

Operations Playbook

Name: **HIMA SREE CHALASANI**

Class: **IFT 422-562: Cloud Security & Operations for IT**

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.....	5
How to connect to an EC2 instance to describe instances.....	6
How to create a batch file to upload update the café website to change its colors.....	7
How to launch an EC2 instance.....	7
How to change the AMI instance on the create-lamp-instance.sh script.....	9
How to tail a log in Linux	10
How to create an Auto Scaling Group in the AWS UI.....	11
How to create a Route 53 health check	11
How to create a Lambda Layer and add it to a Lambda function	12
How to create a Lambda function from a prebuilt package.....	13
How to create a Lambda function from a prebuilt package.....	14
How to collect information about an instance.....	15
How to create two subnets in a subnet group via the AWS CLI.....	16
How to use the mysqldump tool to take a backup of a SQL database and restore it on another SQL instance	18
How to enable VPC Flow Logs via the command line interface.....	19
How to troubleshoot network connectivity on an instance.....	20
How to take a snapshot of an EBS volume	21
How to synchronize files using the command line (aws s3api and aws s3).....	23
How to create a S3 bucket via the CLI.....	27
How to add an event notification to a S3 bucket	29
How to install the CloudWatch Agent.....	30
How to create a CloudWatch Events/CloudWatch EventBridge notification rule	31
How to use the prebuilt stopinator script to turn off instances with the tag value of your full name.....	32
How to resize an EC2 instance using the AWS CLI	33
How to detect drift in a CloudFormation template	34
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	40

How to create a Lambda Layer and add it to a Lambda function	41
How to create a Lambda function from a prebuilt package.....	42
How to setup a VPC	44
How to add a bastion host (Linux) to the public subnet of a VPC to connect to instances in the private subnet.....	46
How to setup IAM so a user can assume an IAM role to access a resource	47
How to setup AWS Config to monitor resources	49
How to add inbound rules to both security groups and network ACLs	51
How to encrypt the root volume of an existing EC2 instance.....	52
How to create a SNS topic	54
How to subscribe to a SNS topic.....	55
How to create a CloudWatch alarm using a metrics-based filter.....	57

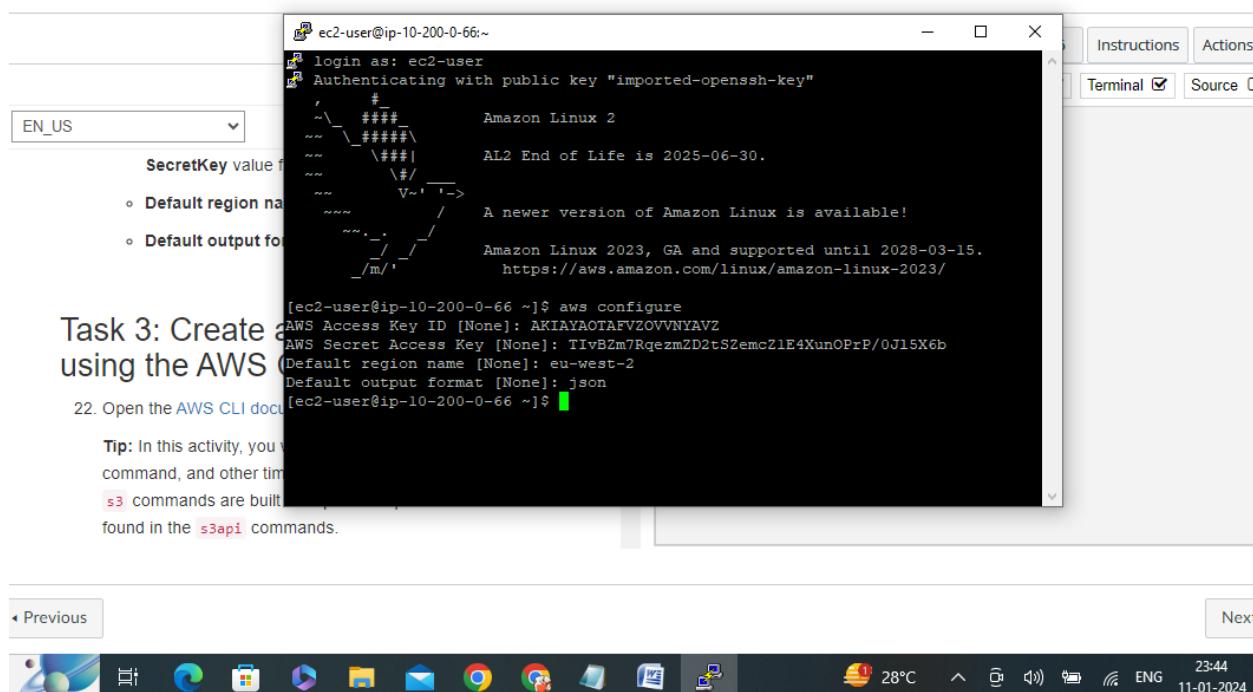
How to connect to the Mom & Pop Cafe Test EC2 instance

1. Ensure you have a copy of the necessary ppk/pem file for instance verification.
2. Open Putty and configure your connection with the specified settings.
3. In the Connection section, set the "Seconds between keepalives" to 30.
4. Input the public IPv4 address of your EC2 instance into the hostname area.
5. Add the ppk/pem file to your connection configuration.
6. Start the session by selecting "open" and log in using the "ec2-user" as the username to gain entry to the instance.

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

1. Initiate an SSH connection to your EC2 instance.
2. Adjust settings in Putty as follows:
 - To prolong the Putty session time, go to the Connections section and adjust the "Seconds between keepalives" setting to 30.
 - Enter the IPv4 address of the Bastion Host in the Hostname/IPv4 input field.
 - Within the Connection tab, access the SSH section, and in the Authentication (Auth) area, you need to load the private key file.
 - Use the "browse" option to upload the ppk file you have previously acquired.
 - After ensuring the checkbox has been marked, click "Open," then proceed with "Approve." Input "ec2-user" for the username to initiate a connection to the instance.

Modules > Module 2 - T... > Activity 2 - Create a Website on S3



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

1. After setting up the AWS CLI within our AWS account, we are required to modify the JSON policy document provided during the lab.
2. Executing the "AWS IAM list-users" command will generate a JSON output listing all users managed by IAM for this account, allowing us to review IAM configurations.
3. Upon receiving the list, we'll replicate the previous output and save it into a file named "lab_policy.json."
4. To inspect the contents of this file, the "cat" command is useful, and for editing purposes, the "vi" editor is recommended. After making all necessary edits, save your changes by executing the "wq" command within the vi editor.

A screenshot of a terminal window titled "Academy Cloud Operations - Module 1.". The window shows a black terminal interface with white text. At the top, the prompt is "ec2-user@ip-10-200-0-70:~". Below the prompt, there is a single character "h" followed by a cursor. The bottom of the terminal window displays the command "himasree@ip-10-200-0-70: ~" and the status "1, 8 All". In the bottom right corner of the terminal window, there is a scroll bar. The terminal window is set against a light gray background.

```
himasree@ip-10-200-0-70:~
```

"lab_policy" 1L, 9B

Operations Playbook Instructions

When building your operations playbook throughout this class make sure you keep the following in mind:



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

1. Navigate to the AWS Management Console and click on "Parameter Store" from the sidebar menu.
2. Upon selecting this feature, you will be asked to enter both a name and a description for the parameter.
3. The software operating on the linked EC2 instance will verify the parameter's presence before it processes any input in designated areas.
4. Should the application detect an additional argument, it will integrate that functionality and reflect the changes on the related web page.
5. The system will seamlessly integrate any new parameters found throughout this procedure into its operations.

Edit parameter

Parameter details

Name: /dashboard/show-beta-features

Description — *Optional*: Display beta features- Hima sree

Tier: Standard (selected)

Type:

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Shell Feedback Type here to search 30°C ENG 13-01-2024

How to connect to an EC2 instance to describe instances

1. Log into the AWS Management Console and navigate to the Systems Manager section.
2. Choose "Session Manager" from the navigation pane on the left.
3. Initiate a session by clicking on "Start session."
4. Execute the command "ls /var/www/html" during the session to inspect the files located on the EC2 instance.
5. Utilize the command "aws ec2 describe-instances" to establish a connection with an EC2 instance.

```
Session ID: user2718047=Chalasani_Hima_Sree-Ofc16ef0ee5d99aa7
Instance ID: i-0e5bfb6641af48907
Terminal
```

```
{
    "Instances": [
        {
            "InstanceId": "i-0e5bfb6641af48907",
            "ImageId": "ami-0fc16ef0ee5d99aa7",
            "Type": "t2.micro",
            "State": "running",
            "SubnetId": "subnet-0ad8a60ac12f",
            "PrivateIpAddress": "172.31.10.10",
            "PublicIpAddress": "54.177.111.10",
            "NetworkInterfaces": [
                {
                    "InterfaceType": "primary",
                    "MacAddress": "enx0a0000000000",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "AllocationId": "eipalloc-0000000000000000",
                            "PrivateIpAddress": "172.31.10.10",
                            "PublicIpAddress": "54.177.111.10"
                        }
                    ],
                    "Primary": true
                }
            ],
            "RootDeviceType": "ebs",
            "RootDeviceName": "/dev/xvda",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/xvda",
                    "Ebs": {
                        "VolumeSize": 8,
                        "VolumeType": "standard"
                    }
                }
            ],
            "MetadataOptions": {
                "HttpPutResponseHopLimit": 1
            },
            "AmiLaunchIndex": 0
        }
    ],
    "Reservations": [
        {
            "ReservationId": "r-0c58ce0bda6d0841e",
            "Groups": [],
            "OwnerId": "065625487961"
        }
    ]
}
```

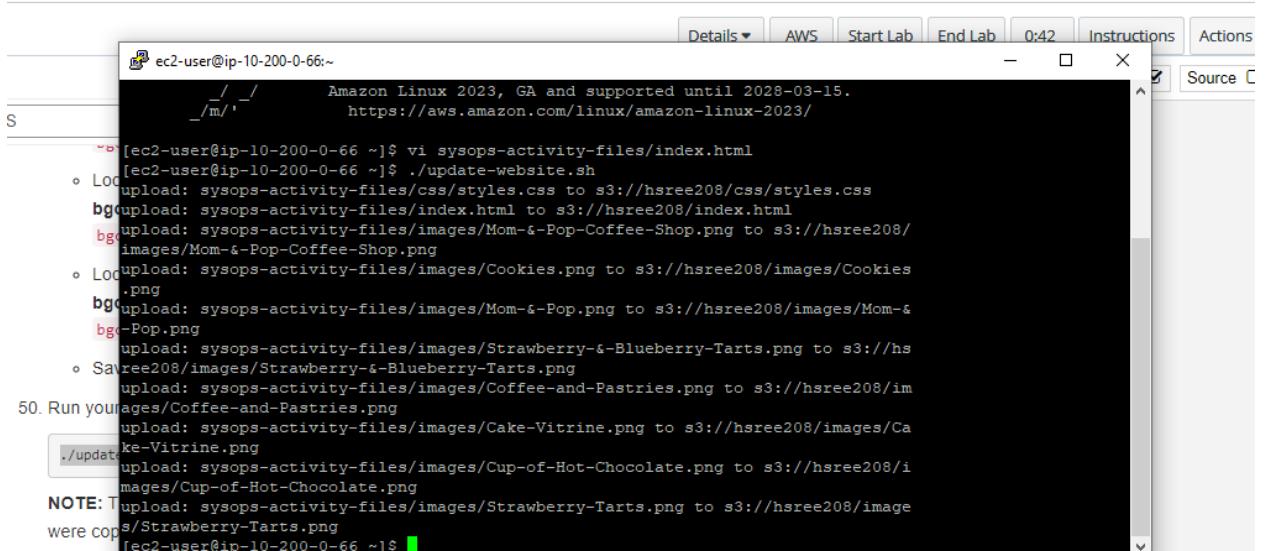
sh-4.2\$ Hima sree Chalasani

Ne... 00:23 ENG 13-01-2024

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

1. Generate a file called "update-website.sh" by employing the "touch" command and moving to the desired directory with "cd."
2. Open the "update-website.sh" file for editing by using the "vi" text editor.
3. Commit any changes you've made to the file by saving them.
4. Use a text editor to access the "vi sysops-activity-files/index.html" file.
5. Once the background color has been altered, save the modifications and proceed to upload the revised file.

s > Module 2 - T... > Activity 2 - Create a Website on S3



```
ec2-user@ip-10-200-0-66:~$ vi sysops-activity-files/index.html
ec2-user@ip-10-200-0-66:~$ ./update-website.sh
upload: sysops-activity-files/css/styles.css to s3://hsree208/css/styles.css
upload: sysops-activity-files/index.html to s3://hsree208/index.html
upload: sysops-activity-files/images/Mom-&-Pop-Coffee-Shop.png to s3://hsree208/images/Mom-&-Pop-Coffee-Shop.png
upload: sysops-activity-files/images/Cookies.png to s3://hsree208/images/Cookies.png
upload: sysops-activity-files/images/Mom-&-Pop.png to s3://hsree208/images/Mom-&-Pop.png
upload: sysops-activity-files/images/Strawberry-&-Blueberry-Tarts.png to s3://hsree208/images/Strawberry-&-Blueberry-Tarts.png
upload: sysops-activity-files/images/Coffee-and-Pastries.png to s3://hsree208/images/Coffee-and-Pastries.png
50. Run your images/Coffee-and-Pastries.png
upload: sysops-activity-files/images/Cake-Vitrine.png to s3://hsree208/images/Cake-Vitrine.png
upload: sysops-activity-files/images/Cup-of-Hot-Chocolate.png to s3://hsree208/images/Cup-of-Hot-Chocolate.png
upload: sysops-activity-files/images/Strawberry-Tarts.png to s3://hsree208/images/Strawberry-Tarts.png
[ec2-user@ip-10-200-0-66:~]$
```

NOTE: The files were copied from the local machine to the S3 bucket.

How to launch an EC2 instance

1. Begin by accessing the EC2 dashboard.
2. Identify and select the specific EC2 instance you intend to launch before starting the creation process.
3. Once an instance is chosen, click on "Launch Instance" to proceed.
4. In the subsequent window, assign a name or identifier to your new instance.
5. Following the naming, select the desired operating system (OS) and application image to serve as the foundation for your instance.
6. Next, choose a key pair for the instance, ensuring to pick the "rockey" key pair in this step.
7. After hitting the "Launch Instance" option, take a moment to review your network configurations carefully.

following the simple steps below.

Name and tags

Key **Info** Value **Info** Resource types **Info**
Name **Bastion Server** Select resource ty... Remove
Instances **X**

Key **Info** Value **Info** Resource types **Info**
IFT422 **Hima sree Chala** Select resource ty... Remove
Instances **X**

Add new tag

You can add up to 48 more tags.

Application and OS Images (Amazon Machine Image)

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences N. Virginia voclabs/user2718047=Chalasani_Hima_Sree @ 0549-8790-9

Services Search [Alt+S] N. Virginia voclabs/user2718047=Chalasani_Hima_Sree @ 0549-8790-9

Summary

Number of instances **Info**
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0a3c3a20c09d6f377

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Review commands

Network settings

VPC - required **Info**
vpc-02e0c2091f821d6eb (Lab VPC)
10.0.0/16

Subnet **Info**
subnet-0c76ab116d8d08143 Public Subnet
VPC: vpc-02e0c2091f821d6eb Owner: 054987909980 Availability Zone: us-east-1a
IP addresses available: 250 CIDR: 10.0.0.0/24 Create new subnet

Auto-assign public IP **Info**
Enable

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
Bastion security group

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences N. Virginia voclabs/user2718047=Chalasani_Hima_Sree @ 0549-8790-9

Services Search [Alt+S] N. Virginia voclabs/user2718047=Chalasani_Hima_Sree @ 0549-8790-9

Summary

Number of instances **Info**
1

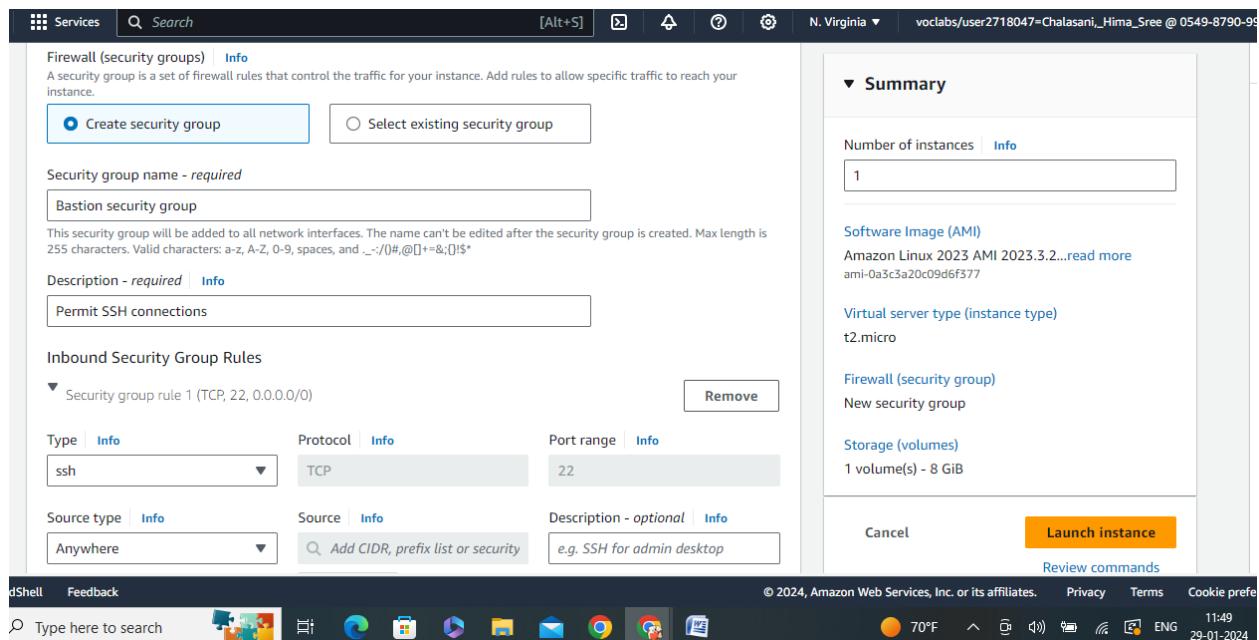
Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0a3c3a20c09d6f377

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Review commands



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

1. The process begins by initiating an SSH connection to an EC2 instance, which is the first step towards modifying the AMI instance.
2. Next, adjust the AWS setup to meet the particular demands of the operation.
3. Then, it's essential to create a backup and work with the "create-lamp-instance.sh" script, which contains important information such as VPC, SubnetID, Security Group, and InstanceID.
4. In case you encounter the "InvalidAMIID.noresult" error, it's recommended to execute the script again. Use the error notification to pinpoint the exact location of the error's origin.
5. Note that the script includes a default AMI ID, which must be replaced with the specific AMI ID you plan to use, to ensure the deployment of the correct AMI instance.
6. Once you have completed the required changes and validated the exact AMI ID, rerun the command to ensure that the AMI ID has been correctly modified as planned.

