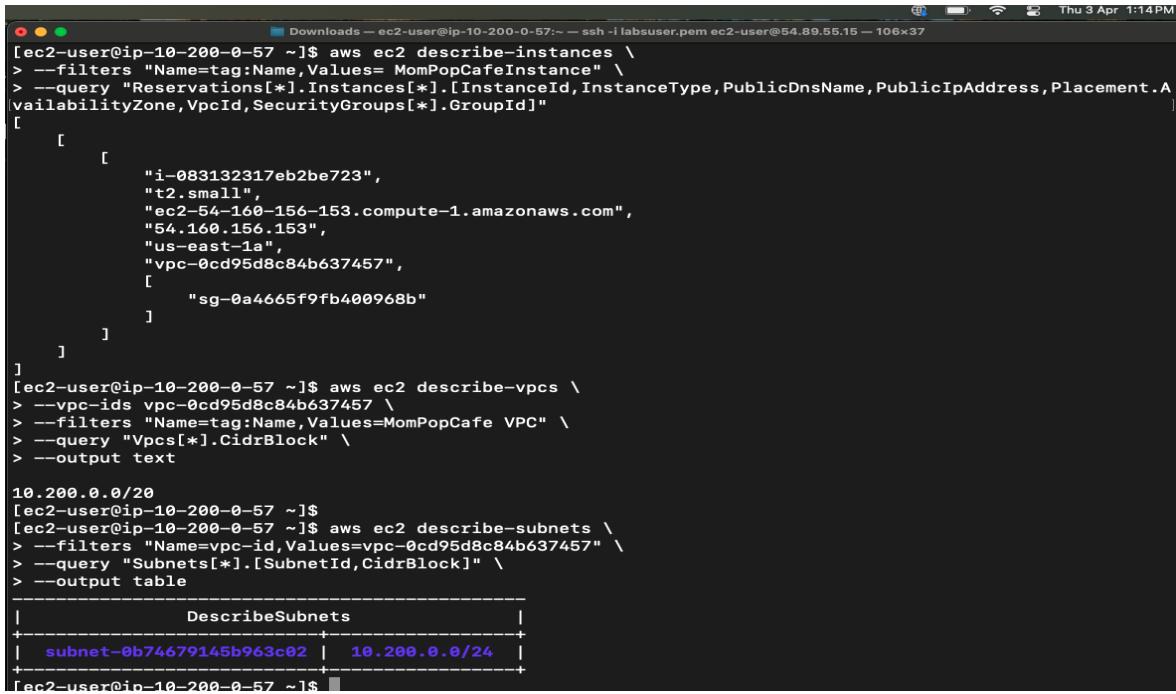


Module 6: Activity 6 - Migrate to Amazon RDS

Task 1.1: Connect to the CLI Host instance by using SSH

Fig 6.1: Running AWS CLI to Describe EC2 Instances



```
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-instances \
> --filters "Name=tag:Name,Values= MomPopCafeInstance" \
> --query "Reservations[*].Instances[*].[InstanceId,InstanceType,PublicDnsName,PublicIpAddress,Placement.AvailabilityZone,VpcId,SecurityGroups[*].GroupId]"
[
  [
    {
      "InstanceId": "i-083132317eb2be723",
      "InstanceType": "t2.small",
      "PublicDnsName": "ec2-54-160-156-153.compute-1.amazonaws.com",
      "PublicIpAddress": "54.160.156.153",
      "AvailabilityZone": "us-east-1a",
      "VpcId": "vpc-0cd95d8c84b637457",
      "SecurityGroups": [
        {
          "GroupId": "sg-0a4665f9fb400968b"
        }
      ]
    }
  ]
]
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-vpcs \
> --vpc-ids vpc-0cd95d8c84b637457 \
> --filters "Name=tag:Name,Values=MomPopCafe VPC" \
> --query "Vpcs[*].CidrBlock" \
> --output text
10.200.0.0/20
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-subnets \
> --filters "Name=vpc-id,Values=vpc-0cd95d8c84b637457" \
> --query "Subnets[*].[SubnetId,CidrBlock]" \
> --output table
+-----+-----+
| Subnet-0b74679145b963c02 | 10.200.0.0/24 |
+-----+-----+
[ec2-user@ip-10-200-0-57 ~]$
```

Note: This image shows the command used to describe the EC2 instances associated with the "MomPopCafeInstance" tag, returning key information about the instance like Instance ID, Type, DNS, IP, Availability Zone, and VPC ID.

Task 1.2: Gather current environment configuration information

Fig 6.2: Retrieving Subnet CIDR Block

```
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-vpcs \
> --filters "Name=tag:Name,Values=MomPopCafe VPC" \
> --query "Vpcs[*].CidrBlock" \
> --output text
10.200.0.0/20
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-subnets \
> --filters "Name=vpc-id,Values=vpc-0cd95d8c84b637457" \
> --query "Subnets[*].[SubnetId,CidrBlock]" \
> --output table
+-----+-----+
|      DescribeSubnets      |
+-----+-----+
| subnet-0b74679145b963c02 | 10.200.0.0/24 |
+-----+-----+
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-availability-zones \
> --filters "Name=region-name,Values=us-east-1" \
> --query "AvailabilityZones[*].ZoneName" \
> --output table
+-----+
| DescribeAvailabilityZones |
+-----+
| us-east-1a                |
| us-east-1b                |
| us-east-1c                |
| us-east-1d                |
| us-east-1e                |
| us-east-1f                |
+-----+
[ec2-user@ip-10-200-0-57 ~]$
```

Note: The command retrieves the CIDR block of the subnet associated with the "MomPopCafe" VPC to ensure proper networking and resource placement.

Fig 6.3: Retrieving Availability Zone Information

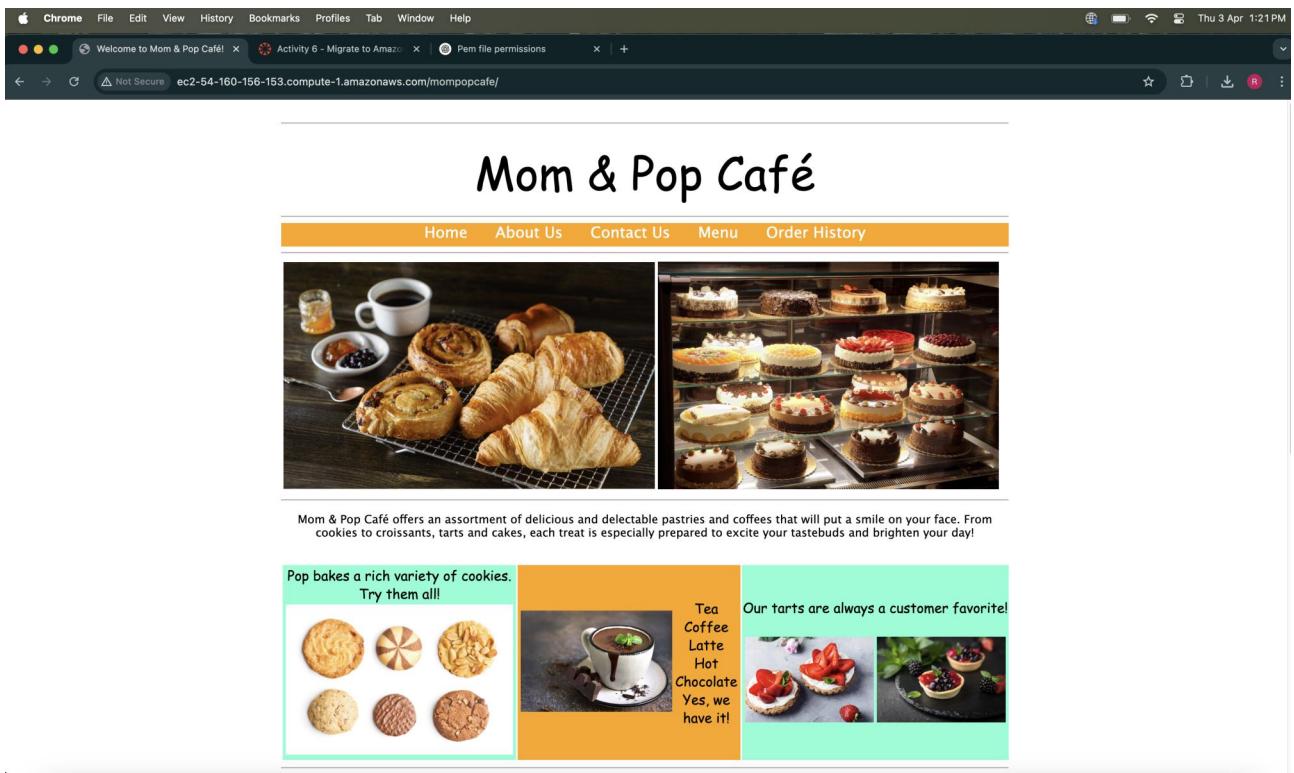
```
[ec2-user@ip-10-200-0-57 ~]$ aws ec2 describe-availability-zones \
> --filters "Name=region-name,Values=us-east-1" \
> --query "AvailabilityZones[*].ZoneName" \
> --output table
+-----+
| DescribeAvailabilityZones |
+-----+
| us-east-1a                |
| us-east-1b                |
| us-east-1c                |
| us-east-1d                |
| us-east-1e                |
| us-east-1f                |
+-----+
[ec2-user@ip-10-200-0-57 ~]$
```

Note:
The
CLI

command retrieves the list of Availability Zones in the "us-east-1" region, which is important for subnet distribution in the migration process.

Task 1.3: Create prerequisite components

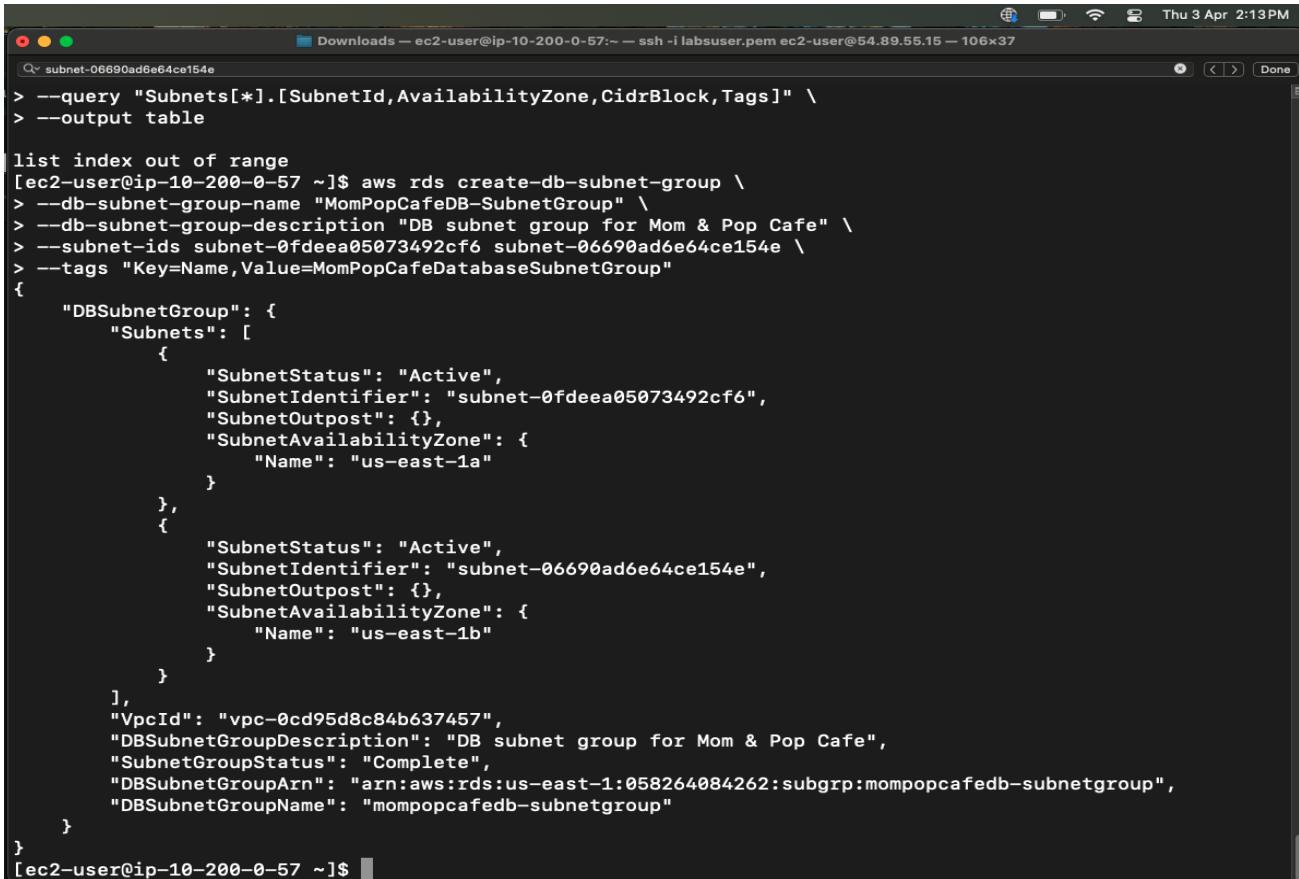
Fig 6.4: Web Server Test Page



Note: This image shows the initial test page of the Mom & Pop Café website hosted on the EC2 instance, verifying the site's availability before migration.

