

Operations Playbook

Name Richard Davis

Date: September 16, 2024

Class/Semester IFT 422 Fall 2024

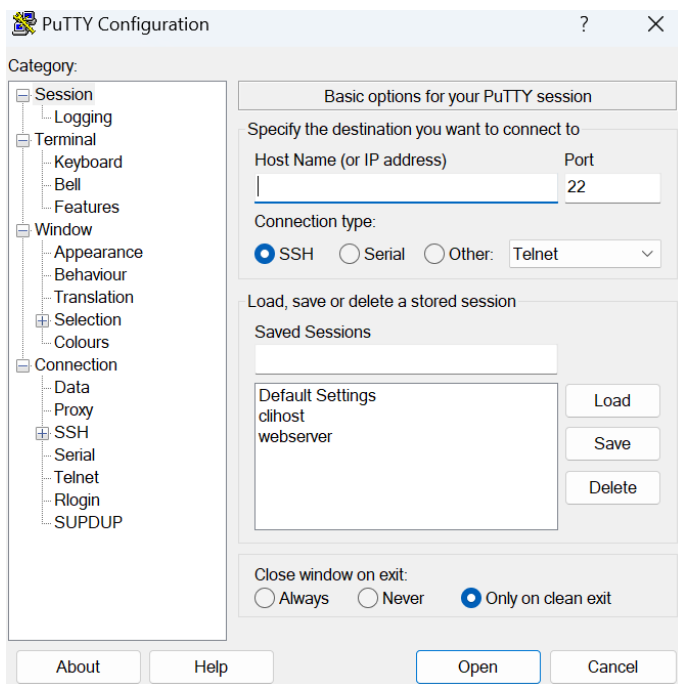
Contents

How to connect to the Mom & Pop Cafe Test EC2 instance	4
How to use the AWS CLI to connect to your AWS account	4
How to make a modification to the lab policy using the AWS CLI	5
How to add a parameter to the parameter store for allowing cookies on the website.....	6
How to connect to an EC2 instance to describe instances	7
How to launch an EC2 instance.....	8
How to fix a misconfigured web server with (_security group_) issue.....	10
How to change the AMI instance on the create-lamp-instance.sh script	11
How to tail a log in Linux.....	12
How to create an Auto Scaling Group in the AWS UI	13
How to create a Route 53 health check.....	14
How to create an Amazon RDS instance using the CLI	16
How to collect information about an instance	16
How to create two subnets in a subnet group via the AWS CLI	17
How to use the mysqldump tool to take a backup of a SQL database and restore it on another SQL instance	17
How to enable VPC Flow Logs via the command line interface.....	17
How to troubleshoot network connectivity on an instance	18
How to take a snapshot of an EBS volume	22
How to synchronize files using the command line (aws s3api and aws s3).....	23
How to create a S3 bucket via the CLI	25
How to add an event notification to a S3 bucket	26
How to create a CloudWatch Events/CloudWatch EventBridge notification rule.....	28
How to use the prebuilt stopinator script to turn off instances with the tag value of your full name ..	29
How to resize an EC2 instance using the AWS CLI	30
How to detect drift in a CloudFormation template	32
How to create an Amazon Athena table.....	35
How to manually review access logs to find anomalous user activity.....	37
How to create a batch file to update the café website to change its colors	39
How to create a Lambda Layer and add it to a Lambda function.....	40
How to create a Lambda function from a prebuilt package	41
How to setup a VPC.....	42

How to add a bastion host (Linux) to the public subnet of a VPC to connect to instances in the private subnet	43
How to setup IAM so a user can assume an IAM role to access a resource.....	44
How to setup AWS Config to monitor resources.....	45
How to add inbound rules to both security groups and network ACLs.....	46
How to encrypt the root volume of an existing EC2 instance	47
How to create a SNS topic	49
How to subscribe to a SNS topic.....	50
How to create a CloudWatch alarm using a metrics-based filter.....	50
How to install the CloudWatch Agent.....	53

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

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



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

1. Once connected to the ec2-user instance, run the configuration command "aws configure"
2. At the prompts enter:

AWS Access Key ID:

AWS Secret Access Key:

Default Region:

Default output format:

```
ec2-user@ip-10-200-0-127:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#  
~\_####_ Amazon Linux 2  
~~\_#####\  
~~\_###| AL2 End of Life is 2025-06-30.  
~~\_#/  
~~V~'-'>  
~~~ / A newer version of Amazon Linux is available!  
~~.-. /  
~~./ / Amazon Linux 2023, GA and supported until 2028-03-15.  
_/_m/'-' / https://aws.amazon.com/linux/amazon-linux-2023/  
ec2-user@ip-10-200-0-127 ~]$ aws configure  
AWS Access Key ID [None]: AKIARBA24KKKLOX2GG4Z  
AWS Secret Access Key [None]: ogO4GhLAbrrfVytclklkNnfc6YJeS5BK8klXufMR6  
Default region name [None]: us-east-1  
Default output format [None]: json  
ec2-user@ip-10-200-0-127 ~]
```

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

1. Use the following commands to find the policy:
"aws iam list-policies"
"aws iam list-policies --scope Local"
"aws iam get-policy-version --policy-arn <arn value> --version-id <version id> >lab_policy.json"
(taken from previous step)
In the last command, the arn value and version id are taken from the lab policy you want to change that is visible from the previous commands. Then the output is piped to the .json file
2. Use the command "Vi lab_policy.json" or "cat lab_policy.json" to see the results.

```
ec2-user@ip-10-200-0-104:~  
RichardDavis  
{  
  "PolicyVersion": {  
    "Document": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": [  
            "iam:get*",  
            "iam:list"  
          ],  
          "Resource": "*",  
          "Effect": "Allow"  
        }  
      ],  
      "VersionId": "v1",  
      "IsDefaultVersion": true,  
      "CreateDate": "2024-08-21T18:04:00:00"  
    }  
  }  
}
```

-- INSERT --

1,13 All

Breaking news
Divers find 4 bo...

Search

11:36 AM
8/21/2024

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

1. Use the AWS Systems Manager Parameter Store
2. Use the Management Console tab. 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 Cafe website.
 - Value: True
4. Click Create parameter
5. The parameter can be specified as a hierarchical path, such as: /dashboard/<option>

aws Services Search [Alt+S] N. Virginia voclabs/user2846850=Richard_Davis @ 0943-6101-6622

AWS Systems Manager > Parameter Store > Create parameter

Create parameter

Parameter details

Name

/web.config/cookie_toggle

When naming a parameter, you can use forward slashes (/) to organize it into a hierarchy. [Learn more about hierarchies](#)

Description — Optional

This feature allows you to turn cookies on or off for the Cafe website. Richard Davis

Tier

Parameter Store offers standard and advanced parameters.

☒ **Standard**

Store up to 10,000 standard parameters. Store parameter values up to 4 KB. Parameter policies and sharing with other AWS accounts are not available. No additional charge.

☐ **Advanced**

Store up to 100,000 advanced parameters. Store parameter values up to 8 KB. Add parameter policies. Share with other AWS accounts. Charges apply.

☐ Standard parameters cannot be shared with other AWS accounts. [Learn more](#)

Type

☒ **String**

Any string value.

☐ **StringList**

Separate strings using commas.

☐ **SecureString**

Encrypt sensitive data using KMS keys from your account or another account.

Data type

text

Value

True

Maximum length 4096 characters

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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
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/availability-zone``
`export AWS_DEFAULT_REGION=${AZ::-1}`
 - `# List information about EC2 instances`
`aws ec2 describe-instances`

```
    },
    "Architecture": "x86_64",
    "RootDeviceType": "ebs",
    "IamInstanceProfile": {
      "Id": "AIPAXELXBVD236OQIF7YP",
      "Arn": "arn:aws:iam::490412812533:instance-profile/App-Role"
    },
    "RootDeviceName": "/dev/xvda",
    "VirtualizationType": "hvm",
    "Tags": [
      {
        "Value": "c130857a331428017397296t1w490412812533",
        "Key": "cloudlab"
      },
      {
        "Value": "arn:aws:cloudformation:us-east-1:490412812533:stack/c130857a331428017397296t1w490412812533/c5d890e0-63f0-11ef-9dc8-121d62c03f89",
        "Key": "aws:cloudformation:stack-id"
      },
      {
        "Value": "SSMInstance",
        "Key": "aws:cloudformation:logical-id"
      },
      {
        "Value": "c130857a331428017397296t1w490412812533",
        "Key": "aws:cloudformation:stack-name"
      },
      {
        "Value": "Managed Instance",
        "Key": "Name"
      }
    ],
    "HibernationOptions": {
      "Configured": false
    },
    "MetadataOptions": {
      "State": "applied",
      "HttpEndpoint": "enabled",
      "HttpTokens": "optional",
      "HttpPutResponseHopLimit": 1
    },
    "AmiLaunchIndex": 0
  },
  "ReservationId": "r-0f088e06abe83541b",
  "RequesterId": "043234062703",
  "Groups": [],
  "OwnerId": "490412812533"
}
sh-4.2$ Richard Davis
```

How to launch an EC2 instance

1. Choose the Services menu, locate the Compute services, and select EC2. Choose the Launch instance button in the middle of the page, and then select Launch instance from the dropdown menu.
2. Name the instance:
Give it the name Bastion Server
Tags allow you to categorize your AWS resources in different ways, such as by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource by their tags. Each tag consists of a Key and a Value, both of which you define.
3. Choose an AMI from which to create the instance:
In the list of available Quick Start AMIs, keep the default Amazon Linux AMI selected.
Select Amazon Linux 2.
4. Choose an Instance Type
In the Instance type panel, keep the default t1.micro selected.
5. Step 4: Choose a key pair

Select the key pair to associate with the instance:

From the Key pair name menu, select vockey.

The vockey key pair you selected will allow you to connect to this instance via SSH after it has launched.

6. Network settings

You will launch the instance in a public subnet within the Lab VPC network.

Next to Network settings, choose Edit.

For VPC, choose the Lab VPC.

For Subnet accept the Public Subnet.

Keep the Auto-assign public IP setting set to Enable.

Under Firewall (security groups), keep the default Create security group option chosen.

7. You will create a new Security Group that permits SSH connections. This security group will allow you to log in to the Bastion Server via SSH.

8. Configure the security group:

Security group name: Bastion security group

Description: Permit SSH connections

Permissions to allow inbound access via SSH (port 22) have already been configured by default.

Keep these settings.

9. Configure storage

In the Configure storage section, keep the default settings.

10. Advanced details

Expand the Advanced details panel and for IAM instance profile, choose Bastion-Role

11. Launch the instance

At the bottom of the Summary panel on the right side of the screen choose Launch instance

You will see a Success message.

Choose View all instances

The Bastion Server instance will first appear in the Pending state, which means it is being launched. The state will then change to Running, which indicates that the instance has started booting. It takes a few minutes for the instance to boot.

Select the Bastion Server instance and review the information in the Details tab that displays in the lower pane.

Notice that the instance has a Public IPv4 address. You can use this IP address to communicate with the instance from the internet.

Before you continue, wait for your instance to display the following:

Instance state: Running

Status check: 2/2 checks passed

This may take a few minutes. Choose the refresh icon at the top of the page every 30 seconds or so to more quickly become aware of the latest status of the instance.

Name
e.g. My Web Server [Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

Amazon Linux 2023 AMI
ami-066784287e358dad1 (64-bit (x86), uefi-preferred) / ami-023508951a94f0c71 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs [Free tier eligible](#)

Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture: 64-bit (x86) Boot mode: uefi-preferred AMI ID: ami-066784287e358dad1 [Verified provider](#)

▼ **Instance type** [Info](#) | [Get advice](#)

Instance type
t1.micro
Family: t1 1 vCPU 0.612 GiB Memory Current generation: false
On-Demand Linux base pricing: 0.02 USD per Hour
On-Demand SUSE base pricing: 0.02 USD per Hour
On-Demand RHEL base pricing: 0.03 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour

[All generations](#) [Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

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

1. Open EC2 on AWS.
Open the Instances page.
Select the Misconfigured Web Server instance.
Copy the public IPv4.
2. Open Putty.exe.
Enter the IPv4 value into the hostname box on the Session Box.
Use the downloaded PPK file and select it in the Connection > SSH > Credentials.
Open the putty configuration.
Have a timeout error.
3. Go back to the AWS instances page.
Select the Misconfigured Web Server instance.