> Modules > Module 3 ... > Activity 3

ec2-user@cli-host:~/sysops-activity-files/starters

```

Authenticating with public key "imported-openssh-key"
,
  #####
  ~\_#####
  ~~ \###|
  ~~ \|#
  ~~ 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/

```

Task 3.1: Observe the output of the AWS CLI command.

22. Change to the directory editing exists, and create these commands:

```

[ec2-user@cli-host ~]$ aws configure
AWS Access Key ID [None]: AKIASNZDIXRAZZJAUUGM
AWS Secret Access Key [None]: NO3aBM3MftF2mB7wPsT+nOZ+hROMKMLkA54tt7
Default region name [None]: us-east-1
Default output format [None]: json

```

Tip: It is always a good practice to start making modifications to the script file in your favorite command-line text editor (such as VI).

```

[ec2-user@cli-host ~]$ cd ~/sysops-activity-files/starters
[ec2-user@cli-host starters]$ cp create-lamp-instance.sh create-lamp-instance.bak
[ec2-user@cli-host starters]$ vi create-lamp-instance.sh

```

23. Open the create-lamp-instance.sh script file in your favorite command-line text editor (such as VI).

vi create-lamp-instance.sh

◀ Previous Next ▶

EN_US

File Edit View Insert Cell Help

47°F ENG 19-01-2024

> Modules > Module 3 ... > Activity 3 - Troubleshoot Create Instance

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="eu-west-2"
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 $profile | grep VpcId | cut -d '"' -f1 | sed -n 1p)
echo "VPC: '$vpc"

```

Line 1: 1,1 Top

- This file is a bash file, so the first line contains `#!/bin/bash`

◀ Previous Next ▶

EN_US

File Edit View Insert Cell Help

47°F ENG 19-01-2024

How to tail a log in Linux

1. Examine the contents of the log file.

2. Use the command "tail -f" to display the most recent 10 entries while also receiving updates in real-time.
3. To adjust the number of lines displayed, employ the '-n' option.
4. To stop monitoring and save any modifications, press CTRL+C; to exit, enter ":q."

How to create an Auto Scaling Group in the AWS UI

1. Sign into your AWS account and navigate to the Services menu to select EC2 on the sidebar.
2. Opt for "Create an Auto Scaling Group" among the listed services.
3. Select a launch template and adjust settings like VPC and Subnet configurations.
4. Link to the load balancer and determine the group's size parameters:
 - Desired capacity: 6
 - Minimum capacity: 4
 - Maximum capacity: 10
5. Implement a scaling policy and append relevant tags.
6. Following these instructions will effectively establish your Auto Scaling Group.

The screenshot shows the AWS EC2 Auto Scaling Groups interface. At the top, there are tabs for 'Launch configurations', 'Launch templates', 'Actions', and a prominent orange 'Create Auto Scaling group' button. Below this is a search bar and a filter bar with dropdowns for 'Launch template/configuration', 'Instances', 'Status', 'Desired capacity', 'Min', and 'Max'. The main content area is titled 'Auto Scaling groups (1/1) Info' and shows a single entry for 'WebServersASGroup'. This entry has a status of 'Successful' and a timestamp of 'At 2024-01-26T02:38:13Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2024-01-26T02:38:25Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.' Below this, another entry shows a second successful launch. The bottom of the screen includes the AWS navigation bar with links for 'Search', 'Back', and various services like S3, Lambda, and CloudWatch. The footer contains copyright information and a date stamp '25-01-2024'.

How to create a Route 53 health check

1. Start by logging into the AWS Management Console and proceed to Route 53. Within the dashboard of Route 53, on the navigation pane to the left, choose "Health Check" before selecting the "Create Health Check" option.
2. Begin the creation process by selecting "Create Health Check" found within the options for Health Checks.
3. Assign a distinctive and informative name to the Health Check.
4. In the "What to Monitor" section, select "Endpoint."

5. Identify and enter the IP address you wish to monitor, which is available in the instance details section, specifically copying the IPv4 Public IP Address from "MomPopCafeInstance1."
6. Adjust the "Requested Interval" to "Fast" and set the "Failure Threshold" to 2 within the advanced settings area.
7. Decide to generate an alarm by choosing "Create Alarm: Yes."
8. Upon request, designate a name for your newly created Simple Notification Service (SNS) topic as desired.
9. Fill in the email address to which notifications will be dispatched once this health check is accomplished.

 Gmail

Hima sree Chalasani <chalahsanihimasree@gmail.com>

ALARM: "Primary-Website-Health-awsroute53-8dbd926c-f6d9-4866-9616-d63ca..." in US East (N. Virginia)

1 message

AWS Notifications <no-reply@sns.amazonaws.com>
To: chalahsanihimasree@gmail.com

Thu, Jan 25, 2024 at 9:05

You are receiving this email because your Amazon CloudWatch Alarm "Primary-Website-Health-awsroute53-8dbd926c-f6d9-4866-9616-d63cac30b841-Low-HealthCheckStatus" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 datapoint [0.0 (26/01/24 04:04:00)] was less than the threshold (1.0)." at "Friday 26 January, 2024 04:05:43 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm:Primary-Website-Health-awsroute53-8dbd926c-f6d9-4866-9616-d63cac30b841-Low-HealthCheckStatus>

Alarm Details:

- Name: Primary-Website-Health-awsroute53-8dbd926c-f6d9-4866-9616-d63cac30b841-Low-HealthCheckStatus
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint [0.0 (26/01/24 04:04:00)] was less than the threshold (1.0).
- Triggered At: Friday 26 January, 2024 04:05:43 UTC
- AWS Account: 339712807827
- Alarm Arn: arn:aws:cloudwatch:us-east-1:339712807827:alarm:Primary-Website-Health-awsroute53-8dbd926c-f6d9-4866-9616-d63cac30b841-Low-HealthCheckStatus

Threshold:

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

Monitored Metric:

- MetricNamespace: AWS/Route53
- MetricName: HealthCheckStatus
- Dimensions: [HealthCheckId = 8dbd926c-f6d9-4866-9616-d63cac30b841]
- Period: 60 seconds
- Statistic: Minimum
- Unit: not specified

State Change Actions:

- OK
- ALARM: [arn:aws:sns:us-east-1:339712807827:Primary-Website-Health]
- INSUFFICIENT_DATA:



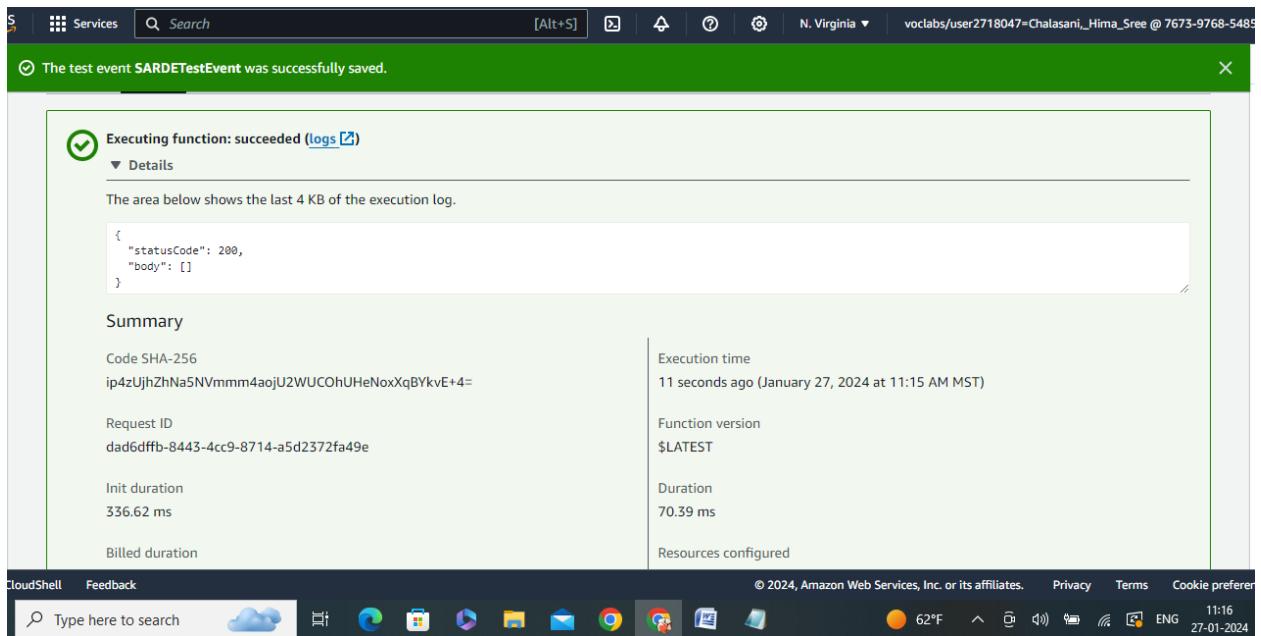
How to create a Lambda Layer and add it to a Lambda function

1. Navigate to the AWS Management Console and select Lambda from the options.
2. For introducing a new layer, access "Layers" and press "Create Layer." During this step, provide a name, detailed description, and select the method for code input.
3. From the creation options, hit "Run" to verify the command executes properly.
4. Initiate the creation of a new function by opting for "Author from Scratch."
5. Proceed to create the function by clicking "Create Function." During this process, fill in details such as the function's name, its runtime environment, and the roles it will execute under.
6. Go to "Add Layers" and choose "Custom Layers" for inclusion.
7. Observe as the interface updates. Following the function's setup, confirm the layer's addition was successful.

The screenshots show the 'Create layer' configuration page in the AWS Lambda console. The top section is 'Layer configuration' with fields for 'Name' (pymysqlLibrary) and 'Description - optional' (PyMySQL 0.9.3 library modules - Hima sree Chalasani). A radio button is selected for 'Upload a .zip file'. Below this is a 'Upload' button with a file selection dialog showing 'pymysql-1.1.0.zip' (107.63 KB). A note says 'For files larger than 10 MB, consider uploading using Amazon S3.' The bottom sections include 'Compatible architectures - optional' (checkboxes for x86_64 and arm64), 'Compatible runtimes - optional' (dropdown showing Python 3.9), and 'License - optional' (empty text area). At the bottom right are 'Cancel' and 'Create' buttons.

How to create a Lambda function from a prebuilt package

1. Launch the AWS Management Console and proceed to the Lambda section.
2. Initiate a fresh function, selecting "Author from Scratch" as the method. Name this function "SalesAnalysisReportDataExtractor," choose Python 3.7 as the runtime, and designate the correct execution role.
3. Incorporate the necessary library along with a custom layer.
4. Proceed to upload a zip file, conduct a validation of the data, and modify the handler entry to "salesAnalysisReportDataExtractor.lambda_handler."



How to create a Lambda function from a prebuilt package

1. Install and configure the AWS CLI on your machine.
2. Use the following command to create a Database (DB) subnet group: `aws rds create-db-subnet-group`.
3. Create a DB parameter group by executing: `aws rds create-db-parameter-group`.
4. Start the RDS instance setup process with the command: `aws rds create-db-instance`.
5. Check the progress of the RDS instance with: `aws rds describe-db-instances`.

↳ Modules > Module 6 - C... > Activity 6 - Migrate to Amazon RDS

```
ec2-user@ip-10-200-0-48:~
```

AWS Access Key ID [None]: AKIATCKASGITW4QX6QLK
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-10-200-0-48 ~]\$ aws ec2 describe-instances \
> --filters "Name=tag:Name,Values= MomPopCafeInstance" \
--query "Reservations[*].Instances[*].[InstanceId, InstanceType, PublicDnsName, P
ublicIpAddress, Placement.AvailabilityZone, VpcId, SecurityGroups[*].GroupId]"
[
[
["i-06638a8d199ff9c5e",
"t2.small",
"ec2-3-80-121-178.compute-1.amazonaws.com",
"3.80.121.178",
"us-east-1a",
"vpc-0a4580fd46c2d4648",
[
"sg-0bc2b8e5ed687e6cf"
]
]
]
]
Information: For readability, long commands are split across multiple lines. To enter them as
separate lines by using the backslash character for the Linux command line, press the Enter key
character for the Linux command line [ec2-user@ip-10-200-0-48 ~]\$. This will allow you to enter
multiple lines at a Linux prompt.

◀ Previous Next ▶

The bottom status bar shows CloudShell, Feedback, a search bar, and system icons like battery and network.

How to collect information about an instance

1. Step 1: Acquire the Instance ID

Execute the command "aws ec2 describe-instances" and isolate the InstanceId by applying a specific query.

2. Step 2: Fetch Details of the Instance

Utilize "aws ec2 describe-instances" alongside the instance-ids option to gather data. Isolate important details such as InstanceType, PublicDnsName, among others, via a query.

3. Step 3: Gather Data on Regional Availability Zones

Invoke "aws ec2 describe-availability-zones" to get details.

From AvailabilityZones[*], deduce the ZoneName.

```

ec2-user@ip-10-200-0-48:~ 
"vpc-0a4580f4d6c2d4648",
[
    "sg-0bc2b8e5ed687e6cf"
]
]

[ec2-user@ip-10-200-0-48 ~]$ ^C
[ec2-user@ip-10-200-0-48 ~]$ aws ec2 describe-vpcs --vpc-ids vpc-0a4580f4d6c2d4648 \
> --filters "Name=tag:Name,Values= MomPopCafe VPC" \
> --query "Vpcs[*].CidrBlock"
[

]

[ec2-user@ip-10-200-0-48 ~]$ aws ec2 describe-subnets \
> --filters "Name=vpc-id,Values=<MomPopCafe VPC ID>" \
> --query "Subnets[*].[SubnetId,CidrBlock]"
[

]
[
    [
        "subnet-0f036b1a6b9a9ee2c",
        "10.200.0.0/24"
    ]
]

In the command, substitute <MomPopCafe VPC ID> with the value that you recorded earlier.
[ec2-user@ip-10-200-0-48 ~]$ 

```

EN_US

25. Record the value that was returned by format:
MomPopCafe VPC IPv4 CIDR block: 10.200.0.0/24

26. Determine the **Subnet ID** and **IPv4 CIDR Block** for MomPopCafe Public Subnet 1, which is part of the VPC. In the SSH window, enter:

```

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

```

In the command, substitute <MomPopCafe VPC ID> with the value that you recorded earlier.

[◀ Previous](#)

[Next ▶](#)

```

ec2-user@ip-10-200-0-48:~ 
AWS Access Key ID [None]: AKIATCKASGITW4QX6QLK
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-10-200-0-48 ~]$ aws ec2 describe-instances \
> --filters "Name=tag:Name,Values= MomPopCafeInstance" \
> --query "Reservations[*].Instances[*].[InstanceId,InstanceType,PublicDnsName,PublicIpAddress,Placement.AvailabilityZone,VpcId,SecurityGroups[*].GroupId]"
[

]
[
    [
        [
            "i-06638a8d199ff9c5e",
            "t2.small",
            "ec2-3-80-121-178.compute-1.amazonaws.com",
            "3.80.121.178",
            "us-east-1a",
            "vpc-0a4580f4d6c2d4648",
            [
                "sg-0bc2b8e5ed687e6cf"
            ]
        ]
]

Information: For readability, long commands are split across multiple lines. To enter them as a single command, separate lines by using the backslash character (\) at the end of each line. This character for the Linux command line allows you to enter multiple lines at a Linux prompt.
[ec2-user@ip-10-200-0-48 ~]$ 

```

EN_US

22. Determine the **Instance ID**, **Instance name**, **Public IP address**, and **Availability Zone** for the MomPopCafeInstance. Use an AWS CLI command that returns the **VPC ID** of its VPC, and the security group. In the SSH window, enter:

```

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

```

Information: For readability, long commands are split across multiple lines. To enter them as a single command, separate lines by using the backslash character (\) at the end of each line. This character for the Linux command line allows you to enter multiple lines at a Linux prompt.

[◀ Previous](#)

[Next ▶](#)

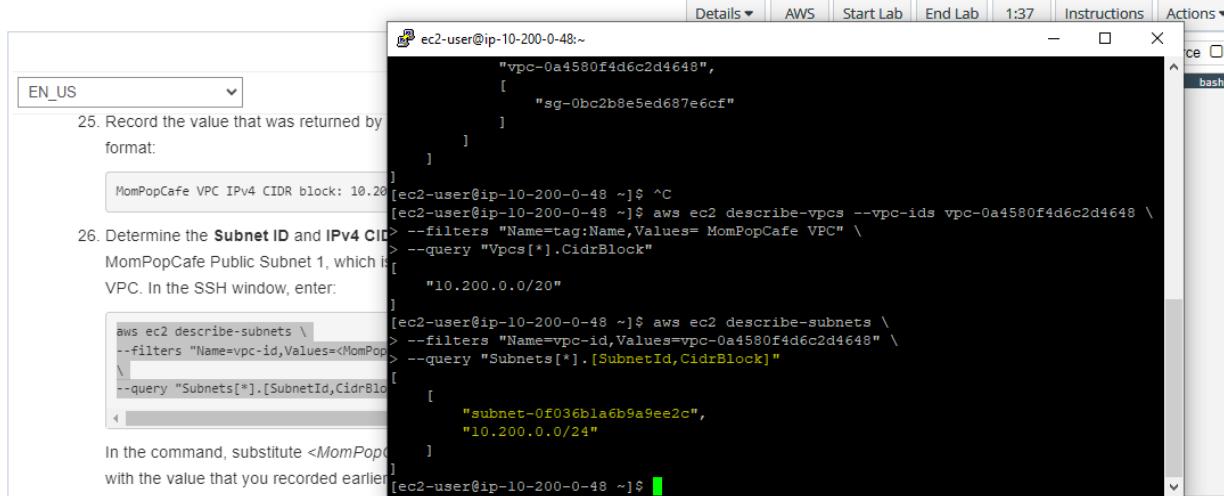


How to create two subnets in a subnet group via the AWS CLI

1. Begin with the creation of the first subnet by executing the `create-subnet` command in AWS EC2. Specify the VPC ID, CIDR block, and the availability zone designated for this subnet.

2. Create another subnet by utilizing the `create-subnet` command within AWS EC2. This time, ensure you input the VPC ID, a different CIDR block, and choose an alternative availability zone from the first subnet.
3. Form a subnet group using the `create-db-subnet-group` command in AWS RDS, incorporating the subnet IDs from the first and second steps into this setup.

> Modules > Module 6 - C... > Activity 6 - Migrate to Amazon RDS



```

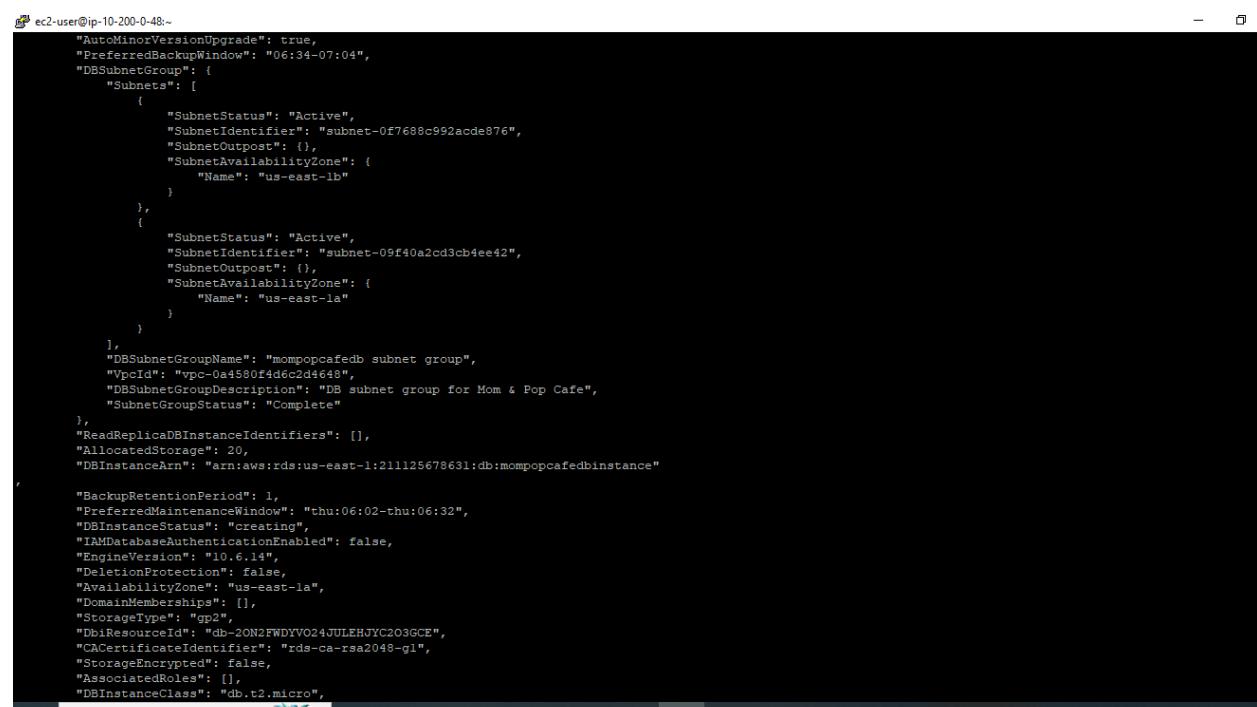
ec2-user@ip-10-200-0-48:~$ aws ec2 describe-vpcs --vpc-ids vpc-0a4580f4d6c2d4648 \
--filters "Name=tag:Name,Values= MomPopCafe VPC" \
--query "Vpcs[*].CidrBlock"
[{"CidrBlock": "10.200.0.0/20"}]
ec2-user@ip-10-200-0-48:~$ aws ec2 describe-subnets \
--filters "Name=vpc-id,Values=<MomPopCafe VPC>" \
--query "Subnets[*].[SubnetId,CidrBlock]"
[{"SubnetId": "subnet-0f036bla6b9a9ee2c", "CidrBlock": "10.200.0.0/24"}]
In the command, substitute <MomPopCafe VPC> with the value that you recorded earlier.
[ec2-user@ip-10-200-0-48 ~]$ 

```

[◀ Previous](#) [Next ▶](#)



56°F ENG 03-02-2024



```

{
  "AutoMinorVersionUpgrade": true,
  "PreferredBackupWindow": "06:34-07:04",
  "DBSubnetGroup": {
    "Subnets": [
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-0f768c992acde876",
        "SubnetOutpost": {},
        "SubnetAvailabilityZone": {
          "Name": "us-east-1b"
        }
      },
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-09f40a2cd3cb4ee42",
        "SubnetOutpost": {},
        "SubnetAvailabilityZone": {
          "Name": "us-east-1a"
        }
      }
    ],
    "DBSubnetGroupName": "mompopcafedb subnet group",
    "vpcId": "vpc-0a4580f4d6c2d4648",
    "DBSubnetGroupDescription": "DB subnet group for Mom & Pop Cafe",
    "SubnetGroupStatus": "Complete"
  },
  "ReadReplaceDBInstanceIdentifiers": [],
  "AllocatedStorage": 20,
  "DBInstanceArn": "arn:aws:rds:us-east-1:211125678631:db:mompopcafedbinstance"
}

"BackupRetentionPeriod": 1,
"PreferredMaintenanceWindow": "thu:06:02-thu:06:32",
"DBInstanceState": "creating",
"IAMDatabaseAuthenticationEnabled": false,
"EngineVersion": "10.6.14",
"DeletionProtection": false,
"AvailabilityZone": "us-east-1a",
"DomainMemberships": [],
"StorageType": "gp2",
"DbiResourceId": "db-2ON2FWDIVV024JULEHJYC2O3GCE",
"CACertificateIdentifier": "rds-ca-rsa2048-g1",
"StorageEncrypted": false,
"AssociatedRoles": [],
"DBInstanceClass": "db.t2.micro",

```

54°F ENG 03-02-2024

```
ec2-user@ip-10-200-0-48:~$ cat rds-db-instance.json
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mompopcafedb subnet group",
        "DBSubnetGroupDescription": "DB subnet group for Mom & Pop Cafe",
        "Status": "Complete"
    },
    "DBInstance": {
        "DBName": "mompopcafedb",
        "DBInstanceIdentifier": "mompopcafedbinstance",
        "DBInstanceClass": "db.t2.micro",
        "DBInstancePort": 0,
        "DBInstanceStatus": "creating",
        "DBParameterGroups": [],
        "DBSecurityGroups": [],
        "DBSnapshotIdentifiers": [],
        "AllocatedStorage": 20,
        "BackupRetentionPeriod": 1,
        "PreferredMaintenanceWindow": "thu:06:02-thu:06:32",
        "IAMDatabaseAuthenticationEnabled": false,
        "EngineVersion": "10.6.14",
        "DeletionProtection": false,
        "AvailabilityZone": "us-east-1a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-2ON2FWDYVO24JULEHJYC2O3GCE",
        "CACertificateIdentifier": "rds-ca-rsa2048-ql",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceArn": "arn:aws:rds:us-east-1:211125678631:db:mompopcafedbinstance"
    }
}

[ec2-user@ip-10-200-0-48 ~]$ Hima sree Chalasani
bash: Hima: command not found
[ec2-user@ip-10-200-0-48 ~]$
```



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

1. Install mysqldump on the primary database server.
2. Use mysqldump to generate a backup of the database with the command: `mysqldump -u <user> -p<password> <database> > backup.sql`.
3. Transfer the backup file to the target server.
4. Create a fresh database on the destination server using the command: `mysql> create database <database name>`.
5. Restore the backup onto the new database using mysql with the command: `mysql -u <user> -p<password> <database name> < backup.sql`.

```
[ec2-user@ip-10-200-0-143 ~]$ mysql --user=root --password='Re:Start!9' \
> --host=mompopcafedbinstance.cpmcmgyw0eqz.us-east-1.rds.amazonaws.com \
> mom_pop_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor. Commands end with ; or \q.
Your MariaDB connection id is 49
Server version: 10.6.14-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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-4-Blueberry-Tarts.jpg
| 6 | Strawberry Tart | Made with fresh ripe strawberries and a delicious whipped cream | 3.50 | 1 | images/Strawberry-Tarts.JPG
| 7 | Coffee | Freshly-ground black or blended Columbian coffee | 3.00 | 2 | images/Coffee.jpg
| 8 | Hot Chocolate | Rich and creamy, and made with real chocolate | 3.00 | 2 | images/Cup-of-Hot-Chocolate.jpg
| 9 | Latte | Offered hot or cold and in various delicious flavors | 3.50 | 2 | images/Latte.jpg
+----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

MariaDB [mom_pop_db]> Hima sree Chalasani
```



How to enable VPC Flow Logs via the command line interface

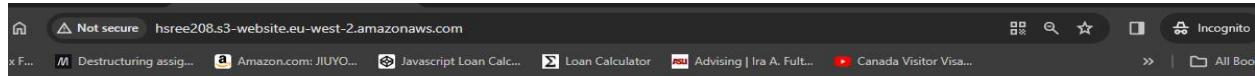
1. Generate an IAM role for VPC Flow Logs: 'aws iam create-role --role-name VPCFlowLogsRole'.
2. Establish a CloudWatch log group: 'aws logs create-log-group --log-group-name VPCFlowLogs'.
3. Activate flow logs on the VPC using 'aws ec2 create-flow-logs', indicating the log destination.

```
[ec2-user@cli-host~]$ aws s3api create-bucket --bucket flowlog5555 --region us-east-1
{
  "Location": "/flowlog5555"
}
[ec2-user@cli-host ~]$ aws ec2 describe-vpcs --query 'Vpcs[].[VpcId,Tags[?Key==`Name`].Value,CidrBlock
]' --filters "Name=tag:Name,Values='VPC1'"
[
  [
    {
      "vpc-0fd9d12917af77872",
      [
        "VPC1",
        "10.0.0.0/16"
      ]
    }
]
[ec2-user@cli-host ~]$ aws ec2 create-flow-logs --resource-type VPC --resource-ids vpc-0fd9d12917af7787
2 --traffic-type ALL --log-destination-type s3 --log-destination arn:aws:s3:::flowlog5555
(
  "Unsuccessful": [],
  "FlowIds": [
    "fl-00af88eb8f0a9990"
  ],
  "ClientToken": "SouhAv29sbTz9y7mc4dqr8k1XCG/6E7EdJuKJ6Ql5Eg="
)
[ec2-user@cli-host ~]$ aws ec2 describe-flow-logs
{
  "FlowLogs": [
    {
      "LogDestinationType": "s3",
      "Tags": [],
      "ResourceId": "vpc-0fd9d12917af77872",
      "CreationTime": "2024-02-09T20:07:31.917Z",
      "TrafficType": "ALL",
      "FlowLogStatus": "ACTIVE",
      "LogFormat": "$version $account-id $interface-id $srcaddr $dstaddr $srcport $dstport $protocol $packets $bytes $start $end $action $log-status",
      "FlowLogId": "fl-00af88eb8f0a9990",
      "MaxAggregationInterval": 600,
      "LogDestination": "arn:aws:s3:::flowlog5555",
      "DeliverLogsStatus": "SUCCESS"
    }
  ]
}
[ec2-user@cli-host ~]$ Hima sree Chalasani
```



How to troubleshoot network connectivity on an instance

1. Ensure that the instance's associated security groups permit both incoming and outgoing traffic on the necessary ports.
2. Validate that Network ACLs linked with the subnet allow essential traffic.
3. Verify the accuracy of the route table configurations, directing traffic to the appropriate destination.
4. Check the instance's operational status and ensure it is running correctly.
5. Confirm that the instance possesses a valid public IP address or Elastic IP if external communication is required.
6. Review system and application logs on the instance for any network connectivity-related error messages.
7. Conduct ping tests from both the instance and external systems to assess reachability.
8. Execute a traceroute to identify the network path and potential bottlenecks.
9. Check for the presence of security software (such as firewalls or antivirus) on the instance that may obstruct traffic.
10. If the instance communicates with resources in another VPC, validate VPC peering configurations.
11. Inspect instance metadata to ensure the accuracy of network settings.
12. Confirm that DNS resolution is functioning correctly.
13. Temporarily adjust security group rules to permit all traffic for testing purposes.
14. Ensure that subnet routing is appropriately configured and functional.



Mom & Pop Café



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



How to take a snapshot of an EBS volume

1. Tailor the settings in Putty to suit your preferences.
2. Set a 30-second interval for keepalives to bolster connection reliability.
3. Enter the public IPv4 address of the EC2 instance into the hostname field.
4. Connect securely by configuring Putty to use the designated PPK/PEM file for authentication.
5. Open the connection by clicking "Open" and inputting "ec2-user" as the username when prompted to gain access.
6. Retrieve the volume ID linked with the specified snapshot using the provided AWS CLI command: aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.{VolumeId:VolumeId}'
7. Obtain the instance ID necessary for terminating the instance with the following AWS CLI command: aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' -query 'Reservations[0].Instances[0].InstanceId'
8. Halt the ongoing process by executing the provided AWS CLI command: aws ec2 stop-instances --instance-ids INSTANCE-ID
9. Create a snapshot of the specified EBS volume using the provided AWS CLI command: aws ec2 create-snapshot --volume-id VOLUME-ID

Module 42 ec2-user@ip-10-5-0-198:~

```
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 wait instance-stopped --instance-id i-0b30de7809cb230cl
[ec2-user@ip-10-5-0-198 ~]$ ^C
[ec2-user@ip-10-5-0-198 ~]$ ^C
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 create-snapshot --volume-id vol-0fa674fed5aaa94d7
{
    "Description": "",
    "Encrypted": false,
    "OwnerId": "381491945051",
    "Progress": "",
    "SnapshotId": "snap-08d6f1c24b009298a",
    "StartTime": "2024-02-15T02:24:11.640Z",
    "State": "pending",
    "VolumeId": "vol-0fa674fed5aaa94d7",
    "VolumeSize": 8,
    "Tags": []
}
[ec2-user@ip-10-5-0-198 ~]$ Hima sree Chalasani
```

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

Continue with the below procedure when the command

Previous Next

