Operations Playbook

Name: Padmasree Ramesh Srinivasan

Date: 4/25/2023

Class/Semester: IFT 598 – Cloud Operations and Security in IT; Spring 2023

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 keep alives - 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.
7. For macOS and Linux Users - make sure that you change the directory to the one wherethe pem file was downloaded and change the permissions on the key to read-only by running the *chmod* command
8. In the terminal window, run the *ssh -I file.pem ec2-user@<public-ip>*. Replace thepublic-IP with the actual public IPv4 address of the EC2 instance.

Figure 1:

Installing the AWS CLI on Linux

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Vasanths-MacBook-Pro:~ padmasree$ cd ~/Downloads
Vasanths-MacBook-Pro:Downloads padmasree$ chmod 400 labsuser.cer
Vasanths-MacBook-Pro:Downloads padmasree$ ssh -i labsuser.cer ec2-user@44.202.96.216
The authenticity of host '44.202.96.216 (44.202.96.216)' can't be established.
ECDSA key fingerprint is SHA256:M3RLUaPq/6G0DJMew0YdcOJhNLIU0qMpjrx3C5f6mWs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.202.96.216' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-200-0-194 ~]$ python3 --version
Python 3.7.15
```

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

1. Ensure that you have an active SSH connection to the instance running on EC2.
2. To install AWS CLI, make sure that Python is installed by running *python3 —version* command.
3. Download and install AWS CLI.
4. Verify that AWS CLI is working by running the *aws help* command and it should displaythe help information for AWS CLI.
5. Run the *aws configure* command to set up the AWS CLI installation. AWS CLI willprompt you for Access key ID, Secret Access key, AWS Region and Output format.
6. In the terminal window, we can test IAM configuration by using *aws iam list-users* command.
7. If the test is successful and AWS CLI is configured correctly, you should see a JSONresponse that shows the list of all the IAM users in the account.
8. In the AWS Management Console browser, choose Services and go to IAM service.
9. In the IAM service page, in the left navigation pane, choose Users, and then choose theuser, which will be a hyperlink.
10. Choose the arrow icon next to lab_policy and choose JSON to view the policy documentin JSON. This should watch with the JSON response that you saw in the terminal.

Figure 2:

IAM configuration details in the AWS Management console.

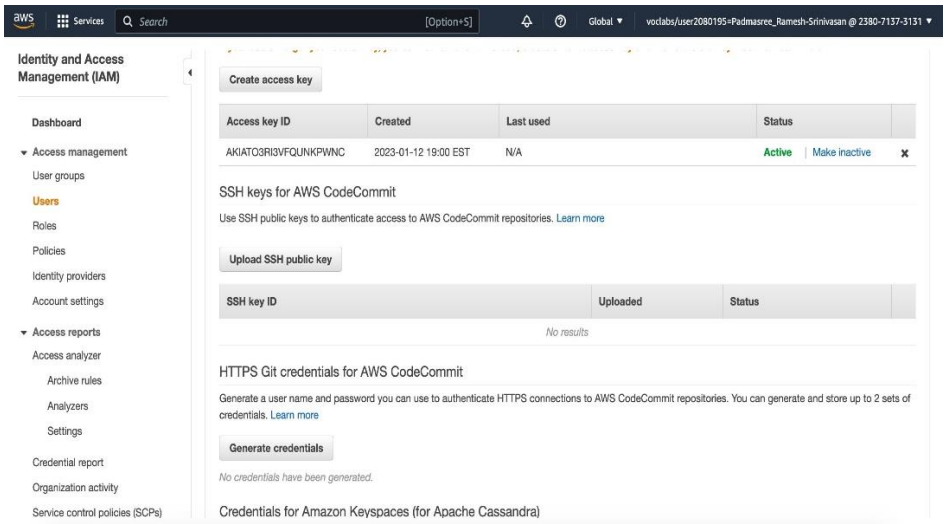


Figure 3:

Observe IAM configuration details using the AW

```
[ec2-user@ip-10-200-0-228 ~]$ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "awsstudent",
            "UserId": "AIDAT03RI3VF7PS3DRDTB",
            "Arn": "arn:aws:iam::238071373131:user/awsstudent",
            "CreateDate": "2023-01-12T23:59:41+00:00"
        }
    ]
}
```

How to make modifications to the lab policy using AWS CLI

1. Get policy using aws iam get-policy. To get information about the specific policy, make surethat you have the arn and the version of the lab_policy.

2. Get the policy document in JSON format using the following command *aws iam get-policy\ — policy-arn —version*

3. Write the JSON document to an output file using *aws iam get-policy\ — policy-arn —version > output.txt.* We can make modifications to the output file using the vi editor and edit the policy using theinsert function.

Figure 1:

Console output from vi lab_policy command

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

1. In the AWS Management Console, on the Services menu, click Systems Manager.
2. In the left navigation pane, under Application Management, click Parameter Store.
3. Click Create parameter and configure:
    - **Name:** /web.config/cookie_toggle
    - **Description**: This feature allows you to turn cookies on or off for the Café website
    - **Value:** True
4. The application that is running on the EC2 instance will automatically check for this parameter. If it finds the parameter, then additional features will be displayed.
5. Open the web browser displaying the Café website page using the public IP address of the EC2 instance.
6. If you refresh the web page, you will notice the option to turn cookies on or off.

Figure 1:

Inventory list output

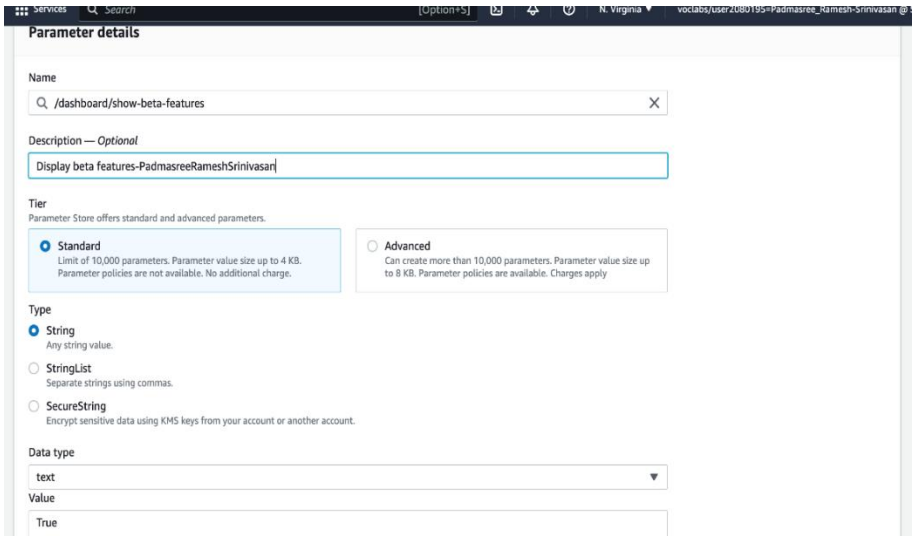
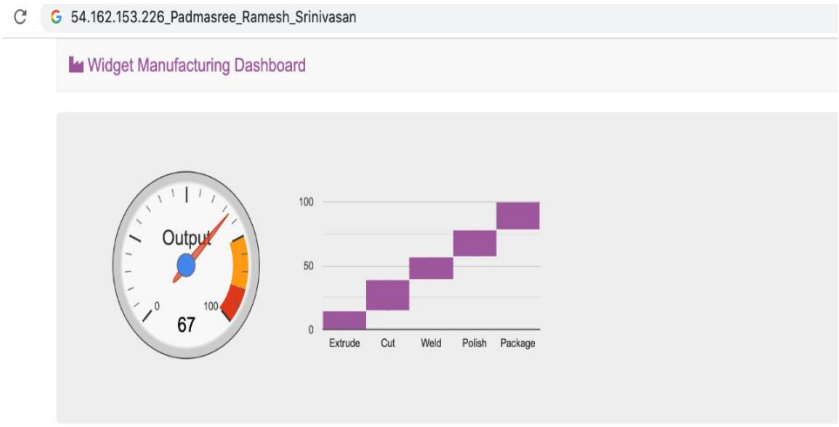


Figure 2:

Widget Manufacturing Dashboard webpage



How to connect to an EC2 instance to describe instances

1.In the Management console, in the left navigation pane, click Session Manager.

2. Click Start Session.

3. Select Managed Instance.

4. Click Start Session. A session window will open in your browser.

5. Click in the session to activate the cursor.

6. Run this command in the session window:

```
ls /var/www/html
```

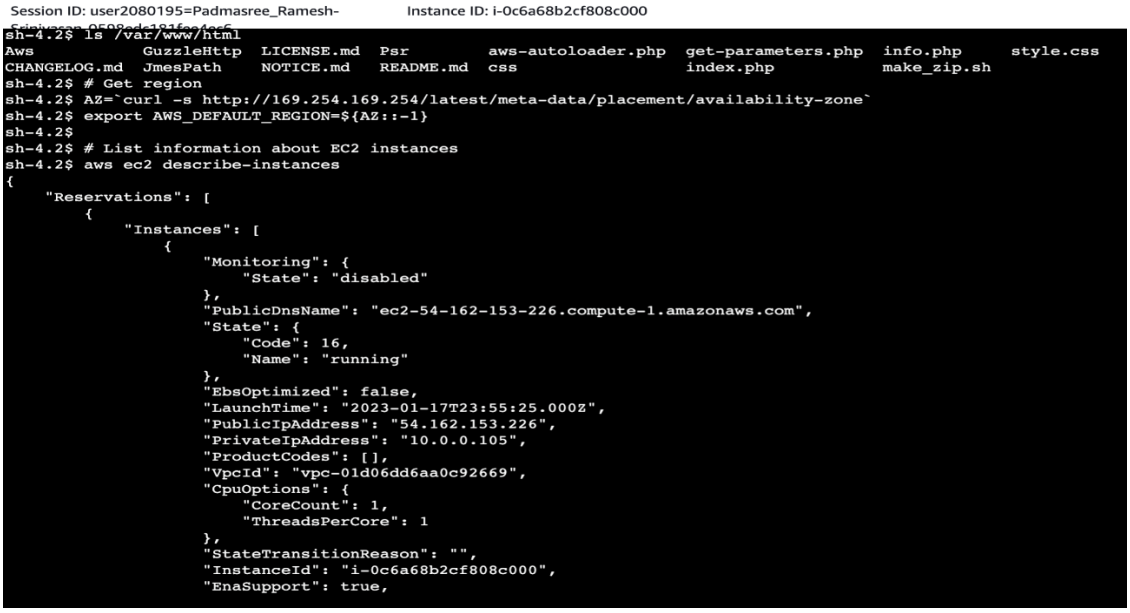
7. You will see application files that were installed on the instance.

8. Run this command in the session window:

```
#Get region
AZ= curl -s http://169.254.169.254/latest/meta-data/placement/availablity-zone
export AWS_DEFAULT_REGION=${AZ::1}
#List information about EC2 instances
aws ec2 describe-instances
```

Figure 1:

Using Session Manager to Access Instances



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

1. Connect to an existing Amazon Linux EC2 instance that has AWS CLI installed in it. Download the PPK or PEM file based on the Operating System that you are using and SSH into the instance using the key pair.
2. Update the AWS CLI software with credentials using the command *aws configure.* When prompted, enter the AWS Access key ID, AWS Secret Access Key, Default region name and the default output format.
3. Create a bucket using the *aws s3api create-bucket* command and specify the region that is closest to the people who are most likely to access it. If you want to use regions outside of *us-east-1*, the appropriate *LocationConstraint* should be specified in order to create the bucket in the desired region.

4. In the AWS Management console, navigate to the IAM console. In the left navigation pane, choose Users and then choose the *awsS3user hyperlink.* In the Permissions tab, observe the details of the policy attached to the user in {}JSON format.
5. In the Security credentials tab, copy the *Console sign-in link* and copy it to your clipboard. Login *as awsS3user* IAM user from a private browser window or a different browser
6. In the SSH terminal, extract the files that you need to host a website and run the *ls* command to make sure that the files are extracted correctly.
7. Configure the bucket that you created earlier for static website hosting that identifies that index document *aws s3 website s3: //<my-bucket>/ -- index-document index.html*
8. Upload the files to the bucket *s3 aws cp . s3//<my-bucket>/ --recursive –acl public-read.* Verify that the files were uploaded by using *aws s3 ls <my-bucket>*.
9. In the AWS S3 console, choose your bucket and in the Properties tab, scroll down tot Static Web hosting and choose the bucket website endpoint link to load it in a new browser tab.
10. In the terminal window, change directories and create an empty file by using the *cd ~ touch update-website.sh* commands.
11. Open the empty file in the vi editor using *vi update-website.sh* command.
12. Add the standard line of bash and then the s3 cp line using the commands *#!/bin/bash aws s3 cp ~/file-name/ s3://<my-bucket>/ --recursive –acl public-read.* Write the changes by pressing ESC and type :wq and then ENTER.
13. Make it an executable batch file using *chmod +x update-website.sh* command.

14. Open the local copy of the index.html file in a text editor using the vi command and modify the file by changing the bgcolor=”aquamarine to bgcolor=”cornsilk”. Locate the other colors in the file and modify it to Gainsboro and forestgreen. Save the changes and run the batch file to update the changes using the *./update-website.sh* command.
15. The batch file will upload every file to Amazon S3 everytime you run it. Return to the browser and refresh the café website to view the changes to the website.

Figure 1:

Creating an S3 bucket using the AWS CLI

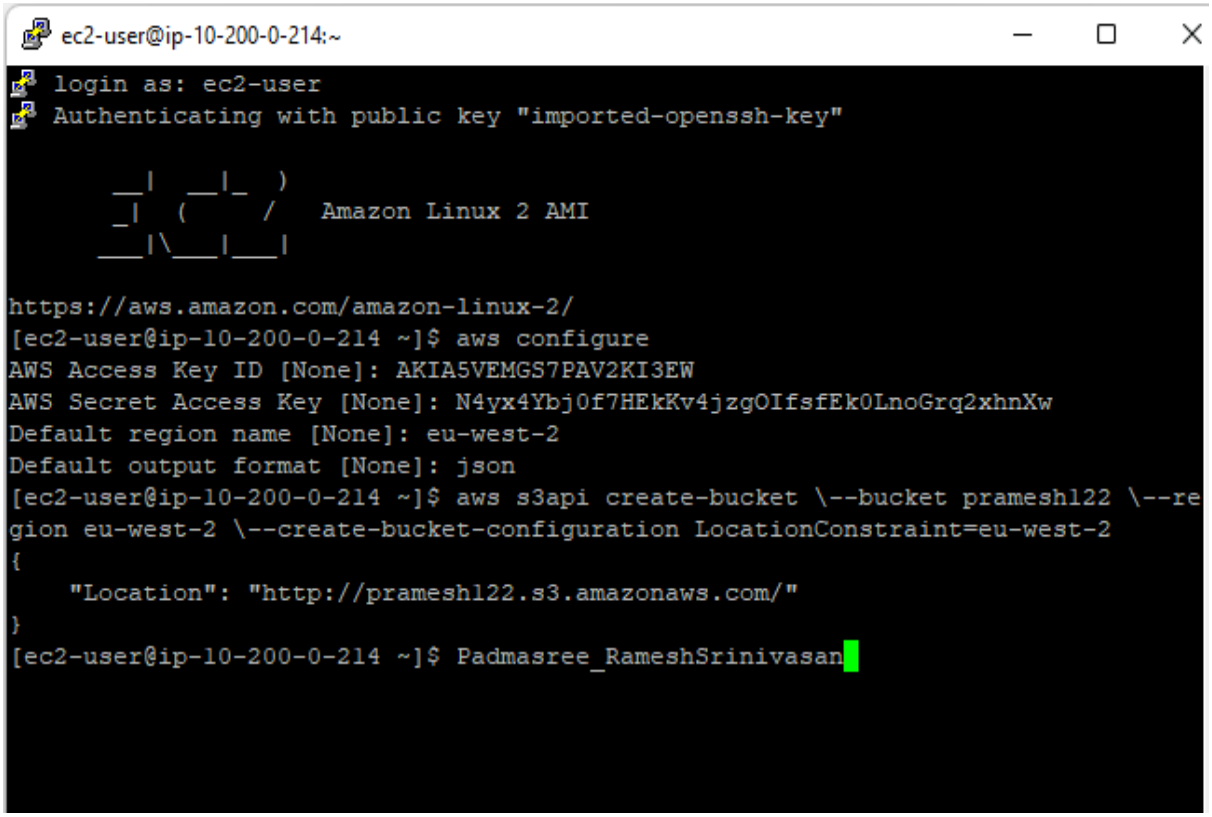


Figure 2:

Logged in as awsS3User

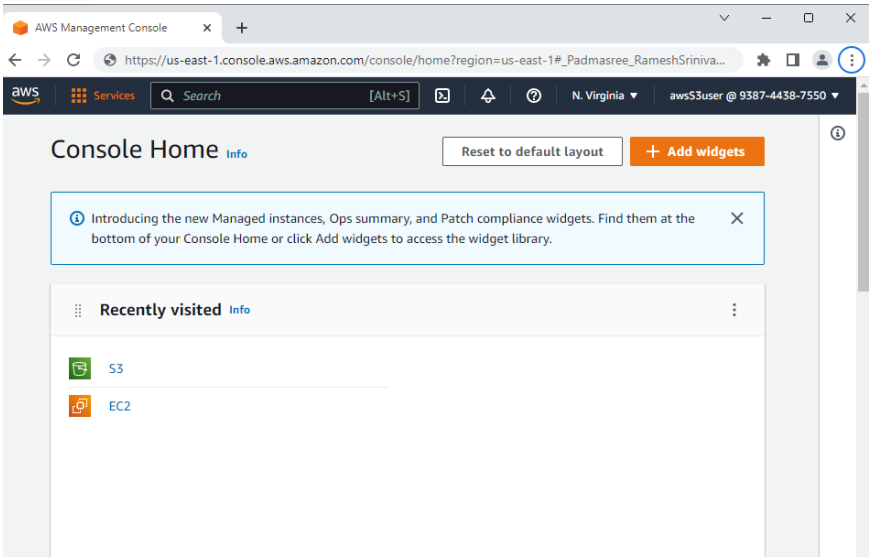


Figure 3:

The created website that is available for viewing

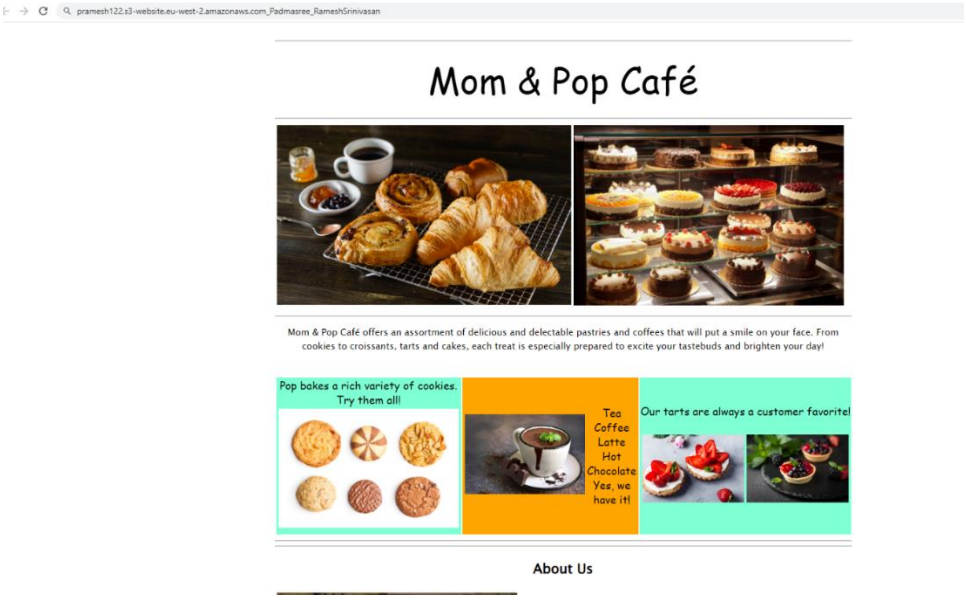


Figure 4:

Output for cat update-website.sh

How to launch an EC2 instance with the following settings:

- **OS/AMI: Amazon Linux 2**
- **Instance Type: t1.micro**

1. To launch an Instance using the Management console, choose the *Services* menu, locate *Compute* services, and select *EC2*.

2. Choose the Launch Instance button and name the instance as Bastion Server. Choose an AMI from the list of available Quick Start AMIs and select the *Amazon Linux 2 AMI(HVM)*. In the Instance type panel, select *t1.micro*.

3. Next in the Network settings, choose the VPC where you want to launch your instance and for subnet, click *on Public Subnet*. Keep the Auto-assign public IP as *Enable*. Under Firewall, click on Create Security group option and *configure a security group*. Give the Security group a name and description. *Allow inbound access via SSH (port 22*).

4. In the *Configure storage* section, keep the *default settings*. In the Advanced details panel, under IAM Instance Profile, choose Bastion Role. This Bastion-Role will grant permission to applications running on the instance to make requests to the web server.

5. At the bottom of the Summary panel, choose *Launch instance* and you will see a Success message. Now choose *View all instances*. The Bastion Server instance will first appear in the

Pending state, which means it has been launched. The state will then change to *Running*, which indicates that the instance has started booting. And the status check will display as *2/2 checks passed*.

6. Select the Bastion Server instance, and review the information in the Details tab that displays below and copy the IPv4 Public IP which you can use to SS HH into the server using the command *ssh -i key-name.pem ec2-user@<public-ip>*

7. To launch an Instance using CLI, in the active SSH session, enter the following commands that obtains the region where the instance is running, calls the AWS Systems Manager and uses the get-parameter command to retrieve the value from the Parameter store. The AMI requested was for Amazon Linux 2 and the AMI ID has been stored in an Environment Variable called AMI.