- ```
ec2-user@ip-10-0-0-12:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"

Amazon Linux 2023
~\#####\
~~_#####\
~~_###|
~~_#/ https://aws.amazon.com/linux/amazon-linux-2023
~~V~'-'->
~~~  
~~.-./-/  
~/m/'-/  
[ec2-user@ip-10-0-0-12 ~]$ Richard Davis
```

1. Connect to the Bastion Host instance via SSH using the appropriate PPK file
2. Update the AWS CLI using the “aws configure” command
3. Provide Access Key, Secret Access Key, region name, and output format
4. Use commands “cd ~/sysops-activity-files/starters” and “cp create-lamp-instance.sh create-lamp-instance.backup”
5. Open VI editor with “vi create-lamp-instance.sh”
6. Find the line of code where instance type is declared
7. Press “i” on keyboard to insert text
8. Replace the instance present with instance desired
9. Press “esc” on keyboard to exit insert mode
10. Type “:wq!” to save file and exit VI editor
11. Use command “./create-lamp-instance.sh” to run the script

```
ec2-user@cli-host:~/sysops-activity-files/starters
#!/bin/bash
DATE=`date '+%Y-%m-%d %H:%M:%S'`
echo
echo "Running create-instance.sh on "$DATE
echo

# Hard coded values
region="us-east-1"
echo "Region: "$region
instanceType="t2.small"
echo "Instance Type: "$instanceType
profile="default"
echo "Profile: "$profile

echo
echo "Looking up account values..."

# get vpcId
vpc=$(aws ec2 describe-vpcs \
--filters "Name=tag:Name,Values='MomPopCafe VPC'" \
--region $region \
--profile $profile | grep VpcId | cut -d '"' -f4 | sed -n 1p)
echo "VPC: "$vpc
```

10.1 Top

## How to tail a log in Linux

1. Connect to the Bastion Host instance via SSH using the appropriate PPK file
2. Update the AWS CLI using the "aws configure" command
3. Provide Access Key, Secret Access Key, region name, and output format
4. Run command "sudo tail -f /var/log/cloud-init-output.log"
5. The tail will run until "CTRL+C" then "ENTER" are pressed
6. You can alter the command to change the number of lines printed with "sudo tail -n..."
7. The original command uses the "-f" to make it a running tail, meaning it will continue tailing until stopped, to avoid this, remove the "-f".

```
ec2-user@web-server:~  
Starting Nmap 6.40 ( http://nmap.org ) at 2024-09-04 20:38 UTC  
Nmap scan report for ec2-54-152-131-25.compute-1.amazonaws.com (54.152.131.25)  
Host is up (0.0014s latency).  
Not shown: 998 filtered ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
8080/tcp   closed http-proxy  
  
Nmap done: 1 IP address (1 host up) scanned in 6.33 seconds  
[ec2-user@web-server ~]$ sudo tail -f /var/log/cloud-init-output.log  
nompopcafe/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.amzn2.0.2 finished at Wed, 04 Sep 2024 20:34:51 +0000. Dat  
source DataSourceEc2.  Up 59.45 seconds  
  
^C  
[ec2-user@web-server ~]$ Richard Davis
```

## How to create an Auto Scaling Group in the AWS UI

1. In the EC2 left navigation pane, scroll to the bottom of the menu, and choose Auto Scaling Groups and choose Create Auto Scaling group again on the following page.
2. In Step 1, Choose launch template or configuration, configure:  
Auto Scaling group name: Enter a name  
Launch template: Choose a template  
Choose Next
3. In the Network pane, configure:  
VPC: Choose VPC  
Subnets: Choose Private Subnet 1 and Private Subnet 2  
Choose Next
4. 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 a load balancer.  
In the Additional settings pane, select Enable group metrics collection within CloudWatch.  
Choose Next
5. In the Group size pane, configure:  
Desired capacity: Enter 7  
Minimum capacity: Enter 5  
Maximum capacity: Enter 10
6. In the Scaling policies pane, choose Target tracking scaling policy, and configure:  
Scaling policy name: Enter a name  
Metric type: Choose Average CPU utilization

Target value: Enter 45

Choose Next

7. On the Add notifications page, choose Next

8. On the Add tags page, choose Add tag and configure:

Key: Enter Name

Value: Enter a name

Choose Next

9. At the bottom of the Review page, choose Create Auto Scaling group

The screenshot displays the AWS Management Console interface for an Auto Scaling group named 'WebServersASGroup'. The top navigation bar shows the AWS logo, 'Services', a search bar, and the user's profile 'voclabs/user2846850=Richard\_Davis @ 2670-1195-7'. The left sidebar shows the 'EC2' menu with 'Auto Scaling groups' selected.

The main content area shows the 'Auto Scaling groups (1/1)' page. It includes a search bar, a table with columns for Name, Launch template/configuration, Instances, Status, and Desired capacity. The table lists one group: 'WebServersASGroup' with launch template 'WebServerLaunceTemplate', version '0', status 'Updating capacity...', and desired capacity '2'.

Below the table, the 'Auto Scaling group: WebServersASGroup' details are shown. The 'Details' tab is active, displaying a table with the following information:

| Group details                                             |                  |                             |                                                                                                                                        |
|-----------------------------------------------------------|------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Auto Scaling group name                                   | Desired capacity | Desired capacity type       | Amazon Resource Name (ARN)                                                                                                             |
| WebServersASGroup                                         | 2                | Units (number of instances) | arn:aws:autoscaling:us-east-1:267011957068:autoScalingGroup:d802ff2c-d25b-4dfc-b7ca-898d619a2659:autoScalingGroupName/WebServerASGroup |
| Date created                                              | Minimum capacity | Status                      |                                                                                                                                        |
| Mon Sep 09 2024 11:24:46 GMT-0700 (Pacific Daylight Time) | 2                | Updating capacity           |                                                                                                                                        |
|                                                           | Maximum capacity |                             |                                                                                                                                        |
|                                                           | 4                |                             |                                                                                                                                        |

The 'Launch template' tab is also visible, showing details for the 'WebServerLaunceTemplate' launch template. The table below contains the following information:

| Launch template                                 |                       |                      |                                                                          |
|-------------------------------------------------|-----------------------|----------------------|--------------------------------------------------------------------------|
| Launch template                                 | AMI ID                | Instance type        | Owner                                                                    |
| lt-06c82b08965e147b9<br>WebServerLaunceTemplate | ami-041e76668ea2a6e1a | t2.micro             | arn:aws:sts::267011957068:assumed-role/voclabs/user2846850=Richard_Davis |
| Version                                         | Security groups       | Security group IDs   | Create time                                                              |
| Default                                         | -                     | sg-0f6c7178b3dafb6c3 | Mon Sep 09 2024 11:20:00 GMT-0700 (Pacific Daylight Time)                |
| Description                                     | Storage (volumes)     | Key pair name        | Request Spot Instances                                                   |
| -                                               | -                     | -                    | No                                                                       |

A link 'View details in the launch template console' is provided at the bottom of the launch template section.

## How to create a Route 53 health check

1. In the AWS Management Console, from the Services menu, choose Route 53.

In the left navigation pane, click Health checks.

2. Click Create health check, and configure the following, leaving all other fields with their default values:  
Name:  
What to monitor:  
Specify endpoint by: IP address  
IP address:  
Path:
3. Expand Advanced configuration and configure the following, leaving all other fields with their default values:  
Request interval: Fast (10 seconds)  
Failure threshold: 2  
This will make your health check respond faster.  
Click Next.
4. Configure the following:  
Create alarm: Yes  
Send notification to: New SNS topic  
Topic name:  
Recipient email address: enter an email address that you can access
5. Click Create health check.

## Create health check

### Step 1: Configure health check

Step 2: Get notified when health check fails

### Configure health check

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name

What to monitor ☒ Endpoint  
☐ Status of other health checks (calculated health check)  
☐ State of CloudWatch alarm

#### Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.  
[Learn more](#)

Specify endpoint by ☒ IP address ☐ Domain name

Protocol

IP address \*

Host name

Port \*

Path

#### Advanced configuration

URL

Health check type Basic - no additional options selected ([View Pricing](#))

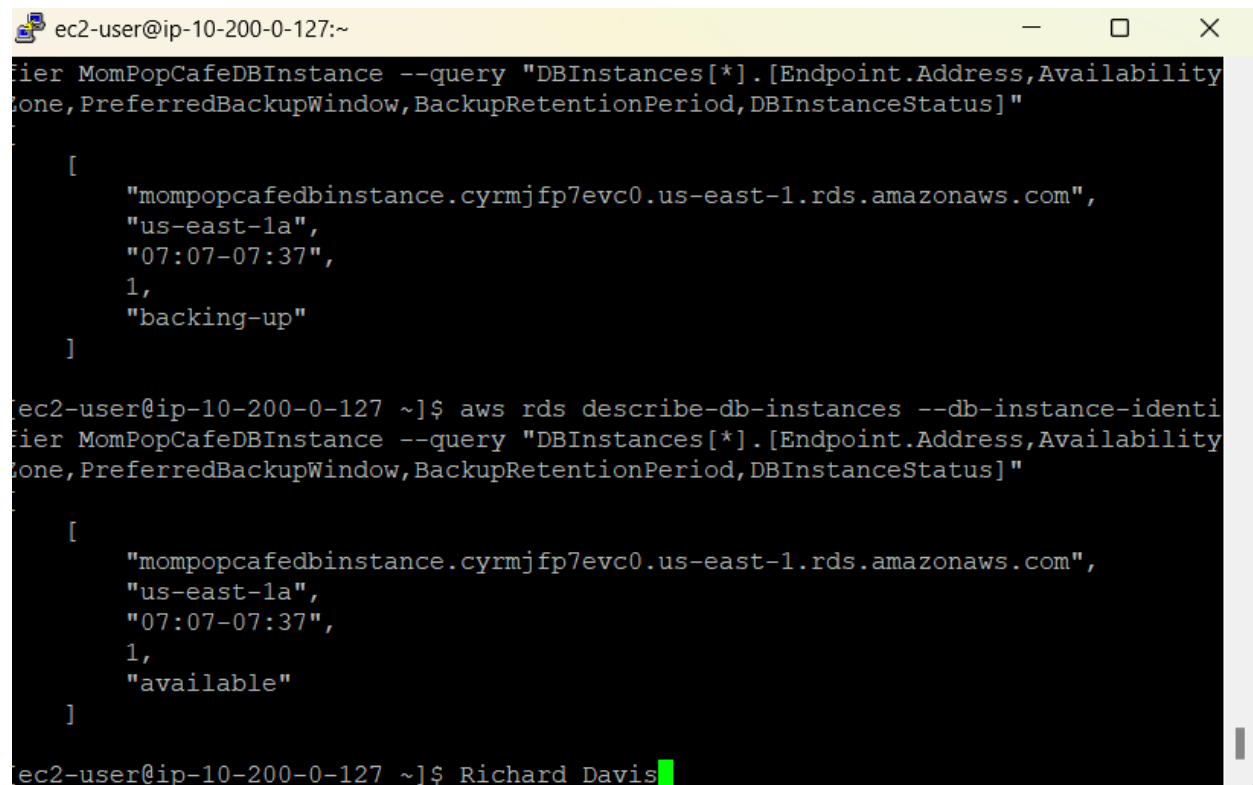
\* Required

Cancel

Next

## How to create an Amazon RDS instance using the CLI

1. After finding all relevant information, use the command:  
`aws rds create-db-instance --db-instance-identifier DBInstance --engine mariadb --engine-version 10.6.14 --db-instance-class db.t3.micro --allocated-storage 20 --availability-zone <Instance Availability Zone> --db-subnet-group-name "DB Subnet Group" --vpc-security-group-ids <DatabaseSG Group ID> --no-publicly-accessible --master-username root --master-user-password 'Re:Start!9'`
2. Monitor the database instance, until it shows a value of available. In the SSH window, enter:  
`aws rds describe-db-instances --db-instance-identifier DBInstance --query "DBInstances[*].[Endpoint.Address,AvailabilityZone,PreferredBackupWindow,BackupRetentionPeriod,DBInstanceStatus]"`



```
ec2-user@ip-10-200-0-127:~  
ier MomPopCafeDBInstance --query "DBInstances[*].[Endpoint.Address,Availability  
Zone,PreferredBackupWindow,BackupRetentionPeriod,DBInstanceStatus]"  
  
[  
  "mompopcafedbinstance.cyrmjfp7evc0.us-east-1.rds.amazonaws.com",  
  "us-east-1a",  
  "07:07-07:37",  
  1,  
  "backing-up"  
]  
  
ec2-user@ip-10-200-0-127 ~]$ aws rds describe-db-instances --db-instance-identi  
fier MomPopCafeDBInstance --query "DBInstances[*].[Endpoint.Address,Availability  
Zone,PreferredBackupWindow,BackupRetentionPeriod,DBInstanceStatus]"  
  
[  
  "mompopcafedbinstance.cyrmjfp7evc0.us-east-1.rds.amazonaws.com",  
  "us-east-1a",  
  "07:07-07:37",  
  1,  
  "available"  
]  
  
ec2-user@ip-10-200-0-127 ~]$ Richard Davis
```

