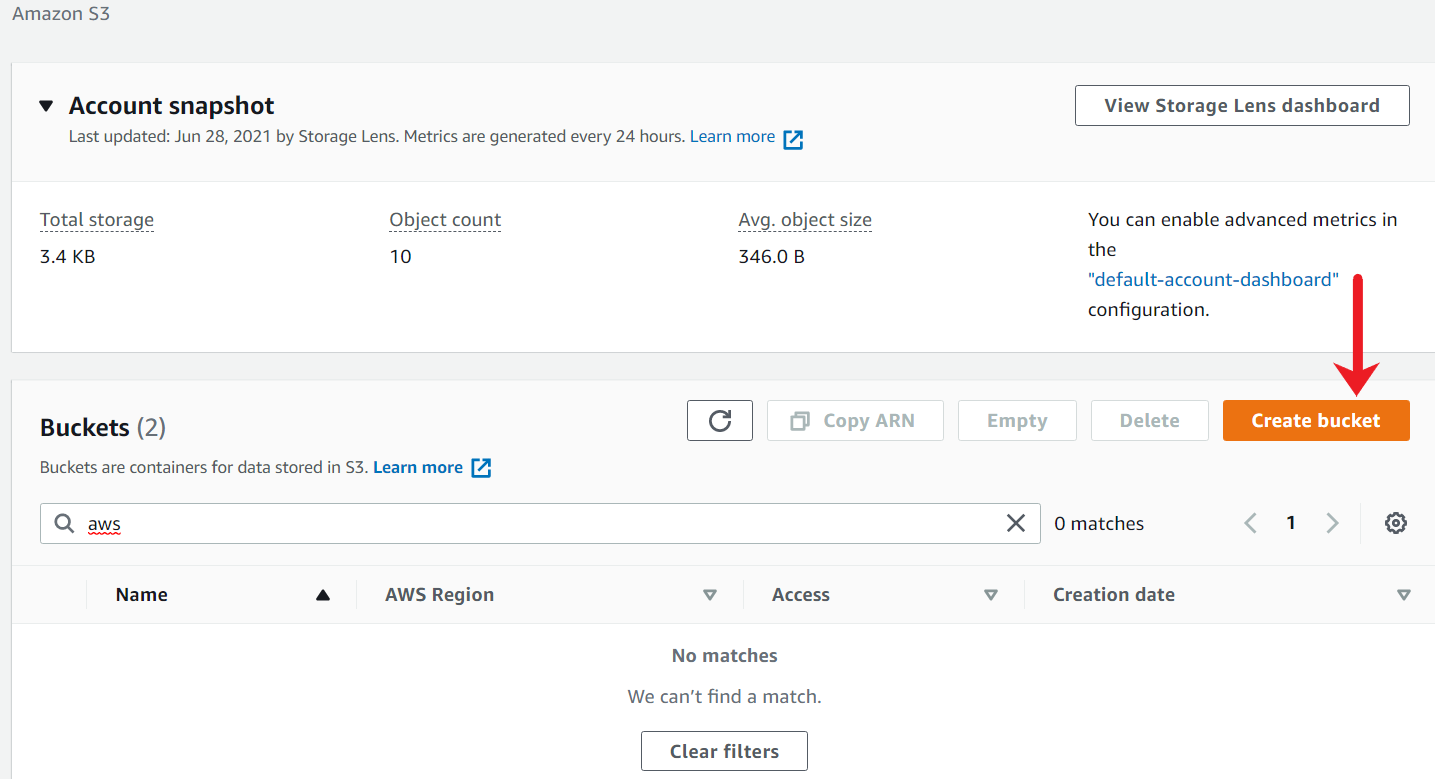
Create S3 Bucket

Create an S3 bucket in the same region as the VPC Endpoint. This bucket will be used to test the connectivity from VPC Endpoint and verify if the application can successfully backup its data.

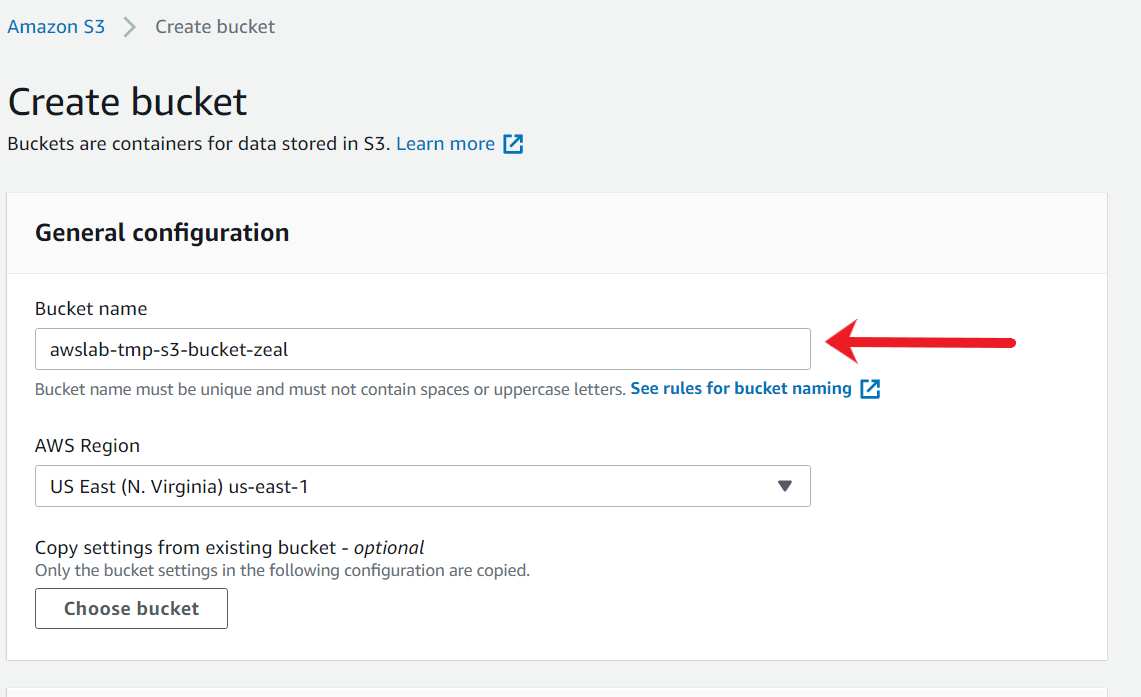
1. Open the S3 console by searching for S3.



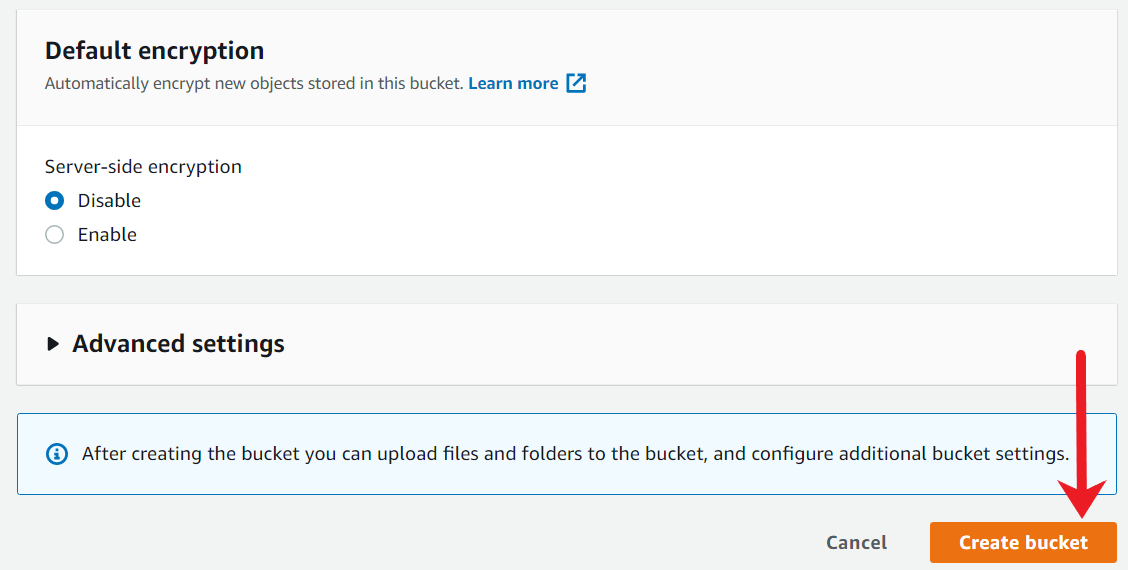
1. In the S3 console, choose Create Bucket.



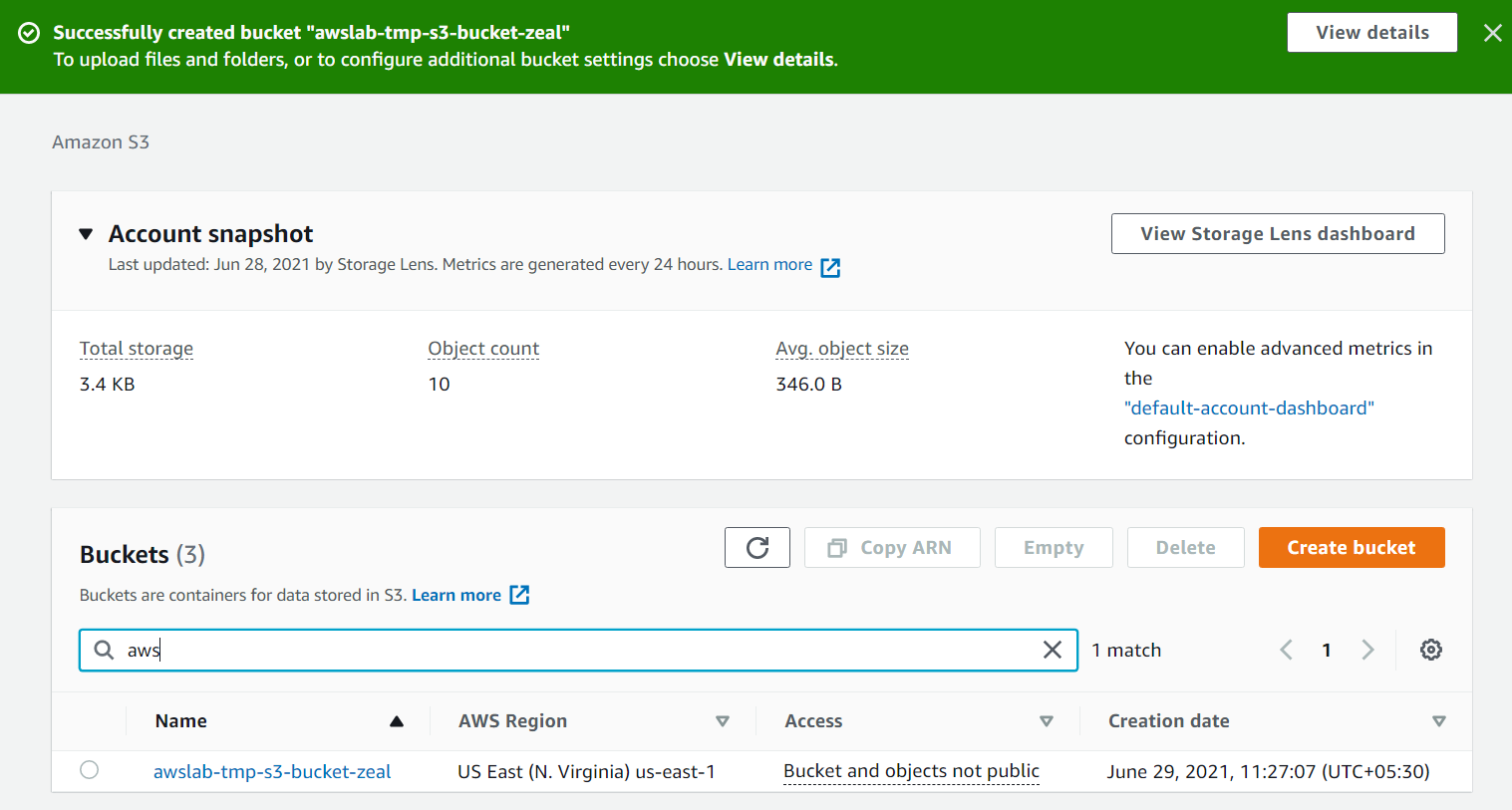
1. In Bucket name, enter a DNS-compliant name for your bucket. The region must be the same where the VPC Endpoint is created.



1. Scroll down to the bottom of the page and choose **Create Bucket**.



1. Once the bucket is created, you should see a success message along with the bucket name in the main S3 console page.

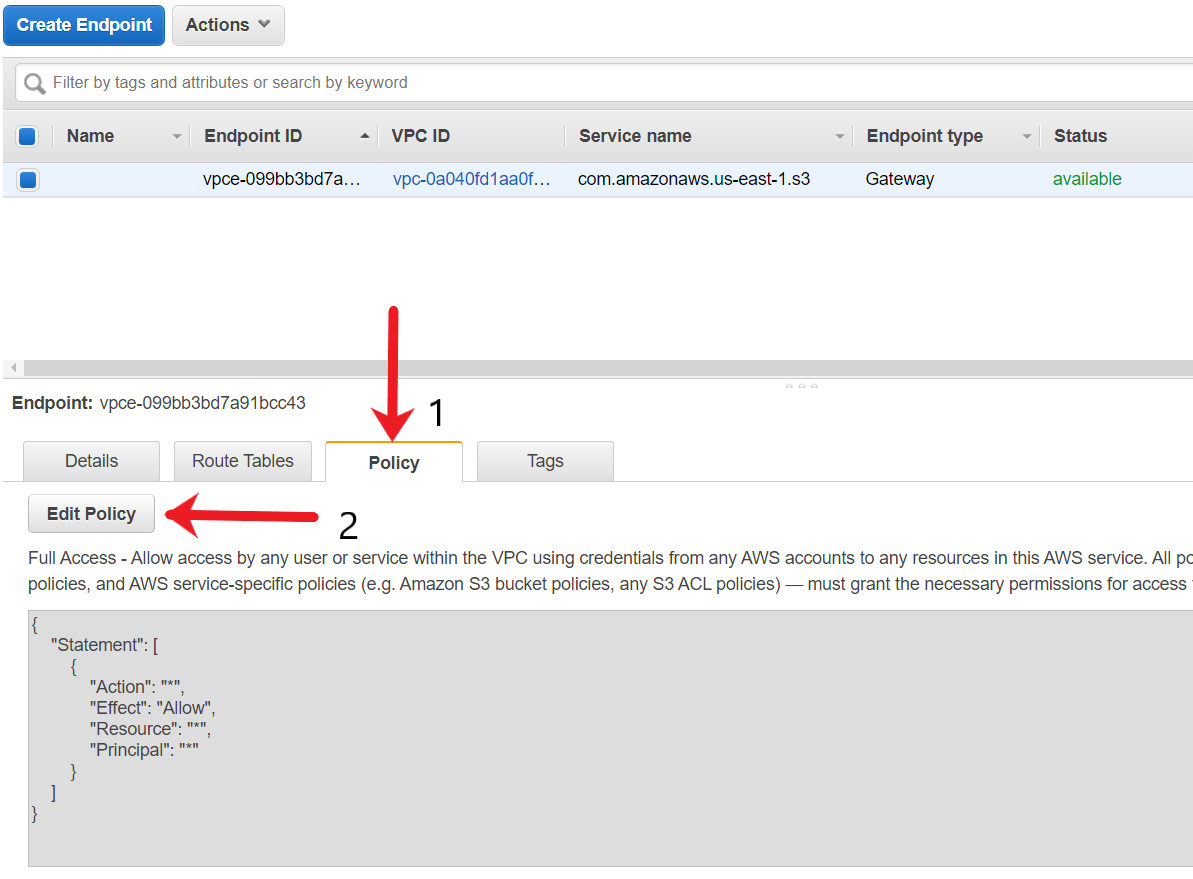


<https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingBucket.html>

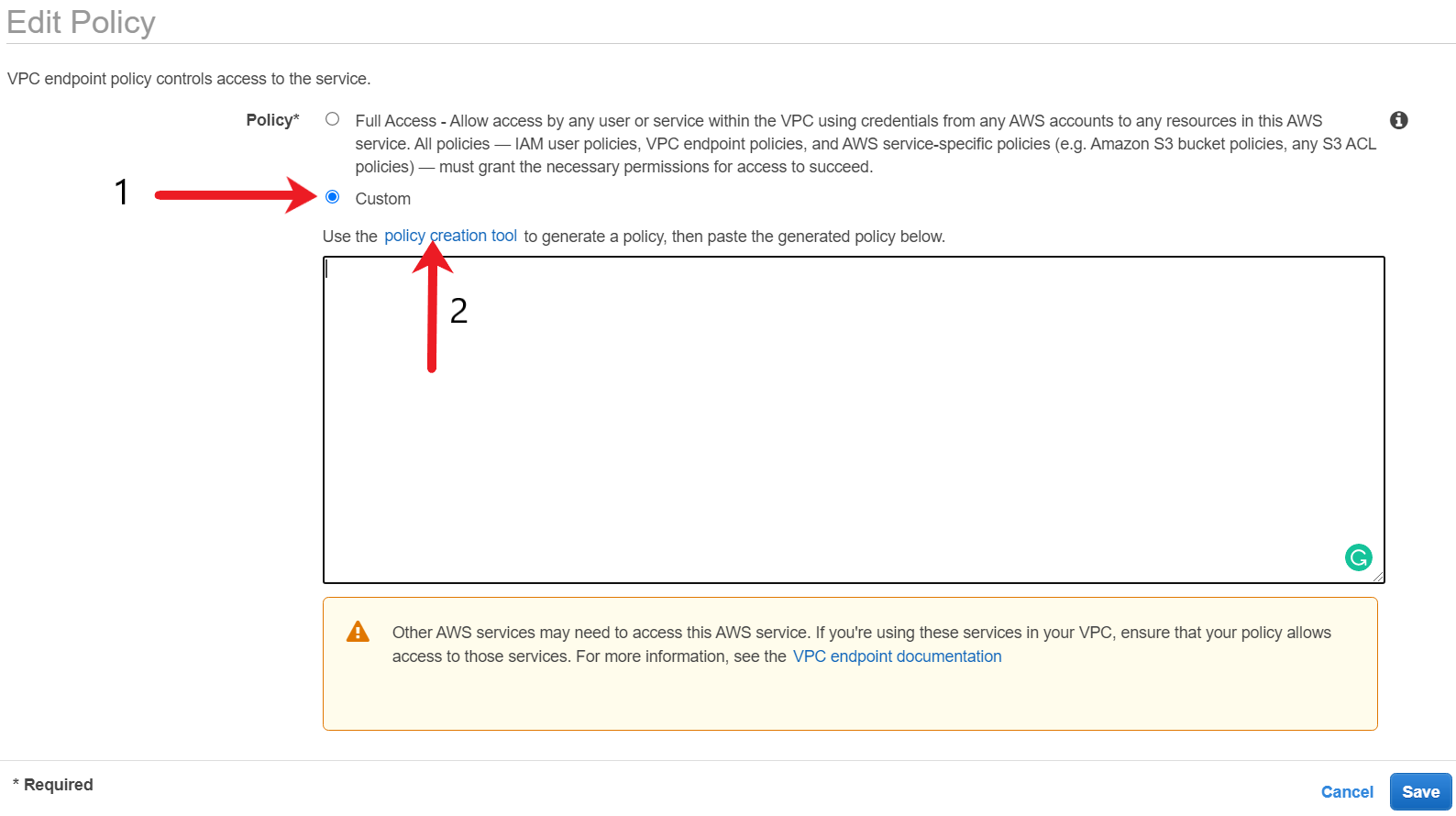
Create a custom Gateway endpoint policy

You need to create a Custom VPC Gateway Endpoint policy that will allow the application to perform its routine backup activity to store backups to the S3 bucket. No other operations should be allowed other than uploading the data to S3.