Task 1.4: Create the Amazon RDS MariaDB instance

Fig 6.5: Create the Amazon RDS MariaDB instance



```
> --query "Subnets[*].[SubnetId,AvailabilityZone,CidrBlock,Tags]" \
> --output table

list index out of range
[ec2-user@ip-10-200-0-57 ~]$ aws rds create-db-subnet-group \
> --db-subnet-group-name "MomPopCafeDB-SubnetGroup" \
> --db-subnet-group-description "DB subnet group for Mom & Pop Cafe" \
> --subnet-ids subnet-0fdeea05073492cf6 subnet-06690ad6e64ce154e \
> --tags "Key=Name,Value=MomPopCafeDatabaseSubnetGroup"
{
    "DBSubnetGroup": {
        "Subnets": [
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-0fdeea05073492cf6",
                "SubnetOutpost": {},
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                }
            },
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-06690ad6e64ce154e",
                "SubnetOutpost": {},
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1b"
                }
            }
        ],
        "VpcId": "vpc-0cd95d8c84b637457",
        "DBSubnetGroupDescription": "DB subnet group for Mom & Pop Cafe",
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:058264084262:subgrp:mompopcafedb-subnetgroup",
        "DBSubnetGroupName": "mompopcafedb-subnetgroup"
    }
}
[ec2-user@ip-10-200-0-57 ~]$
```

Note: The order confirmation page shows successful order processing, confirming that the application functionality is intact after the database migration to RDS. Configuring the AWS CLI environment to define the AWS account credentials, Region name, and output format to use with the command aws configure and enter the values which are asked to configure.

Fig 6.6: Order Confirmation Page



Mom & Pop Café

Home Menu Order History

Order Confirmation

Thank for your order! It will be available for pickup within 15 minutes. Your order number and details are shown below.

Order Number: 1 Date: 2025-04-03 Time: 13:23:31 Total Amount: \$25.00

Item	Price	Quantity	Amount
Croissant	\$1.50	3	\$4.50
Donut	\$1.00	3	\$3.00
Strawberry Tart	\$3.50	5	\$17.50

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

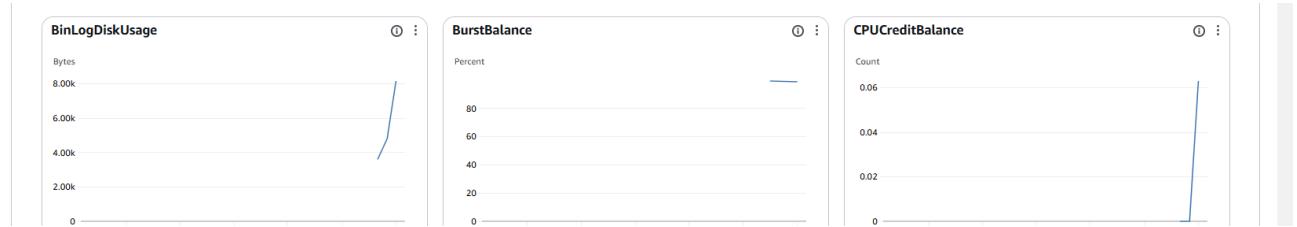
Note: The screenshot displays the order confirmation page for the "Mom & Pop Café" website. It shows a successful order with itemized details, demonstrating the continued functionality of the website post-migration.

Fig 6.7: SQL statement to retrieve the data in the product table

```
MariaDB [mom_pop_db]> select * from product;
+----+-----+-----+-----+
| id | product_name           | description                                | price
| product_group | image_url          |                                         |
+----+-----+-----+-----+
+----+-----+-----+-----+
| 1 | Croissant             | Fresh, buttery and fluffy... Simply delicious! | 1.50
|   1 | images/Croissants.jpg |                                         |
| 2 | Donut                  | We have more than half-a-dozen flavors!      | 1.00
|   1 | images/Donuts.jpg    |                                         |
| 3 | Chocolate Chip Cookie | Made with Swiss chocolate with a touch of Madagascar vanilla | 2.50
|   1 | images/Chocolate-Chip-Cookies.jpg |                                         |
| 4 | Muffin                 | Banana bread, blueberry, cranberry or apple | 3.00
|   1 | images/Muffins.jpg    |                                         |
| 5 | Strawberry Blueberry Tart | Bursting with the taste and aroma of fresh fruit | 3.50
|   1 | images/Strawberry-&-Blueberry-Tarts.jpg |                                         |
| 6 | Strawberry Tart       | Made with fresh ripe strawberries and a delicious whipped cream | 3.50
|   1 | images/Strawberry-Tart.jpg |                                         |
+----+-----+-----+-----+
```

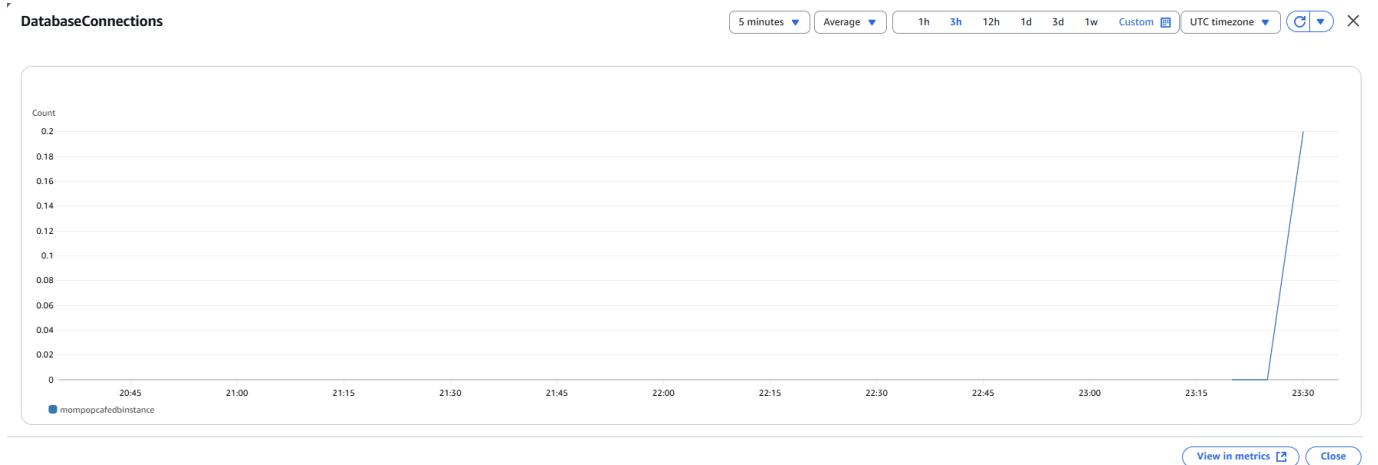
Note: checking whether migrated application data to Amazon RDS by first establishing an SSH connection to the application server. Since, creating a backup of the local database using `mysqldump`. Finally, restore this backup file to the newly created Amazon RDS instance, ensuring successful data transfer.

Fig 6.8: Monitor the Amazon RDS database



Note: In the above image, we can see the performance metrics of our DB, One of the benefits of using Amazon RDS is the ability to monitor the performance of a database instance. Amazon RDS automatically sends metrics to Amazon CloudWatch every minute for each active database.

Fig 6.9: Database connections monitoring



Note: DB Connections graph shows a line that indicates that 1 connection is in use. This connection was established by the interactive SQL session from the MomPopCafeInstance.

Fig 6.10: Module 6 Knowledge Check Results

The screenshot shows a browser window with the URL awsacademy.instructure.com/courses/107810/assignments/198235?module_item_id=10090934. The page title is "Module 6 Knowledge Check". On the left, there is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area displays the results of the knowledge check:

Module 6 knowledge check results

Your score: 100% (100 points)
Required score: 70% (70 points)

Result: Congratulations! You have completed this module.
To continue, choose **Next** in the lower-right corner.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Submission
Apr 3 at 9:07pm
Submission Details
Grade: 100 (100 pts possible)
Graded Anonymously: no
Comments:
No Comments

Note: This screenshot shows the successful completion of the Module 6 Knowledge Check, confirming the understanding of the migration process and Amazon RDS concepts.

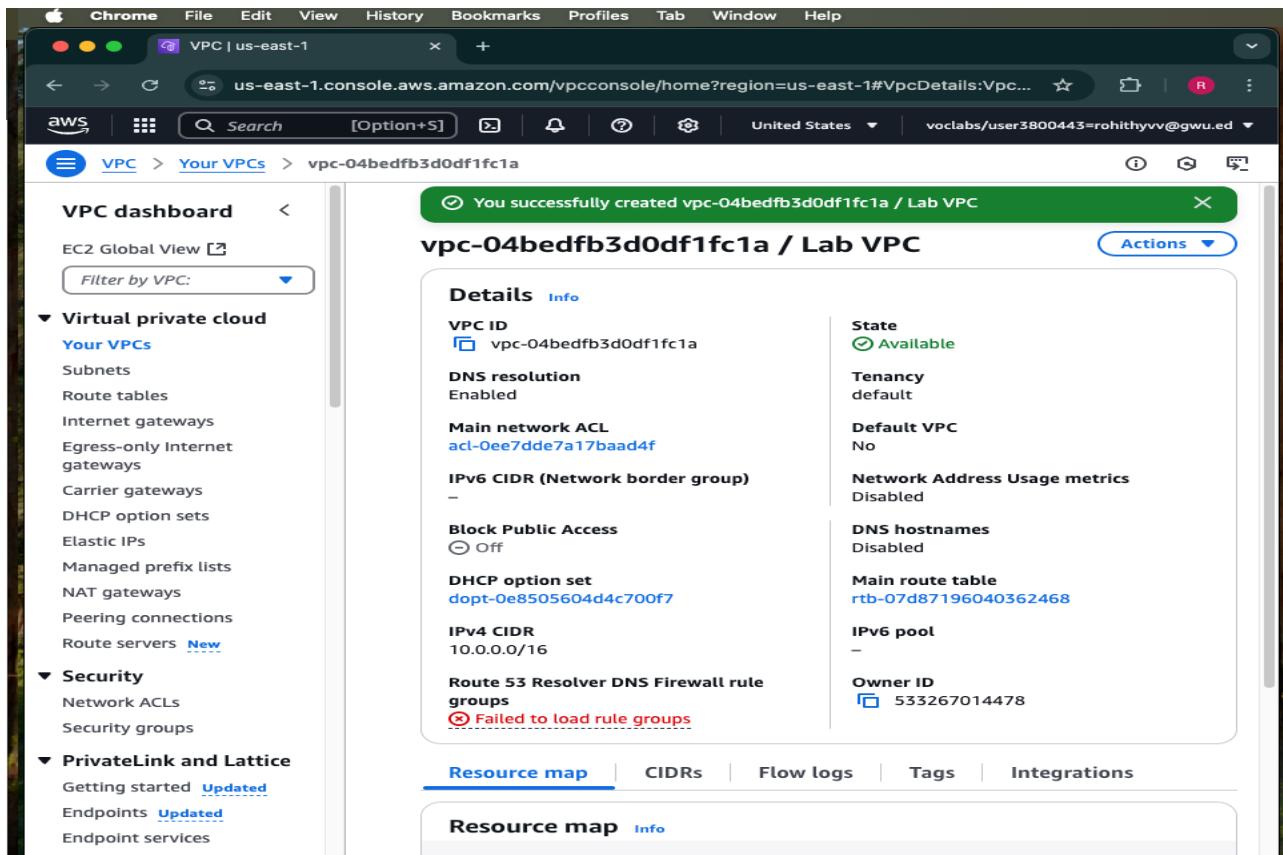
Summary:

- Amazon Aurora ensures high availability by replicating data across three Availability Zones, supporting automatic failover, read replicas, and using a distributed storage system.
- AWS Database Migration Service (DMS) helps migrate data securely and efficiently between databases.
- AWS Schema Conversion Tool (SCT) converts the source database schema to be compatible with the target database.
- Migrating a database involves converting the schema with SCT and then replicating the data using DMS.