```
#Set the Region
AZ= curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone export AWS_DEFAULT_REGION=${AZ::-1}
 # Obtain latest Linux AMI
AMI = $(aws ssm get-parameters -names /aws/service/ami-amazon-linux-latest/amazon2-ami-hvm-x86_64-gp2 -query 'Parameters[0].[value]' -output tex
echo $AMI
 #To retrieve the SubnetID for the Public Subnet
SUBNET =$(aws ec2 describe-subnets -filters 'Name=tag:Name, Values=Public Subnet' -query Subnets[].SubnetID -output text)
Echo $SUBNET
 # To retrieve the Security group ID
SG=$(aws ec2 describe-security-groups -filters Name=group-name, Values=WebSecurityGroup -query SecurityGroups[].GroupId -output text)
```

8. To launch an instance that will act as a Web Server, you need to provide a User Data script that will automatically run when the instance launches. Use the command below for downloading the UserData Script. The script installs a web server, downloads the zip file containing the web application and installs the web application.

```
#To download User Data script
wget https://aws-tc-largeobjects.s3.amazonaws.com/ILT-TF-200-ACSOPS-1/lab-2-ec2-linux/UserData.txt
 # To view the contents of the script
cat UserData.txt
#Launch the instance using the following command.
INSTANCE=$(\
aws ec2 run-instances \
--image-id $AMI \
--subnet-id $SUBNET \
--security-group-ids $SG \
--user-data file:///home/ec2-user/UserData.txt \
--instance-type t1.micro \
--tag-specifications 'ResourceType=instance, Tags=[{Key=Name, Value=Web Server}]' \
--query 'Instances[*].InstanceId' \
--output text \
)
echo $INSTANCE
```

10. To describe all the information related to the instance in JSON format use the *command aws ec2 describe-instances –instance-ids $INSTANCE*. To get specific information about the instance, like the instance State, use the command

```
aws ec2 describe-instances -instance-ids $INSTANCE -query 'Reservations[].Instances[].State.Name' -output text
```

11. Once the commands return a Running State response, you can retrieve a URL to the instance with the following command

```
aws ec2 describe-instances -instance-ids $INSTANCE -query Reservations[].Instances[].PublicDnsname -output text.
```

12. Once the DNS name of the instance is returned, paste the DNS name into a new web browser. The web page should be displayed which demonstrates that the web server was successfully launched and configured using the AWS CLI.

Figure 1

Adding a name tag for the instance

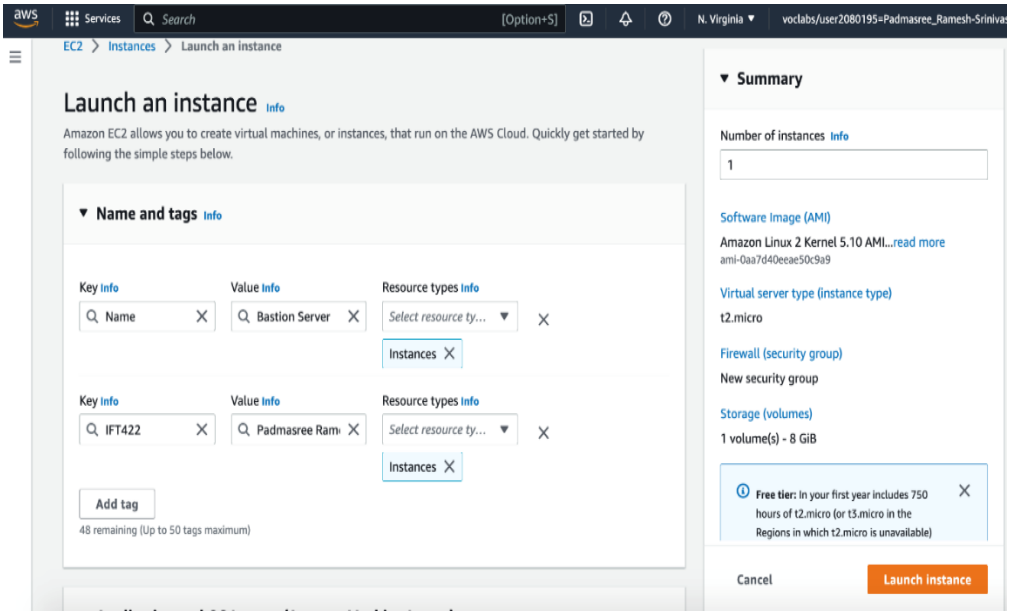


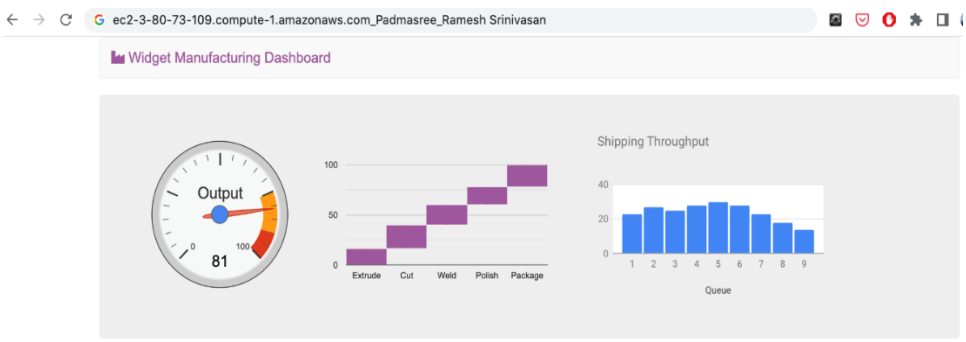
Figure 2

Launching an EC2 instance using AWS CLI

```
100%[=========================================================================================================================

2023-01-27 04:06:50 (18.3 MB/s) - 'UserData.txt' saved [308/308]

[ec2-user@ip-10-0-0-25 ~]$ cat UserData.txt
#!/bin/bash
# Install Apache Web Server
yum install -y httpd

# Turn on web server
systemctl enable httpd.service
systemctl start  httpd.service

# Download App files
wget https://aws-tc-largeobjects.s3.amazonaws.com/CUR-TF-200-RESOPS/lab2vocareum/dashboard-app.zip
unzip dashboard-app.zip -d /var/www/html/
[ec2-user@ip-10-0-0-25 ~]$ INSTANCE=$(\
> aws ec2 run-instances \
> --image-id $AMI \
> --subnet-id $SUBNET \
> --security-group-ids $SG \
> --user-data file:///home/ec2-user/UserData.txt \
> --instance-type t2.micro \
> --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Web Server}]' \
> --query 'Instances[*].InstanceId' \
> --output text \
> )
[ec2-user@ip-10-0-0-25 ~]$ echo $INSTANCE
i-0c5bd2156367cc3bb
[ec2-user@ip-10-0-0-25 ~]$ Padmasree_RameshSrinivasan
```

Figure 3

Testing the Webserver



How to fix a misconfigured webserver with Security Group issue:

1. Click on the misconfigured Webserver instance and go through the instance details in the panel that displays below.

2. Click on the Security tab. The given security group Inbound rules only allows HTTP traffic on Port 80 and the Outbound rules allows all traffic on all ports.

3. To allow SSH access to the web server, Click on Security groups under Network & Security from the left pane, select the misconfigured Webserver security group and click on Edit Inbound rules.

4. Click on Add rule. Under type, select SSH and type Port range as 22 and source type as My IP.

5. Save changes. Make sure that you have the key pair to SSH into the instance.

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

1. Open the create-lamp-instance.sh script file in a text editor such as VI using the command *vi create lamp-instance.sh*

2. Analyze the contents of the script. The describe-regions command queries for the MomPopCafe VPC and captures the VPC ID and the region where the LAMP instance should be deployed. The script also looks up the Subnet ID, keypair name and AMI ID values that will be needed to create an EC2 instance. The script creates a security group with ports 22 and 80 open. The instance Details variable is used to call to create the instance.

3. Exit the editor by pressing q*!* command.

4. Display the contents of the user-data script by running the *cat create-lamp-instance-userdata.txt* This command will install a web server, PHP and a database server.

5. Run the script using the *./create-lamp-instance.sh* command.

6. If you run into any errors while running the script, open the script again using the vi create lamp-instance.sh command.

7. Locate the line in the bash script that leads to the error. Make sure that the AMI is in the same region where you want to create the instance.

8. To be able to connect with the instance using SSH, make sure that the instance is using the correct key pair in the script. Add the key-pair<value> in the create-instances section.

9. The web server runs on TCP port 80. You can load the website in a browser using the *http://<public-ip*> where public-ip is the IPv4 Public Ip address of the instance.

Figure 1

Output of cat create-lamp-instance.userdata.txt command

```
[ec2-user@cli-host starters]$ cp create-lamp-instance.sh create-lamp-instance.backup
[ec2-user@cli-host starters]$ vi create-lamp-instance.sh
[ec2-user@cli-host starters]$ cat create-lamp-instance-userdata.txt
#!/bin/bash
yum -y update
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum -y install httpd mariadb-server

systemctl enable httpd
systemctl start httpd

systemctl enable mariadb
systemctl start mariadb

echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php

usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www

#Check /var/log/cloud-init-output.log after this runs to see errors, if any.

#
# Download and unzip the Mom & Pop Cafe application files.
#

# Database scripts
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/ILT-TF-200-ACSOPS-1/activity-
3/momPopDb.tar.gz
tar -zxvf momPopDb.tar.gz

# Web application files
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/ILT-TF-200-ACSOPS-1/activity-
3/mompopcafe.tar.gz
tar -zxvf mompopcafe.tar.gz -C /var/www/html/

#
# Run the scripts to set the database root password, and create and populate the applicat
ion database.
# Check the following logs to make sure there are no errors:
#
#       /momPopDb/set-root-password.log
#       /momPopDb/create-db.log
#
cd momPopDb
./set-root-password.sh
./create-db.sh
hostnamectl set-hostname web-server
[ec2-user@cli-host starters]$ Padmasree_Ramesh Srinivasan
```

How to tail a log in Linux

1. In the terminal window where you have an active SSH connection to the LAMP instance, run the following command to see the log file entries.

```
sudo tail -f /var/log/cloud-init-output.log
```

If you want to view the entire log file, use the *sudo cat /var/log/cloud-init-output.log*

2. On an Amazon Linux instance, the user-data file commands are run by the **cloud-init service.**

3. Observe the log file entries. The message is related to the installation of the database and PHP.

4. You can also see messages related to momPopDb files that were downloaded and extracted onto this instance, including the message: *Crate Database script completed.*

5. To change the number of lines that are displayed in the tail command, use the *-n* (number of lines) option or you can use a hyphen "-" and the number.

Example: tail -5 /var/log/output.log

6. To create a real-time, streaming output of a changing file, use the -f or –follow options

Example: tail -f /var/log/output.log

7. When you are finished with your observations, use the *Ctrl+C* command to exit the tail utility.

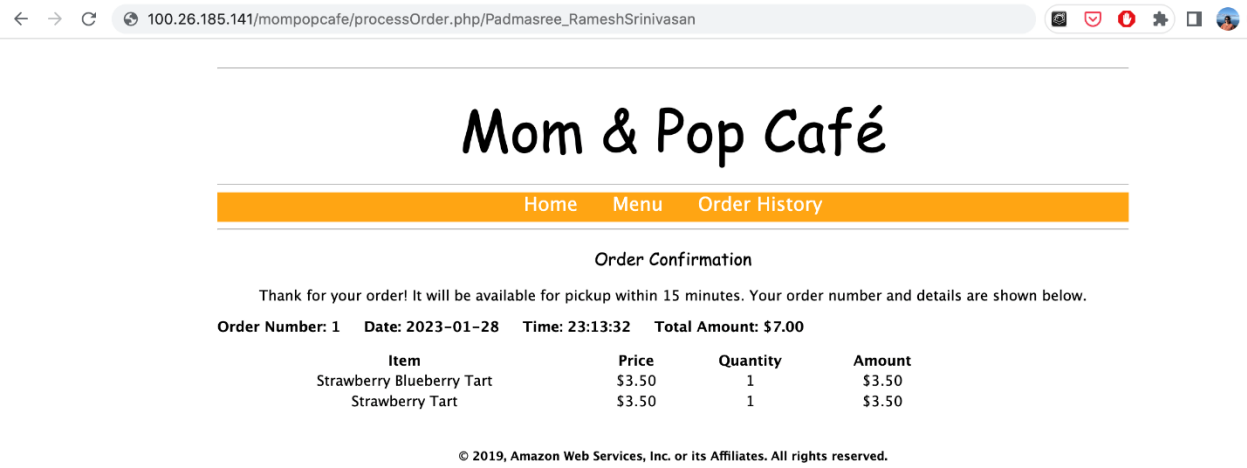
Figure 2:

Output of the tail command on the cloud-int-output.log

```
  Installing : 2:nmap-ncat-6.40-13.amzn2.x86_64
  Installing : 2:nmap-6.40-13.amzn2.x86_64
  Verifying  : 2:nmap-6.40-13.amzn2.x86_64
  Verifying  : 2:nmap-ncat-6.40-13.amzn2.x86_64

Installed:
  nmap.x86_64 2:6.40-13.amzn2

Dependency Installed:
  nmap-ncat.x86_64 2:6.40-13.amzn2

Complete!
[ec2-user@web-server ~]$ nmap -Pn 100.26.185.141

Starting Nmap 6.40 ( http://nmap.org ) at 2023-01-29 04:06 UTC
Nmap scan report for ec2-100-26-185-141.compute-1.amazonaws.com (100.26.185.141)
Host is up (0.00064s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 4.44 seconds
[ec2-user@web-server ~]$ sudo tail -f /var/log/cloud-init-output.log
mompopcafe/css/menu.css

Set Root Password script completed.
Please check the set-root-password.log file to verify successful execution.


Create Database script completed.
Please check the create-db.log file to verify successful execution.

Cloud-init v. 19.3-46.amzn2 finished at Sun, 29 Jan 2023 04:05:14 +0000. Datasource DataSourceEc2.  Up 97.81 seconds


^C
[ec2-user@web-server ~]$ cat create-db.log
cat: create-db.log: No such file or directory
[ec2-user@web-server ~]$ sudo tail -f /var/log/cloud-init-output.log
mompopcafe/css/menu.css

Set Root Password script completed.
Please check the set-root-password.log file to verify successful execution.


Create Database script completed.
Please check the create-db.log file to verify successful execution.