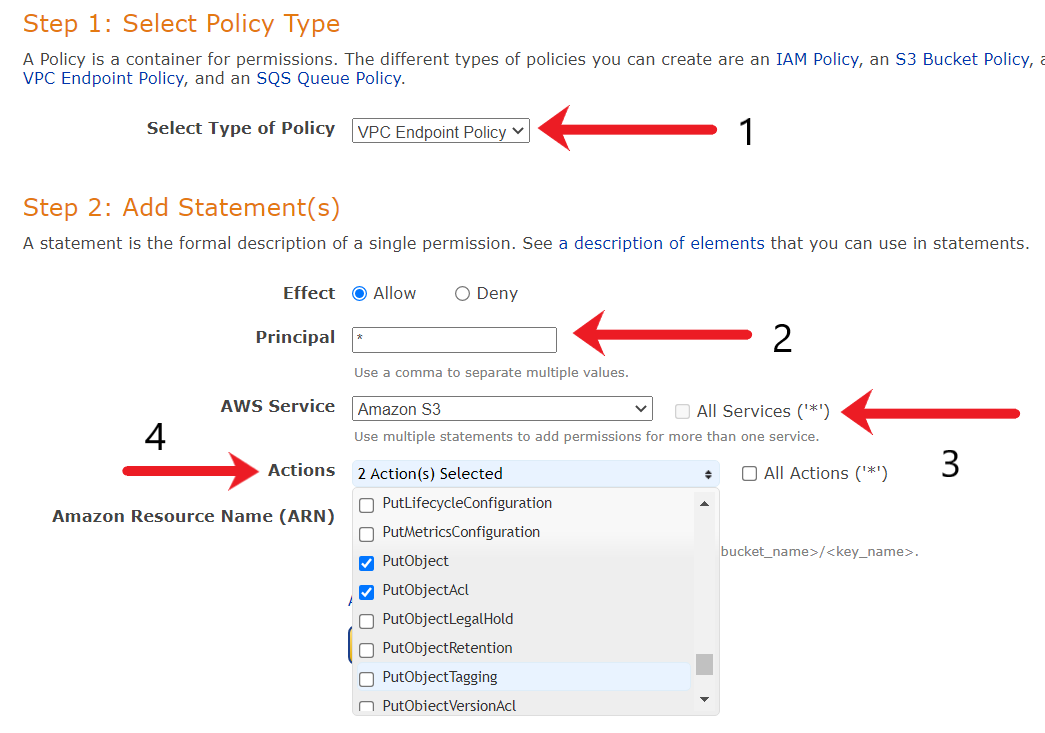
1. Within the Endpoints console, choose your endpoint and select the Policy tab and click on Edit Policy.



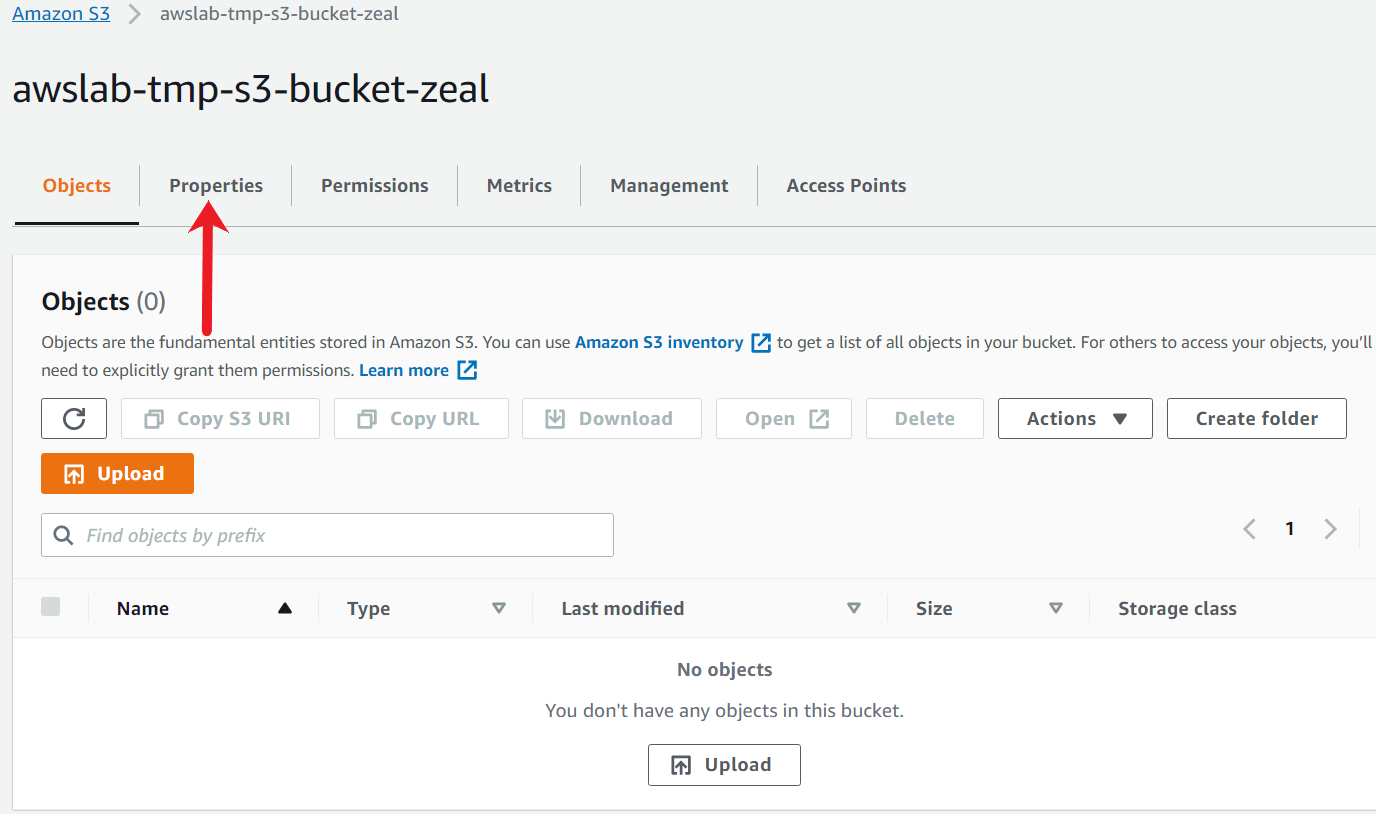
1. Select the Policy as **Custom**and click on **policy creation tool**.



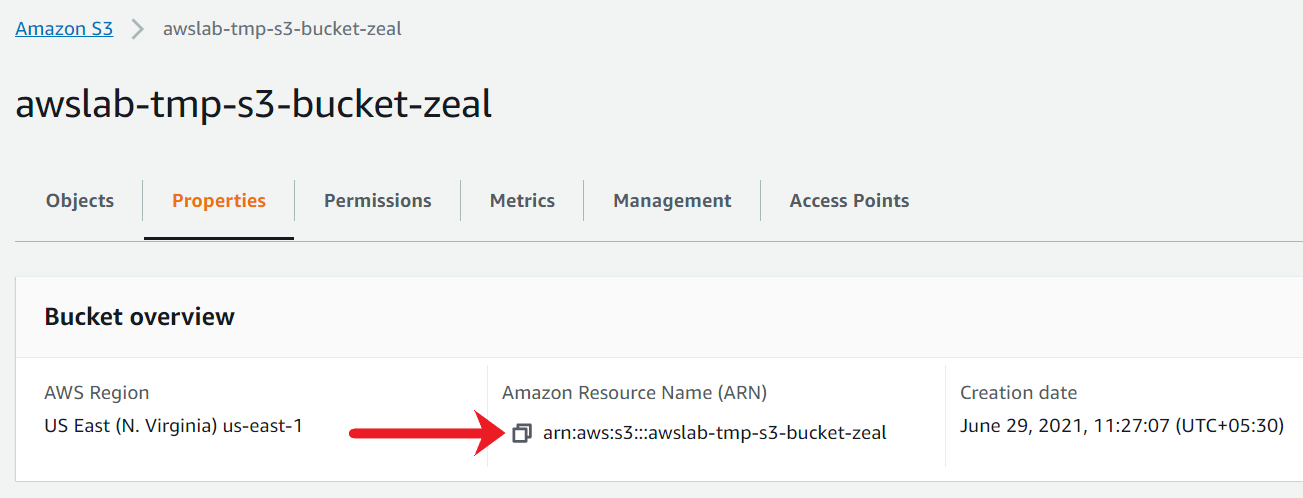
1. The policy type should be VPC Endpoint Policy. The principle should be \* . AWS Service should be Amazon S3 and Actions should be PutObject and PutObjectAcl.



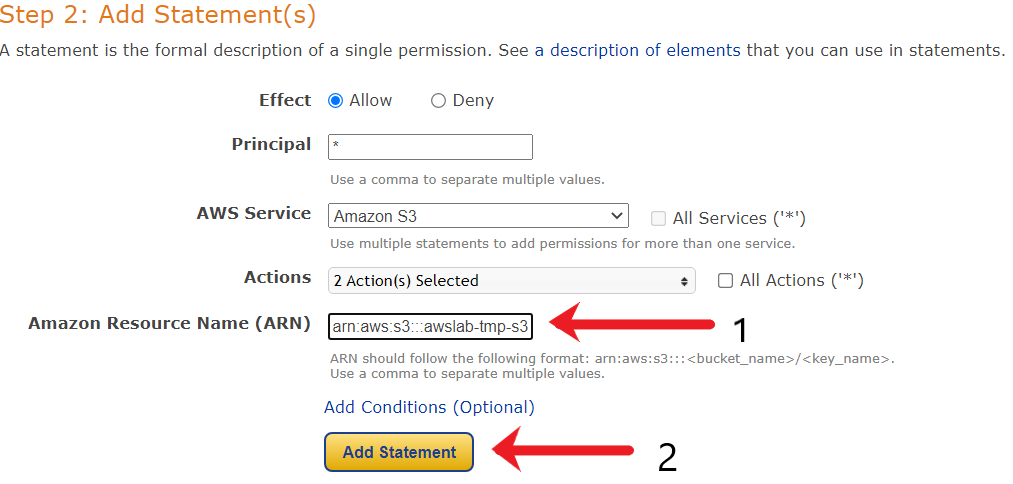
1. For the next step, we need to get the ARN of our S3 bucket. Open the S3 bucket that you created in a new tab and click on Properties tab.



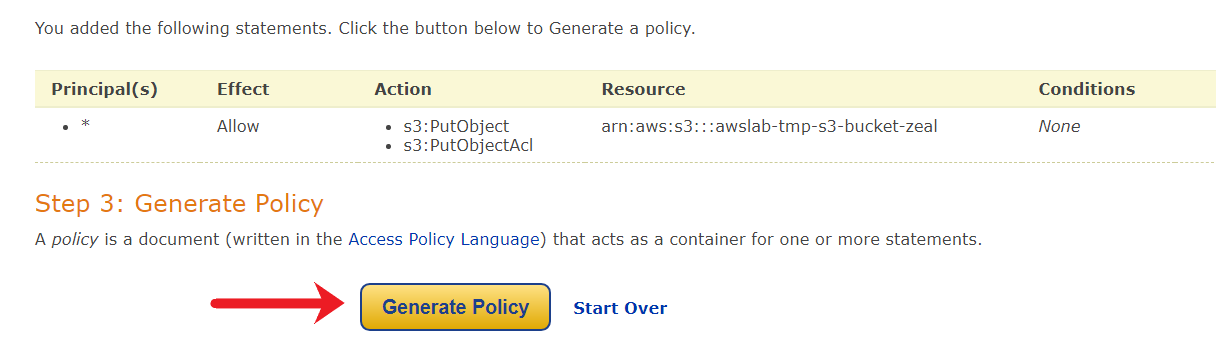
Copy the ARN of the S3 bucket.



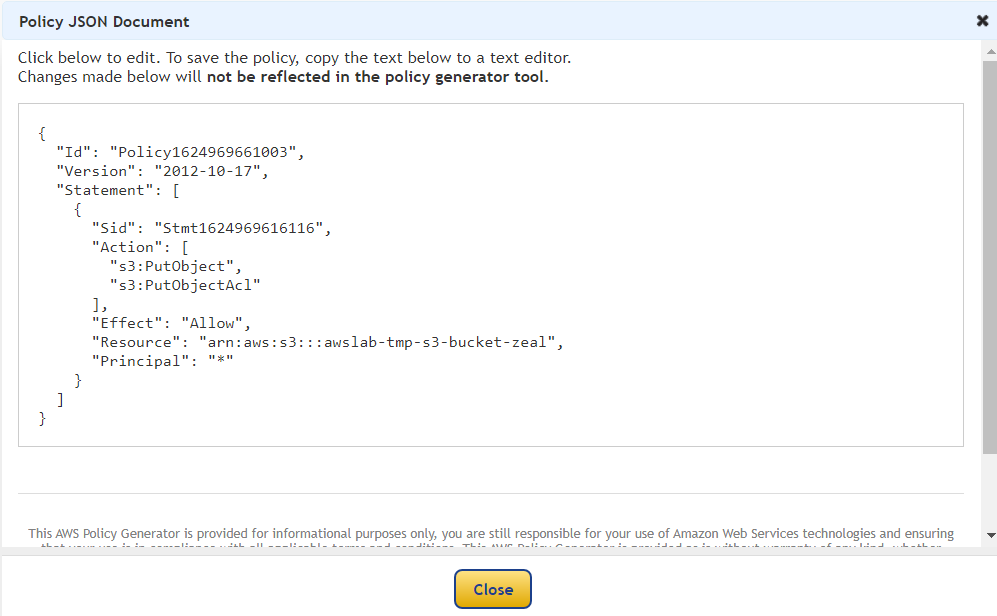
1. Copy the ARN of the S3 bucket in the ARN box and choose **Add Statement**.



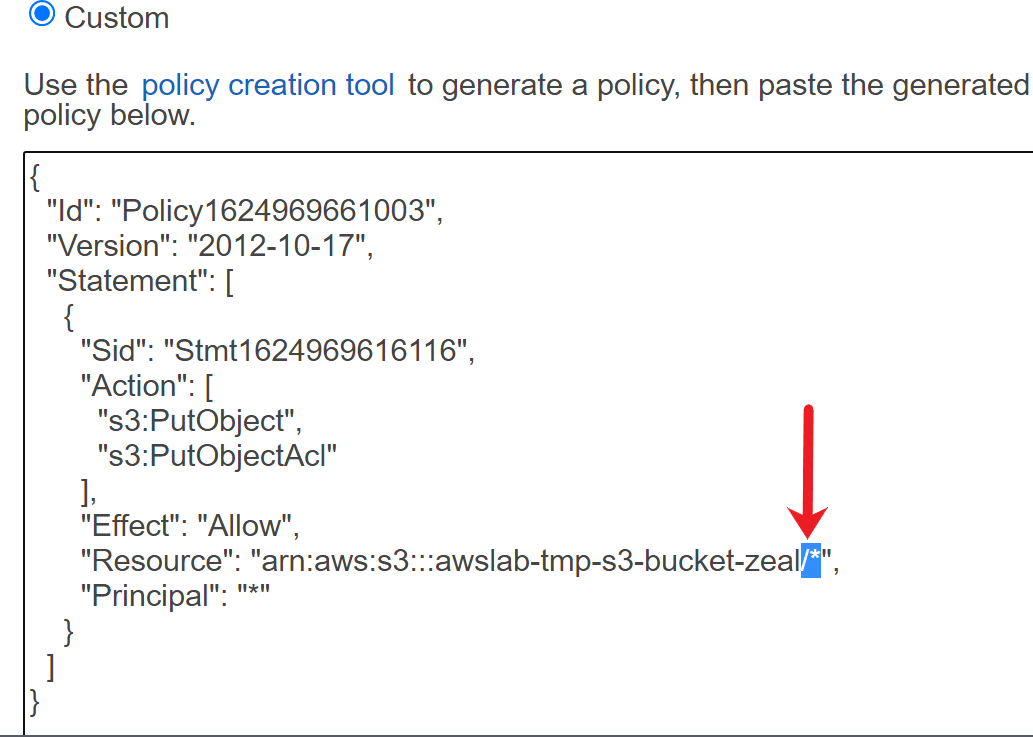
1. Scroll to the bottom of the page. Verify if the statement are as per your selection and choose **Generate Policy**.



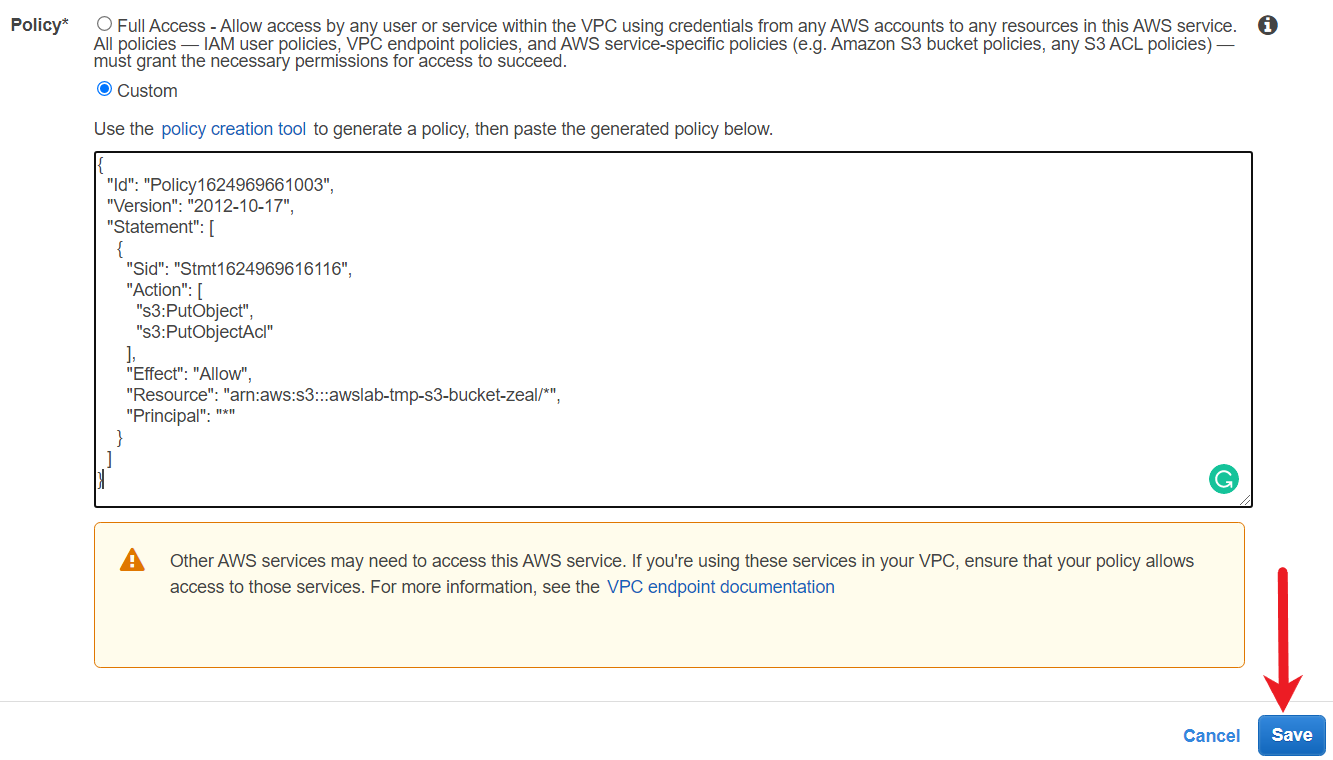
1. After choosing Generate Policy, an endpoint policy would be created.Copy the contents of the endpoint policy.