```
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 describe-instances --filter 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.VolumeId'
{
    "VolumeId": "vol-0fa674fed5aaa94d7"
}
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 describe-instances --filters 'Name=tag:Name,Values=Processor' --query 'Reservations[0].Instances[0].InstanceId'
"i-0b30de7809cb230cl"
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 stop-instances --instance-ids INSTANCE-ID
An error occurred (InvalidInstanceId.Malformed) when calling the StopInstances operation: The instance ID 'INSTANCE-ID' is malformed
[ec2-user@ip-10-5-0-198 ~]$ ^C
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 stop-instances --instance-ids i-0b30de7809cb230cl
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-0b30de7809cb230cl",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
[ec2-user@ip-10-5-0-198 ~]$ Hima sree Chalasani
```

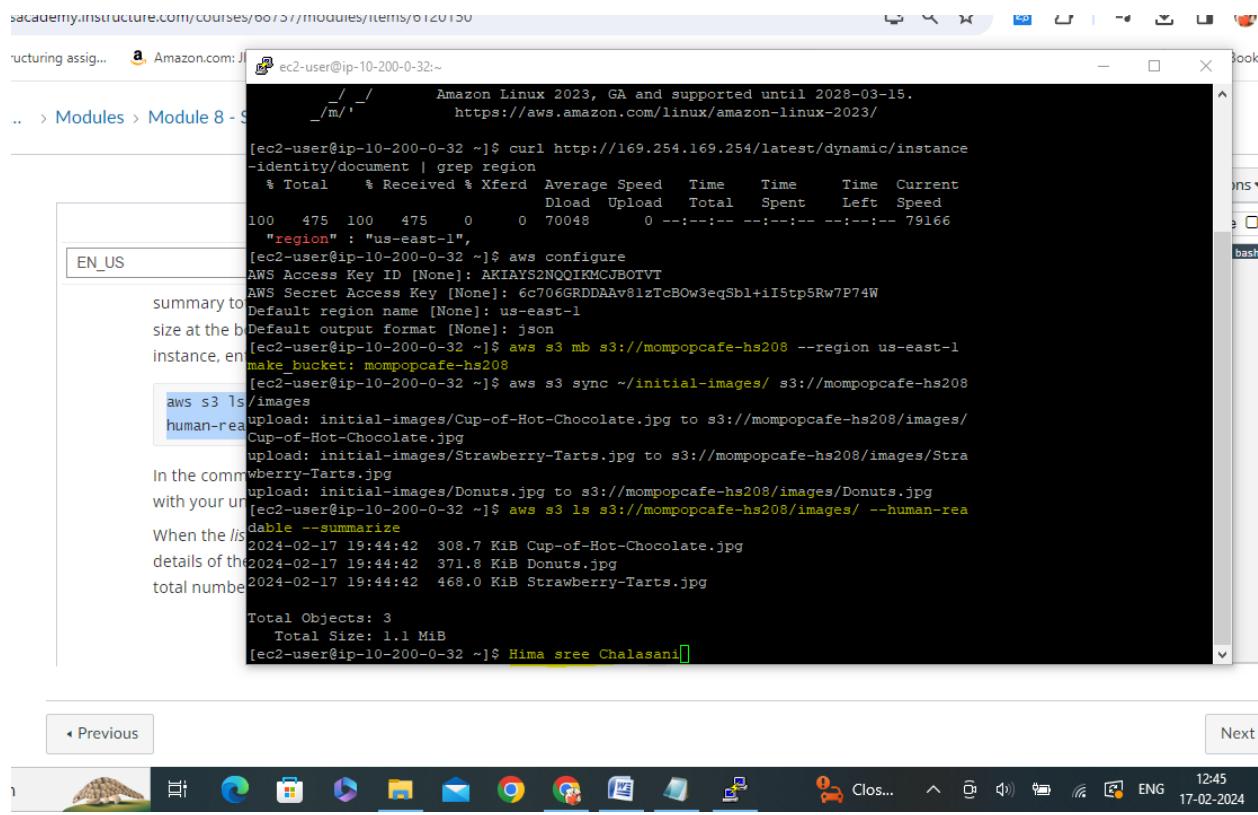
Type here to search

```
instructure [ec2-user@ip-10-5-0-198:~]
  "Tags": []
}
[ec2-user@ip-10-5-0-198 ~]$ ^C
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 wait snapshot-completed --snapshot-id snap-08d6flc24b009298a
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 start-instances --instance-ids i-0b30de7809cb230cl
> Module
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-0b30de7809cb230cl",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 wait instance-running --instance-id i-0b30de7809cb230cl
[ec2-user@ip-10-5-0-198 ~]$ aws ec2 start-instances --instance-ids i-0b30de7809cb230cl
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 16,
                "Name": "running"
            },
            "InstanceId": "i-0b30de7809cb230cl",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
[ec2-user@ip-10-5-0-198 ~]$ Hima sree Chalasani
```

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

1. Activate Versioning for the Bucket: Ensure that versioning is turned on for the bucket to effectively utilize Amazon's s3api commands.
2. Synchronize Local Files with Amazon S3: Use the `aws s3 sync` command to match your local files with those on Amazon S3.
3. Remove Local File: Delete the file from your local storage device.
4. Restore Deleted File from Amazon S3: Utilize `aws s3api list-object-versions` and `aws s3api get-object` commands to retrieve a previous version of a deleted file from Amazon S3. Use AWS S3 Sync to restore the file. Sample files are available for download. Access the processor instance and download the sample file by following the link provided in the lab instructions.
5. Extract Downloaded File: Execute the `unzip files.zip` command in the terminal to decompress the downloaded file.
6. Activate Versioning: Enable versioning using the command `aws s3api put-bucket-versioning --s3mmr --versioning-configuration Status=Enabled`.
7. Synchronize Folder Content with S3 Bucket: Match the content of the folder with the S3 bucket using `aws s3 sync files s3://s3mmr/files/`.
8. Delete Local File: Remove the file from local storage using the command `rm files/file1.txt`.

9. Delete File from S3 Using --delete Option: Use `aws s3 sync` with the `--delete` option to erase the same file from the server: `aws s3 sync files s3://S3-BUCKET-NAME/files/ --delete`.
10. Confirm Remote Deletion: Verify the deletion of the file from the server by running the command `ls s3 aws s3://smmr/files/`.
11. Synchronize Old File Version with Amazon S3: Match the previous version of the file with Amazon S3 using the command `aws s3api get-object --bucket smmr --key files/file1.txt --version-id VERSION-ID files/file1.txt`.
12. Re-Synchronize File or Folder to Amazon S3: Execute `aws s3 sync files s3://S3-BUCKET-NAME/files/` to re-synchronize the content of a file or folder with Amazon S3.



The screenshot shows a Linux terminal window titled "ec2-user@ip-10-200-0-32:~". The terminal displays the following AWS commands and their output:

```

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

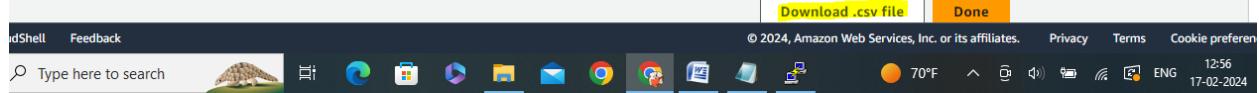
[ec2-user@ip-10-200-0-32 ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
  % Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload Total   Spent    Left Speed
100  475  100  475    0     0  70048      0 --:--:-- --:--:-- 79166
"region" : "us-east-1",
[ec2-user@ip-10-200-0-32 ~]$ aws configure
AWS Access Key ID [None]: AKIAYS2NQQIKMCUBOTV7
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-10-200-0-32 ~]$ aws s3 mb s3://mompopcafe-hs208 --region us-east-1
make_bucket: mompopcafe-hs208
[ec2-user@ip-10-200-0-32 ~]$ aws s3 sync ~/initial-images/ s3://mompopcafe-hs208
upload: initial-images/Cup-of-Hot-Chocolate.jpg to s3://mompopcafe-hs208/images/
Cup-of-Hot-Chocolate.jpg
upload: initial-images/Strawberry-Tarts.jpg to s3://mompopcafe-hs208/images/Strawberry-Tarts.jpg
upload: initial-images/Donuts.jpg to s3://mompopcafe-hs208/images/Donuts.jpg
[ec2-user@ip-10-200-0-32 ~]$ aws s3 ls s3://mompopcafe-hs208/images/ --human-readable --summarize
2024-02-17 19:44:42  308.7 KiB Cup-of-Hot-Chocolate.jpg
2024-02-17 19:44:42  371.8 KiB Donuts.jpg
2024-02-17 19:44:42  468.0 KiB Strawberry-Tarts.jpg

Total Objects: 3
  Total Size: 1.1 MiB
[ec2-user@ip-10-200-0-32 ~]$ Hima sree Chalasani

```

The terminal window is part of a web-based interface, with the URL visible at the top: <https://academy.instructure.com/courses/0075/modules/items/012150>. Navigation buttons for "Previous" and "Next" are visible below the terminal window. The system tray at the bottom right shows the date and time as "17-02-2024 12:45" and the language as "ENG".

The screenshot shows the AWS IAM Access Key creation page. At the top, there's a green header bar with the message "Access key created" and a note: "This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time." Below this, there are two tabs: "Access key" (selected) and "Secret access key". Under "Access key", the access key ID is displayed as "AKIAY52NQQIKAC6Q2INH" and the secret access key is shown as a long string of asterisks followed by a "Show" button. To the left, there are sections for "Step 2 - optional" (Set description tag) and "Step 3" (Retrieve access keys). On the right, there's a section titled "Access key best practices" with a bulleted list: "Never store your access key in plain text, in a code repository, or in code.", "Disable or delete access key when no longer needed.", "Enable least-privilege permissions.", and "Rotate access keys regularly.". A link to "best practices for managing AWS access keys" is provided at the bottom of this section.



The screenshot shows a Gmail inbox with one message from "AWS Notifications <no-reply@sns.amazonaws.com>" with the subject "Amazon S3 Notification". The message content is a JSON object representing an S3 event: {"Service": "Amazon S3", "Event": "s3:TestEvent", "Time": "2024-02-17T20:27:43.177Z", "Bucket": "mompopcafe-hs208", "RequestId": "CZRHZHD3CDCAVG7N", "HostId": "bYGlz5S0gXtbtLyC3rzHsFHQLd/CBo5pDDN+7SuQtq0eiHgSkUQnYTrnPwRExDxUUgqJ1nr8s4="}. Below the message, there's a link to unsubscribe: "https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:590183694868:s3NotificationTopic:f73a4124-9e9a-43ba-b483-9e8c1686cd40&Endpoint=chalaanihimasree@gmail.com". A note at the bottom says "Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support".



```
ec2-user@ip-10-200-0-32:~$ aws configure
AWS Access Key ID [*****OTVT]: AKIAYS2NQQIKAC6Q2INH
AWS Secret Access Key [*****P74W]: 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
Default region name [us-east-1]:
Default output format [json]: json
[ec2-user@ip-10-200-0-32 ~]$ aws s3api put-object --bucket mompopcafe-hs208 --key images/Caramel-Delight.jpg --body ~/new-images/Caramel-Delight.jpg
{
    "ETag": "\"3lac30da619244b0ce786f106e4f3df7\"",
    "ServerSideEncryption": "AES256"
}
the CLI [ec2-user@ip-10-200-0-32 ~]$ Hima sree Chalasani
```

In the code editor, there is a snippet of AWS CLI command execution. It shows the configuration of AWS credentials and the upload of a file named 'Caramel-Delight.jpg' to an S3 bucket named 'mompopcafe-hs208'. The file is uploaded in JSON format. The command 'aws s3api put-object' is used with parameters: --bucket, --key, and --body. The file is located at '~/new-images/Caramel-Delight.jpg'. The response includes the ETag and ServerSideEncryption details.

I08. Check th
subscri

us

Next



```
es > Module ec2-user@ip-10-200-0-32:~
[ec2-user@ip-10-200-0-32 ~]$ aws configure
AWS Access Key ID [*****OTVT]: AKIAYS2NQQIKAC6Q2INH
AWS Secret Access Key [*****P74W]: 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
Default region name [us-east-1]:
Default output format [json]: json
[ec2-user@ip-10-200-0-32 ~]$ aws s3api put-object --bucket mompopcafe-hs208 --key images/Caramel-Delight.jpg --body ~/new-images/Caramel-Delight.jpg
{
    "ETag": "\"3lac30da619244b0ce786f106e4f3df7\"",
    "ServerSideEncryption": "AES256"
}
[ec2-user@ip-10-200-0-32 ~]$ aws s3api get-object --bucket mompopcafe-hs208 --key images/Donuts.jpg Donuts.jpg
{
    "AcceptRanges": "bytes",
    "ContentType": "image/jpeg",
    "LastModified": "Sat, 17 Feb 2024 19:44:42 GMT",
    "ContentLength": 380753,
    "ETag": "\"405b0bcc53cb5ab713c967dc1422b4f4\"",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
The Amazon
Mom & Pop
company. P
he receives
uploaded. H
An error occurred (AccessDenied) when calling the PutObjectAcl operation: Access Denied
exchange d [ec2-user@ip-10-200-0-32 ~]$ Hima sree Chalasani
```

In this session, the user attempts to download a file ('Donuts.jpg') from the S3 bucket. The command 'aws s3api get-object' is used with parameters: --bucket, --key, and a local file path. The response includes metadata such as AcceptRanges, ContentType, LastModified, ContentLength, ETag, ServerSideEncryption, and Metadata. However, the user receives an 'Access Denied' error when trying to set the object's ACL using 'aws s3api put-object-acl' due to insufficient permissions.

ious

Next



How to create a S3 bucket via the CLI

1. Set up and customize the AWS CLI on your personal computer.
2. Select a name for the bucket that is unique worldwide.
3. Decide on the AWS region where the bucket will reside.
4. Establish the S3 bucket employing the aws s3api create-bucket command: 'aws s3 create-bucket --bucket \$bucket_name --region \$aws_region'.
5. Confirm the creation of the bucket by running: 'aws s3 ls'

Modules > Module 2 - T... > Activity 2 - Create a Website on S3

The screenshot shows a terminal window titled 'Terminal' with the following content:

```
[ec2-user@ip-10-200-0-66 ~]$ aws configure
AWS Access Key ID [None]: AKIAYAOTAFVZOVNYYAVZ
AWS Secret Access Key [None]: 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
Default region name [None]: eu-west-2
Default output format [None]: json
```

A tip box is overlaid on the terminal window:

Task 3: Create a bucket using the AWS CLI

Tip: In this activity, you will use the AWS CLI to run the `aws s3api create-bucket` command, and other times you will use the AWS Management Console. `s3` commands are built on top of `s3api` commands, so you can learn more about them by reading the `s3api` documentation.

Below the terminal window, there is a task bar with various icons and system status indicators:

◀ Previous Next ▶

28°C 23:44
11-01-2024

N_US

```
[ec2-user@ip-10-200-0-66 ~]$ aws s3api create-bucket --bucket hsree208 --region eu-west-2 --create-bucket-configuration LocationConstraint=eu-west-2
{
    "Location": "http://hsree208.s3.amazonaws.com/"
}
[ec2-user@ip-10-200-0-66 ~]$ Hima sree Chalasani
```

Task 3: Create a bucket using the AWS CLI

22. Open the AWS CLI terminal window.

Tip: In this task, you will use the AWS CLI command-line interface. The AWS CLI is a command-line interface that allows you to interact with AWS services from the command line. It is available for both Windows and Linux operating systems. You can find it in the AWS Lambda console under the "Tools" tab.

23. Use the `aws s3` command to create a new bucket in AWS S3. The bucket name must be unique and can contain up to three random lowercase letters followed by three random numbers. This will create a good bucket name for your website.



Modules > Module 2 - T... > Activity 2 - Create a Website on S3

EN_US

```
[ec2-user@ip-10-200-0-66:~/sysops-activity-files]$ aws s3api put-bucket-ownership-controls \
> --bucket hsree208 \
> --ownership-controls "Rules=[{ObjectOwnership=BucketOwnerPreferred}]"
[ec2-user@ip-10-200-0-66:~/sysops-activity-files]$ aws s3api put-public-access-block \
> --bucket hsree208 \
> --public-access-block-configuration "BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=false,RestrictPublicBuckets=false"
[ec2-user@ip-10-200-0-66:~/sysops-activity-files]$ aws s3 website s3://hsree208/ --index-document index.html
[ec2-user@ip-10-200-0-66:~/sysops-activity-files]$ aws s3 cp . s3://hsree208/ --recursive --acl public-read
upload: css/styles.css to s3://hsree208/css/styles.css
upload: ./index.html to s3://hsree208/index.html
upload: images/Mom-&-Pop-Coffee-Shop.png to s3://hsree208/images/Mom-&-Pop-Coffee-Shop.png
upload: images/Mom-&-Pop.png to s3://hsree208/images/Mom-&-Pop.png
upload: images/Strawberry-&-Blueberry-Tarts.png to s3://hsree208/images/Strawberry-&-Blueberry-Tarts.png
upload: images/Cookies.png to s3://hsree208/images/Cookies.png
upload: images/Coffee-and-Pastries.png to s3://hsree208/images/Coffee-and-Pastries.png
NOTE: upload: images/Cup-of-Hot-Chocolate.png to s3://hsree208/images/Cup-of-Hot-Chocolate.png
your account has been updated.
upload: images/Cake-Vitrine.png to s3://hsree208/images/Cake-Vitrine.png
upload: images/Strawberry-Tarts.png to s3://hsree208/images/Strawberry-Tarts.png
[ec2-user@ip-10-200-0-66:~/sysops-activity-files]$ aws s3 ls hsree208
aws s3 ls
--bucket hsree208
--owner-id 00000000000000000000000000000000
2024-01-12 07:17:27      3020 index.html
refererr [ec2-user@ip-10-200-0-66:~/sysops-activity-files]$
```

Task 6: Set ownership controls by using AWS CLI

33. Run the following AWS CLI commands and ACloud Academy will upload the files to the bucket.

NOTE: If you are using a different AWS account, you need to update the account ID in the command.



How to add an event notification to a S3 bucket

1. Generate an SNS topic.
2. Retrieve the Amazon Resource Name (ARN) of the topic.
3. Grant permission to enable S3 to publish messages to the SNS topic.
4. Configure the event notification settings for the S3 bucket.

The screenshot shows the 'Create subscription' step in the AWS Lambda console. The 'Topic ARN' field contains 'arn:aws:sns:us-east-1:590183694868:s3NotificationTopic'. The 'Protocol' dropdown is set to 'Email'. The 'Endpoint' field contains 'chalasanihimasree@gmail.com'. A note at the bottom states: 'After your subscription is created, you must confirm it.' The browser status bar at the bottom indicates the date as 17-02-2024.

The screenshot shows an email from 'AWS Notifications <no-reply@sns.amazonaws.com>' to 'chalasanihimasree@gmail.com' with the subject 'Amazon S3 Notification'. The email body contains a JSON object with details about the test event: {"Service": "Amazon S3", "Event": "s3:TestEvent", "Time": "2024-02-17T20:27:43.177Z", "Bucket": "mompopcafe-hs208", "RequestId": "CZRHZHD3CDCAVG7N", "HostId": "bYGkyZ5S0gXCtLjC3rHsFHQLd/CBo5pDDN+7SuQtq0eHgSkUQnYTrnPwRExDUJgqJ1m024="}. It includes unsubscribe and support links. The browser status bar at the bottom indicates the date as 17-02-2024.



How to install the CloudWatch Agent

1. Access the Systems Manager console.
2. Navigate to the Run Command section and opt for AWS-ConfigureAWSPackage.
3. Customize the installation parameters for the CloudWatch Agent (Action, Name, Version).
4. Designate the specific EC2 instance as the target.
5. Commence the installation process for the CloudWatch Agent.
6. Monitor the installation progress and await confirmation of success.
7. Validate the successful installation through command output.
8. Review instance details and expand the status of the command.
9. Ensure the appearance of the "Successfully installed CloudWatchAgent" message.
10. Establish the configuration of the CloudWatch Agent by generating a config file within Parameter Store.
11. Define the desired logs and metrics for collection within the config file.

The screenshot shows the AWS Systems Manager Run Command interface. At the top, it displays the Command ID: fb8d1914-5a4b-4efd-984a-db6724f82c4f. Below the command ID are four buttons: 'Cancel command' (disabled), 'Rerun', and 'Copy to new'. A 'Command status' section shows the overall status as 'Success' (green checkmark) and detailed status as 'Success' (green checkmark). It also lists the number of targets (1), completed tasks (1), errors (0), and delivery timed outs (0). The 'Targets and outputs' section shows a single target: Instance ID i-0d90a61b0225d8180, Instance name ip-10-0-0-245.ec2.internal, Status as 'Success' (green checkmark), Detailed Status as 'Success' (green checkmark), and Start time as Sat, 02 Mar 2024 19:17:45 GMT. There is a 'View output' button next to the search bar. At the bottom, there is a section titled 'CloudWatch alarm' with a disclosure arrow. The footer of the browser window includes the AWS logo, a toolbar with various icons, and system status information like temperature (72°F), battery level, and network connectivity.

Output on i-0d90a61b0225d8180

Step 1 - Command description and status

Status	Detailed status	Response code	Step name	Start time	Finish time
Success	Success	0	ControlCloudWatchAgentWindows	Sat, 02 Mar 2024 19:17:45 GMT	Sat, 02 Mar 2024 19:17:45 GMT

▶ Output

▶ Error

Step 2 - Command description and status

Status	Detailed status	Response code	Step name	Start time	Finish time
Success	Success	0	ControlCloudWatchAgentLinux	Sat, 02 Mar 2024 19:17:45 GMT	Sat, 02 Mar 2024 19:17:46 GMT

Search

72°F ENG 12:19 02-03-2024

How to create a CloudWatch Events/CloudWatch EventBridge notification rule

1. Execute aws configure to configure AWS CLI credentials.
2. Utilize the AWS CloudWatch Events put-rule command:

```
aws events put-rule --name RuleName --event-pattern '{ "source": [ "aws.ec2"], "detail-type": [ "EC2 Instance State-change Notification"], "resources": ["arn:aws:ec2:Region:AccountID:instance/InstanceID"], "detail": { "state": [ "stopped"] } }'
```
3. Execute the AWS CloudWatch Events put-targets command:

```
aws events put-targets --rule RuleName --targets Id=1,Arn=arn:aws:lambda:Region:AccountID:function:LambdaFunctionName
```
4. Validate the rule and associated target by checking the CloudWatch Events Console or executing aws events list-targets-by-rule --rule RuleName.

The screenshot shows the AWS EventBridge Create rule wizard interface. It consists of five main steps:

- Step 1: Define rule detail**: Shows a table with the following data:

Rule name	Status	Event bus
Instance_Stopped_Terminated	Enabled	default
Description		Rule type
		Standard rule
- Step 2: Build event pattern**: Displays an event pattern JSON snippet with a "Copy" button:


```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"],
4   "detail": {
5     "state": ["stopped", "terminated"]
6   }
7 }
```
- Step 3: Select target(s)**: Shows a table of targets:

Details	Target Name	Type	Arn	Input	Role
▶ Default_CloudWatch_Alarms_Topic	Default_CloudWatch_Alarms_Topic	SNS topic	arn:aws:sns:us-east-1:938486009943:Default_CloudWatch_Alarms_Topic	Matched event	-
- Step 4: Configure tag(s)**: Shows a table for tags with a "Create rule" button at the bottom right:

Tags (0)	
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.	
Key	Value
No tags associated with this resource.	

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

1. Retrieve the pre-assembled Stopinator script from the AWS GitHub repository:
2. curl -O https://raw.githubusercontent.com/awsdocs/aws-doc-sdk-examples/main/python/example_code/cloudwatch/stopinator.py
3. Grant executable permissions to the script: chmod +x stopinator.py
4. Execute the script: ./stopinator.py HimasreeChalasani

The screenshot shows a terminal window titled "Module" with the following content:

```
[ec2-user@ip-10-200-0-32 ~]$ aws configure
AWS Access Key ID [*****OTVT]: AKIAYS2NQQIKAC6Q2INH
AWS Secret Access Key [*****P74W]: 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
Default region name [us-east-1]:
Default output format [json]: json
[ec2-user@ip-10-200-0-32 ~]$ aws s3api put-object --bucket mompopcafe-hs208 --key images/Caramel-Delight.jpg --body ~/new-images/Caramel-Delight.jpg
{
    "ETag": "\"3lac30da619244b0ce786f106e4f3df7\"",
    "ServerSideEncryption": "AES256"
}
the CLI [ec2-user@ip-10-200-0-32 ~]$ Hima sree Chalasani
```

In the code block, the command `aws s3api put-object` is used to upload a file named "Caramel-Delight.jpg" to an S3 bucket named "mompopcafe-hs208". The file is located at `~/new-images/Caramel-Delight.jpg`. The response includes the ETag and ServerSideEncryption details.

Below the terminal window, the taskbar shows various application icons and system status indicators. The system tray displays the date and time as 17-02-2024, 13:32, with a temperature of 71°F.

How to resize an EC2 instance using the AWS CLI

1. Utilize `aws configure` to configure AWS CLI credentials.
2. Execute the AWS EC2 `modify-instance-attribute` command:
`aws ec2 modify-instance-attribute --instance-id InstanceID --instance-type NewInstanceType`
3. Substitute `InstanceID` with the EC2 instance ID that requires resizing.
4. Replace `NewInstanceType` with the desired instance type.
5. Validate the modifications in the AWS EC2 Console or execute `aws ec2 describe-instances --instance-ids InstanceID` to verify the instance type.
6. Optionally, it may be necessary to halt and restart the instance for the changes to take effect:
7. `aws ec2 stop-instances --instance-ids InstanceID`
`aws ec2 start-instances --instance-ids InstanceID`

The screenshot shows a terminal window titled 'instructure' with the command prompt '[ec2-user@ip-10-5-0-198:~]'. The window displays the following sequence of events:

- A CloudFormation template is being executed, showing 'StartingInstances' with 'InstanceState' pending.
- The user runs 'aws ec2 wait snapshot-completed --snapshot-id snap-08d6f1c24b009298a' to check for completed snapshots.
- The user runs 'aws ec2 start-instances --instance-ids i-0b30de7809cb230cl' to start the instances.
- The 'InstanceState' changes from 'pending' to 'running'.
- The user runs 'aws ec2 wait instance-running --instance-id i-0b30de7809cb230cl' to ensure the instances are running.
- The user runs 'aws ec2 start-instances --instance-ids i-0b30de7809cb230cl' again, which is unnecessary as the instances are already running.

The terminal ends with the message '[ec2-user@ip-10-5-0-198 ~]\$ Hima sree Chalasani'.

The desktop taskbar at the bottom shows various application icons and system status indicators, including the date and time (14-02-2024, 19:28), battery level (61%), and network connection.

How to detect drift in a CloudFormation template

1. Utilize aws configure to establish AWS CLI credentials.
2. Execute the AWS CloudFormation detect-stack-drift command: `aws cloudformation detect-stack-drift --stack-name StackName`
3. Keep track of the drift detection status: `aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-id DriftDetectionId`
4. Review the drift results: `aws cloudformation describe-stack-resource-drifts --stack-name StackName`
5. Alternatively, the drift detection process can be automated through CloudWatch Events or AWS Config, if desired.

```

ec2-user@cli-host:~$ aws cloudformation describe-stacks \
--stack-name myStack \
--query 'StackResources[*].[ResourceType,ResourceStatus]' \
--output table
[ec2-user@cli-host ~]$ aws cloudformation describe-stacks \
--stack-name myStack \
--output table
+
+-----+-----+
|             | DescribeStacks
+-----+-----+
|             |     +-----+-----+
|             |     |   Stacks
|             |     +-----+-----+
|             |     |     +-----+-----+
|             |     |     | CreationTime | 2024-02-23T04:59:47.849Z
|             |     |     | Description  | Lab template
|             |     |     | DisableRollback | False
|             |     |     | EnableTerminationProtection | False
|             |     |     | StackId      | arn:aws:cloudformation:us-east-1:211125697966:stack/myStack/5e3902f0-d208-11ee-a17c-0e150b3c5969
|             |     |     | StackName    | myStack
|             |     |     | StackStatus  | CREATE_COMPLETE
+-----+-----+
|             |     +-----+-----+
|             |     |   Capabilities
|             |     +-----+-----+
|             |     |     CAPABILITY_NAMED_IAM
+-----+-----+
|             |     +-----+-----+
|             |     |   DriftInformation
|             |     +-----+-----+
|             |     |     StackDriftStatus | NOT_CHECKED
+-----+-----+
|             |     +-----+-----+
|             |     |   Outputs
|             |     +-----+-----+
|             |     |     OutputKey          | OutputValue
|             |     +-----+-----+
|             |     |     BucketName        | mystack-mybucket-y56urvs1j:2f
|             |     |     PublicCIP         | 18.209.44.132
+-----+-----+
|             |     +-----+-----+
|             |     |   Parameters
|             |     +-----+-----+
|             |     |     ParameterKey | ParameterValue | ResolvedValue
|             |     +-----+-----+
|             |     |     KeyName       | vockey
|             |     |     LabVpcCidr  | 10.0.0.0/20
|             |     |     PublicSubnetCidr | 10.0.0.0/24
|             |     |     AmazonLinuxAMIID | /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
+-----+-----+
[ec2-user@cli-host ~]$ Hima sree Chalasani

```



61°F 22:02 22-02-2024

How to create an Amazon Athena table

1. Access the Amazon Athena Console.
2. Choose the Database: From the left panel, select the database in which we intend to establish the table.
3. Execute a CREATE TABLE Query:
Example:

```
CREATE TABLE IF NOT EXISTS table_name ( column1_name datatype1,
column2_name datatype2, ... );
```
4. Initiate the execution of the CREATE TABLE statement by clicking the "Run Query" button.
5. Upon completion of the query execution, confirm the creation of the table by inspecting the "Tables" tab within the Athena console.

Athena

Search [Alt+S] N. Virginia vocabs/user2718047=Chalasani_Hima_Sree @ 5332-6736-5860

Editor Recent queries Saved queries Settings Workgroup primary

Data Data source AwsDataCatalog Database default Tables and views Create Filter tables and views Tables (1) cloudtrail_logs_monitoring7777 eventversion string useridentity struct<type:string,principalId:string,arn:string,accountId:string,invokedBy:string,accessKeyId:string,userName:string,sessionContext:struct<attributes:struct<mfaAuthenticated:string,creationDate:string>,sessionId:struct<type:string,principalId:string,arn:string,accountId:string,username:string>,ec2RoleDelivery:string,webIdFederationData:map<string, string>>>

Query 1 :
1 SELECT *
2 FROM cloudtrail_logs_monitoring7777
3 LIMIT 5

SQL Ln 2, Col 36 Run again Explain Cancel Clear Reuse query results

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 72°F ENG 14:24 02-03-2024

Athena

Search [Alt+S] N. Virginia vocabs/user2718047=Chalasani_Hima_Sree @ 5332-6736-5860

Editor Recent queries Saved queries Settings Workgroup primary

Data Data source AwsDataCatalog Database default Tables and views Create Filter tables and views Tables (1) cloudtrail_logs_monitoring7777 eventversion string useridentity struct<type:string,principalId:string,arn:string,accountId:string,invokedBy:string,accessKeyId:string,userName:string,sessionContext:struct<attributes:struct<mfaAuthenticated:string,creationDate:string>,sessionId:struct<type:string,principalId:string,arn:string,accountId:string,username:string>,ec2RoleDelivery:string,webIdFederationData:map<string, string>>>

Query results Query stats

Completed Time in queue: 98 ms Run time: 629 ms Data scanned: 22.51 KB

Results (5) Copy Download results

Search rows

#	eventversion	useridentity
1	1.09	{type=AWSService, principalId=null, arn=null, accountId=n}

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 72°F ENG 14:24 02-03-2024

The screenshot shows the AWS Athena console with a completed query execution. The results table has three columns: #, eventversion, and useridentity. The data is as follows:

#	eventversion	useridentity
1	1.09	{type=AWSService, principalid=null, arn=null, accountid=null}
2	1.08	{type=AssumedRole, principalid=AROAXYKJWH7SPNJL4GCP}
3	1.10	{type=AssumedRole, principalid=AROAXYKJWH7SHO6PQGP}
4	1.08	{type=AssumedRole, principalid=AROAXYKJWH7SPNJL4GCP}
5	1.09	{type=AWSService, principalid=null, arn=null, accountid=null}

How to manually review access logs to find anomalous user activity

1. Determine the location of access logs.
2. Understand the format of access logs.
3. Ensure that logs are collected and stored in a centralized repository for streamlined analysis. Solutions such as AWS CloudWatch Logs, Elasticsearch, or Splunk can facilitate this process.
4. Establish a baseline representing typical user behavior. Familiarize yourself with common access patterns, IP addresses, user agents, and standard actions.
5. Regularly monitor access logs and configure alerts for detecting specific events or irregular patterns suggestive of anomalous behavior.
6. Be vigilant for any unusual patterns within the access logs.
7. Scrutinize user agents for anomalies, as malicious entities might utilize uncommon or altered user agents.
8. Investigate instances of abnormal access patterns by delving into the corresponding logs. Identify the impacted user accounts and associated resources.
9. Correlate access logs with other log types, such as authentication or system logs, to gain a comprehensive understanding of user activities.
10. Incorporate threat intelligence feeds to recognize IP addresses or user agents linked with documented malicious behavior.
11. Deploy User Behavior Analytics (UBA) tools to automate the detection of aberrant user actions using machine learning algorithms.
12. Document and report any identified findings. Maintain an incident response plan to effectively manage confirmed security incidents.

```

[ec2-user@web-server:~/ctraillogs/AWSLogs/533267365860/CloudTrail/us-east-1/2024/03/02
    ]
    "responseElements": null,
    "sharedEventID": "e10122b5-a8c7-4de6-9273-cd7d940c1d78",
    "sourceIPAddress": "cloudtrail.amazonaws.com",
    "userAgent": "cloudtrail.amazonaws.com",
    "userIdentity": {
        "invokedBy": "cloudtrail.amazonaws.com",
        "type": "AWSService"
    }
}
]
[ec2-user@web-server 02]$ ip=34.207.185.115
[ec2-user@web-server 02]$ for i in $(ls); do echo $i && cat $i | python -m json.tool | grep sourceIPAddress ; done
533267365860_CloudTrail_us-east-1_20240302T20452_AXf0ozp00xZ2uny6.json
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "54.212.125.109",
"sourceIPAddress": "54.212.125.109",
"sourceIPAddress": "54.212.125.109",
"sourceIPAddress": "75.237.68.36",
533267365860_CloudTrail_us-east-1_20240302T20452_BuTtXqS57VSVSQP.json
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "75.237.68.36",
533267365860_CloudTrail_us-east-1_20240302T20452_O0awusJjBy3rIIOK.json
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "cloudtrail.amazonaws.com",
"sourceIPAddress": "cloudtrail.amazonaws.com",
[ec2-user@web-server 02]$ Hima sree Chalasani

```

