AWS Operations Playbook

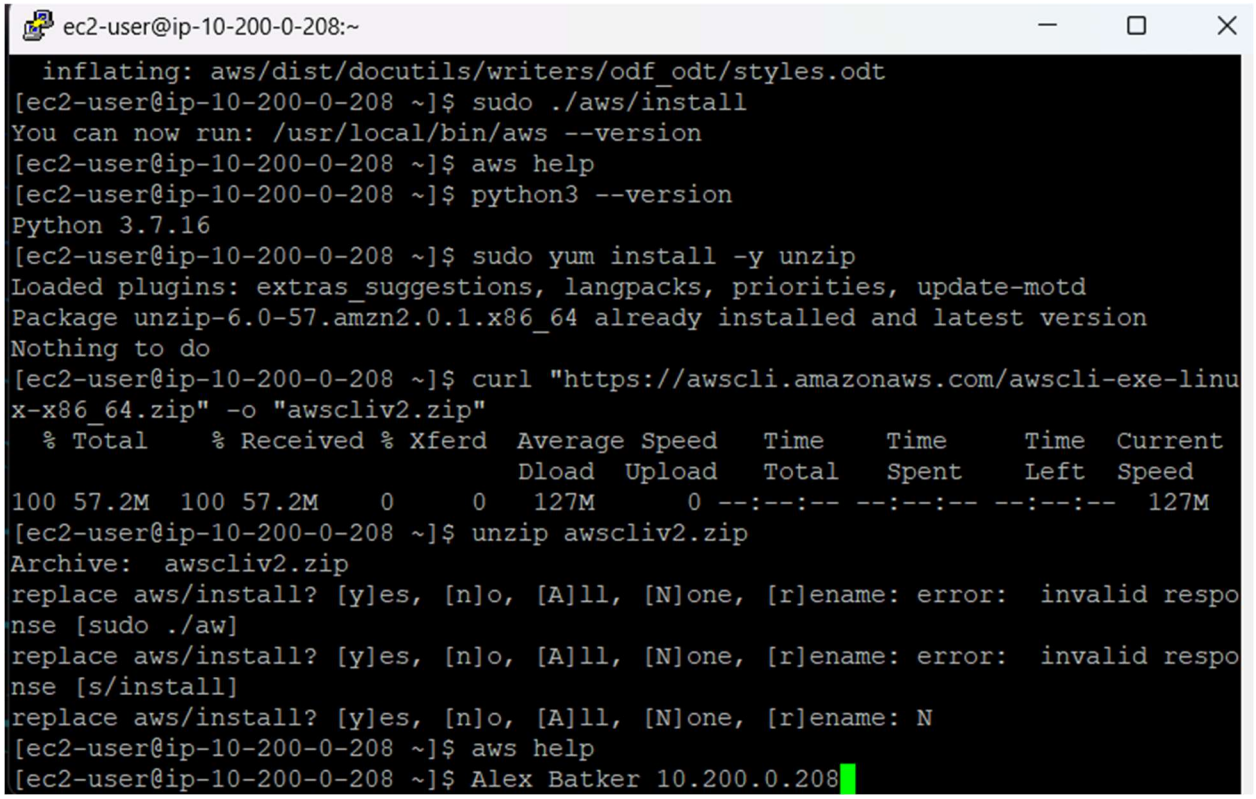
Alexander Batker

2024

Contents

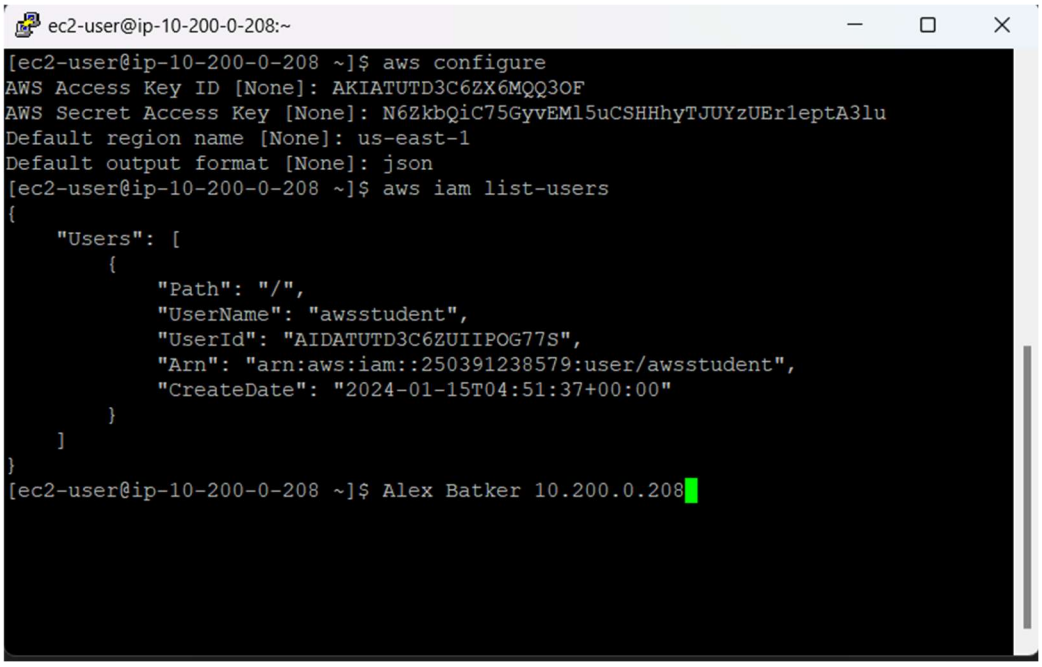
How to connect to the Mom & Pop Cafe Test EC2 instance

1. Ensure you have a copy of the ppk/pem file used to authenticate with your instance
2. Open putty and configure the connection to the following settings
3. Connection - Seconds between keepalives - Set to 30
4. Add the public IPv4 address of the EC2 instance to the hostname field
5. Add the ppk/pem file to the connection
6. Click on open and use the user "ec2-user" to connect to the instance



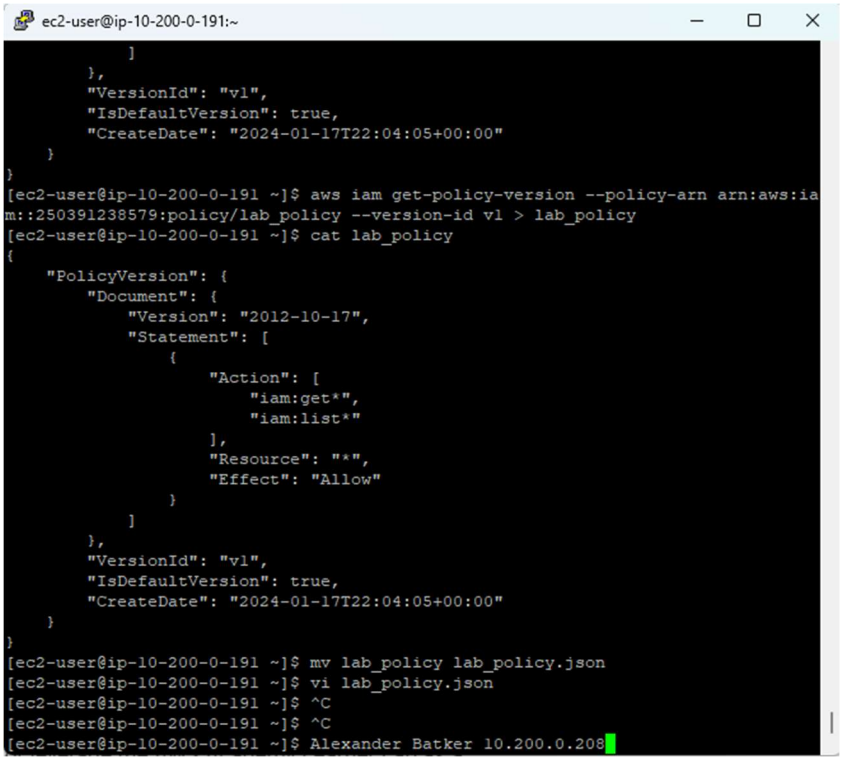
How to use the AWS CLI to connect to your AWS account

1. Verify Python is installed and run ‘sudo yum install -y unzip’
2. Download and install AWS CLI
    a. curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
    b. unzip awscliv2.zip
    c. sudo ./aws/install
3. Ensure you have both the AWS Access Key ID and AWS Secret Access Key
4. Paste both keys in their respective order and then enter the region name and default output format (json)
5. Verify the account by typing ‘aws iam list-users’

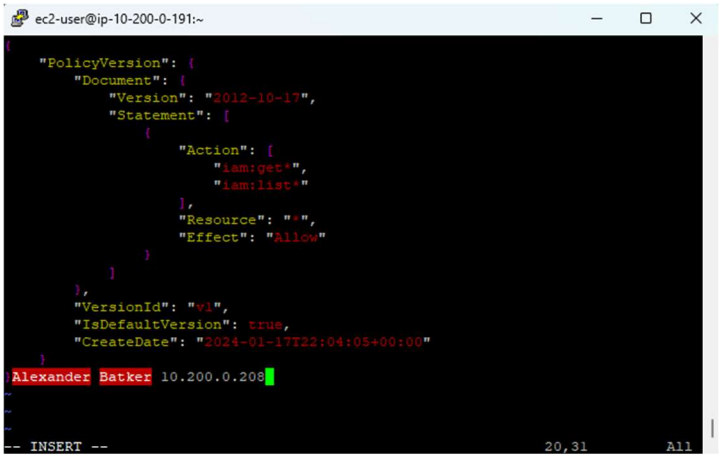


How to make a modification to the lab policy using the AWS CLI

1. Find policy by using scope ‘aws iam list-policies –scope Local’
2. Note the policy *Arn* to get the policy version and enter this command: aws iam get-policy --policy-arn <*Arn*>
    a. Note the version id for the next portion of the play.
3. Get policy version to output the json file to display ‘aws iam get-policy-version --policy-arn <*Arn*> --version-id <*id*>
4. Now verifying the correct output, pipe the output of that command to a new file with a .json extension.

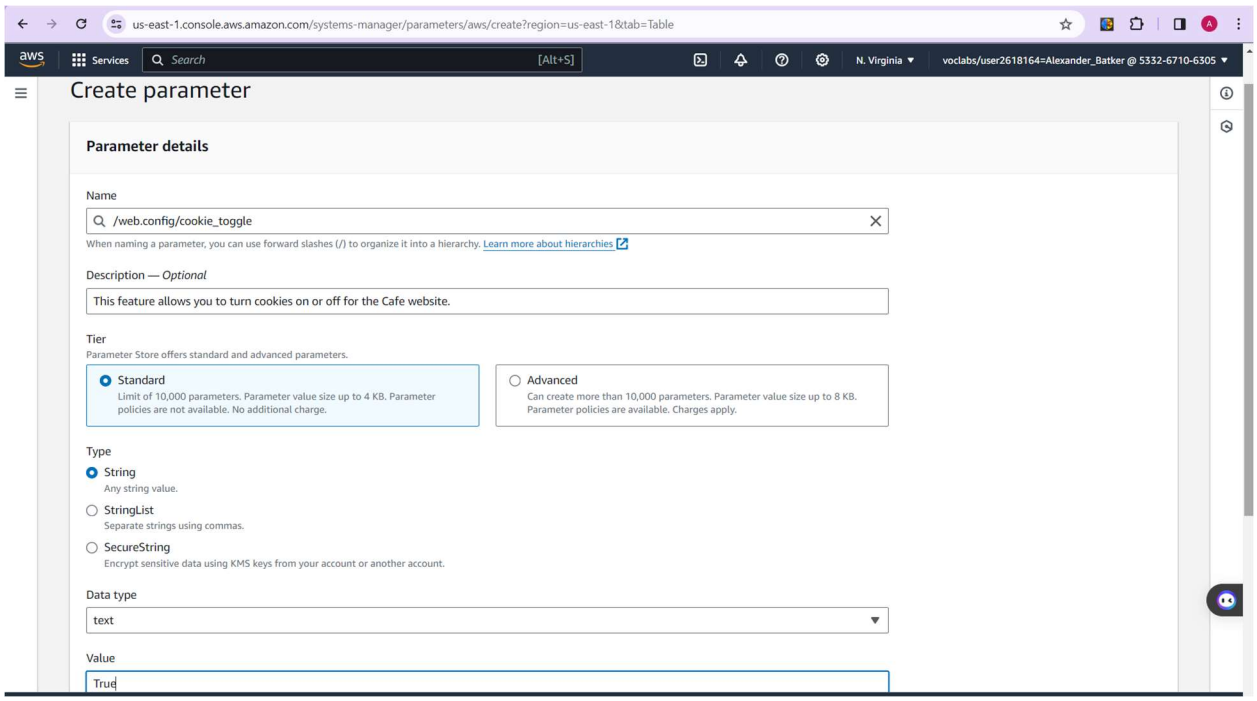


5. To make a modification in CLI, type ‘vi lab_policy.json’ to enter editing mode.



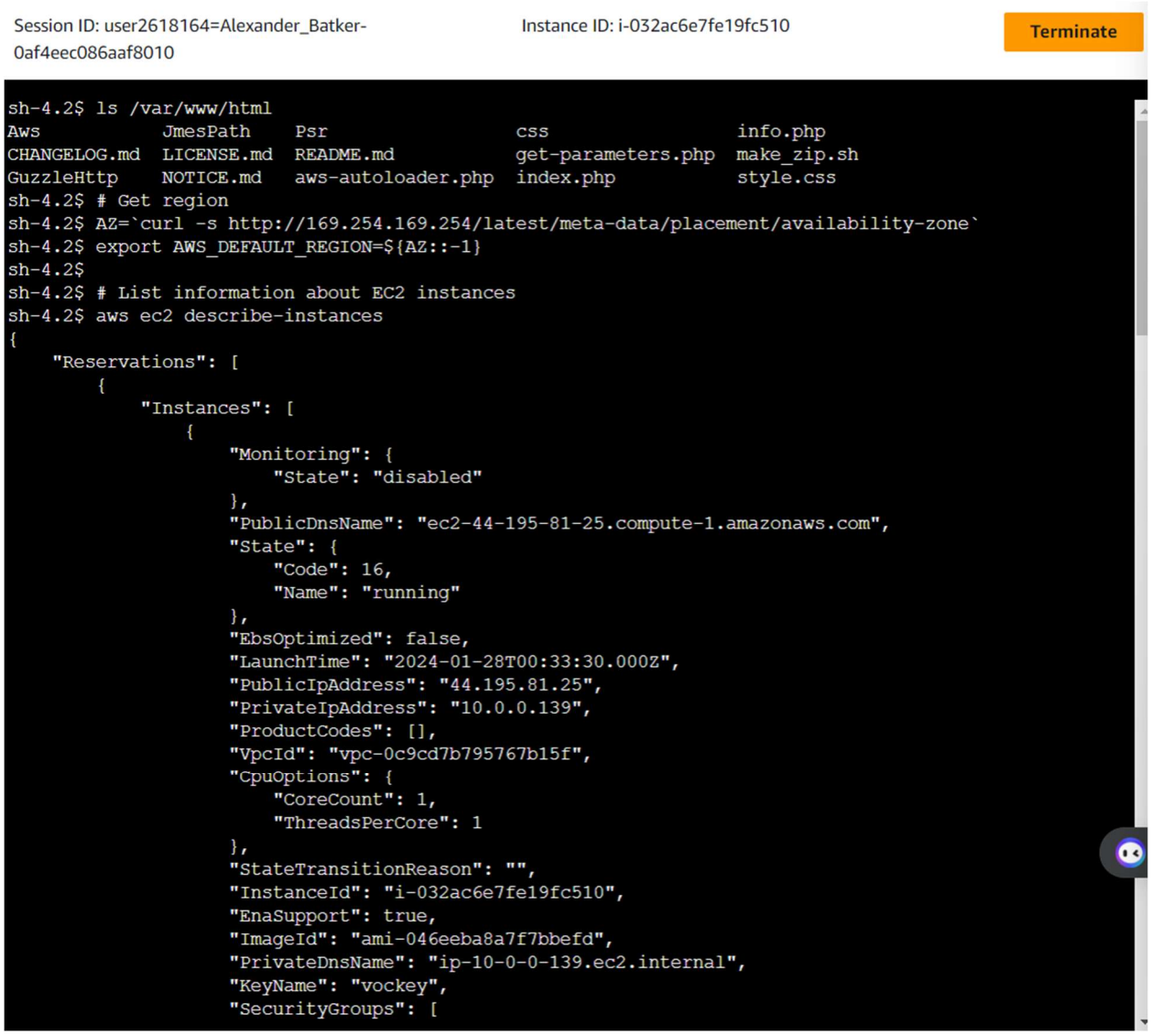
How to add a parameter to the parameter store for allowing cookies on the website

1. Ensure you are in the Systems Manager AWS service and select Parameter Store under Application Management
2. Click ‘Create Parameter’ and configure:
    a. Name: /web.config/cookie_toggle
    b. Description: This feature allows you to turn cookies on or off for the Café website
    c. Value: True