## How to collect information about an instance

1. Determine the Instance ID, Instance Type, Public DNS name, Public IP address, and Availability Zone of the instance. Use the command:  
`aws ec2 describe-instances --filters "Name=tag:Name,Values= NameOfInstance" --query "Reservations[*].Instances[*].[InstanceId,InstanceType,PublicDnsName,PublicIpAddress,Placement.AvailabilityZone,VpcId,SecurityGroups[*].GroupId]"`
2. Determine the IPv4 CIDR block of the VPC. In the SSH window, enter:  
`aws ec2 describe-vpcs --vpc-ids <VPC ID> --filters "Name=tag:Name,Values= VPC" --query "Vpcs[*].CidrBlock"`
3. Determine the Subnet ID and IPv4 CIDR block of 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=<Instance VPC ID>" --query "Subnets[*].[SubnetId,CidrBlock]"
```

4. Determine the list of Availability Zones in the Region. In the SSH window, enter:  

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

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

1. With the VPC IPv4 CIDR block known, enter the command:  

```
aws ec2 create-subnet --vpc-id <VPC ID> --cidr-block 10.200.2.0/23 --availability-zone <Instance Availability Zone[a]>
```
2. For the second subnet use the same command:  

```
"aws ec2 create-subnet --vpc-id <VPC ID> --cidr-block 10.200.10.0/23 --availability-zone <Instance Availability Zone[b]>
```

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

1. In the SSH window, enter:  

```
mysqldump --user=root --password='Re:Start!9' --databases name_of_db --add-drop-database > nameofdb-backup.sql
```
2. if you want to view the backup using the Linux “less” command, in the SSH window, enter:  

```
less nameofdb-backup.sql
```
3. To restore the database from the backup, use the command:  

```
mysql --user=root --password='Re:Start!9' --host=<RDS Instance Database Endpoint Address> < nameofdb-backup.sql
```
4. To verify, open a mysql session, use the command:  

```
mysql --user=root --password='Re:Start!9' --host=<RDS Instance Database Endpoint Address> name_of_db
```
5. Enter the SQL statement “select \* from product;”
6. You can exit the mysql session with “exit”.

### How to enable VPC Flow Logs via the command line interface

1. Log into your CLI Host via SSH
2. Create the S3 bucket that will hold the flow logs by using the following command:  

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

In the command, replace #### with four random numbers and replace both occurrences of <region> with the region where the EC2 instances were created (for example, eu-west-2). If the region is us-east-1, delete the --create-bucket-configuration LocationConstraint=<region> portion of the command before you run it.
3. Run the following command to get the VPC ID for VPC1, which you must have to enable VPC Flow Logs:  

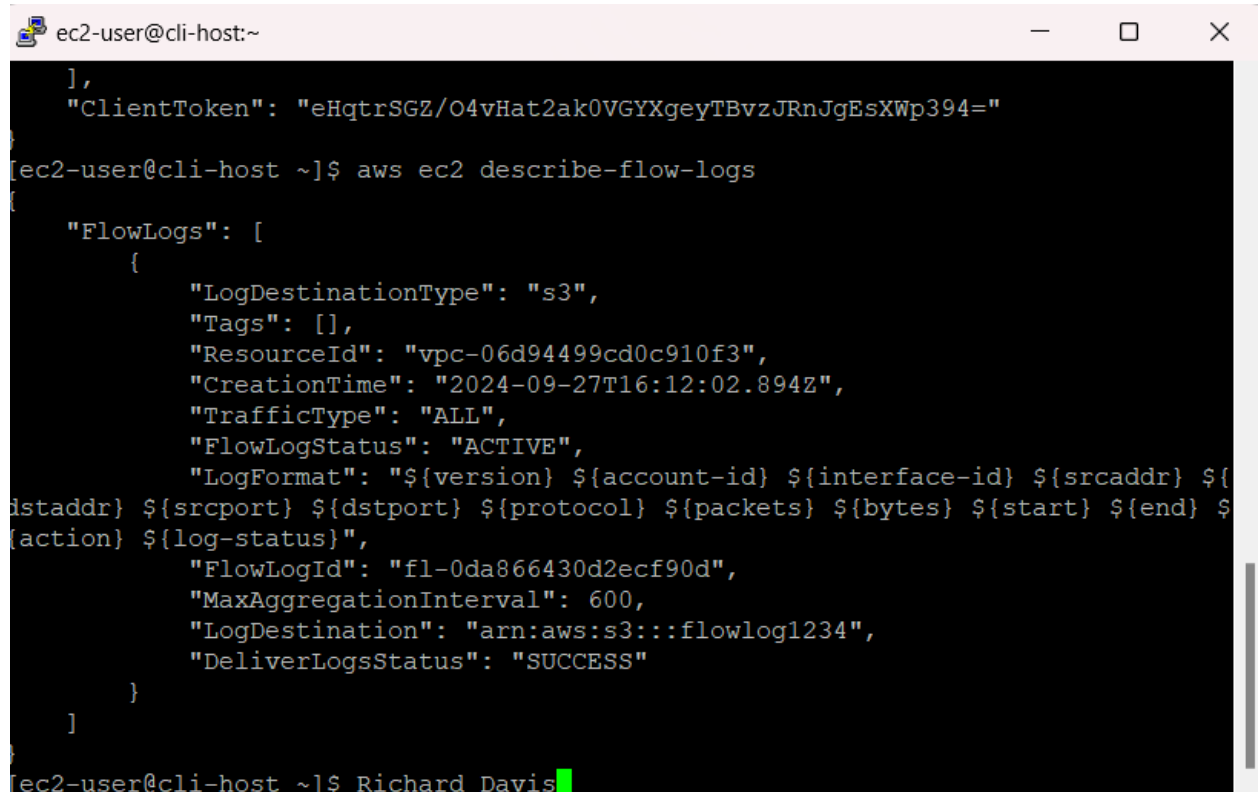
```
aws ec2 describe-vpcs --query 'Vpcs[*].[VpcId,Tags[?Key==`Name`].Value,CidrBlock]' --filters "Name=tag:Name,Values='VPC1'"
```
4. Enable VPC Flow Logs on VPC1 by running the following command.

```
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####>
```

5. In the command above, replace <flowlog####> with the actual bucket name. Also replace <vpc-id> with the actual VPC ID of VPC1.
6. If the command runs successfully, you should see that a FlowLogId and a ClientToken are returned.
7. Run the following command to confirm 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:~  
    ],  
    "ClientToken": "eHqtrSGZ/O4vHat2ak0VGyXgeyTBvzJRnJgEsXWp394=",  
  },  
  "FlowLogs": [  
    {  
      "LogDestinationType": "s3",  
      "Tags": [],  
      "ResourceId": "vpc-06d94499cd0c910f3",  
      "CreationTime": "2024-09-27T16:12:02.894Z",  
      "TrafficType": "ALL",  
      "FlowLogStatus": "ACTIVE",  
      "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end} ${action} ${log-status}",  
      "FlowLogId": "fl-0da866430d2ecf90d",  
      "MaxAggregationInterval": 600,  
      "LogDestination": "arn:aws:s3:::flowlog1234",  
      "DeliverLogsStatus": "SUCCESS"  
    }  
  ]  
}  
ec2-user@cli-host ~]$ Richard Davis
```

## How to troubleshoot network connectivity on an instance

1. Log into the CLI host using SSH
2. (Method 1) Use nmap with the ServerIP (ex. nmap 44.206.229.197)  
Check security groups with command  

```
aws ec2 describe-security-groups --group-ids <sg-####>
```

  
View output to see if there may be a rule blocking connection.

```

ec2-user@cli-host:~
bal.net

#
~\#### Amazon Linux 2
~~\#####
~~\#####\
~~\###| AL2 End of Life is 2025-06-30.
~~\#/
~~V~'-'>
~~~
~~~.
~~~/_/
~/m/'

A newer version of Amazon Linux is available!

Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

ec2-user@cli-host ~]$ nmap 44.206.229.197

Starting Nmap 6.40 (http://nmap.org) at 2024-09-27 18:02 UTC
Nmap scan report for ec2-44-206-229-197.compute-1.amazonaws.com (44.206.229.197)
Host is up (0.0012s latency).
```

3. (Method 2) Check the route tables with command  
`aws ec2 describe-route-tables --filter "Name=association.subnet-id,Values='<subnet id>'"`  
Check the route tables and add a route if needed with command  
`aws ec2 create-route --route-table-id <rtb id> --destination-cidr-block <cidr-block> --gateway-id <igw id>`
4. (Method 3) If you still cannot connect to the instance, reconnect to CLI host  
Check the network ACL with the command  
`aws ec2 describe-network-acls --filter "Name=association.subnet-id,Values='VPC1PublicSubnetID'" --query 'NetworkAcls[*].[NetworkAclId,Entries]'`  
Check the outputs and see if something is blocking connection
5. If there is, to delete the ACL entry that is blocking connection, use the command  
`aws ec2 delete-network-acl-entry --network-acl-id <acl id> --ingress --rule-number <rule number>`  
Connection should be reestablished.

```
ec2-user@cli-host:~
 "Egress": false,
 "CidrBlock": "0.0.0.0/0",
 "RuleAction": "deny"
 }
]
]
[ec2-user@cli-host ~]$ aws ec2 delete-network-acl-entry --network-acl-id acl-036
ae0dlffa598443 --ingress --rul-number 40
Note: AWS CLI version 2, the latest major version of the AWS CLI, is now stable
and recommended for general use. For more information, see the AWS CLI version 2
installation instructions at: https://docs.aws.amazon.com/cli/latest/userguide/
install-cliv2.html
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help
aws: error: argument --rule-number is required
[ec2-user@cli-host ~]$ aws ec2 delete-network-acl-entry --network-acl-id acl-036
ae0dlffa598443 --ingress --rule-number 40
[ec2-user@cli-host ~]$
```

6. (Method 4) To parse and search network logs to troubleshoot, reconnect to the CLI host
7. Create a local directory on the CLI Host where you can download the flow log files:  
mkdir flowlogs  
Change the directory to the new directory:  
cd flowlogs  
List the S3 buckets to recall the bucket name:  
aws s3 ls  
Download the flow logs by running the following command (replace <flowlog####> with the actual bucket name).  
aws s3 cp s3://<flowlog####>/ . --recursive  
If the command is successful, you should see that many files are downloaded.
8. Use the cd command and ls commands repeatedly (or cd followed by pressing TAB multiple times) as needed. The logs will be in an AWSLogs/<account-num>/vpcflowlogs/<region>/yyyy/mm/dd subdirectory.  
Notice that the file names all end in log.gz, which indicates that they are compressed as GNU zip files.  
Run this command to extract the logs:  
gunzip \*.gz  
Run ls again. Notice that all files are now extracted.
9. Copy one of the file names that were returned by the ls command that you ran.  
Enter head in the terminal window, followed by a space, and then paste the copied file name.  
Run the command.  
Notice that the header row indicates the kind of data that each log entry contains. Each entry contains information, such as the IP address of the source of the event (in the fourth column),

the destination port (seventh column), start and end timestamps (in Unix timestamp format), and the action that resulted (one of ACCEPT or REJECT).

10. Run a grep command that looks in each log file in the current directory, and returns lines that contain the word REJECT:

```
grep -rn REJECT .
```

This command should return a large dataset because it includes every event where the VPC settings rejected the request.

Find out how many records were returned:

```
grep -rn REJECT . | wc -l
```

The result shows the number of lines in your result set.

11. Refine your search by only looking for lines that contain 22 (which is the port number where you attempted to connect to the web server when access was blocked):

```
grep -rn '22' . | grep REJECT
```

This command should return a smaller number of results.

12. To isolate the result set—so that it only displays the log entries that correspond to the failed SSH connection attempts that you made—you must filter the results further.

Recall that your failed attempts to use SSH to connect the web server were initiated from your local machine. In this next step, you will determine the IP address by which your local machine is addressable from the internet.

13. Find the IP address to which your local computer is addressable from the internet.

Log in to the AWS Management Console.

Go to the EC2 service in the same Region where your EC2 instances are running.

Choose Security Groups.

Choose WebSecurityGroup and then choose the Inbound tab.

Choose Edit, then choose Add Rule.

In the third row that was just created, for Source, choose My IP.

Copy the IP address from the Classless Inter-Domain Routing (CIDR) block that is automatically populated (it will end in /32).

Copy only the IP address, not the /32 suffix.

Then, choose Cancel. You do not need to modify any security groups in this account. The purpose of this step is to capture this IP address.

14. Back in the CLI Host SSH terminal session, run a more refined query on the flow logs (replace <ip-address> with the IP address that you copied):

```
grep -rn '22' . | grep REJECT | grep <ip-address>
```

The number of lines in the result set should now match the number of times you tried and failed to use SSH to connect the web server instance.

Notice that the elastic network interface ID is in each of the log entries that were returned by your query.

15. Run the following command (replace <WebServerIP> with the actual IP address):

```
aws ec2 describe-network-interfaces --filters "Name=association.public-ip,Values='<WebServerIP>'" --query 'NetworkInterfaces[*].[NetworkInterfaceId,Association.PublicIp]'
```

The result set should confirm that the network interface ID that is recorded in the flow log matches the network interface that is assigned to the web server instance (as part of the network interface).

16. Translate the timestamps to human-readable form.

Notice the two long numbers that appear towards the end of each log entry, before the REJECT term.

These numbers are Unix-formatted timestamps. The first timestamp indicates the start time of each event that was captured. The second timestamp indicates the end time. You can convert them to human-readable form by using the Linux date command line utility.