Module 7: Activity 7 – Networking

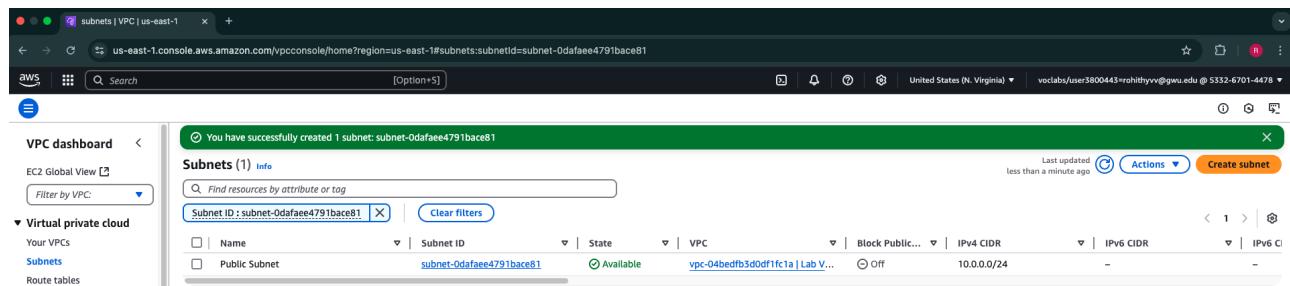
Lab 4 : Configure VPC

Fig 7.1 : Created Lab VPC for the application



Note: This image corresponds to the successful creation of a VPC in the AWS console, showing the details of the created Lab VPC, which is the starting step of the lab.

Fig 7.2 : Create Subnets



Note: This image reflects the creation of a public subnet within the "Lab VPC." It shows the subnet details like the ID and CIDR block (10.0.0.0/24). The subnet is marked as "Available," indicating its readiness for further configuration.

Fig 7.3 : Create an Internet Gateway

igw-08cac41539acd604e / Lab IGW

Details **Info**

Internet gateway ID: igw-08cac41539acd604e
State: Attached
Owner: 533267014478

Tags

Key	Value
Name	Lab IGW

Note: This image shows the successful creation of the Internet Gateway (IGW) for the Lab VPC. It is attached to the VPC, enabling internet connectivity for resources in the public subnet, which is crucial for enabling internet-bound traffic.

Fig 7.4 : Configure Route Tables

rtb-0c9a699f709f6b336 / Public Route Table

Details **Info**

Route table ID: rtb-0c9a699f709f6b336
Main: No
Owner ID: 533267014478

Explicit subnet associations: -

Edge associations: -

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Note: This image illustrates the creation of a public route table for the Lab VPC, which includes routes to direct internet-bound traffic (0.0.0.0/0) to the internet gateway. The route table setup is an essential step for configuring the public subnet.

Fig 7.5 : Launch a Bastion Server in the Public Subnet

The screenshot shows the AWS VPC console with the URL us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails. The main content area displays the details of a Public Route Table named "rtb-0c9a699f709f6b336". A green success message at the top states: "Updated routes for rtb-0c9a699f709f6b336 / Public Route Table successfully". The "Routes" tab is selected, showing two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-08cac4153...	Active	No
10.0.0.0/16	local	Active	No

The left sidebar includes sections for Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and PrivateLink and Lattice (Getting started, Endpoints, Endpoint services). The bottom right corner contains links for Privacy, Terms, and Cookie preferences.

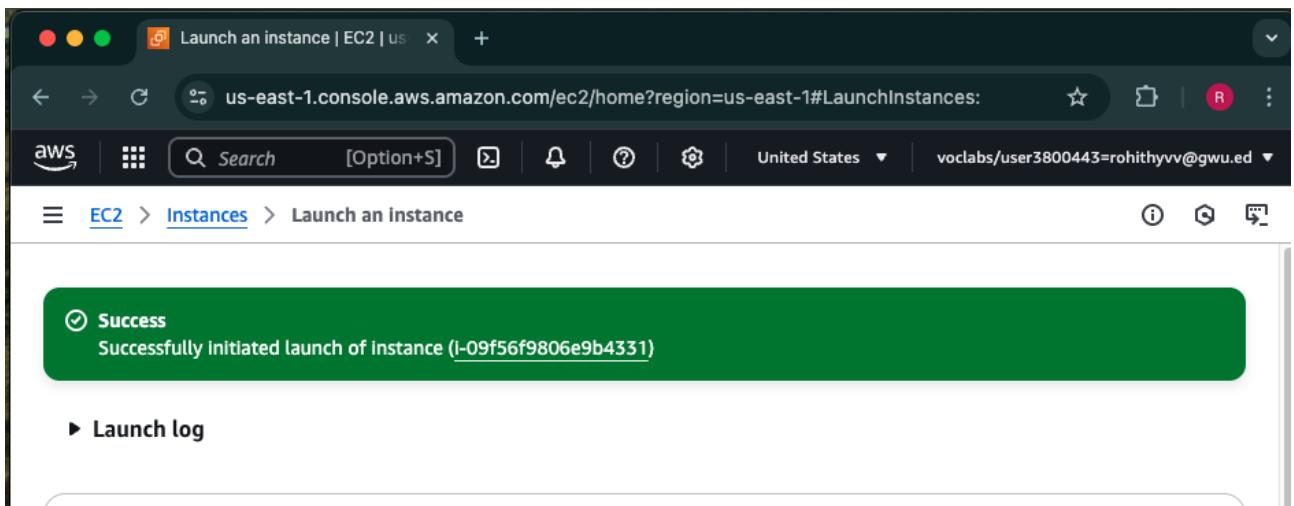
Note: This image shows the EC2 instance (Bastion Server) launched in the public subnet. It serves as a jump host for securely accessing instances in the private subnet. The instance has a public IP for direct SSH access.

Fig 7.6 : Create a NAT Gateway

This screenshot is identical to Fig 7.6, showing the AWS VPC console with the same Public Route Table and its associated routes. A green success message at the top states: "You have successfully updated subnet associations for rtb-0c9a699f709f6b336 / Public Route Table." The "Routes" tab is selected, showing the same two routes as in Fig 7.6.

Note: This image depicts the creation of the NAT Gateway, which is placed in the public subnet. The NAT Gateway enables instances in the private subnet to access the internet securely while preventing inbound access from the internet.

Fig 7.7 : Testing the Private Subnet



Note: This image shows the update to the private route table, configuring it to route outbound traffic to the NAT Gateway. It confirms that private subnet resources can reach the internet through the NAT Gateway.

Fig 7.7 : Launch Instance and Test Connectivity

The screenshot shows the AWS VPC NAT gateways details page for a specific NAT gateway. The main panel displays the following information:

- Details** section:
 - NAT gateway ID: nat-0df01e1b83f267c0f
 - NAT gateway ARN: arn:aws:ec2:us-east-1:533267014478:natgateway/nat-0df01e1b83f267c0f
 - VPC: vpc-04bedfb3d0df1fc1a / Lab VPC
 - State: Available
 - Primary private IPv4 address: 10.0.0.192
 - Created: Friday 4 April 2025 at 14:00:29 GMT-4
- Connectivity type**: Public
- Primary public IPv4 address**: 75.101.225.98
- Subnet**: subnet-0dafaae4791bace81 / Public Subnet
- State message**: Info
- Primary network interface ID**: eni-0a4c92a7d73110e7f
- Deleted**: -

The left sidebar shows the VPC dashboard and various VPC-related options like EC2 Global View, Virtual private cloud, Security, and PrivateLink and Lattice.

Note: This image illustrates the successful launch of an EC2 instance within the Lab VPC. The next steps include connecting to this instance and testing connectivity, validating that the network configuration works as expected.

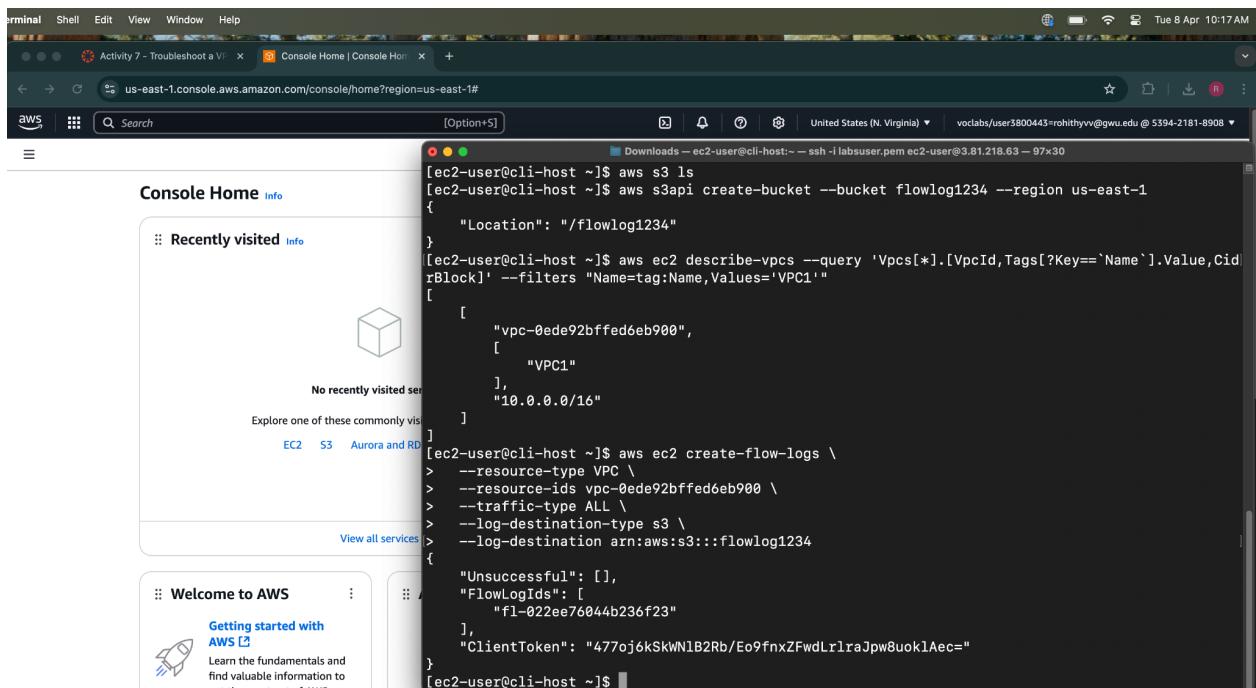
Activity 7 : Troubleshoot a VPC

Fig 7.8 : Configure the AWS CLI on the CLI Host EC2 Instance



Note : In the above image we discover the region in which the CLI Host instance is running and Update the AWS CLI software with the credentials.

Fig 7.8 : Confirming VPC Flow Logs Settings



Note: This screenshot captures the configuration interface for enabling VPC Flow Logs. It helps monitor and troubleshoot network traffic by logging information about IP traffic going to and from network interfaces in the specified VPC.

Fig 7.9 : Investigating EC2 Web Server Instance with AWS CLI

The screenshot shows a terminal window running on an AWS Lambda environment. The user has run the command `aws ec2 describe-instances --filter "Name=ip-address,Values=3.88.6.153" --query 'Reservations[*].Instances[*].[State,PrivateIpAddress,InstanceId,SecurityGroups,SubnetId,KeyName]'`. The output shows a single instance with the state "running", private IP address "3.88.6.153", and security group "vockey". The instance is in subnet "subnet-020e9b9cd6764eb77".

```

"ClientToken": "477oj6kSkWN1B2Rb/Eo9fnxZFwdLrlraJpw8uok1Aec"
[ec2-user@cli-host ~]$ http://3.88.6.153
-bash: http://3.88.6.153: No such file or directory
[ec2-user@cli-host ~]$ aws ec2 describe-instances \
> --filter "Name=ip-address,Values=3.88.6.153" \
> --query 'Reservations[*].Instances[*].[State,PrivateIpAddress,InstanceId,SecurityGroups,SubnetId,KeyName]'
[
  [
    {
      "Code": 16,
      "Name": "running"
    },
    "10.0.1.10",
    "i-0effd04438b77d7cc",
    [
      {
        "GroupName": "c148596a383148819890684t1w539421818908-WebSecurityGroup-naMVMC
        "GroupId": "sg-05cf7a8910d1d9b2c"
      }
    ],
    "subnet-020e9b9cd6764eb77",
    "vockey"
  ]
]
[ec2-user@cli-host ~]$ 

```

Note: This screenshot shows the use of the `aws ec2 describe-instances` command to fetch instance metadata for troubleshooting a connection issue to a public IP (3.88.6.153). The instance is confirmed as “running” in subnet vockey with security group details provided.

Fig 7.10 : Web Server Connectivity Verified with Public Access and Route Table Configuration

The screenshot shows a browser window displaying the message "Hello From Your Web Server!" at the IP address 3.88.6.153. Below the browser, a terminal window shows the creation of a route table and the execution of an `nmap` scan, which finds port 80 open on the host.

Browser Output:

Hello From Your Web Server!