Cloud-init v. 19.3-46.amzn2 finished at Sun, 29 Jan 2023 04:05:14 +0000. Datasource DataSourceEc2.  Up 97.81 seconds
^C
[ec2-user@web-server ~]$ Padmasree RameshSrinivasan
```

Figure 3:

Screenshot of the order history page



How to create an AutoScaling Group in the AWS UI using Step scaling and Min: 5 nodes, Max: 10 nodes, Target: 7 nodes:

1. In the AWS Management console, on the Services menu, choose EC2 and then choose Instance from the left navigation pane. Choose the instance based on which you want to create a new AMI and copy the Public IPv4 address from the Details tab.

2. Make sure that you have the key pair to SSH into the instance. Use Putty/terminal window to SSH into the instance.

3. The script *UserData.txt* that was already installed in the running instance performs a number of initialization tasks like updating all the installed software and installs a small PHP web application that you can use to simulate a high CPU load on the instance. Inspect the script using the command *more UserData.txt*

4. Make note of the *AMI ID, HTTPAccess and SubnetID* and replace the values in the following command. This command output will provide you with an Instance ID. This value will be referred to as NEW-INSTANCE-ID in the subsequent steps.

aws ec2 run-instances –key-name vockey –instance-type t2.micro –image-id <AmiID> --user-data file:///home/ec2-user/UserData.txt --security-group-ids <HTTPAccess> --subnet-id <SubnetID> --associate-public-ip-address –tag-specifications 'Resourcetype=instance, Tags=[{Key=Name, Value=WebServerBaseImage}]' –output text –query 'Instances[*].InstanceId'

#To monitor instance status.

aws ec2 wait instance-running –instance-ids <NEW-INSTANCE-ID>

#To obtain the public DNS name of the new web server*aws ec2 describe-instances –instance-id <NEW-INSTANCE-ID> --query 'Reservations[0].Instances[0].NetworkInterfaces[0].Association.PublicDNSName'*

#In the web browser, replace PUBLIC-DNS-ADDRESS with the value you copied in the last step without the quotation marks

http://PUBLIC-DNS-ADDRESS/index.php

#Create a new AMI based on the new instance you just created.

aws ec2 create-image –name WebServer –instance-id <NEW-INSTANCE-ID>

5. On the Services menu, choose EC2. In the left navigation pane, choose Load balancers. Choose Create Load Balancer. Under Application Load Balancer, choose Create.

- Load Balancer Name – webserverloadbalancer

- In the Network Mapping section, select the VPC and the two Public Subnets because the load balancer will be internet facing.
- In the Security groups section, choose the HTTPAccess security group and remove the default security group.
- In the Listeners and routing section, choose Create Target Group.
    - Choose target type: Instances.
    - Target group name: webserver-app
    - Health check path: Enter /index.php
    - Healthy threshold – Enter 2
    - Interval -10 second.
- Choose Next and the Register targets screen appears. Targets are the individual instances that will respond to requests from the load balancer. You don't have any web application instances yet. So, we can skip this step.
- Review the settings and choose Create Target Group
- Return to the browser tab where you already were creating the Load Balancer.
- In the Listeners and routing section, choose the refresh icon.
- For the Listener HTTP:80 row, set the Default action to forward to webserver-app.
- Scroll to the bottom and choose Create Load Balancer
- The load balancer is successfully created. Choose View Load balancer.

6. The autoscaling group will use Launch Template to know which AMI to use to create new instances. In the left navigation pane, choose Launch Template

- Choose Create Launch Template.
- Launch template name – WebServerLaunchTemplate.
- In the AMI pane, search for WebServer and select it.
- In the Instance type, select t2.micro
- In the Network settings, for security groups, select HTTPAccess,
- In the Advanced details section, set Detailed CloudWatch monitoring to Enable.
- Choose Create Launch Template and you can see a success message. Choose View Launch Templates.

7. The AutoScaling group will create a minimum number of EC2 instances that will reside behind the load balancer. In the left navigation pane, choose AutoScaling Groups.

- Choose Create AutoScaling Group
- Choose Launch Template or configuration, configure
- Auto Scaling Group name: enter WebServerASGroup.
- Choose Launch Template or configuration, configure
    - Auto Scaling Group name: enter WebServerASGroup
    - Launch template: Choose WebServerLaunchTemplate
- Choose Next.
- In the Network pane, choose,
    - VPC

  - Select the Private Subnet 1 and Private Subnet 2
- Choose Next
- In the load balancing pane, choose Attach to an existing load balancer.
- In the attach to an existing load balancer pane, for Existing load balancer target groups, choose webserver-app
- Enable the group metrics collection within CloudWatch
- Choose Next
- In the Group size, configure.
  - Desired capacity – 7
  - Minimum capacity -5
  - Maximum capacity -10
- In the Scaling policy, select Step scaling policy.
  - Scaling Policy name – MyScalingPolicy
  - Choose metric type: Average CPU Utilization
- Choose Next
- On the Add notifications page, choose Next,
- On the tags page, choose Add tag and create a tag with key: Name and Value: webapp.
- Choose Next
- At the bottom of the Review page, choose Create Auto Scaling Group.

8. In the left navigation pane, choose Instances and verify that your desired number of instances named WebApp is being created as part of the AutoScaling Group. Wait for the instances to complete initialization and the status checks.

9. In the left navigation pane, choose Target Groups. Select your Target Group(webserver-app). In the Targets tab, in the lower half of the screen verify the creation of the instances and refresh until the Status of these instances changes to Healthy.

10. In the left navigation pane, choose Load balancer and select webserverloadbalancer.

11. From the description tab below, copy the DNS value. In the new web browser, copy the URL into the address bar and press Enter. On the Webpage, choose Start Stress.

12. This calls the application stress in the background which causes the CPU utilization on the instance to spike to 100 percent.

13. Return to the Auto Scaling Groups. Select WebServerASGroup. Choose Activity tab and scroll down to the Activity History pane and in a few minutes, you should see a new entry which indicates that your AutoScaling Group is launching new instance.

Figure 1:

Create a new instance

Figure 2:

Creating the load balancer in the console

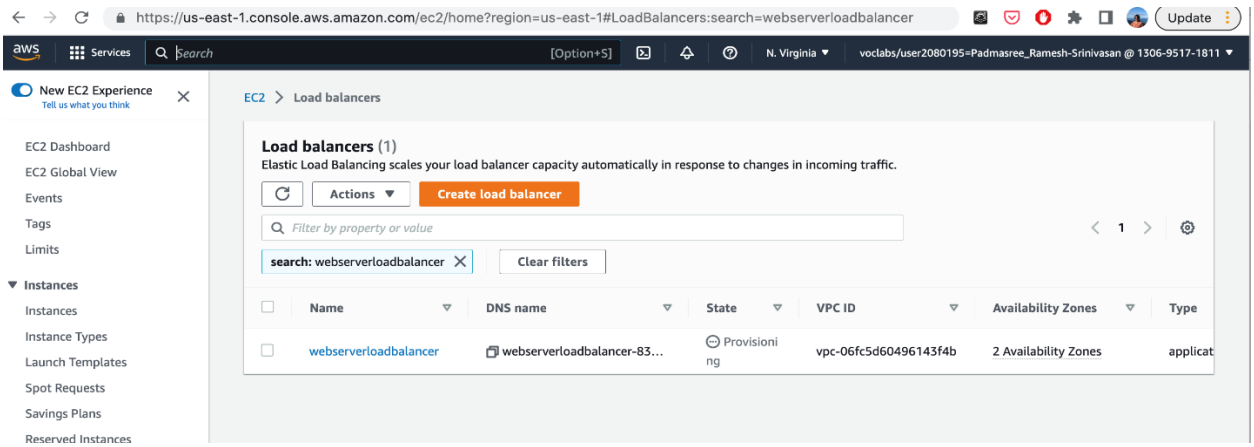
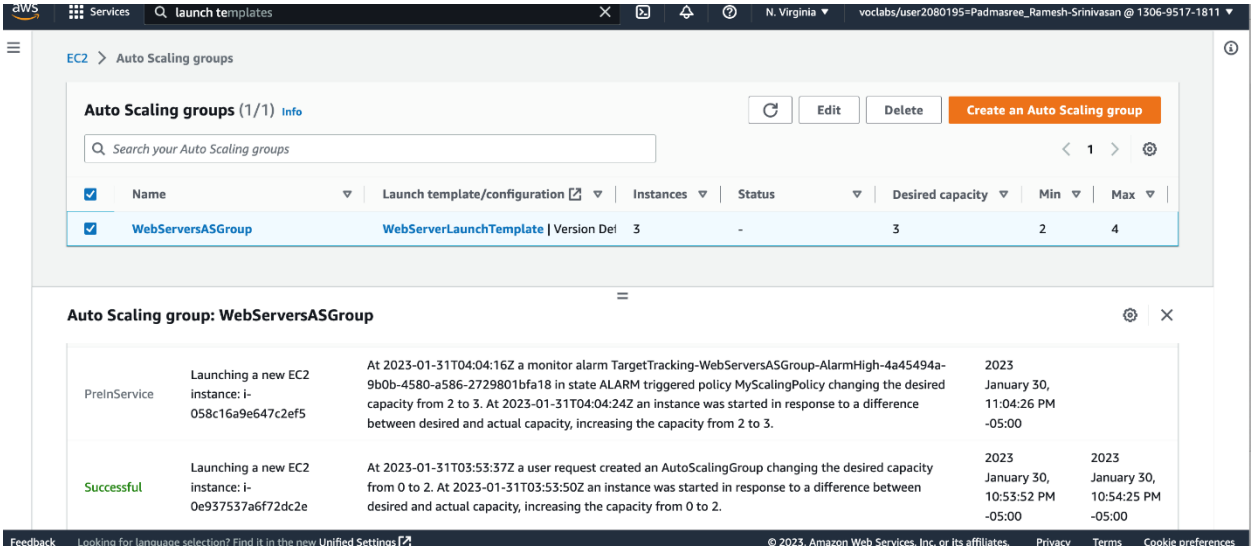


Figure 3:

Autoscaling activity:



How to create a Route 53 health check

1. In the AWs Management console, on the Services menu, select EC2 and then click the Instances. Two EC2 instances with MomPopCaféInstance1 is running on Public Subnet 1 and MomPopCaféInsatnce2 is running on Public Subnet 2. The role applied to the instance allows the instance access to AWS Systems Manager service.

2. Copy the IPv4 Public IP value from the MomPopCafeInstance1 and paste it into the URL address bar in a new browser tab. A message “Hello from Web Server” appears. Append the /mompopcafe to the end of the URL and load that page. The Mom and Pop Café website should load. Repeat the same steps with MomPopCafeInstance2 also and the Mom and Pop Café website should load as well.

3. In the AWS Management console, from the Services menu, choose Systems Manager and then Parameter store. View all the defined parameters and check the box next to /mompopcafe/showServerInfo and click Edit. In the Value field, change False to True and Save changes. Changing this will show additional information on the Mom Pop café web pages that will prove useful.

4. Return to the Mom and Pop webpage and try submitting an order.

5. In the AWS Management console, choose Services and then Route 53. In the left navigation pane, click Health checks. Click *Configure health check* and configure the following:

- **Name**: *Primary-Website-Health*
- **What to monitor**: *Endpoint*
- **Specify Endpoint by**: *IP Address*
- **IP Address:** *Paste in the Public IPv4 Address of MomPopCafeInstance1*
- **Path:** */mompopcafe*

Expand Advanced configuration and configure the following:

- Request Interval: *Fast (10 seconds)*
- Failure Threshold: 2
- Click *Next*

Configure the following:

- Create Alarm: *yes*
- Send Notification to: *New SNS topic*
- Topic Name: *Primary-Website-Health*
- Recipient email address: *enter an email address*

Click Create Health check. Route 53 will now check the health of your site by periodically requesting the domain name you provided and verifying that it returns a successful response. Refresh to see if the Health check shows a "*Healthy" status.*

6. Place a check in the box next to *Primary-Website-Health* and then click the *Monitoring* tab. This gives you a view of the status of the health check over time. It may take a few seconds before the chart becomes available. Click the refresh page section icon to update the view.

7. Check your email. You should have received an email from *AWS Notifications*. Click the *Confirm subscription* link contained in the email to finish setting up the email alerting that you configured when you created the health check.

8. You can configure failover routing based on the health check by creating a A record. In the Route 53 console, click on *Hosted zone* on the left navigation pane. Create A records using *Failing routing policy* for both instances as *Primary and secondary*.

9. You can verify the Failover functionality by changing the Instance state of the MomPopCaféIsntance1 to *Stop.* Then, go to Route 53 and click on Health checks. In the Primary-Website-health, in the lower pane, click *Monitoring.*

10. The status of Primary-Website-Health changes to *Unhealthy* after a few minutes. If you refresh the MomPopCafe webpage, you can see that the website is served from your MomPopCafeInstance2 instance.

11. Also, you should receive an email from AWS Notifications that the Primary-Website-Health is in *Alarm state* along with details of what triggered the alarm.

Figure 1:

Email subscription confirmation

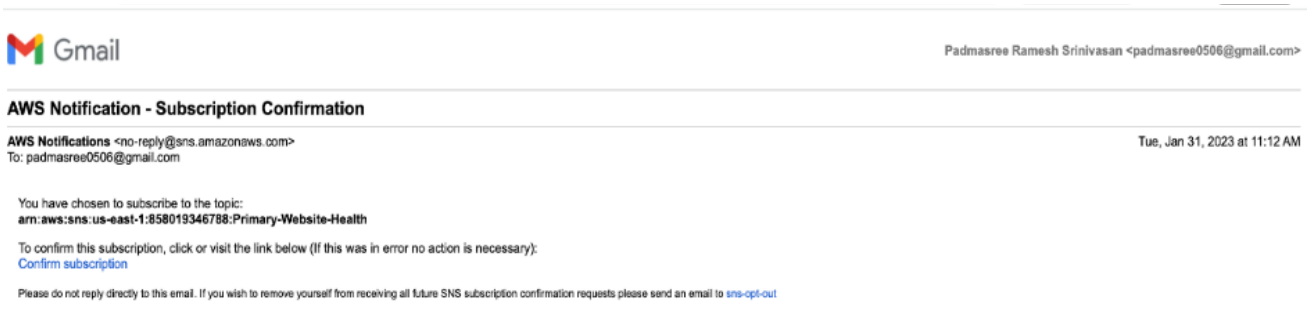


Figure 2:

MomPopCafe Website

Figure 3:

Email from AWS Notifications

Gmail — Padmasree Ramesh Srinivasan <padmasree0506@gmail.com>

ALARM: "Primary-Website-Health-awsroute53-8c104adf-cf78-4aa1-aaa3-85a8e..." in US East (N. Virginia)

AWS Notifications <no-reply@sns.amazonaws.com> Tue, Jan 31, 2023 at 11:38 AM
To: padmasree0506@gmail.com

You are receiving this email because your Amazon CloudWatch Alarm "Primary-Website-Health-awsroute53-8c104adf-cf78-4aa1-aaa3-85a8eb7b3ba9-Low-HealthCheckStatus" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 datapoint [0.0 (31/01/23 16:37:00)] was less than the threshold (1.0)." at "Tuesday 31 January, 2023 16:38:01 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/Primary-Website-Health-awsroute53-8c104adf-cf78-4aa1-aaa3-85a8eb7b3ba9-Low-HealthCheckStatus

Alarm Details:
- Name: Primary-Website-Health-awsroute53-8c104adf-cf78-4aa1-aaa3-85a8eb7b3ba9-Low-HealthCheckStatus
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint [0.0 (31/01/23 16:37:00)] was less than the threshold (1.0).
- Timestamp: Tuesday 31 January, 2023 16:38:01 UTC
- AWS Account: 858019346788
- Alarm Arn: arn:aws:cloudwatch:us-east-1:858019346788:alarm:Primary-Website-Health-awsroute53-8c104adf-cf78-4aa1-aaa3-85a8eb7b3ba9-Low-HealthCheckStatus

Threshold:
- The alarm is in the ALARM state when the metric is LessThanThreshold 1.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace: AWS/Route53
- MetricName: HealthCheckStatus
- Dimensions: [HealthCheckId = 8c104adf-cf78-4aa1-aaa3-85a8eb7b3ba9]
- Period: 60 seconds
- Statistic: Minimum
- Unit: not specified

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:us-east-1:858019346788:Primary-Website-Health]
- INSUFFICIENT_DATA:

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

1. In the AWS Management console, select Services > IAM. Choose *Roles*. In the Roles page, choose *the salesAnalysisReportRole* and the *salesAnalysisReportDERole* and review the permissions granted to these roles.
2. To create a Lambda layer, download the required files to your local machine *-pymysql-0.9.3.zip and salesAnalysisReportDataExtractor.zip.* The salesAnalysisReportDataExtractor is a Python implementation of a Lambda function that makes use of the PymySQL open source client library to access MySQL Mom & Pop Café database.
3. In the AWS Management console, select Services > Lambda. Choose *Layers* and then Choose *Create Layer.* Configure the following settings:
    - Name: pymysqlLibrary
    - Description: PyMySQL 0.9.3 library modules

- Code entry type: Upload a .zip file
- Choose *Upload*, navigate to the folder where you downloaded pymysql-0.9.3.zip and open it.
- Compatible runtimes: Choose Python 3.7
- Choose *Create*
- The message *Successfully created layer pymysqlLibrary version 1* is displayed.

4. In the *Function overview* panel, under salesAnalysisReportDataExtractor, choose Layers. In the Layers panel at the bottom of the page, choose *Add a layer*. In the Add layer page, configure:

    - Choose a layer: Select *Custom layers card.*
    - Custom layers: pymysqlLibrary
    - Version: 1
    - Choose Add

    The function detail page is displayed with the overview panel that shows a count of (1) in the Layers node.

How to create a Lambda function from a prebuilt package:

1. In the Function detail page, scroll down to the *Runtime Setting* panel, and choose Edit. In the handler field, change the value to *salesAnalysisReportDataExtractor.lambda_handler*. Choose Save. Scroll back to the *Code Source* panel and choose *Upload From*. Select *.zip* file. Choose *Upload*, then navigate to *salesAnalysisReportDataExtractor.zip* that you downloaded earlier and choose *Save*. Now the Lambda function code is imported.

2. This function requires network access to the Mom & Pop Café database which runs in an EC2 LAMP instance. To configure the network settings of this function, choose the Configuration tab, then choose VPC, and choose Edit. Configure the VPC settings:
    - VPC: MomPopCafeVPC
    - Subnets: MomPopCafe Public Subnet 1
    - Make sure that the Inbound Security group has access to Port 80, 22 and 3306. Choose Save.
3. To invoke the salesAnalysisReportDataExtractor function, you need to supply values for the Mom & Pop Database connection parameters that are stored in the AWS Systems manager Parameter store.
    - In a new browser tab, open the AWS Management console, select Services > Systems Manager. Choose Parameter store and make note of the following values- /mompopcafe/dBrl,dBName,dBUser,dBPassword
    - In the Lambda Management Console tab, and in the salesAnalysisReportDataExtractor function details page, choose the Test tab. Configure the test event as:
        - New event: (selected)
        - Template: hello-world
        - Name: SARDETestEvent
        - Replace JSON object in the editor pane with the values of dbUrl, dbName, dbUser and dbPassword.
    - Save changes and choose Test. You should see a green box message "*Execution result: succeeded(logs)"*

4. To create a SNS topic, use the Simple Notification service and configure a Standard topic and then create a subscription to that topic using Email protocol. Confirm subscription to that topic from your email account.
5. SSH into the CLI host instance using the key pair. Update the AWS CLI software using aws configure command and credentials AWS Access key ID, Secret Access key, default region name and output format.
6. Verify that the file containing the code for the salesAnalysisReport lambda function is already on the instance by running the command.