```

[ec2-user@web-server:~/ctraillogs/AWSLogs/533267365860/CloudTrail/us-east-1/2024/03/02
    ]
    "sourceIPAddress": "54.212.125.109",
    "sourceIPAddress": "54.212.125.109",
    "sourceIPAddress": "75.237.68.36",
533267365860_CloudTrail_us-east-1_20240302T20452_BuTtXqS57VSVSQP.json
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "75.237.68.36",
533267365860_CloudTrail_us-east-1_20240302T20452_O0awusJjBy3rIIOK.json
"sourceIPAddress": "34.207.185.115",
"sourceIPAddress": "75.237.68.36",
"sourceIPAddress": "cloudtrail.amazonaws.com",
"sourceIPAddress": "cloudtrail.amazonaws.com",
[ec2-user@web-server 02]$ for i in $(ls); do echo $i && cat $i | python -m json.tool | grep eventName ; done
533267365860_CloudTrail_us-east-1_20240302T20452_AXf0ozp00xZ2uny6.json
"eventName": "CreateSecurityGroup",
"eventName": "CreateSecurityGroup",
"eventName": "CreateSecurityGroup",
"eventName": "ListNotificationHubs",
"eventName": "CreateSecurityGroup",
"eventName": "CreateSecurityGroup",
"eventName": "CreateSecurityGroup",
"eventName": "GetParametersByPath",
"eventName": "GetParametersByPath",
"eventName": "GetParametersByPath",
"eventName": "DescribeMetricFilters",
"eventName": "DescribeMetricFilters",
"eventName": "DescribeMetricFilters",
"eventName": "ListNotificationHubs",
"eventName": "DescribeKeyPairs",
"eventName": "AssumeRole",
"eventName": "AssumeRole",
"eventName": "DescribeInstanceInformation",
533267365860_CloudTrail_us-east-1_20240302T20452_BuTtXqS57VSVSQP.json
"eventName": "GetInsightSelectors",
"eventName": "GetEventSelectors",
"eventName": "UpdateInstanceInformation",
"eventName": "DescribeAddresses",
533267365860_CloudTrail_us-east-1_20240302T20452_O0awusJjBy3rIIOK.json
"eventName": "CreateSecurityGroup",
"eventName": "DescribeAlarms",
"eventName": "GetBucketAcl",
"eventName": "GetBucketAcl",
[ec2-user@web-server 02]$ Hima sree Chalasani

```

To find using Athena:

1. Utilize the + icon to initiate a new tab for inserting a fresh query.
2. Implement filters on the results by employing a WHERE clause, such as WHERE eventsource = 'ec2.amazonaws.com'.

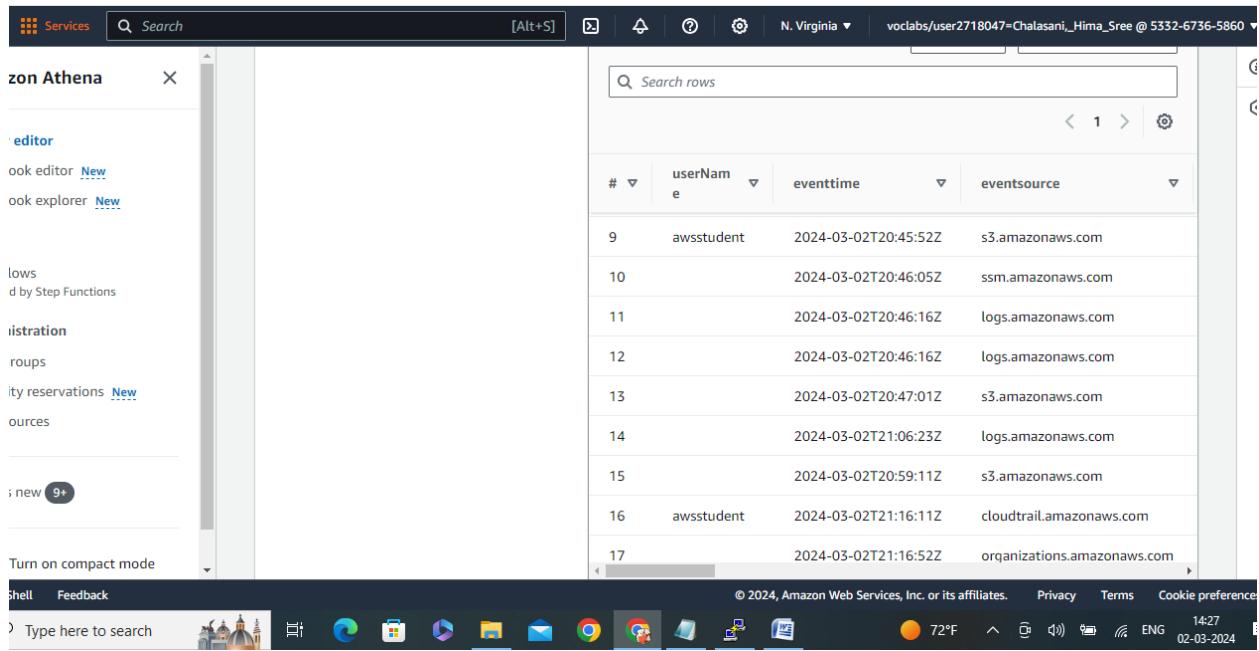
3. Examine the acquired data and enhance the query to detect occurrences of the term "security."
4. After analyzing the outcomes, construct the following query to retrieve all users active within the account throughout the day:
5.

```
SELECT userIdentity.userName, eventtime, eventsource, eventname, requestparameters
FROM cloudtrail_logs_monitoring7777
LIMIT 30
```

The screenshot shows the AWS Management Console with the Athena service selected. A modal window titled "Introducing cost-based optimizer" is open, explaining how it uses table and column statistics from AWS Glue to improve query execution plans. Below the modal, the "Query editor" interface is visible. It has tabs for "Editor", "Recent queries", "Saved queries", and "Settings". The "Workgroup" dropdown is set to "primary". There are two queries listed: "Query 1" and "Query 2". "Query 2" is the one being run, containing the provided SQL code. The "Data" section shows the configuration for the query, including "Data source" set to "AwsDataCatalog" and "Database" set to "default". A "Tables and views" section with a "Create" button is also present. The status bar at the bottom right shows the date and time as "02-03-2024 14:26".

The screenshot shows the results of the query execution. The "Results (30)" pane displays a table with 30 rows. The columns are labeled "#", "userName", "eventtime", and "eventsources". The data shows various user names and their corresponding event times and sources. The results table includes a header row and 30 data rows. The status bar at the bottom right shows the date and time as "02-03-2024 14:26".

#	userName	eventtime	eventsources
1		2024-03-02T21:18:45Z	redshift.amazonaws.com
2		2024-03-02T20:42:43Z	sts.amazonaws.com
3		2024-03-02T20:42:44Z	sts.amazonaws.com
4	awsstudent	2024-03-02T20:40:07Z	ec2.amazonaws.com
5		2024-03-02T20:44:04Z	monitoring.amazonaws.com
6		2024-03-02T20:44:45Z	s3.amazonaws.com
7		2024-03-02T20:44:48Z	s3.amazonaws.com
8		2024-03-02T20:53:48Z	redshift.amazonaws.com



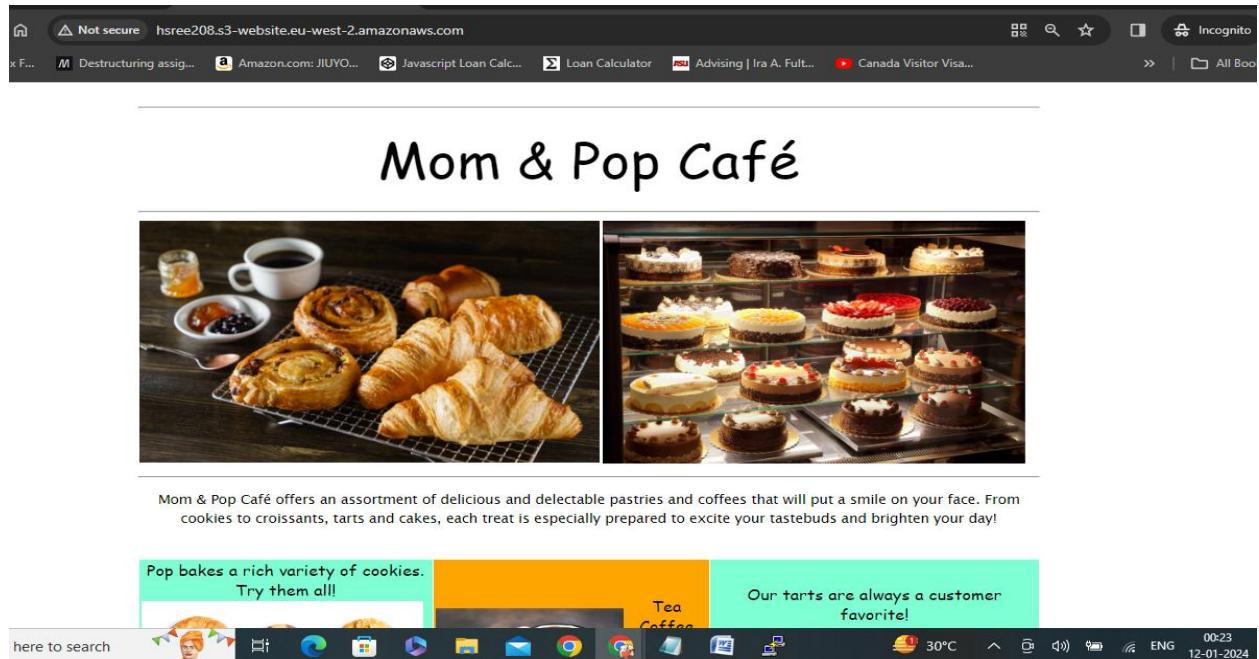
How to create a batch file to update the café website to change its colors

13. Launch a text editor, such as Notepad, and initiate a new file.
14. Navigate to the directory containing the website files.
15. Create the Batch File: In the text editor, generate a batch file with a .bat extension. For instance, name it update_colors.bat.
16. Modify the Batch File: Customize the batch file to incorporate commands for updating the CSS file. Below is an example of the batch file content:

```

batch
@echo off
echo Updating website colors...
set old_color=#ff0000
set new_color=#00ff00
set css_file=styles.css
powershell -Command "(Get-Content %css_file%) -replace '%old_color%', '%new_color%' | Set-Content %css_file%"
echo Colors updated successfully.

```
17. Save the batch file.
18. Execute the color update by either double-clicking the batch file or running it from the command prompt.



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

Create a Lambda Layer:

1. Prepare Layer Content: Arrange the content of the layer in a directory structure, such as:

```
my_layer/
  └── python
      └── lib
          └── python3.8
              └── site-packages
                  └── my_custom_code.py
```
2. Package the Layer Content: Navigate to the directory containing the layer content and execute the following command to zip it:

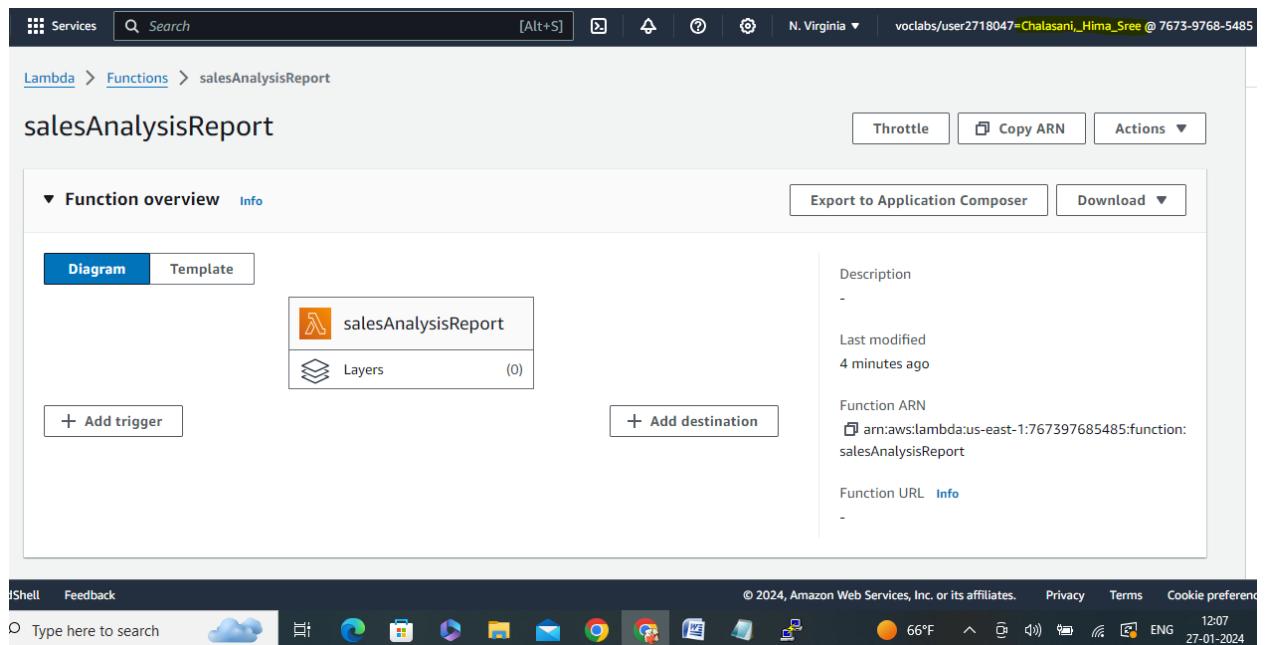
```
cd my_layer
cd my_layer
zip -r my_layer.zip
```
3. Document the Layer: Update the operations playbook with comprehensive information about the layer, including its purpose, dependencies, and any considerations regarding runtime compatibility.

Add the Layer to a Lambda Function:

6. Identify Target Function: Identify the Lambda function in the operations playbook where the layer will be incorporated.
7. Specify Layer Configuration: Document the specific details of adding the layer, including runtime compatibility and any versioning considerations.
8. Verify Lambda Configuration: Ensure that the Lambda function configuration in the playbook reflects the addition of the layer.
9. Document Changes: Update the operations playbook to incorporate details about the layer addition, such as the layer name, version, and a concise description.

Using the Layer in Lambda Function Code:

10. Code Integration: Document in the playbook how the Lambda function code should integrate with the layer, specifying any dependencies or modules to import from the layer.
11. Deployment Instructions: Offer clear instructions on deploying the updated Lambda function, emphasizing the importance of rigorous testing.
12. Testing Procedure: Document a step-by-step procedure for testing the Lambda function to ensure seamless access to the content of the layer.



How to create a Lambda function from a prebuilt package

19. Open the Lambda Console.
20. Select "Create function."
21. Opt for "Author from scratch."
22. Provide a name for the function and choose the appropriate runtime compatible with the prebuilt package, such as Python, Node.js, or Java.
23. Choose an existing role containing the necessary permissions or create a new role.
24. Proceed by clicking "Create function."
25. Within the "Function code" section, select "Upload a .zip file" under "Code entry type."
26. Upload the prebuilt package ZIP file from the local machine by clicking "Upload."
27. If the handler information differs from the default, specify it accordingly.
28. Include environment variables as required by the function.
29. Modify other settings like timeout, memory allocation, and VPC configuration if necessary.

30. Save the function configuration by clicking "Save."

31. Deploy the function by clicking "Deploy"

After uploading the code and conducting testing.

The screenshot shows the 'Create layer' configuration page in the AWS Lambda console. The top navigation bar includes 'Services', a search bar, and account information: N. Virginia, voclabs/user2718047=Chalasani_Hima_Sree @ 7673-9768-5485. Below the navigation is the 'Layer configuration' section with the following fields:

- Name:** pymysqlLibrary
- Description - optional:** PyMySQL 0.9.3 library modules - Hima sree Chalasani
- Upload a .zip file:** (radio button selected)
- Upload a file from Amazon S3:** (radio button unselected)
- Upload:** A button with a cloud icon.
- pymysql-1.1.0.zip** (highlighted in blue) - 107.63 KB
- Compatible architectures - optional:** Info
- CloudShell Feedback:** A search bar and a toolbar with various icons.
- © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences**
- 58°F ENG 10:34 27-01-2024**

The bottom section of the configuration page contains:

- pymysql-1.1.0.zip** (highlighted in blue) - 107.63 KB
- For files larger than 10 MB, consider uploading using Amazon S3.**
- Compatible architectures - optional:** Info
- Choose the compatible instruction set architectures for your layer.**
- x86_64
- arm64
- Compatible runtimes - optional:** Info
- Choose up to 15 runtimes.**
- Runtimes:** A dropdown menu showing 'Python 3.9' (highlighted in blue).
- Cancel** and **Create** buttons.
- Shell Feedback:** A search bar and a toolbar with various icons.
- © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences**
- 59°F ENG 10:38 27-01-2024**

pymysqlLibrary

Version details

Version	2	Version ARN	arn:aws:lambda:us-east-1:767397685485:layer:pymysqlLibrary:2	Description	PyMySQL 0.9.3 library modules - Hima sree Chalasani
Created	1 minute ago	License	-	Compatible runtimes	python3.9
Compatible architectures	-				

Versions | Functions using this version

CodeShell Feedback

The test event SARDETestEvent was successfully saved.

Test event Info

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

Create new event Edit saved event

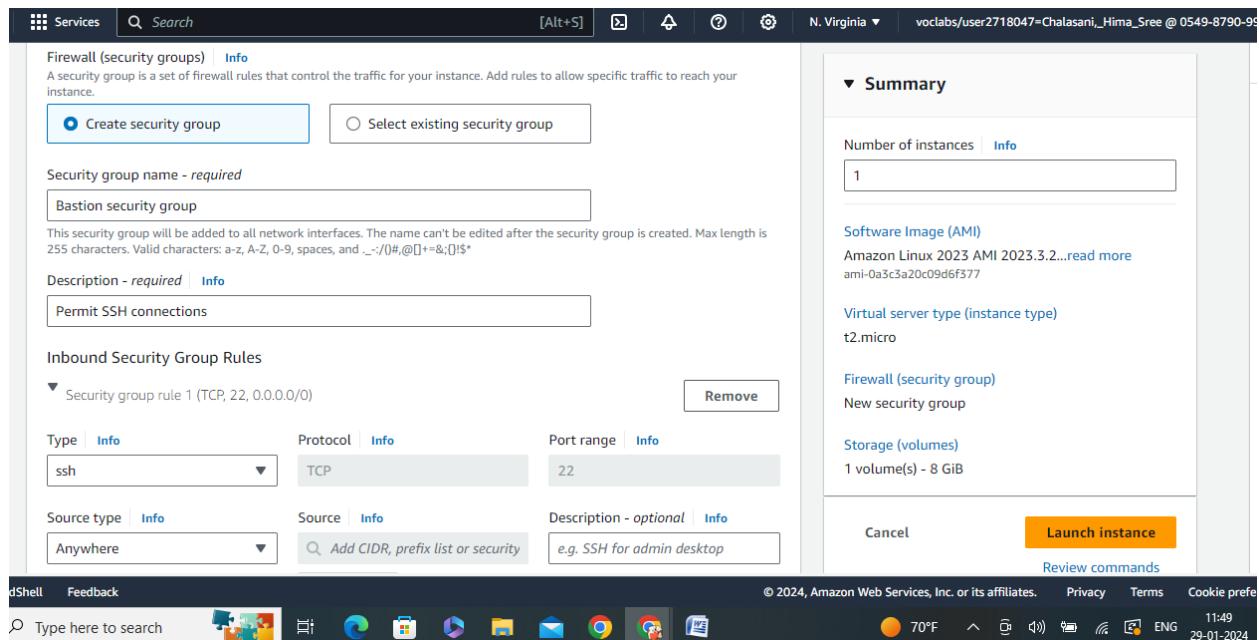
CloudShell Feedback

How to setup a VPC

1. Access the VPC service from the AWS Management Console.
2. Navigate to "VPCs" in the left sidebar and proceed to click on the "Create VPC" button.
3. Specify a name for the VPC and define its IPv4 CIDR block, for example, 10.0.0.0/16.
4. If necessary, enable DNS hostnames by selecting the corresponding option.
5. Go to "Subnets" in the left sidebar and click on "Create subnet."

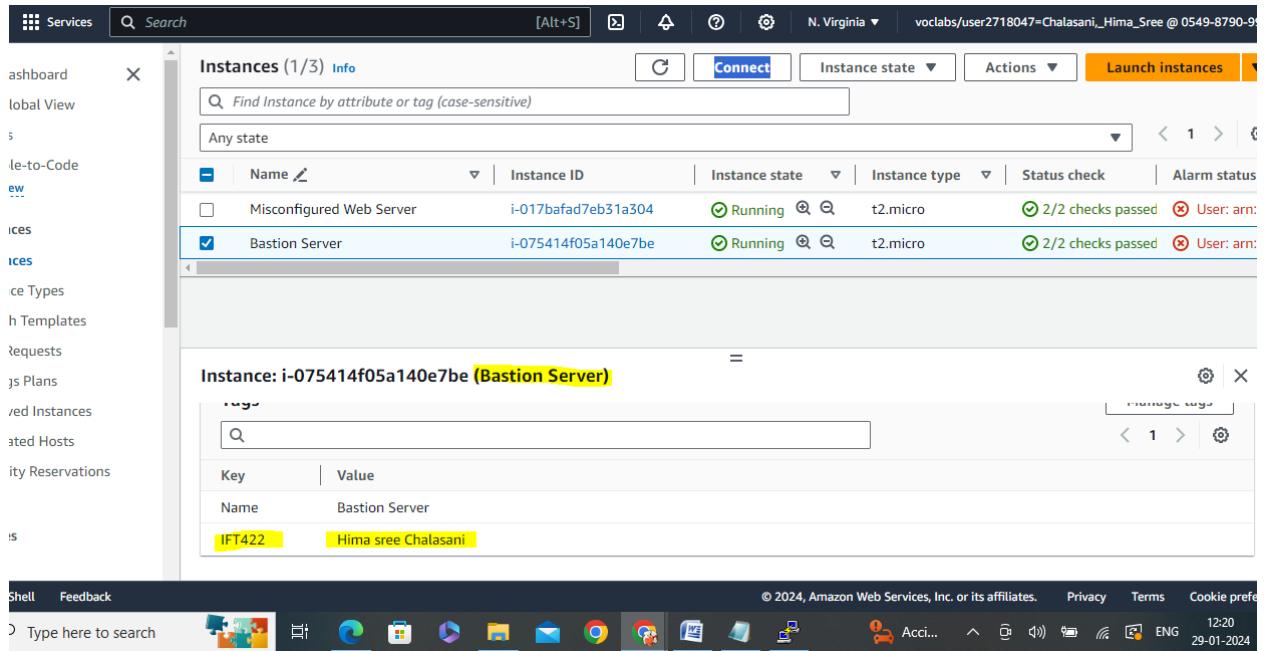
6. Associate the subnet with the VPC, assign a name, and specify its IPv4 CIDR block, such as 10.0.1.0/24.
7. Access "Internet Gateways" from the left sidebar.
8. Click on "Create internet gateway" and link it to the VPC.
9. Proceed to "Route Tables" in the left sidebar.
10. Choose the default route table and append a route for 0.0.0.0/0 directed to the internet gateway.
11. Access "Security Groups" from the left sidebar and establish security groups for instances.
12. Navigate to "Instances" in the left sidebar.
13. Launch instances into the public subnet and assign the designated security group.

The screenshot shows the AWS CloudFormation console interface. On the left, there's a navigation pane with 'CloudFormation' selected. The main area has tabs for 'Create New Stack', 'My Stacks', and 'Outputs'. Under 'Outputs', the 'HelloWorld' stack is listed with the value 'arn:aws:lambda:us-east-1:123456789012:function:HelloWorld'.



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

1. Initiate the launch of an EC2 instance within the public subnet.
2. Select an appropriate Amazon Machine Image (AMI) according to preference, such as Amazon Linux.
3. Configure the instance with a security group allowing inbound SSH (port 22) traffic from a specific IP address.
4. Enhance stability by considering the allocation of an Elastic IP to the bastion host, ensuring a consistent IP address.
5. Access the VPC dashboard and navigate to "Security Groups."
6. Establish a new security group dedicated to instances within the private subnet, permitting inbound SSH traffic solely from the bastion host.
7. Proceed to the VPC dashboard and locate "Route Tables."
8. Create a fresh route table tailored for the private subnet.
9. Associate the private subnet with the recently generated route table.
10. Integrate a route to the public subnet CIDR block via the bastion host.
11. Initiate the launch of instances within the private subnet.
12. Associate these instances with the security group established in step 6.



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

1. Access the IAM console, then select "Roles" from the left sidebar and proceed to click on "Create Role."
2. Opt for "Another AWS account" as the trusted entity type. Input the Account ID of the AWS account to which the user is associated.
3. Associate a policy with the role, providing the necessary permissions.
4. Within the "Trust relationships" tab, formulate a trust relationship policy permitting the IAM principal of the user to assume the role. For instance:

```
Json {
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::user-account-id:root"
    },
    "Action": "sts:AssumeRole"
}
```
5. Generate an IAM user account for the individual designated to assume the role.
6. Assign policies to the user, granting permissions to assume the role.
7. The IAM user can utilize either the AssumeRole API or the AWS Management Console to undertake the role.
8. If opting for the console, the user should navigate to the "Roles" section, select the designated role, and then click on "Switch Role."

Permissions Groups Tags (1) Security credentials Access Advisor

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

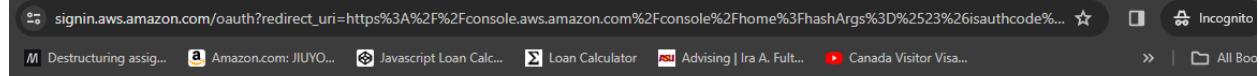
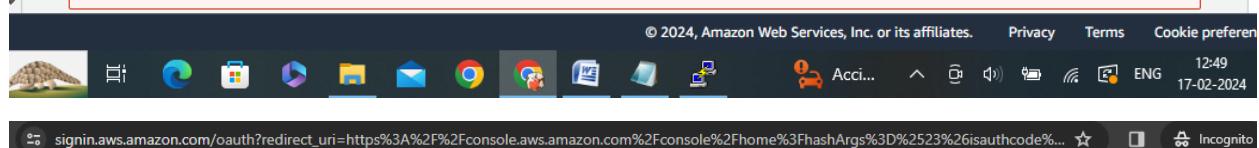
Filter by Type

Search All types

Policy name	Type	Attached via
AmazonS3ReadOnlyAccess	AWS managed	Directly

▶ Permissions boundary (not set)

You need permissions
User: arn:aws:sts::590183694868:assumed-role/voclabs/user2718047=Chalasani_Hima_Sree is not authorized to perform: access-analyzer>ListPolicyGenerations on resource: arn:aws:access-analyzer:us-east-1:590183694868:*



Sign in as IAM user

Account ID (12 digits) or account alias
590183694868

IAM user name
mediacouser

Password
.....

Remember this account

Sign in

Sign in using root user email
Forgot password?

INFRASTRUCTURE

Optimize cloud infrastructure costs and accelerate application innovation

[Learn more >](#)



You don't have permission to get the Bucket Versioning setting
Without s3:getBucketVersioning permission, we cannot determine if this delete action will add a delete marker to your objects or permanently delete them. [Learn more about Identity and access management in Amazon S3](#)

If a folder is selected for deletion, all objects in the folder will be deleted, and any new objects added while the delete action is in progress might also be deleted. If an object is selected for deletion, any new objects with the same name that are uploaded before the delete action is completed will also be deleted.
[Learn more](#)

Name	Type	Last modified	Size
Feedback	Image	2024-02-17T13:11:11Z	70F

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Type here to search 70F 13:11 ENG 17-02-2024

How to setup AWS Config to monitor resources

1. Sign in to the AWS Management Console and access the AWS Config service.
2. Select either "Get started" or "Set up AWS Config."
3. Specify the AWS resources to be monitored, such as All resources, and proceed by clicking "Next."
4. Choose the AWS Config rules to activate, whether AWS managed rules or custom rules, and adjust rule configurations as necessary.
5. Select an existing Amazon S3 bucket or create a new one to store AWS Config's configuration history. Configure bucket settings accordingly and proceed by clicking "Next."
6. Review the configured settings.
7. If all settings appear accurate, click "Confirm."
8. Proceed by clicking "Continue" to finalize the setup.
9. AWS Config will commence recording configuration alterations and evaluating rules.

Servicess Search [Alt+S] N. Virginia vocabs/user2718047=Chalasani_Hima_Sree @ 5332-6715-6046

AWS Config > Rules > Add rule

Step 1 Specify rule type

Step 2 Configure rule

Step 3 Review and create

Configure rule

Customize any of the following fields

Details	
Name	A unique name for the rule. 128 characters max. No special characters or spaces. ec2-volume-inuse-check
Description - optional	Describe what the rule evaluates and how to fix resources that don't comply. Checks whether EBS volumes are attached to EC2 instances.
Managed rule name	EC2_VOLUME_INUSE_CHECK



Servicess Search [Alt+S] N. Virginia vocabs/user2718047=Chalasani_Hima_Sree @ 5332-6715-6046

Select rule type

Step 3 Review and create

Add AWS managed rule
Customize any of the following rules to suit your needs.

Create custom Lambda rule
Create custom rules and add them to AWS Config. Associate each custom rule with an AWS Lambda function, which contains the logic that evaluates whether your AWS resources comply with the rule.

Create custom rule using Guard
Create custom rules using Guard Custom Policy that evaluates whether your AWS resources comply with the rule.

AWS Managed Rules (392)

Name	Labels	Supported evaluation mode	Description
required-tags	AWS	DETECTIVE	Checks whether your resources have the tags that you specify.

Cancel Next



The screenshot shows the AWS CloudWatch Metrics Insights interface. A search query has been run, resulting in a table of metrics. The table includes columns for Key, Type, Value, and Description. One row shows 'deleteOnTermination' set to 'boolean'. Below this, a 'Resources in scope' section lists EC2 volumes. The first volume, 'vol-0aa35e94a7a7a0dff', is marked as 'Compliant' with a green checkmark icon. The second volume, 'vol-04b395b81c1a67535', is marked as 'Noncompliant' with a red warning icon.

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

Adding Inbound Rules to Security Groups:

1. Access the EC2 Dashboard.
2. Navigate to "Security Groups" under "Network & Security" in the left sidebar.
3. Select the security group linked with the EC2 instances requiring adjustments.
4. Within the "Inbound rules" tab, click on "Edit inbound rules."
5. Specify rules for the desired protocol, port range, and source IP or security group.
6. Confirm the changes by clicking "Save rules."

The screenshot shows the 'Edit inbound rules' page for a specific security group. It displays two existing rules. Rule 1 allows 'All traffic' from the security group 'sg-06da990cedaf1459 - default'. Rule 2 allows 'All traffic' from the CIDR range '0.0.0.0/8'. Each rule has a 'Delete' button to its right. At the bottom of the page are buttons for 'Cancel', 'Preview changes', and 'Save rules'.

Adding Inbound Rules to Network ACLs:

1. Go to the VPC Dashboard.
2. Click on "Network ACLs" under "Security" in the left sidebar.
3. Choose the network ACL associated with the subnet containing the targeted EC2 instances.
4. Click "Edit inbound rules" in the "Inbound Rules" tab.
5. Define rules for the desired protocol, port range, and source IP or CIDR range.
6. As Network ACLs use rule numbers for sequencing, ensure correct ordering of rules.
7. Apply the modifications by clicking "Save rules."

The screenshot shows the 'Edit inbound rules' page for a specific Network ACL. The interface includes a breadcrumb navigation bar at the top: VPC > Network ACLs > acl-041b1f76cd31bd7a9 / MainACL > Edit inbound rules. Below this is a header with tabs for 'Edit inbound rules' and 'Info'. A note below the tabs states: 'Inbound rules control the incoming traffic that's allowed to reach the VPC.' The main content area is a table titled 'Rule number' with columns for 'Info', 'Type', 'Protocol', 'Port range', 'Source', 'Allow/Deny', and 'Info'. There are three rows of rules:

- Row 1: Rule number 100, Type 'All traffic', Protocol 'All', Port range 'All', Source '0.0.0.0/0', Allow/Deny 'Allow', Info 'Info'.
- Row 2: Rule number 200, Type 'Custom TCP', Protocol 'TCP (6)', Port range '0', Source '0.0.0.0/0', Allow/Deny 'Allow', Info 'Info'.
- Row 3: A new row with Rule number '...', Type 'All traffic', Protocol 'All', Port range 'All', Source '0.0.0.0/0', Allow/Deny 'Deny', Info 'Info'.

At the bottom of the table are buttons for 'Add new rule' and 'Sort by rule number'. To the right of the table are buttons for 'Cancel', 'Preview changes', and a prominent orange 'Save changes' button. The footer of the page includes links for CloudShell, Feedback, and various AWS terms like © 2023, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

How to encrypt the root volume of an existing EC2 instance

10. Navigate to the EC2 section within the AWS Management Console and access the instances.
11. Choose the desired instance, such as "LabInstance," and verify whether the storage tab indicates encryption status.
12. Halt the instance to enable snapshot creation by accessing the instance details and changing its status to "stopped."
13. Take note of the volume ID and availability zone of the unencrypted EBS volume by inspecting the storage tab and capturing a screenshot.
14. Click on the action button and select "Create Snapshot." Provide the necessary key and value pairs, such as a name for the snapshot and indication of it being from an unencrypted volume, then proceed with the snapshot creation.
15. Within the Elastic Block Store, navigate to the snapshot section. Locate the previously generated unencrypted volume snapshot, choose to create a volume from it, and configure the settings as follows:

- Set the Availability Zone to us-east-1.
 - Select the MyKMSKey key for encryption.
 - Click on "Create Volume."
16. Establish a customer-managed key by selecting "Customer Managed Key," initiate the creation of a new key, choose "Standard" for key usage, enable encryption and decryption capabilities, and assign a name to the key (e.g., MyKMSKey). Proceed through the default settings, review all configurations, and finalize the key creation process by clicking "Create."

Review

Key configuration

Key type Symmetric	Key spec SYMMETRIC_DEFAULT	Key usage Encrypt and decrypt
Origin AWS KMS	Regionality Single-Region key	

You cannot change the key configuration after the key is created.

Alias and description

Alias	Description
-------	-------------

Key Management Service (KMS)

AWS managed keys
Customer managed keys

Custom key stores
AWS CloudHSM key stores
External key stores

ee63e42e-1816-48e3-88fa-adf22784ee55

General configuration

Alias MyKMSKey	Status Enabled	Creation date Apr 24, 2024 22:35 MST
ARN arn:aws:kms:us-east-1:975050305321:key/ee63e42e-1816-48e3-88fa-adf22784ee55	Description -	Regionality Single Region

Key policy

Key administrators (1)

AWS Services Search [Alt+S] N. Virginia vocabs/user2718047=Chalasani_Hima_Sree @ 9750-5030

Key Management Service (KMS)

AWS managed keys Customer managed keys

Custom key stores AWS CloudHSM key stores External key stores

Key policy

Key administrators (1)

Choose the IAM users and roles who can administer this key through the KMS API. You might need to add additional permissions for the users or roles to administer this key from this console.

Learn more

Search Key administrators Add Remove

Name	Path	Type
voclabs	/	Role

Key deletion

Allow key administrators to delete this key

Key users (1)

Add Remove

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

How to create a SNS topic

- Utilize the AWS services search bar to locate and select "Simple Notification Service" (SNS).
- Navigate to the SNS dashboard and access the "Topics" section from the left sidebar. Proceed to click on the "Create topic" button.
- Input a name for the topic and optionally include a display name if desired.
- Complete the topic creation process by clicking on "Create topic."
- Upon successful creation, the newly formed topic will be displayed in the list. Click on the recently generated topic.
- On the Topic Details page, the Amazon Resource Name (ARN) can be copied for future reference.

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

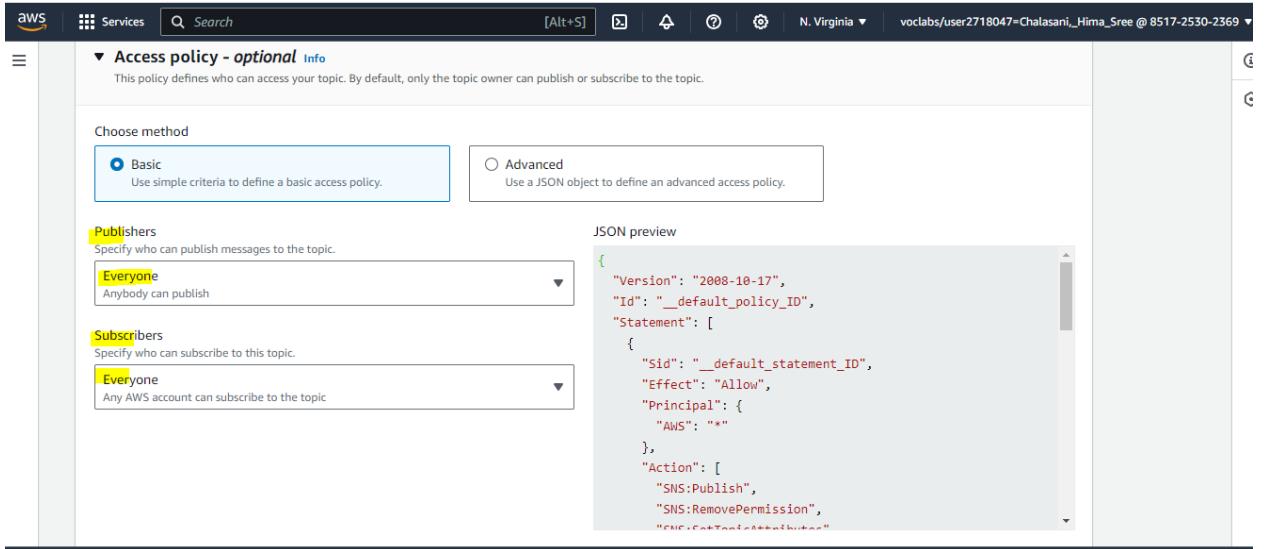
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

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

Display name - optional

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



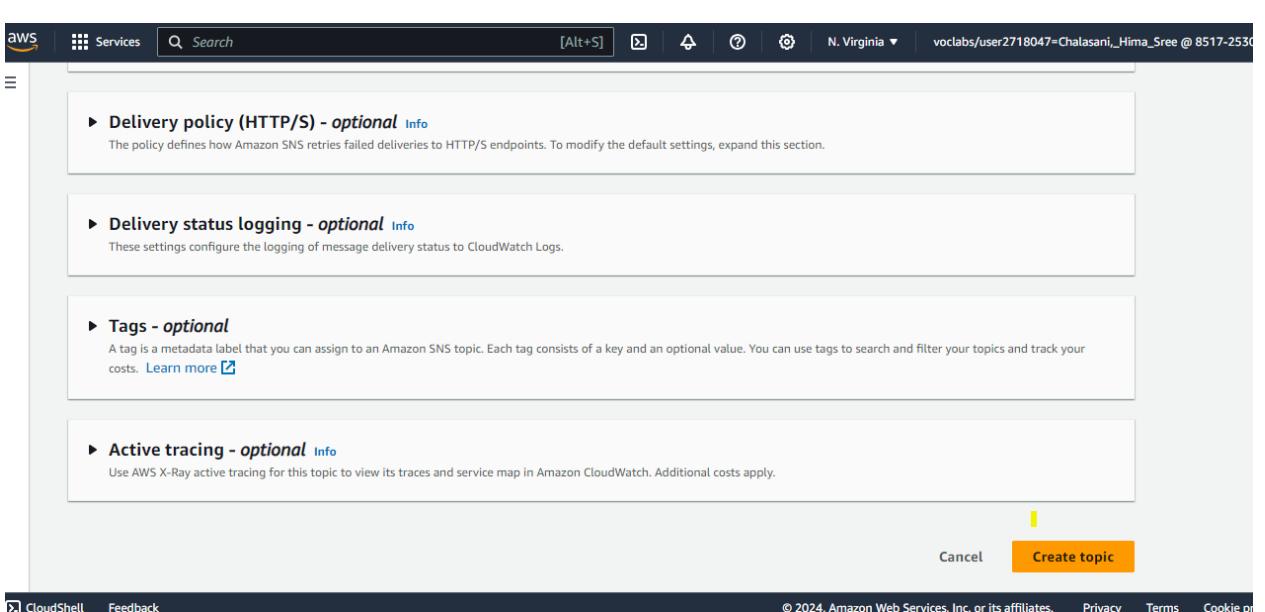
The screenshot shows the 'Access policy - optional' section of the AWS SNS Topic creation page. It includes a 'Choose method' section with 'Basic' selected (using simple criteria) and 'Advanced' (using a JSON object). Below are 'Publishers' and 'Subscribers' dropdowns, both set to 'Everyone'. To the right is a 'JSON preview' pane displaying the generated policy:

```

{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "SNS:Publish",
        "SNS:RemovePermission",
        "SNS:CreateTopic"
      ]
    }
  ]
}

```

At the bottom are 'CloudShell', 'Feedback', and copyright information.

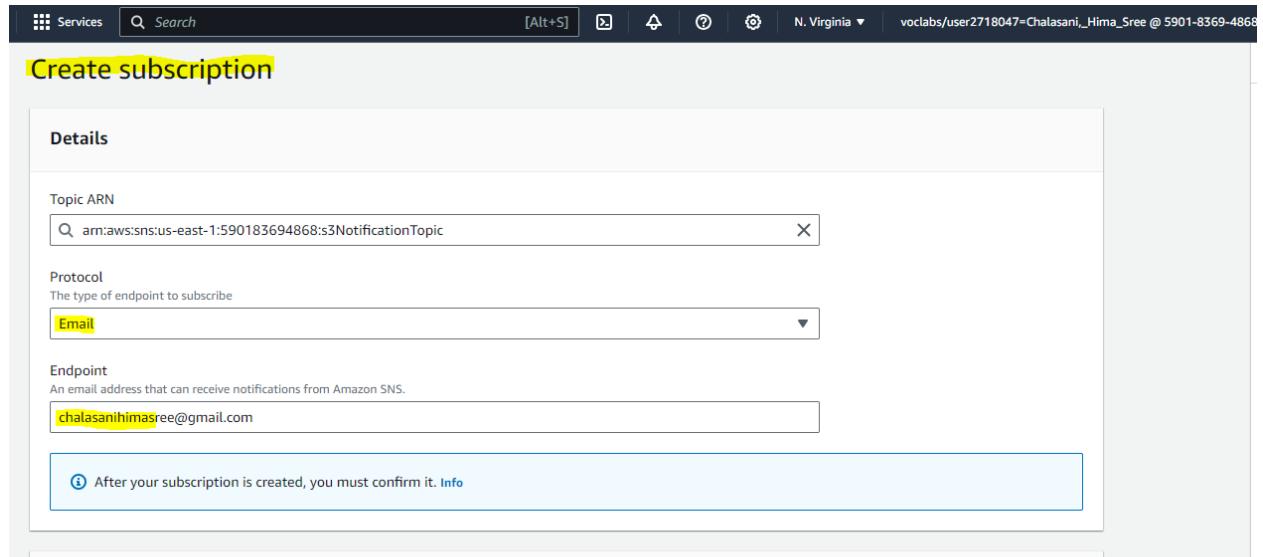


The screenshot shows the 'Delivery policy (HTTP/S) - optional' and 'Delivery status logging - optional' sections. It also includes 'Tags - optional' (describing metadata labels) and 'Active tracing - optional' (describing AWS X-Ray integration). At the bottom are 'Cancel' and 'Create topic' buttons.

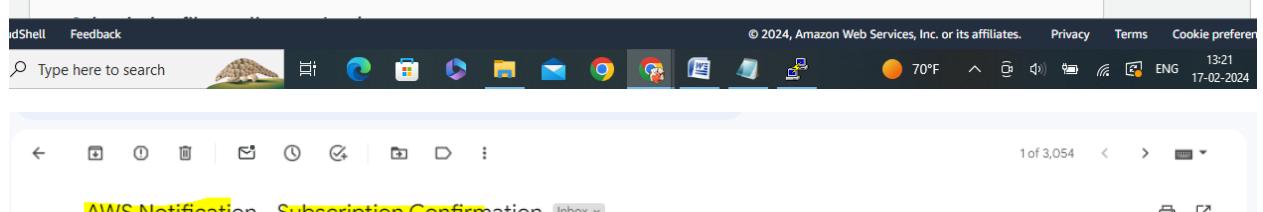
How to subscribe to a SNS topic

17. Navigate to the AWS services search bar and input "SNS" to locate and select "Simple Notification Service."
18. Access the SNS dashboard and proceed to click on "Topics" situated in the left sidebar.
19. Select the specific topic to which subscription is desired.
20. Within the Topic Details page, initiate the subscription process by clicking on the "Create subscription" button.
21. Choose the preferred protocol for message delivery, such as email, HTTP, or HTTPS.
22. Provide the requisite information corresponding to the selected protocol, such as an email address or endpoint URL.
23. Finalize the subscription creation by clicking on "Create subscription."

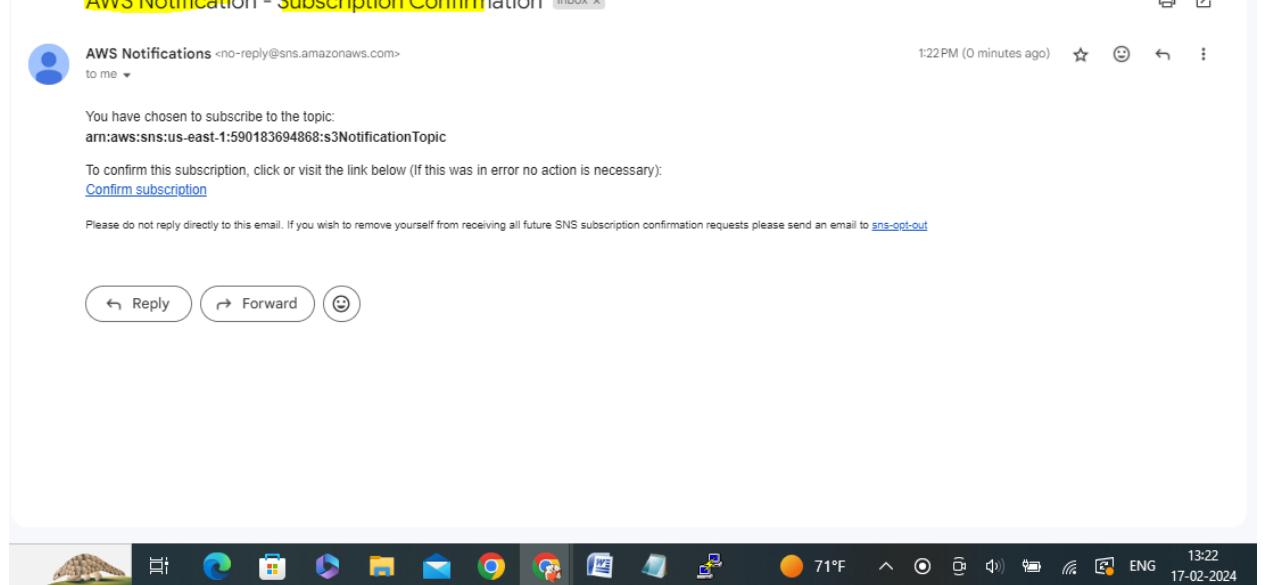
24. Depending on the chosen protocol, it might be necessary to confirm the subscription, which may involve actions like clicking a confirmation link sent via email.



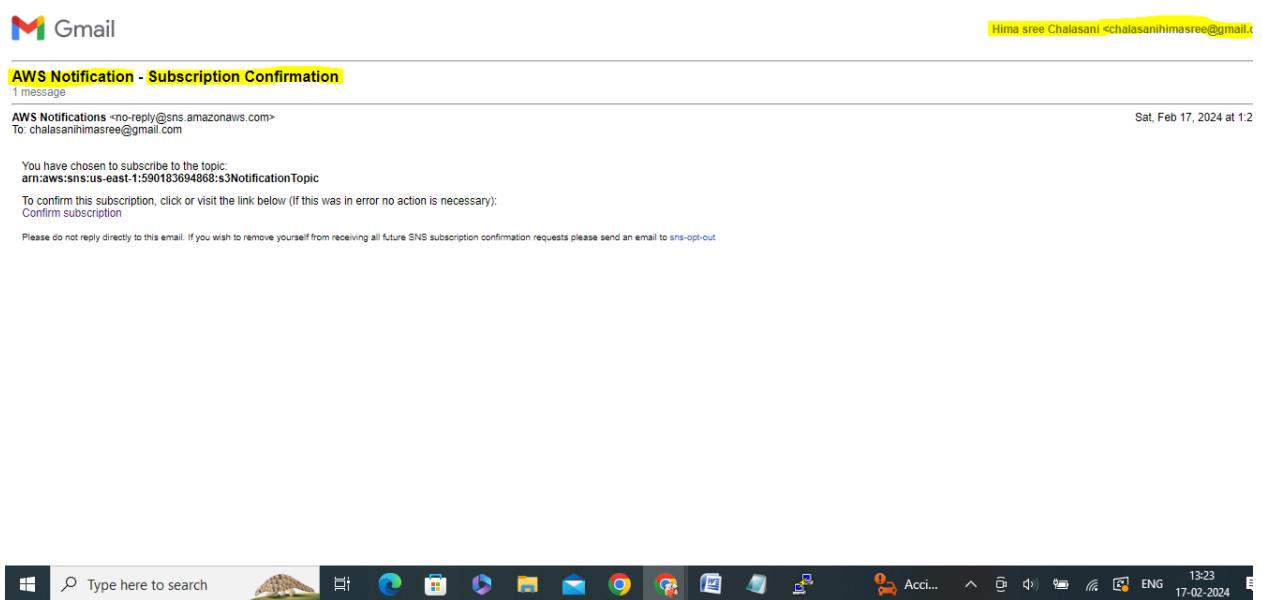
The screenshot shows the 'Create subscription' page in the AWS Management Console. The 'Topic ARN' field contains 'arn:aws:sns:us-east-1:590183694868:s3NotificationTopic'. The 'Protocol' dropdown is set to 'Email'. The 'Endpoint' field contains 'chalasanihima_sree@gmail.com'. A blue info box at the bottom left says 'After your subscription is created, you must confirm it.' Below the form is a success message: 'Subscription created successfully. Subscriptions: 1'. The status bar at the bottom shows the date as 17-02-2024 and the time as 13:21.



The screenshot shows the 'AWS Lambda - Function Overview' page. It displays the function name 'S3toSNS', the runtime as 'Python 3.9', and the last deployment timestamp as '2024-02-17T13:21:00Z'. The status is 'Active'. The status bar at the bottom shows the date as 17-02-2024 and the time as 13:21.



The screenshot shows the 'AWS Lambda - Function Overview' page for the 'S3toSNS' function. It shows the function name, runtime, last deployment timestamp, and active status. The status bar at the bottom shows the date as 17-02-2024 and the time as 13:22.



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

1. Utilize the AWS services search bar to input "CloudWatch" and then proceed to select "CloudWatch" from the results.
2. Navigate to the CloudWatch dashboard and access the "Logs" section located in the left sidebar.
3. Choose the specific log group for which a metric filter needs to be established.
4. Initiate the creation of a Metric Filter by clicking on the corresponding button.
5. Craft a filter pattern tailored to match the desired log events.
6. Name the filter and finalize its creation by selecting "Create Filter."
7. Return to the CloudWatch dashboard and access the "Alarms" section from the left sidebar.
8. Begin the alarm creation process by clicking on "Create Alarm."
9. Select the metric linked with the previously established filter.
10. Define the conditions for triggering the alarm, such as setting threshold values.
11. Configure the actions to be executed upon the alarm state activation.
12. Provide a distinctive name and description for the alarm.
13. Complete the alarm creation process by clicking on "Create Alarm."

AWS Services Search [Alt+S] N. Virginia v vclabs/user2718047=Chalasani_Hima_Sree @ 8517-2530-2

CloudWatch

- Favorites and recents
- Dashboards
- ▶ Alarms 0 0 0
- ▼ Logs
 - Log groups**
 - Log Anomalies
 - Live Tail
 - Logs Insights
- ▼ Metrics
 - All metrics
 - Explorer
 - Streams
- ▶ X-Ray traces

Metric details

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

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](#)

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.

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
This default value is only applied to the metric when the pattern does not match. If you leave this blank, no value is published.

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

AWS Services Search [Alt+S] N. Virginia v vclabs/user2718047=Chalasani_Hima_Sree @ 8517-2530-2369

CloudWatch

- Favorites and recents
- Dashboards
- ▶ Alarms 0 0 0
- ▼ Logs
 - Log groups**
 - Log Anomalies
 - Live Tail
 - Logs Insights
- ▼ Metrics
 - All metrics
 - Explorer
 - Streams
- ▶ X-Ray traces

Step 2 [Assign metric](#)

Step 3 [Review and create](#)

Step 1: Pattern [Edit](#)

Create filter pattern

Filter pattern

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

Step 2: Metric [Edit](#)

Assign metric

Filter name	Metric name
ConsoleLoginErrors	ConsoleLoginFailureCount
Metric namespace	Metric value
CloudTrailMetric	1
Default value	Unit
-	-

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

AWS CloudWatch Metrics Filter Configuration

Metric filters (1/1)

ConsoleLoginErrors

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

Metric: `CloudTrailMetric / ConsoleLoginFailureCount`

Metric value: 1

Default value: -

Unit: -

Create alarm

AWS CloudWatch Alarms - Create alarm

Step 1: Specify metric and conditions

Metric

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

No unit: 4

Namespace: CloudTrailMetric

Metric name: ConsoleLoginFailureCount

Statistic: Sum

Period: 5 minutes

Graph Data:

Time	Value
00:00	2
01:00	3
02:00	4

aws Services Search [Alt+S] N. Virginia v vclabs/user2718047=Chalasani_Hima_Sree @ 8517-2530

Conditions

Threshold type

Static
Use a value as a threshold

Anomaly detection
Use a band as a threshold

Whenever ConsoleLoginFailureCount is...

Define the alarm condition.

Greater
> threshold

Greater/Equal
>= threshold

Lower/Equal
<= threshold

Lower
< threshold

than...

Define the threshold value.

3

Must be a number

► Additional configuration