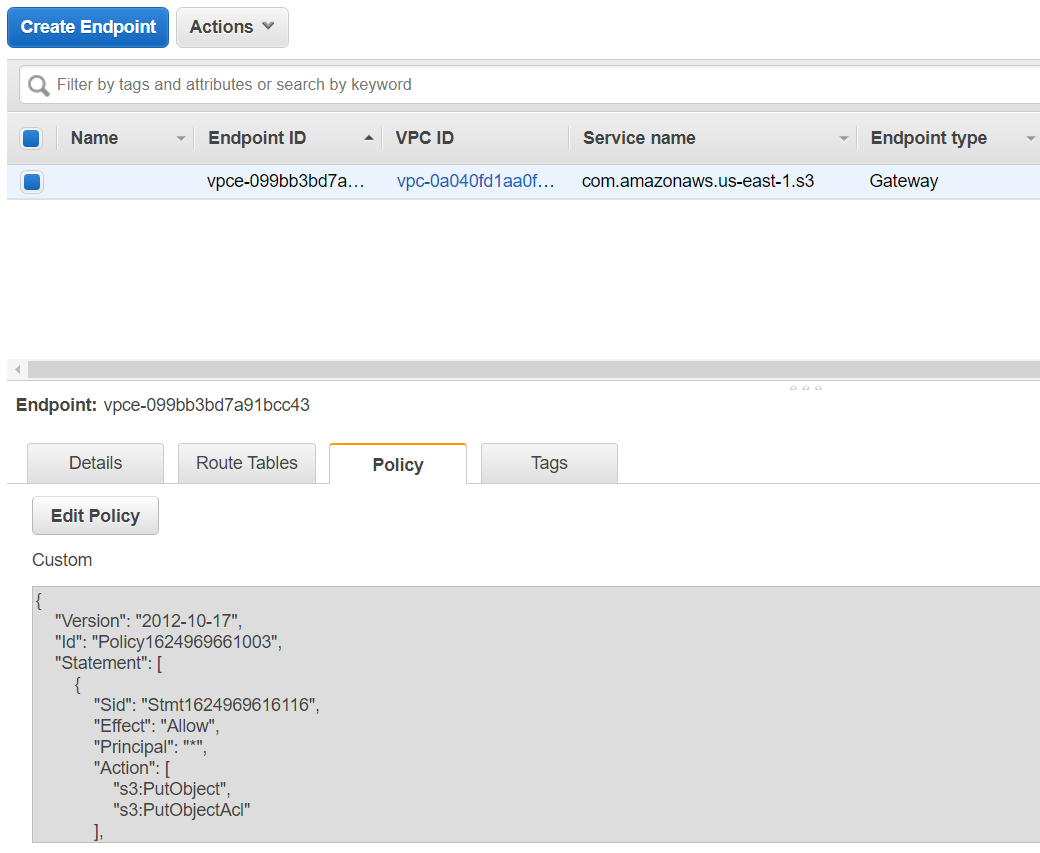
1. Within the Endpoint custom policy console, paste the custom policy. Make sure to add a /\* at the end of ARN.



1. Scroll to the bottom of the page and choose **Save**.



1. After the custom endpoint policy is created, you should see the updated policy within the VPC Endpoint Policy Tab.

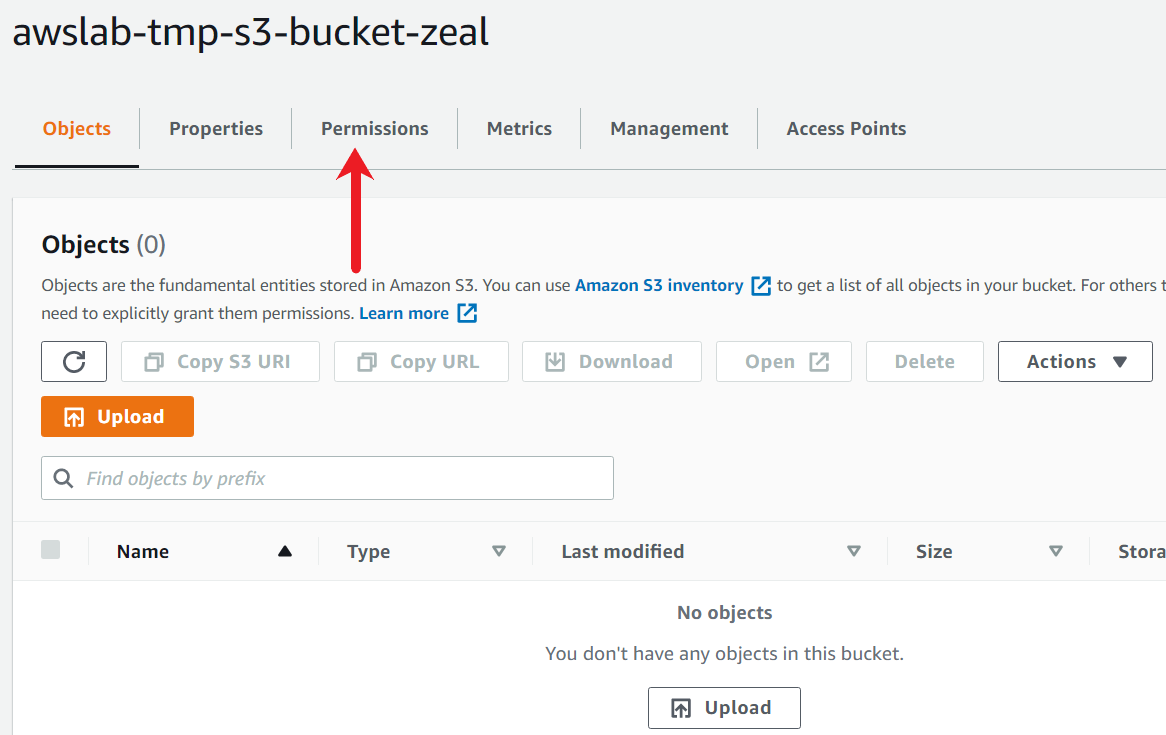


<https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints-access.html>

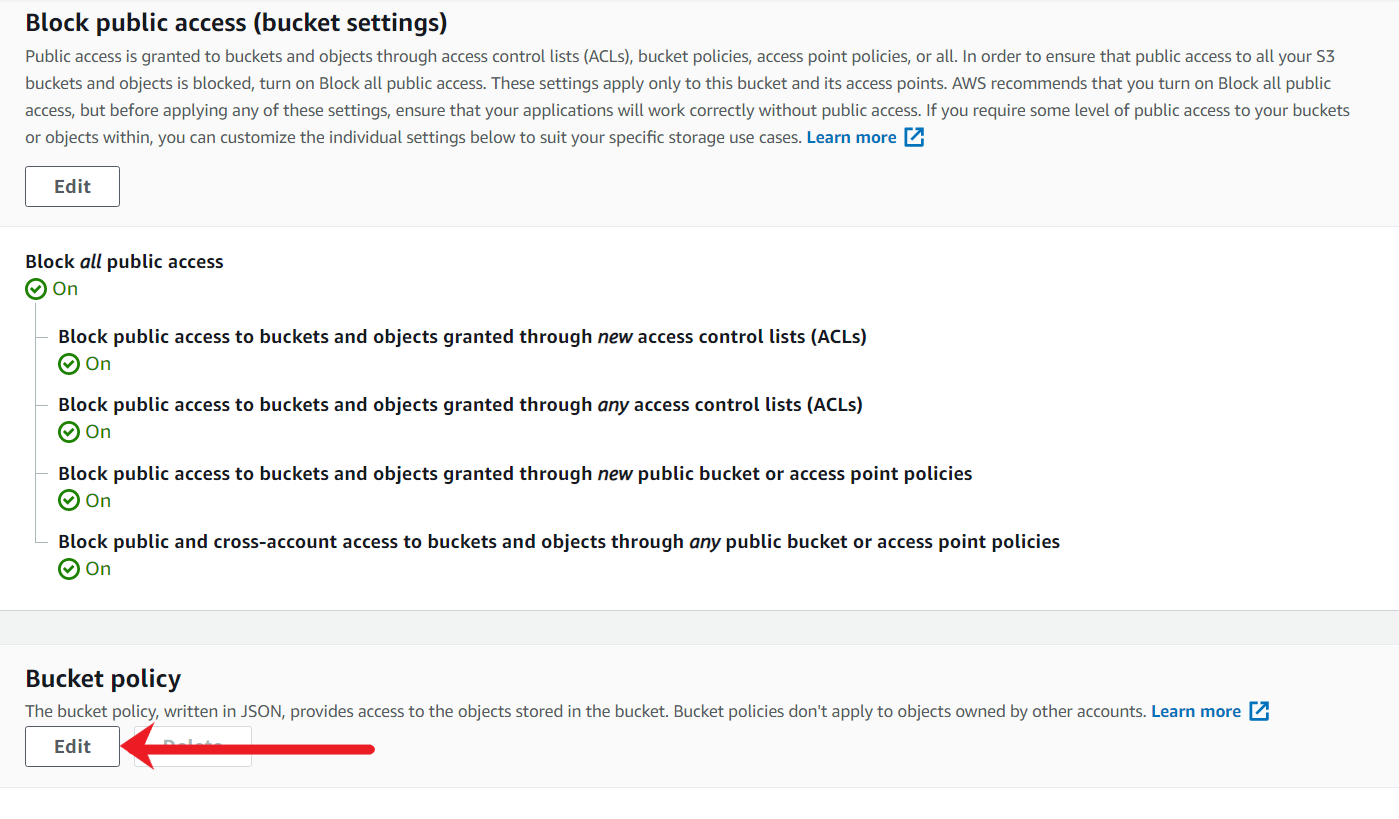
Create S3 Bucket with bucket policy

You need to create a custom S3 bucket policy that will only allow connections from the VPC Endpoint. All types of actions should be allowed from VPC Endpoints.

1. From the S3 console, choose the name of the bucket that you want to create a bucket policy.
2. Choose Permissions.



1. In the Bucket Policy, choose Edit.



1. In the Bucket policy editor text box, type or copy and paste a new bucket policy. The bucket policy is a JSON file. The policy in this example denies all access to the S3 bucket if the source traffic is not coming from our VPC Endpoint ID.

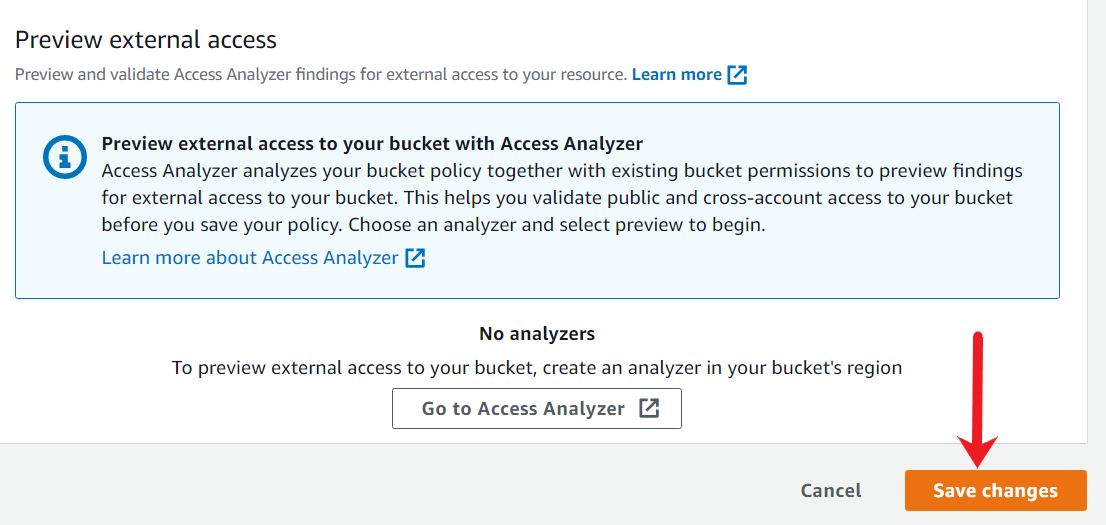
For easy reference, you can use the following sample policy and modify the bucket ARN and the VPC Endpoint ID according to your requirement.

* 1. {
  2. "Version": "2012-10-17",
  3. "Id": "Policy1415115909152",
  4. "Statement": [
  5. {
  6. "Sid": "Access-to-specific-VPCE-only",
  7. "Principal": "\*",
  8. "Action": "s3:\*",
  9. "Effect": "Deny",
  10. "Resource": ["arn:aws:s3:::awsexamplebucket1",
  11. "arn:aws:s3:::awsexamplebucket1/\*"],
  12. "Condition": {
  13. "StringNotEquals": {
  14. "aws:SourceVpce": "vpce-1a2b3c4d"
  15. }
  16. }
  17. }
  18. ]
  19. }

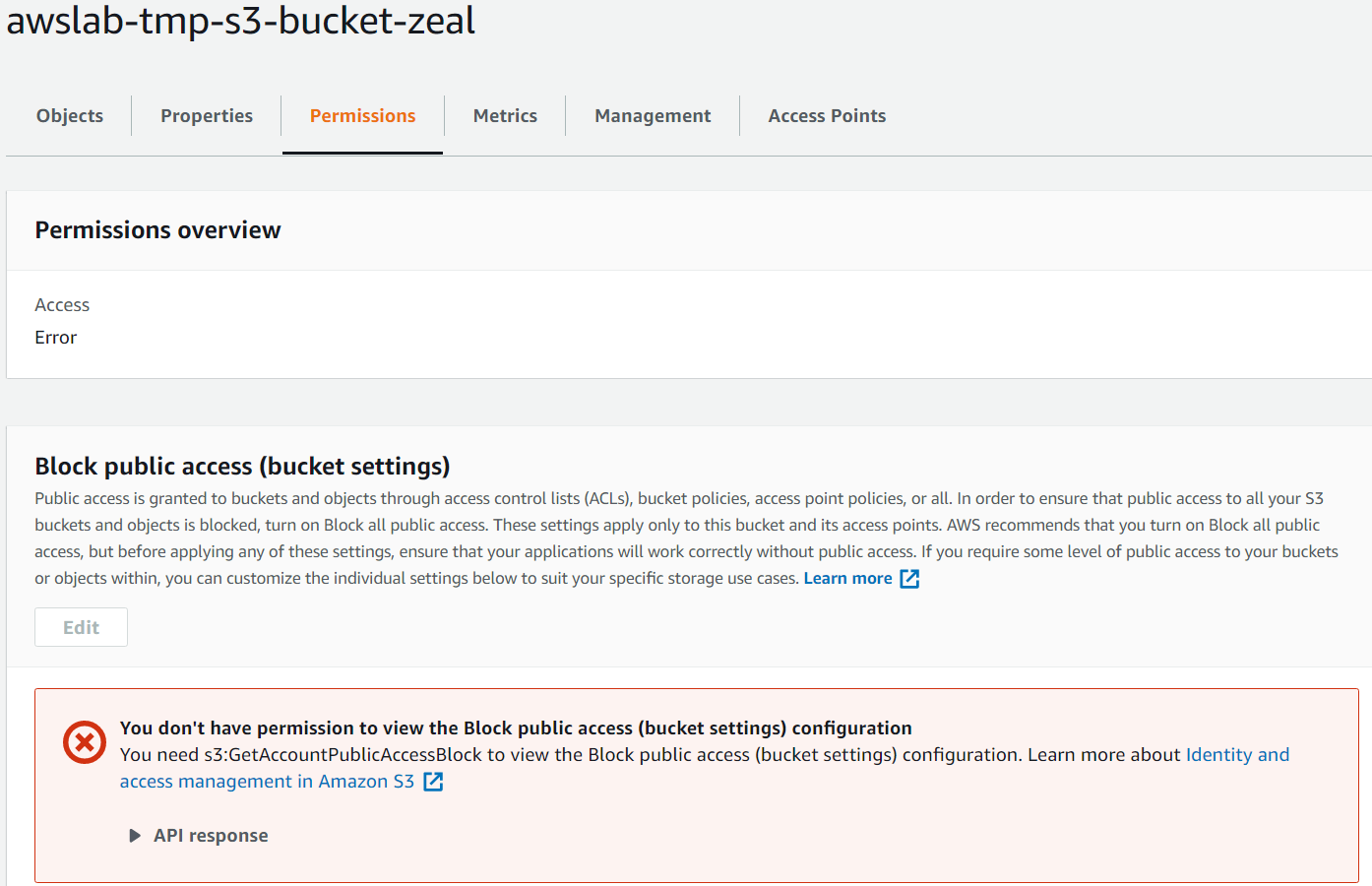
The following screenshot denotes the final S3 bucket policy used for this demo.



1. Scroll to the bottom of the page and choose Save changes.



1. After the new custom bucket policy is created, you should see multiple error and permission-related messages for the S3 bucket. This is as expected as the bucket only allows VPC Endpoint-related connections.