```
cd activity files
ls
```

   Retrieve the ARN of the salesAnalysisReportRole IAM Role from the IAM console, Roles > Role name. The Role ARN appears in the Summary page.

7. To create the salesAnalysisReport function, type in the following command:

```
aws lambda create-function \
--function-name salesAnalysisReport \
--runtime python3.7 \
--zip-file fileb://salesAnalysisReport.zip \
--handler salesAnalysisReport.lambda_handler \
--region <region> \
--role <salesAnalysisReportRoleARN>
```

   This command will return a JSON object describing the attributes of the function.

8. In the Lambda console, choose Functions and then choose the salesAnalysisReport hyperlink. In the Configuration tab, choose Environment variables, choose Edit and then add a new environment variable with key as topicARN and ARN value of the salesAnalysisReportTopic that was created earlier. Choose Save.

9. Choose the Test tab and configure a test event and choose Test. You should see a Succeeded message. Check your email. You should receive an email that contains the report of the orders placed on the website.

Figure 1:

Create Lambda layer.

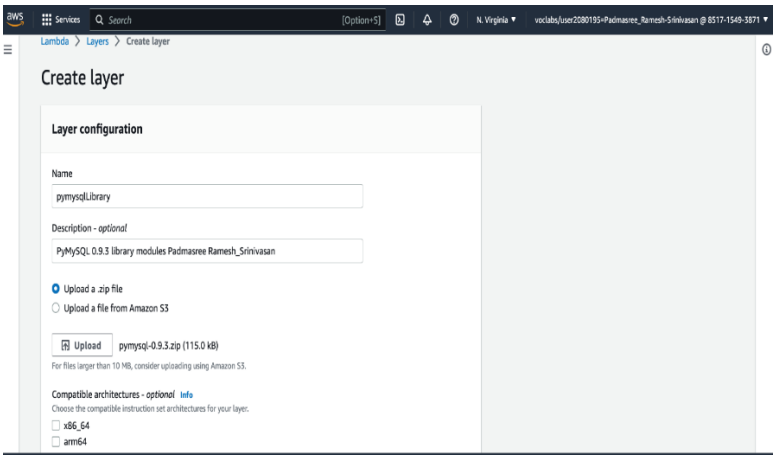


Figure 2:

Troubleshoot failing Lambda function.

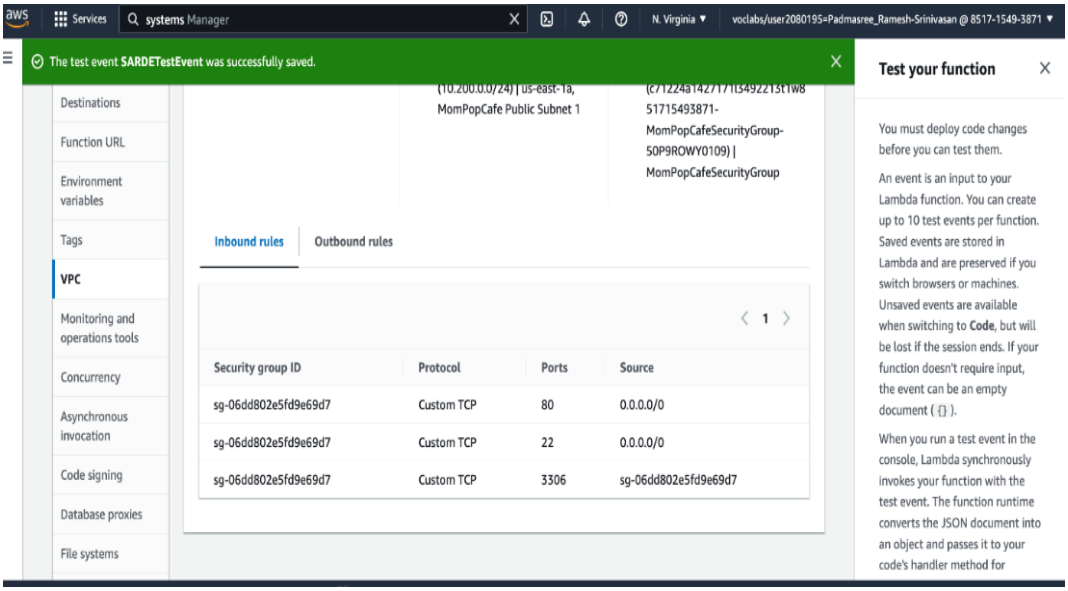


Figure 3:

Subscription confirmation email

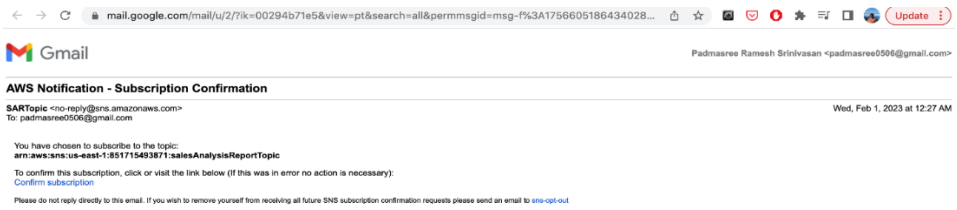


Figure 4:

salesAnalysisReport function

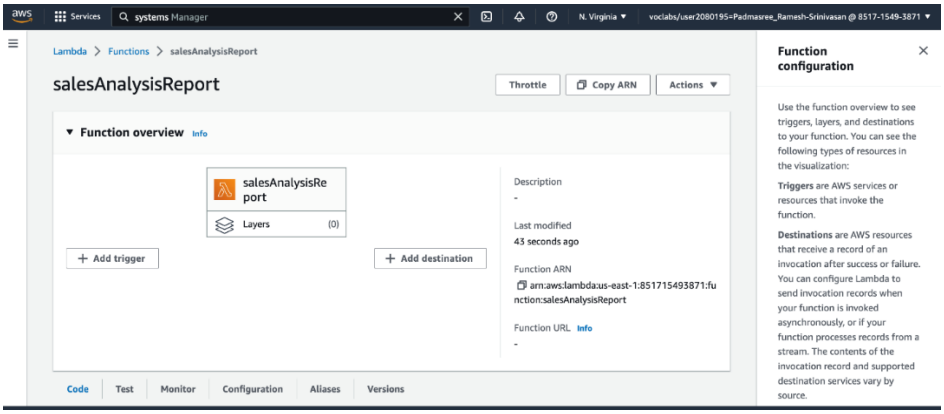


Figure 5:

Daily Sales analysis report email

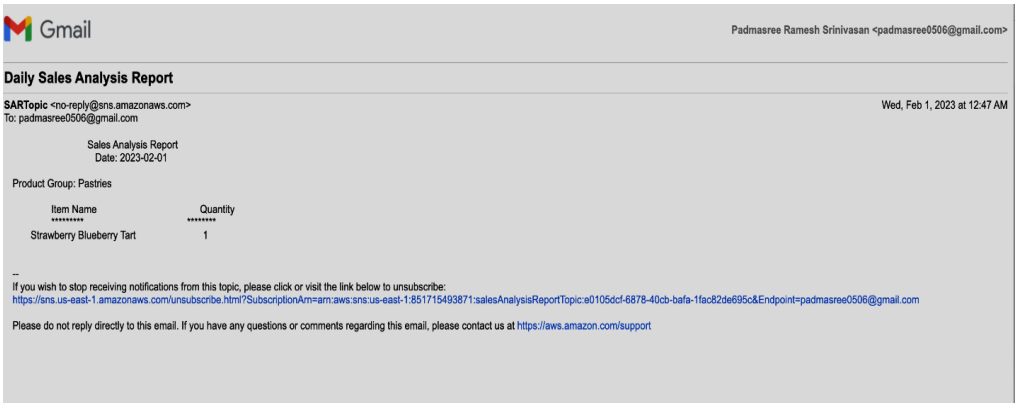
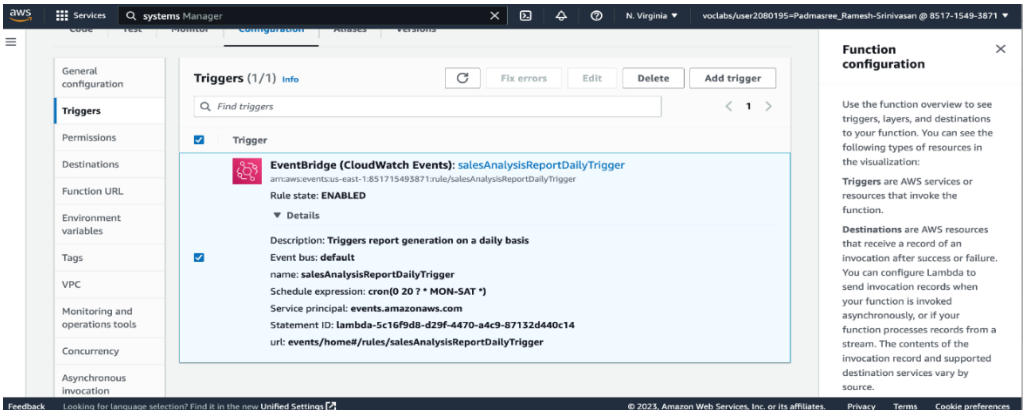


Figure 6:

Eventbridge rule



How to setup a VPC:

1. In the search box to the right of Services, search for and choose **VPC** to open the VPC console. The VPC console offers a VPC Wizard that can automatically create several VPC architectures. In the left navigation pane, choose **Your VPCs.**

   A default VPC is provided so that you can launch resources as soon as you start using AWS.

2. Choose Create VPC and configure:
   - **Name tag:** Lab VPC
   - **IPv4 CIDR block:** 10.0.0.0/16
   - Choose **Create VPC.**
3. Choose Actions and select **Edit DNS hostnames.** Under DNS hostnames, select Enable, then choose Save changes.

   Any Amazon EC2 instances launched into the VPC will now automatically receive a DNS hostname.

4. A subnet is a sub-range of IP addresses within the VPC. The public subnet will be used for internet-facing resources. In the navigation pane, choose Subnets. Choose Create Subnet and configure:
    - **VPC ID:** Lab VPC
    - **Subnet name:** Public subnet
    - **Availability zone:** Select the first AZ in the list.
    - **IPv4 CIDR block:** 10.0.0.0/24
    - Choose **Create subnet.**
5. The private subnet will be used for resources that are to remain isolated from the Internet.
    - **VPC ID** : Lab VPC
    - **Subnet name**: Private subnet
    - **Availability zone:** Select the first AZ in the list.
    - **IPv4 CIDR block**: 10.0.2.0/23

    Now the VPC has two subnets.
6. An Internet gateway provides a target for route tables to connect to the internet and also performs network address translation (NAT) for instances that have been assigned IPv4 public Ip addresses. In the left navigation pane, choose **Internet gateways**. Choose Create Internet gateway and configure:
    - **Name tag**: Lab IGW
    - Choose **Create internet gateway**.

    To attach the internet gateway to Lab VPC, choose **Actions** then Attach to VPC and configure:
        - Under Available VPCs choose **Lab VPC.**

- Choose **Attach internet gateway.**

Now the public subnet as connection to the internet.

7. A **route table** contains a set of rules called routes that are used to determine where network traffic is directed. In the left navigation pane, choose **Route tables**.

   Select the route table that shows Lab VPC in the VPC column. Choose the Name column, enter the name as**: Private Route table**, then choose Save. Choose the **Routes** tab in the lower half of the page. There is currently only one route. It shows all traffic destined for 10.0.0.0/16 will be routed locally.

8. To create a new public route table to send public traffic to the internet gateway, choose **Create route table** and configure:

   - **Name tag**: Public Route Table
   - **VPC:** Lab VPC
   - Choose **Create route table**.

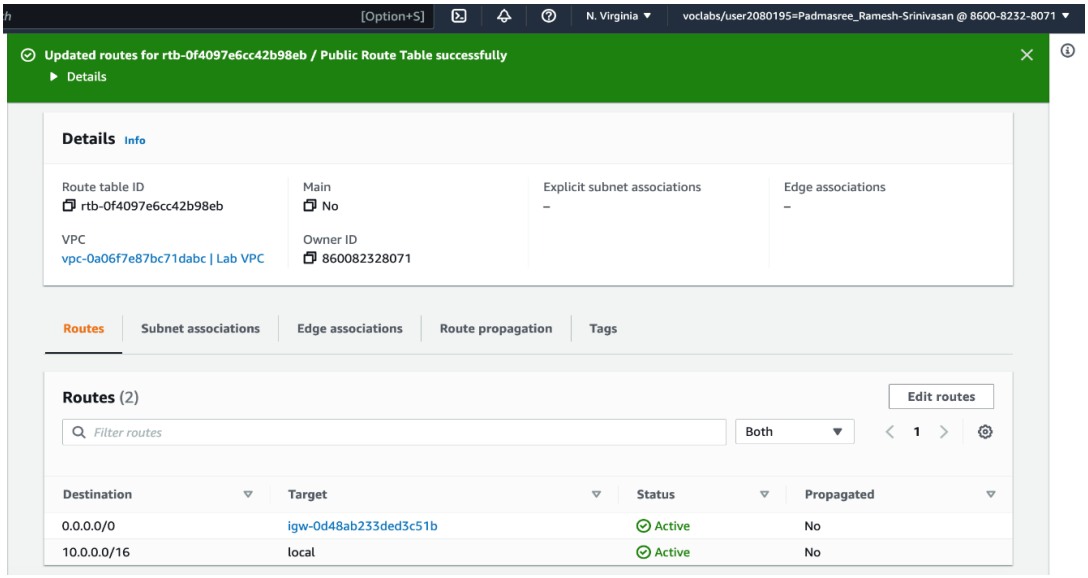
   In the **Routes** tab, choose **Edit routes**. You will now add a route to direct internet-bound traffic (0.0.0.0/0) to the Internet gateway.

   Choose **Add route** then configure:

   - **Destination**: 0.0.0.0/0
   - **Target**: Select Internet gateway then select Lab IGW from the list
   - Choose **Save changes.**

9. To associate the new Route Table with the public subnet, choose **Subnet associations** tab. Choose **Edit subnet associations**.

10. Select ☑ the row with Public Subnet. Choose **Save associations**.



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

1. A bastion server also known as a Jump box is an Amazon EC2 instance in a Public subnet that is securely configured to provide access to resources in a private subnet. System Operators can connect to the Bastion Server and then jump into resources in the private subnet.
2. In the search box to the right of **Services,** search for and choose **EC2** to open the EC2 console.
3. From the Launch Instance menu, choose Launch Instance and configure:
    - Name and tags:
        - **Name:** Bastion Server
    - Application and OS images:
        - **Quick Start:** Amazon Linux
        - **AMI:** Amazon Linux 2 AMI (HVM)

- Instance type:
    - **Instance type:** t2.micro
- Key pair (login):
    - **Key pair name:** vockey
- Network settings:
    - Choose **Edit**
    - **VPC**: Lab VPC
    - **Subnet**: Public Subnet
    - **Auto-assign public IP**: Enable
    - **Security group name**: BastionSG
    - **Description**: BastionSG
    - **Inbound security groups rules**: Keep the default setting which will provide SSH access.
- Configure storage:
    - Use the default settings (no changes)
- Review the **Summary** displayed on the right of the screen and choose **Launch Instance**
- On the next page, choose **View all instances.**

The Bastion server will be launched in the Public Subnet.

4. To launch an Amazon EC2 instance in the Private subnet, configure:

- Name and tags:
    - **Name**: Private Instance
- Application and OS Images:

- **Quick Start** Amazon Linux
- **AMI**: Amazon Linux 2 AMI (HVM)

- Instance type:
  - **Instance type**: t2.micro
- Key pair(login):
  - **Key pair name:** vockey
- Network settings:
  - Choose **Edit**
  - **VPC:** Lab VPC
  - **Subnet**: Private Subnet
  - **Auto-Assign public IP**: Disable
  - **Security group name**: PrivateSG
  - **Description;** PrivateSG
  - **Inbound security groups rules**: Keep the default setting which will provide SSH access.
- Configure storage:
  - Use the default settings.
- Review the Summary displayed on the right of the screen and choose **Launch Instance**
- On the next page, choose **View all instances**.

5. To log in to the instance launched in the Private subnet, we will first log in to the Bastion Server in the public subnet and then log into the Private Instance from the Bastion Server.

Select the Bastion Server and copy the address shown under Public IPv4 address to your clipboard.

6. For Mac and Linux users, in the AWS Management console, on the Services menu, choose EC2. In the left navigation pane, choose Instances. Select the Bastion Host. Copy the address shown under Public IPv4 address from the Description in the lower pane. Make sure that your have the key pair .pem file saved.
7. Open a terminal window, and change directory cd to the directory where the labsuser.pem file was downloaded.

```
cd ~/Downloads
```

Change the permissions on the key to be read only, by running this command:

```
chmod 400 labsuser.pem
```

Return to the terminal window and run this command (replace <public-ip> with the Public IPv4 value you copied to your clipboard earlier in the lab):

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

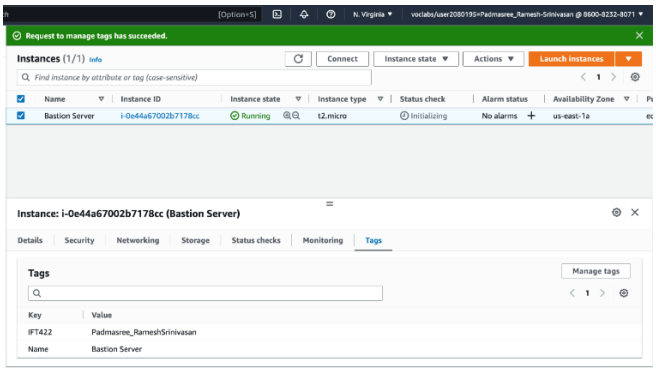
Type yes when prompted to allow a first connection to this remote SSH server.

8. To log into the private instance in the Private Subnet, select the Private Instance and deselect any other instances. Copy the address shown under Private IPv4 addresses to your clipboard. This private IP address starting with 10.0.2.x or 10.0.3.x is not reachable directly from the Internet which is why you first logged into the Bastion Server and then yoy will log into the Private Instance.
9. Run this command into the ssh session, replacing Private-IP with the address you just copied to your clipboard.

```
ssh PRIVATE-IP
```

10. If you are prompted with “Are you sure”, enter yes.

    Now you should be connected to the Private instance. This was accomplished by first connecting to the Bastion Server in the Public Subnet, then connecting to the Private Instance in the Private Subnet.



How to enable VPC Flow Logs via the command line interface

1. Create the S3 bucket that will hold the flow logs using the following command:

```
aws s3api create-bucket --bucket flowlog####
--region <region>
--create-bucket-configuration LocationConstraint=<region>
```

In this command, replace #### with four random numbers and replace both occurrences of **<region>** with the region where the EC2 instances were created.

2. Run the following command to get the VPC ID for VPC1, which you must have to enable VPC Flow logs:

```
aws ec2 describe-vpcs
--query 'Vpcs[*].[VpcId,Tags[?Key==`Name`].Value,CidrBlock]'
--filters "Name=tag:Name,Values='VPC1'"
```

3. Enable VPC Flow Logs on VPC1 by running the following command:

```
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####>
```

In the command above, replace <flowlog####> with the actual bucket name. Also replace <vpc-id> with the actual VPC ID of VPC1. The VPC ID was returned by the describe-vpcs command that you ran.

4. If the command runs successfully, you should see that a **FlowLogID** and a **ClientToken** are returned.
5. Run the following command to confirm that the flow log was created:

```
aws ec2 describe-flow-logs
```

The command output should show a single flow log was created with a FlowLogStatus of **ACTIVE** and a log destination that points to your S3 bucket.