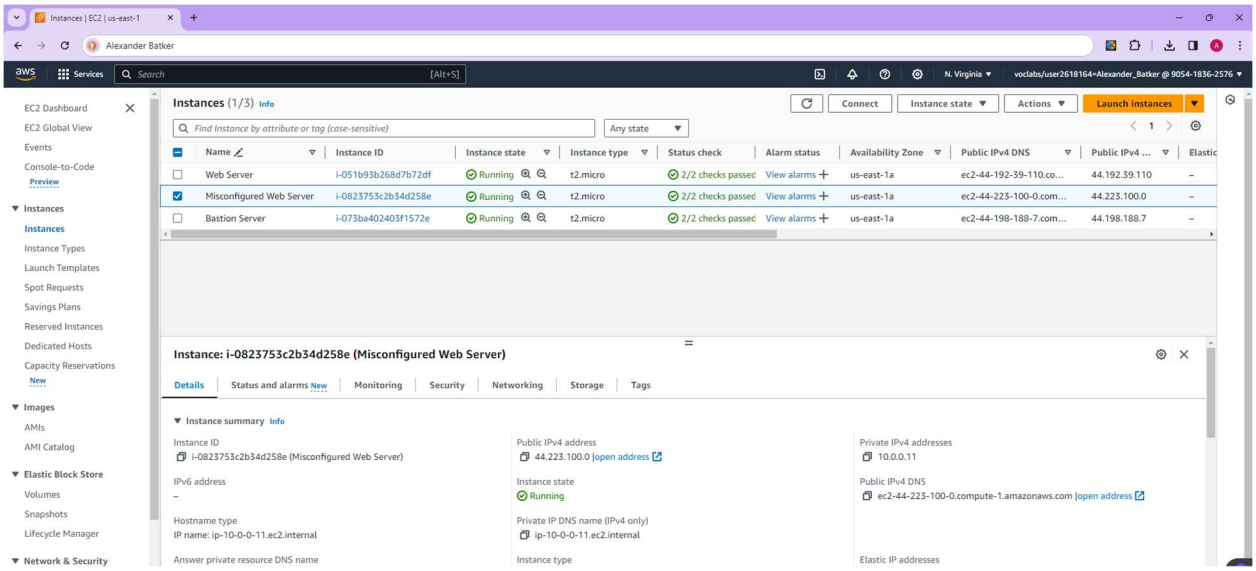
How to connect to an EC2 instance to describe instances

1. Ensure you are in AWS Systems Manager Service, on the left pane navigate to Session Manager in the management console.
2. Click ‘Start Session’ and selected the respective instance, then click ‘Start session’
3. Run the following command in the CLI ‘aws ec2 describe-instances’



How to launch an EC2 instance with Amazon Linux 2, t1.micro

1. Logged into the AWS console: Select EC2 from services and check ‘Misconfigured Web Server’
2. Note the public IPv4 Address and Public IPv4 DNS Name



3. The SSH connection did not work. Check inbound rules by clicking on the security group of the Misconfigured Web Server
4. Click edit inbound rules and add SSH with port range 22 and Protocol TCP and Save Rules.
5. Open Putty.exe and ensure the public ipv4 address is copied and entered into the ‘Session’ ip address for connection. Use the proper PPK downloaded and insert it into Credentials -> SSH -> Auth in the private key section. Click open and Accept

How to fix a misconfigured web server with (_____) issue

1. Try to open the public ip address of the web server and see that it does not appear after taking the time to load
2. Checked the cloud init logs with ‘cat /var/log/cloud-init-output.log’ and found a user script error

```
Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
2024-02-05 03:55:43,381 - cc_scripts_user.py[WARNING]: Failed to run module scripts-user (scripts in /var/lib/cloud/instance/scripts)
2024-02-05 03:55:43,381 - util.py[WARNING]: Running module scripts-user (<module 'cloudinit.config.cc_scripts_user' from '/usr/lib/python3.9/site-packages/cloudinit/config/cc_scripts_user.
y'>) failed
Cloud-init v. 22.2.2 finished at Mon, 05 Feb 2024 03:55:43 +0000. Datasource DataSourceEc2.  Up 37.54 seconds
[ec2-user@ip-10-0-0-11 ~]$ ^C
[ec2-user@ip-10-0-0-11 ~]$ ^C
[ec2-user@ip-10-0-0-11 ~]$ sudo cloud-init status
status: error
[ec2-user@ip-10-0-0-11 ~]$ cat /var/lib/cloud/instance/scripts
cat: /var/lib/cloud/instance/scripts: Is a directory
[ec2-user@ip-10-0-0-11 ~]$ ls /var/lib/cloud/instance/scripts
part-001
[ec2-user@ip-10-0-0-11 ~]$ cat part-001
cat: part-001: No such file or directory
[ec2-user@ip-10-0-0-11 ~]$ cat /var/lib/cloud/instance/scripts/part-001
cat: /var/lib/cloud/instance/scripts/part-001: Permission denied
[ec2-user@ip-10-0-0-11 ~]$
```

3. Ran these lines of code and found the issue in the user script was a typo.

```
[ec2-user@ip-10-0-0-11 ~]$ sudo cat /var/lib/cloud/instance/scripts/part-001
#!/bin/bash
yum install -y httpd php
/usr/bin/systemctl enable httpd
/usr/bin/systemctl start httpdd 2>/tmp/errors.txt
[ec2-user@ip-10-0-0-11 ~]$ ^C
[ec2-user@ip-10-0-0-11 ~]$ sudo /var/lib/cloud/instance/scripts/part-001
Last metadata expiration check: 1:01:34 ago on Mon Feb  5 03:55:31 2024.
Package httpd-2.4.58-1.amzn2023.x86_64 is already installed.
Package php8.2-8.2.9-1.amzn2023.0.3.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

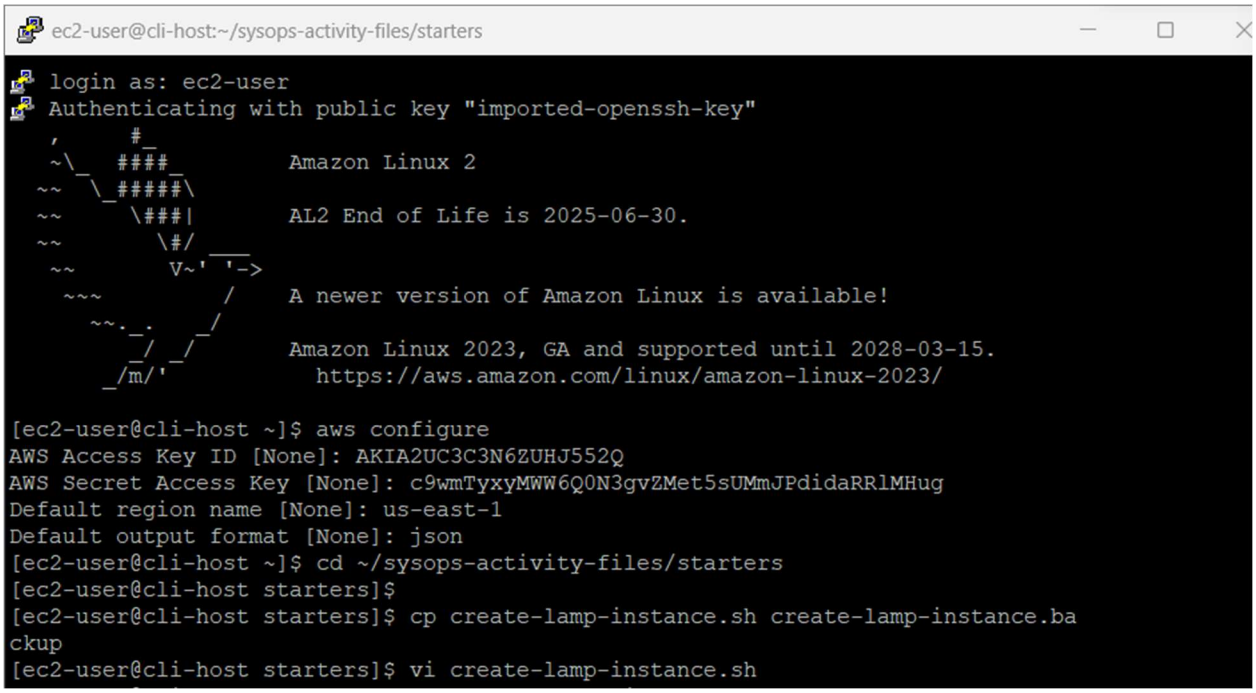
4. Run vi editor and fix the type. Type ‘i’ to go into insert mode and make the change. Then type ‘esc’ then ‘:wq’ .
5. Ensure the server is actually properly running and active by checking its status.

```
[ec2-user@ip-10-0-0-11 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Mon 2024-02-05 05:06:51 UTC; 9s ago
       Docs: man:httpd.service(8)
   Main PID: 27730 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1114)
     Memory: 13.2M
        CPU: 68ms
     CGroup: /system.slice/httpd.service
             ├─27730 /usr/sbin/httpd -DFOREGROUND
             ├─27732 /usr/sbin/httpd -DFOREGROUND
             ├─27733 /usr/sbin/httpd -DFOREGROUND
             ├─27734 /usr/sbin/httpd -DFOREGROUND
             └─27735 /usr/sbin/httpd -DFOREGROUND