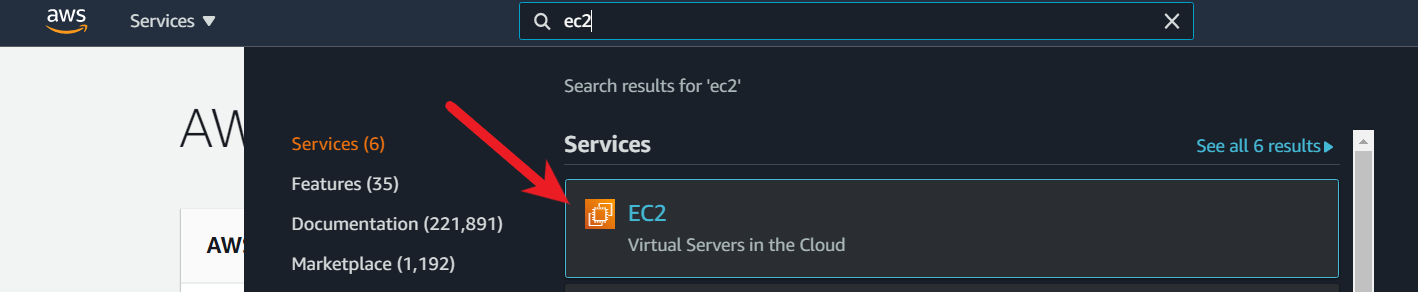
<https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies-vpc-endpoint.html>

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies.html>

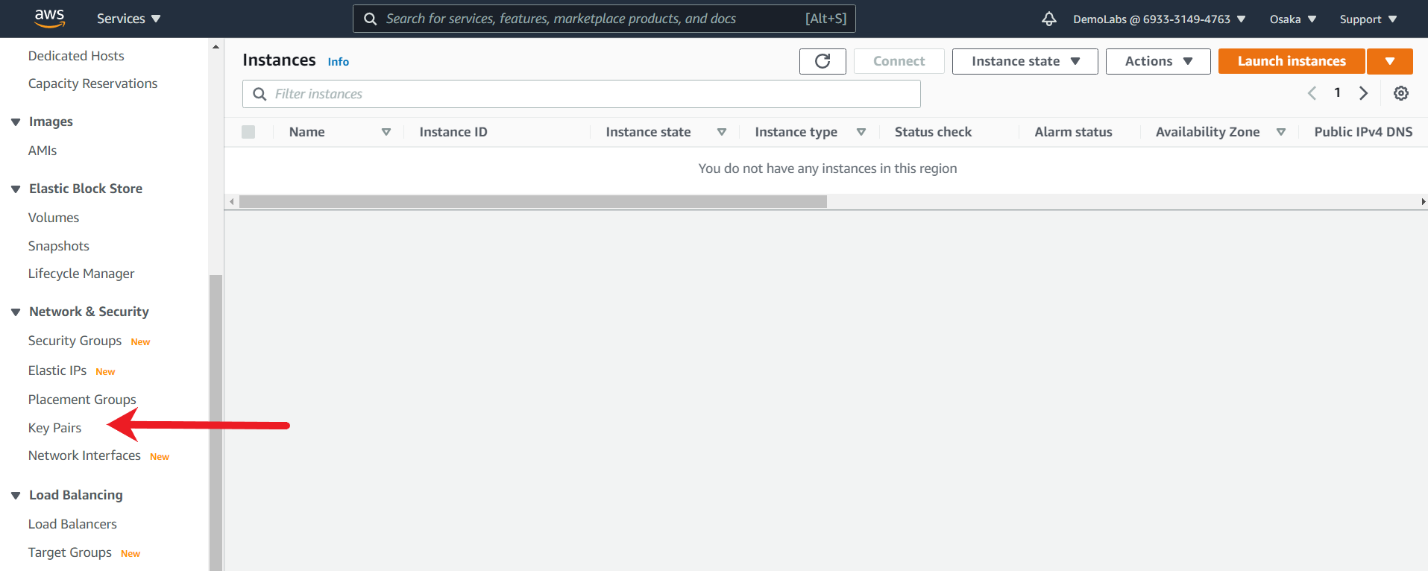
Create temporary PEM rivate key for ec2

You need to create a temporary key to log in to the EC2 instances to test the connectivity to the S3 bucket via VPC Endpoint.

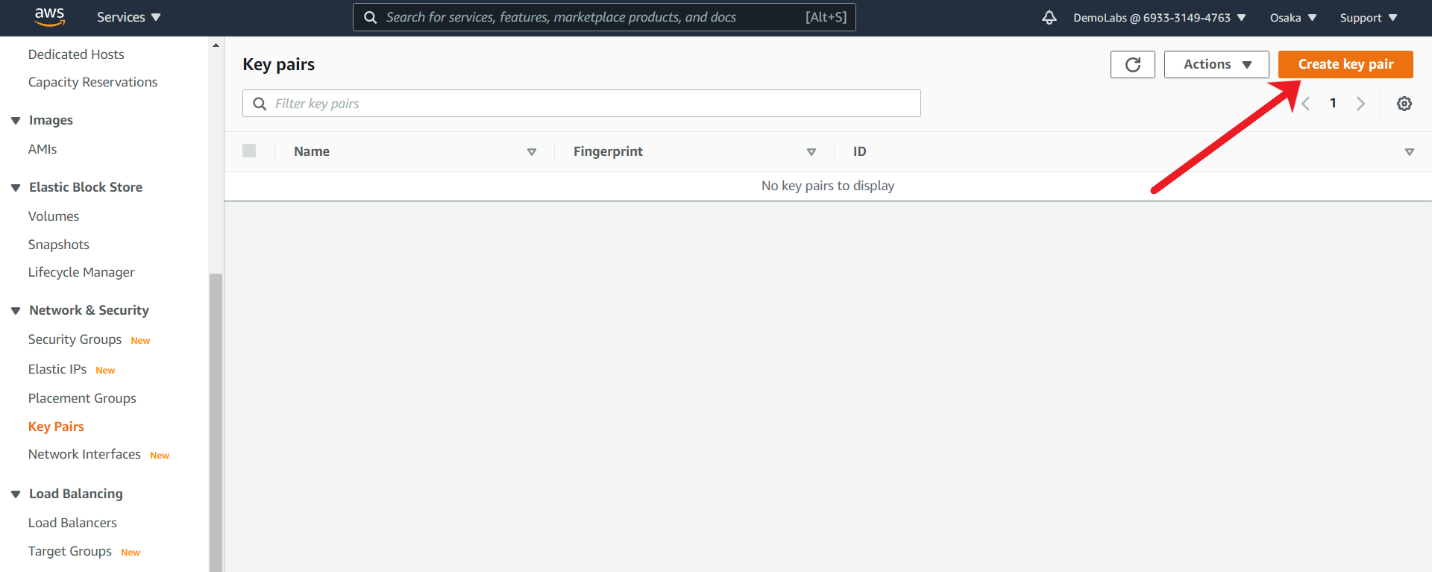
1. Open the Amazon EC2 console by searching for EC2.



1. In the navigation pane, under Network & Security, choose Key Pairs.



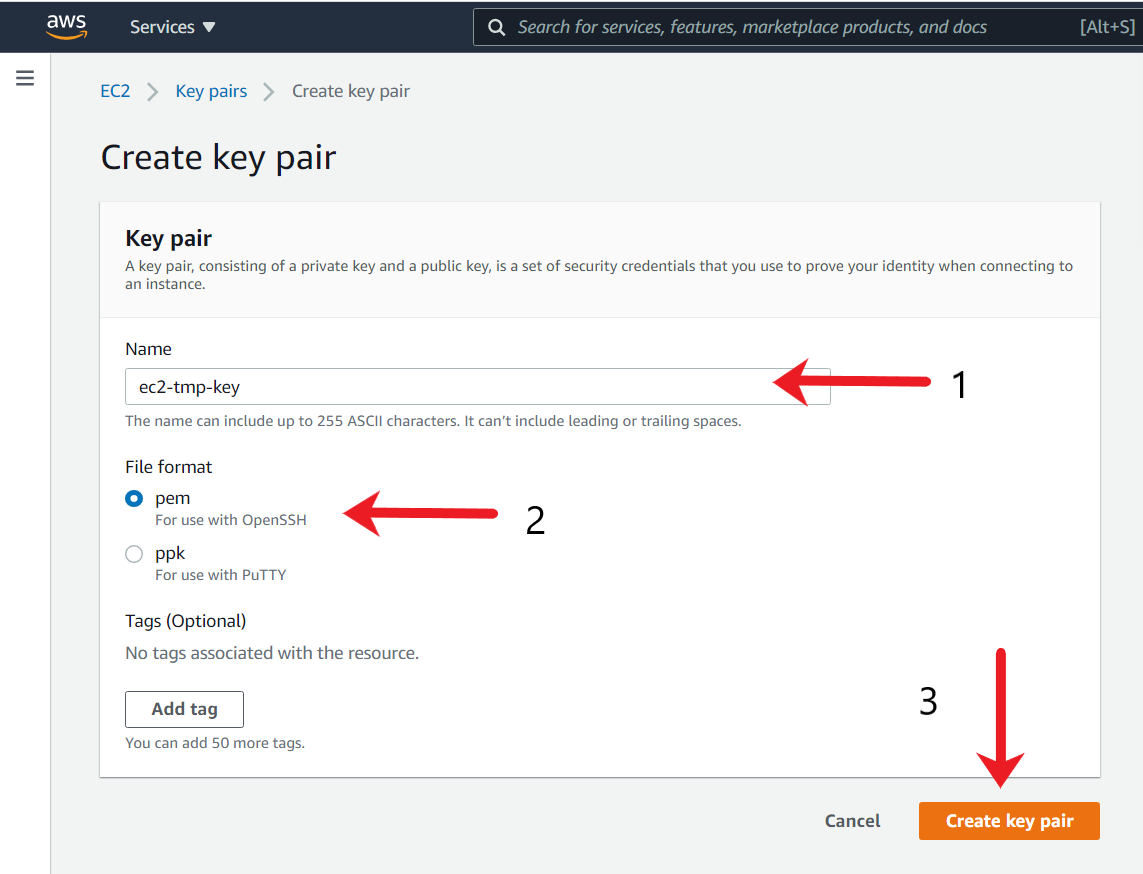
1. From the Key-Pair console, Choose Create key pair.



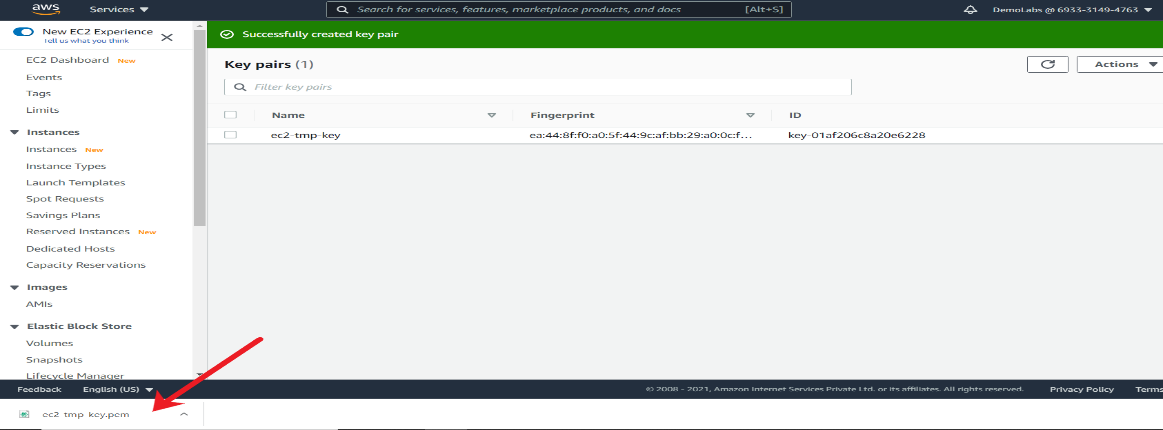
1. For Name, enter a descriptive name for the key pair.

For File format, choose the format in which to save the private key. For this use-case, I have selected PEM format, however if you are using tools like Putty, you can even select PPK.

Choose Create key pair.



The private key file is automatically downloaded by your browser. Save the private key file in a safe place.

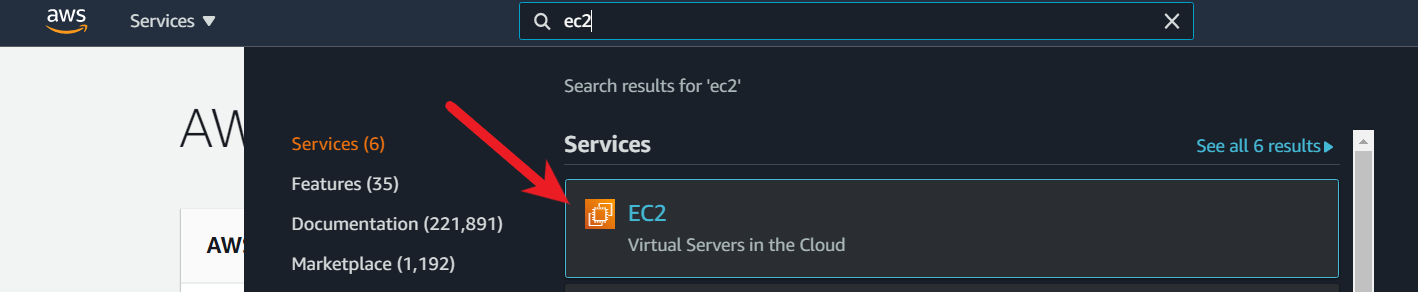


<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>

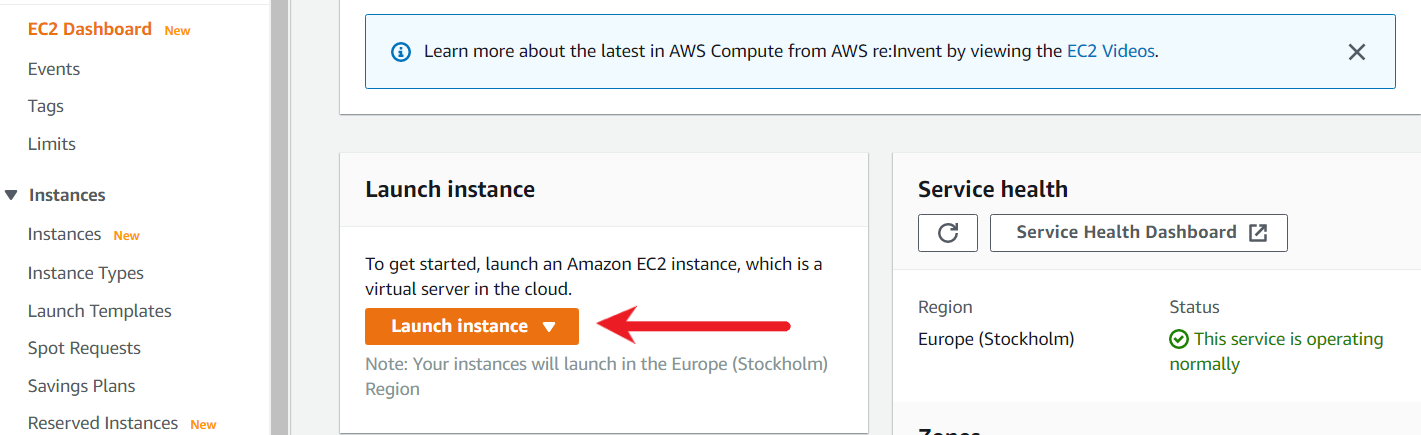
Launche ec2 to test endpoint connectivity

You need to launch 2 EC2 instances to test the entire setup. The EC2 instance in the Public subnet will be used as a Bastion to connect to the EC2 in a private subnet.

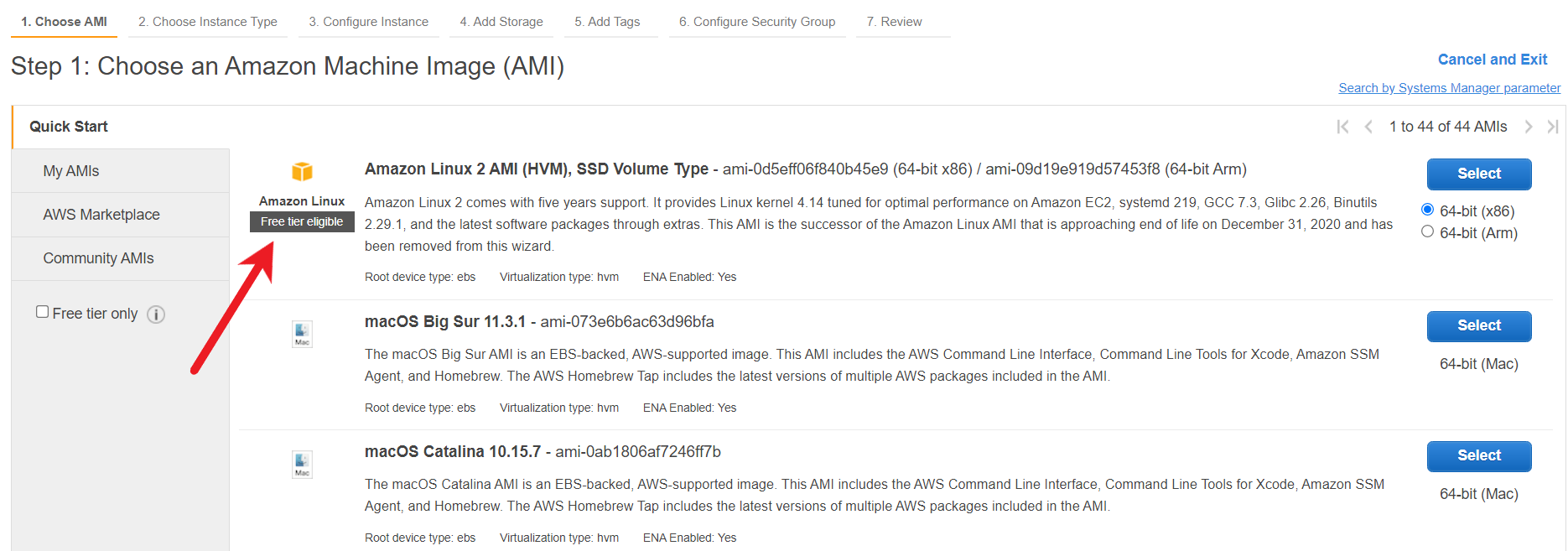
1. Open the Amazon EC2 console by searching for EC2.