```
date -d @1554496931
```

Run the date -d @ command for one of the captured timestamps from one of the filtered REJECT results. It should indicate a time from today that corresponds with when you were working through this activity. Run the date command to compare the result to the current time.

## How to take a snapshot of an EBS volume

1. To get a full description of the Processor instance, copy the following command and run it from within your instance:

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

This command uses the --filter tag to limit the results description to the new instance that you created in the previous section. The command will respond with a full, JSON-based description of the instance and all of its attributes. You will now modify this command to return just the subset of data—the Amazon EBS volume information—that you are interested in.

2. To narrow down the results of the previous command further, copy the following command and run it from within your instance:

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

This modified command uses the --query attribute to specify a JMESPath query that returns only the volume ID of the only volume (the root volume) attached to the Processor instance. You should receive a response similar to this:

```
{ "VolumeId": "vol-1234abcd" }
```

This value will be referred to as volume-id in subsequent commands.

3. Before taking a snapshot, you will shut down the Processor instance, which requires its instance ID. To obtain the instance ID, copy the following command and run it from within your instance:

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

This value will be referred to as instance-id in subsequent commands.

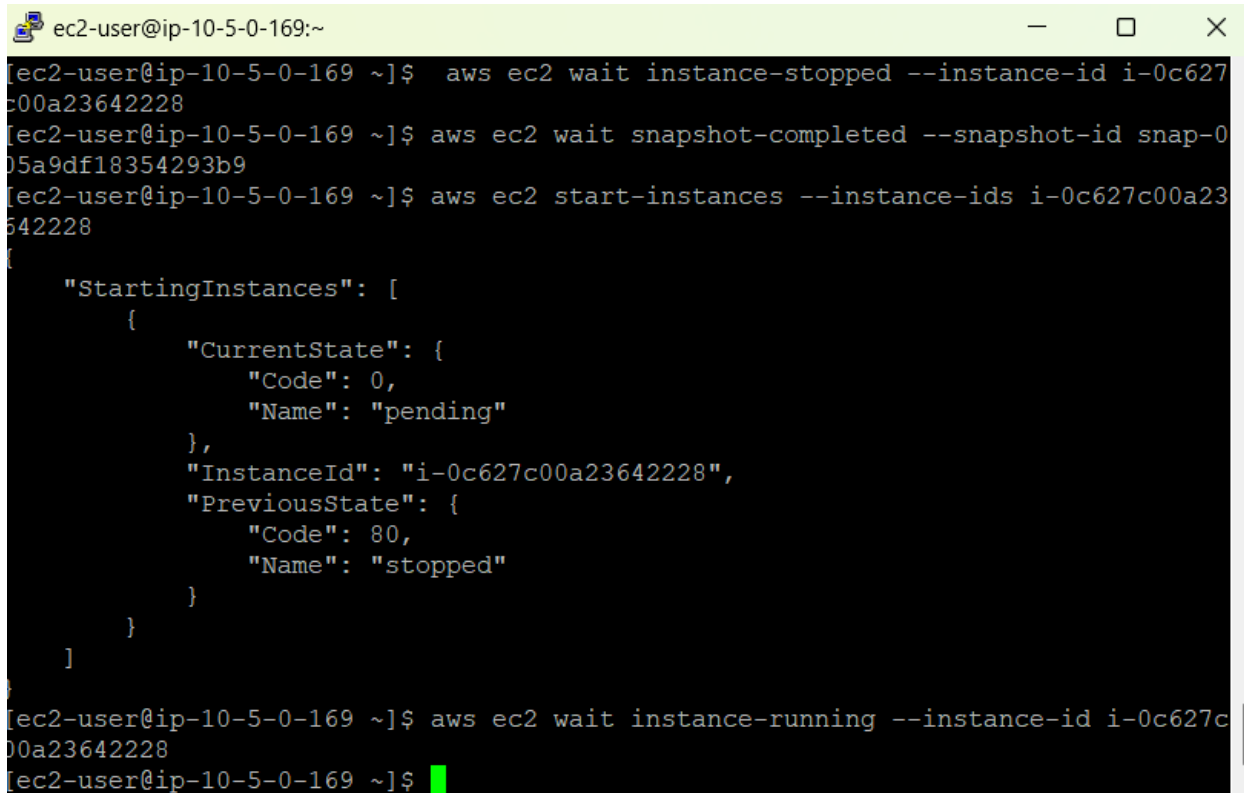
4. To shut down the Processor instance, copy the following command, replace INSTANCE-ID with your instance id, and run it from within your instance:

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

Before moving to the next step in this procedure, verify that the Processor instance has stopped by running the following command, replacing INSTANCE-ID with your instance id. When the Processor instance has stopped, the command will return to a prompt.

```
aws ec2 wait instance-stopped --instance-id INSTANCE-ID
```

5. To create your first snapshot of the root volume of your Processor instance, copy the following command, replace VOLUME-ID\_ with your volume id, and run it in your SSH window:  
`aws ec2 create-snapshot --volume-id VOLUME-ID`  
The command will return a set of information that includes a SnapshotId value that uniquely identifies the new snapshot. This value will be referred to as snapshot-id in subsequent commands.
6. To check the status of your snapshot, copy the following command, replace SNAPSHOT-ID your snapshot-id, and run it in your SSH window:  
`aws ec2 wait snapshot-completed --snapshot-id SNAPSHOT-ID`  
Continue with the below procedure when the command completes.
7. To restart the Processor instance, copy the following command, replace the INSTANCE-ID to your instance id and run it in your SSH window:  
`aws ec2 start-instances --instance-ids INSTANCE-ID`
8. To check on the status of the restart operation, copy the following command, replace INSTANCE-ID with your instance id, and run it in your SSH window:  
`aws ec2 wait instance-running --instance-id INSTANCE-ID`



```
ec2-user@ip-10-5-0-169:~
[ec2-user@ip-10-5-0-169 ~]$ aws ec2 wait instance-stopped --instance-id i-0c627c00a23642228
[ec2-user@ip-10-5-0-169 ~]$ aws ec2 wait snapshot-completed --snapshot-id snap-005a9df18354293b9
[ec2-user@ip-10-5-0-169 ~]$ aws ec2 start-instances --instance-ids i-0c627c00a23642228
{
 "StartingInstances": [
 {
 "CurrentState": {
 "Code": 0,
 "Name": "pending"
 },
 "InstanceId": "i-0c627c00a23642228",
 "PreviousState": {
 "Code": 80,
 "Name": "stopped"
 }
 }
]
}
[ec2-user@ip-10-5-0-169 ~]$ aws ec2 wait instance-running --instance-id i-0c627c00a23642228
[ec2-user@ip-10-5-0-169 ~]$
```

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

1. Login to the Processor instance.
2. To download the sample files on the Processor instance, copy the following command and run it from within your instance:  
`wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/CUR-TF-200-RESOPS/lab5vocareum/files.zip`

3. To unzip the directory, use the following command:  
`unzip files.zip`

#### Synchronizing Files

4. Before synchronizing content with your Amazon S3 bucket, you will need to enable versioning on your bucket. To enable versioning, copy the following command (replacing S3-BUCKET-NAME with your bucket name) and run it from within your instance:  
`aws s3api put-bucket-versioning --bucket S3-BUCKET-NAME --versioning-configuration Status=Enabled`
5. To synchronize the contents of the files folder with your Amazon S3 bucket, copy the following command (replacing S3-BUCKET-NAME with your bucket name) and run it from within your instance:  
`aws s3 sync files s3://S3-BUCKET-NAME/files/`
6. To confirm the state of your files, use the following command (replacing S3-BUCKET-NAME with your bucket name):  
`aws s3 ls s3://S3-BUCKET-NAME/files/`
7. To delete one of the files on the local drive, use the following command:  
`rm files/file1.txt`
8. To delete the same file from the server, use the `--delete` option to the `aws s3 sync` command. Copy the following command (replacing S3-BUCKET-NAME with your bucket name) and run it from within your instance:  
`aws s3 sync files s3://S3-BUCKET-NAME/files/ --delete`
9. Verify that the file was deleted remotely on the server:  
`aws s3 ls s3://S3-BUCKET-NAME/files/`
10. 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`
11. Because 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 (replacing VERSION-ID with your version-id and S3-BUCKET-NAME with your bucket name) and run it from within your instance:  
`aws s3api get-object --bucket S3-BUCKET-NAME --key files/file1.txt --version-id VERSION-ID files/file1.txt`
12. To verify that the file has been restored locally, use the following command:  
`ls files`
13. To re-sync the contents of the files/ folder to Amazon S3, copy the following command (replacing S3-BUCKET-NAME with your bucket name) and run it from within your instance:  
`aws s3 sync files s3://S3-BUCKET-NAME/files/`
14. Finally, to verify that a new version of file1.txt has been pushed to Amazon S3, copy the following command (replacing S3-BUCKET-NAME with your bucket name) and run it from within your instance:  
`aws s3 ls s3://S3-BUCKET-NAME/files/`



```

ec2-user@ip-10-5-0-50:~
[ec2-user@ip-10-5-0-50 ~]$ aws s3api get-object --bucket rdavis39-s3 --key files
/file1.txt --version-id R1DvkI9YTS4aDcXgTmqEMcmPz3BgCEmp files/file1.txt
{
 "AcceptRanges": "bytes",
 "ContentType": "text/plain",
 "LastModified": "Thu, 03 Oct 2024 18:49:40 GMT",
 "ContentLength": 30318,
 "VersionId": "R1DvkI9YTS4aDcXgTmqEMcmPz3BgCEmp",
 "ETag": "\"b76b2b775023e60be16bc332496f8409\"",
 "ServerSideEncryption": "AES256",
 "Metadata": {}
}
[ec2-user@ip-10-5-0-50 ~]$ ls files
file1.txt file2.txt file3.txt
[ec2-user@ip-10-5-0-50 ~]$ aws s3 sync files s3://rdavis39-s3/files/
upload: files/file1.txt to s3://rdavis39-s3/files/file1.txt
[ec2-user@ip-10-5-0-50 ~]$ aws s3 ls files s3://rdavis39-s3/files/

Unknown options: s3://rdavis39-s3/files/
[ec2-user@ip-10-5-0-50 ~]$ aws s3 ls s3://rdavis39-s3/files/
2024-10-03 18:54:06 30318 file1.txt
2024-10-03 18:49:40 43784 file2.txt
2024-10-03 18:49:40 96675 file3.txt
[ec2-user@ip-10-5-0-50 ~]$ Richard Davis

```

## How to create a S3 bucket via the CLI

1. Create the <mompopcafe-xxxxnnn> S3 bucket. Because an S3 bucket name must be unique across all existing bucket names in Amazon S3, you will add a suffix to the name with a format of -xxxxnnn. For xxx, substitute your initials. For nnn, substitute a random number. In the SSH window for the CLI Host instance, enter:  
`aws s3 mb s3://<mompopcafe-xxxxnnn> --region <region>`  
 In the command, substitute <mompopcafe-xxxxnnn> with your unique S3 bucket name. Also, substitute <region> with the region where your CLI Host instance is running.  
 When the make bucket (mb) command completes successfully, it returns the name of the bucket.
2. Load some images in the S3 bucket under the /images prefix. Sample image files are provided in the initial-images folder on the CLI Host. In the SSH window for the CLI Host instance, enter:  
`aws s3 sync ~/initial-images/ s3://<mompopcafe-xxxxnnn>/images`  
 In the command, substitute <mompopcafe-xxxxnnn> with your unique S3 bucket name.  
 As the synchronize (sync) command runs, you will see the names of the image files being uploaded.
3. List the bucket contents by 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. In the SSH window for the CLI Host instance, enter:  
`aws s3 ls s3://<mompopcafe-xxxxnnn>/images/ --human-readable --summarize`  
 In the command, substitute <mompopcafe-xxxxnnn> with your unique S3 bucket name.  
 When the list (ls) command completes, you will see the details of the image files that were uploaded, and their total number and size.