Feb 05 05:06:51 ip-10-0-0-11.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 05 05:06:51 ip-10-0-0-11.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 05 05:06:51 ip-10-0-0-11.ec2.internal httpd[27730]: Server configured, listening on: port 80
[ec2-user@ip-10-0-0-11 ~]$
```

6. Copy and paste the ipv4 dns address in another tab and verify the success!

How to change the AMI instance on the create-lamp-instance.sh script

1. Connect to an AWS EC2 instance via SSH and putty with proper credentials
2. Look at the script and open it in vi:



3. Type ':set number' in vi to see code lines labeled Navigate to #Get AMI ID on line #38

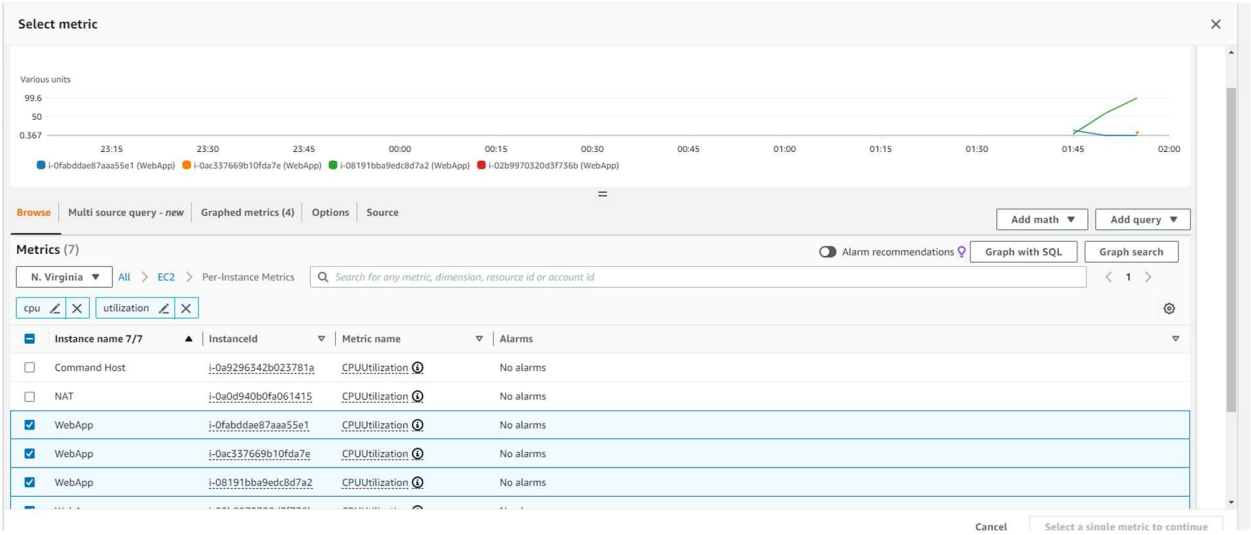
4. Type 'i' and change the imageID. Type ':wq' to save and exit after typing 'esc' when done changing the AMI instance.

How to create an Auto Scaling Group in the AWS UI

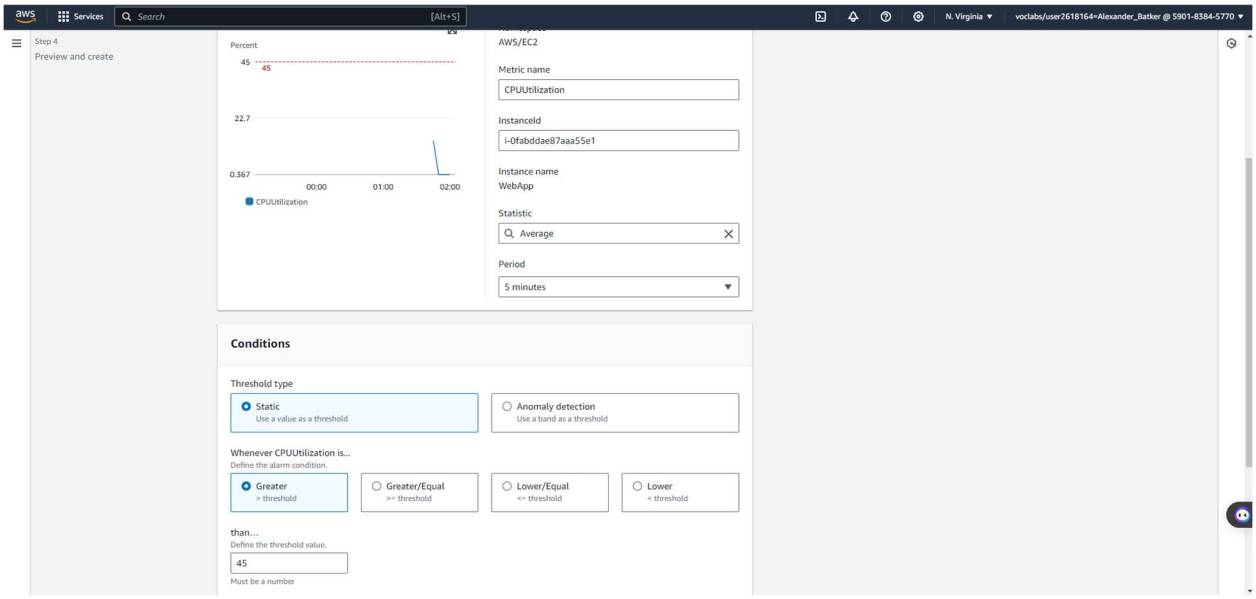
Step Scaling

Minimum 5, Maximum 10. 7 Nodes.

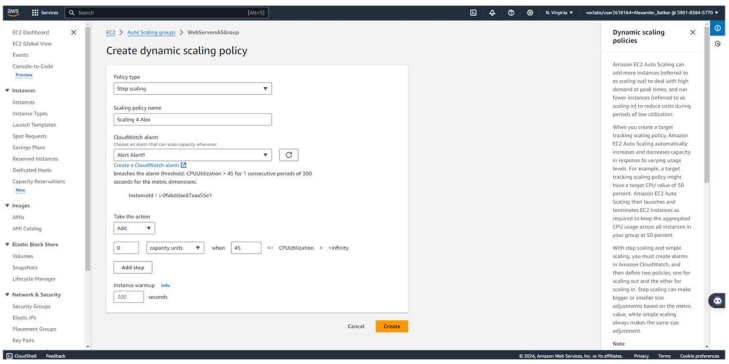
1. Go to AWS services, EC2, then on the left panel go down to Auto Scaling Groups and click Create Auto Scaling Group
2. Choose template or create a launch template (usually already template by this time)
3. On the next page, choose correct VPC and corresponding private subnets
4. On the next page, choose attach to existing load balancer and select corresponding target group
5. Finish Creation of Auto Scaling Group and go to Automatic Scaling Tab and create new scaling
6. Create CloudWatch metric to monitor CPU utilization on correct server



7. Choose conditions for CloudWatch metric

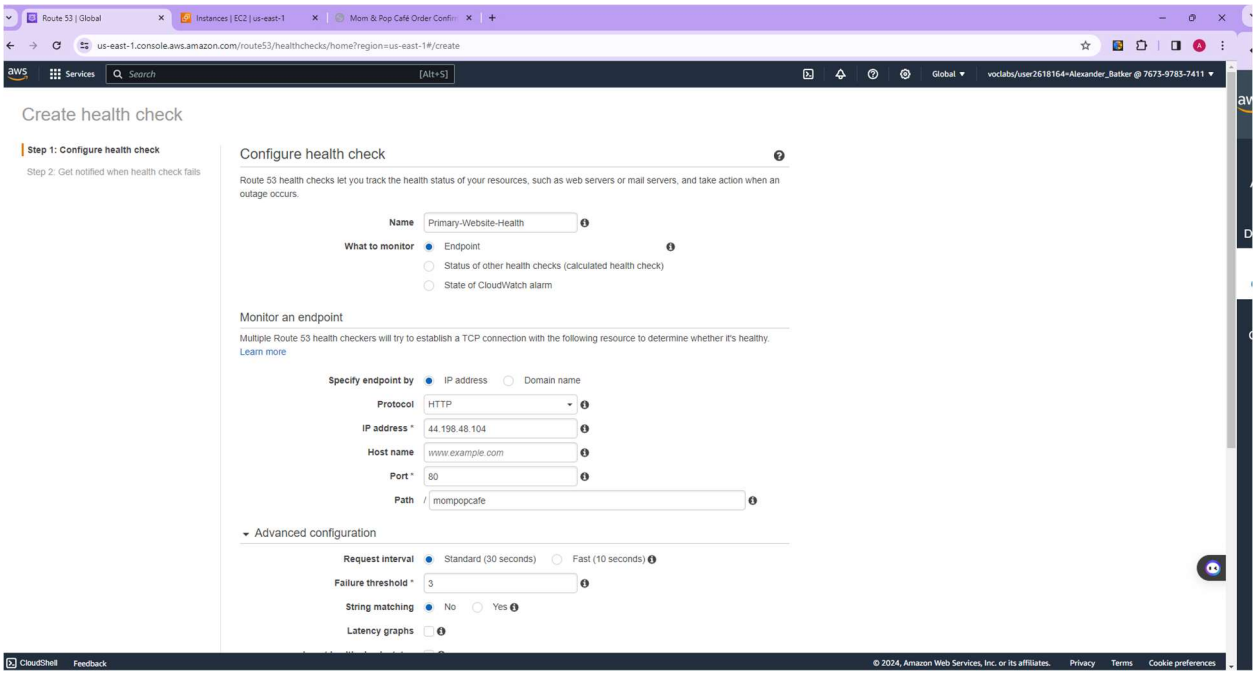


8. Go back to creating a dynamic scaling policy and choose Step scaling then the CloudWatch Alarm



How to create a Route 53 health check

1. Navigate to AWS Services in the AWS UI and click health check, then click create health check.
2. Configure health check to monitor HTTP traffic .

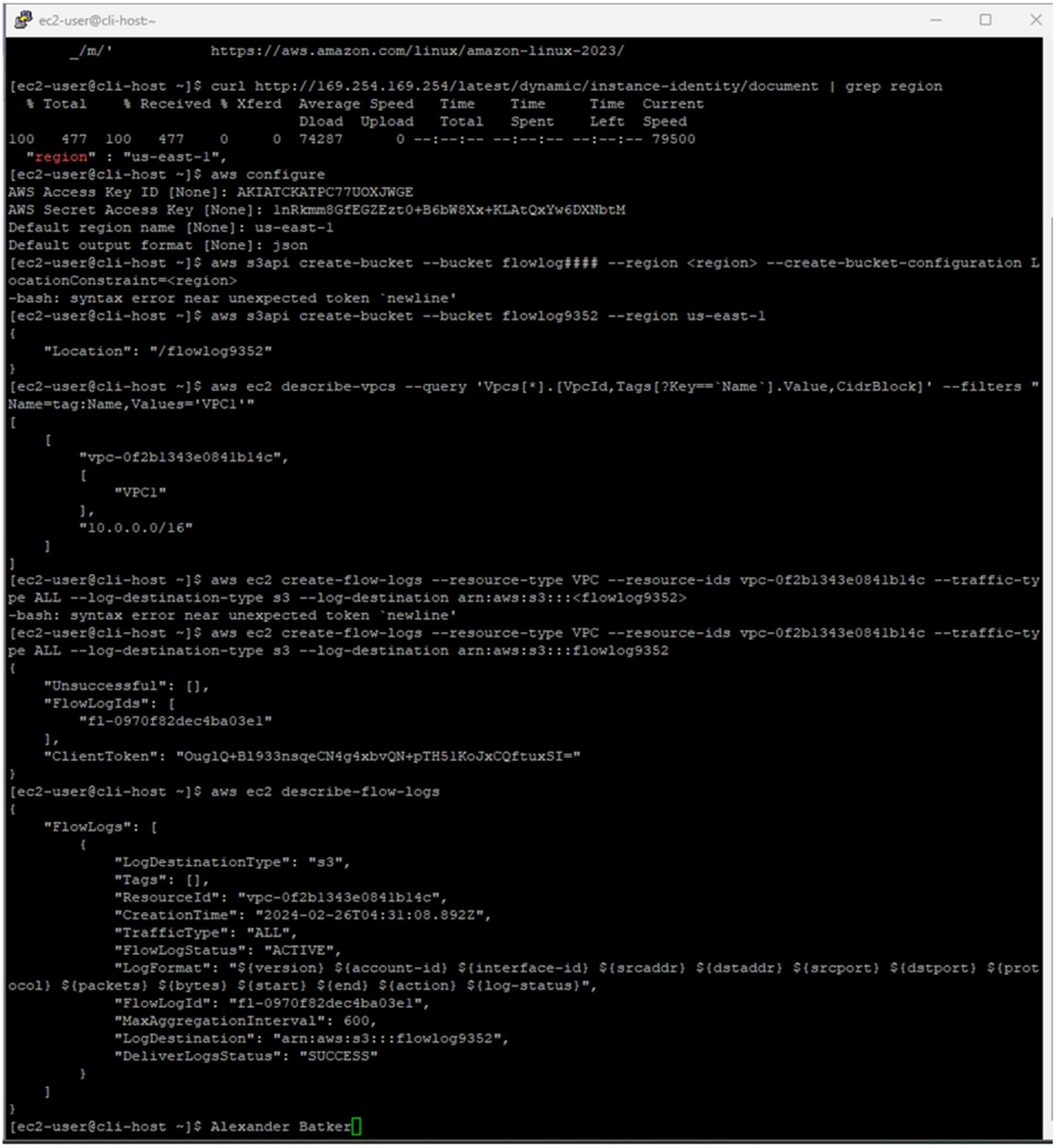


3. Click next and then check yes on create alarm to send new SNS topic to accessible email.
4.

How to enable VPC Flow Logs via the command line interface

1. Connect to the CLI via putty with the proper ipv4 address and set connection settings to 30 seconds. Have the PPK key handy to paste into the SSH -> Credentials -> Auth for keys.
2. Update aws with aws configure command. Enter access key, secret access key, default region name, and default output format (which is json)
3. Create an S3 bucket with this command ‘aws s3api create-bucket --bucket flowlog#### --region <region> --create-bucket-configuration LocationConstraint=<region>’ except replace the # signs with a memorable number and insert region the instance is in. This will hold the flow logs.
4. Next, enable VPC flow logs on the proper VPC with the following two commands entered sequentially:
    a. aws ec2 describe-vpcs --query 'Vpcs[*].[VpcId,Tags[?Key==`Name`].Value,CidrBlock]' --filters "Name=tag:Name,Values='VPC1'"
    b. aws ec2 create-flow-logs --resource-type VPC --resource-ids <vpc-id> --traffic-type ALL --log-destination-type s3 --log-destination arn:aws:s3:::<flowlog####>
        i. First command to find VPC ID, then insert VPC ID into second command along with the correct flowlog#### created.
5. Check if the flow logs were properly created with this cli command:

a. aws ec2 describe-flow-logs



How to troubleshoot network connectivity on an instance

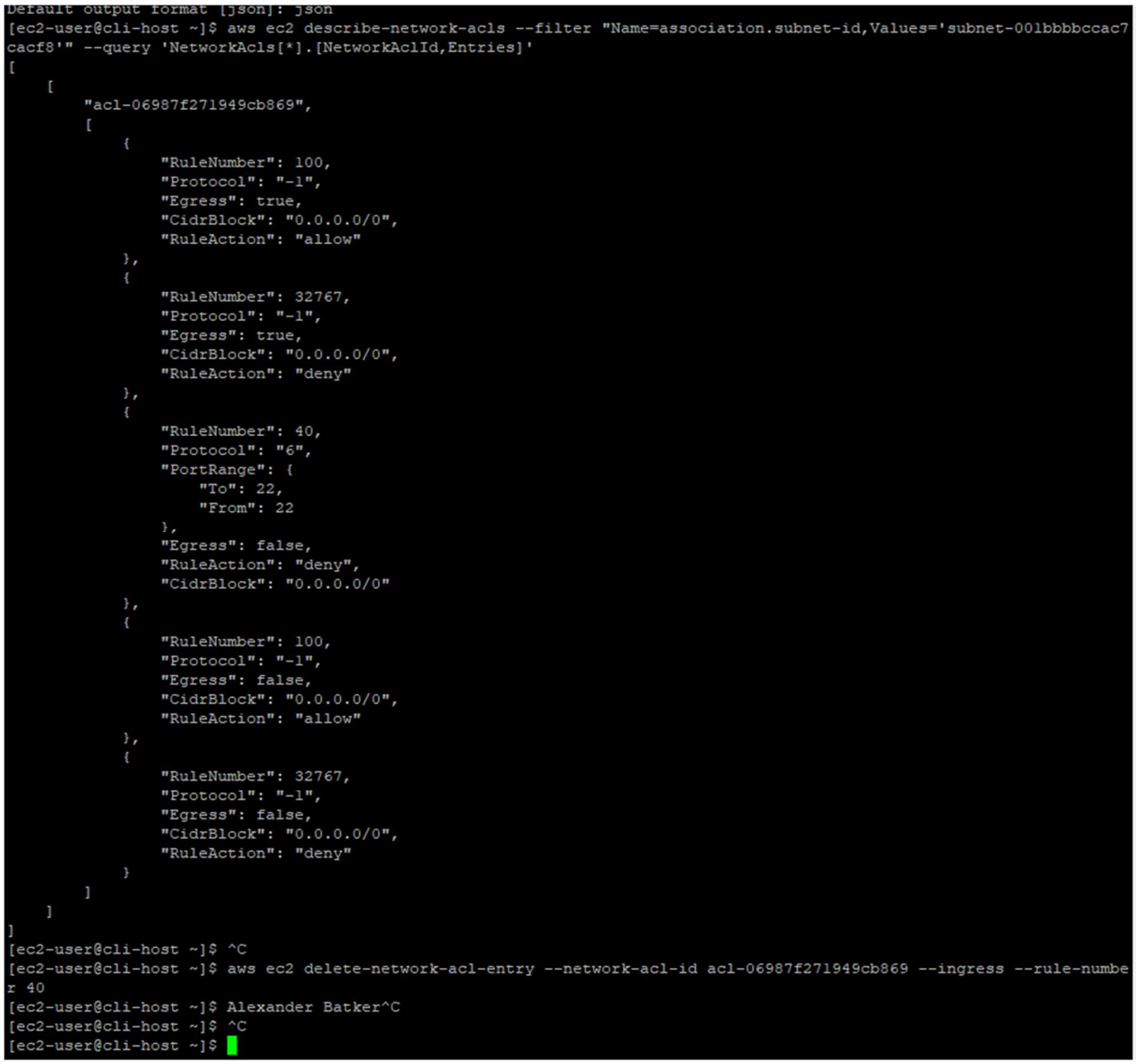
1. While connected to the cli-host, download nmap and then check the ports of the webserverIP with these two commands:
    a. sudo yum install -y nmap
    b. nmap <webserveripaddress>
2. Since there are no open ports, check the security group affiliation with the following command (webserverSG is entered as an option)
    a. aws ec2 describe-security-groups --group-ids sg-07fe72c2537b76a3d

3. This shows that there is no route that allows outbound traffic to the internet. Now check the route table with this command:
    a. aws ec2 describe-route-tables --filter "Name=association.subnet-id,Values='subnet-001bbbbccac7cacf8'"
4. There is only a local route specified in the associated route table (as shown below), so a route must be created in order to access the webserver. The gateway ID must be found using this command and then the following command creates the route.
    a. aws ec2 describe-internet-gateways
    b. aws ec2 create-route --route-table-id rtb-04b02ef95c9e269d7 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-09e2d8be677cb858f

```
[ec2-user@cli-host ~]$ aws ec2 create-route --route-table-id rtb-04b02ef95c9e269d7 --destination-cidr-block 0.0.0.0/
 --gateway-id igw-09e2d8be677cb858f
{
    "Return": true
}
[ec2-user@cli-host ~]$ igw-09e2d8be677cb858f
-bash: igw-09e2d8be677cb858f: command not found
[ec2-user@cli-host ~]$ aws ec2 describe-internet-gateways
{
    "InternetGateways": [
        {
            "OwnerId": "211125762239",
            "Tags": [
                {
                    "Value": "VPC2InternetGateway",
                    "Key": "aws:cloudformation:logical-id"
                },
                {
                    "Value": "arn:aws:cloudformation:us-east-1:211125762239:stack/c104021a240432015949859t1w211125762
239/0bd2c200-d45e-11ee-b85c-1212618374a5",
                    "Key": "aws:cloudformation:stack-id"
                },
                {
                    "Value": "VPC2 Internet Gateway",
                    "Key": "Name"
                },
                {
                    "Value": "c104021a240432015949859t1w211125762239",
                    "Key": "aws:cloudformation:stack-name"
                },
                {
                    "Value": "c104021a240432015949859t1w211125762239",
                    "Key": "cloudlab"
```

5. Since the web server still cannot be connected to, the network access control list associated with this subnet should be examined. After running these commands below, we can see that rule number 40 is denying ssh traffic and we need to delete that rule. Command ‘a’ shows the problem and command ‘b’ deletes that NACL rule.
    a. aws ec2 describe-network-acls --filter "Name=association.subnet-id,Values='subnet-001bbbbccac7cacf8'" --query 'NetworkAcls[*].[NetworkAclId,Entries]'

b. aws ec2 delete-network-acl-entry --network-acl-id acl-06987f271949cb869 --ingress --rule-number 40



How to take a snapshot of an EBS volume

1. Connect to the command host of the processor via CLI
2. Know the volumeID and instance ID to stop the instance associated with the S3 bucket.
    a. aws ec2 stop-instances --instance-ids INSTANCE-ID
    b. Replace INSTANCE-ID with the ID
3. Create the snapshot with this command
    a. aws ec2 create-snapshot --volume-id VOLUME-ID
    b. aws ec2 wait snapshot-completed --snapshot-id SNAPSHOT-ID
        i. This command shows when completed

```
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'
{
    "VolumeId": "vol-008957c19a4418882"
}
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].InstanceId'
"i-0f5f420e166b1a06f"
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 stop-instances --instance-ids ^C
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 stop-instances --instance-ids i-0f5f420e166b1a06f
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-0f5f420e166b1a06f",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 wait instance-stopped --instance-id INSTANCE-ID

Waiter InstanceStopped failed: An error occurred (InvalidInstanceID.Malformed): Invalid id: "INSTANCE-ID"
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 wait instance-stopped --instance-id ^C
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 wait instance-stopped --instance-id i-0f5f420e166b1a06f
[ec2-user@ip-10-5-0-204 ~]$
[ec2-user@ip-10-5-0-204 ~]$
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 create-snapshot --volume-id VOLUME-ID

An error occurred (InvalidParameterValue) when calling the CreateSnapshot operation: Value (VOLUME-ID) for parameter volumeId is invalid. Expected: 'vol-...'.
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 create-snapshot --volume-id ^C
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 create-snapshot --volume-id vol-008957c19a4418882
{
    "Description": "",
    "Encrypted": false,
    "OwnerId": "992382826753",
    "Progress": "",
    "SnapshotId": "snap-03af8de8e02a61e54",
    "StartTime": "2024-03-03T20:48:34.364Z",
    "State": "pending",
    "VolumeId": "vol-008957c19a4418882",
    "VolumeSize": 8,
    "Tags": []
}
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 wait snapshot-completed --snapshot-id
^C
[ec2-user@ip-10-5-0-204 ~]$ aws ec2 wait snapshot-completed --snapshot-id snap-03af8de8e02a61e54
[ec2-user@ip-10-5-0-204 ~]$
```

4. Restart the instance and utilize these commands:
    a. aws ec2 start-instances --instance-ids INSTANCE-ID
    b. aws ec2 wait instance-running --instance-id INSTANCE-ID

How to synchronize files using the command line (aws s3api and aws s3)

1. Log into the AWS CLI and ensure you have files that are relatively disposable for this experiment.
2. Utilize these commands in order for syncing the files with your S3 bucket:

```
[ec2-user@ip-10-5-0-43 ~]$ aws s3api list-object-versions --bucket alexbatkeraws --prefix files/file1.txt
{
    "Versions": [
        {
            "ETag": "\"b76b2b775023e60be16bc332496f8409\"",
            "Size": 30318,
            "StorageClass": "STANDARD",
            "Key": "files/file1.txt",
            "VersionId": "cqrrXnINRIMirUo7z3xr5Ck2RpoyChf9",
            "IsLatest": false,
            "LastModified": "2024-03-03T21:14:55.000Z",
            "Owner": {
                "DisplayName": "awslabsc0w733771lt1708403252",
                "ID": "f6df3b8a0331db23380a20d8671a5964778b30559d5a3f796a74c64cccbe6a97"
            }
        }
    ],
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "awslabsc0w733771lt1708403252",
                "ID": "f6df3b8a0331db23380a20d8671a5964778b30559d5a3f796a74c64cccbe6a97"
            },
            "Key": "files/file1.txt",
            "VersionId": "YMv5c79ev0mHNtT0YKS1A.avdFR_Gr8_",
            "IsLatest": true,
            "LastModified": "2024-03-03T21:16:08.000Z"
        }
    ],
    "RequestCharged": null
}
[ec2-user@ip-10-5-0-43 ~]$ aws s3api get-object --bucket S3-BUCKET-NAME --key files/file1.txt --version-id VERSION-ID files/file1.txt

An error occurred (NoSuchBucket) when calling the GetObject operation: The specified bucket does not exist
[ec2-user@ip-10-5-0-43 ~]$ aws s3api get-object --bucket alexbatkeraws --key files/file1.txt --version-id VERSION-ID files/file1.txt