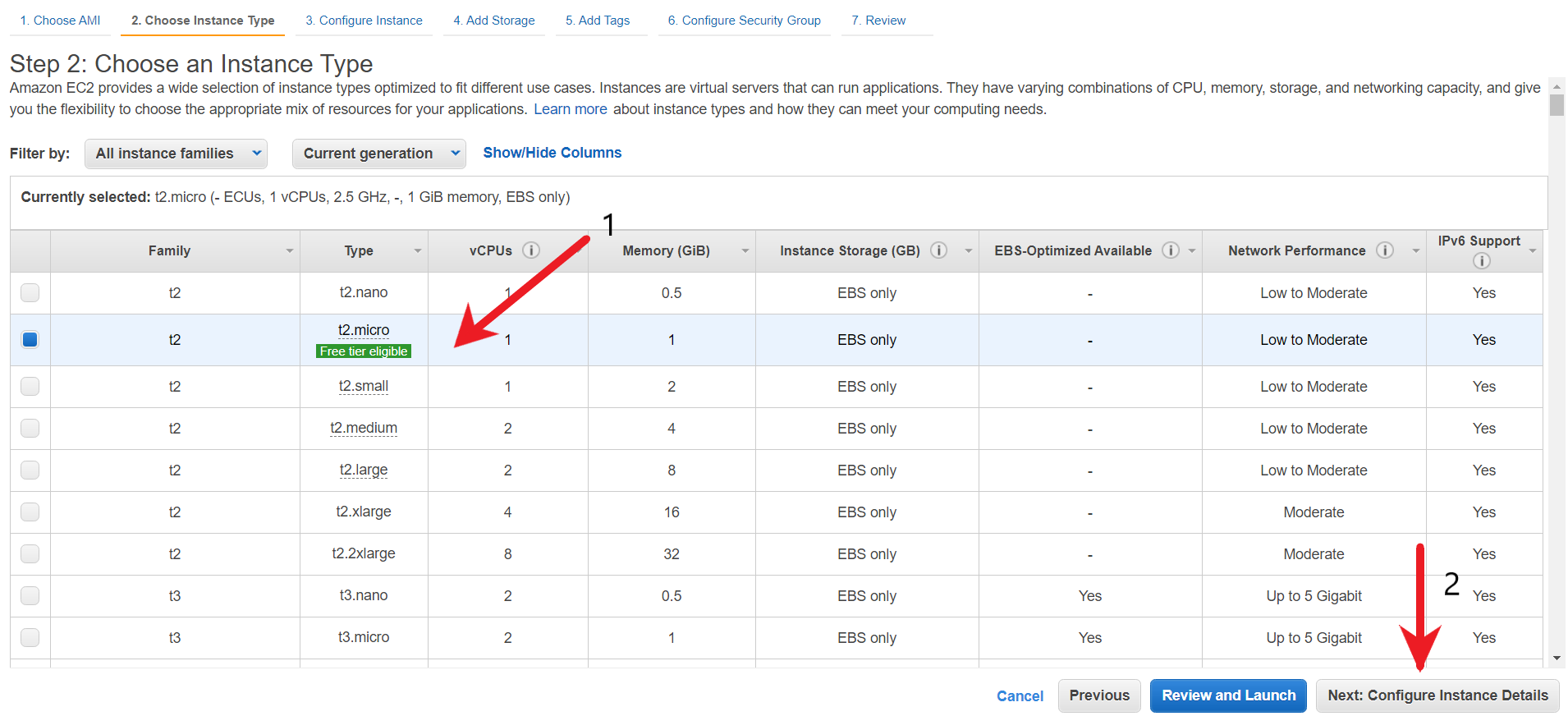
1. From the console dashboard, choose Launch Instance, and select Launch instance from the options that pop up.



1. Choose an Amazon Machine Image (AMI), locate Amazon Linux 2 AMI (HVM), SSD Volume Type, and then choose Select. (This AMI is labeled "Free tier eligible" and can be found at the top of the list.)



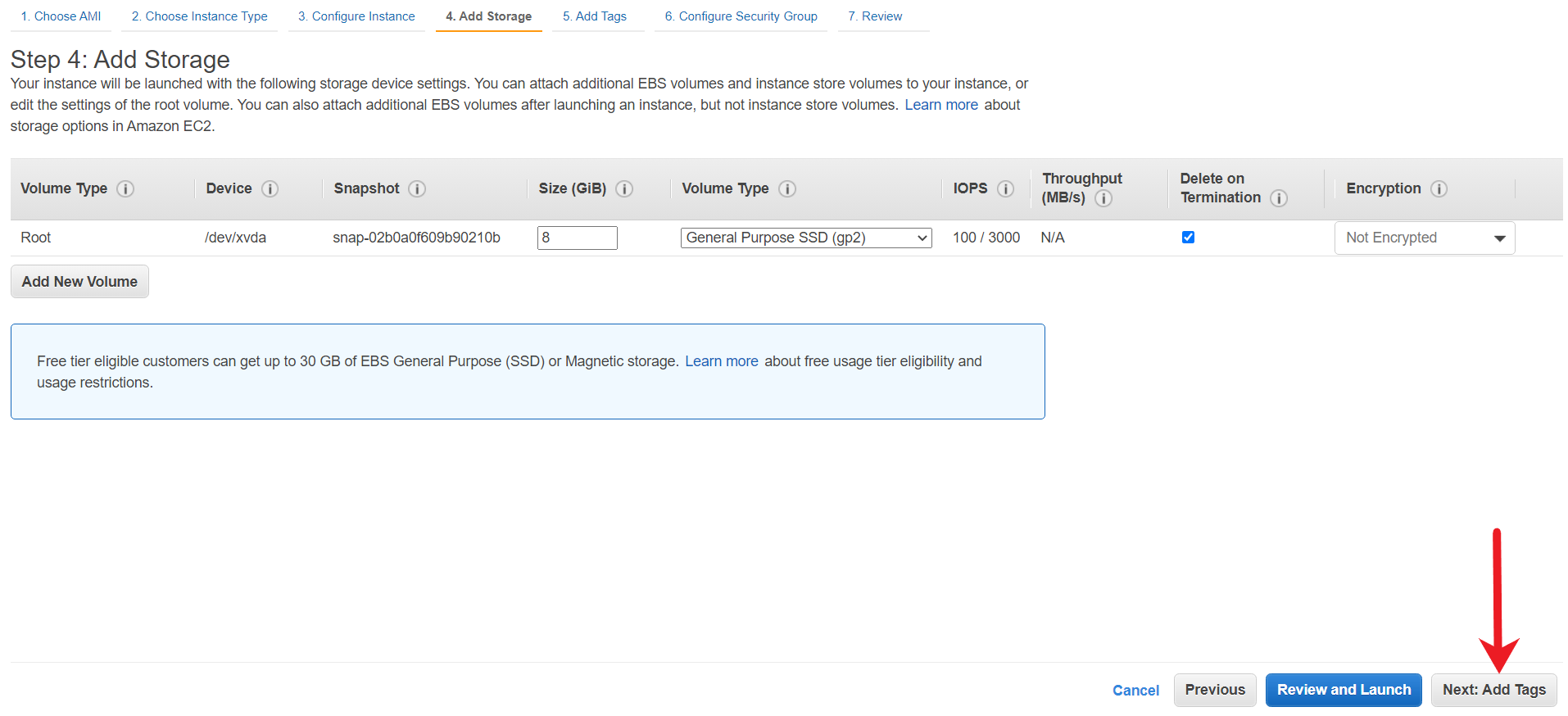
1. Choose an Instance Type page, choose the free tier eligible t2.micro type as the hardware configuration for your instance, and then choose Next: Configure Instance Details.



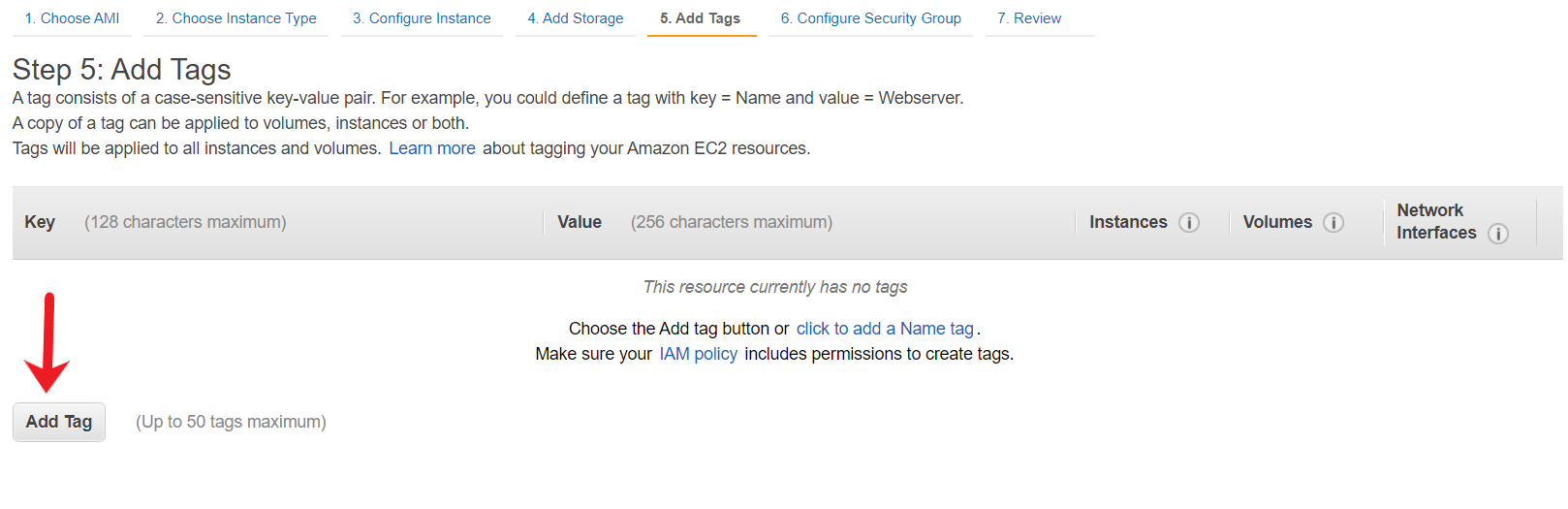
1. In the “Configure Instance Details” page, choose the private subnet that is associated with VPC endpoint and click on “Next: Add Storage”.



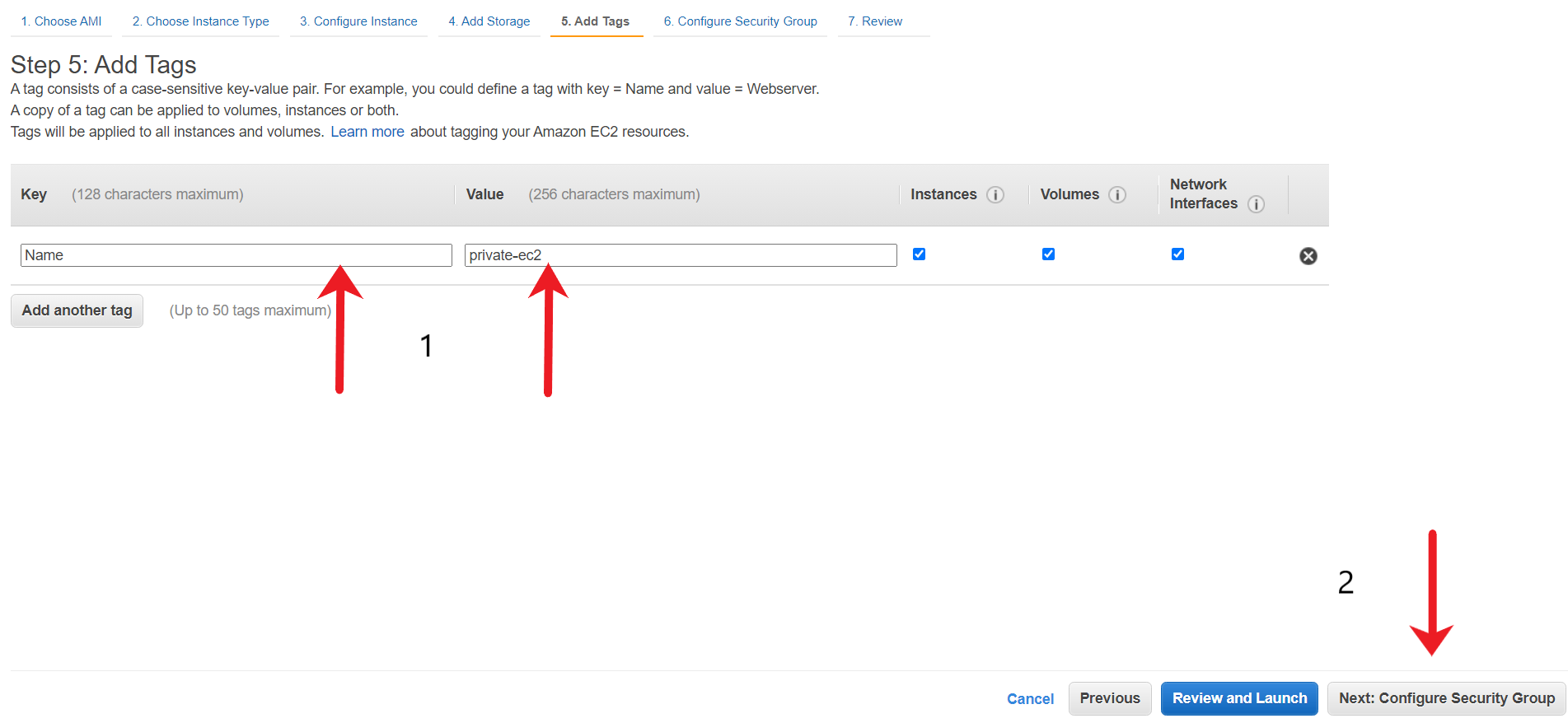
1. Add Storage page unchanged, and then choose Next: Add Tags.



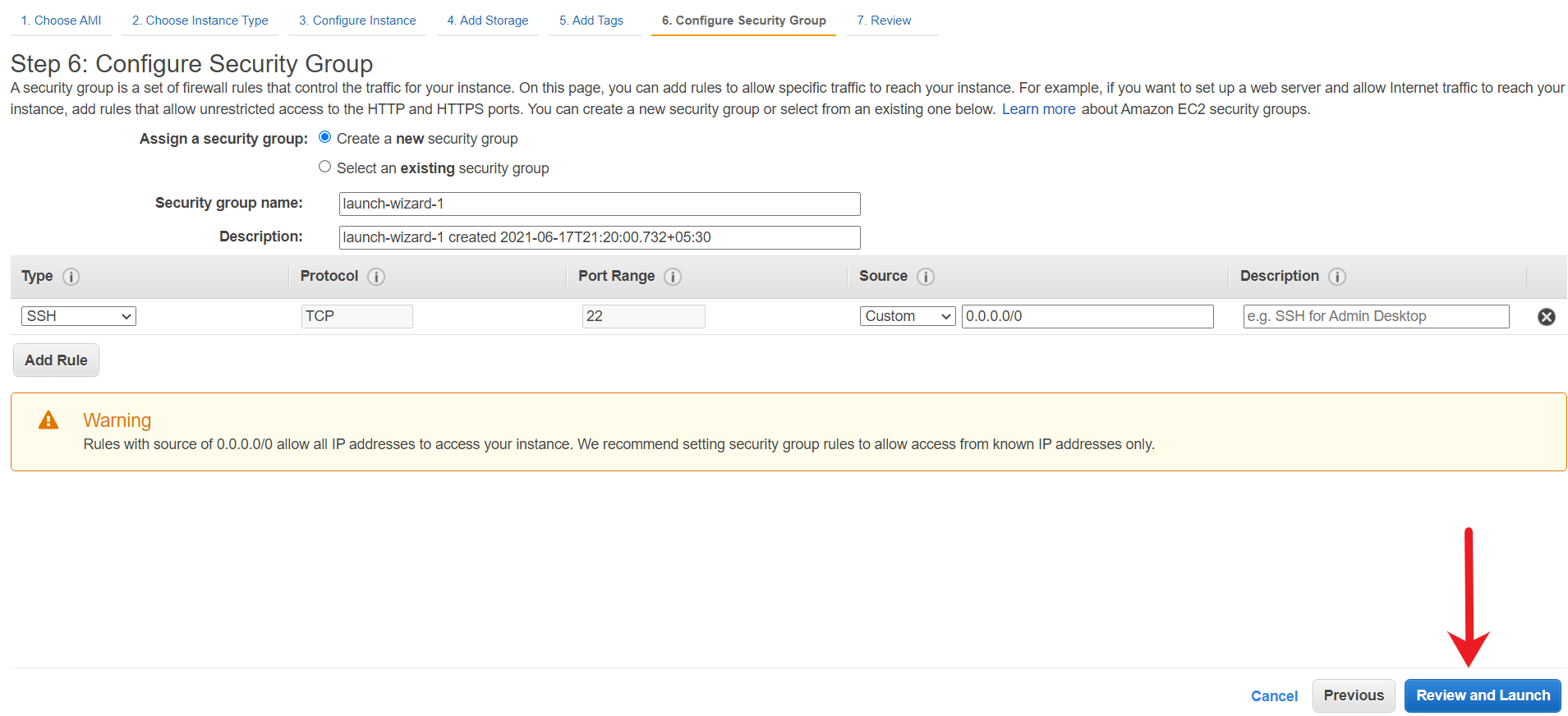
1. In the Tag page, click on Add tags.



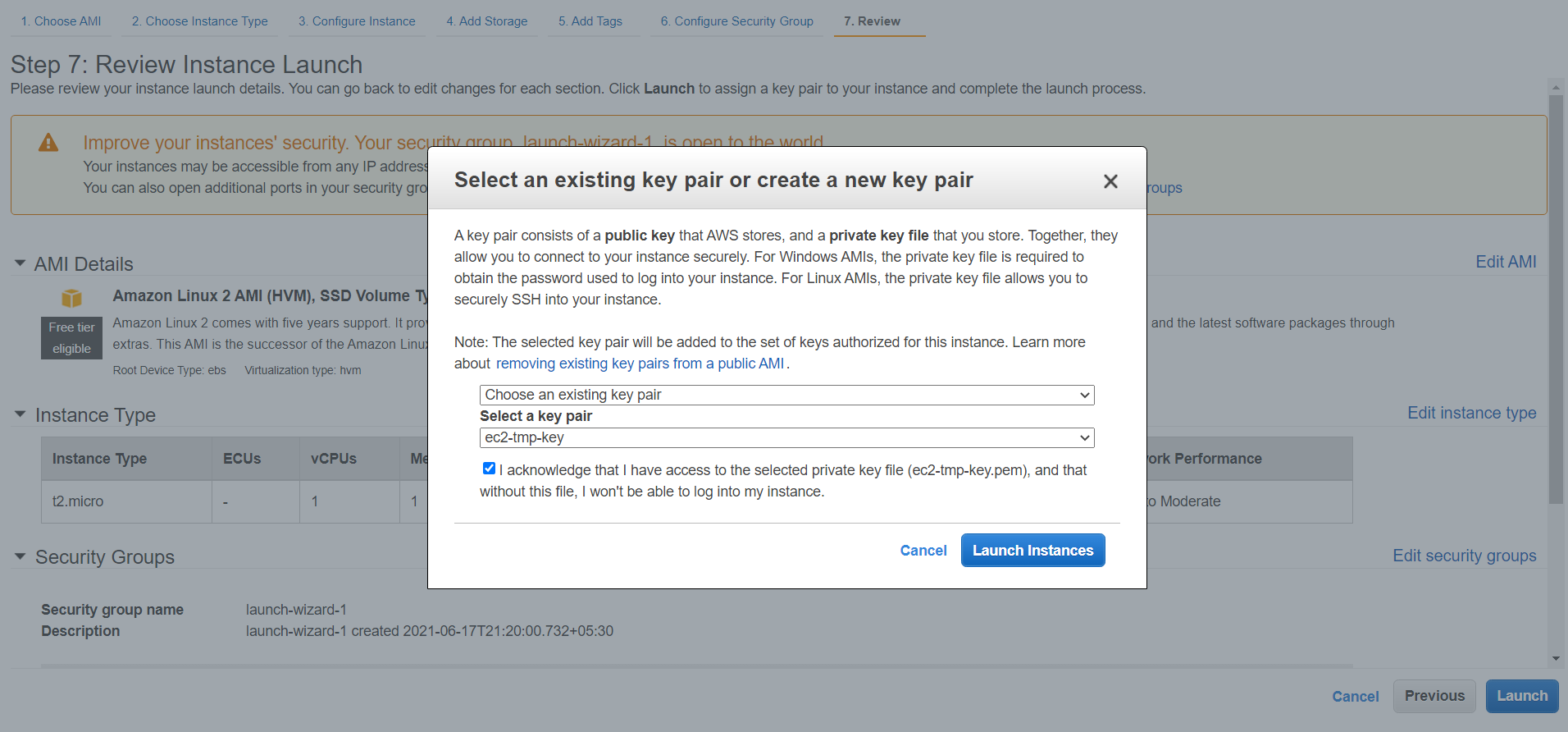
Create a new tag with the key as Name and Value as private-ec2 and then choose Next: Security Group.