Terminal Output:

```

"Ipv6Ranges": []
}
]
[ec2-user@cli-host ~]$ aws ec2 describe-route-tables \
> --filters "Name=association.subnet-id,Values=subnet-020e9b9cd6764eb77" \
> --query 'RouteTables[*].Routes'
[
  [
    {
      "GatewayId": "local",
      "DestinationCidrBlock": "10.0.0.0/16",
      "State": "active",
      "Origin": "CreateRouteTable"
    }
  ]
]
[ec2-user@cli-host ~]$ aws ec2 create-route \
> --route-table-id rtb-04499a3ca29ad1ae \
> --destination-cidr-block 0.0.0.0/0 \
> --gateway-id igw-04f574156f691b780
{
  "Return": true
}
[ec2-user@cli-host ~]$ nmap -Pn 3.88.6.153

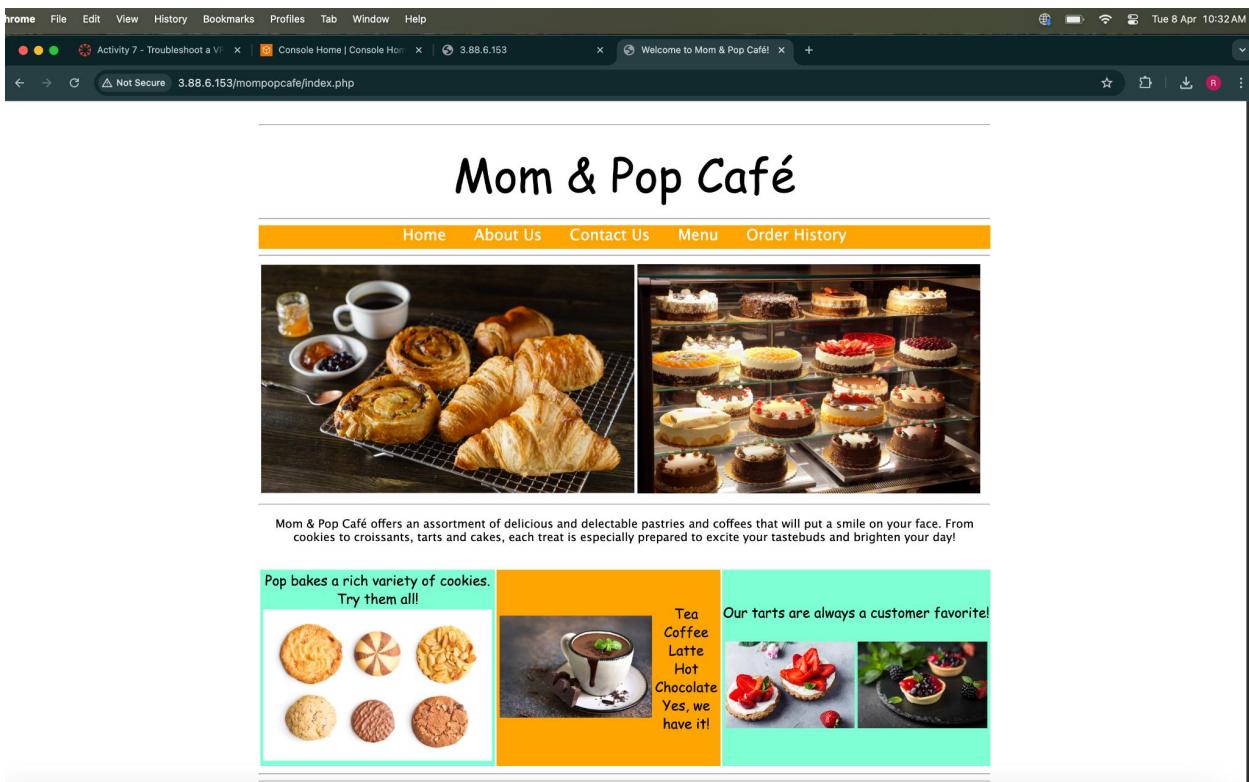
Starting Nmap 6.40 ( http://nmap.org ) at 2025-04-08 14:30 UTC
Nmap scan report for ec2-3-88-6-153.compute-1.amazonaws.com (3.88.6.153)
Host is up (0.0013s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 6.54 seconds
[ec2-user@cli-host ~]$ 

```

Note: This screenshot confirms successful access to a web server at IP 3.88.6.153, displaying the message “Hello From Your Web Server!” in a browser. Terminal commands show that a route to the internet was created via `igw-0457f4516f691b780`, and port 80 is open, verified using `nmap`.

Fig 7.11 : Successful Deployment of the Mom & Pop Café Web Application



Note: This screenshot confirms the successful launch of the full Mom & Pop Café website hosted at 3.88.6.153/mompopcafe/index.php. The page displays a complete frontend with navigation, pastries and cake visuals, and engaging marketing content—verifying that the EC2 web server and PHP application are functioning correctly.

Fig 7.12 : Module 7 Knowledge Check – Perfect Score Achievement

The screenshot shows a web browser window with the URL awsacademy.instructure.com/courses/107810/assignments/1198236/module_item_id=10090945. The page title is "ACOV1EN-LTI13-107810 > Assignments > Module 7 Knowledge Check". The main content area displays the results of the knowledge check. It shows "Your score: 100% (100 points)" and "Required score: 70% (70 points)". A message below states: "Result: Congratulations! You have completed this module. To continue, choose Next in the lower-right corner." On the right side, there is a sidebar with submission details: "Submission Apr 3 at 10:42pm", "Submission Details", "Grade: 100 (100 pts possible)", "Graded Anonymously: no", and "Comments: No Comments". The sidebar also includes a navigation menu with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help.

Note: This screenshot shows successful completion of the "Module 7 Knowledge Check" on AWS Academy with a perfect score of 100%. The required passing score was 70%. This confirms mastery of VPC configuration concepts, including subnets, route tables, and internet gateway setup. Summary:

- CIDR Blocks Define IP Ranges in VPC: When creating a VPC, a CIDR block determines the range of IP addresses available for the network and its subnets.
- Subnets Enable Segmentation and Isolation: VPC CIDR blocks are divided into smaller subnets for logical grouping, isolation, and traffic control, supporting both public and private subnets.
- NAT Devices Enable Internet Access for Private Subnets: NAT gateways allow instances in private subnets to access the internet securely while blocking incoming traffic.
- VPC Peering and VPN Enable Cross-Network Communication: VPC peering allows private IP routing between VPCs, while Site-to-Site VPN connects on-premises data centers to AWS securely over the internet.
- Route Tables and Security Groups Control Traffic: Route tables manage traffic flow at the subnet level, while security groups manage access permissions at the instance level.

Module 8: Activity 8 –

Fig 8.1 : Successful S3 Bucket Creation via AWS CLI after Credential Configuration

```
Activity 8 - Work with Amazon EC2 | Instances | EC2 | us-east-1 | + | Tue 8 Apr 11:38
awsacademy.instructure.com/courses/107810/modules/items/10090954

AWS Secret Access Key [None]: 7prjgWbCgEX/K//qh11rIGNHdG2Lkv/FSCRwFDxW
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-10-200-0-117 ~]$ aws s3 mb s3://mompopcafe-ryv123 --region us-east-1
make_bucket failed: s3://mompopcafe-ryv123 An error occurred (InvalidAccessKeyId) when calling the CreateBucket operation: The AWS Access Key Id you provided does not exist in our records.
[ec2-user@ip-10-200-0-117 ~]$ aws configure
AWS Access Key ID [*****5W05]: AKIA33TYBIKDJD5W05
AWS Secret Access Key [*****FDxW]: 7prjgWbCgEX/K//qh11rIGNHdG2Lkv/FSCRwFDxW
Default region name [us-east-1]: us-east-1
Default output format [json]: us-east-1
[ec2-user@ip-10-200-0-117 ~]$ aws s3 mb s3://mompopcafe-ryv123 --region us-east-1
make_bucket: mompopcafe-ryv123
[ec2-user@ip-10-200-0-117 ~]$ 
```

Credentials

Cloud Access

AWS CLI: Show

Cloud Labs

Remaining session time: 01:50:51(111 minutes)

Session started at: 2025-04-08T08:07:04-0700

Session to end at: 2025-04-08T10:17:05-0700

Accumulated lab time: 00:19:00 (19 minutes)

ips -- public:44.193.209.247, private:10.200.0.117

SSH key Show Download PEM Download PPK

AWS SSO Download URL

SecretKey 7prjgWbCgEX/K//qh11rIGNHdG2Lkv/FSCRwFDxW

CliHostPublicIP 44.193.209.247

AccessKey AKIA33TYBIKDJD5W05

Note: This image shows a successful S3 bucket creation (mompopcafe-ryv123) using the AWS CLI after resolving an initial "InvalidAccessKeyId" error. The user corrected the AWS access credentials via aws configure, enabling proper authentication for the operation.

Fig 8.2 : EC2 CLI Host Configuration and S3 Bucket Creation Validation

EC2 > Instances

Instances (1/1) Info

Last updated about 1 hour ago

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
CLI Host	i-0da0398a70f888a42	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-44-193-209-247.co...

i-0da0398a70f888a42 (CLI Host)

Details Status and alarms Monitoring Security Network

Instance summary Info

Instance ID i-0da0398a70f888a42

IPv4 address copied 44.193.209.247

IPv6 address -

Instance state Running

```
Downloads - ec2-user@ip-10-200-0-117:~$ ssh -l labuser.pem ec2-user@44.193.209.247 -80x24
AWS Secret Access Key [None]: 7prjgWbCgEX/K//qh11rIGNHdG2Lkv/FSCRwFDxW
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-10-200-0-117 ~]$ aws s3 mb s3://mompopcafe-ryv123 --region us-east-1
make_bucket failed: s3://mompopcafe-ryv123 An error occurred (InvalidAccessKeyId) when calling the CreateBucket operation: The AWS Access Key Id you provided does not exist in our records.
[ec2-user@ip-10-200-0-117 ~]$ aws configure
AWS Access Key ID [*****5W05]: AKIA33TYBIKDJD5W05
AWS Secret Access Key [*****FDxW]: 7prjgWbCgEX/K//qh11rIGNHdG2Lkv/FSCRwFDxW
Default region name [us-east-1]: us-east-1
Default output format [json]: us-east-1
[ec2-user@ip-10-200-0-117 ~]$ aws s3 mb s3://mompopcafe-ryv123 --region us-east-1
make_bucket: mompopcafe-ryv123
[ec2-user@ip-10-200-0-117 ~]$ 
```

Note: The screenshot displays the AWS EC2 dashboard with a running CLI Host instance and the terminal where AWS CLI commands were executed. After resolving an InvalidAccessKeyId error by configuring credentials, the user successfully created the S3 bucket mompopcafe-ryv123 in the us-east-1 region.

Fig 8.3 : Image Upload to S3 Bucket from EC2 CLI Host via AWS CLI Sync

The screenshot shows the AWS EC2 Instances page with a single instance named "CLI Host" listed. The terminal output in the bottom right corner of the browser window shows the user running the command "aws s3 sync ~/initial-images/ s3://mompopcafe-ryv123/images/" which successfully uploaded three files: "Donuts.jpg", "Strawberry-Tarts.jpg", and "Cup-of-Hot-Chocolate.jpg".

```

Downloads - ec2-user@ip-10-200-0-117:~ ssh -l labuser.pem ec2-user@44.193.209.247 - 80x24
[ec2-user@ip-10-200-0-117 ~]$ aws configure
AWS Access Key ID [*****5W05]: AKIA33TYBIKDJD5W05
AWS Secret Access Key [*****FDxW]: 7prjgWbCgEX/K/qh11rIGNHdG2Lkv/FSC
RwFDxW
Default region name [us-east-1]: us-east-1
Default output format [json]: us-east-1
[ec2-user@ip-10-200-0-117 ~]$ aws configure
AWS Access Key ID [*****5W05]: AKIA33TYBIKDJD5W05
AWS Secret Access Key [*****FDxW]: 7prjgWbCgEX/K/qh11rIGNHdG2Lkv/FSC
RwFDxW
Default region name [us-east-1]: us-east-1
Default output format [us-east-1]: json
[ec2-user@ip-10-200-0-117 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafe-ryv123 --region us-east-1
make_bucket: mompopcafe-ryv123
[ec2-user@ip-10-200-0-117 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafe-ryv123/images/Donuts.jpg
upload: initial-images/Donuts.jpg to s3://mompopcafe-ryv123/images/Donuts.jpg
upload: initial-images/Strawberry-Tarts.jpg
upload: initial-images/Cup-of-Hot-Chocolate.jpg to s3://mompopcafe-ryv123/images/Cup-of-Hot-Chocolate.jpg
[ec2-user@ip-10-200-0-117 ~]$ 

```

Note: This screenshot shows successful synchronization of local images from the EC2 instance to the S3 bucket mompopcafe-ryv123. The user used aws s3 sync to upload images such as Donuts.jpg, Strawberry-Tarts.jpg, and Cup-of-Hot-Chocolate.jpg under the images/ directory.

Fig 8.4 : S3 Bucket File Listing and Size Verification from EC2 Instance

The screenshot shows the AWS EC2 Instances page with a single instance named "CLI Host" listed. The terminal output in the bottom right corner of the browser window shows the user running the command "aws s3 sync ~/initial-images/ s3://mompopcafe-ryv123/images/" followed by "aws s3 ls s3://mompopcafe-ryv123/images/ --human-readable --summarize" to list the uploaded objects. The total size is 1.1 MiB.

```

Downloads - ec2-user@ip-10-200-0-117:~ ssh -l labuser.pem ec2-user@44.193.209.247 - 80x24
[ec2-user@ip-10-200-0-117 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafe-ryv123/images/
[ec2-user@ip-10-200-0-117 ~]$ aws s3 ls s3://mompopcafe-ryv123/images/ --human-readable --summarize
Total Objects: 3
  Total Size: 1.1 MiB
[ec2-user@ip-10-200-0-117 ~]$ 
[ec2-user@ip-10-200-0-117 ~]$ 

```

Note: The image illustrates the successful upload of three image files to the S3 bucket mompopcafe-ryv123/images/, followed by listing them using the aws s3 ls command. The total size of the uploaded objects is 1.1 MiB, confirming transfer integrity.