An error occurred (InvalidArgument) when calling the GetObject operation: Invalid version id specified
[ec2-user@ip-10-5-0-43 ~]$ aws s3api get-object --bucket alexbatkeraws --key files/file1.txt --version-id VERSION-ID files/file1.txt^C
[ec2-user@ip-10-5-0-43 ~]$ aws s3api get-object --bucket alexbatkeraws --key files/file1.txt --version-id cqrrXnINRIMirUo7z3xr5Ck2RpoyChf9 files/file1.txt
{
    "AcceptRanges": "bytes",
    "LastModified": "Sun, 03 Mar 2024 21:14:55 GMT",
    "ContentLength": 30318,
    "ETag": "\"b76b2b775023e60be16bc332496f8409\"",
    "VersionId": "cqrrXnINRIMirUo7z3xr5Ck2RpoyChf9",
    "ContentType": "text/plain",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
[ec2-user@ip-10-5-0-43 ~]$ ls files
file1.txt  file2.txt  file3.txt
[ec2-user@ip-10-5-0-43 ~]$ aws s3 sync files s3://S3-BUCKET-NAME/files/
fatal error: An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
[ec2-user@ip-10-5-0-43 ~]$ aws s3 sync files s3://alexbatkeraws/files/
upload: files/file1.txt to s3://alexbatkeraws/files/file1.txt
[ec2-user@ip-10-5-0-43 ~]$ aws s3 ls s3://alexbatkeraws/files/
2024-03-03 21:19:40      30318 file1.txt
2024-03-03 21:14:55      43784 file2.txt
2024-03-03 21:14:55      96675 file3.txt
[ec2-user@ip-10-5-0-43 ~]$ Alexander Batker
```

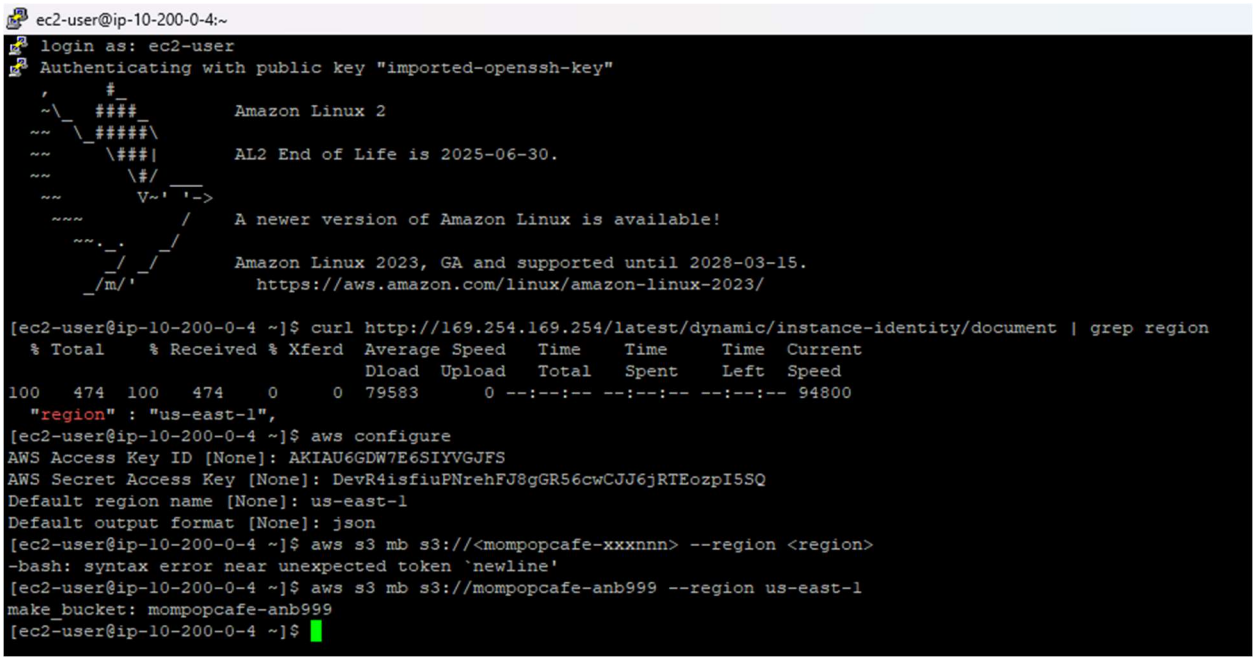
a. ‘aws s3api put-bucket-versioning --bucket S3-BUCKET-NAME --versioning-configuration Status=Enabled’ This command enables versioning for files on the S3 bucket
b. ‘aws s3 sync files s3://S3-BUCKET-NAME/files/’ This syncs contents of files with S3 bucket
c. ‘aws s3 ls s3://S3-BUCKET-NAME/files/’ This confirms initial state of files in S3 bucket

3. Experiment and delete a file or two and sync:
    a. ‘rm files/file1.txt’
    b. ‘ aws s3 sync files s3://S3-BUCKET-NAME/files/ --delete’
    c. ‘aws s3 ls s3://S3-BUCKET-NAME/files/’ This verifies deletion of file or files
4. Now, we recover the old file with these commands:
    a. ‘aws s3api list-object-versions --bucket S3-BUCKET-NAME --prefix files/file1.txt’
    b. ‘aws s3api get-object --bucket S3-BUCKET-NAME --key files/file1.txt --version-id VERSION-ID files/file1.txt’
    c. ‘aws s3 sync files s3://S3-BUCKET-NAME/files/’
    d. ‘aws s3 ls s3://S3-BUCKET-NAME/files/’

```
[ec2-user@ip-10-5-0-43 ~]$ wget https://aws-tc-largeobjects.s3-us-west-2.amazona
ws.com/CUR-TF-200-RESOPS/lab5vocareum/files.zip
--2024-03-03 21:13:05--  https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/
CUR-TF-200-RESOPS/lab5vocareum/files.zip
Resolving aws-tc-largeobjects.s3-us-west-2.amazonaws.com (aws-tc-largeobjects.s3
-us-west-2.amazonaws.com)... 3.5.84.193, 3.5.85.173, 52.92.194.242, ...
Connecting to aws-tc-largeobjects.s3-us-west-2.amazonaws.com (aws-tc-largeobject
s.s3-us-west-2.amazonaws.com)|3.5.84.193|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72110 (70K) [application/zip]
Saving to: ‘files.zip’

100%[======================================>] 72,110      --.-K/s   in 0.1s

2024-03-03 21:13:06 (559 KB/s) - ‘files.zip’ saved [72110/72110]

[ec2-user@ip-10-5-0-43 ~]$ unzip files.zip
Archive:  files.zip
  inflating: files/file1.txt
  inflating: files/file2.txt
  inflating: files/file3.txt
[ec2-user@ip-10-5-0-43 ~]$ aws s3api put-bucket-versioning --bucket alexbatkeraws --versioning-configuration Status=Enabled
[ec2-user@ip-10-5-0-43 ~]$ aws s3 sync files s3://alexbatkeraws/files/
upload: files/file1.txt to s3://alexbatkeraws/files/file1.txt
upload: files/file2.txt to s3://alexbatkeraws/files/file2.txt
upload: files/file3.txt to s3://alexbatkeraws/files/file3.txt
[ec2-user@ip-10-5-0-43 ~]$ aws s3 ls s3://S3-BUCKET-NAME/files/

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
[ec2-user@ip-10-5-0-43 ~]$ aws s3 ls s3://alexbatkeraws/files/
2024-03-03 21:14:55      30318 file1.txt
2024-03-03 21:14:55      43784 file2.txt
2024-03-03 21:14:55      96675 file3.txt
[ec2-user@ip-10-5-0-43 ~]$ rm files/file1.txt
[ec2-user@ip-10-5-0-43 ~]$ aws s3 sync files s3://S3-BUCKET-NAME/files/ --delete
fatal error: An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
[ec2-user@ip-10-5-0-43 ~]$ aws s3 sync files s3://alexbatkeraws/files/ --delete
delete: s3://alexbatkeraws/files/file1.txt
[ec2-user@ip-10-5-0-43 ~]$ aws s3 ls s3://S3-BUCKET-NAME/files/

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
[ec2-user@ip-10-5-0-43 ~]$ aws s3 ls s3://alexbatkeraws/files/
2024-03-03 21:14:55      43784 file2.txt
2024-03-03 21:14:55      96675 file3.txt
[ec2-user@ip-10-5-0-43 ~]$ Alexander Batker
```

How to create a S3 bucket via the CLI

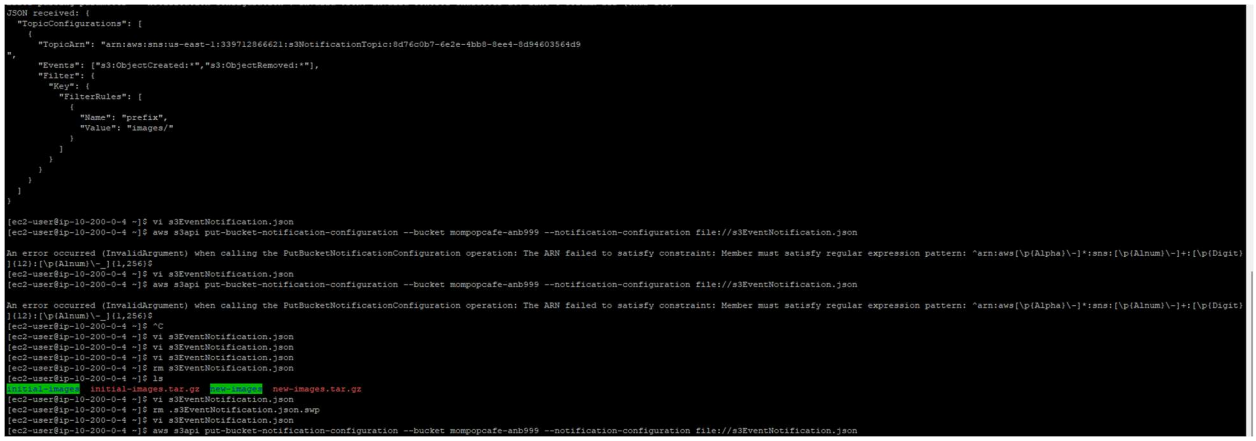
1. Log into AWS CLI with proper credentials, SSH key, with known region and bucket name ready
2. Command to create S3 bucket in CLI:
    a. aws s3 mb s3://<nameofbucket-xxxnnn> --region <region>



How to add an event notification to a S3 bucket

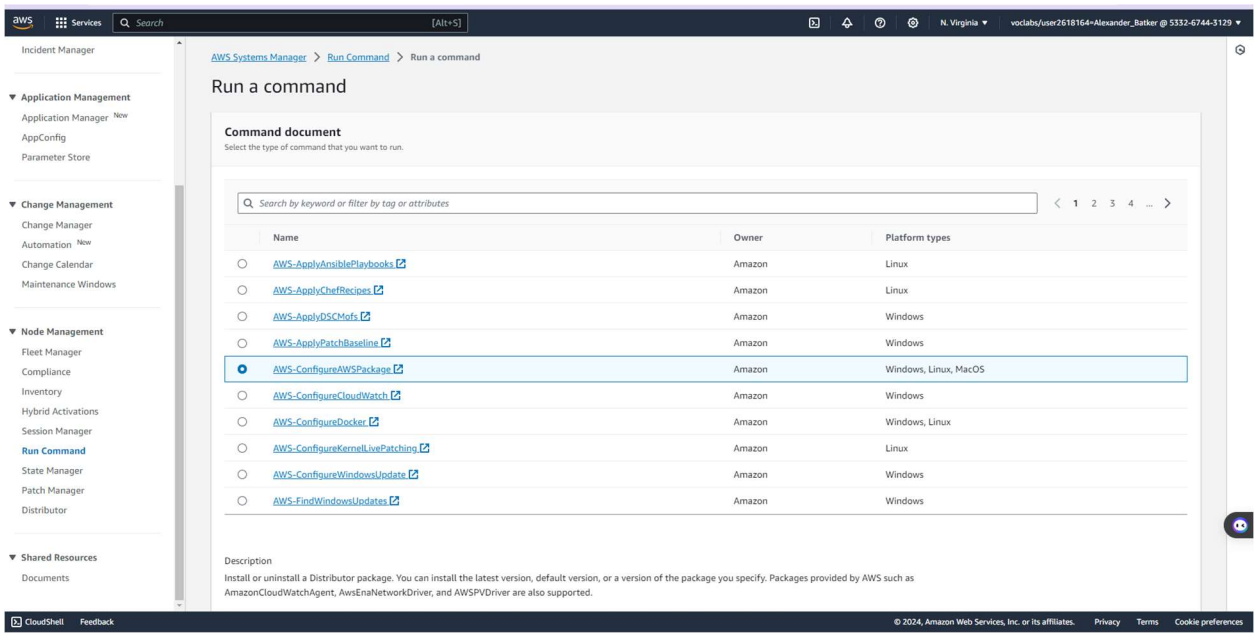
1. Create s3NotificationTopic SNS topic
    a. Navigate to SNS in AWS, then Topics, and create topic. Insert relevant info into proper fields and note the Topic ARN.
2. Grant AWS S3 permission to publish the topic
    a. Using JSON format, please give AWS S3 permission to publish topic with the following methodology:
        i. Go to s3NotificationTopic pane, choose ‘Edit’, expand Access policy – optional section then insert JSON formatted policy
3. Subscribe user to topic
    a. Then, create a subscription by clicking ‘Create Subscription’ and set endpoint as email.
4. Add event notification config to S3 bucket
    a. In AWS CLI, insert command to create and edit event notification:
        i. vi s3EventNotification.json

ii. aws s3api put-bucket-notification-configuration --bucket <uniquebucketname-xxxnnn> --notification-configuration file://s3EventNotification.json



How to install the CloudWatch Agent

1. First, navigate to systems manager -> run command -> run a command and choose ‘AWS-configureawspackage’.



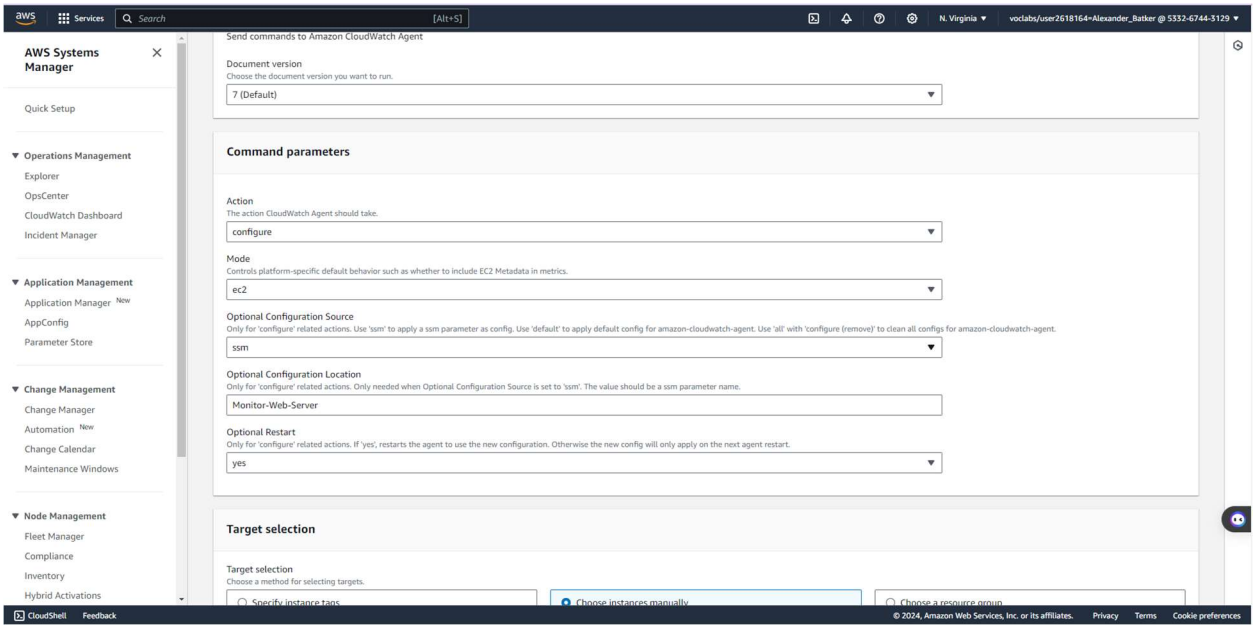
2. Name the cloudwatchagent and choose your target.
3. Click ‘Run’ and wait for the status to switch to ‘Success’. Then, navigate to Parameter Store to create a new parameter.
4. Click ‘Create Parameter’ and name/describe the parameter properly. Then, enter the following configuration under ‘Value’:

```
{
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
```

```
        "log_group_name": "HttpAccessLog",
        "file_path": "/var/log/httpd/access_log",
        "log_stream_name": "{instance_id}",
        "timestamp_format": "%b %d %H:%M:%S"
       },
       {
        "log_group_name": "HttpErrorLog",
        "file_path": "/var/log/httpd/error_log",
        "log_stream_name": "{instance_id}",
        "timestamp_format": "%b %d %H:%M:%S"
       }
      ]
     }
    }
  },
  "metrics": {
   "metrics_collected": {
    "cpu": {
     "measurement": [
      "cpu_usage_idle",
      "cpu_usage_iowait",
      "cpu_usage_user",
      "cpu_usage_system"
     ],
     "metrics_collection_interval": 10,
     "totalcpu": false
    },
    "disk": {
     "measurement": [
      "used_percent",
      "inodes_free"
     ],
     "metrics_collection_interval": 10,
     "resources": [
      "*"
     ]
    },
    "diskio": {
     "measurement": [
      "io_time"
     ],
     "metrics_collection_interval": 10,
     "resources": [
      "*"
     ]
    },
    "mem": {
     "measurement": [
      "mem_used_percent"
     ],
```

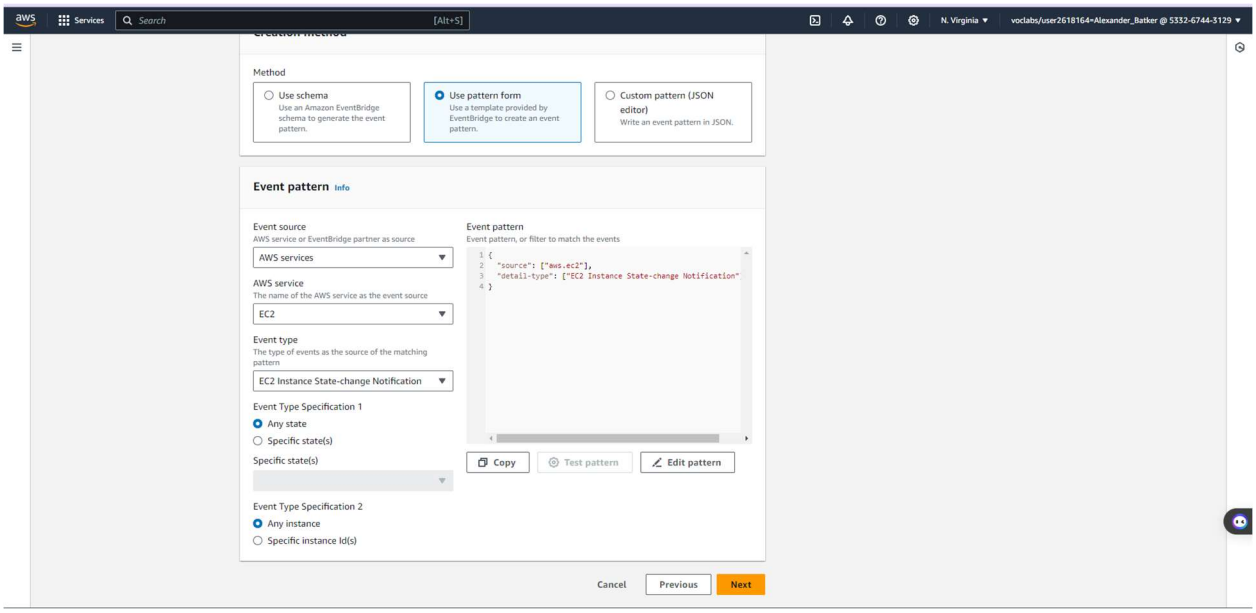
```
      "metrics_collection_interval": 10
    },
    "swap": {
     "measurement": [
      "swap_used_percent"
     ],
     "metrics_collection_interval": 10
    }
   }
  }
}
```

5. Navigate back to ‘Run Command’ and search for ‘AmazonCloudWatch-ManageAgent’ and choose it with these settings:



How to create a CloudWatch Events/CloudWatch EventBridge notification rule

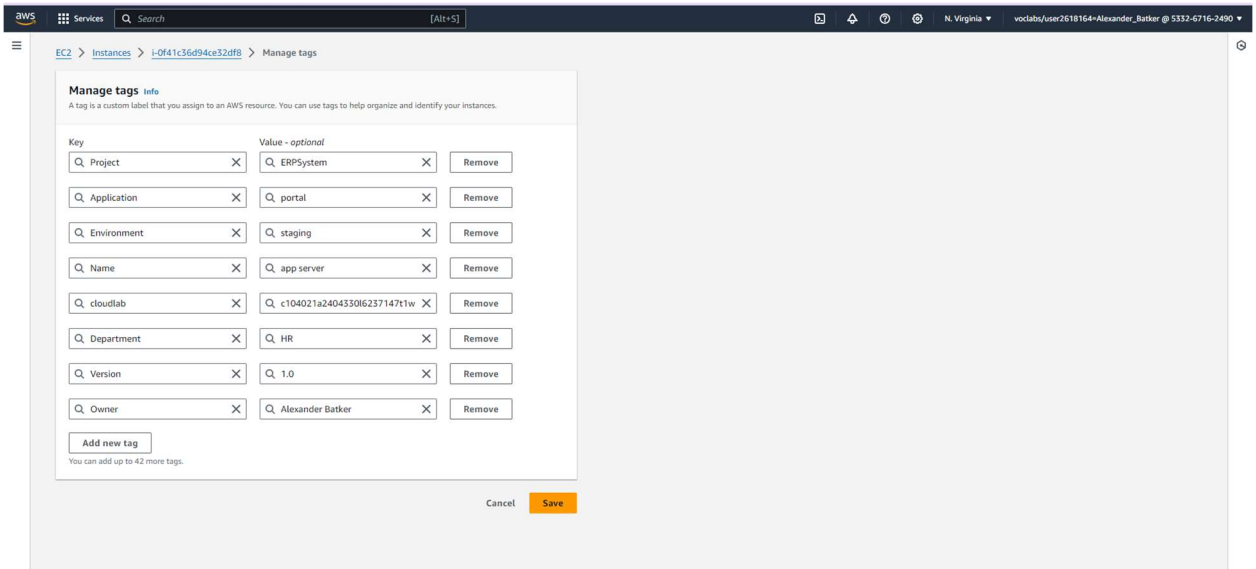
1. In AWS Cloudwatch, in the left pane, choose Rules under Events. -> Create Rule.
2. Choose event pattern pertaining to the what event you would like to be notified for:



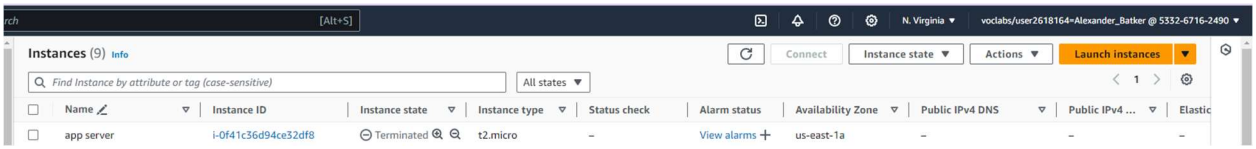
3. Choose target and select SNS topic.
4. Add tags if preferred. Then, ‘Create Rule’.

How to use the prebuilt stopinator script to turn off instances with the tag value of your full name

1. First, ensure the prebuilt stopinator script is in order and you are logged into a command host related to instance(s) in the scenario.
2. Ensure there is a tag with a proper key and full name



3. Run the script with the tag and full name value after the name of the script
4. Go back to the AWS console to check if the instance was terminated.



How to detect drift in a CloudFormation template

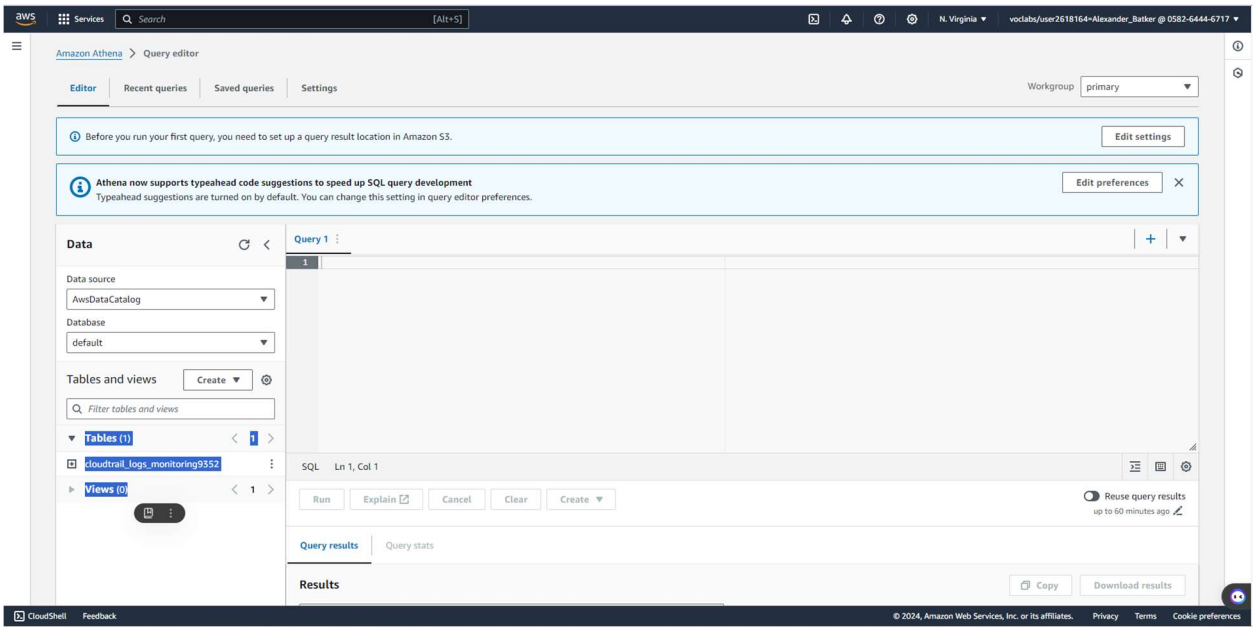
1. Connect to the CLI host associated with the cloudformation template with proper credentials
2. Run this command to start drift detection and note the output of the “StackDriftDetectionId”:
    a. aws cloudformation detect-stack-drift --stack-name myStack
3. Run this command to monitor the status of the drift detection. Replace driftID with StackDriftDetectionId found in the previous step:
    a. aws cloudformation describe-stack-drift-detection-status \
    b. --stack-drift-detection-id driftId
4. Run the following command with ‘myStack’ replaced with the actual stack name
    a. aws cloudformation describe-stack-resources \
    b. --stack-name myStack \
    c. --query 'StackResources[*].[ResourceType,ResourceStatus,DriftInformation.StackResourceDriftStatus]' \

d. --output table

```
[ec2-user@cli-host ~]$ aws cloudformation describe-stack-resources \
> --stack-name myStack \
> --query 'StackResources[*].[ResourceType,ResourceStatus,DriftInformation.StackResourceDriftStatus]' \
> --output table
-------------------------------------------------------------------------------
|                          DescribeStackResources                             |
+-----------------------------------------+-----------------+---------------+
|  AWS::EC2::InternetGateway              |  CREATE_COMPLETE |  IN_SYNC      |
|  AWS::EC2::VPC                          |  CREATE_COMPLETE |  IN_SYNC      |
|  AWS::S3::Bucket                        |  CREATE_COMPLETE |  IN_SYNC      |
|  AWS::EC2::Route                        |  CREATE_COMPLETE |  IN_SYNC      |
|  AWS::EC2::RouteTable                   |  CREATE_COMPLETE |  IN_SYNC      |
|  AWS::EC2::SubnetRouteTableAssociation  |  CREATE_COMPLETE |  NOT_CHECKED  |
|  AWS::EC2::Subnet                       |  CREATE_COMPLETE |  IN_SYNC      |
|  AWS::EC2::VPCGatewayAttachment         |  CREATE_COMPLETE |  NOT_CHECKED  |
|  AWS::CloudFormation::WaitCondition     |  CREATE_COMPLETE |  NOT_CHECKED  |
|  AWS::CloudFormation::WaitConditionHandle |  CREATE_COMPLETE |  NOT_CHECKED  |
|  AWS::EC2::SecurityGroup                |  CREATE_COMPLETE |  MODIFIED     |
|  AWS::EC2::Instance                     |  CREATE_COMPLETE |  IN_SYNC      |
+-----------------------------------------+-----------------+---------------+
[ec2-user@cli-host ~]$ aws cloudformation describe-stack-resource-drifts \
> --stack-name myStack \
> --stack-resource-drift-status-filters MODIFIED
{
    "StackResourceDrifts": [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:533267140408:stack/myStack/47d795a0-dbf7-11ee-90b7-1
            "ActualProperties": "{\"GroupDescription\":\"Enable access to web server\",\"GroupName\":\"WebSer
\":\"0.0.0.0/0\",\"FromPort\":80,\"IpProtocol\":\"tcp\",\"ToPort\":80}],\"Tags\":[{\"Key\":\"Name\",\"Value\"
            "ResourceType": "AWS::EC2::SecurityGroup",
            "Timestamp": "2024-03-06T20:30:00.240Z",
            "PhysicalResourceId": "sg-019bb12f1f5bd56ba",
            "StackResourceDriftStatus": "MODIFIED",
            "ExpectedProperties": "{\"GroupDescription\":\"Enable access to web server\",\"GroupName\":\"WebS
\"0.0.0.0/0\",\"FromPort\":80,\"IpProtocol\":\"tcp\",\"ToPort\":80}],\"Tags\":[{\"Key\":\"Name\",\"Value\":\"
            "PropertyDifferences": [
                {
                    "PropertyPath": "/SecurityGroupIngress/0/CidrIp",
                    "ActualValue": "68.12.25.81/32",
                    "ExpectedValue": "0.0.0.0/0",
                    "DifferenceType": "NOT_EQUAL"
                }
            ],
            "LogicalResourceId": "WebSecurityGroup"
        }
    ]
}
[ec2-user@cli-host ~]$ aws cloudformation update-stack \
> --stack-name myStack \
> --template-body file://template1.yaml \
> --parameters ParameterKey=KeyName,ParameterValue=vockey