```
    ]
]
[ec2-user@cli-host ~]$ aws ec2 create-flow-logs --resource-type VPC --resource-ids vpc-003ee5011692a5fa4 --traffic-type ALL --log-destination-type s3 --log-destination arn:aws:s3:::
flowlog6789
{
    "Unsuccessful": [],
    "FlowLogIds": [
        "fl-029511e24fba7c3fa"
    ],
    "ClientToken": "Js8clIJxl4cbv361OK2F8Zrgz8czjEA64cZCAJ2TsDU="
}
[ec2-user@cli-host ~]$ aws ec2 describe-flow-logs
{
    "FlowLogs": [
        {
            "LogDestinationType": "s3",
            "Tags": [],
            "ResourceId": "vpc-003ee5011692a5fa4",
            "CreationTime": "2023-02-15T16:53:48.922Z",
            "TrafficType": "ALL",
            "FlowLogStatus": "ACTIVE",
            "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end} ${action} ${log-status
}",
            "FlowLogId": "fl-029511e24fba7c3fa",
            "MaxAggregationInterval": 600,
            "LogDestination": "arn:aws:s3:::flowlog6789",
            "DeliverLogsStatus": "SUCCESS"
        }
    ]
}
[ec2-user@cli-host ~]$ Padmasree_RameshSrinivasan
```

How to troubleshoot network connectivity on an instance

1. Even after setting up the web server instance that is running, if you cannot load the webpage the web server instance should serve and if you also cannot use SSH to connect to the web server instance, the following troubleshooting steps must be followed.
2. **Method 1:** Use the **nmap** utility to check what ports are open on the web server EC2 instance. To do this, you must first install the utility on the CLI Host instance by running *sudo yum install -y nmap*. Then run nmap <WebServerIP>, where <WebServerIP> is the actual public IP address of the web server instance.

   If nmap cannot find any open ports, we have to further investigate if there is something else blocking access to the instance.
3. **Method 2**: Check the security group details by using the aws ec2 describe-security-groups command.
    - It is easier to analyze the results of the command if you use the group-ids parameter. Also, the describe-instances command that you ran also returned the security group ID.
    - After you run the describe-security-groups command, analyze the resulting output. Make sure that the security group settings that are applied to the web server EC2 instance allow connectivity.
4. **Method 3:** Check the route table settings for the route table that is associated with the subnet where the web server is running.
    - Use the aws *ec2 describe-route-tables command*

- When you run the command, you might find it helpful to apply a filter like: *--filter "Name= association.subnet-id,Values='<VPC1PubSubnetID>'"*

Replace <VPCPubSubnetID> with the actual subnet ID value. The subnet ID value was also returned when you ran the describe-instances command.

5. **Method 4**: When you analyze the output of the describe-route-tables command, recall that the subnet is labeled as public.
    - If you notice any issues with the routes, you must define a new route. Use the *aws ec2 create-route command.*
    - You must know the route-table-id and gateway-id to successfully create a route.
    - You could use *aws ec2 describe-internet-gateways* to get the gateway-id if you want.
6. Now return to the web browser tab where you tried to load the web server page and refresh the webpage. The http://WebServerIP page should display Hello from your web server!
7. Try to use SSH to connect to the EC2 instance that hosts the web server. To connect to the web server instance
    - Disconnect your current SSH session with the CLI Host by entering exit.
    - Use the UP key on your keyboard to load the same SSH connection details that you used previously. Change the Public IP address at the end of the connection details to the <WebServerIP> address. Also add -o ConnectionTimeout=10 to the command.

- So, after 10 seconds, the attempt to connect via SSH also fails. You get an operation timed out error.

8. The web server is running. Make sure that you are using the correct key pair. Make sure that the route table entry to connect the subnet where the web server instance is running to the internet. Also check whether the security group allows connections on port 22.
    - Re-establish the SSH connection to the CLI Host and check the network access control list (NACL)

```
aws ec2 describe-network-acls
--filter "Name=association.subnet-id,Values='VPC1PublicSubnetID'"
--query 'NetworkAcls[*].[NetworkAclId,Entries]'
```

    - Analyze the output result from the above command. Use the delete-network-acl-entry command to delete any NACL entries that causes the issue.
9. Try to establish an SSH session from your computer to the web server again and confirm that you can connect.

```
[ec2-user@cli-host ~]$ aws ec2 describe-instances --filter "Name=ip-address,Values='52.90.108.109'" --query 'Reservations[*].Instances[*].[State,PrivateIpAddress,InstanceId,SecurityGroups,SubnetId,KeyName]'
[
    [
        [
            {
                "Code": 16,
                "Name": "running"
            },
            "10.0.1.9",
            "i-065dfaa310681e6b8",
            [
                {
                    "GroupName": "c71224a142717713579468t1w968770671120-WebSecurityGroup-WZVNK99HZEOM",
                    "GroupId": "sg-00b971e9a4d23fc9b"
                }
            ],
            "subnet-0f55565f5c0e070bf",
            "vockey"
        ]
    ]
]
[ec2-user@cli-host ~]$ Padmasree_RameshSrinivasan
```

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

1. Three EC2 Linux instances have been launched in two public subnets and one in private subnet. The instances in the public subnets are configured as proxy servers to forward traffic to the application server in the private subnet. They both allow HTTP traffic on port 80 from anywhere to be directed to the webserver instance.
2. In the VPC console, in LabVPC, choose Subnets. Select the checkbox for PrivateSubnet and then choose Route Table tab. Choose the Routes tab. Notice that all IP address in the LabVPC CIDR range (10.0.0.0/16) will be routed to local targets within the VPC. All other traffic (0.0.0.0/0) will be routed to NAT gateway. Likewise, PublicSubnetA route table routes 0.0.0.0/0 traffic to an Internet gateway.
3. Modify the AppServerSG inbound rules so that the allowed source for HTTP traffic is the ProxySG security group instead of the ProxyServer1 internal IP address.
    - In the Aamzon EC2 console, choose Security Groups. Select the AppServerSG group.
    - Choose the Inbound rules tab and then click Edit Inbound rules.
    - Delete the existing inbloudn rule and choose Add rule
    - Type: Choose HTTP
    - Source: Choose Custom.
    - In the Source field, enter sg and choose ProxySG security group.
    - Select the ProxyServer2 instance.
    - Choose Actions menu and then choose Security > Change security groups. Remove the ProxySG2 security group.
    - In the search box, search for and choose the ProxySG security group.

- Choose Add security group, and then choose Save.

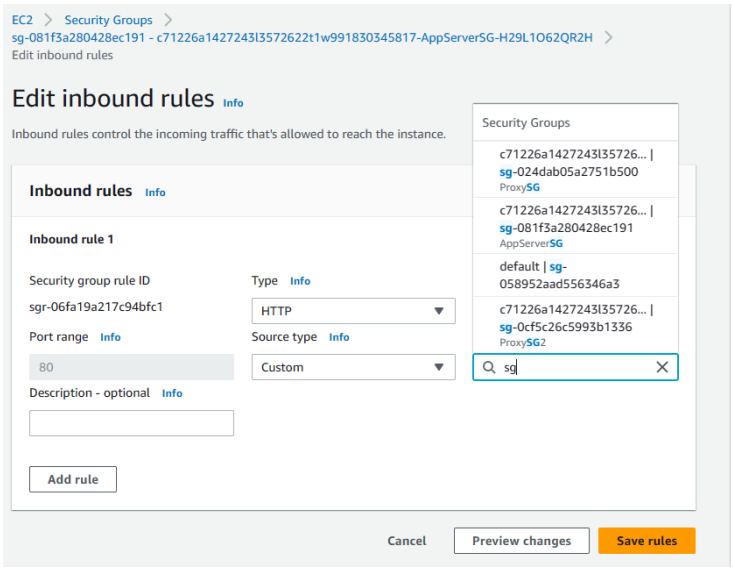
4. To test the website access from ProxyServer1 and ProxyServer2 , copy the respective Public IPv4 values and paste them in different browser tabs. The website loads.
5. The AppServerSG security group is updated to allow traffic from any instances associated with the ProxySG security group.
6. To control access to EC2 instances in a VPC using network ACLs,
    - Add a new inbound rule on the LabVPC network ACL to deny all inbound traffic on Port 80
    - In the Amazon VPC console, choose Network ACLs. Select the ACL that is associated with LabVPC.
    - Choose the Inbound Rules tab, and then choose Edit inbound rules.
    - Choose Add new rule, and configure as follows:
    - Rule number: Enter 99
    - Type: Choose HTTP
    - Allow/Deny: Choose Deny
    - Choose Save changes.

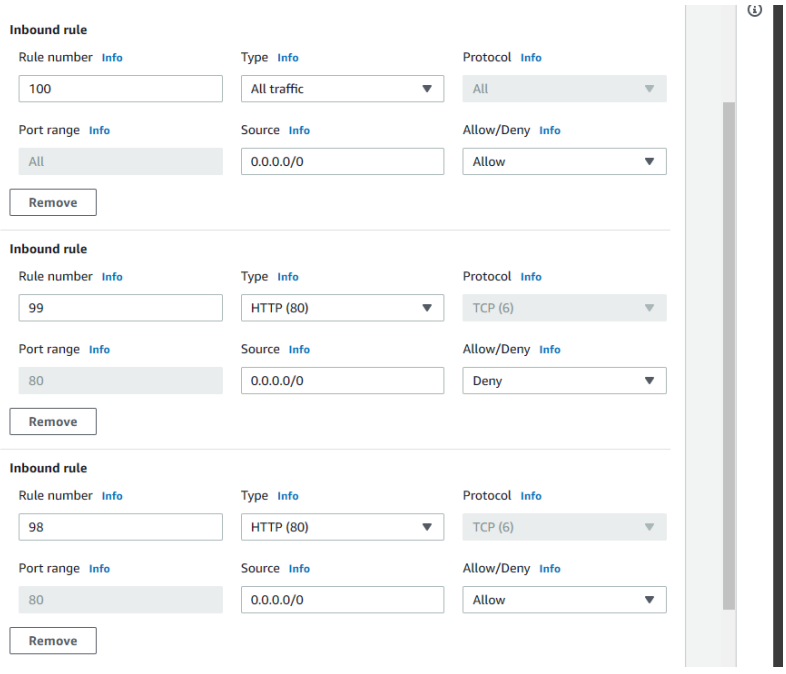
    Now when you test accessing the website from ProxyServer1, the connection times out. The ProxySG security group allows inbound port 80 traffic, but the connection fails because both the network ACL and security group would need to allow it.

    - In the LabVPC network ACL still selected, choose Edit Inbound rules.
    - Choose Add new rule and configure.
        - Rule number: Enter 98

- Type: Choose HTTP
- Allow/Deny : Choose Allow
- Choose Save changes.

Now when you test accessing the website from ProxyServer1, the website rules. This is because network ACL have a rule number. Rules are evaluated in order, starting with the lowest numbered value.





How to create an Amazon RDS instance using the CLI:

1. Connect to the CLI Host instance by using SSH. Choose and download the labsuser.pem. Open a terminal window and change directory cd to the directory where the labsuser.pem file was downloaded.

   cd ~/Downloads

   Change the permissions on the key to be read only using the command:

   chmod 400 labsuser.pem

   In the description tab of the CLI Host instance, copy the IPv4 public IP value and use it in the following command:

   ssh -i labsuser.pem ec2-user@<public-ip>

   Type yes when prompted to allow a first connection to this remote SSH server.

2. Update the AWS CLI software with the credentials.

   aws configure

   At the prompts, enter the AWS Access Key ID, AWS Secret Access Key, Default region name and the default output format.

3. In the SSH window enter:

```
aws rds create-db-instance \
--db-instance-identifier MomPopCafeDBInstance \
--engine mariadb \
--engine-version 10.2.43 \
--db-instance-class db.t2.micro \
--allocated-storage 20 \
--availability-zone <MomPopCafeInstance Availability Zone> \
--db-subnet-group-name "MomPopCafeDB Subnet Group" \
--vpc-security-group-ids <MomPopCafeDatabaseSG Group ID> \
--no-publicly-accessible \
--master-username root --master-user-password 'Re:Start!9'
```

In the command, substitute <MomPopCafeInstance Availability Zone> and <MomPopCafeDatabaseSG GroupID> .

4. Monitor the status of the database instance and wait until it shows a value of *available.* In the SSH window enter:

```
aws rds describe-db-instances \
--db-instance-identifier MomPopCafeDBInstance \
--query "DBInstances[*].[Endpoint.Address,AvailabilityZone,PreferredBackupWindow,
BackupRetentionPeriod,
DBInstanceStatus]"
```

5. Keep repeating the command until the status shows available, then record the value that is returned for Endpoint Address.

```
[ec2-user@ip-10-200-0-107 ~]$ aws ec2 describe-instances \--filters "Name=tag:Name,Values= MomPopCafeInstance" \--query "Reservations[*].Instances[*].[InstanceId,InstanceType,PublicDnsName,PublicIpAddress,Placement.AvailabilityZone,VpcId,SecurityGroups[*].GroupId]"
[
    [
        [
            "i-0355ea0a9cf6c60fd",
            "t2.small",
            "ec2-3-94-251-5.compute-1.amazonaws.com",
            "3.94.251.5",
            "us-east-1a",
            "vpc-091d0c74c5c787ae4",
            [
                "sg-0754b0aee93da57c1"
            ]
        ]
    ]
]
[ec2-user@ip-10-200-0-107 ~]$ Padmasree RameshSrinivasan
```

How to collect information about an instance including the following:

- **Instance ID,**
- **Instance type**
- **Public DNS name**
- **Public IP address**
- **Availability zone**
- **VPC id**
- **Group ID**
- **IPv4 CIDR block of the VPC**

- **Subnet ID**
- **IPv4 CIDR block of the subnet**
- **List of availability zones in the region**

1. Connect to the CLI host Instance through SSH by using the correct key pair and the Public IPv4 address.

```
aws ec2 describe-instances \
--filters "Name=tag:Name,Values= MomPopCafeInstance" \
--query "Reservations[*].Instances[*].[InstanceId,InstanceType,PublicDnsName,
PublicIpAddress,Placement.AvailabilityZone,VpcId,SecurityGroups[*].GroupId]"
```

2. After the above command runs, it returns the values for the queried attributes like Instance Id, Instance type, Public DNS name, Public IP Address, Availability zone, VPC id, and the MomPopCafeSecurityGroup GroupID.
3. Determine the IPv4 CIDR block of the MomPopCafe VPC. In the SSH window enter:

```
aws ec2 describe-vpcs --vpc-ids <MomPopCafeInstance VPC ID> \
--filters "Name=tag:Name,Values= MomPopCafe VPC" \
--query "Vpcs[*].CidrBlock"
```

4. Determine the Subnet ID and IPv4 CIDR block of MomPopCafe Public Subnet 1, which is the only subnet in the VPC. In the SSH window enter:

```
aws ec2 describe-subnets \
--filters "Name=vpc-id,Values=<MomPopCafeInstance VPC ID>" \
--query "Subnets[*].[SubnetId,CidrBlock]"
```

5. Determine the list of Availability Zones in the region. In the command, substitute <region> with the actual region name. In the SSH window enter:

```
aws ec2 describe-availability-zones \
--filters "Name=region-name,Values=<region>" \
--query "AvailabilityZones[*].ZoneName"
```

List of Attributes of the Mom and Pop Café instance:

- **Instance ID** - i-0f032b6f11d258e0d
- **Instance type** – t2.small
- **Public DNS name** – ec2-54-86-143-161.compute-1.amazonaws.com
- **Public IP address**- 54.86.143.161
- **Availability zone** – us-east-1a
- **VPC ID**- vpc-0260015fd05061996
- **Group ID of MomPopCafeSecurityGroup**- sg-0ecfa5f3491df55ec
- **IPv4 CIDR block of MomPopCafe VPC**- 10.200.0.0/20
- **Subnet ID of MomPopCafe Public Subnet 1**- subnet-059e109046517837a
- **IPv4 CIDR block of MomPopCafe Public Subnet 1**- 10.200.0.0/24
- **List of Availability zones in the Region**- us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f

How to create two subnets in a subnet group via the AWS CLI:

1. In the SSH window, enter :

```
aws ec2 create-subnet \
--vpc-id <MomPopCafeInstance VPC ID> \
--cidr-block 10.200.2.0/23 \
--availability-zone <MomPopCafeInstance Availability Zone>
```

   In the command, substitute <MomPopCafeInstance VPC ID> and <MomPopCafeInstance Availability Zone> with the values there were recorded earlier.

2. After the command completes, record the returned subnetID as MomPopCafe Private Subnet 1 ID value.
3. Create the MomPopCafeDB Private Subnet 2. It is an empty private subnet that is defined in a different Availability Zone than the MomPopCafeInstance.

   So, you must assign a CIDR address block to the subnet that is within the address range of the VPC, but does not overlap with the address range of any other subnet in the VPC. Also

choose a different availability zone for the second subnet's availability zone. In the SSH window,

```
aws ec2 create-subnet \
--vpc-id <MomPopCafeInstance VPC ID> \
--cidr-block 10.200.10.0/23 \
--availability-zone <availability-zone>
```

After the command completes, record the returned SubnetId as MomPopCafe Private Subnet 2 ID.

4. Create the MomPopCafeDB Subnet Group. A Database subnet group is a collection of subnets in a VPC. In the SSH window, enter:

```
aws rds create-db-subnet-group \
--db-subnet-group-name "MomPopCafeDB Subnet Group" \
--db-subnet-group-description "DB subnet group for Mom & Pop Cafe" \
--subnet-ids <MomPopCafe Private Subnet 1 ID> <MomPopCafe Private Subnet 2 ID> \
--tags "Key=Name,Value= MomPopCafeDatabaseSubnetGroup"
```

After this command is completed, it returns the attributes of the DB Subnet Group.

How to use the mysqldump tool to take a backup of a SQL database and restore it on another SQL instance:

1. Open an SSH session to the MomPopCafeInstance and login as ec2-user.
2. Use the mysqldump utitly to create a backup of the local mom_pop_db database. This utility program is part of the MySQL database product. In the SSH window, enter:

```
mysqldump --user=root --password='Re:Start!9' \
--databases mom_pop_db --add-drop-database > mompopdb-backup.sql
```

This command generates SQL statements in a file named mompopdb-backup.sql, which can be run to reproduce the schema and data of the original mom_pop_db database.

3. Review the contents of the backup file. Open the mompopdb-backup.sql file in a text editor. In the SSH window, enter:

   less mompopdb-backup.sql

4. Restore the back up to the Amazon RDS database by using the mysql command and specify the endpoint address of the RDS instance. In the SSH window,

```
mysql --user=root --password='Re:Start!9' \
--host=<RDS Instance Database Endpoint Address> \
< mompopdb-backup.sql
```

   This command creates a mysql connection to the RDS instance and runs the SQL statements in the mompopdb-backup.sql file.

5. Finally, verify that the mom_pop_db was successfully created and populated in the Amazon RDS instance. In the SSH window enter:

```
mysql --user=root --password='Re:Start!9' \
--host=<RDS Instance Database Endpoint Address> \
mom_pop_db
```

6. Next, enter the SQL statement to retrieve the data in the product table:

   *select * from product;*

   The query should return 9 rows from the table.