```
ec2-user@ip-10-200-0-105:~
t-1
-bash: mompopcafe-red123: No such file or directory
[ec2-user@ip-10-200-0-105 ~]$ aws s3 sync ~/initial-images/ s3://<mompopcafe-red123>/images
-bash: mompopcafe-red123: No such file or directory
[ec2-user@ip-10-200-0-105 ~]$ aws s3 mb s3://mompopcafe-red123 --region us-east-1
make_bucket: mompopcafe-red123
[ec2-user@ip-10-200-0-105 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafe-red123/images
upload: initial-images/Cup-of-Hot-Chocolate.jpg to s3://mompopcafe-red123/images/Cup-of-Hot-Chocolate.jpg
upload: initial-images/Donuts.jpg to s3://mompopcafe-red123/images/Donuts.jpg
upload: initial-images/Strawberry-Tarts.jpg to s3://mompopcafe-red123/images/Strawberry-Tarts.jpg
[ec2-user@ip-10-200-0-105 ~]$ aws s3 ls s3://mompopcafe-red123/images/ --human-readable --summarize
2024-10-03 19:21:49 308.7 KiB Cup-of-Hot-Chocolate.jpg
2024-10-03 19:21:49 371.8 KiB Donuts.jpg
2024-10-03 19:21:49 468.0 KiB Strawberry-Tarts.jpg

Total Objects: 3
Total Size: 1.1 MiB
[ec2-user@ip-10-200-0-105 ~]$ Richard Davis
```

## How to add an event notification to a S3 bucket

1. In Services choose Simple Notification Service.
2. If necessary, choose the menu icon ( ) on the left to open the navigation pane.
3. In the navigation pane, select Topics.
4. Choose Create topic.
5. Choose Standard.
6. In the Name box, enter s3NotificationTopic.
7. Choose Create topic.  
A message is displayed indicating that the s3NotificationTopic was successfully created.
8. Copy and paste the value of the topic ARN field in a text editor to save it.  
You will need to supply it when you create the topic's access policy in the next steps and also later.
9. Configure the topic's access policy. In the s3NotificationTopic pane, choose Edit.
10. 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-xxxxnnn>"
 }
 }
 }
]
}

```

In the JSON object, substitute <ARN of s3NotificationTopic> with the value of the topic ARN that you recorded earlier, and <mompopcafe-xxxxnnn> with your unique S3 bucket name. Also remember to remove the enclosing angle brackets (< >) during the substitution.

Take a moment to review the intent of this policy. It grants the mompopcafe S3 share bucket the permission to publish messages to the s3NotificationTopic.

11. Choose Save changes.

12. Lastly, subscribe Pop to the topic as the mompopuser who will receive the event notifications from the S3 share bucket.

Choose Create subscription.

In the topic ARN box, the s3NotificationTopic already appears.

13. In the Protocol menu, select Email.

14. In the Endpoint box, enter an email address that you can access.

Note: For the purposes of this activity, you are going to pretend that you are Pop so you receive the S3 event notifications.

Choose Create subscription. A message is displayed confirming that the subscription was created successfully.

Check the inbox for the email address that you provided. How to install the CloudWatch Agent

**New Feature**  
Amazon SNS now supports in-place message archiving and replay for FIFO topics. [Learn more](#)

Amazon SNS > Topics > Create topic

## Create topic

**Details**

Type [Info](#)

Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)
 

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ Standard
 

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (\_).

Display name - optional [Info](#)

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

Maximum 100 characters.

► **Encryption - optional**

Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

▼ **Access policy - optional** [Info](#)

This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

Choose method

☒ Basic
 

Use simple criteria to define a basic access policy.

☐ Advanced
 

Use a JSON object to define an advanced access policy.

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

- Go to Amazon EventBridge
- Choose Create rule  
Enter a name  
Choose Next
- 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
- 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.  
On the Configure tags - optional page, choose Next

## 5. On the Review and create page, choose Create rule

The screenshot shows the Amazon EventBridge console. The left sidebar contains navigation links for Dashboard, Developer resources, Buses, Pipes, Scheduler, Integration, and Schema registry. The main content area displays the 'Instance\_Stopped\_Terminated' rule details. The rule is enabled and has a standard type. The event pattern is defined as follows:

```
1 {
2 "source": ["aws.ec2"],
3 "detail-type": ["EC2 Instance State-change Notification"],
4 "detail": {
5 "state": ["stopped", "terminated"]
6 }
7 }
```

The console also shows tabs for Event pattern, Targets, Monitoring, and Tags. A 'Copy' button is available for the event pattern.

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

1. From the Linux shell, run the stopinator.php script:  
`./stopinator.php -t"Richard Davis"`  
The output should look like this, indicating that two instances will be stopped in your current AWS region. (Your results will differ depending on the region in which your lab is running.)
2. Region is us-east-1  
No instances to stop in region  
Region is us-west-1  
No instances to stop in region  
Region is us-west-2  
Found instance i-9552ba9f  
Found instance i-d35fb7d9  
Stopping all identified instances...  
[...]  
No instances to stop in region  
Region is sa-east-1  
No instances to stop in region
3. On the Services menu, choose EC2.

4. In the navigation pane, choose Instances.
5. Verify that two instances are stopping or have already been stopped.
6. Return to the SSH session for Command Host, and from the Linux prompt, restart your instances with the following command:  
`./stopinator.php -t"Richard Davis" -s`  
 Return to the EC2 Management Console window and verify that the two instances that were previously shut down are now restarting.

```

ec2-user@ip-10-5-0-180:~/aws-tools
PHP Notice: Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 115

 Stopping 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 120

 No instances to stop in Array.
Region is us-west-1
PHP Notice: Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 120

 No instances to stop in Array.
Region is us-west-2
PHP Notice: Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 120

 No instances to stop in Array.
Region is us-west-3
PHP Notice: Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 120

 No instances to stop in Array.
[ec2-user@ip-10-5-0-180 aws-tools]$ aws-tools]$./stopinator.php -t"Project=ERP System;Environment=development" -s
-bash: aws-tools]$: command not found
[ec2-user@ip-10-5-0-180 aws-tools]$ aws-tools]$./stopinator.php -t"Richard Davis" -s

```

## How to resize an EC2 instance using the AWS CLI

1. Stop the local database and uninstall it from the Mom & Pop Café instance. In the SSH window for the MomPopCafeInstance, enter:  
`sudo systemctl stop mariadb`  
`sudo yum -y remove mariadb-server`
2. If the last command runs successfully, you will see a Complete! message in the output.
3. Close the SSH window for the MomPopCafeInstance because you no longer need it.
4. Determine the Instance ID of the MomPopCafeInstance. Switch to the SSH window for the CLI Host instance and enter:  
`aws ec2 describe-instances \`  
`--filters "Name=tag:Name,Values= MomPopCafeInstance" \`  
`--query "Reservations[*].Instances[*].InstanceId"`
5. Record the value returned as:  
 MomPopCafeInstance Instance ID: i-nnnnnnnnnn

6. Stop the Mom & Pop 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 recorder earlier.
7. Change the instance type to t2.micro. In the SSH window for the 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 the value that you recorder earlier.  
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>`  
In the command, substitute <MomPopCafeInstance Instance ID> with the value that you recorder earlier.
9. Check the current state of the instance, and wait until the status shows running. In the SSH window for the CLI Host instance, enter:  
`aws ec2 describe-instances \`  
`--instance-ids <MomPopCafeInstance Instance ID> \`  
`--query`  
`"Reservations[*].Instances[*].[InstanceType,PublicDnsName,PublicIpAddress,State.Name]"`  
In the command, substitute <MomPopCafeInstance Instance ID> with the value that you recorder 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 that are returned by the command by using the following format:  
Downsized MomPopCafeInstance Public DNS Name: ec2-zzz-zzz-zzz-zzz.eu-west-2.compute.amazonaws.com  
Downsized MomPopCafeInstance Public IP Address: nnn.nnn.nnn.nnn  
Information: Because you restarted the instance, Amazon EC2 will assign a different Public DNS name and Public IP address to the instance than what it had before.
11. Test the Mom & Pop Café website to make sure that it is functional. In a browser window, enter the following URL:  
`http://<Downsized MomPopCafeInstance Public DNS Name>/mompopcafe`  
Substitute <Downsized MomPopCafeInstance Public DNS Name> with the value that you recorded.

```
ec2-user@cli-host:~
 }
]
}
[ec2-user@cli-host ~]$ aws ec2 modify-instance-attribute \
> --instance-id <MomPopCafeInstance Instance ID> \
> --instance-type "{\"Value\": \"t2.micro\"}"
-bash: MomPopCafeInstance: No such file or directory
[ec2-user@cli-host ~]$ aws ec2 modify-instance-attribute --instance-id i-03a23124df3725f3e --instance-type "{\"Value\": \"t2.micro\"}"
[ec2-user@cli-host ~]$ aws ec2 start-instances --instance-ids i-03a23124df3725f3e
{
 "StartingInstances": [
 {
 "InstanceId": "i-03a23124df3725f3e",
 "CurrentState": {
 "Code": 0,
 "Name": "pending"
 },
 "PreviousState": {
 "Code": 80,
 "Name": "stopped"
 }
 }
]
}
```

```
ec2-user@cli-host:~
 }
]
}
[ec2-user@cli-host ~]$ aws ec2 describe-instances \
> --instance-ids <MomPopCafeInstance Instance ID> \
> --query "Reservations[*].Instances[*].[InstanceType,PublicDnsName,PublicIpAddress,State.Name]"
-bash: MomPopCafeInstance: No such file or directory
[ec2-user@cli-host ~]$ aws ec2 describe-instances --instance-ids i-03a23124df3725f3e --query "Reservations[*].Instances[*].[InstanceType,PublicDnsName,PublicIpAddress,State.Name]"
[
 [
 [
 "t2.micro",
 "ec2-3-81-134-158.compute-1.amazonaws.com",
 "3.81.134.158",
 "running"
]
]
]
[ec2-user@cli-host ~]$
```

## 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



The command should return a StackDriftDetectionId.

2. Monitor the status of the drift detection by running the following command (replace <driftId> with the actual value of StackDriftDetectionId):

```
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
```

The output from the command is extensive. Try a different approach.

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

5. The following command outputs the results as a table:

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

This output is easier to read because of the query parameter, which is written in JMESPath.

Notice that not all resources are checked for drift. However, the resources that are checked for drift show a status.

On this stack, all checked resources have a status of IN\_SYNC, except for the security group that you manually modified, which has a status of MODIFIED.

Also notice that though you placed an object in the S3 bucket, the bucket still shows a status of IN\_SYNC. If you had modified some property of the bucket, then the bucket would show a status of MODIFIED. However, only adding files to a bucket does not register as drift in AWS CloudFormation.

6. 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 the PropertyDifferences section of the output. It should show that port 22 is now open only to your IP address, instead of the 0.0.0.0/0 Classless Inter-Domain Routing (CIDR) block that is defined in the AWS CloudFormation template.

7. 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. You must manually resolve these issues to eliminate the drift.

```

ec2-user@cli-host:~
(NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist
[ec2-user@cli-host ~]$ aws s3 cp myfile s3://mystack-mybucket-zxxqlhdppdde/
upload: ./myfile to s3://mystack-mybucket-zxxqlhdppdde/myfile
[ec2-user@cli-host ~]$ aws s3 ls mystack-mybucket-zxxqlhdppdde
2024-10-09 20:38:19 0 myfile
[ec2-user@cli-host ~]$ aws cloudformation detect-stack-drift --stack-name myStack
{
 "StackDriftDetectionId": "86debbbc0-867e-11ef-bad8-0ea92c7d3591"
}
[ec2-user@cli-host ~]$ aws cloudformation describe-stack-drift-detection-status \
> --stack-drift-detection-id 86debbbc0-867e-11ef-bad8-0ea92c7d3591
{
 "StackId": "arn:aws:cloudformation:us-east-1:150225838146:stack/myStack/5a3a9f40-867d-11ef-b11f-1264993986c7",
 "StackDriftDetectionId": "86debbbc0-867e-11ef-bad8-0ea92c7d3591",
 "StackDriftStatus": "DRIFTED",
 "Timestamp": "2024-10-09T20:39:05.852Z",
 "DetectionStatus": "DETECTION_COMPLETE",
 "DriftedStackResourceCount": 1
}
[ec2-user@cli-host ~]$

```

```

ec2-user@cli-host:~
-----DescribeStackResources-----
+-----+-----+-----+
| AWS::EC2::InternetGateway | CREATE_COMPLETE | IN_SYNC |
| AWS::EC2::VPC | CREATE_COMPLETE | IN_SYNC |
| AWS::S3::Bucket | CREATE_COMPLETE | IN_SYNC |
| AWS::EC2::Route | CREATE_COMPLETE | IN_SYNC |
| AWS::EC2::RouteTable | CREATE_COMPLETE | IN_SYNC |
| AWS::EC2::SubnetRouteTableAssociation | CREATE_COMPLETE | NOT_CHECKED |
| AWS::EC2::Subnet | CREATE_COMPLETE | IN_SYNC |
| AWS::EC2::VPCGatewayAttachment | CREATE_COMPLETE | NOT_CHECKED |
| AWS::CloudFormation::WaitCondition | CREATE_COMPLETE | NOT_CHECKED |
| AWS::CloudFormation::WaitConditionHandle | CREATE_COMPLETE | NOT_CHECKED |
| AWS::EC2::SecurityGroup | CREATE_COMPLETE | MODIFIED |
| AWS::EC2::Instance | CREATE_COMPLETE | IN_SYNC |
+-----+-----+-----+
[ec2-user@cli-host ~]$ aws cloudformation describe-stack-resource-drifts \
> --stack-name myStack \
> --stack-resource-drift-status-filters MODIFIED
{
 "StackResourceDrifts": [
 {
 "StackId": "arn:aws:cloudformation:us-east-1:150225838146:stack/myStack/5a3a9f40-867d-11ef-b11f-1264993986c7",