Fig 8.5 : Test the mediacouser permissions

The screenshot shows the AWS Management Console interface for Amazon S3. On the left, there's a sidebar with various options like General purpose buckets, Directory buckets, Table buckets, etc. The main area displays an account snapshot and a list of general purpose buckets. A new bucket, 'mompopcafe-ryv123', has been created and is listed in the table. The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The creation date is April 8, 2025, at 11:29:48 UTC-04:00.

Note: This screenshot confirms the successful creation of the S3 bucket mompopcafe-ryv123 in the AWS Management Console. The bucket is listed under the US East (N. Virginia) region (us-east-1), with the creation timestamp recorded as April 8, 2025.

Fig 8.6 : Verification of Uploaded Folder in S3 Bucket mompopcafe-ryv123

The screenshot shows the AWS Management Console interface for the S3 bucket 'mompopcafe-ryv123'. The left sidebar shows the bucket structure. The main page lists objects in the 'Objects' tab. A single folder named 'images/' is listed. The table includes columns for Name, Type, Last modified, Size, and Storage class. The 'Type' column shows 'Folder' for the 'images/' entry.

Note: The image shows the S3 console view of the mompopcafe-ryv123 bucket. It confirms that the images/ folder was successfully uploaded, indicating that the mediacouser has sufficient permissions to list and access objects within the bucket.

Fig 8.7 : Confirmation of Image Uploads in S3 Bucket Folder images/

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight [11]

AWS Marketplace for S3

images/

Objects [3] **Properties**

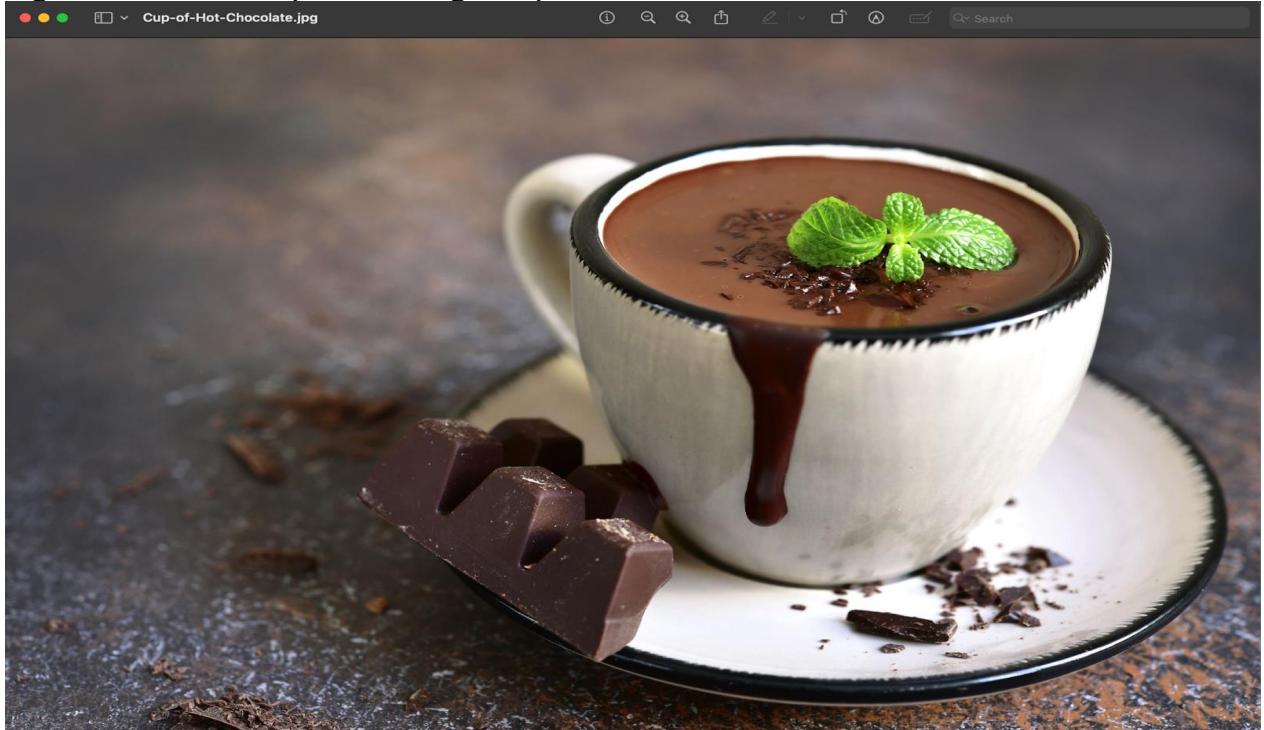
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix Show versions

Name	Type	Last modified	Size	Storage class
Cup-of-Hot-Chocolate.jpg	jpg	April 8, 2025, 12:32:20 (UTC-04:00)	308.7 KB	Standard
Donuts.jpg	jpg	April 8, 2025, 12:32:20 (UTC-04:00)	371.8 KB	Standard
Strawberry-Tarts.jpg	jpg	April 8, 2025, 12:32:20 (UTC-04:00)	468.0 KB	Standard

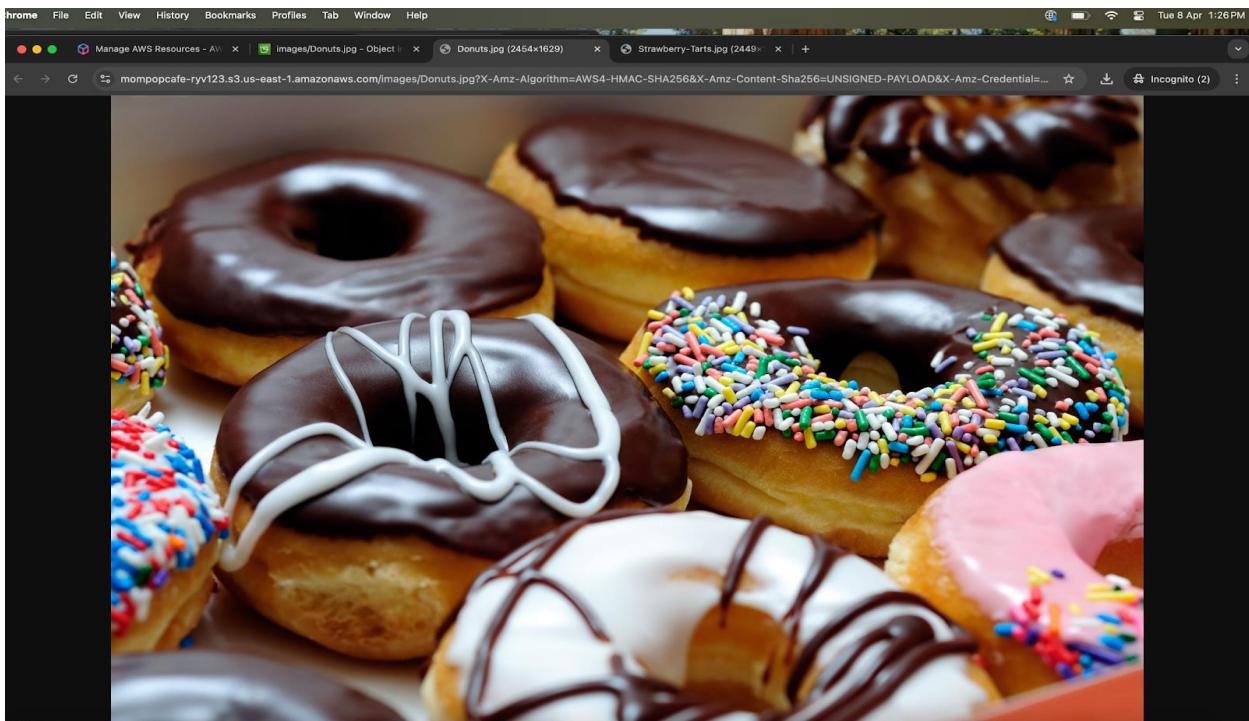
Note: This screenshot verifies that three image files—Cup-of-Hot-Chocolate.jpg, Donuts.jpg, and Strawberry-Tarts.jpg—have been successfully uploaded to the images/ folder in the S3 bucket mompopcafe-ryv123, with accurate timestamps and storage class settings.

Fig 8.8 : Preview of Uploaded Image: Cup of Hot Chocolate



Note: This image, titled Cup-of-Hot-Chocolate.jpg, was successfully uploaded to the S3 bucket mompopcafe-ryv123. It visually confirms the content integrity post-upload, showcasing a rich cup of hot chocolate garnished with mint and chocolate shavings.

Fig 8.8 : Preview of Uploaded Image: Donuts Assortment



Note: This image titled Donuts.jpg was uploaded to the S3 bucket mompopcafe-ryv123/images/. It displays a colorful assortment of donuts topped with chocolate glaze, sprinkles, and icing, confirming successful file accessibility and visual integrity after upload.

Fig 8.9 :Preview of Uploaded Image: Strawberry Tarts



Note: This image, named Strawberry-Tarts.jpg, showcases two freshly prepared tarts topped with whipped cream and sliced strawberries. Successfully uploaded to the mompopcafe-ryv123/images/S3 folder, it verifies image accessibility and visual quality after upload to the cloud.

Fig 8.10 :Successful Upload Confirmation of Donuts_1.jpg

The screenshot shows the AWS S3 console interface. At the top, there are several tabs including "Manage AWS Resources - AW", "Upload objects - S3 bucket", "Cup-of-Hot-Chocolate.jpg", "Donuts.jpg", "Strawberry-Tarts.jpg", and "New Incognito tab". The main content area displays a green success message: "Upload succeeded. For more information, see the Files and folders table." Below this, a summary table shows the destination as "s3://mompopcafe-ryv123/images/" with one file uploaded successfully (Succeeded) and zero files failed. A "Files and folders" tab is selected, showing a table with one item: "Donuts_1.jpg" (image/jpeg, 371.8 KB, Succeeded). A note at the bottom says: "After you navigate away from this page, the following information is no longer available."

Note: This screenshot confirms the successful upload of Donuts_1.jpg to the S3 bucket mompopcafe-ryv123/images/. The file uploaded without errors, achieving 100% success. The image size is 371.8 KB, and the confirmation assures data integrity post-upload.

Fig 8.11 :SNS Topic Overview – s3NotificationTopic

The screenshot shows the AWS SNS console. The left sidebar has navigation links for "Amazon SNS", "Dashboard", "Topics" (which is selected), "Subscriptions", and "Mobile" (with "Push notifications" and "Text messaging (SMS)"). The main content area is titled "s3NotificationTopic". It shows a "Details" section with "Name" (s3NotificationTopic), "Display name" (empty), "ARN" (arn:aws:sns:us-east-1:81522178522:s3NotificationTopic), and "Topic owner" (81522178522). Below this are tabs for "Subscriptions", "Access policy", "Data protection policy", "Delivery policy (HTTP/S)", "Delivery status logging", "Encryption", "Tags", and "Integrations". The "Subscriptions" tab is selected, showing a table with one row: "Subscriptions (0)". The table includes columns for "ID", "Endpoint", "Status", and "Protocol". A note below the table says: "No subscriptions found. You don't have any subscriptions to this topic." A "Create subscription" button is located at the bottom of the table.

Note: This image shows the SNS topic s3NotificationTopic created to receive S3 event notifications. It includes details like ARN, topic type, and owner ID, and confirms that no subscriptions have yet been added to the topic.

Fig 8.12 :SNS Topic Access Policy for S3 Event Publishing

The screenshot shows the AWS SNS 'Edit topic' page for an 's3NotificationTopic'. The 'Display name - optional' field contains 'My Topic'. The 'Access policy - optional' section shows a JSON editor with the following policy:

```

1 {
2     "Version": "2008-10-17",
3     "Id": "S3PublishPolicy",
4     "Statement": [
5         {
6             "Sid": "AllowPublishFromS3",
7             "Effect": "Allow",
8             "Principal": {
9                 "Service": "s3.amazonaws.com"
10            },
11            "Action": "SNS:Publish",
12            "Resource": "arn:aws:sns:us-east-1:815221785222:s3NotificationTopic",
13            "Condition": {
14                "ArnLike": {
15                    "aws:SourceArn": "arn:aws:s3:::mompopcafe-ryv123"
16                }
17            }
18        }
19    ]
20}

```

The 'Data protection policy - optional' section is collapsed.