An error occurred (ValidationError) when calling the UpdateStack operation: No updates are to be performed.
[ec2-user@cli-host ~]$ Alexander Batker
```

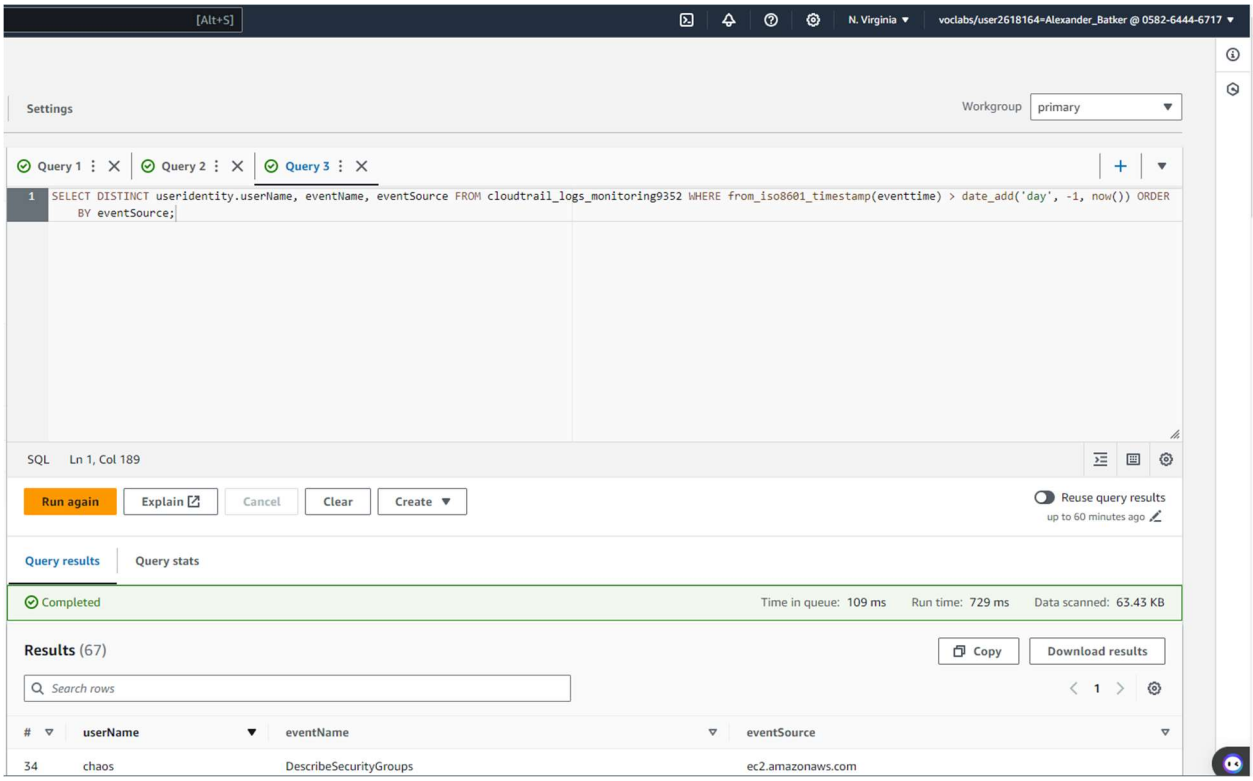
How to create an Amazon Athena table

1. Navigate to CloudTrail on AWS Management Console. Navigate to ‘Event History’ -> Create Athena table.
2. Choose S3 bucket associated with the CloudTrail log files and Create Table!
3. Navigate to Athena via the search bar for Services and then to Query Editor. This verifies the creation of the Athena database table.



How to manually review access logs to find anomalous user activity

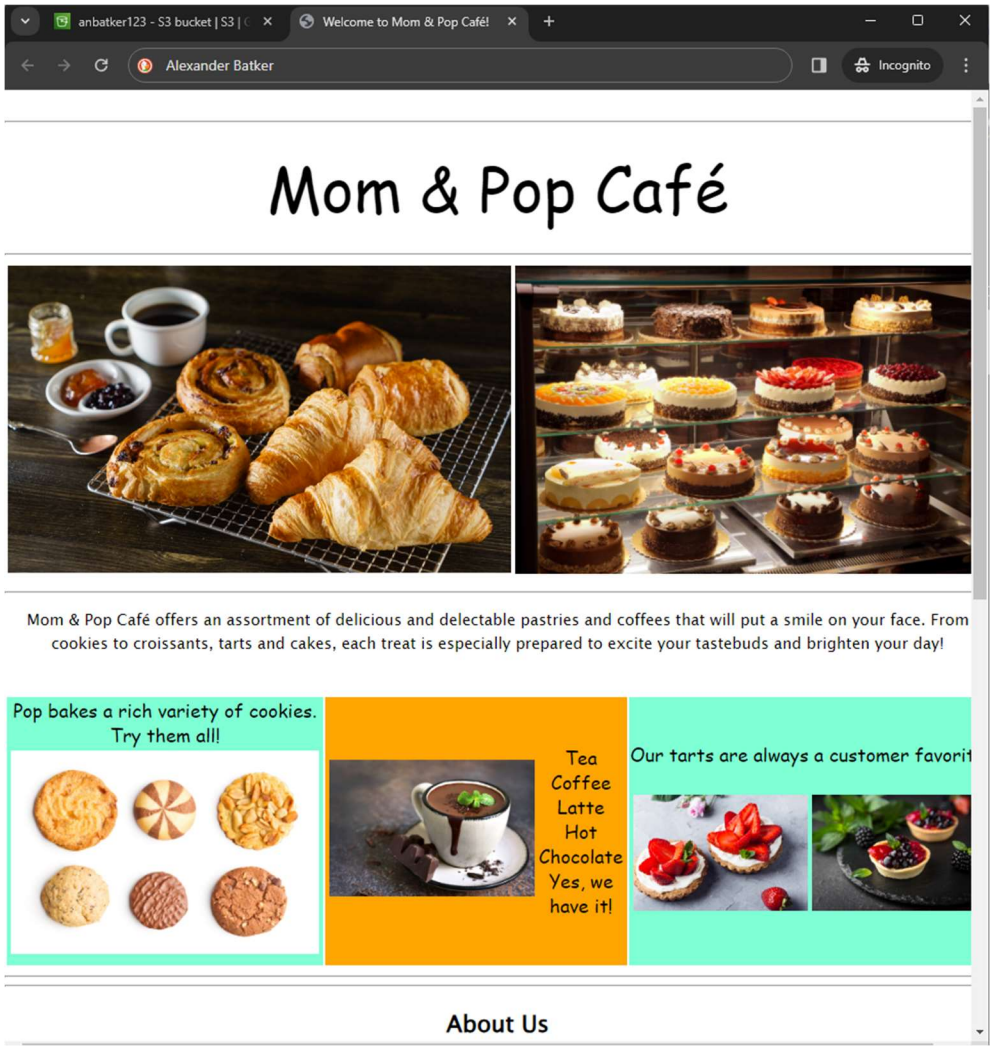
1. When in the Query Editor for AWS Athena, start with a broad, more generalized query to search for the anomalous user activity.
    a. SELECT useridentity.userName, eventtime, eventsource, eventname, requestparameters
    b. FROM cloudtrail_logs_monitoring####
    c. LIMIT 30
2. Add more constraints and filters to the query in order to track down the user activity in question
    a. Filters such as WHERE eventsource = ‘ec2.amazonaws.com’
3. Identify anomalous user by their activity through the narrowing of your queries.



How to create a batch file to update the café website to change its colors

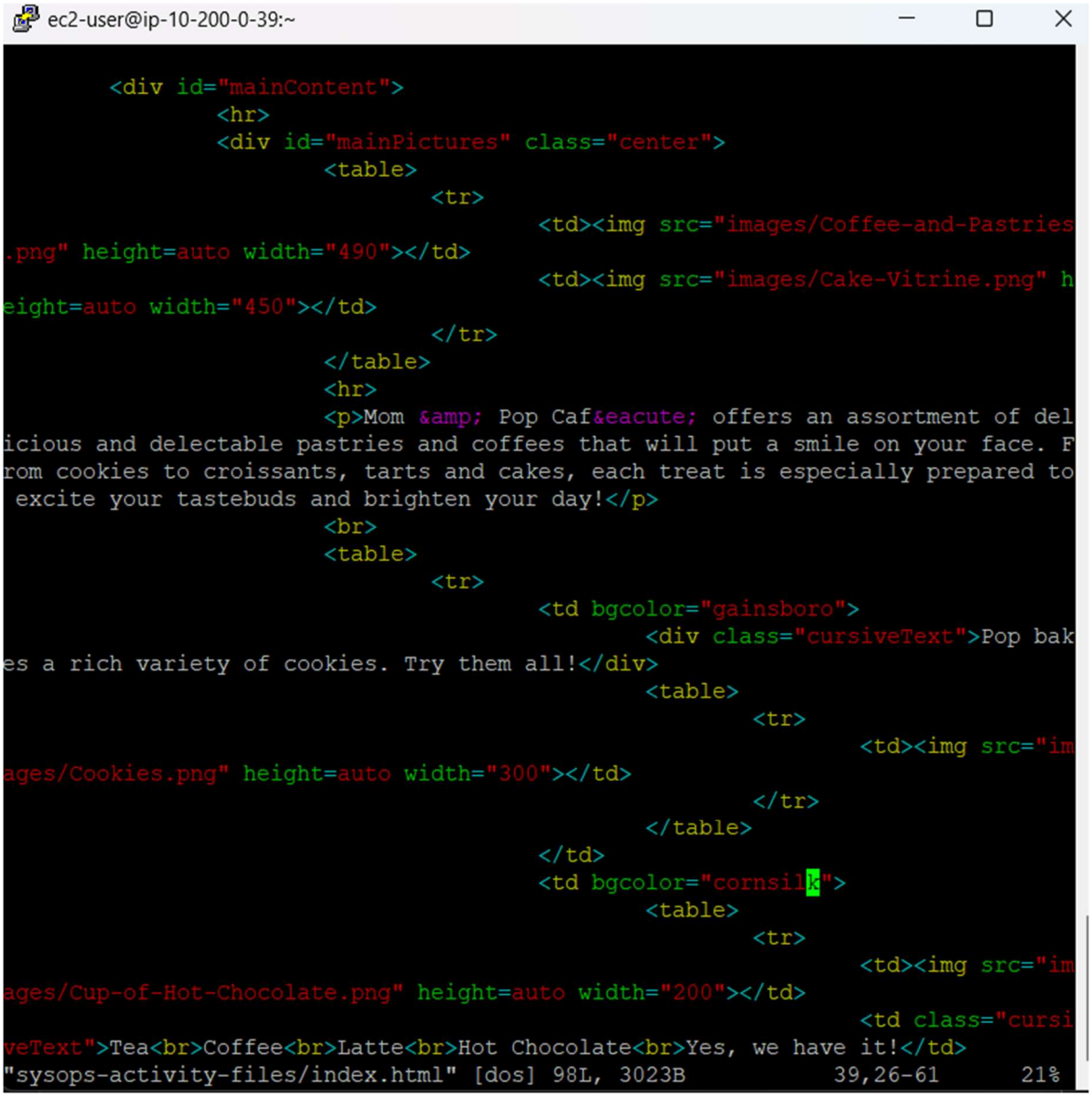
1. Ensure that you are in a putty session with the correct public ipv4 address and login to your AWS account via AWS CLI
2. Double-check that the files are uploaded to the S3 bucket
    a. ‘aws s3 cp . s3://<my-bucket>/ --recursive --acl public-read’ or ‘aws s3 ls <my-bucket>’
3. Navigate to AWS S3 console and click on the hyperlink of your bucket that hosts the website
4. Click on properties, scroll to static website hosting panel, and then choose ‘bucket website endpoint’ link to load the website into a new browser tab.

a. This ensures you can visually see your updates to the café website.



5. Run the following commands back in the putty session AWS CLI to create a new file with your updates and then open the empty file in vi editor.
    a. cd ~
    b. touch update-website.sh
    c. vi update-website.sh
6. Copy this line to the first line of the bash file with the ‘s3 cp’ line that contains the bucket
    a. #!/bin/bash
    b. aws s3 cp ~/sysops-activity-files/ s3://<my-bucket>/ --recursive --acl public-read

7. Run these two commands to make it an executable batch file and open the local copy in vi:
    a. chmod +x update-website.sh
    b. vi sysops-activity-files/index.html
8. Click ‘a’ to go into edit mode and navigate to the html code containing the object’s color you want to change and then type ‘esc’, then ‘:wq’ to exit and save.



9. Click refresh back on the browser showing the webpage to check the color changes.

anbatker123 - S3 bucket | S3 | C

Welcome to Mom & Pop Café!

Alexander Batker

Incognito

Mom & Pop Café

Mom & Pop Café offers an assortment of delicious and delectable pastries and coffees that will put a smile on your face. From cookies to croissants, tarts and cakes, each treat is especially prepared to excite your tastebuds and brighten your day!

Pop bakes a rich variety of cookies. Try them all!

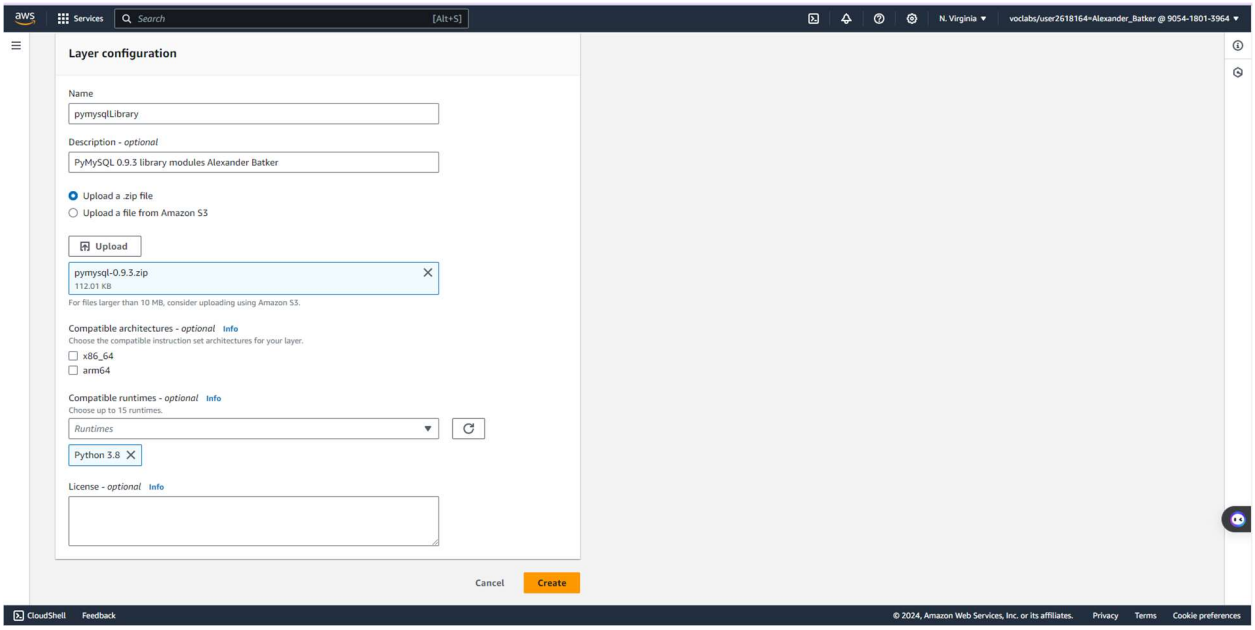
Tea
Coffee
Latte
Hot Chocolate
Yes, we have it!

Our tarts are always a customer favorit

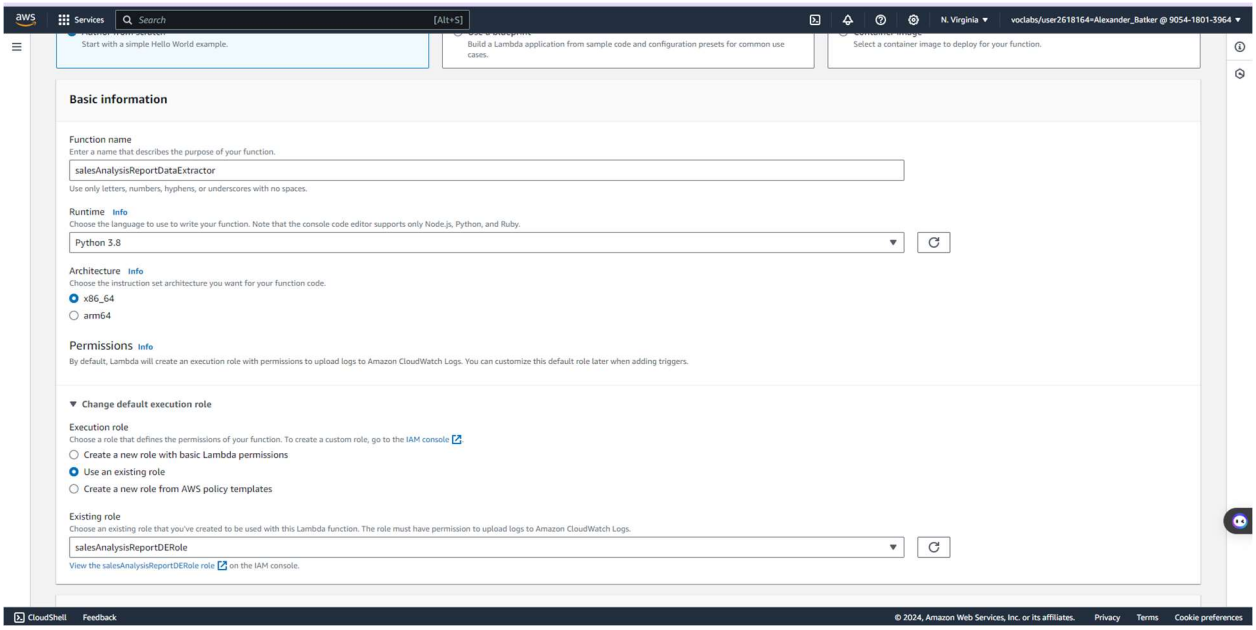
About Us

How to create a Lambda Layer and add it to a Lambda function

1. Search for Lambda on the search bar and click Lambda to go to the Lambda module. Then click the left panel to expand it and click on layers, then click Create layer.
2. Choose zip file from machine and upload it. Choose compatible runtimes.



3. Go back to Lambda home page and create function.
4. Name function and match runtime to that of the layer created in Lambda. Add existing role that has proper permissions.