```

```
ec2-user@cli-host:~
...0.0.0/0\", \"FromPort\":22, \"IpProtocol\": \"tcp\", \"ToPort\":22}, {\"CidrIp\": \"0
...0.0.0/0\", \"FromPort\":80, \"IpProtocol\": \"tcp\", \"ToPort\":80}], \"Tags\": [{\"K
ey\": \"Name\", \"Value\": \"WebServerSG\"}], \"VpcId\": \"vpc-011b2d8559b8ee0f5\"},

 \"PropertyDifferences\": [
 {
 \"PropertyPath\": \"/SecurityGroupIngress/0/CidrIp\",
 \"ActualValue\": \"76.240.119.242/32\",
 \"ExpectedValue\": \"0.0.0.0/0\",
 \"DifferenceType\": \"NOT_EQUAL\"
 }
],
 \"LogicalResourceId\": \"WebSecurityGroup\"
 }
]
}
[ec2-user@cli-host ~]$ aws cloudformation update-stack \
> --stack-name myStack \
> --template-body file://template1.yaml \
> --parameters ParameterKey=KeyName,ParameterValue=vockey

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

## How to create an Amazon Athena table

1. open the CloudTrail console.
2. In the navigation pane, choose Event history.
3. 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. The Event history page can be a useful tool, and you are free to explore it. However, in this activity, you will use Amazon Athena.
4. From the Event history page, choose Create Athena table.
5. Storage location: Choose the monitoring#### Amazon S3 bucket where you configured CloudTrail to store log files.
6. Take a moment to analyze how the Amazon Athena CREATE TABLE statement is formed.
7. It will create a database column for each of the standard name-value pairs in each JSON-formatted CloudTrail log entry. Refer back to the image of the JSON format of a typical log entry in Task 3.4 to confirm this.
8. At the bottom of the CREATE TABLE SQL statement, notice the LOCATION statement. This indicates the Amazon S3 location where the table data will be stored. In this case, the data is already there. You are defining the table schema that will be used to parse existing JSON-structured data.  
For details on AWS CloudTrail record structure, see <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-event-reference.html>.  
For details on how this Amazon Athena table was created, see the CREATE EXTERNAL TABLE document at <https://docs.aws.amazon.com/athena/latest/ug/cloudtrail-logs.html>.
9. After you are done analyzing the CREATE TABLE details, choose Create table.

10. The table is created with a default name that includes the name of the Amazon S3 bucket.
11. In the search box next to Services search for and select the Athena service to open the Athena console.

### Create a table in Amazon Athena



You can use Amazon Athena to analyze events that are stored in a trail's Amazon S3 bucket. Athena is an interactive query service that helps you analyze data in S3 buckets by using standard SQL. Athena charges for running queries.[Learn more](#)

Storage location

monitoring1103



Choose an S3 bucket that contains CloudTrail log files

Athena table name

cloudtrail\_logs\_monitoring1103

This name is auto-generated. You can rename it in Amazon Athena.

### Athena table query

Copy

```
1 CREATE EXTERNAL TABLE cloudtrail_logs_monitoring1103 (
2 eventVersion STRING,
3 userIdentity STRUCT<
4 type: STRING,
5 principalId: STRING,
6 arn: STRING,
7 accountId: STRING,
8 invokedBy: STRING,
9 accessKeyId: STRING,
10 userName: STRING,
11 sessionContext: STRUCT<
12 attributes: STRUCT<
13 mfaAuthenticated: STRING,
14 creationDate: STRING>,
15 sessionIssuer: STRUCT<
16 type: STRING,
17 principalId: STRING,
18 arn: STRING,
19 accountId: STRING,
20 username: STRING>,
21 ec2RoleDelivery: STRING,
22 webIdFederationData: MAP<STRING,STRING>>>,
23 eventTime STRING,
24 eventSource STRING,
25 eventName STRING,
26 awsRegion STRING,
27 sourceIpAddress STRING,
28 userAgent STRING,
29 errorCode STRING,
30 errorMessage STRING,
31 requestParameters STRING,
32 responseElements STRING,
33 additionalEventData STRING,
34 requestId STRING,
```

Cancel

Create table

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

1. In the left panel of the Athena Query Editor, you should see the `cloudtrail_logs_monitoring####` table.
2. Choose the plus icon next to table name to reveal the column names.  
Analysis: Notice how each standard child element that exists in a CloudTrail log record in JSON format has a corresponding column name in this database. The `useridentity` database column is a struct, because it contains more than a single name-value pair. Similarly, the `resources` database column is an array.
3. Start by setting up a query results location and then running a simple query to get an idea of the data that is available in the logs.
4. Choose the View settings button that appears above the query panel, then choose Manage.
5. Choose Browse S3, select your `monitoring####` bucket, and select Choose.
6. In the Location of query result box, add `/results/` to the value, so that it now reads `s3://monitoring####/results/` where `monitoring####` is the name of the bucket you created earlier.  
Choose Save.
7. Return to the Editor tab.
8. Paste the following SQL query into the New query 1 panel. Replace `####` with the numbers in your actual table, and choose Run.  

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

This query returns 5 rows of data. Look at the result set (scroll to the right in the Results panel to see additional column data).

The `useridentity`, `eventtime`, `eventsources`, `eventname`, `requestparameters` columns look like they contain interesting data.

That `useridentity` column has lots of detail that make it more difficult to read though. You will now return only the user name for that column.
9. Run a new query that selects only those columns that were previously mentioned. This time, limit the results to 30 rows:  

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

This information is interesting, but recall what you are looking for.

Specifically, someone modified the security group that is associated with the Cafe Web Server instance, and you want to know who it was.
10. TIP #1: Choose the + icon next to New query 1 to create a second query tab. This way, you can preserve older queries without deleting them.  
TIP #2: Try filtering by events that are related to the EC2 service. Remember that you can add WHERE clauses, such as `WHERE eventsources = 'ec2.amazonaws.com'`  
TIP #3: To ensure you are querying the entire log set, remove the LIMIT clause from your query.  
TIP #4: Take a look at the kind of data that is captured in the `eventname` column. Can you further refine your SQL query so that it looks for only events that contain the word Security? Remember that SQL allows you to use compound WHERE clauses that look for pattern matches.

For example: WHERE columnName = 'some value' AND otherColumnName LIKE '%part of some value%'

TIP #5: After you have successfully filtered all security-related actions in the log, analyze the eventnames further. Do any of them look suspicious? Can you adjust the WHERE clause to search for a particular eventname?

TIP #6: If you are still looking for the entry that shows who opened port 22 to the world, here is a general query that is often useful to run. This query might help identify the action:

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

The screenshot displays the Amazon Athena Query Editor interface. On the left, the 'Data' panel shows the 'cloudtrail\_logs\_monitoring1103' table with its schema, including fields like 'eventversion', 'eventtime', 'eventsource', 'eventname', and 'requestparameters'. The 'Tables and views' section lists the available tables. The main editor area contains a SQL query: 

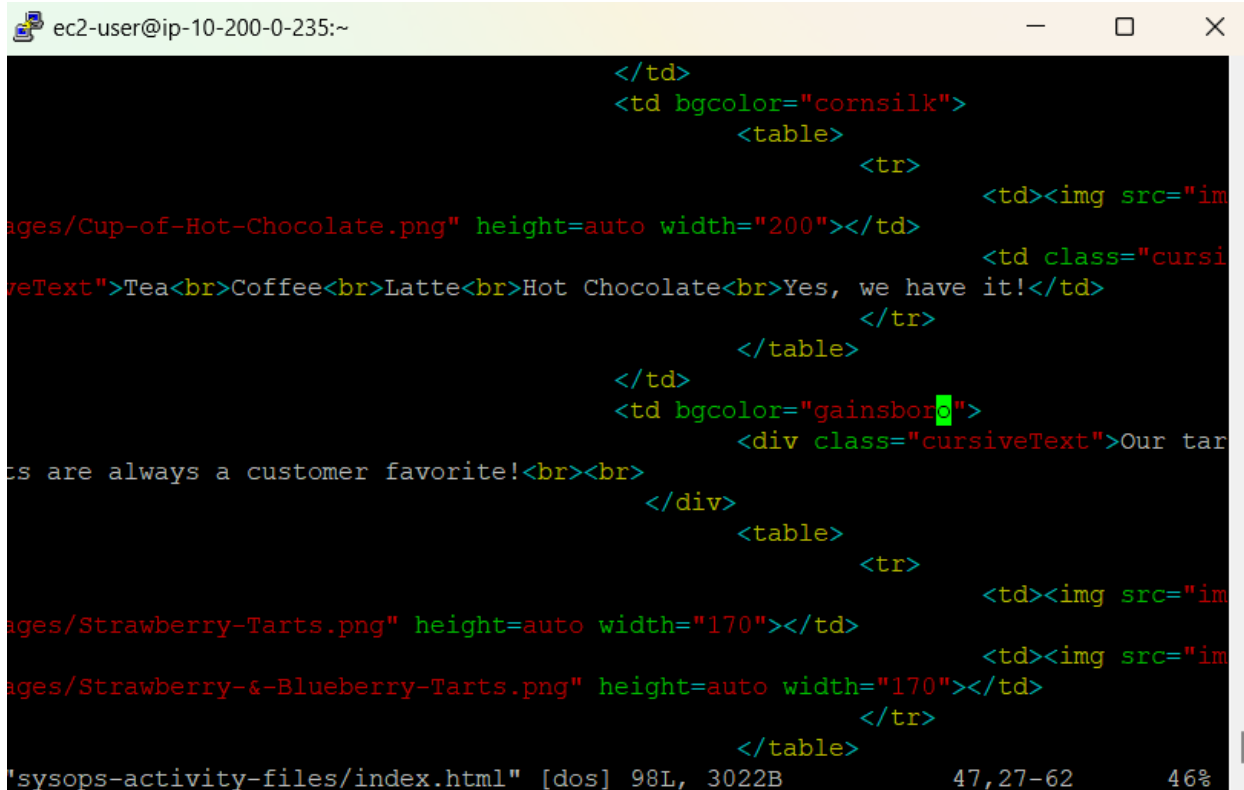
```
1 SELECT useridentity.userName, eventtime, eventsource, eventName, requestparameters
2 FROM cloudtrail_logs_monitoring1103
3 WHERE useridentity.userName = 'chaos' AND eventName LIKE '%Security%'
4
```

 Below the query editor, the 'Query results' section shows a 'Completed' status with performance metrics: 'Time in queue: 63 ms', 'Run time: 795 ms', and 'Data scanned: 94.99 KB'. The results are displayed in a table with 247 rows, showing columns for '#', 'userName', 'eventtime', 'eventsource', and 'eventName'. The results show multiple entries for the user 'chaos' performing 'DescribeSecurityGroups' actions from 'ec2.amazonaws.com' at various times on 2024-10-15.

| # | userName | eventtime            | eventsource       | eventName              |
|---|----------|----------------------|-------------------|------------------------|
| 1 | chaos    | 2024-10-15T19:17:12Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 2 | chaos    | 2024-10-15T19:17:24Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 3 | chaos    | 2024-10-15T19:17:35Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 4 | chaos    | 2024-10-15T19:17:46Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 5 | chaos    | 2024-10-15T19:17:58Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 6 | chaos    | 2024-10-15T19:18:09Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 7 | chaos    | 2024-10-15T19:18:20Z | ec2.amazonaws.com | DescribeSecurityGroups |
| 8 | chaos    | 2024-10-15T19:18:32Z | ec2.amazonaws.com | DescribeSecurityGroups |

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

1. Use PuTTY to SSH to Amazon EC2 instances by using the Public DNS or IPv4 address of the Bastion Host, use the .ppk file for credentials, login as "ec2-user"
2. Update AWS CLI software using command "aws configure", use AWS AccessKey, AWS SecretKey, region, and output format.
3. Create an empty file using command "touch update-website.sh"
4. Use the VI editor with command "vi update-website.sh", enter editing mode by pressing "A"
5. Enter the following code into the file, where "<my-bucket>" is replaced with the actual bucket name:
  - `#!/bin/bash`  
`aws s3 cp ~/sysops-activity-files/ s3://<my-bucket>/ --recursive --acl public-read`
6. Quit the file by pressing ESC, typing ":wq" and then pressing ENTER.
7. Use the command "chmod +x update-website.sh" to make the file executable.
8. Open the index.html file with the VI editor
9. In the file, find the locations of the code "bgcolor=" and change the color from 'aquamarine' to 'gainsboro' and 'orange' to 'cornsilk', save and exit the file using the same method as before.