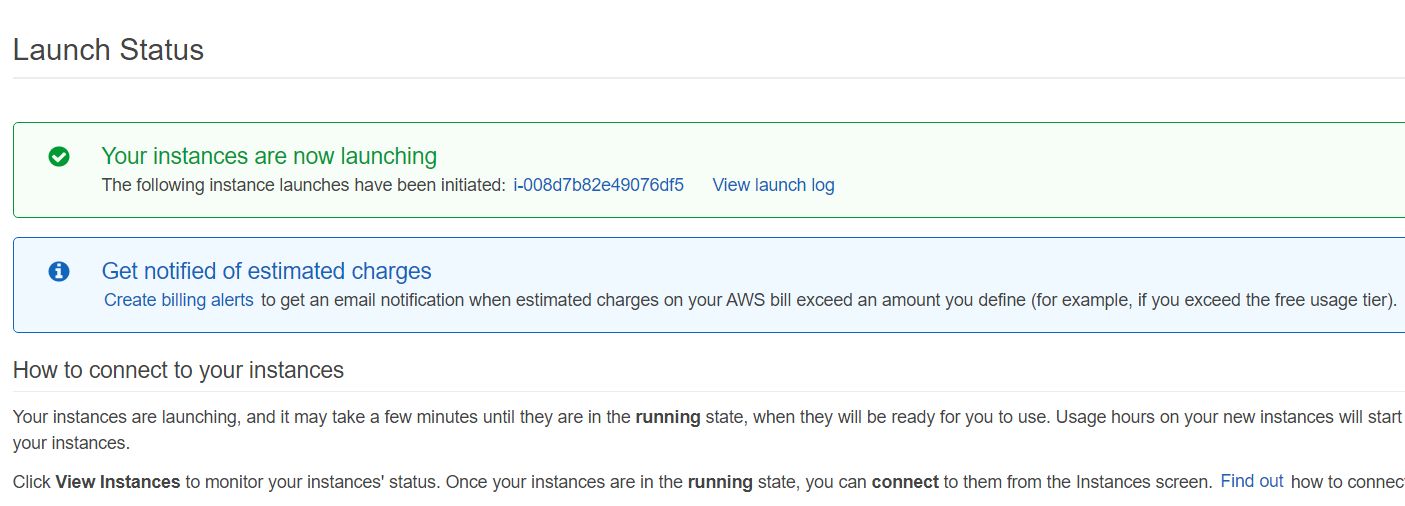
1. Leave the Security Group page unchanged. Make sure it allows Port 22 and then choose Review and Launch.



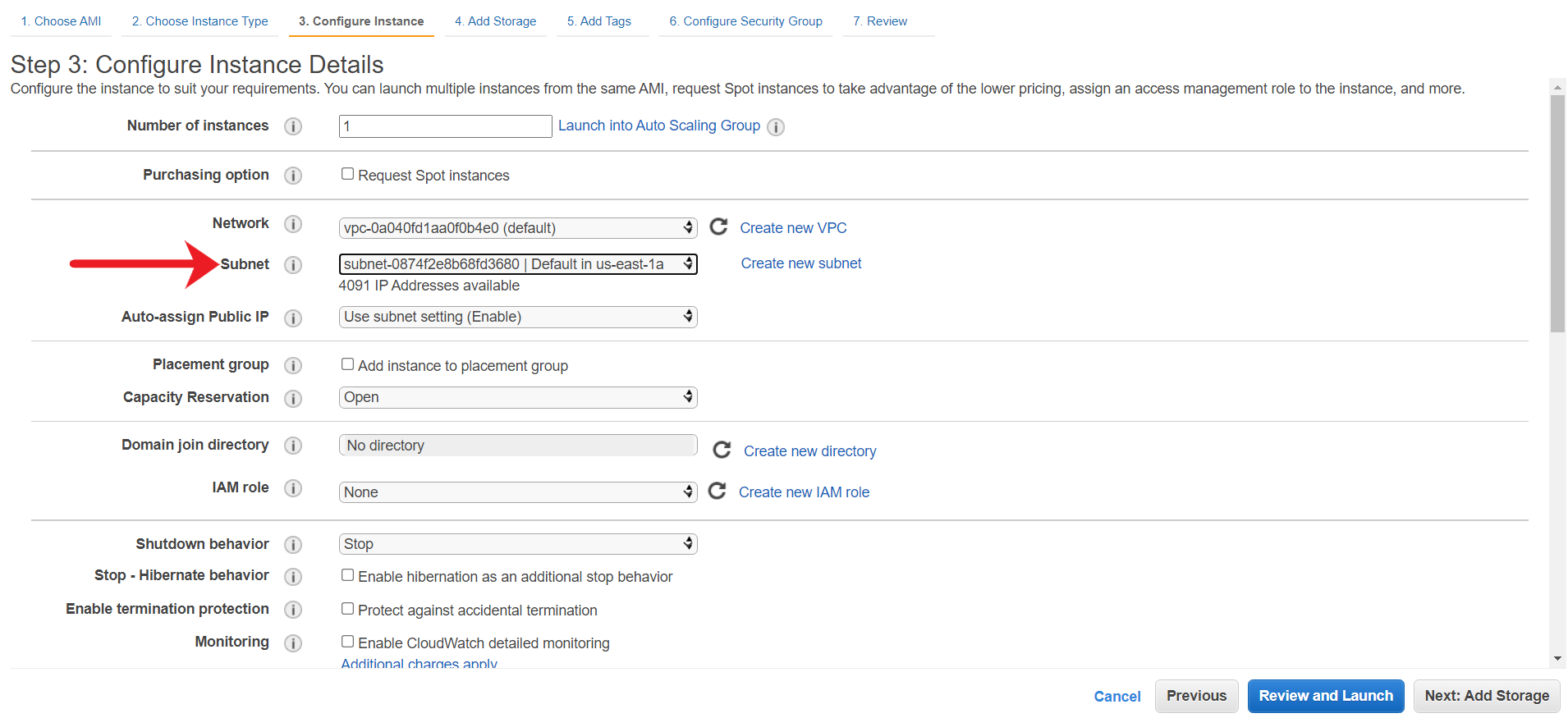
1. On the Review Instance Launch page, choose Launch. When prompted for a key pair, choose the temporary key pair that you had created. When you are ready, select the acknowledgment check box, and then choose Launch Instances.

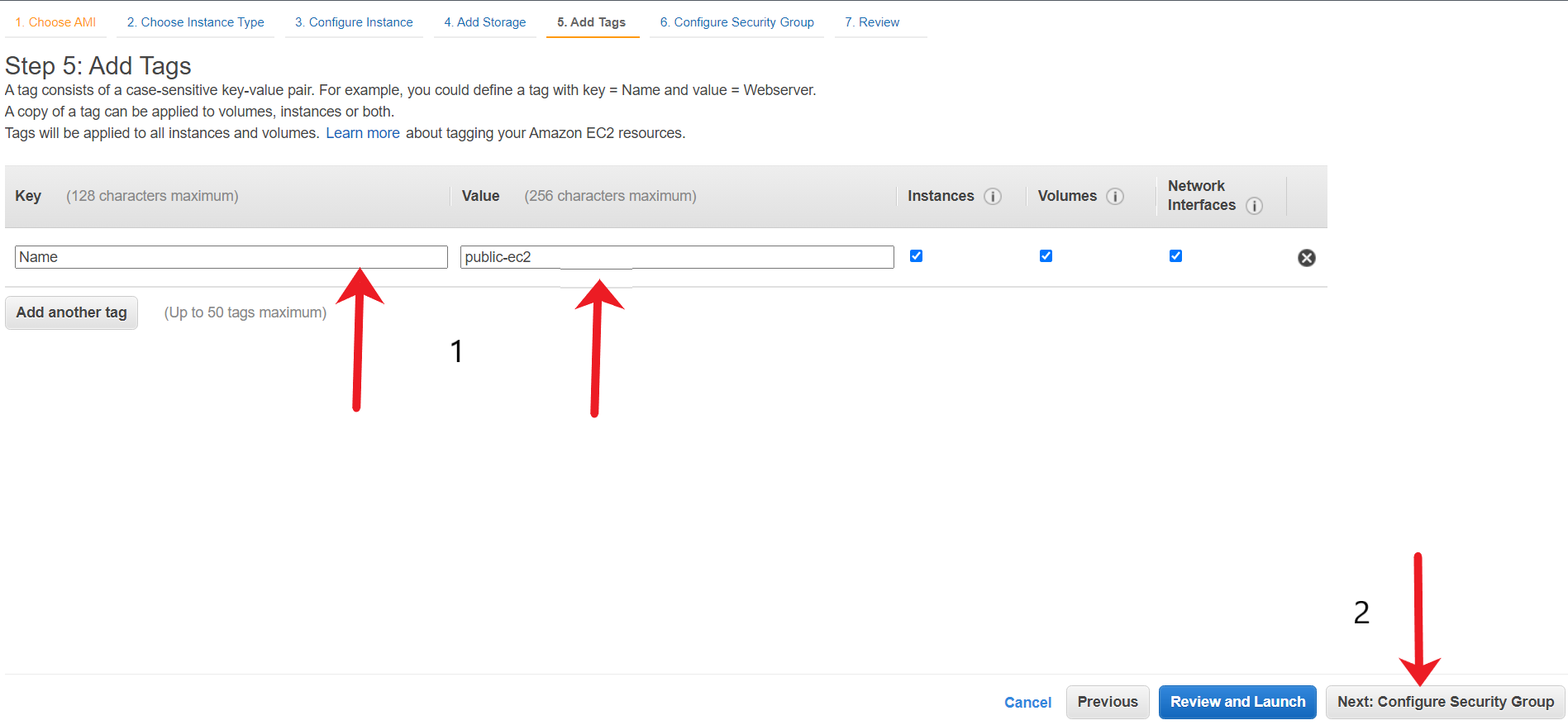


1. Select the instance ID to go to the EC2 console to see the status of the EC2 instance.



1. In a similar way, launch one more EC2 instance in a public subnet (us-east-1a for this demo). Add the tag with Name as public-ec2.





1. After both the EC2 instances are launched, verify from the EC2 console. You should see two EC2 with the right set of names.

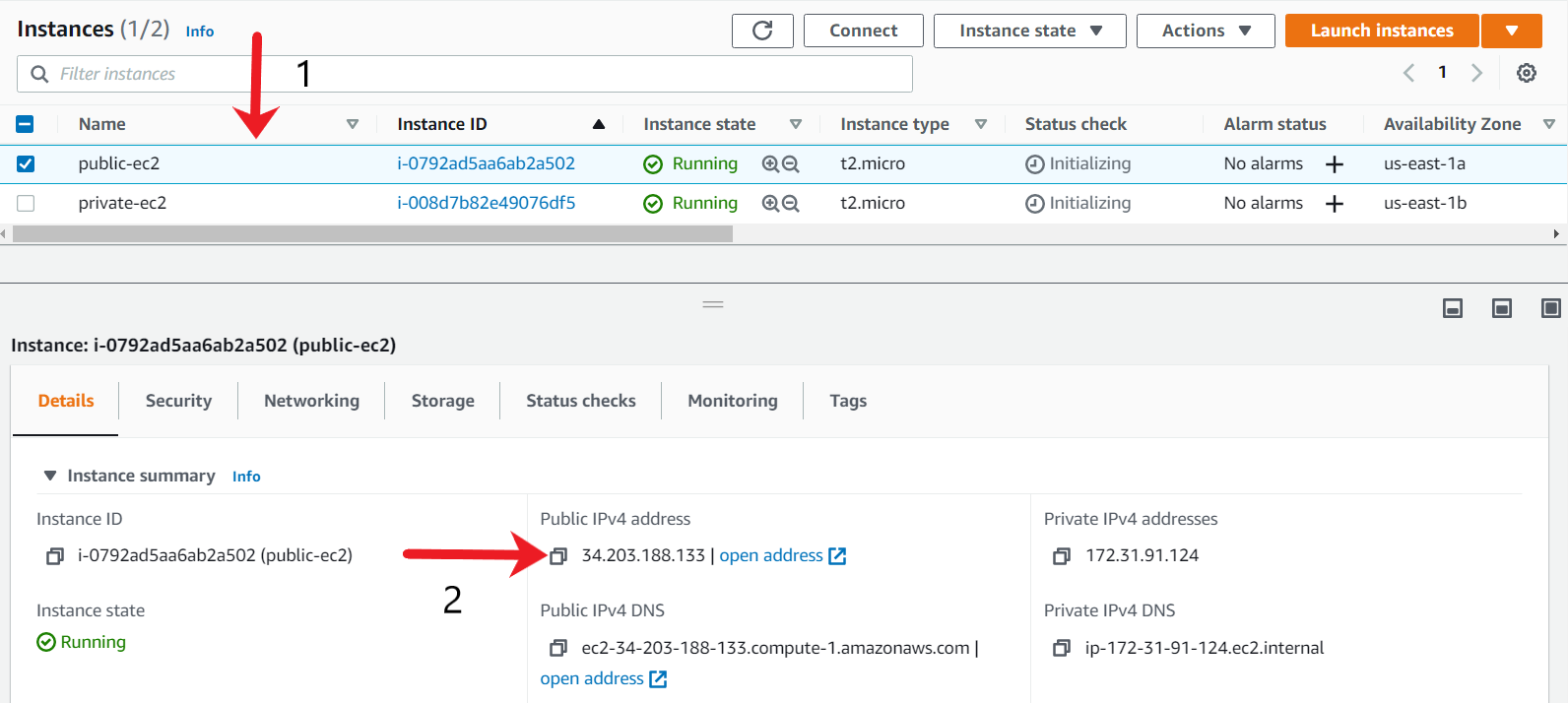


<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html>

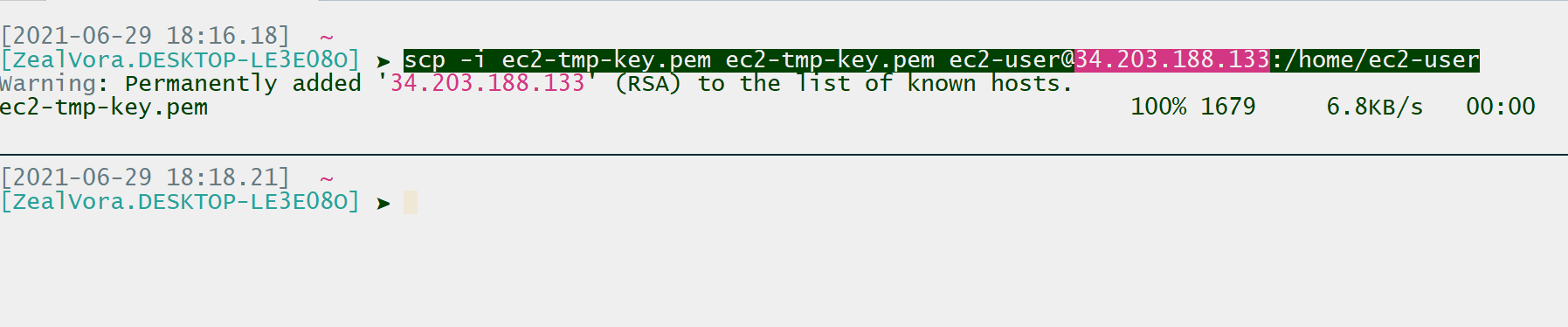
Verify connectivity between ec2 and s3 via endpoind

You need to test the connectivity between the EC2 instance in the private subnet to the S3 bucket to verify if upload related operation is working as expected. Connect to the Private EC2 instance through Public EC2 (acting as a bastion host) and verify the connectivity.

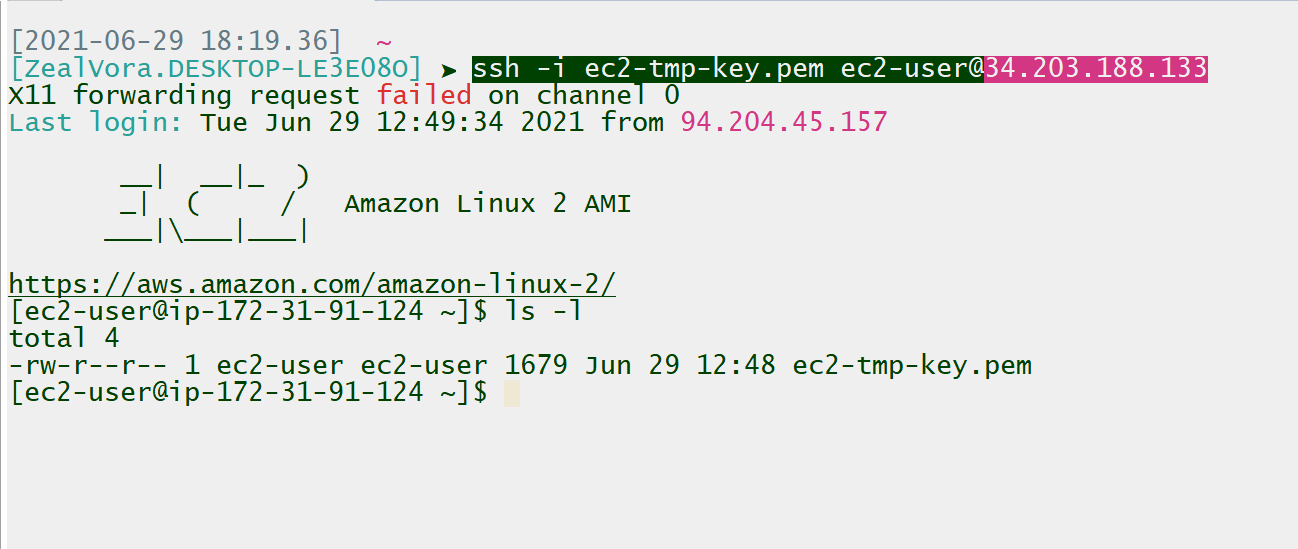
1. Fetch the Public IP of the EC2 instance running in the private subnet.



1. Copy the Private PEM key-pair to the EC2 instance using scp command as we will be using the key-pair to connect to Private EC2.