```
MariaDB [mom_pop_db]> select * from product;
+----+--------------------------+--------------------------------------------------------------------+-------+---------------+--------------------------------
-------+
| id | product_name             | description                                                        | price | product_group | image_url
      |
+----+--------------------------+--------------------------------------------------------------------+-------+---------------+--------------------------------
-------+
|  1 | Croissant                | Fresh, buttery and fluffy... Simply delicious!                     |  1.50 |             1 | images/Croissants.jpg
      |
|  2 | Donut                    | We have more than half-a-dozen flavors!                            |  1.00 |             1 | images/Donuts.jpg
      |
|  3 | Chocolate Chip Cookie    | Made with Swiss chocolate with a touch of Madagascar vanilla       |  2.50 |             1 | images/Chocolate-Chip-Cookies.jpg
      |
|  4 | Muffin                   | Banana bread, blueberry, cranberry or apple                        |  3.00 |             1 | images/Muffins.jpg
      |
|  5 | Strawberry Blueberry Tart | Bursting with the taste and aroma of fresh fruit                   |  3.50 |             1 | images/Strawberry-&-Blueberry-Tar
ts.jpg |
|  6 | Strawberry Tart          | Made with fresh ripe strawberries and a delicious whipped cream    |  3.50 |             1 | images/Strawberry-Tarts.jpg
      |
|  7 | Coffee                   | Freshly-ground black or blended Columbian coffee                   |  3.00 |             2 | images/Coffee.jpg
      |
|  8 | Hot Chocolate            | Rich and creamy, and made with real chocolate                      |  3.00 |             2 | images/Cup-of-Hot-Chocolate.jpg
      |
|  9 | Latte                    | Offered hot or cold and in various delicious flavors               |  3.50 |             2 | images/Latte.jpg
      |
+----+--------------------------+--------------------------------------------------------------------+-------+---------------+--------------------------------
-------+
9 rows in set (0.00 sec)

MariaDB [mom_pop_db]> Padmasree_RameshSrinivasan
```

How to take a snapshot of an EBS volume:

1. On the AWS Management console, on the Services menu, click S3. Click Create Bucket. In the Create Bucket dialog box, type Bucket name as s3-bucket-name. Leave the Region as default. Click Create.
2. On the Services menu, click EC2. Click Instances. Select the Processor. Click on Actions, then Security, followed by Modify IAM role. Select the S3BucketAccess role under IAM role. Click Apply. This IAM Role attached to the Processor Host gives it permission to interact with your Amazon S3 bucket.
3. Connect to the Command Host through SSH using the following command.

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

4. To get a full description of the Processor instance,

```
aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor'
```

To narrow down the results of the previous command further,

```
aws ec2 describe-instances
--filter 'Name=tag:Name,Values=Processor'
--query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'
```

This command will return the volume ID of the only volume (root volume) attached to the Processor Instance.

5. Before taking a snapshot, you will shut down the Processor Instance for that we need to obtain the instance Id.

```
aws ec2 describe-instances
--filters 'Name=tag:Name,Values=Processor'
--query 'Reservations[0].Instances[0].InstanceId'
```

6. To shut down the Processor instance, copy the following command, replace INSTANCE-ID with the instance ID obtained from the previous command.

```
aws ec2 stop-instances --instance-ids INSTANCE-ID
```

7. To create your first snapshot of the root volume of your Processor instance, copy the following command , replace VOLUME-ID with your volume-id

```
aws ec2 create-snapshot --volume-id VOLUME-ID
```

This command will return the SnapshotId value that uniquely identifies the new snapshot.

8. To check the status of your snapshot, copy the following command, replace SNAPSHOT-ID with your snapshot-id,

```
aws ec2 wait snapshot-completed --snapshot-id SNAPSHOT-ID
```

9. To restart the Processor instance, copy the following command, replace the INSTANCE-ID to your instance id.

```
aws ec2 start-instances --instance-ids INSTANCE-ID
```

```
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-03fb0901818953dd8",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
[ec2-user@ip-10-5-0-59 ~]$ aws ec2 wait instance-running --instance-id i-03fb0901818953dd8
[ec2-user@ip-10-5-0-59 ~]$ Padmasree_RameshSrinivasan
```

How to synchronize files from your local computer to an S3 bucket then enable versioning on it using the command line (aws s3api and aws s3):

1. Run this command on the EC2 instance to download a sample set of files:

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/
CUR-TF-200-RESOPS/lab5vocareum/files.zip
```

2. Unzip these files using the command.

```
unzip files.zip
```

3. Before synchronizing content with your Amazon S3 bucket, you will need to enable versioning to your bucket.

```
aws s3api put-bucket-versioning
--bucket S3-BUCKET-NAME --versioning-configuration Status=Enabled
```

   Replace S3-BUCKET-NAME with your bucket name.

4. To synchronize the contents of the files with your Amazon S3 bucket, copy the following command.

```
aws s3 sync files s3://S3-BUCKET-NAME/files/
```

   This command should confirm that it has copied each of the 3 files to your Amazon S3 bucket.

5. To confirm the state of your files, use the following command.

```
aws s3 ls s3://S3-BUCKET-NAME/files/
```

6. To delete one of the files on the local drive,

```
rm files/file1.txt
```

7. To delete the same file from the server, use the –delete option to the aws s3 sync command.

```
aws s3 sync files s3://S3-BUCKET-NAME/files/ --delete
```

8. Verify that the file was deleted remotely on the server.

```
aws s3 ls s3://S3-BUCKET-NAME/files/
```

9. Now try to recover the old version of file1.txt. To view a list of past versions of this file, use the aws s3api list-object-versions command:

```
aws s3api list-object-versions --bucket S3-BUCKET-NAME --prefix files/file1.txt
```

   The output will contain a DeleteMarkers and a Versions block. You should have only a single Versions entry. Find the VersionId field and copy its value.

10. There is no direct command to restore an older version of an Amazon S3 object to its own bucket, you will need to re-download the old version and then sync again to Amazon S3. To download the previous version of file1.txt, copy the following command.

```
aws s3api get-object
--bucket S3-BUCKET-NAME --key files/file1.txt --version-id VERSION-ID files/file1.txt
```

11. To verify that the file has been restored locally, use the following command.

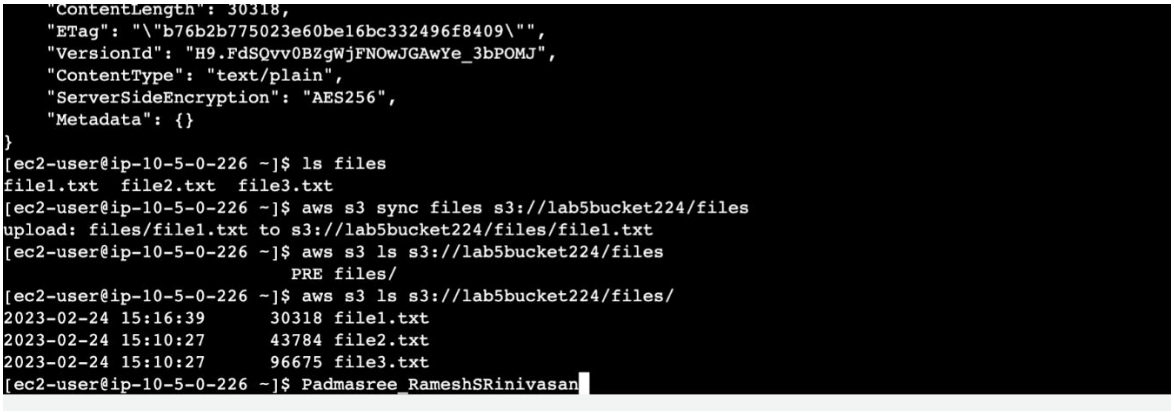
```
ls files
```

12. To re-sync the contents of the files to Amazon S3, copy the following command.

```
aws s3 sync files s3://S3-BUCKET-NAME/files/
```

13. To verify that a new version of the file1.txt has been pushed to Amazon S3,

```
aws s3 ls s3://S3-BUCKET-NAME/files/
```



How to create a S3 bucket via the CLI:

1. Open a SSH session to the CLI Host instance by using the labsuser.pem file. Copy the Public IPv4 address and run the following command:

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

2. To configure the region in which the CLI Host is running.

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
```

Update the AWS CLI software with the credentials.

```
aws configure
```

At the prompts, enter the following information:

- AWS Access Key ID
- AWS Secret Access Key
- Default region name: Type in the name of the region where your EC2 instances are running, which you discovered in the previous command.
- Default output format: json

3. Create the <mompopcafe-xxxnnn> S3 bucket. For xxxnnn, substitute a random number so that the S3 bucket name will be unique across all existing bucket names in Amazon S3.

```
aws s3 mb s3://<mompopcafe-xxxnnn> --region <region>
```

When the *make bucket* (mb) command completes successfully, it returns the name of the bucket.

4. Load some images in the S3 bucket under the /images prefix.

```
aws s3 sync ~/initial-images/ s3://<mompopcafe-xxxnnn>/images
```

As the *synchronize(sync)* command runs, you will see the names of the image files being uploaded.

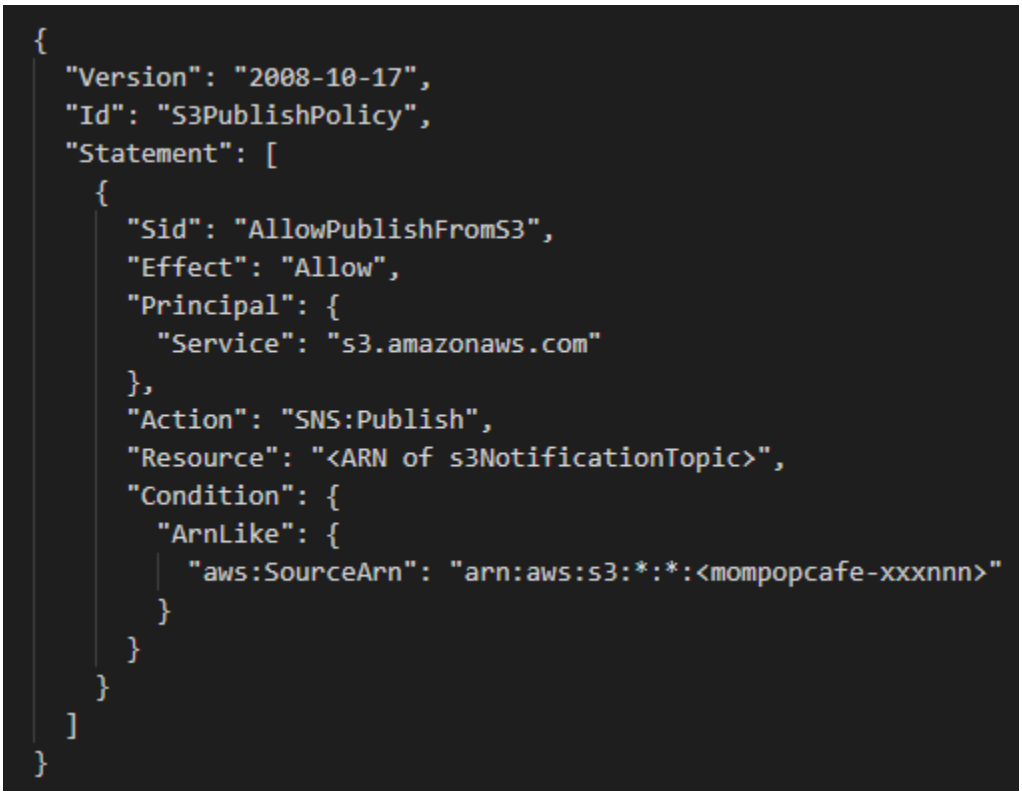
5. List the bucket contents using the s3 ls command. Choose to display the list in human-readable form with summary totals for the number of objects and their total size at the bottom.

```
aws s3 ls s3://<mompopcafe-xxxnnn>/images/ --human-readable --summarize
```

```
[ec2-user@ip-10-200-0-182 ~]$ aws s3 mb s3://mompopcafe-prs225 --region us-east-1
make_bucket: mompopcafe-prs225
[ec2-user@ip-10-200-0-182 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafeprs223/images
[ec2-user@ip-10-200-0-182 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafe-prs225/images
upload: initial-images/Cup-of-Hot-Chocolate.jpg to s3://mompopcafe-prs225/images/Cup-of-Hot-Chocolate.jpg
upload: initial-images/Strawberry-Tarts.jpg to s3://mompopcafe-prs225/images/Strawberry-Tarts.jpg
upload: initial-images/Donuts.jpg to s3://mompopcafe-prs225/images/Donuts.jpg
[ec2-user@ip-10-200-0-182 ~]$ aws s3 ls s3://mompopcafe-prs225/images/ --human-readable --summarize
2023-02-24 03:21:27  308.7 KiB Cup-of-Hot-Chocolate.jpg
2023-02-24 03:21:27  371.8 KiB Donuts.jpg
2023-02-24 03:21:27  468.0 KiB Strawberry-Tarts.jpg

Total Objects: 3
   Total Size: 1.1 MiB
[ec2-user@ip-10-200-0-182 ~]$
```

How to add an event notification to an S3 bucket:

1. In the AWS Management Console, search for and choose Simple Notification Service. In the left navigation pane, select Topics. Choose Create topic. Choose Standard. In the Name box, enter **s3NotificationTopic.** Choose Edit. Expand the Access Policy-Optional section. Replace the contents of the JSON editor with the following policy:



```
{
  "Version": "2008-10-17",
  "Id": "S3PublishPolicy",
  "Statement": [
    {
      "Sid": "AllowPublishFromS3",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "<ARN of s3NotificationTopic>",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:<mompopcafe-xxxnnn>"
        }
      }
    }
  ]
}
```

   In the JSON object, substitute <ARN of s3NotificationTopic> with the value of the topic ARN, and <mompopcafe-xxxnnn> with your unique S3 bucket name.

2. The intent of the above policy is – it grants the mompopcafe S3 share bucket the permission to publish messages to the s3NotificationTopic. Choose **Save changes.** Subscribe Pop to the topic as the mompopuser who will receive the event notifications from the S3 share bucket.

3. Choose **Create Subscription.** In the **topicARN** box, **s3NotificationTopic** already appears. In the **Protocol** menu, select **Email.** In the **Endpoint** box, enter an email address that you can access.
4. Choose **Create subscription.** A message is displayed confirming that the subscription was created successfully.
5. Check the inbox for the email address that you provided. You should see an email message with the subject *AWS Notification – Subscription confirmation.* Open the email message and choose **Confirm subscription.** A new browser tab opens and displays a page with the message *Subscription confirmed!*
6. Now to create an event notification configuration file that identifies the events that Amazon S3 will publish and the topic destination where AmazonS3 will send the event notifications. In the SSH window for the CLI Host instance, edit a new file named s3EventNotification.json by entering:

   vi s3EventNotification.json

   In the editor, change to *insert* mode by entering *i.*

   Customize the following JSON configuration by substituting the <ARN of s3NotificationTopic> with the value of the topicARN created earlier.

The above policy requests that Amazon S3 publish an event notification to the s3NotificationTopic whenever an ObjectCreated or ObjectRemoved event is performed on objects inside an S3 resource with a prefix of images/.

Press ESC to exit *insert* mode.

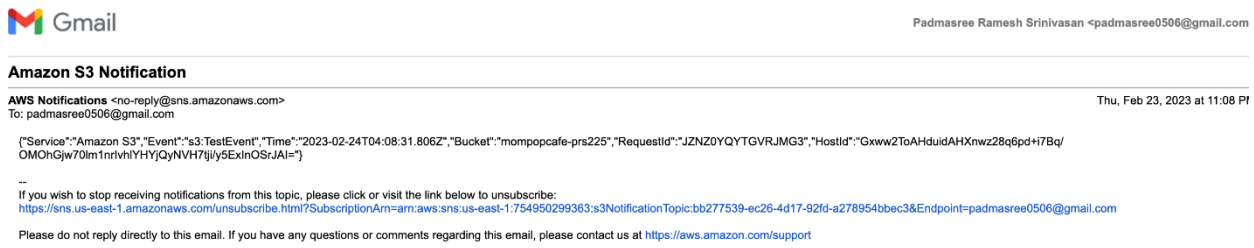
To save the file and exit the editor, enter *:wq*.

7. Associate the event configuration file with the S3 share bucket. In the SSH window for the CLI Host instance, enter:

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

Substitute the <mompopcafe-xxxnnn> with your unique S3 bucket name.

8. Wait for a few moments and then check the inbox for the email address that you used to subscribe to the topic. You should see an email message with the subject Amazon S3 Notification. Notice that the value of the "Event" key is "s3:TestEvent". This notification was sent by Amazon S3 as a test of the event notifications configuration that was set up.



How to encrypt the root volume of an existing EC2 instance:

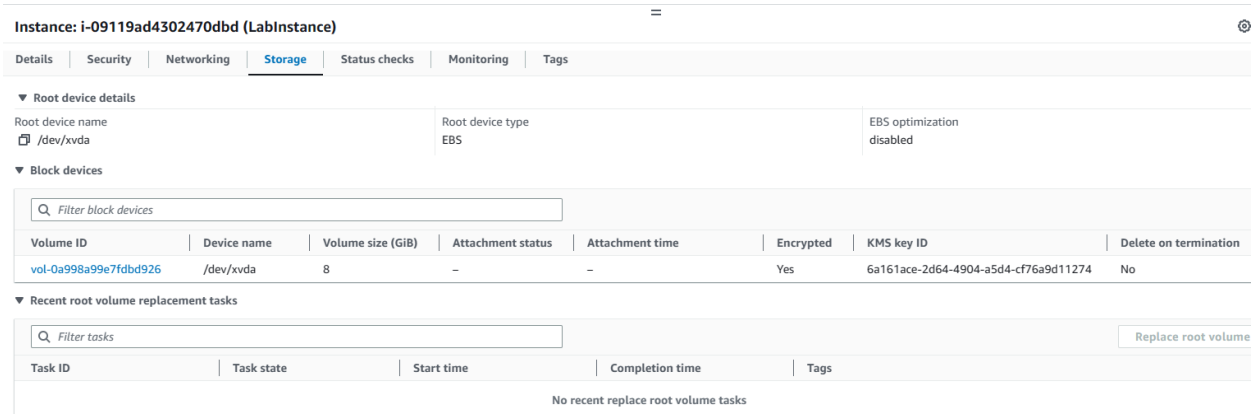
1. In the search box to the right of **Services**, search for and choose **EC2** to open the Amazon EC2 console. In the navigation pane, choose **Instances**. Choose the link for **LabInstance** instance ID. Choose the **Storage** tab. In the **Block Devices** section, notice that the volume

that is attached indicates it is *not encrypted.* This is the root volume, which contains the guest OS installation.

2. Stop the instance.
    - In the breadcrumbs at the top of the page, choose Instances.
    - Select LabInstance and choose Instance State > **Stop instance.**
    - To confirm the action, choose **Stop.**
3. Create a snapshot of the root EBS volume of the existing EC2 instance.
    - Choose the **Storage** tab.
    - In the **Block devices** section, choose the link for the VolumeID.
    - Choose the link for the volume ID again.
    - Note the availability zone where the volume exists.
    - Choose Actions > **Create Snapshot.**
    - Choose Add tag and add a tag with the following information.
        - **Key**: Enter *Name*
        - **Value:** Enter *Unencrypted Root Volume*
    - Choose **Create Snapshot**
4. Label the volumes.
    - In the navigation pane, under Elastic Block Store, choose **Volumes**.
    - Notice that two volumes are now listed.
    - For the volume with a Volume state of In-use, change the volume name:
        - Hover on the Name field, choose the pencil and paper icon.
        - In the Edit Name box, enter *Old unencrypted root volume*.
        - Choose **Save.**

- Follow the same steps to change the name of the volume with a Volume state of Available to *New unencrypted root volume.*

5. Swap the root volume that the EC2 instance uses.
    - Select Old unencrypted root volume, and then choose **Actions > Detach volume.**
    - To confirm, choose **Detach.**
    - Select New encrypted root volume, and choose **Actions > Attach volume** and configure the following:
        - **Instance**: Choose (LabInstance) (stopped)
        - **Device** Name: Enter /dev/xvda
    - Choose **Attach volume**.
6. Notice that the root volume is now encrypted.
    - Return to the **Instances** screen and select LabInstance.
    - Choose the **Storage** tab, and notice that the attached volume is now encrypted and has **an AWS KMS key ID.**



7. Return to the Amazon EC2 console and start LabInstance again. The instance successfully starts now.

How to detect drift in a CloudFormation template:

1. To start drift detection on your stack, run the following command:

```
aws cloudformation detect-stack-drift --stack-name myStack
```

This command should return a StackDriftDetectionId.

2. Monitor ethe status of the drift detection by running the following command:

```
aws cloudformation describe-stack-drift-detection-status \
--stack-drift-detection-id driftId
```

Notice that the output shows **"StackDriftStatus"."DRIFTED"**

3. Finally, describe the resources that drifted by running the following describe-stack-resource-drifts command:

```
aws cloudformation describe-stack-resource-drifts \
--stack-name myStack
```

4. Run a describe-stack-resources command with a query parameter that will return only the resource type, resource status, and drift status.