```
ec2-user@ip-10-200-0-235:~
 </td>
 <td bgcolor="cornsilk">
 <table>
 <tr>
 <td></td>
 <td class="cursiveText">Tea
Coffee
Latte
Hot Chocolate
Yes, we have it!</td>
 </tr>
 </table>
 </td>
 <td bgcolor="gainsboro">
 <div class="cursiveText">Our tarts are always a customer favorite!

 </div>
 <table>
 <tr>
 <td></td>
 <td></td>
 </tr>
 </table>
'sysops-activity-files/index.html' [dos] 98L, 3022B 47,27-62 46%
```

10. Use the command "./update-website.sh" to run the batch file.

```

ec2-user@ip-10-200-0-235:~
 8 ls
 9 rm static-website.tar.gz
10 ls
11 aws s3api put-bucket-ownership-controls --bucket <my-bucket> \
12 aws s3api put-bucket-ownership-controls --bucket rdavis113 --ownership-co
ntrols "Rules=[{ObjectOwnership=BucketOwnerPreferred}]"
13 aws s3api put-public-access-block --bucket rdavis113 --public-access-bloc
k-configuration "BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=
false,RestrictPublicBuckets=false"
14 aws s3 website s3://rdavis113/ --index-document index.html
15 aws s3 cp . s3://rdavis113/ --recursive --acl public-read
16 aws s3 ls rdavis113
17 history
[ec2-user@ip-10-200-0-235 sysops-activity-files]$ cd ~
[ec2-user@ip-10-200-0-235 ~]$ touch update-website.sh
[ec2-user@ip-10-200-0-235 ~]$ vi update-website.sh
[ec2-user@ip-10-200-0-235 ~]$ chmod +x update-website.sh
[ec2-user@ip-10-200-0-235 ~]$ vi sysops-activity-files/index.html
[ec2-user@ip-10-200-0-235 ~]$./update-website.sh
[ec2-user@ip-10-200-0-235 ~]$ cat update-website.sh
#!/bin/bash
#aws s3 cp ~/sysops-activity-files/ s3://rdavis113/ --recursive --acl public-read
[ec2-user@ip-10-200-0-235 ~]$ Richard Davis

```

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

1. In the Lambda Function overview panel, your function, choose Layers.
2. In the Layers panel at the bottom of the page, choose Add a layer.  
In the Add layer page, configure as follows:
3. Choose a layer: Select the Custom layers card  
Custom layers:  
Version:
4. Choose Add.



**Add layer**

**Function runtime settings**

|                       |                        |
|-----------------------|------------------------|
| Runtime<br>Python 3.8 | Architecture<br>x86_64 |
|-----------------------|------------------------|

**Choose a layer**

**Layer source** [Info](#)  
Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also [create a new layer](#).

☒ **AWS layers**  
Choose a layer from a list of layers provided by AWS.

☐ **Custom layers**  
Choose a layer from a list of layers created by your AWS account or organization.

☐ **Specify an ARN**  
Specify a layer by providing the ARN.

**AWS layers**  
Layers provided by AWS that are compatible with your function's runtime.

Choose ▼

Cancel Add

## How to create a Lambda function from a prebuilt package

1. In the AWS Management Console, select Services > Lambda.
2. Choose Layers.
3. Choose Create layer.  
Configure the layer settings as follows:  
Name:  
Description:  
Code entry type: Upload a .zip file  
Choose Upload, navigate to the folder where your zip file is and open it.  
Compatible runtimes: Choose Python 3.8 .
4. Choose Create.

**aws** Services Search [Alt+S] N. Virginia voclabs/user2846850=Richard\_Davis @ 5016-6908-7689

Lambda > Functions > Create function

## Create function [Info](#)

Choose one of the following options to create your function.

☒ **Author from scratch**  
 Start with a simple Hello World example.

☐ **Use a blueprint**  
 Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**  
 Select a container image to deploy for your function.

### Basic information

**Function name**  
 Enter a name that describes the purpose of your function.  
  
 Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime** [Info](#)  
 Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
 [Refresh](#)

**Architecture** [Info](#)  
 Choose the instruction set architecture you want for your function code.  
☒ x86\_64  
☐ arm64

**Permissions** [Info](#)  
 By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

**Execution role**  
 Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
 ☒ Use an existing role
 ☐ Create a new role from AWS policy templates

**Existing role**  
 Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
 [Refresh](#)  
[View the salesAnalysisReportDERole role](#) on the IAM console.

► **Advanced settings**

Cancel **Create function**

## How to setup a VPC

1. Navigate to the VPC service.
2. Choose Create VPC and configure:  
 Name tag: VPC name  
 IPv4 CIDR block: 10.0.0.0/16
3. Choose Create VPC.
4. Choose Actions and select Edit DNS hostnames.
5. Under DNS hostnames, select Enable, then choose Save changes

Services
Search
[Alt+S]
N. Virginia
voclabs/user2846850=Richard\_Davis @ 1058-3378-2841

VPC > Your VPCs > Create VPC

## Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

### VPC settings

**Resources to create** [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only
☐ VPC and more

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

my-vpc-01

**IPv4 CIDR block** [Info](#)

☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/24

CIDR block size must be between /16 and /28.

**IPv6 CIDR block** [Info](#)

☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

**Tenancy** [Info](#)

Default

### Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource

Add tag

You can add 50 more tags

Cancel
Create VPC

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

1. Navigate to EC2 service.
2. Select Launch Instance.
3. Configure:
  - Name: Bastion Server
  - Application and OS Images:
    - Quick Start: Amazon Linux
    - AMI: Amazon Linux 2023 AMI (HVM)
4. Instance Type:
  - Instance Type: t2.micro
  - Key pair (login):
    - Key pair name: vockey
5. Network settings:
  - Choose Edit

VPC: Lab VPC

Subnet: Public Subnet

Auto-assign public IP: Enable

Security group name: BastionSG

Description: BastionSG

6. Inbound security groups rules: Keep the default setting which will provide SSH access.

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

vockey [Create new key pair](#)

▼ **Network settings** [Info](#)

VPC - *required* [Info](#)

vpc-0e9385fae834fc4a3 (Lab VPC) 10.0.0.0/16 [Create new VPC](#)

Subnet [Info](#)

subnet-08d5163f20f49844f Public Subnet [Create new subnet](#)

VPC: vpc-0e9385fae834fc4a3 Owner: 105833782841 Availability Zone: us-east-1a  
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.0.0/24

Auto-assign public IP [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - *required*

BastionSG

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .-:/!#,@[]+=&{}!\$\*

Description - *required* [Info](#)

BastionSG

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

| Type <a href="#">Info</a> | Protocol <a href="#">Info</a> | Port range <a href="#">Info</a> |
|---------------------------|-------------------------------|---------------------------------|
| ssh                       | TCP                           | 22                              |

| Source type <a href="#">Info</a> | Source <a href="#">Info</a>                                          | Description - <i>optional</i> <a href="#">Info</a> |
|----------------------------------|----------------------------------------------------------------------|----------------------------------------------------|
| Anywhere                         | <a href="#">Add CIDR, prefix list or security group</a><br>0.0.0.0/0 | e.g. SSH for admin desktop                         |

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


1. Find out what role is needed to access a resource.
2. In the upper right corner of the page, select the username.
3. Then choose **Switch Role**
4. Configure the following:

Account ID

Role

Display name: leave blank

## 5. Choose Switch Role

English ▼

### Switch Role

Switching roles enables you to manage resources across Amazon Web Services accounts using a single user. When you switch roles, you temporarily take on the permissions assigned to the new role. When you exit the role, you give up those permissions and get your original permissions back. [Learn more](#)

**Account ID**  
The 12-digit account number or the alias of the account in which the role exists.

**IAM role name**  
The name of the role that you want to assume which can be found at the end of the role's ARN. For example, provide the **TestRole** role name from the following role ARN: `arn:aws:iam::123456789012:role/TestRole`.

**Display name - optional**  
This name will appear in the console navigation bar when active. Choose a name to help identify the permission set assigned to the role.

**Display color - optional**  
The selected color displays in the console navigation when this role is active

☐ None ▼

Cancel

Switch Role

### How to setup AWS Config to monitor resources

1. In the search box to the right of **Services**, search for and choose **Config**.
2. Choose **Get started**, and configure the following settings:
3. Under **Recording strategy**. Choose **Specific resource types**.
4. **Resource type**: Choose **AWS EC2 SecurityGroup**. For **Frequency** choose **Continuous**.
5. **IAM role for AWS Config** Choose **Choose a role from your account**.
6. **Existing roles**: Choose **AwsConfigRole**.  
Note: Recall that **AwsConfigRole** was the second role that you analyzed in the previous task.
7. 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**.
8. On the **AWS Managed Rules** page, choose **Next** at the bottom of the page.
9. Review the **AWS Config** setup details, and then choose **Confirm**.

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

Allow/Deny: Choose Deny.

Choose Save changes.

7. In the Amazon EC2 console, in the navigation pane, choose Security Groups.

8. Select the AppServerSG security group.

9. Choose the Inbound rules tab, and then choose Edit inbound rules.

10. Choose Add rule, and configure as follows:

Type: Choose SSH.

Source: Choose Custom.

Next, paste the BastionPrivateIP value from the lab instructions into the Source field.

After pasting, add /32 to the end of the IP address.

Choose Save rules.

**Edit inbound rules** [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

| Security group rule ID | Type       | Protocol | Port range | Source | Description - optional |                        |
|------------------------|------------|----------|------------|--------|------------------------|------------------------|
| sgr-06ceab2e253925ed0  | HTTP       | TCP      | 80         | Cu...  | Richard Davis          | <a href="#">Delete</a> |
| -                      | Custom TCP | TCP      | 0          | Cu...  |                        | <a href="#">Delete</a> |

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

[Cancel](#) [Preview changes](#) [Save rules](#)

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