1. SSH into the Public EC2 instance. If you run a ls -l command, you should see the private key available.



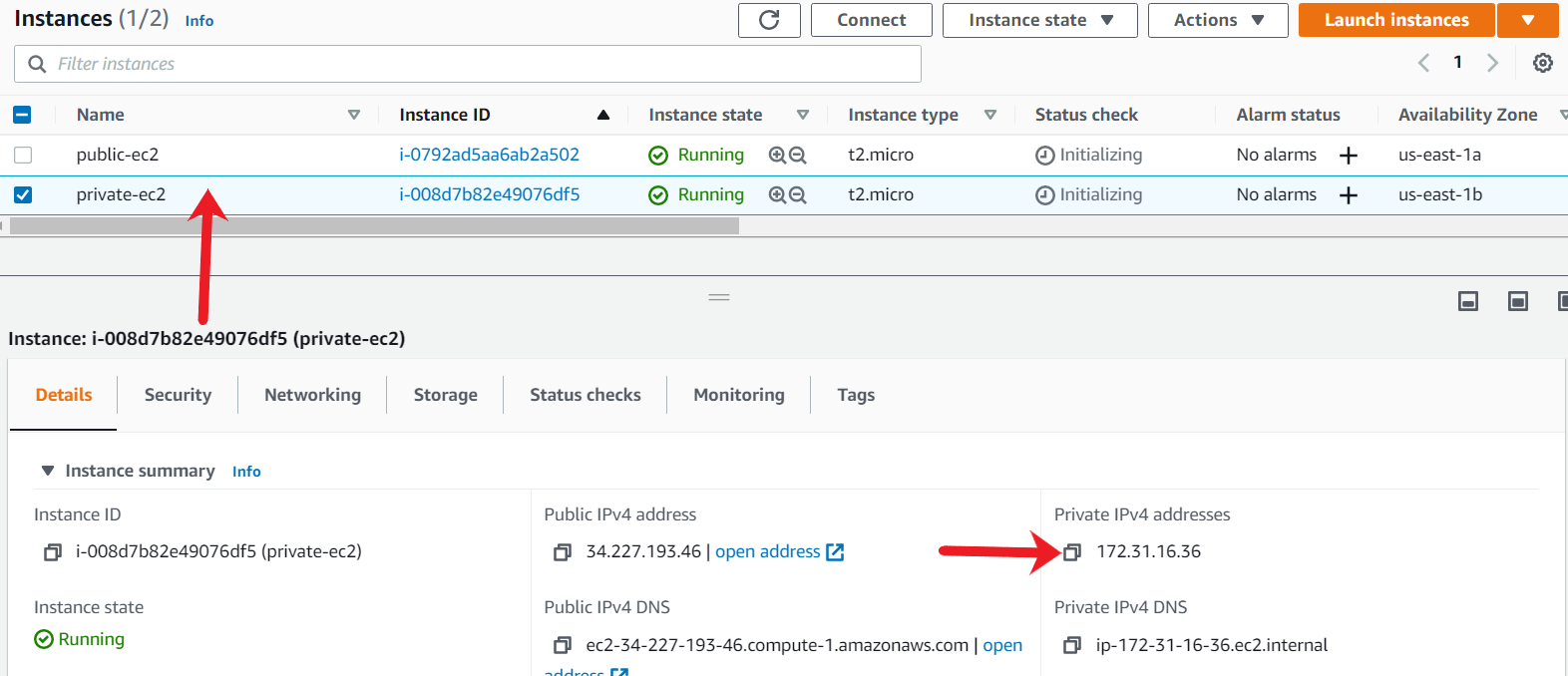
In-case if you get a “bad permission” error, you can change the permission of the file to 400 with the following command:

* 1. chmod 400 ec2-tmp-key.pem

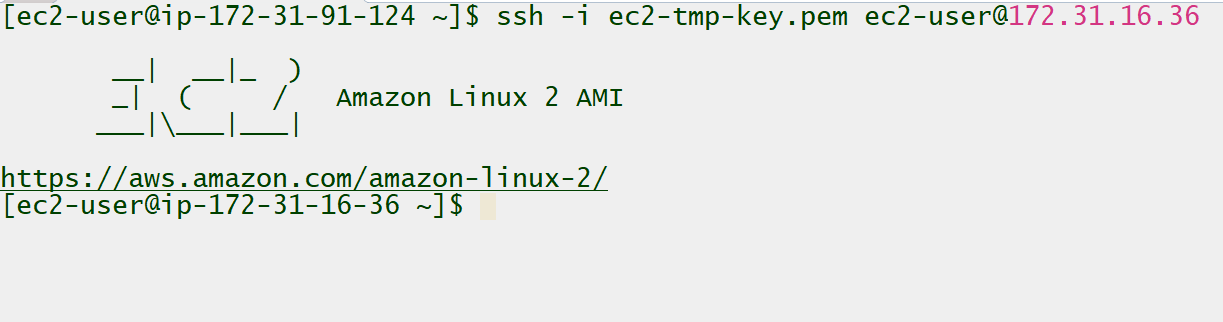
1. Change the permission of the private key to 400.



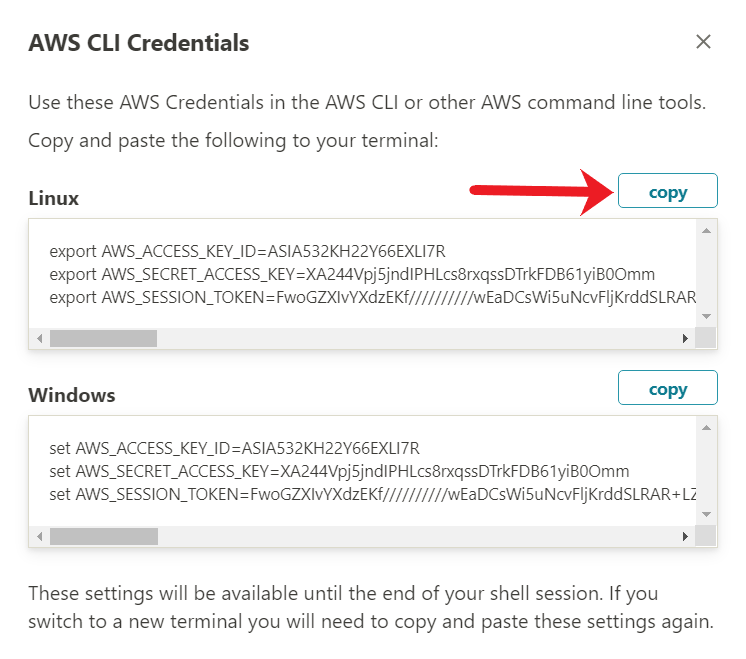
1. Fetch the Private IP associated with the private-ec2 instance.



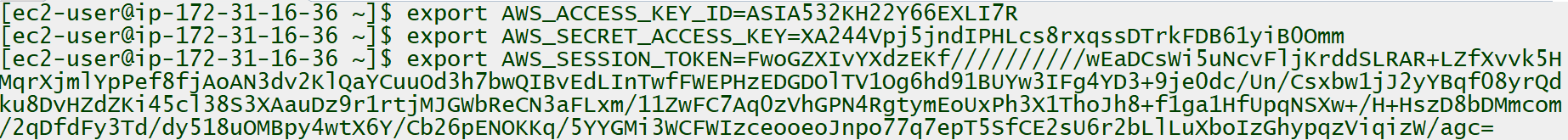
1. From the existing SSH session to public EC2, connect to private-ec2 via SSH using it’s private IP.



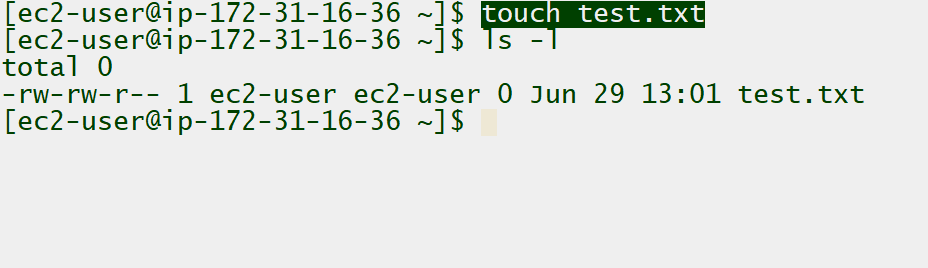
1. Copy the access/secret keys and the token from theWorkspace CLI.



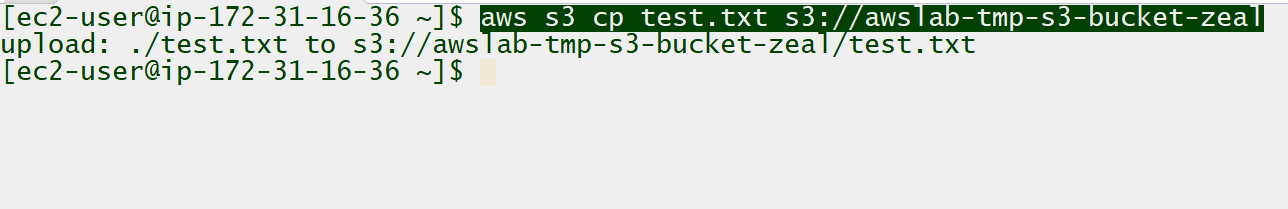
1. Paste the access/secret keys and the token within the SSH terminal of the Private EC2 instance.



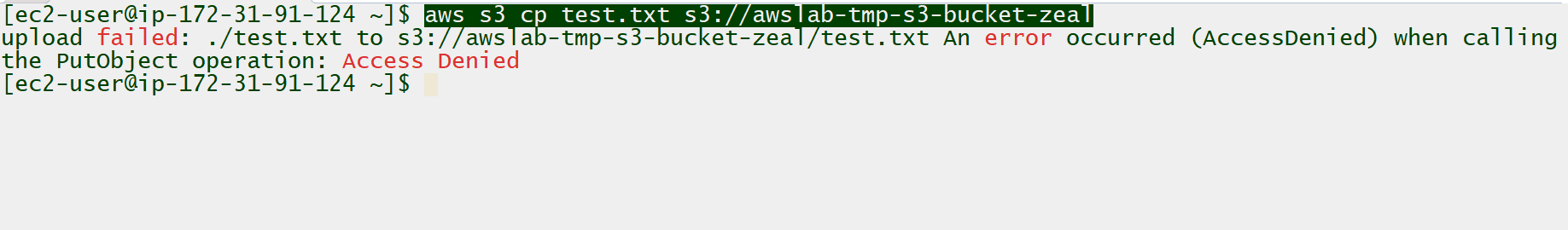
1. To test if the Put operation is working, create a next file named as test.txt.



1. Upload the test.txt file to your S3 bucket using the AWS CLI command. You should see the upload operation to be successful.



1. Logout from the private EC2 instance using CTRL+D (in windows) and you should be back to public EC2 instance terminal. You will need to re-export the AWS Access/Secret keys from the Workspace again within the terminal.
2. After you have exported the access/secret keys and the token in the terminal, try creating the same test.txt file and test if the upload functionality is working. You should get an **Access Denied** operation since Bucket policy only allows connections from VPC Endpoint.

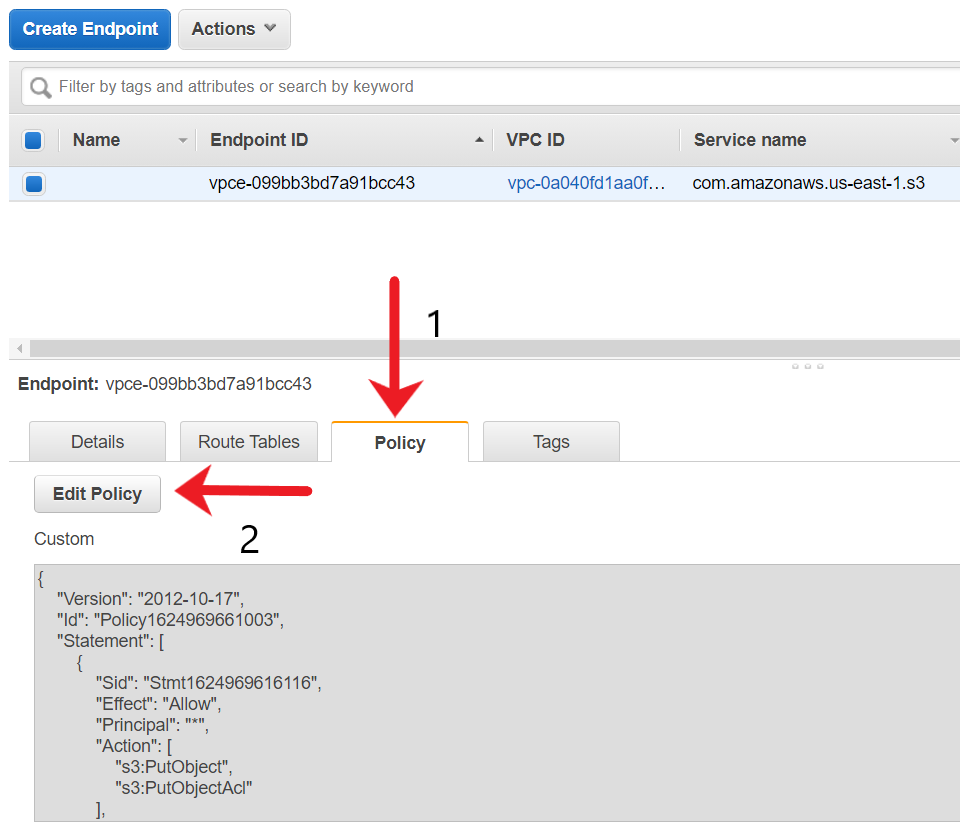


Clean Up

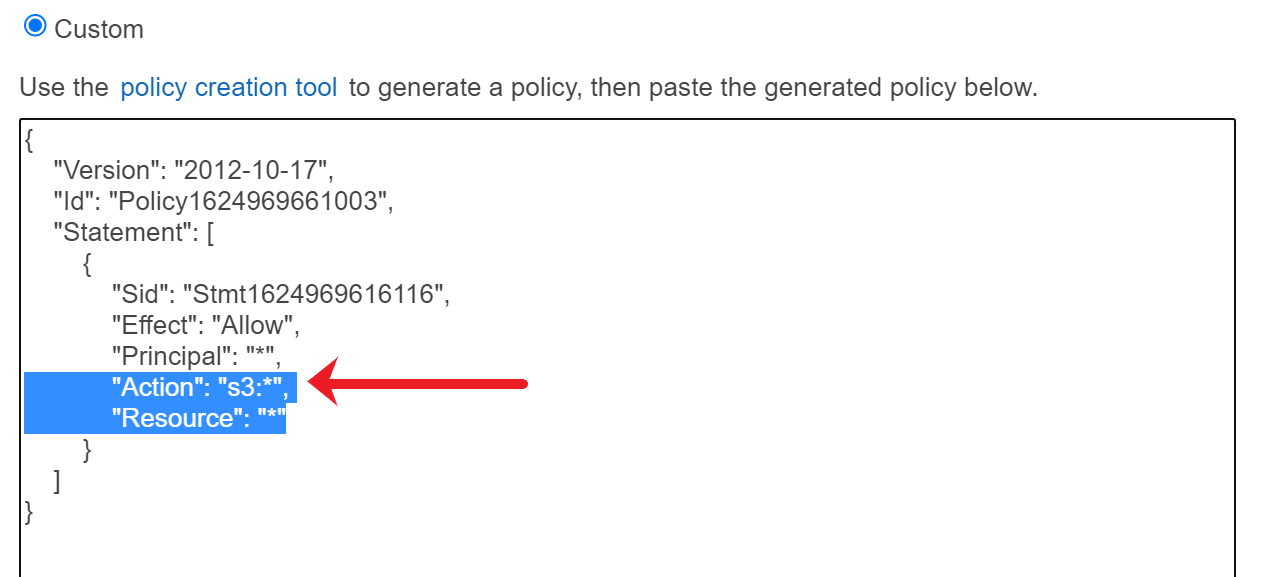
As we have successfully demonstrated a proof of concept for a VPC Gateway Endpoints, we'll clean up our AWS resources at this time. Delete all the resources created for this proof of concept.

1. To delete the S3 bucket, you MUST change the VPC Endpoint policy to allow Delete related operations on the S3 resource. Modify the VPC Endpoint policy and change the Action to s3:\* from the older s3:Put.

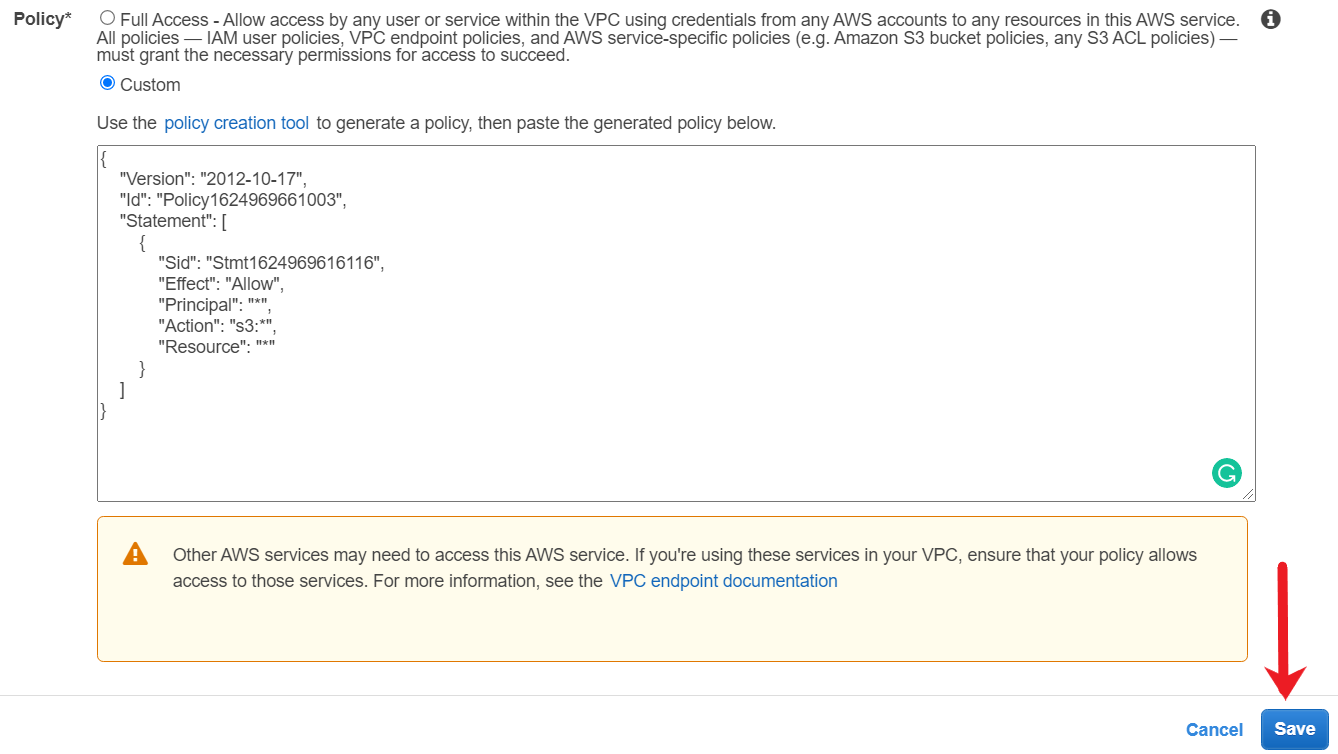
Go to the VPC Endpoint console, and under the Policy tab, click on Edit Policy.