Note: This screen captures the custom access policy configured to allow the S3 service to publish notifications to the SNS topic. The policy grants S3:Publish permission specifically from the mompopcafe-ryv123 bucket using aws:SourceArn

Fig 8.13 :Bucket Notification Configuration and Test Object Upload

The screenshot shows the AWS SNS 's3NotificationTopic' configuration page. A green success message says 'Topic s3NotificationTopic saved successfully.' Below it, the 'Subscriptions' tab is selected, showing 'Subscriptions (0)'. A terminal window is overlaid on the right side, displaying the command to put an object into an S3 bucket and the resulting JSON response, which includes the ETag and ServerSideEncryption metadata.

```

[ec2-user@ip-10-200-0-117 ~]$ aws s3api put-bucket-notification-configuration \
> --bucket mompopcafe-ryv123 \
> --notification-configuration file://s3EventNotification.json
[ec2-user@ip-10-200-0-117 ~]$ aws s3api put-object \
> --bucket mompopcafe-ryv123 \
> --key images/Caramel-Delight.jpg \
> --body ~/new-images/Caramel-Delight.jpg
{
    "ETag": "\"31ac30da619244b0ce786f106e4f3df7\"",
    "ServerSideEncryption": "AES256"
}
[ec2-user@ip-10-200-0-117 ~]$ 

```

Note: This composite image confirms the setup of an S3 bucket notification for prefix images/, followed by a successful object upload (Caramel-Delight.jpg) using the aws s3api put-object command, verifying notification trigger behavior with encryption metadata.

Fig 8.14 :Pending Email Subscription Confirmation for SNS Topic

The screenshot shows the AWS SNS console with a subscription details page. The URL in the address bar is `us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/subscription/arm:aws:sns:us-east-1:815221785222:s3NotificationTopic:2e6149e8-d862-404b-b74a-c2191b19ad51`. The page title is "Subscription: 2e6149e8-d862-404b-b74a-c2191b19ad51". A blue banner at the top says "New Feature" about High Throughput FIFO topics. The left sidebar shows "Amazon SNS" selected under "Subscriptions". The main content area has tabs for "Details", "Subscription filter policy" (which is selected), and "Redrive policy (dead-letter queue)". The "Details" tab shows the ARN, Endpoint (rohitv@gmail.com), Topic (s3NotificationTopic), and Subscription Principal (arn:aws:iam::815221785222:role/voclabs). The "Status" is listed as "Pending confirmation" with "EMAIL" as the protocol. Below the "Subscription filter policy" tab, it says "No filter policy configured for this subscription. To apply a filter policy, edit this subscription." with an "Edit" button.

Note: This image shows a pending SNS subscription created using the email protocol. The endpoint rohitv@gmail.com awaits confirmation to begin receiving notifications from the s3NotificationTopic. The subscription is linked to the IAM role vocabs.

Module 8: Storing and Archiving

Lab 5 Managing Storage

Fig 8.15 :Identifying EC2 Volume and Instance for Backup

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and Target Groups. The main area displays two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Processor	i-00dfc886cd50a2eac	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-80-229-229.co...	98.80.229.229	-
Command Host	i-02def3ec6a38ea976	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1a	ec2-3-80-253-233.com...	3.80.253.233	-

Details for the Processor instance:

- Instance ID:** i-00dfc886cd50a2eac
- Public IPv4 address:** 98.80.229.229 [open address]
- Private IP address:** 10.5.0.192
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-5-0-192.ec2.internal
- Instance type:** t2.micro

Note: This screenshot shows the use of AWS CLI to identify the Volume ID and Instance ID of an EC2 instance tagged as "Processor." These identifiers are essential for accurate snapshot creation.

Fig 8.16 :Stopping EC2 Instance and Creating a Snapshot

```
Downloads - ec2-user@ip-10-5-0-81:~ - ssh -i labsuser.pem ec2-user@3.80.253.233 - 80x24
  ~\_\_ #####_          Amazon Linux 2
  ~~ \_\#####\_
  ~~   \###|          AL2 End of Life is 2026-06-30.
  ~~   \#/  ___
  ~~   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/

-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file
or directory
[ec2-user@ip-10-5-0-81 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' \
> --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:Vol
umeId}'
{
  "VolumeId": "vol-0341839bed98b0bcb"
}
[ec2-user@ip-10-5-0-81 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' \
> --query 'Reservations[0].Instances[0].InstanceId'
"i-00dfc886cd50a2eac"
[ec2-user@ip-10-5-0-81 ~]$
```

Note: Note. After the snapshot is confirmed, the EC2 instance is restarted to resume service. This image reflects how AWS CLI confirms the instance's transition from stopped to running state.

Fig 8.17 :Listing All S3 Buckets in Console

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'Amazon S3', 'General purpose buckets', and 'Storage Lens'. The main area is titled 'files/'. It shows a table of objects with columns for Name, Type, Last modified, Size, and Storage class. There are three objects: file1.txt (txt, April 4, 2025, 29.6 KB, Standard), file2.txt (txt, April 4, 2025, 42.8 KB, Standard), and file3.txt (txt, April 4, 2025, 94.4 KB, Standard). Each object row has a 'Copy S3 URI' button.

Note. The AWS S3 Console lists all created buckets. This confirms the presence and region of the lab bucket used for file operations and version control.

Fig 8.18 : Inspecting Metadata of Uploaded Object

This screenshot shows the AWS S3 console for the 'file3.txt' object. The left sidebar includes 'Amazon S3', 'General purpose buckets', and 'Storage Lens'. The main panel is titled 'file3.txt' and has tabs for 'Properties', 'Permissions', and 'Versions'. Under 'Object overview', it shows the owner (awslabsc0w4072543t1648263812), AWS Region (US East (N. Virginia) us-east-1), last modified (April 4, 2025, 15:54:55 (UTC-04:00)), size (94.4 KB), type (txt), and key (files/file3.txt). To the right, detailed metadata is listed: S3 URI (s3://rohit-lab5-bucket/files/file3.txt), Amazon Resource Name (ARN) (arn:aws:s3:::rohit-lab5-bucket/files/file3.txt), Entity tag (ETag) (f491957be664c931c52fc1d39fffc709f), and Object URL (https://rohit-lab5-bucket.s3.us-east-1.amazonaws.com/files/file3.txt).

Note. Metadata view of file3.txt in the AWS Console. Shows S3 URI, object URL, ETag, and modification timestamps for referencing and API access.

Fig 8.19 : Downloading File from S3 to Local System

This screenshot shows a Mac OS X desktop with a dark-themed file browser window. The sidebar on the left shows 'Documents', 'Desktop', and 'Downloads'. The main area shows a file named 'file2.txt' with a blue status bar indicating its size (44 KB), type (Plain Text), and last modified time (Today at 4:11PM).

Note. Shows successful upload of file2.txt back into the bucket. This reflects how overwriting a file creates a new version in versioned S3 buckets.

Fig 8.20 : Uploading File from Local System

Upload succeeded
For more information, see the [Files and folders](#) table.

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary													
Destination s3://rohit-lab5-bucket/files/	Succeeded 1 file, 42.8 KB (100.00%)												
	Failed 0 files, 0 B (0%)												
Files and folders Configuration													
Files and folders (1 total, 42.8 KB)													
<table border="1"> <thead> <tr> <th>Name</th> <th>Folder</th> <th>Type</th> <th>Size</th> <th>Status</th> <th>Error</th> </tr> </thead> <tbody> <tr> <td>file2.txt</td> <td>-</td> <td>text/plain</td> <td>42.8 KB</td> <td>Succeeded</td> <td>-</td> </tr> </tbody> </table>		Name	Folder	Type	Size	Status	Error	file2.txt	-	text/plain	42.8 KB	Succeeded	-
Name	Folder	Type	Size	Status	Error								
file2.txt	-	text/plain	42.8 KB	Succeeded	-								

Note. Shows successful upload of file2.txt back into the bucket. This reflects how overwriting a file creates a new version in versioned S3 buckets.

Fig 8.21 : Listing S3 Objects Using AWS CLI

```
[ec2-user@ip-10-5-0-81 ~]$ aws s3 ls s3://rohit-lab5-bucket/files/
2025-04-04 20:14:27      43784 file2.txt
2025-04-04 19:54:55      96675 file3.txt
[ec2-user@ip-10-5-0-81 ~]$ ]
```

Note. AWS CLI lists current files in the S3 bucket. Verifies programmatic access matches UI state and confirms the synchronization of object management.

Fig 8.22 : Module 8 Knowledge Check – Perfect Score Achievement

Module 8 knowledge check results

Your score: 100% (100 points)

Required score: 70% (70 points)

Result: Congratulations! You have completed this module.

To continue, choose **Next** in the lower-right corner.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Note: The image confirms successful completion of the Module 8 Knowledge Check with a perfect score of 100%. The user exceeded the passing requirement of 70%, demonstrating full mastery of the content.

Summary:

- Amazon EBS Snapshots are incremental backups of EBS volumes, stored in Amazon S3, and only save changes after the first snapshot, helping reduce storage costs and improving backup efficiency.
- Amazon S3 is a scalable object storage service used for storing and accessing data online, offering different storage classes like Standard, Intelligent-Tiering, Infrequent Access, and Glacier for varying access needs.
- Instance Store provides temporary block-level storage physically attached to the host, ideal for fast-changing data like caches, buffers, and scratch space, but is not persistent.
- AWS Data Migration Tools such as AWS DataSync, Snowball, and Transfer for SFTP help move large volumes of data securely and efficiently between on-premises systems and AWS storage services.

Module 9: Monitoring and Security

Lab 6 : Monitoring Infrastructure

Fig 9.1 : CloudWatch Agent on the Web Server

The screenshot shows the AWS Systems Manager 'Run a command' interface. On the left, a sidebar lists various tools like Node Tools, Change Management Tools, and Application Tools. The main area is titled 'Run a command' and contains a 'Command document' section. A search bar at the top of this section has 'Document name prefix: Equals: AWS-ConfigureAWSPackage'. Below it is a table with one row selected: 'AWS-ConfigureAWSPackage' (Name), 'Amazon' (Owner), and 'Windows, Linux, MacOS' (Platform types). The 'Description' section explains that it installs or uninstalls a Distributor package. The 'Document version' dropdown is set to '1 (Default)'. The 'Command parameters' section shows an 'Action' dropdown set to 'Install'.

This screenshot shows the initiation of a Run Command using AWS-ConfigureAWSPackage to install or uninstall a Distributor package. The document targets EC2 instances to configure packages such as CloudWatch Agent, helping streamline monitoring or software configuration tasks.

Fig 9.2 : Command Execution Status - AWS Systems Manager

The screenshot shows the AWS Systems Manager 'Command ID: 98878713-5ab1-4db1-893f-c3c222a3a2a4' status page. The 'Command status' section shows an overall success status with 1 target completed and 0 errors. The 'Targets and outputs' section lists one instance: 'ip-10-0-0-48.ec2.internal' with a status of 'Success'. Below this are sections for 'CloudWatch alarm', 'Command description', and 'Command parameters'.

Note: The command AWS-ConfigureAWSPackage was successfully executed on the target EC2 instance. Both the overall and detailed status show success, confirming that the package (likely CloudWatch Agent) was correctly installed without any errors or delivery timeouts.

Fig 9.3 : Command Execution Status - AWS Systems Manager

The screenshot shows the AWS Systems Manager Parameter Store interface. A green success message at the top states "Create parameter request succeeded! View details". Below it, the "My parameters" section lists a single parameter named "Monitor-Web-Server" with a value of "String" and a creation date of "Tue, 08 Apr 2025 21:31:25 GMT".

Note: The parameter Monitor-Web-Server was created successfully under the Parameter Store. This standard string parameter can be used for storing configuration values securely and centrally to manage and reference in automation tasks like EC2 monitoring or script execution.

Fig 9.4 : Parameter Creation in AWS Systems Manager Parameter Store

The screenshot shows the AWS Systems Manager Run Command results for a command with ID "5c521c83-c11b-4517-b676-5a220e6541ce". The "Command status" table shows 1 target, 1 completed, 0 errors, and 0 delivery timed out. The "Targets and outputs" table shows one instance with a success status. The "CloudWatch alarm" and "Command description" sections are collapsed.

Overall status	Detailed status	# targets	# completed	# error	# delivery timed out
Success	Success	1	1	0	0

Instance ID	Instance name	Status	Detailed Status	Start time	Finish time
i-0c8c28587c4cf82b5	ip-10-0-0-48.ec2.internal	Success	Success	Tue, 08 Apr 2025 21:34:03 GMT	Tue, 08 Apr 2025 21:34:03 GMT

Note: The parameter Monitor-Web-Server was created successfully under the Parameter Store. This standard string parameter can be used for storing configuration values securely and centrally to manage and reference in automation tasks like EC2 monitoring or script execution.

Fig 9.5 :CloudWatch Log Event Showing 403 Error

The screenshot shows the AWS CloudWatch Log Events interface. On the left, a sidebar navigation includes sections for AI Operations, Alarms, Metrics, X-Ray traces, Events, Application Signals, Network Monitoring, and Insights. The main content area is titled "Log events" and displays a single log entry from the "HttpAccessLog" log group. The log entry timestamp is 2025-04-08T21:43:39.732Z, and the message content is: "35.82.31.94 - - [08/Apr/2025:21:43:35 +0000] \"GET / HTTP/1.1\" 403 3630 \"\" \"Mozilla/5.0 (Linux; Android 8.0.0; moto e5 cruise Build/OCPS27... No newer events at this moment. Auto retry paused. Resume".