1. Go to EC2>Instances.

2. 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.

4. Choose the Storage tab.

5. In the Block devices section, choose the link for the Volume ID.

6. Choose the link for the Volume ID again.

7. Note the Availability Zone where the volume exists (for example, us-east-1a or us-east-1b).

Important: You will need this information in a moment.

8. 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.

9. Create an encrypted volume from the unencrypted snapshot.

10. In the navigation pane, under Elastic Block Store, choose Snapshots.

Choose the link for the Unencrypted Root Volume snapshot ID that you just created.

Wait until the Snapshot status shows Completed.

Notice that the encryption status of the snapshot is Not encrypted.

Choose Actions > Create volume from snapshot, and configure the following:

Availability Zone: Choose the Availability Zone where the existing volume exists.

Select Encrypt this volume.

KMS key: Choose MyKMSKey.

Choose Create volume.

Label the volumes.

11. In the navigation pane, under Elastic Block Store, choose Volumes.

Notice that two volumes are now listed.

12. For the volume with a Volume state of In-use, change the volume name:

Hover on the Name field, and 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 encrypted root volume

13. Swap the root volume that the EC2 instance uses.

Select Old unencrypted root volume, and then choose Actions > Detach volume.

To confirm, choose Detach.

14. Select New encrypted root volume, and then choose Actions > Attach volume and configure the following:

Instance: Choose (LabInstance) (stopped).

Device name: Enter /dev/xvda

Note: This is the device name where the existing instance expects to find the root volume.

15. Choose Attach volume.

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 a AWS KMS key ID.

Note: You might need to refresh the page to see the latest information for the attached volume.



aws Services Search [Alt+S] N. Virginia voclabs/user2846850=Richard\_Davis @ 9332-2201-1980

EC2 > Snapshots > snap-0c1a00bbaa8be76ed > Create volume

## Create volume [Info](#)

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

### Volume settings

Snapshot ID  
☐ snap-0c1a00bbaa8be76ed (Unencrypted Root Volume)

Volume type [Info](#)

Size (GiB) [Info](#)  
  
 Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS [Info](#)  
  
 Min: 3000 IOPS, Max: 16000 IOPS. The value must be an integer.

Throughput (MiB/s) [Info](#)  
  
 Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

Availability Zone [Info](#)

Fast snapshot restore [Info](#)  
☐ Not enabled for selected snapshot

Encryption  
 Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.  
☒ Encrypt this volume

KMS key [Info](#)

KMS key description  
 -

KMS key owner  
☐ 933222011980 (This account)

KMS key ID  
☐ 5863453a-7ab6-43a5-806a-3026487c0967

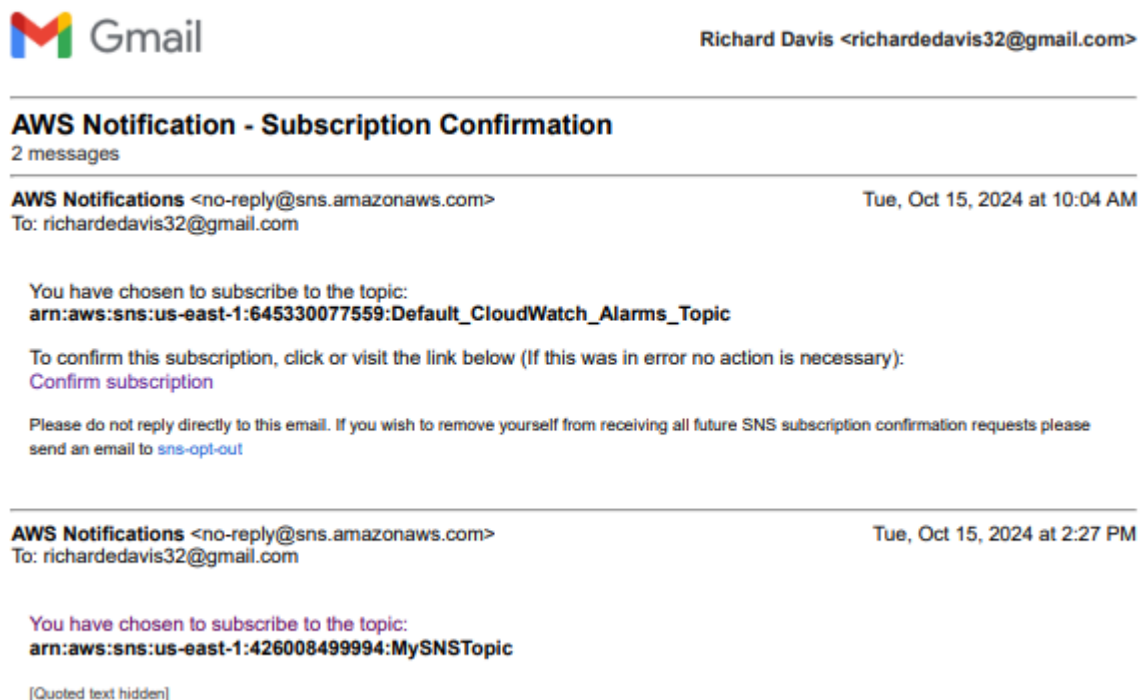
KMS key ARN

## How to create a SNS topic

1. Open the Amazon SNS console.  
In the navigation pane, choose Topics.
2. Choose Create topic, and configure the following:  
Type: Choose Standard.  
Name: Enter MySNSTopic
3. Expand the Access policy - optional section.  
Define who can publish messages to the topic: Choose Everyone.  
Define who can subscribe to this topic: Choose Everyone.  
At the bottom of the page, choose Create topic.

## How to subscribe to a SNS topic

1. To create an email subscription to the SNS topic, choose Create subscription, and configure the following:  
Topic ARN: Notice that the Amazon Resource Number (ARN) of the topic that you just created is already filled in.  
Protocol: Choose Email.  
Endpoint: Enter an email address where you can receive emails during this lab.  
Scroll to the bottom of the page and choose Create subscription.
2. 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.  
Note: Recall that when you created the CloudTrail trail, you configured it to create this log group.  
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

Observe the settings. This alarm will be invoked 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 credentials 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 user name.

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 user name: Enter test

Password: test

Choose Sign in.

Note: Each time that you attempt to log in, you will see a message indicating that your authentication information is incorrect. This is expected!

6. Re-establish your access to the AWS account.

Close all browser tabs where you have the AWS Management Console open.

On the lab instructions page, choose the AWS link above these instructions to log in again as the voclabs user.

Important: Your attempts to log in to the console as the test user cleared the previous authentication information from your browser's cache. Therefore, you need to re-authenticate to gain access to the console.

7. Graph the metric that you created.

Navigate to the CloudWatch console.

In the navigation pane, expand Metrics, and then choose All metrics.

In the Metrics section, under Custom namespaces, choose CloudTrailMetrics.

Note: If CloudTrailMetrics does not yet appear, wait until the SNS notification is received.

Choose Metrics with no dimensions.

Choose ConsoleLoginFailureCount and then choose Graph this metric only.

In the graph area at the top of the page, a small blue dot should appear. The dot indicates that a login failure was detected.

5. Check the alarm status and details in the CloudWatch console.

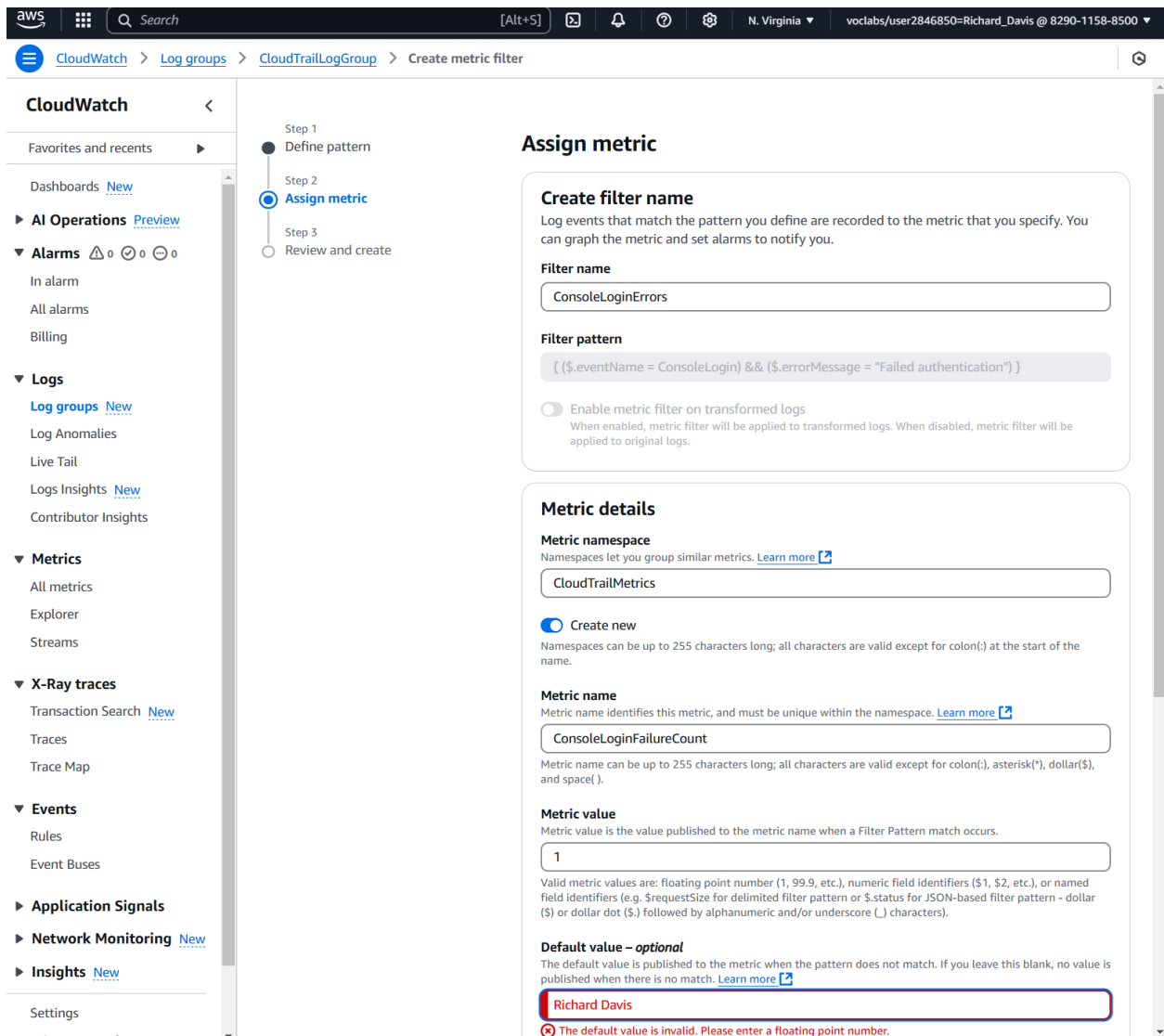
In the navigation pane, expand Alarms, and then choose All alarms.

The State for the FailedLogins alarm should be In alarm.

Note: If the alarm doesn't show this state, wait a minute or two. To refresh the page, choose the refresh icon.

Tip: To find out if the alarm was invoked recently, choose the link for the FailedLogins alarm name, and then choose the History tab.

6. Check the inbox of the email address that you subscribed to the SNS topic.



**CloudWatch**

Log groups > CloudTrailLogGroup > Create metric filter

**Assign metric**

**Create filter name**  
Log events that match the pattern you define are recorded to the metric that you specify. You can graph the metric and set alarms to notify you.

**Filter name**  
ConsoleLoginErrors

**Filter pattern**  
{ (\$.eventName = ConsoleLogin) && (\$.errorMessage = "Failed authentication") }

☐ Enable metric filter on transformed logs  
When enabled, metric filter will be applied to transformed logs. When disabled, metric filter will be applied to original logs.

**Metric details**

**Metric namespace**  
Namespaces let you group similar metrics. [Learn more](#)

CloudTrailMetrics

☒ **Create new**  
Namespaces can be up to 255 characters long; all characters are valid except for colon(:) at the start of the name.

**Metric name**  
Metric name identifies this metric, and must be unique within the namespace. [Learn more](#)

ConsoleLoginFailureCount

Metric name can be up to 255 characters long; all characters are valid except for colon(:), asterisk(\*), dollar(\$), and space( ).

**Metric value**  
Metric value is the value published to the metric name when a Filter Pattern match occurs.

1

Valid metric values are: floating point number (1, 99.9, etc.), numeric field identifiers (\$1, \$2, etc.), or named field identifiers (e.g. \$requestSize for delimited filter pattern or \$.status for JSON-based filter pattern - dollar (\$) or dollar dot (\$.) followed by alphanumeric and/or underscore (\_) characters).

**Default value – optional**  
The default value is published to the metric when the pattern does not match. If you leave this blank, no value is published when there is no match. [Learn more](#)

Richard Davis

**The default value is invalid. Please enter a floating point number.**

## How to install the CloudWatch Agent

1. Open the Systems Manager console.  
In the left navigation pane, choose Run Command under Node Management.
2. Choose Run a Command
3. 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.
6. At the bottom of the page, choose Run  
Wait for the Overall status to change to Success. You can occasionally choose refresh 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
8. Expand Step 2 - Command description and status.  
You should see the message: Successfully installed  
arn:aws:ssm:::package/AmazonCloudWatchAgent

The screenshot shows the AWS Systems Manager console interface. At the top, a green notification bar states: "Command ID: 9c820614-a5ec-4f3d-8757-9158e4c713c3 was successfully sent!". The left sidebar contains navigation links for various AWS Systems Manager features, including Quick Setup, Operations Management, Application Management, Change Management, and Node Management. The main content area displays the details for a specific command. The command ID is "9c820614-a5ec-4f3d-8757-9158e4c713c3". Below the command ID, there are buttons for "Cancel command", "Rerun", and "Copy to new". The "Command status" section shows the overall status as "Success" and the detailed status as "Success". It also provides counts for targets (1), completed (1), error (0), and delivery timed out (0). The "Targets and outputs" section includes a search bar and a table with one entry. The table has columns for Instance ID, Instance name, Status, Detailed Status, and Start time. The entry shows an instance ID of "i-0274ed005249f74fa", an instance name of "ip-10-0-0-72.ec2.internal", and a status of "Success".

**AWS Systems Manager** ×

Quick Setup

▼ **Operations Management**

- Explorer
- OpsCenter
- CloudWatch Dashboard
- Incident Manager

▼ **Application Management**

- Application Manager [New](#)
- AppConfig
- Parameter Store [New](#)

▼ **Change Management**

- Change Manager
- Automation [New](#)
- Change Calendar
- Maintenance Windows

▼ **Node Management**

- Fleet Manager
- Compliance
- Inventory
- Hybrid Activations
- Session Manager
- [Run Command](#)

Command ID: 9c820614-a5ec-4f3d-8757-9158e4c713c3 was successfully sent!

[AWS Systems Manager](#) > [Run Command](#) > Command ID: 9c820614-a5ec-4f3d-8757-9158e4c713c3

### Command ID: 9c820614-a5ec-4f3d-8757-9158e4c713c3

[Cancel command](#) [Rerun](#) [Copy to new](#)

#### Command status

Overall status  
✔ Success

Detailed status  
✔ Success

# targets  
1

# completed  
1

# error  
0

# delivery timed out  
0

#### Targets and outputs

[View output](#)

Search command invocations

|   | Instance ID         | Instance name             | Status    | Detailed Status | Start time               |
|---|---------------------|---------------------------|-----------|-----------------|--------------------------|
| ○ | i-0274ed005249f74fa | ip-10-0-0-72.ec2.internal | ✔ Success | ✔ Success       | Tue, 15 Oct 2024 18:40:5 |