```
aws cloudformation describe-stack-resources \
--stack-name myStack \
--query 'StackResources[*].[ResourceType,ResourceStatus,DriftInformation.StackResourceDriftStatus]' \
--output table
```

The output is easier to read because of the query parameter, which is written in JMESPath. On this stack, all checked resource have a status of IN_SYNC, except for the resource which has a status **MODIFIED.**

5. Retrieve the specific details of the drift for the resource that has a StackResourceDriftStatus of MODIFIED.

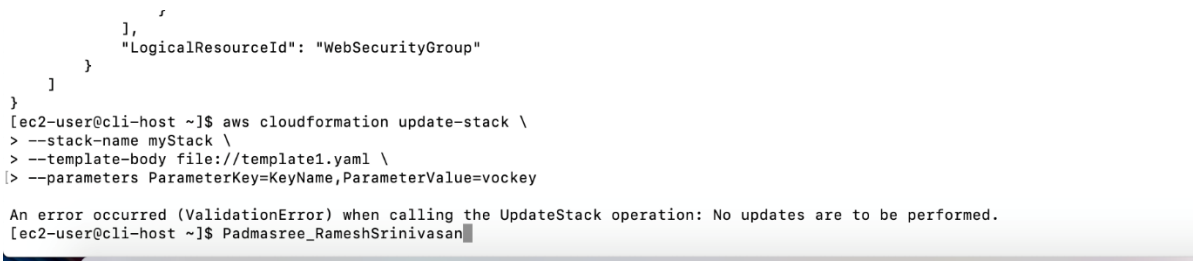
```
aws cloudformation describe-stack-resource-drifts \
--stack-name myStack \
--stack-resource-drift-status-filters MODIFIED
```

Notice that the PropertyDifferences section of the output that shows specific details about the resources that were MODIFIED.

6. Try updating the stack:

```
aws cloudformation update-stack \
--stack-name myStack \
--template-body file://template1.yaml \
--parameters ParameterKey=KeyName,ParameterValue=vockey
```

The output should indicate that an error occurred. This is expected. The update-stack command will not automatically resolve drift, though drift has occurred.



How to install the CloudWatch agent:

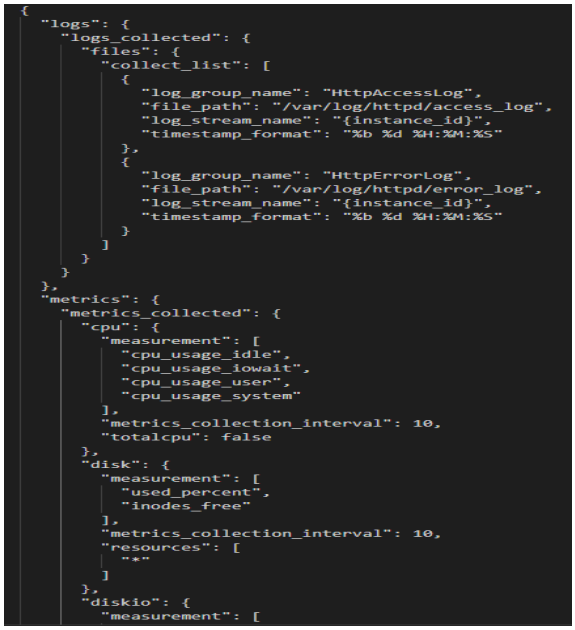
1. The CloudWatch agent can be used to collect metrics from Amazon EC2 instances and on-premises servers including System level metrics from Amazon EC2 instances such as CPU allocation, free disk space and memory utilization. System level metrics from on-premises servers, System and application logs from both Linux and Windows servers and custom metrics.

2. In the AWS Management console, in the search box next to Services, search for and select the Systems Manager service to open the Systems Manager console. In the left navigation pane, choose Run command under Node Management.
3. Choose Run a command. Select the radio button next to AWS-ConfigureAWSPackage.
4. Go down to the Command parameters section and configure
    - Action: Install
    - Name: AmazonCloudWatchAgent
    - Version: latest
5. In the Targets section, select Choose instances manually and then select ☒Web Server. This configuration will install the CloudWatch agent on the Web Server. At the bottom of the page, choose Run.
6. Wait for the Overall status to change to Success. You can occasionally choose Refresh button towards the top of the page to update the status. You can view the output from the job to confirm that it ran successfully.
7. Under Targets and outputs, choose the instance-id displayed under Instance ID. Expand ➔Step 2- Command description and status. You should see the message : *Successfully installed arn:aws:ssm:::package/AmazonCloudWatchAgent.* To configure the CloudWatch agent to collect the web server logs and general system metrics, we can store the configuration file in the AWS Systems Manager Parameter Store, which can then be fetched by the CloudWatch agent.
8. In the left navigation pane, choose Parameter Store. Choose Create Parameter and then configure:

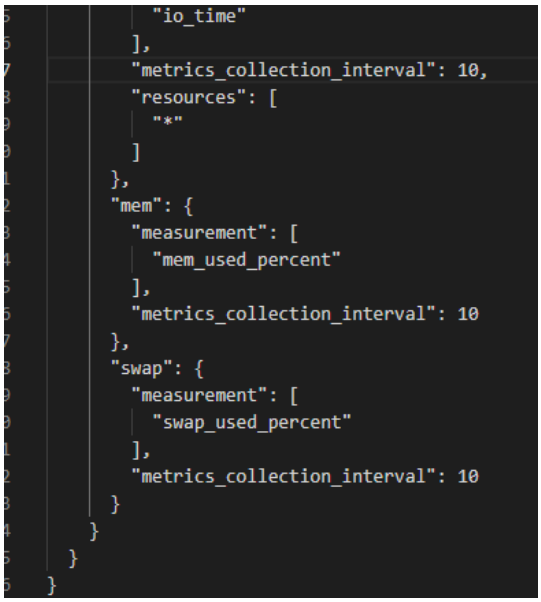
    Name : Monitor-Web-Server

Description: Collect web logs and system metrics

Value: Paste the code below



```
{
  "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
      }
    }
  }
}
```

The above configuration defines the following items to be monitored:

- **Logs:** Two web server log files to be collected and sent to CloudWatch Logs.
- **Metrics**: CPU, disk, and memory metrics to send to Amazon CloudWatch Metrics.

9. Choose **Create parameter.**

   This parameter will be referenced when starting the CloudWatch Agent.

10. To start the CloudWatch Agent on the Web server, in the left navigation pane, choose Run command. Choose **Run command.** Choose  then:

    - Document name prefix
    - Equals
    - AmazonCloudWatch-ManageAgent
    - Press Enter

    Choose AmazonCloudWatch-ManageAgent(choose the name itself). A new web browser tab will open, showing the definition of the command. Choose the Content tab and scroll to the bottom to see the actual script that will run on the target instance.

11. Return to the Run a command tab you were using earlier. Verify that you have selected the radio button next to AmazonCloudWatch-ManageAgent.

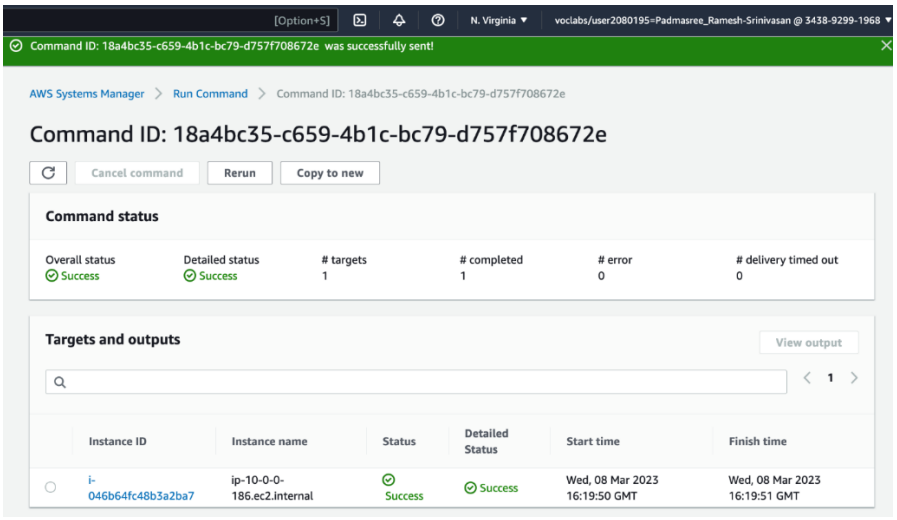
12. In the Command parameters section, configure:

    - **Action**: *configure*
    - **Mode**: ec2
    - **Optional configuration source**: *ssm*
    - **Optional configuration location:** *Monitor-Web-Server*
    - **Optional Restart**: yes

    This configures the Agent to use the configuration you previously stored in the Parameter store.

13. In the Targets panel below, select **Choose instances manually.** In the **Instances** section, select ☑ Web-Server. Choose **Run**.

14. Wait for the Overall status to change to *Success.* You can occasionally choose Refresh towards the top of the page to update the status. The CloudWatch agent is now running on the instance, sending log and metric data to Amazon CloudWatch.



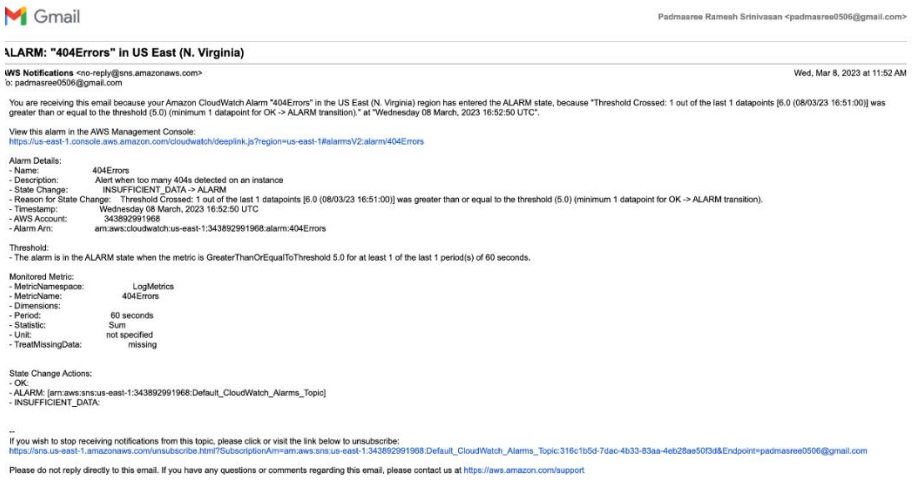
How to create a CloudWatch Events/CloudWatch EventBridge notification rule:

1. Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources. In Amazon CloudWatch, in the left navigation pane, expand Events and choose Rules.
    - At the top of the page, below the CloudWatch Events is now EventBridge notification, choose Go to Amazon EventBridge.

    Choose **Create rule**.
2. For **Name** enter *Instance_Stopped_Terminated*. Choose Next.
3. In the Event pattern section near the bottom of the page, configure the following settings:
    - **Event source:** From the drop down list, choose Aws services
    - **AWS Service:** From the drop down list, choose EC2.

- **Event type:** From the drop down list, choose EC2 Instance State-change notification.
- Select Specific state(s)
- From the drop down list, choose *stopped and terminated.*
- Choose Next.

4. In the Target 1 section, configure the following settings:
   - From the **Select a target** drop down list, choose *SNS topic.*
   - From the **Topic** drop-down list, choose *Default_CloudWatch_Alarms_Topic*
5. Choose Next. On the Review and create page, choose Create rule.
6. On the Services menu, choose EC2. In the left navigation pane, choose Instances.
7. Select Web Server. Choose Instance state, then Stop Instance, then Stop.
8. The Web Server instance will enter the stopping state. After a minute, it will enter the stopped state. After a minute it will enter the Stopped state. Now you should receive an Email or SMS based on your preferred way of notification in the SNS topic. The message is formatted in JSON.


Gmail

Padmasree Ramesh Srinivasan <padmasree0506@gmail.com>

ALARM: "404Errors" in US East (N. Virginia)

WS Notifications <no-reply@sns.amazonaws.com> Wed, Mar 8, 2023 at 11:52 AM
o: padmasree0506@gmail.com

You are receiving this email because your Amazon CloudWatch Alarm "404Errors" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [6.0 (08/03/23 16:51:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Wednesday 08 March, 2023 16:52:50 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/404Errors

Alarm Details:
- Name: 404Errors
- Description: Alert when too many 404s detected on an instance
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [6.0 (08/03/23 16:51:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Wednesday 08 March, 2023 16:52:50 UTC
- AWS Account: 343892991968
- Alarm Arn: arn:aws:cloudwatch:us-east-1:343892991968:alarm:404Errors

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 5.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace: LogMetrics
- MetricName: 404Errors
- Dimensions:
- Period: 60 seconds
- Statistic: Sum
- Unit: not specified
- TreatMissingData: missing

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:us-east-1:343892991968:Default_CloudWatch_Alarms_Topic]
- INSUFFICIENT_DATA:

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:343892991968:Default_CloudWatch_Alarms_Topic:316c1b5d-7dac-4b33-83aa-4eb28ae50f3d&Endpoint=padmasree0506@gmail.com

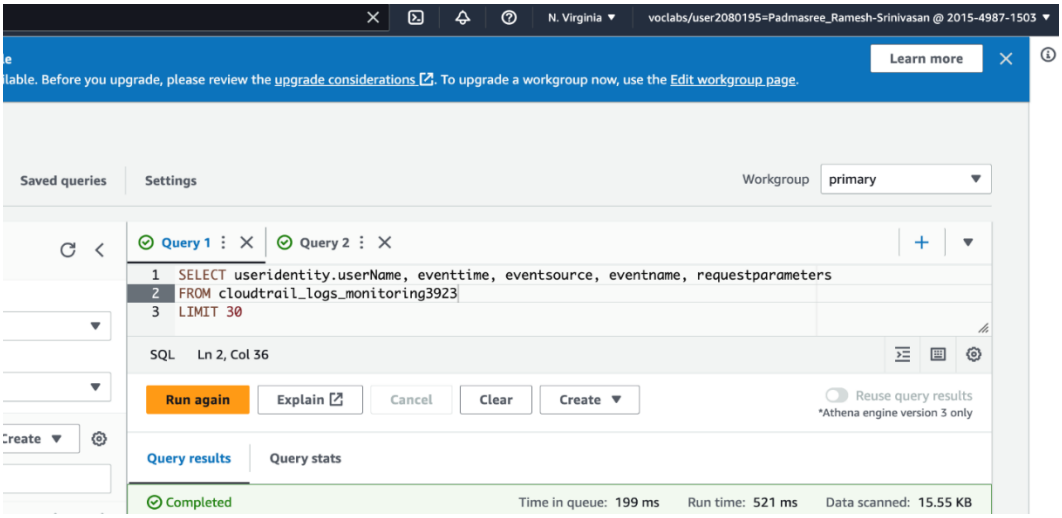
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

How to create an Amazon Athena table:

1. In the AWS Management Console, in the search box next to Services, search for and select the **CloudTrail** service to open the CloudTrail console.
2. In the navigation pane, choose **Event History**. Notice that CloudTrail provides this event history interface where you can apply filters and conduct a basic search based on parameters, such as Event name or Resource type.
3. From the Event History page, choose **Create Athena table.**
    - Storage location: Choose the **monitoring####** Amazon S3 bucket where you configured CloudTrail to store log files.
4. Choose **Create table**. The table is created with a default name that includes the name of the Amazon S3 bucket.
5. In the left panel of the Athena Query Editor, you should see the clooudtrail_logs_monitoring#### table. Choose the plus icon next to table name to reveal the column names. Each standard child element that exists in the CloudTrail log record in JSON format has a corresponding column name in this database.
6. To set up a query results location and then to query the data that is available in the logs, choose View settings button that appears above the query panel, then choose Manage. Choose Browse S3, select your monitoring#### bucket and then select Choose.
7. In the location of query result box, add /results/ where monitoring#### is the name of the bucket you created earlier. Choose Save and return to the Editor tab. Paste the SQL query in the New query 1 panel and choose Run.

```
SELECT *
FROM cloudtrail_logs_monitoring####
LIMIT 5
```

This query will return 5 rows of data.



How to manually review access logs to find anomalous user activity:

1. Run a query to query the results that is available in the logs using Amazon Athena.