5. After creating the function, click on the created function and choose layers. Under layers, add a layer.
6. Add the layer created earlier, and now the layer will be present under the Lambda module under the Function overview area.

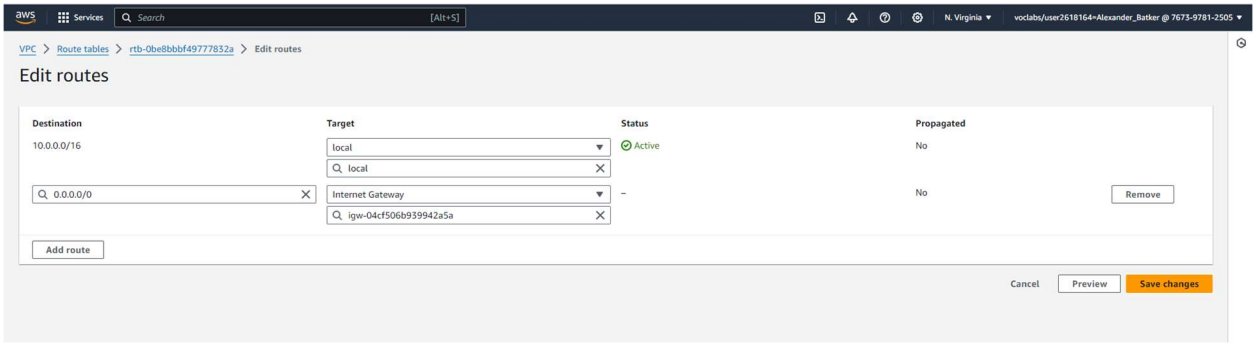
How to create a Lambda function from a prebuilt package

1. Connect to instance from AWS CLI using proper credentials.
2. Run the code seen below with prebuilt package details.
3. Check AWS UI to ensure lambda function utilizing a prebuilt package was created.

How to setup a VPC

1. In the AWS console, search for VPC and select VPC. Create VPC and ensure to tag the name of the VPC. Enter the proper IPv4 CIDR block.
2. Now on actions, depending on what the actions display, click ‘edit DNS hostnames’ or ‘edit VPC’. Enable DNS hostnames in order to allow EC2 instances to automatically receive a DNS hostname.
3. Navigate to ‘subnets’ on the left pane and select the corresponding VPC ID and create a public subnet first. Use same availability zone for both private and public subnets. Allow private subnet to have more available addresses.
4. Next, create an internet gateway. On the left pane, select internet gateways and create an internet gateway in the internet gateway module. Tag the IGW and create the internet gateway.
5. Attach the IGW to a VPC by choosing ‘Actions’ then ‘Attach to VPC’.

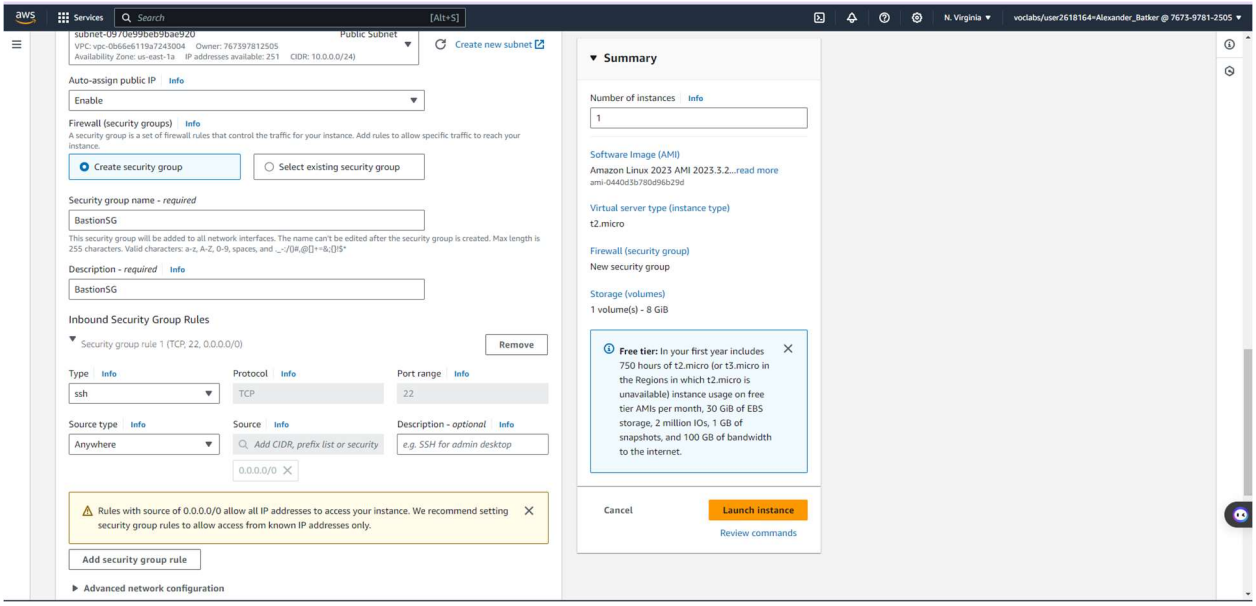
6. After this, we configure the route tables. In the left pane, select ‘Route Tables’. Create public and private route tables if not already created. Keep private subnet to route all traffic destined for range of the VPC. Allow public route table to direct 0.0.0.0/0 traffic to the internet gateway and save changes.



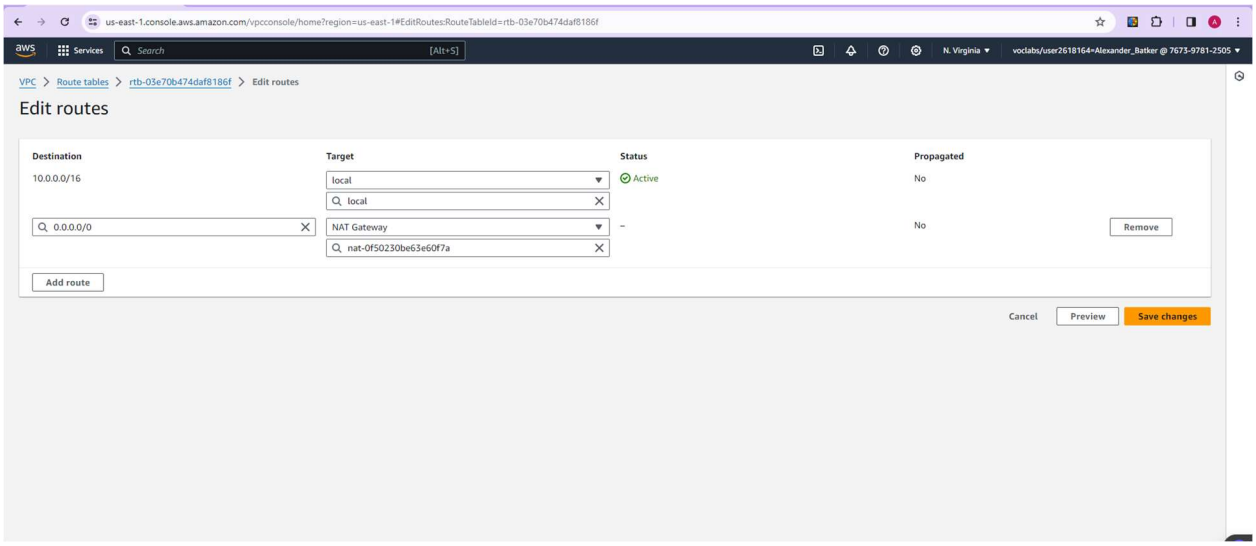
7. Click on the ‘Subnet associations’ tab and go to ‘explicit subnet associations’. Click ‘edit’ and then check the public subnet to be associated with this public route table.

How to add a bastion host (Linux) to the public subnet of a VPC to connect to instances in the private subnet

1. Navigate to ec2 services in AWS and click ‘launch instance’. Choose a free tier Amazon Linux 2023 AMI and a free teir t2.micro instance type.
2. Choose vockey as key pair login, then on network settings choose the corresponding VPC and the public subnet. Create a SG for the BastionSG if not already created. Inbound security rule should be set to SSH.



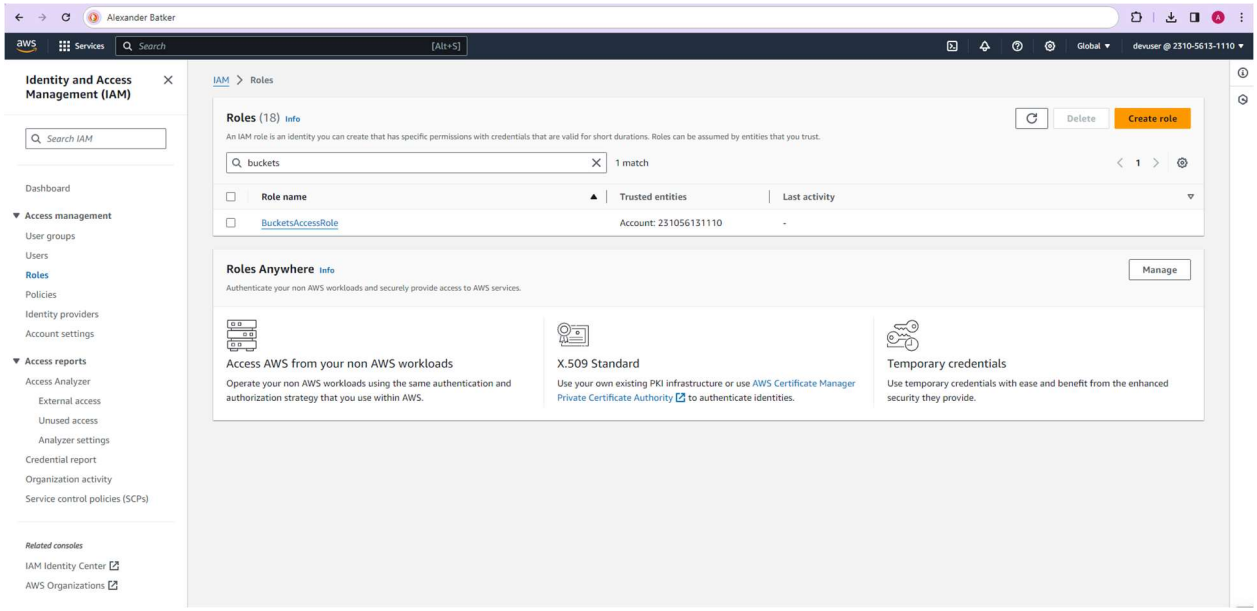
3. After the creation of the bastion server, navigate back to VPC and navigate to NAT gateways, then click ‘create nat gateway’.
4. Choose the public subnet and click allocate elastic IP, then create NAT gateway.
5. Navigate back to the route tables interface, click routes and select the private route table, then select the routes tab. Edit the route to add a route of the NAT gateway at 0.0.0.0/0



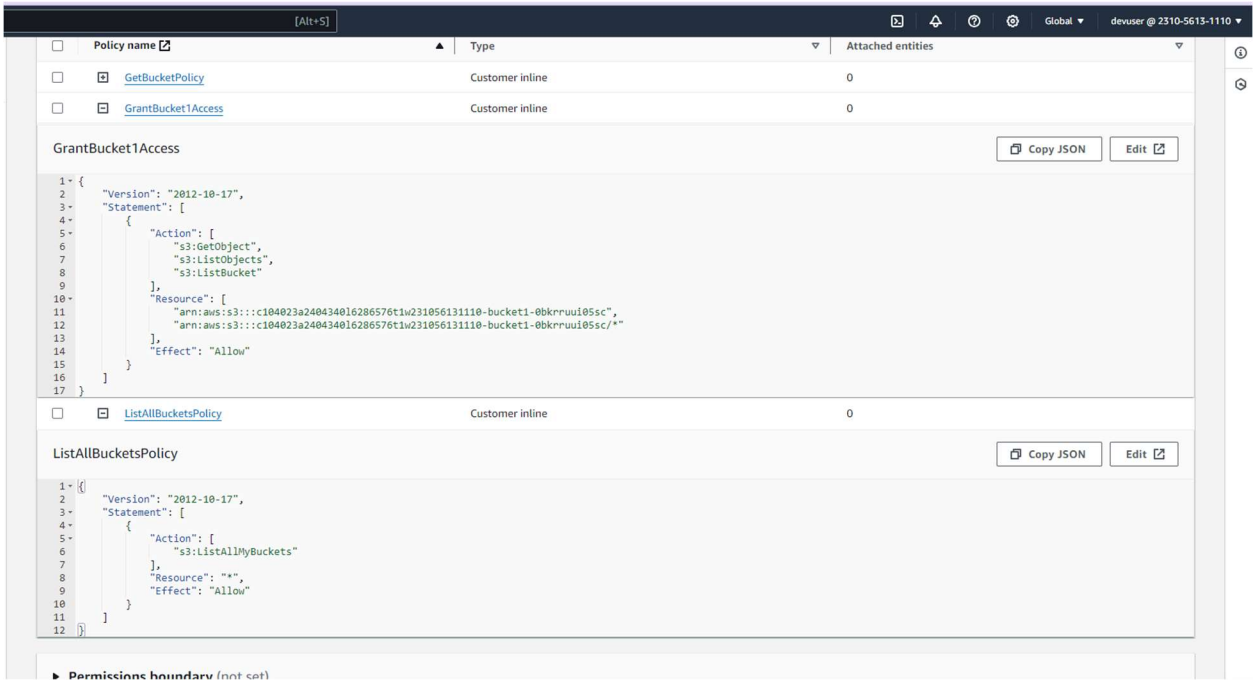
6. Now, the connection of the bastion host to instances in the private subnet should be functional.

How to setup IAM so a user can assume an IAM role to access a resource

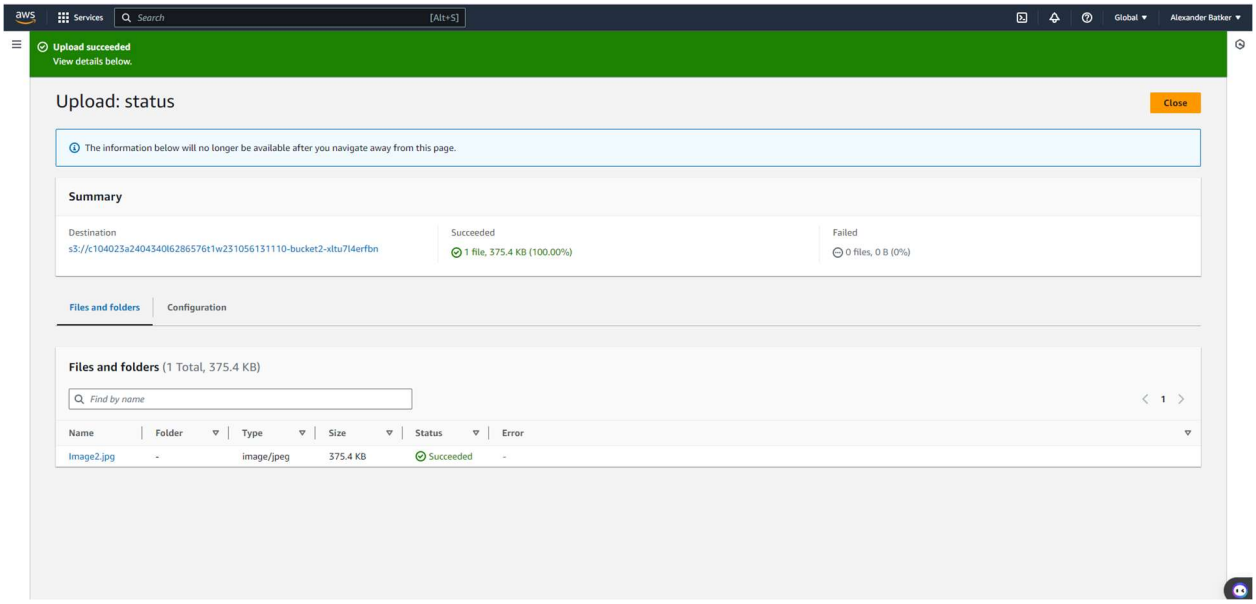
1. When logged into AWS management console as a user, navigate to IAM -> Roles -> [Search for access role wanted for scenario, for this case it is BucketsAccessRole].

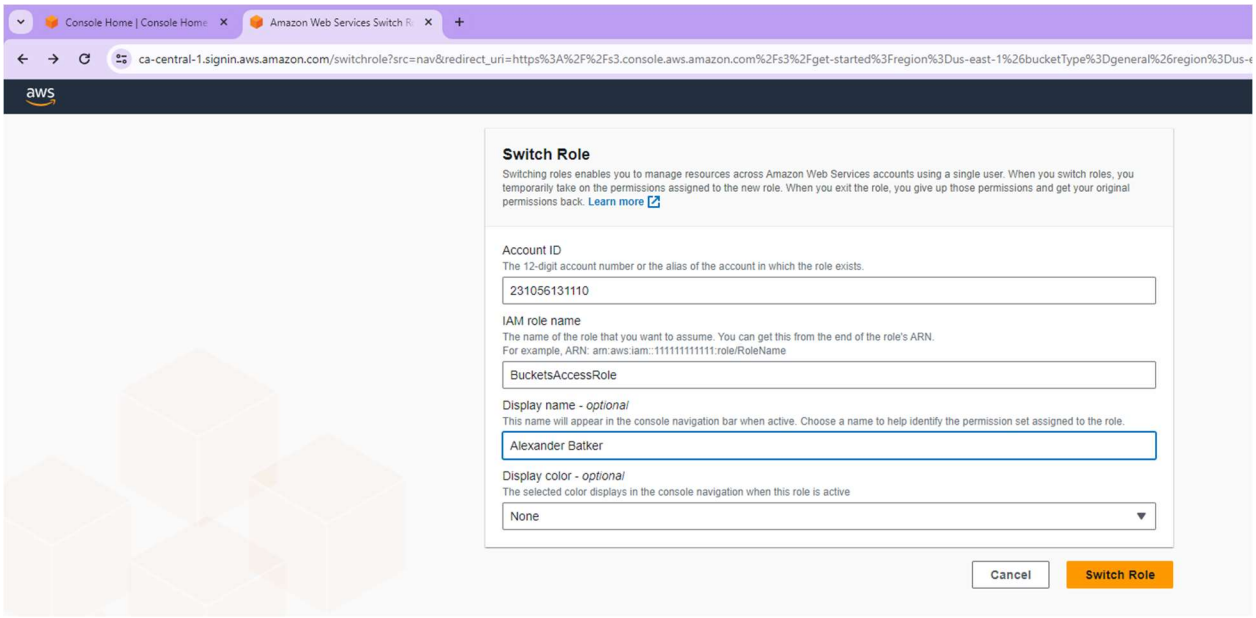


2. Navigate to access json file desired, and save the json file to local computer.



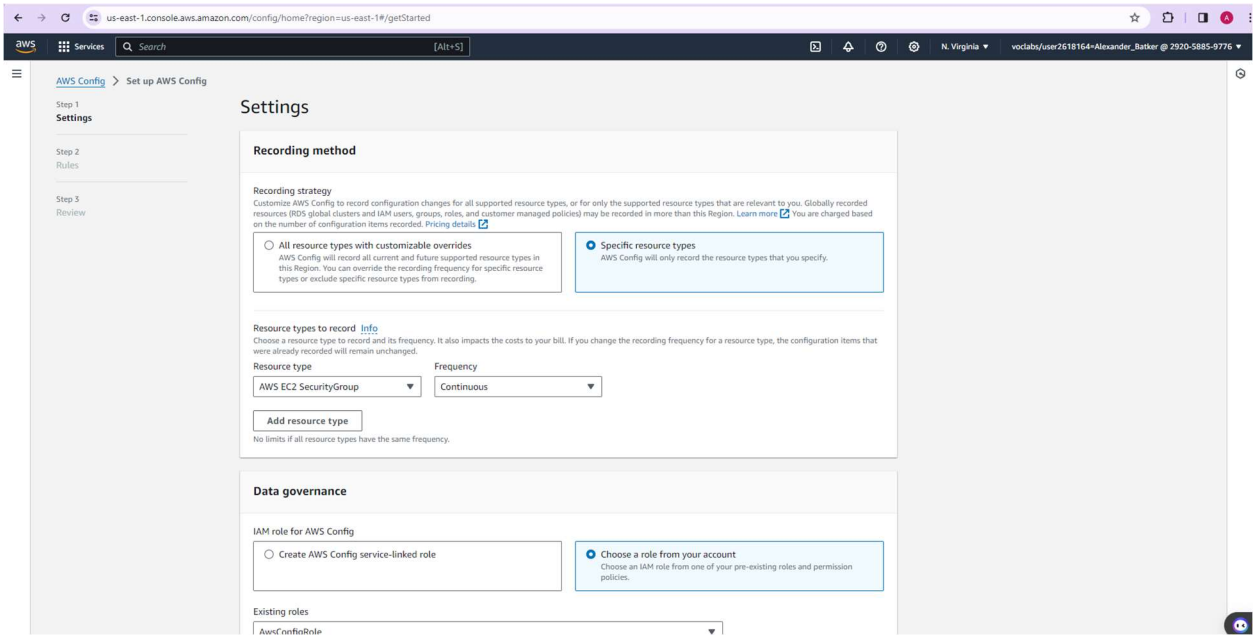
3. When logged into AWS as a user, click the top right where the username appears and then click ‘switch role’.
4. Have account ID on hand and choose the role name and switch role. Try doing an action that was once frowned upon by AWS before the IAM role was assumed. (Uploading a file to an S3 bucket that was once denied permission to do so.