Note: This CloudWatch log entry shows a 403 Forbidden error response captured in the log group HttpAccessLog. The request was denied, possibly due to permission issues, and the log confirms monitoring is active and capturing HTTP access errors.

Fig 9.5 : CloudWatch Metric Filter Creation for 403 Errors

The screenshot shows the "Create metric filter" wizard, Step 3: Review and create. It displays two tabs: Step 1: Pattern and Step 2: Metric. In Step 1, the pattern is defined as "[ip, id, user, timestamp, request, status_code=403, size]". In Step 2, the metric is assigned with the name "403Errors", namespace "LogMetrics", value "1", and unit "-". The "Create metric filter" button is highlighted in orange at the bottom right.

Note: This screen finalizes the creation of a CloudWatch metric filter for detecting HTTP 403 errors. The filter pattern extracts logs with status_code=403, assigning them to the metric 403Errors in namespace LogMetrics, enabling alarm triggers or monitoring dashboards.

Fig 9.6: Web Server Test Page



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server instal

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to ["webmaster@example.com"](mailto:webmaster@example.com).

If you are the website administrator:

You may now add content to the directory `/var/www/html`. Your website will see this page, and not your content. To prepare instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by Apache.



Note: we paste the WebServerIP copied in the browser to see a web server Test Page, now will generate log data by attempting to access a page that does not exist.

Fig 9.7 : Creating a Metric Filter in CloudWatch Logs

Metric filter "404Errors" has been created.

Metric filters (1)

404Errors

Filter pattern
[ip, id, user, timestamp, request, status_code=404, size]

Metric
LogMetrics / 404Errors

Metric value
1

Default value
-

Applied on transformed logs
-

Unit
-

Dimensions
-

Note :

Note: We will configure a Filter to identify 404 Errors in the log file. This would normally be an indication that the web server is generating invalid links that users are clicking.

Fig 9.8: Alarm Triggered

Some subscriptions are pending confirmation
Amazon SNS doesn't send messages to an endpoint until the subscription is confirmed

Alarms (1)

Name	State	Last state update (UTC)	Conditions	Actions
404Errors	In alarm	2025-03-15 02:19:53	404Errors >= 5 for 1 datapoints within 1 minute	Actions enabled Warning

Note:

Note; Attempting to go to pages that do not exist by adding a page name after the IP address, graph shown on the CloudWatch page should turn red to indicate that it is in the ALARM state.

Activity 9 : Working with AWS CloudTrail

Fig 9.9: EC2 Instance Running - Cafe Web Server

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Instances (selected), Images, Elastic Block Store, Network & Security, and more. The main content area shows a table of instances. One instance is selected: "Cafe Web Server" (i-08f491347839aa4c4). The instance details are shown in a modal window. Key details include:

- Instance ID:** i-08f491347839aa4c4
- Public IPv4 address:** 3.236.189.103
- Instance state:** Running
- Private IP address:** 10.0.0.241
- Private IP DNS name (IPv4 only):** ip-10-0-0-241.ec2.internal
- Instance type:** t2.micro
- VPC ID:** ec2-3-236-189-103.compute-1.amazonaws.com

Note: The EC2 instance named "Cafe Web Server" is in a running state in the us-east-1a availability zone. It's a t2.micro instance with a public IPv4 address, allowing you to host web services accessible via the internet.

Fig 9.10: Editing Inbound Rules for SSH Access

The screenshot shows the AWS Security Groups page. A specific security group rule is being edited. The rule details are as follows:

- Type:** SSH
- Protocol:** TCP
- Port range:** 22
- Source:** My IP (with an additional entry for 69.143.0.226/32)
- Description:** (empty)

Note: This screen shows modification of a security group to allow SSH access. The rule allows TCP traffic on port 22 from a specific IP address (69.143.0.226/32), securing the instance by limiting remote access to only the user's machine.

Fig 9.11: Security Group Rule Successfully Modified

The screenshot shows the AWS EC2 Security Groups console. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-0ec26ce16bf789c3c | WebSecurityGroup) Details". Below this, the security group details are shown: Security group name (WebSecurityGroup), Security group ID (sg-0ec26ce16bf789c3c), Description (Enable HTTP and SSH ingress), Owner (020733825915), VPC ID (vpc-03775bf342aa87e17). The Inbound rules section shows one rule: sgr-09b5e4eeded0bb56, IPv4, SSH, TCP, port 22, source 69.143.0.226/32.

Note: Inbound rules for the security group WebSecurityGroup were successfully updated. SSH access (TCP port 22) is now restricted to a single IP, enhancing security by preventing unrestricted remote login attempts from unauthorized networks.

Fig 9.12: CloudTrail Trail Setup and Logging Status

The screenshot shows the AWS CloudTrail Trails console. A blue info message at the top says: "What's new: Strengthen your data perimeter and implement better detective controls for your VPC endpoints by enabling Network activity events on your Trail or CloudTrail Lake. Learn more". The Trails table lists one trail named "Monitor": Name (Monitor), Home region (US East (N. Virginia)), Multi-region trail (Yes), Insights (Disabled), Organization trail (No), S3 bucket (aws-cloudtrail-logs-020733825915-d57b25fb), Log file prefix (-), CloudWatch Logs log group (-), and Status (Logging).

Note: A CloudTrail trail named "Monitor" is configured to capture account activity in the US East (N. Virginia) region. The trail is logging successfully and is set up as a multi-region trail, storing logs in the specified S3 bucket.

Fig 9.13: Website has been hacked with an random image is uploaded in the webpage of mompopcafe

Mom & Pop Café

Home About Us Contact Us Menu Order History



Note: Notice that the website has been hacked. As we enabled CloudTrail before this happened, CloudTrail can give valuable information about what users have been doing in my account.

Fig 9.14 : Deleting inbound rule which hacker created

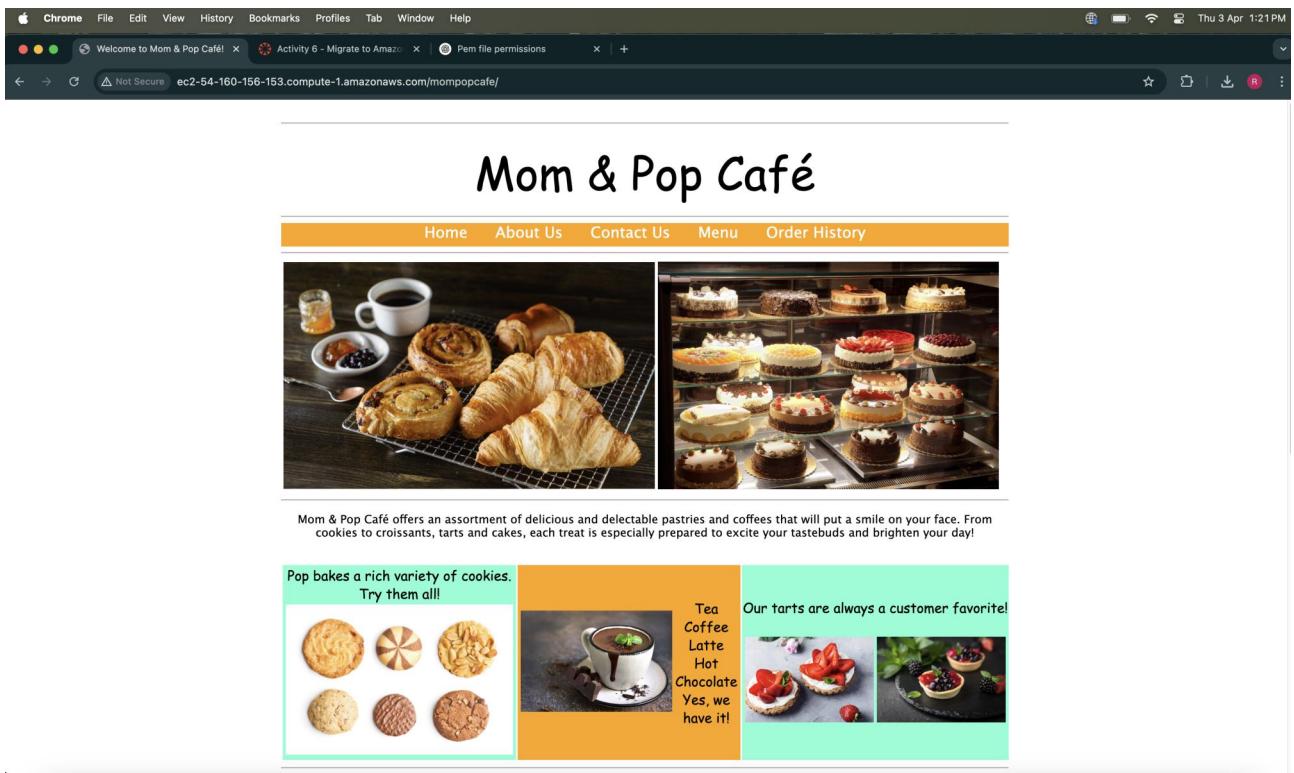
[Edit inbound rules](#) Info

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Delete
sgr-08a524e4a4b1087e5	SSH	TCP	22	Custom	73.180.168.123/32	Delete
sgr-0f1db300b5ca0f8d7	SSH	TCP	22	Custom	0.0.0.0/0	Delete
sgr-036311b83849e335c	HTTP	TCP	80	Custom	0.0.0.0/0	Delete

Note: With the WebSecurityGroup selected, in **Inbound rules** should delete the inbound rule that allows port 22 access from 0.0.0.0/0 (the one the hacker created)

Fig 9.15 : After deleting the image and fixing the website



Note: After deleting the inbound rule rule which hacker created and corrupt files uploaded by him, when we refresh our mompopcafe website it should look fine.

Fig 9.16 : Delete the AWS hacker user

User Name	Path	Group	Last Activity	MFA	Password Age	Console Last Sign-in	Access Key ID
awsstudent	/		-	-	-	-	Active - AKIAZQAVNM...

Note: The hacker not only accessed the EC2 instance hosting the website, but they also managed to run an AWS CLI command that opened port 22 in the security group to the entire internet. In this step we removed the chaos IAM user from the account.

Fig 9.17 : End of module 9

The screenshot shows a browser window with the URL awsacademy.instructure.com/courses/107810/assignments/1196238?module_item_id=10090965. The page title is "Module 9 Knowledge Check". On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area displays the results of the knowledge check:

Module 9 knowledge check results

Your score: 100% (100 points)
Required score: 70% (70 points)

Result: Congratulations! You have completed this module.
To continue, choose **Next** in the lower-right corner.

At the bottom right, it says "Submission" with the date "Apr 3 at 9:36pm", "Submission Details", "Grade: 100 (100 pts possible)", "Graded Anonymously: no", and "Comments: No Comments".

Note: This image shows the completion of module 9

Summary:

- Amazon CloudWatch provides real-time monitoring of AWS resources by collecting metrics, logs, and events. It supports alarms and automated actions based on thresholds or scheduled rules.
- CloudWatch Logs help collect and store log data from AWS services, EC2 instances, and custom sources, allowing for search, alerting, and troubleshooting.
- AWS CloudTrail records all account activity related to actions taken through the AWS Management Console, SDKs, CLI, and other services, storing the logs in S3 for auditing and compliance purposes.
- Amazon Athena allows you to run SQL queries directly on data stored in Amazon S3, making it useful for analyzing service logs like CloudTrail and VPC Flow Logs without data movement.
- AWS Config continuously records resource configurations and evaluates them against desired states, supporting compliance auditing, change tracking, and security analysis.

Module 11 : Creating Automated and Repeatable Deployments

Lab 8 : Automation with CloudFormation

Fig 11.1: Deploy a CloudFormation Stack

The screenshot shows the AWS CloudFormation console interface. On the left, the navigation sidebar includes sections for CloudFormation (Stacks, Stack details, Drifts, StackSets, Exports), Infrastructure Composer (laC generator), Registry (Public extensions, Activated extensions, Publisher), Spotlight, and Feedback. A prominent blue callout box in the center-left area introduces the new Hooks experience and capabilities, mentioning Lambda functions or Guard domain-specific language (DSL). The main content area is titled 'Lab' and shows the 'Events' tab selected. It displays a table of events for the 'Lab' stack, with 28 entries. The first entry is 'CREATE_COMPLETE'. Subsequent entries show the creation of various resources like PublicRouteTableAssociation, PublicRoute, and PublicRouteTableAssociation, all in 'CREATE_COMPLETE' status. Other entries are in 'CREATE_IN_PROGRESS' status, with notes indicating 'Resource creation Initiated'. The table columns include Timestamp, Logical ID, Status, Detailed status, and Status reason.