```
SELECT useridentity.userName, eventtime, eventsource, eventname, requestparameters
FROM cloudtrail_logs_monitoring####
LIMIT 30
```

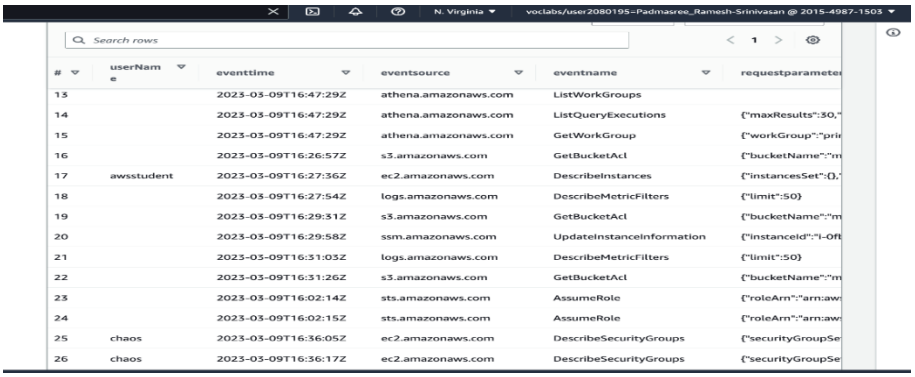
The useridentity column has a lot of information that make it more difficult to read. So just returning the username for that column will simplify the result. We can also use WHERE clauses to refine the SQL query.

2. To find out who made the security group change, type in the following query.

```
SELECT DISTINCT useridentity.userName, eventName, eventSource
FROM cloudtrail_logs_monitoring####
WHERE from_iso8601_timestamp(eventtime) > date_add('day', -1, now())
ORDER BY eventSource;
```

The above query returns a list of all users who were active in the account in the past day, and the distinct actions they have taken.

3. The name of the AWS user who created the security hole in the Security group, the exact time that they hacked the security group, the IP address from which they hacked it and the method they used to perform- console or programmatic access can be identified using the information from the result set.

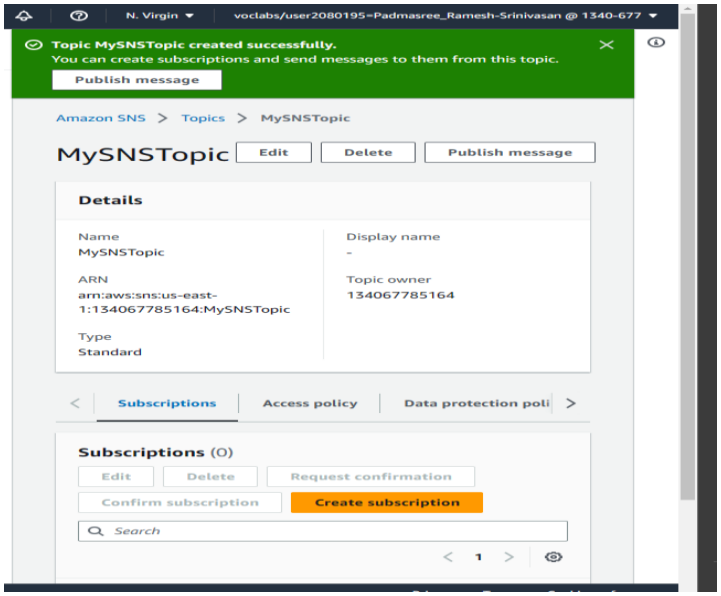


How to create a SNS topic:

1. Amazon SNS is a fully managed messaging service for both application to application(A2P) communication. The A2P functionality provides the ability to send messages to users at scale through SMS, mobile push and email. In the search box to the right of Services, search for and choose **Simple Notification Service** to open the Amazon SNS console.

- To open the navigation pane, choose the menu icon in the upper-left corner.
- Choose Create topic, and configure the following:
  - Type: Choose Standard
  - Name: Enter MySNSTopic
  - Expand the Access policy -optional section
  - Define who can publish messages to the topic: Choose Everyone

- At the bottom of the page, choose Create topic.



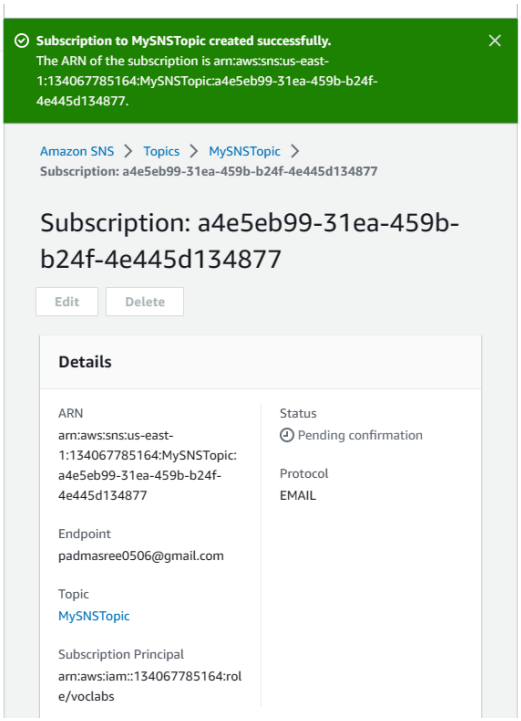
How to subscribe to a SNS Topic

To create an email subscription to the SNS topic, Choose Create subscription, and configure the following:

- Topic ARN: Notice that the ARN of the topic you just created is already filled in.
- Protocol: Choose Email.
- Endpoint: Enter an email address where you can receive emails.
- Scroll to the bottom of the page, and choose Create Subscription.

Check your email and confirm the subscription.

- Check your email for a message from AWS Notifications
- In the email body, choose the Confirm subscription link.
- A webpage opens and displays a message that the subscription was successfully confirmed.



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

1. Create a CloudWatch metric filter:
    - In the search box to the right of Services, search for and choose CloudWatch to open the CloudWatch console.
    - In the navigation pane, expand Logs and then choose Log Groups.
    - Select the check box for CloudTrailLogGroup.
    - Choose Actions > Create metric filter, and then configure the following:
        - Filter pattern: Copy and paste the following code:

```
{ ($.eventName = ConsoleLogin) && ($.errorMessage = "Failed authentication") }
```

        - Choose Next.
        - Filter Name: Enter ConsoleLoginErrors
        - Metric namespace: Enter CloudTrailMetrics

        - Metric name: Enter ConsoleLoginFailureCount
        - Metric value: Enter 1.
    - At the bottom of the page, choose Next.
    - Choose Create metric filter.

2. Create a CloudWatch alarm based on the metric filter.

    - On the Metric filters tab, select the check box to the right of the ConsoleLoginErrors metric filter that you just created.
    - Choose Create alarm. A new browser tab opens.
    - On the Specify metric and conditions page, in the Conditions section, configuring the following alarm details:
        - Whenever ConsoleLoginFailureCount is: Choose Greater/Equal.
        - Than: Enter 3.

          This alarm will be involved whenever the sum of the ConsoleLoginFailureCount metric that you defined is greater than or equal to 3 within any 5 minute period.
        - Choose Next.
    - On the Configure actions page, configure the following:
        - Select an SNS topic: Choose Select an existing topic.
        - Send a notification to.: Choose MySNSTopic.
        - Choose Next.
    - On the Add name and description page, configure the following:
        - Alarm name: Enter FailedLogins
        - Choose Next.

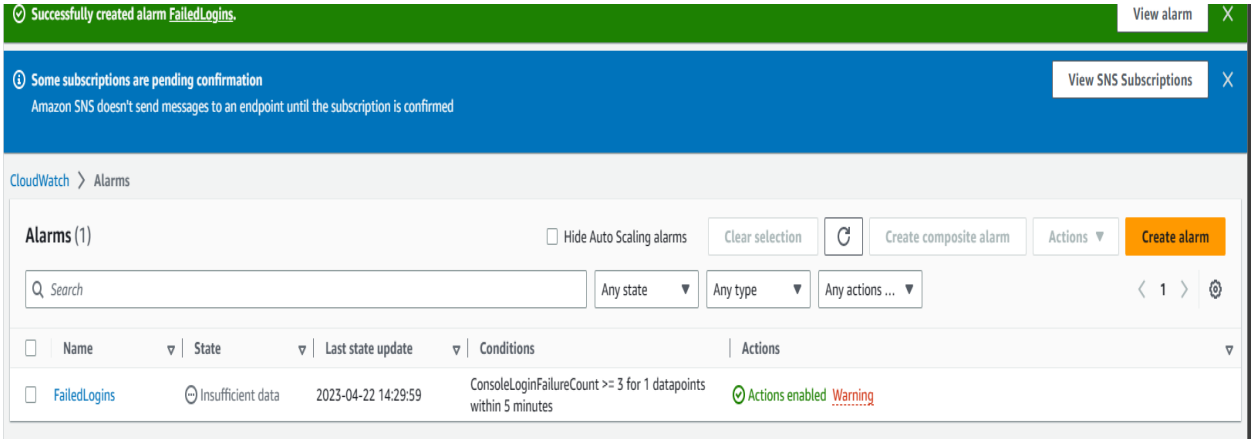
- Scroll to the bottom of the page and choose Create Alarm.

3. Test the CloudWatch alarm by attempting to log in to the console with incorrect credential at least three times.
    - In the search box to the right of Services, search for and choose IAM to open the IAM console.
    - In the navigation pane, choose Users.
    - Choose the link for the test username.
    - Choose the Security Credentials tab, and then copy the Console sign-in link.
    - Paste the copied link into a new browser tab to load the console sign-in page.
    - Enter credentials, including an incorrect password, and attempt to sign in. Repeat this at least three times.
        - IAM username: Enter test
        - Password: test
        - Choose sign in.

        Each time that you attempt to log in, you will see a message indicating that your authentication information is incorrect.
4. Re-establish your access to the AWS account. Close all browser tabs where you have the AWS Management Console open and log in again using the correct credentials.
5. Check the alarm status and details in the CloudWatch console.
    - In the navigation pane, expand Alarms, and then choose All alarms.

      The State for the FailedLogins alarm should be In alarm.

6. Check the inbox of the email address that you subscribed to the SNS Topic. You should have received a message about multiple failed login attempts.



How to use the prebuilt Stopinator script to turn off instances with the tag value of your name:

To use a prebuilt script to stop and start a set of instances tagged as your full name, first SSH into the Command Host instance using the correct keypair. Then, on the Command Host instance, cd into the directory aws-tools in the home directory.

cd aws-tools

1. Open the stopinator.php and examine its contents:

nano stopinator.php

The stopinator.php script is a simple script that uses the AWS SDK for PHP to stop and restart instances based on a set of tags. The script will look in every AWS region for instances that match the specified tags. The script takes the following arguments:

- **-t:** A set of tags in the following format: *name=value;name=value.* The script converts these tags into the format expected by the AWS PHP call

EC2::DescribeInstance(). If this optional parameter is absent, the script will identify and shut down all running Amazon EC2 instances in the account.

- **-s:** A Boolean parameter, no arguments are required. When this parameter is present, instances identified by -t are started instead of stopped.

2. Exit your nano editor.
3. From the Linux shell, run the stopinator.php script:

```
./stopinator.php -t"Project=Padmasree;Environment=development"
```

4. On the Services menu, choose EC2. In the navigation pane, choose Instances. Verify that the instances are stopping or have already been stopped.
5. Return to the SSH session for Command Host, and from the Linux prompt, restart your instances with the following command.

```
./stopinator.php -t"Project=Padmasree;Environment=development" -s
```

```
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 110

        No instances to start in Array
Region is us-east-1
        Identified instance i-00a4299464cc8c4fc
        Identified instance i-0cdff41d59b801761
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 105

        Starting identified instances in Array...
Region is us-east-2
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 110

        No instances to start in Array
Region is us-west-1
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 110

        No instances to start in Array
Region is us-west-2
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 110

        No instances to start in Array
[ec2-user@ip-10-5-0-236 aws-tools]$ Padmasree_RameshSRinivasan
```

How to resize an EC2 instance using the AWS CLI:

1. Connect to the Mom & Pop instance by using SSH using the labsuser.pem keypair and the following command:

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

   Type yes when prompted to allow connection to this remote SSH server.

2. Configure the AWS CLI commands on the instance, discover the region in which the CLI Host instance is running:

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
```

3. Update the AWS CLI software with the credentials:

   aws configure

   At the prompts, enter the AWS Access Key ID, AWS Secret Access Key, Default region name and the default output format. Leave the terminal window SSH session open.

4. Determine the Instance ID of the MomPopInstance. In the SSH window for the CLI host, enter:

```
aws ec2 describe-instances \
--filters "Name=tag:Name,Values= MomPopCafeInstance" \
--query "Reservations[*].Instances[*].InstanceId"
```

5. Stop the Mom & Mom Café instance and change its instance type to t2.micro. In the SSH window for the CLI Host instance, enter:

```
aws ec2 stop-instances --instance-ids <MomPopCafeInstance Instance ID>
```

   In the command, substitute <MomPopCafeInstance Instance ID> with the value that you recorded earlier.

6. Change the instance type to t2.micro. In the SSH window for CLI Host instance, enter:

```
aws ec2 modify-instance-attribute \
--instance-id <MomPopCafeInstance Instance ID> \
--instance-type "{\"Value\": \"t2.micro\"}"
```

In the command, substitute <MomPopCafeInstance Instance ID> with he value from earlier commands.

7. If the command completes successfully, no output is returned.
8. Start the Mom & Pop Café instance. In the SSH window for the CLI Host instance, enter:

```
aws ec2 start-instances --instance-ids <MomPopCafeInstance Instance ID>
```

9. Check the current state of the instance, and wait until the status shows *running.* In the SSH window for the CLI host, enter:

```
aws ec2 describe-instances \
--instance-ids <MomPopCafeInstance Instance ID> \
--query "Reservations[*].Instances[*].[InstanceType,PublicDnsName,PublicIpAddress,State.Name]"
```

In the command, substitute <MomPopCafeInstance InstanceID> with the value that you recorded earlier.

10. The instance might take a few moments to reach the running state. Periodically repeat the command until you can confirm that it is running. Also, record the PublicDNSName and PublicIPAddress values and test whether the Mom & Pop website is functional by entering the URL in a browser window:

http://<Downsized MomPopCafeInstance Public DNS Name>/mompopcafe

```
[ec2-user@cli-host ~]$ aws ec2 describe-instances \
> --instance-ids i-0a3f53c94109de130 \
> --query "Reservations[*].Instances[*].[InstanceType,PublicDnsName,PublicIpAddress,State.Name]"
[
    [
        [
            "t2.micro",
            "ec2-34-207-232-194.compute-1.amazonaws.com",
            "34.207.232.194",
            "running"
        ]
    ]
]
[ec2-user@cli-host ~]$ Padmasree_RameshSrinivasan
```

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

1. Sign into the AWS Management console as devuser and open the Amazon S3 console.

   When you try to download an object from bucket 1, it shows Access Denied.

2. Assume the BucketsAccessRole IAM role in the Console:

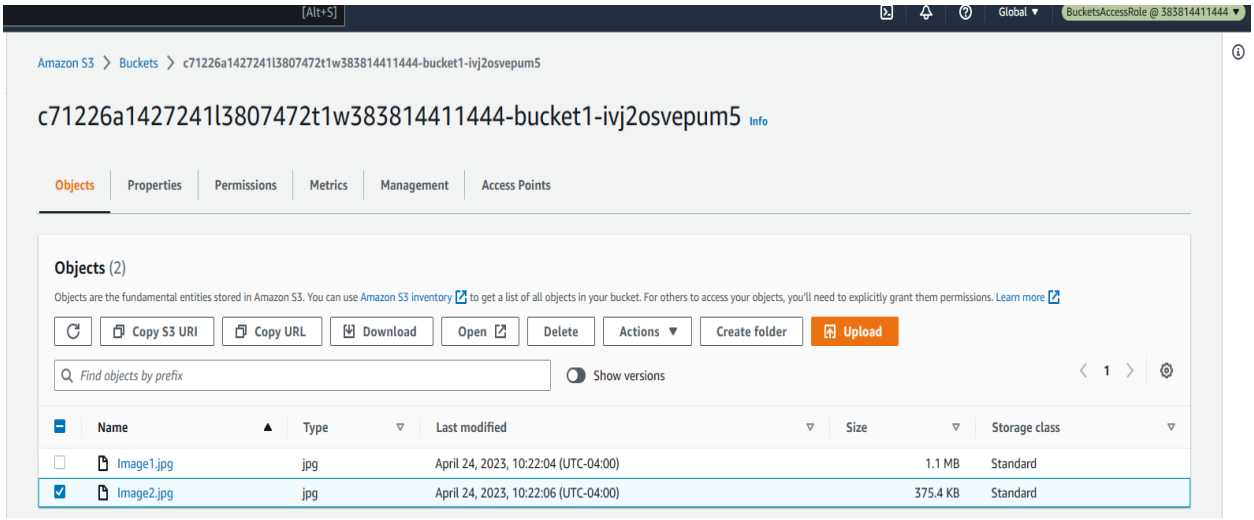
   - In the upper-right corner of the page, choose devuser, and then choose Switch role.
   - If the Switch role page appears, choose Switch Role.
   - Configure the following:
     - Account: Enter the Account ID value
     - Role: Enter BucketAccessRole
     - Choose Switch Role.

3. Now when you try to download an object from Amazon S3 again:

   - In the Amazon S3 console, choose the bucket and select a file and choose Download.
   - Open the file to verify that the file downloaded.

4. Observe the details of the bucket policy that is applied.

- On the details page for the bucket, choose the Permissions tab.
- In the Bucket policy, review the policy that is applied to that bucket.
- The policy Statement ID is S3Write. The principal is the BucketsAccessRole IAM Role that you assumed. This role is allowed to call the actions s3:GetObject and s3:PutObject on the resource, which is bucket2.
- The second SID is ListBucket. The principal is BucketsAccessRole. This role is allowed to call the action s3:ListBucket on the resource, which is bucket2.



How to setup AWS Config to monitor resources:

1. To set up AWS Config:
    - In the search box to the right of **Services,** search for and choose **Config.**
    - Choose **Get started**, and configure the following settings:
        - **Resource types to record**: Choose Record specific resource types.
        - **Resource category**: Choose AWS resources.
        - **Resource type**: Choose AWS EC2 SecurityGroup
        - **AWS Config Role**: Choose Choose a role from your account

- **Existing Roles:** Choose AwsConfigRole.
- In the Delivery method section, notice that AWS Config will store findings in an S3 bucket by default. Keep the default settings, and choose Next.
- On the AWS Managed Rules page, choose Next at the bottom of the page.
- Review the AWS Config setup details, and then choose **Confirm**.

A banner appears briefly, and then the AWS Config dashboard displays.

2. To observe the resource inventory that AWS Config created, in the navigation pane, choose Resources.

   The Resource Inventory page displays and lists the Amazon EC2 resources in your account.