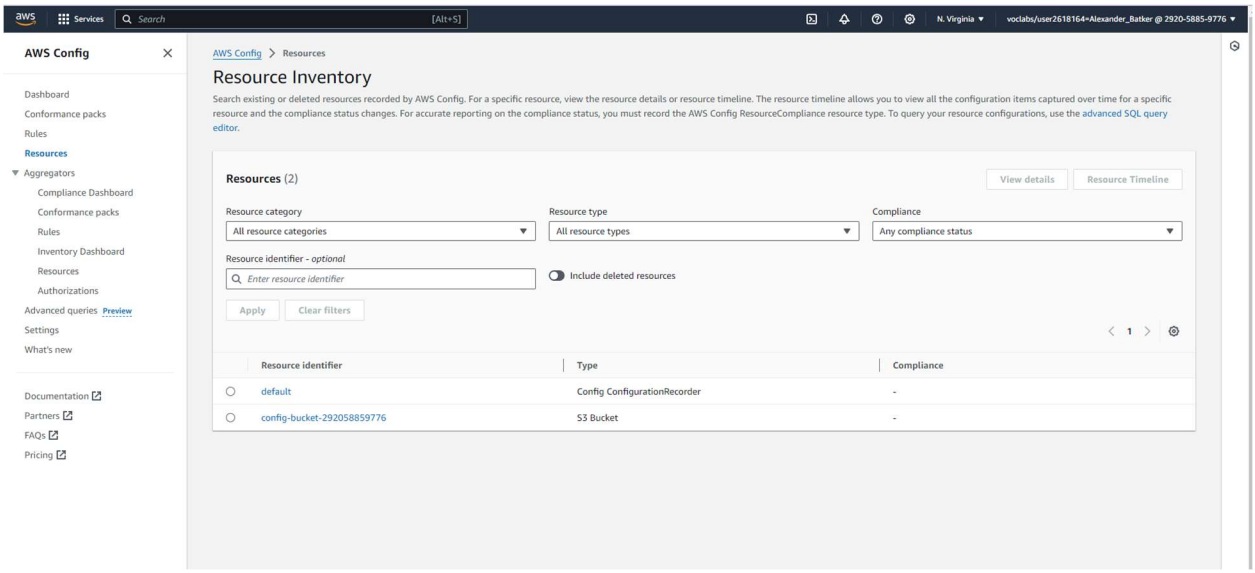


How to setup AWS Config to monitor resources

1. While logged into AWS Management Console, navigate to AWS Config -> Get Started.

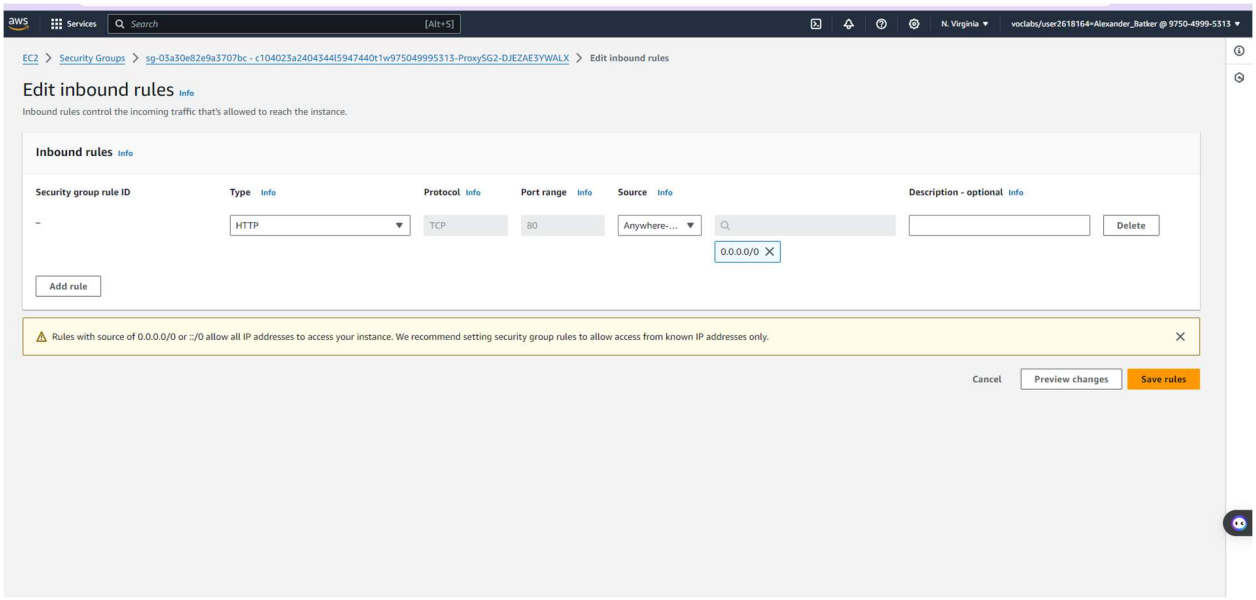


2. Choose monitoring/recording strategy that best fits the scenario, and choose an existing AWS IAM Role if possible.
3. Keep delivery methods and AWS Managed Rules as default if all necessary permissions/rules are set in place already.
4. Navigate to AWS Config -> Resources on the left pane to verify the creation of the config-bucket that stores all AWS Config data from the created AWS Config from earlier steps.



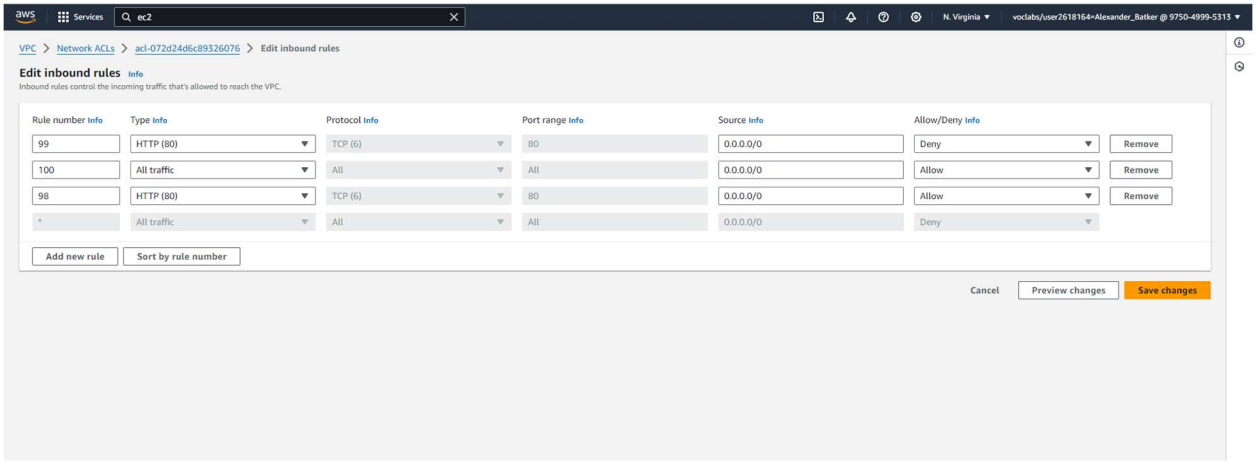
How to add inbound rules to both security groups and network ACLs

1. First, navigate to the ec2 module in AWS, then security groups and select the security group needing to be modified.
2. Ensure inbound rules is the selected module, and click 'edit inbound rules'. For this example, HTTP is the type of inbound rule with a source from any public IPv4 address.



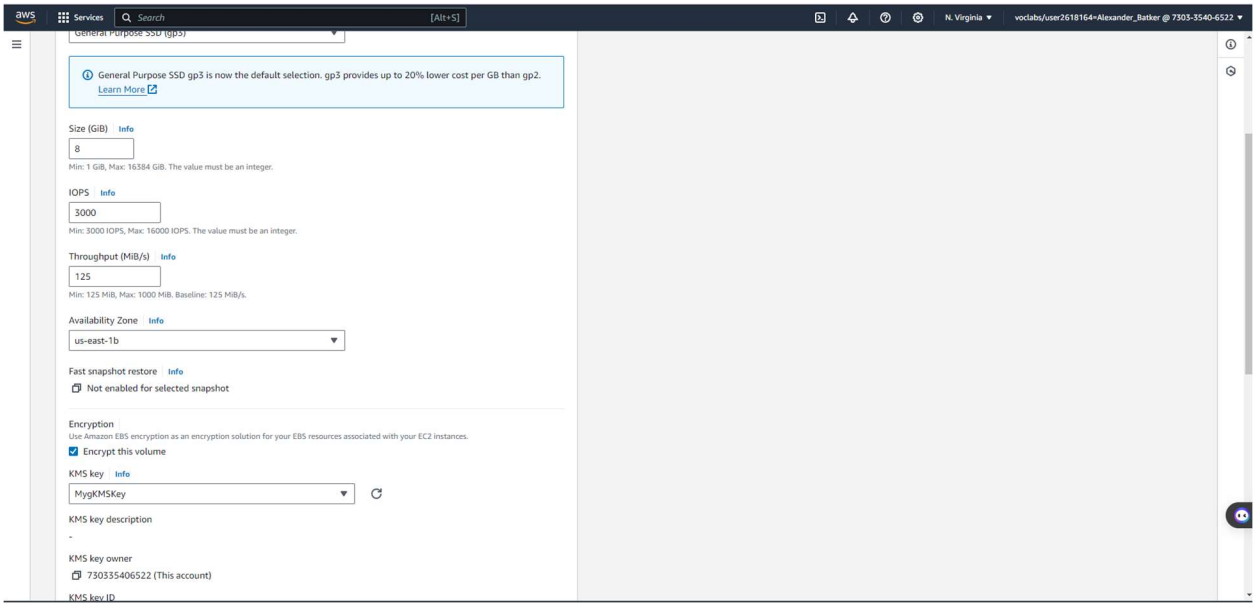
3. Now, navigate to the VPC module in AWS, then select Network ACLs and then select the proper network ACL. Select inbound rules and then 'edit inbound rules'.

4. Click ‘add new rule’ and choose the type of rule, source, and allow/deny.



How to encrypt the root volume of an existing EC2 instance

1. Stop EC2 instance that is chosen for root volume encryption in AWS console
2. Next, to create snapshot of unencrypted root EBS volume by:
    a. Going to storage on that instance ID link, block devices, then click ‘Volume ID’ link
    b. Note the region in which the instance is located
    c. Options -> create snapshot and add descriptors
3. Next, to create an encrypted volume from unencrypted snapshot
    a. Go to EBS on the left and choose ‘Snapshots’
    b. Choose link to the unencrypted snapshot ID and let its status show ‘Completed before next step.
    c. Actions -> Create volume snapshot
    d. Choose availability zone (region) noted from earlier and choose proper KMS key
    e. Create volume

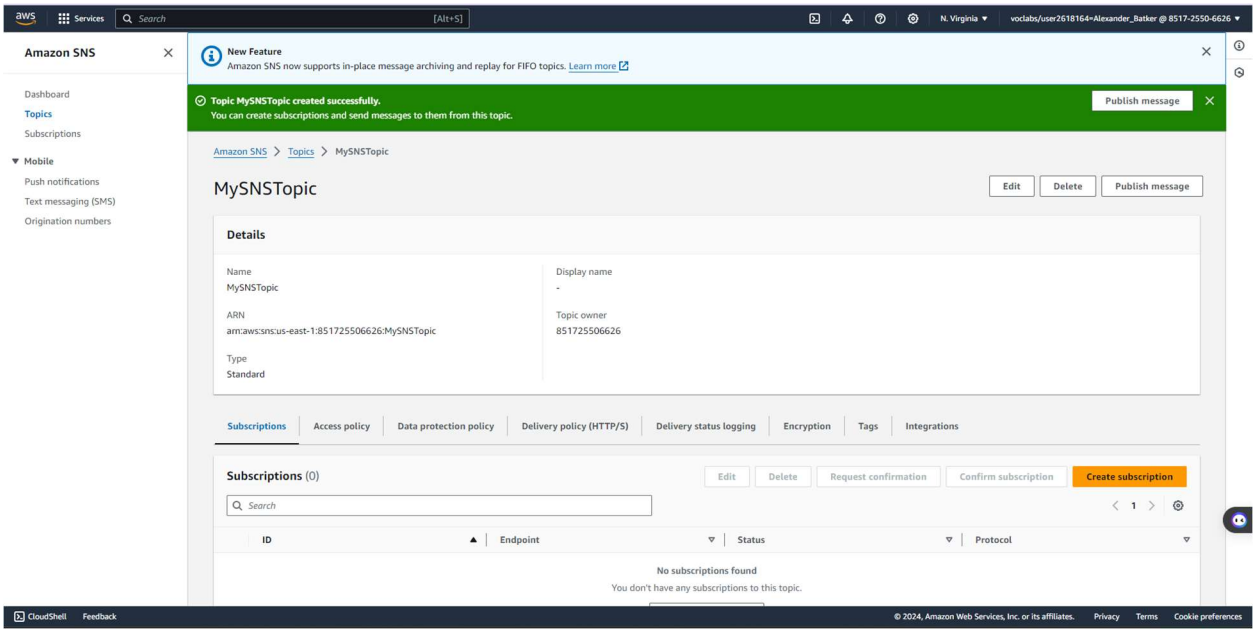


4. Change names of both volumes to proper names of encrypted and unencrypted
5. Swap root volume of EC2 instance:

a. Select old unencrypted volume and then -> Actions -> Detach volume
b. Select new encrypted volume and then -> Actions -> Attach volume
    i. Choose stopped instance and device name where existing instance expects to find that root volume
c. Attach Volume by clicking ‘Attach Volume’

How to create a SNS topic

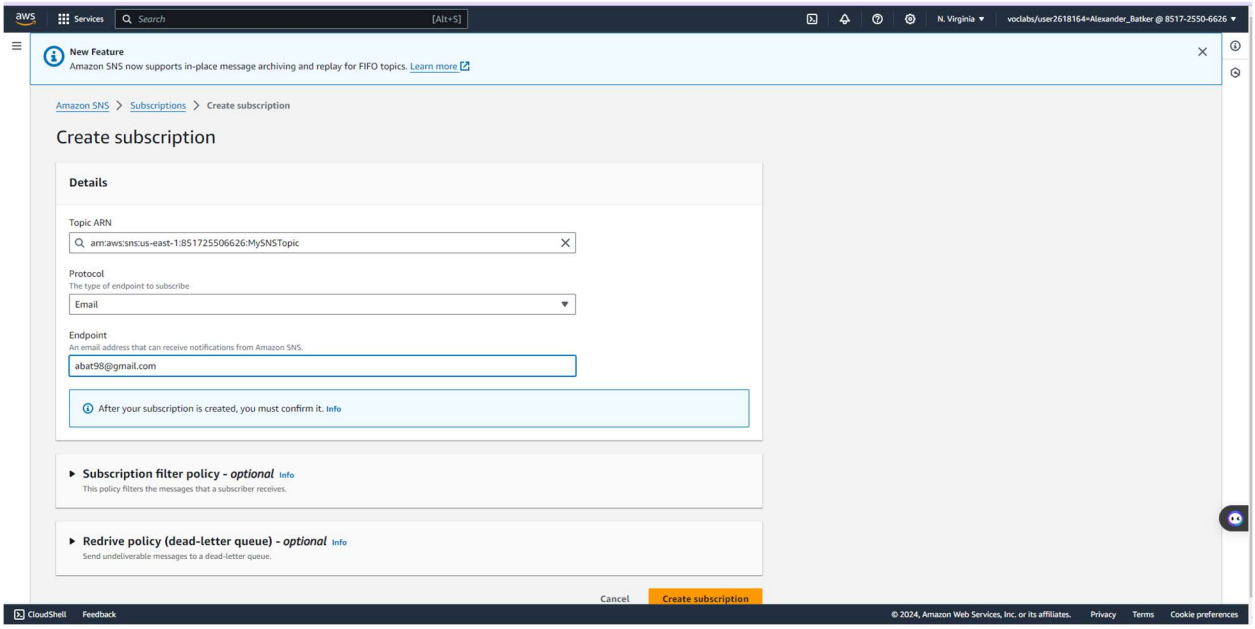
1. Navigate to SNS AWS Console
2. In the left pane, choose topics, then Create Topic.
3. Choose standard type, define who can publish and subscribe according to use case.
4. Create Topic!



How to subscribe to a SNS topic

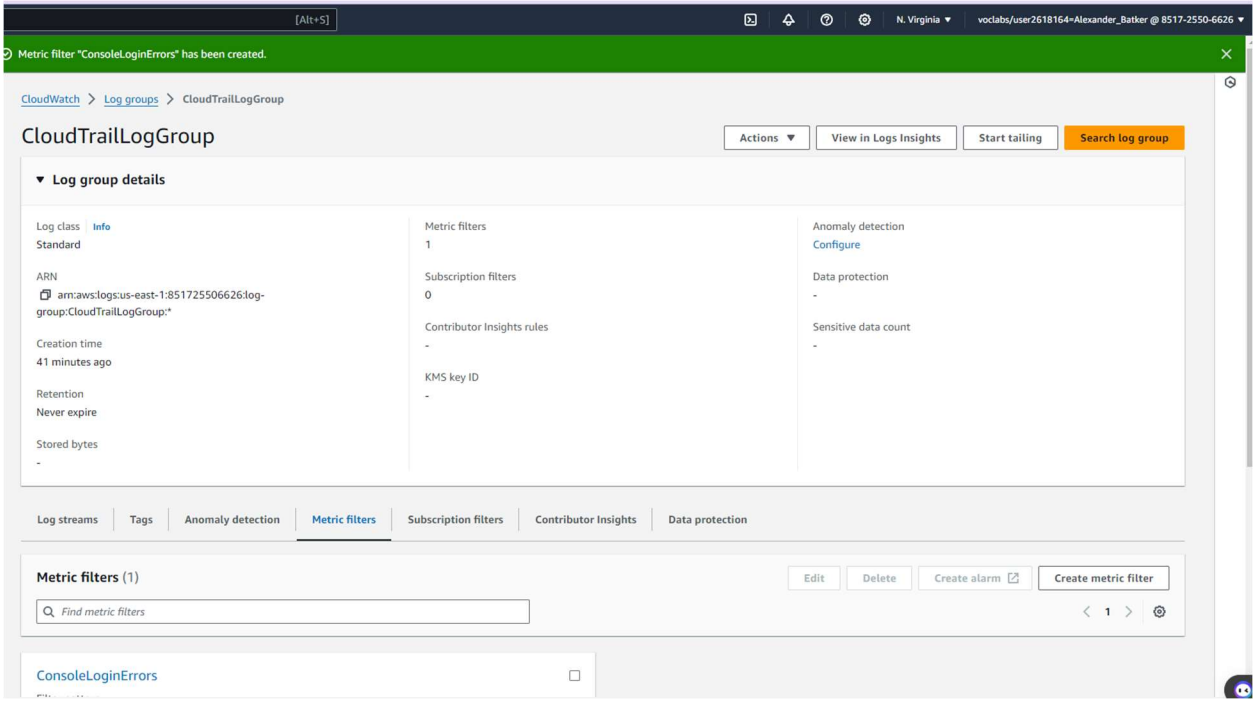
1. Navigate to AWS SNS console and click on the SNS topic needed to be subscribed to.
2. Click create subscription then configure the protocol, ensure ARN is correct, as well as any endpoints.

3. Click Create Subscription



How to create a CloudWatch alarm using a metrics-based filter

1. Navigate to CloudWatch AWS Console, then to Logs -> Log Groups and select the box for the proper cloud trail log group.
2. Insert a filter pattern with code or select a premade one, then choose next.
3. Create namespace, metric name, value, and any relevant tags should be added.



4. Scroll down to metric filters tab, then check the box on the top right of the metric created and choose Create Alarm.

5. Finally, specify metric conditions for the alarm and choose the SNS topic or other preferred endpoint for the alarm to go to and Create Alarm.