Timestamp	Logical ID	Status	Detailed status	Status reason
2025-04-04 10:46:39 UTC-0400	Lab	CREATE_COMPLETE	-	-
2025-04-04 10:46:37 UTC-0400	PublicRouteTableAssociation	CREATE_COMPLETE	-	-
2025-04-04 10:46:37 UTC-0400	PublicRoute	CREATE_COMPLETE	-	-
2025-04-04 10:46:37 UTC-0400	PublicRouteTableAssociation	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-04-04 10:46:36 UTC-0400	PublicRoute	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-04-04 10:46:35 UTC-0400	PublicRouteTableAssociation	CREATE_IN_PROGRESS	-	-
2025-04-04 10:46:35 UTC-0400	PublicRoute	CREATE_IN_PROGRESS	-	-

Note: This screenshot shows the successful creation of a CloudFormation stack named "Lab." All stack components, including public route tables and VPC resources, have reached the CREATE_COMPLETE status, confirming that infrastructure deployment was successful and error-free.

Fig 11.2: CloudFormation Resource List – Post Stack Creation

The screenshot shows the AWS CloudFormation console interface, similar to Fig 11.1. The left sidebar includes sections for CloudFormation (Stacks, Stack details, Drifts, StackSets, Exports), Infrastructure Composer (laC generator), Registry (Public extensions, Activated extensions, Publisher), Spotlight, and Feedback. The main content area shows the 'Resources' tab selected under the 'Lab' stack. It displays a table of resources with 9 items. All resources are in 'CREATE_COMPLETE' status. The table columns include Logical ID, Physical ID, Type, Status, and Module. Resources listed include AppSecurityGroup, IGW, LabVPC, MyS3Bucket, PublicRoute, PublicRouteTable, PublicRouteTableAssociation, and PublicSubnet.

Logical ID	Physical ID	Type	Status	Module
AppSecurityGroup	sg-0ba4d303a437cf0c	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
IGW	igw-0777fe96ebd916fc	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
LabVPC	vpc-01decf746f1579d2a	AWS::EC2::VPC	CREATE_COMPLETE	-
MyS3Bucket	lab-my3bucket-4k8p5puce800	AWS::S3::Bucket	CREATE_COMPLETE	-
PublicRoute	rtb-00bf91f43820aee2b[0..0/0]	AWS::EC2::Route	CREATE_COMPLETE	-
PublicRouteTable	rtb-00bf91f43820aee2b	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PublicRouteTableAssociation	rtbasoc-0f20c3df87f7298c4	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicSubnet	subnet-0729376b0a1e5ea9	AWS::EC2::Subnet	CREATE_COMPLETE	-

Note: This screen displays the resources created as part of the "Lab" stack. Items include a security group, internet gateway, VPC, S3 bucket, route table, subnet, and associations—all with CREATE_COMPLETE status, indicating a successful provisioning of AWS infrastructure.

Fig 11.3: CloudFormation Stack Deletion – Resources Cleaned Up

The screenshot shows the AWS CloudFormation console with the URL <https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/resources?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A47112556170%3Astack%2FLab%2F>. The left sidebar shows the navigation path: CloudFormation > Stacks > Lab. The main area displays the 'Stacks (0)' section, indicating 'No stacks' and a 'Create stack' button. On the right, the 'Resources' tab is selected, showing a table of 9 resources that have been deleted successfully:

Logical ID	Physical ID	Type	Status	Module
AppSecurityGroup	sg-0ba4d303a437cf0c	AWS::EC2::SecurityGroup	DELETE_COMPLETE	-
IGW	igw-0777fe96ebd916fcba	AWS::EC2::InternetGateway	DELETE_COMPLETE	-
LabVPC	vpc-01decf746f1579d2a	AWS::EC2::VPC	DELETE_COMPLETE	-
MyS3Bucket	lab-mys3bucket-4k8p5puvx8oo	AWS::S3::Bucket	DELETE_COMPLETE	-
PublicRoute	rtb-00bf91f43820aee2b 0.0.0/0	AWS::EC2::Route	DELETE_COMPLETE	-
PublicRouteTable	rtb-00bf91f43820aee2b	AWS::EC2::RouteTable	DELETE_COMPLETE	-
PublicRouteTableAssociation	rtbassoc-0f20c3df87f7298c4	AWS::EC2::SubnetRouteTableAssociation	DELETE_COMPLETE	-
PublicSubnet	subnet-0729376b0a1e5ea9	AWS::EC2::Subnet	DELETE_COMPLETE	-
VPCtoIGWConnection	IGWVpc-01decf746f1579d2a	AWS::EC2::VPCCGatewayAttachment	DELETE_COMPLETE	-

Note: The confirms that all resources from the "Lab" CloudFormation stack, including VPC, S3 bucket, and internet gateway, have been deleted successfully (DELETE_COMPLETE status). The stack is no longer listed, ensuring clean deprovisioning of infrastructure.

Activity 11: Troubleshoot CloudFormation

Fig 11.5 : Practice querying JSON-formatted data by using JMESPath



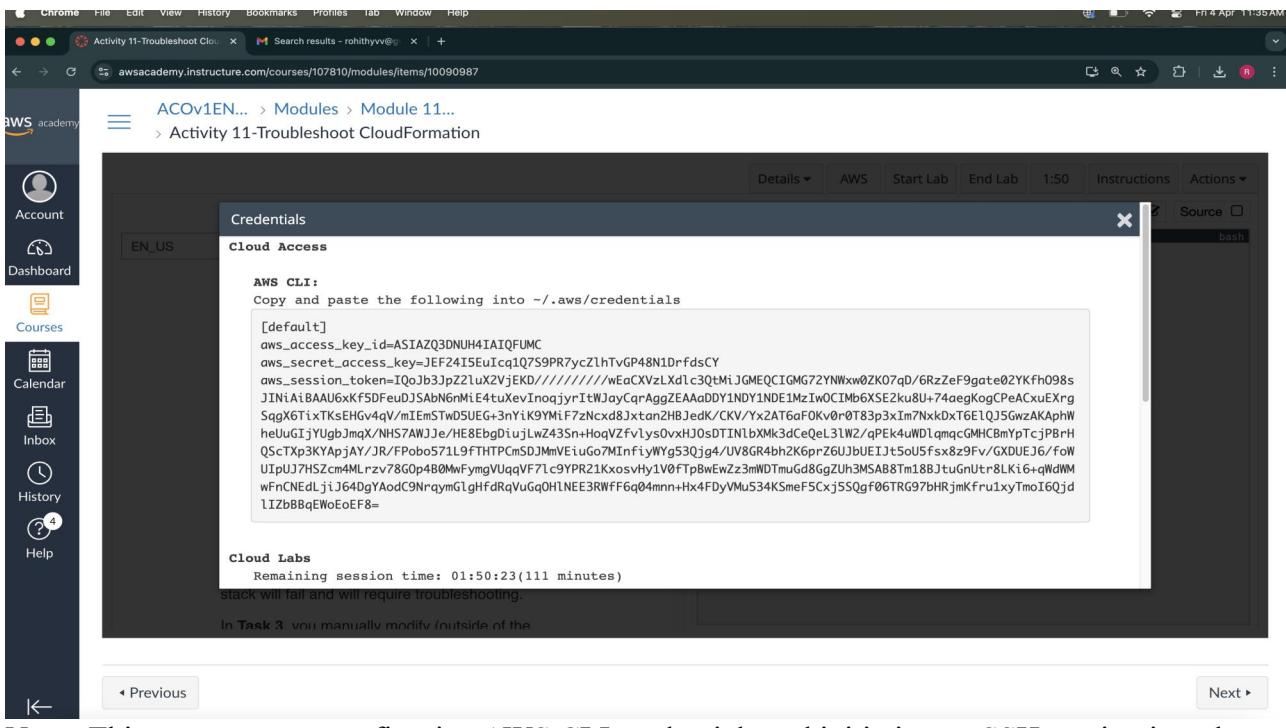
Note: The image shows the JMESPath website, in the document window that currently displays the locations JSON document, copied from the JSON document provided in the lab

Fig 11.6: To retrieve the LogicalResourceId of the EC2 instance resource



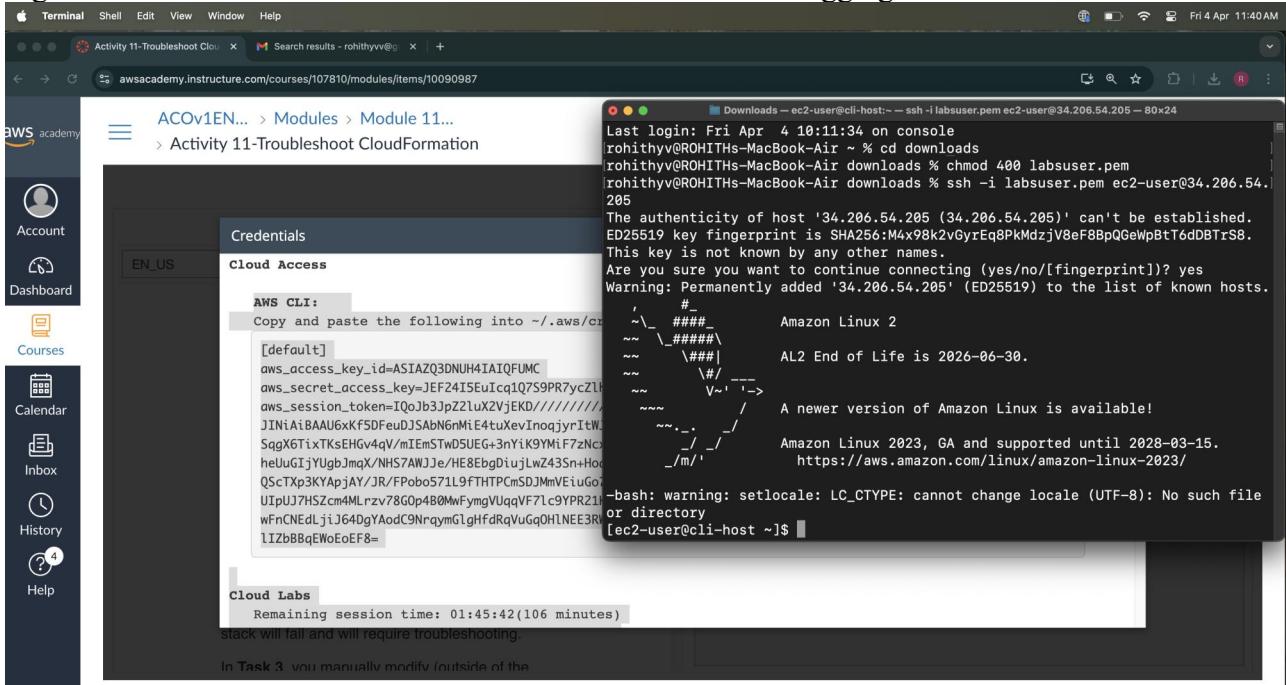
Note: The image shows "StackResources[?ResourceType == 'AWS::EC2::Instance'].LogicalResourceId" is executed to retrieve the LogicalResourceId of the EC2 instance resource. The instructions often have AWS CLI commands that include --query or --filter parameters. These parameters use JMSEPath expressions to filter the output that is returned by an AWS CLI command.

Fig 11.7: CloudFormation CLI Setup and EC2 SSH Session for Troubleshooting



Note: This step captures configuring AWS CLI credentials and initiating an SSH session into the EC2 CLI host. By accessing the instance terminal, users can manually investigate and resolve CloudFormation stack failures directly from the command line interface.

Fig 11.8: Secure CLI Authentication and CloudFormation Debugging via EC2 Access



Note: The screenshot demonstrates secure authentication using temporary AWS credentials and SSH key-based login into an EC2 instance. This is a preparatory step to perform hands-on debugging of AWS CloudFormation issues within a lab environment using the CLI tools.

Fig 11.13: End of module 11

The screenshot shows a browser window with the AWS CloudFormation Module 11 Knowledge Check results. The URL is awsacademy.instructure.com/courses/107810/assignments/1198230/module_item_id=10090989. The page title is "Module 11 Knowledge Check". On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area displays the results of the knowledge check:

Module 11 knowledge check results

Your score: 100% (100 points)
Required score: 70% (70 points)

Result: Congratulations! You have completed this module.
To continue, choose **Next** in the lower-right corner.

At the bottom, it says "© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved."

On the right side of the results box, there's a "Submission" section showing the grade and submission details:

- Grade: 100 (100 pts possible)
- Graded Anonymously: no
- Comments: No Comments

At the bottom of the page, there are "Previous" and "Next" navigation buttons.

Note: This image shows the completion of module 11

Summary:

- User data allows you to run custom scripts at instance launch, offering a simple way to automate configuration, though it becomes harder to manage at scale in enterprise environments.
- Amazon Machine Images (AMIs) help reduce deployment time by pre-configuring EC2 instances with software and settings specific to your organization's needs.
- Configuration tools like Chef, Puppet, and Ansible let you apply reusable templates to manage and update instance configurations dynamically.
- AWS offers managed services like OpsWorks for Chef/Puppet and CloudFormation for defining infrastructure as code, enabling consistent, repeatable deployments.