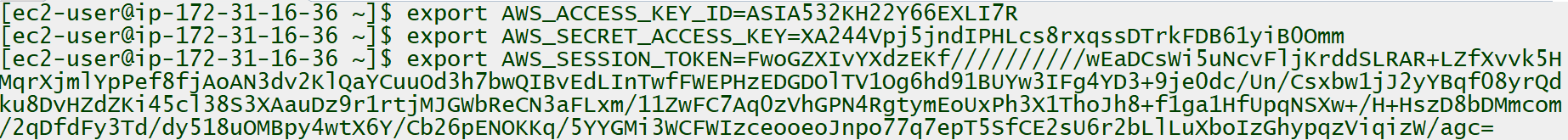
1. Remove the actions of PutObject and PutObjectAcl and replace it with s3:\* so that all operations are now allowed, including deletion-related ones. You can also replace the resource ARN to \* similar to the below screenshot.

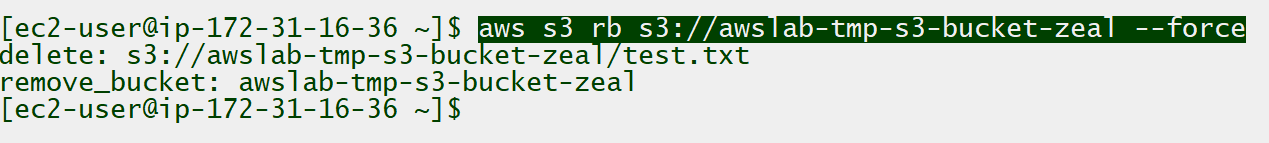


Once you have customized the policy, save the VPC Endpoint Policy by choosing Save.

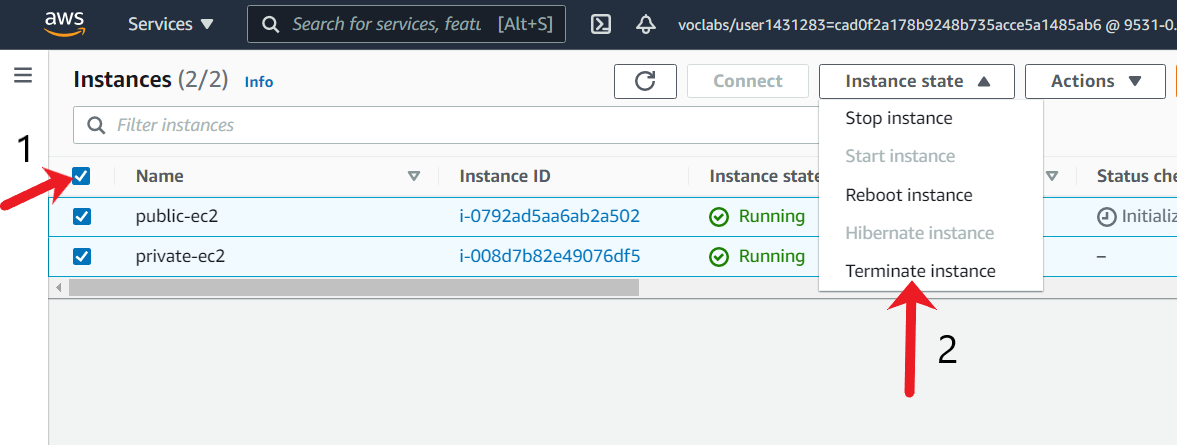


1. Connect to the Private EC2 instance via SSH and run the AWS CLI command to delete the S3 bucket and all its contents. Make sure to export the AWS Access/Secret keys and then token obtained from the workspace before running the AWS CLI commands.

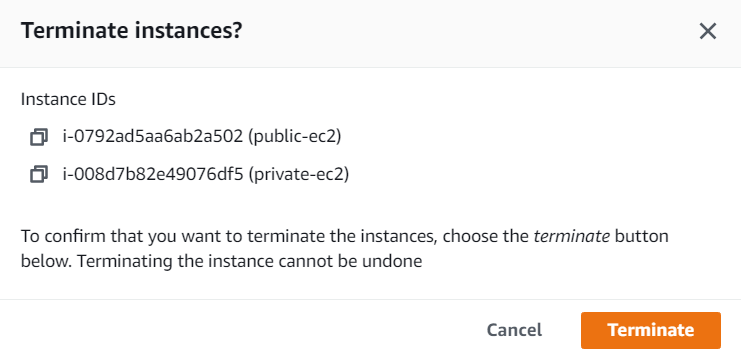




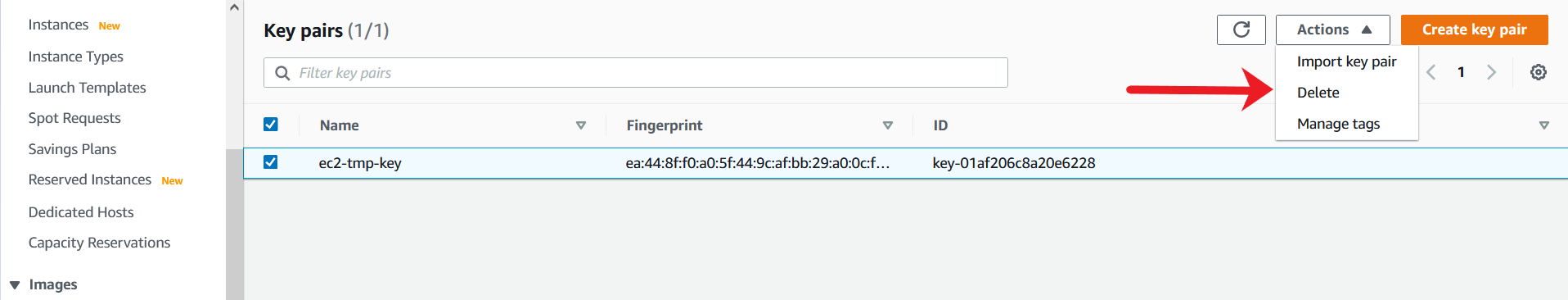
1. To terminate the EC2 instances, choose both the EC2 instances , click on Instance State and choose Terminate.

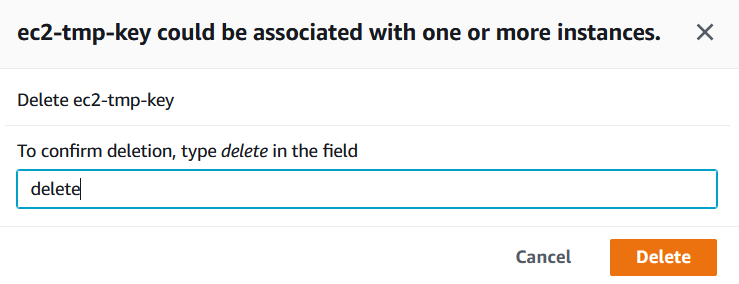


In the new confirmation window, choose Terminate.

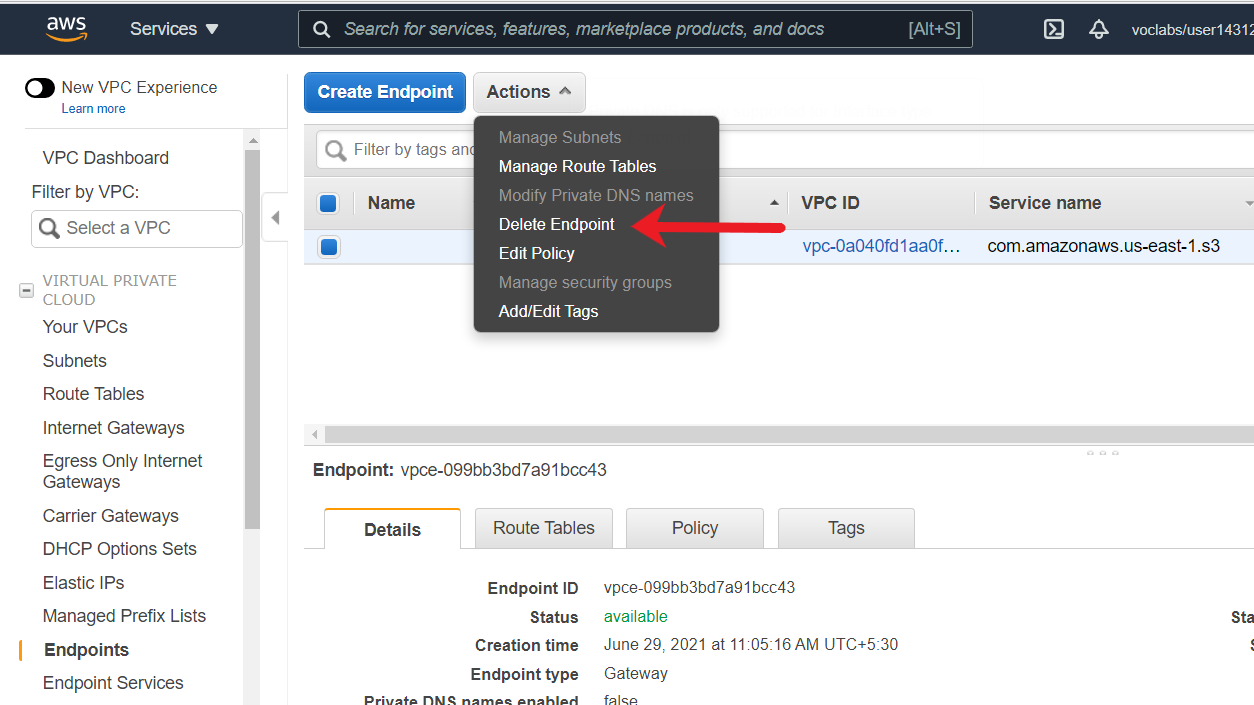


1. To delete the EC2 key-pair, open the Amazon EC2 console. In the navigation pane, choose Key Pairs. Select the key pair to delete and click on Action, and select Delete. In the confirmation field, enter delete and then choose Delete.

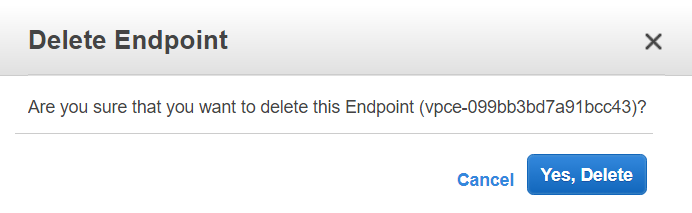




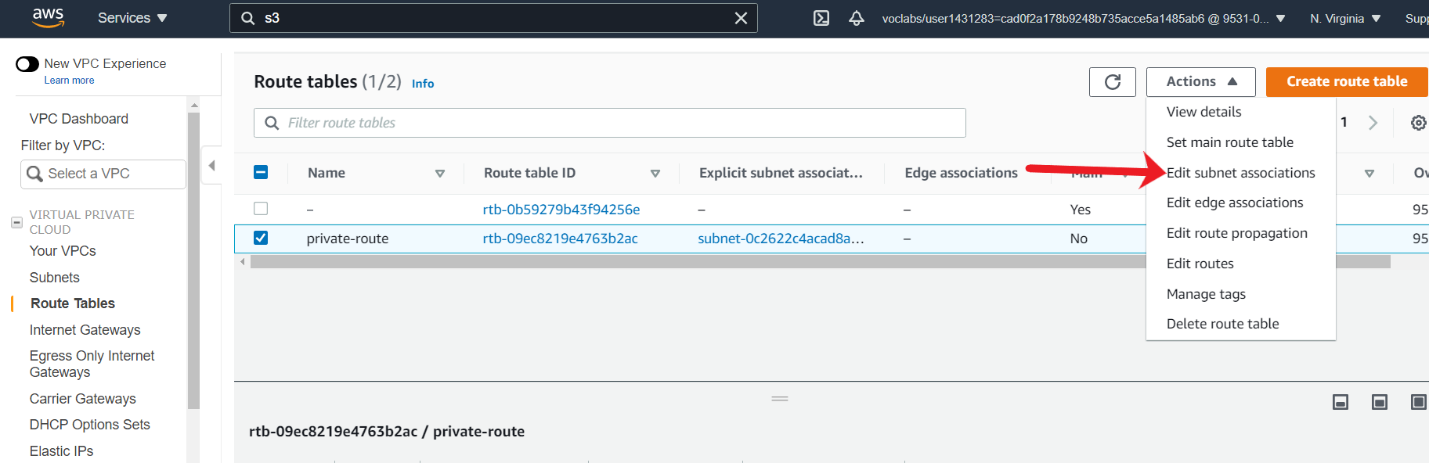
1. To delete the VPC Endpoint, go to the VPC console and select the Endpoints tab from the navigation menu. Select your endpoint, click on Actions and choose Delete Endpoint.



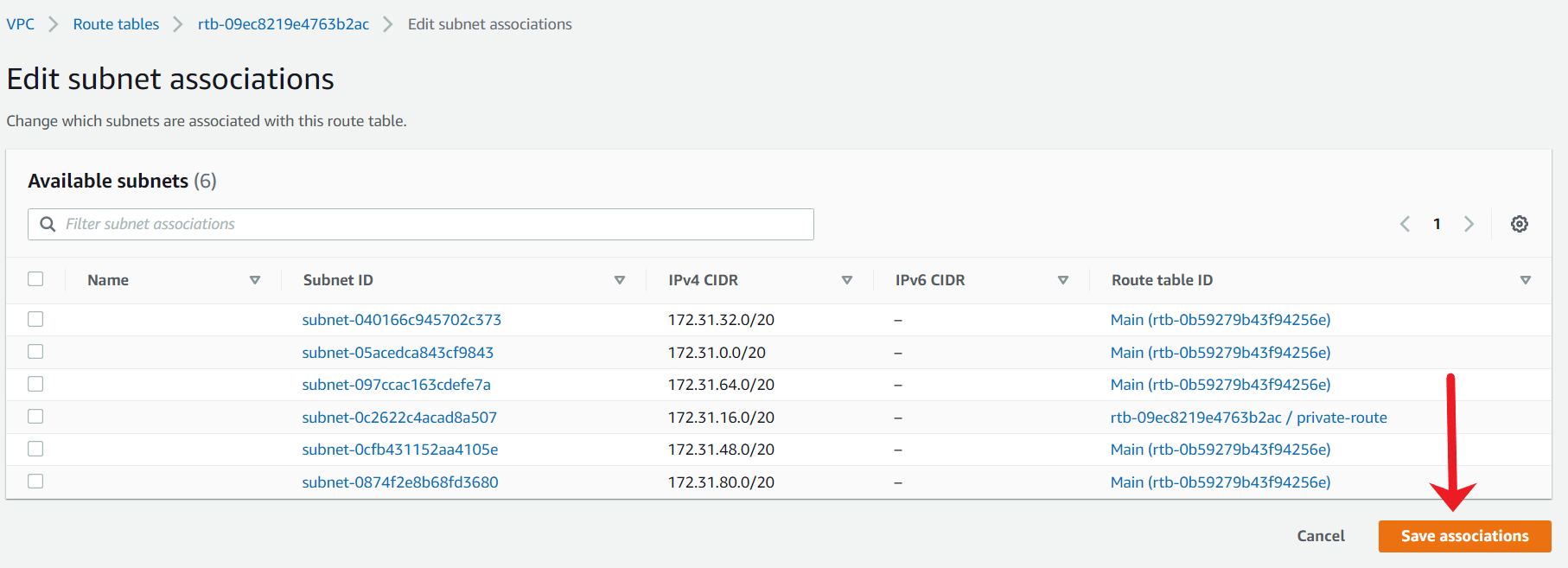
On the confirmation window, choose **Yes, Delete**.



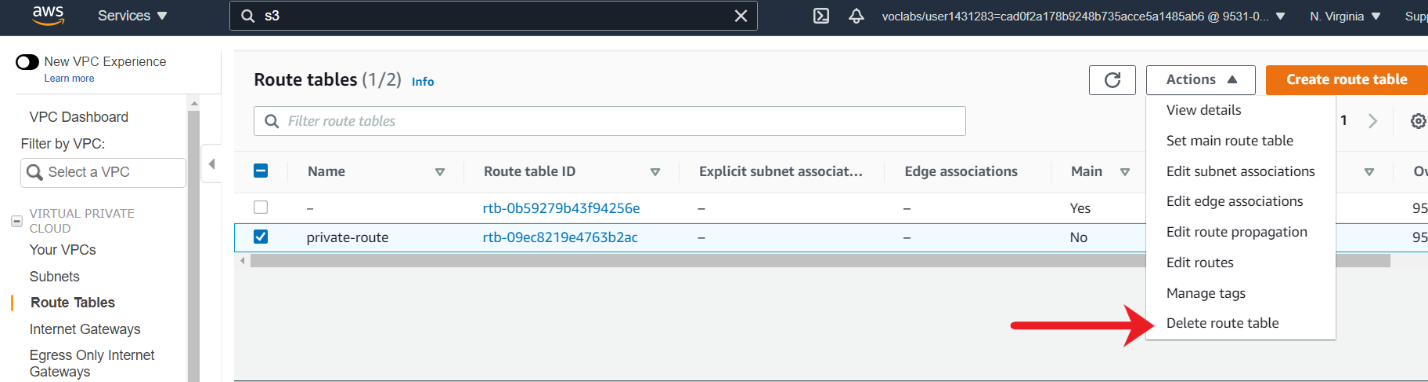
1. To delete the custom route tablet, go to the VPC console and select the Route Tables tab from the navigation menu. Select your custom route table and click on Actions and choose Edit subnet Associations.



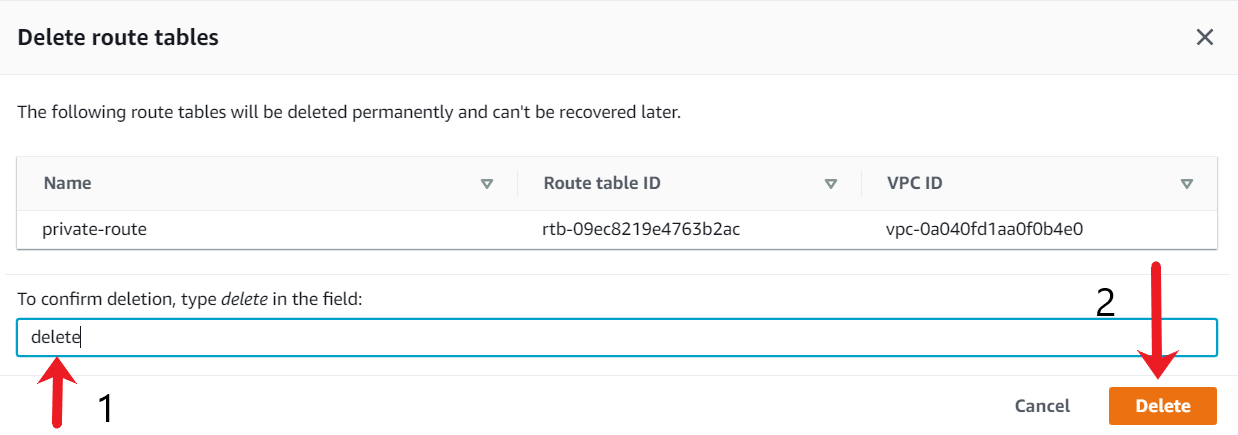
De-associate all the subnets and then choose Save associations.



Once all subnets are de-associated, choose your custom route table from the main console, select Actions, and choose Delete route table.



On the confirmation window, type delete in the confirmation window and click on Delete.



Mark task as complete

Report issue

