

C.ABDUL HAKEEM COLLEGE OF ENGINEERING & TECHNOLOGY

Hakeem Nagar, Melvisharam -632509, Vellore District, TamilNadu, India.

(Approved by AICTE, New Delhi and Affiliated to Anna University,
Chennai) (Regd. Under Sec 2(F) & 12(B) of the UGC Act 1956)

Name of the Candidate:

Year: III

Semester: VI

Degree/Branch: B.Tech./IT

Subject Name: Cloud Service Management

Sub.Code: CCS336

University Register Number:

CERTIFICATE

Certified that this is the bonafide record of work done by the above student in **CCS336-
CLOUD SERVICE MANAGEMENT** during the year 2023 - 2024.

Signature of Head of the Department

Signature of Lab In-charge

Submitted for the University Practical Examination held on _____

EXAMINERS

Date: _____

Centre code: 5106

Internal: _____

External: _____

INDEX

S No	Date	Name of the Experiment	Page No	Mark out Of 10	Signature
1		Create a cloud organization in AWS/google cloud or any equivalent open source software like open stack, open nebula with role-based action	2		
2		Create a cost model for a web application using various services and make an analysis for cost benefit	10		
3		Create alerts for usage of cloud resources	15		
4		Create billing alerts for your cloud organization	23		
5		Compare cloud cost for a simple web application across AWS, azure and GCP and suggest the best one	30		

EXP NO:1

DATE:

**CREATE A CLOUD ORGANISATION IN
AWS/GOOGLE CLOUD OR ANY
EQUIVALENT OPEN SOURCE SOFTWARE
LIKE OPENSTACK, EUCALYPTUS, OPEN
NEBULA WITH ROLE BASED ACTIONS**

AIM:

To create a cloud organisation in aws/google cloud or any equivalent open source softwares like openstack, ecalyptus, open nebula with role based actions.

PROCEDURE:

1. Create the user account in aws.
2. Create the user and corresponding user group.
3. Add the user to the user group with role based access.

1. CREATING THE USER ACCOUNT IN AWS

1(a). Visit the aws management console.



Sign in

☒ **Root user**

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**

User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

— New to AWS? —

Create a new AWS account

CATEGORY

AI Use Case Explorer


Discover AI use cases, customer success stories, and expert-curated implementation plans.

Explore now >



Click on the create a new AWS account.

1(b). For creating the account enter the E-mail Id and verify your id using verification code sent by AWS to the corresponding E-mail Id.



Sign up for AWS

Confirm you are you

Making sure you are secure -- it's what we do.

We sent an email with a verification code to **pdhanushkumar96@gmail.com**. (not you?)


Enter it below to confirm your email.

Verification code

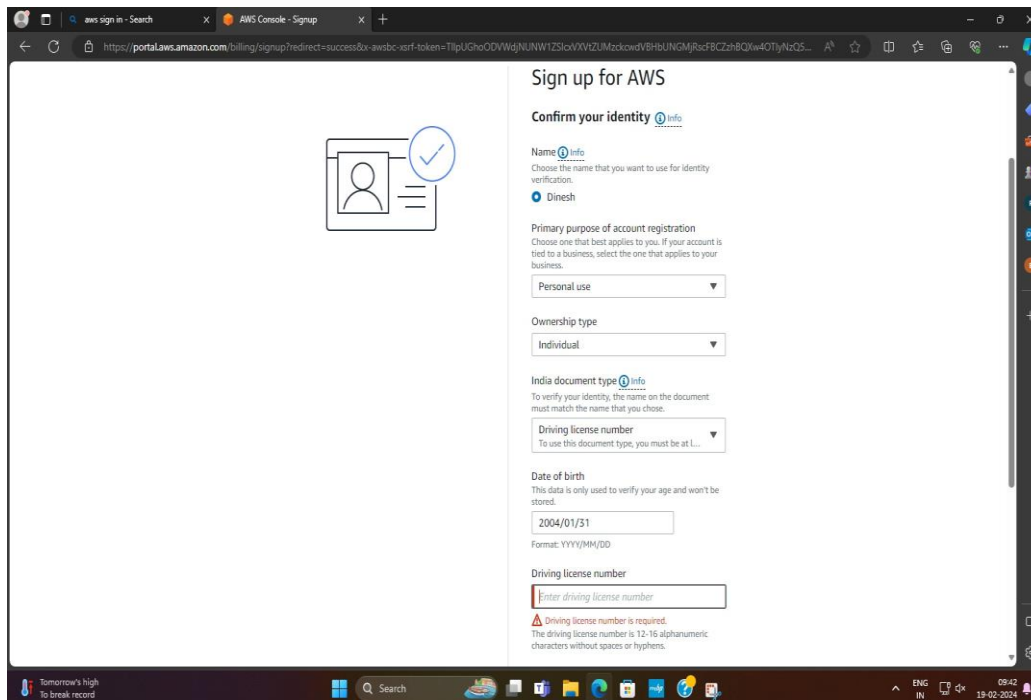
Verify

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.



1(c) . After the verification ,Enter your details and then fill up your account details for the Identity verification. It can charge 2rupee from your account for the card verification.



The screenshot shows the AWS sign-up page for identity verification. The page is titled "Sign up for AWS" and "Confirm your identity". It includes a "Name" field with a dropdown menu showing "Dinesh". Below this is a "Primary purpose of account registration" section with a dropdown menu showing "Personal use". The "Ownership type" section has a dropdown menu showing "Individual". The "India document type" section has a dropdown menu showing "Driving license number". The "Date of birth" section has a text input field showing "2004/01/31". The "Driving license number" section has a text input field with a placeholder "Enter driving license number" and a warning message: "Driving license number is required. The driving license number is 12-16 alphanumeric characters without spaces or hyphens."

Then the aws account is created.

2.CREATE THE USER AND CORRESPONDING USER GROUP.

2(a) .Open the aws management console in any browser and log in as the Root user using your corresponding e-mail id and the password.



Sign in

☒ Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ IAM user

User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

New to AWS?

Create a new AWS account

CATEGORY

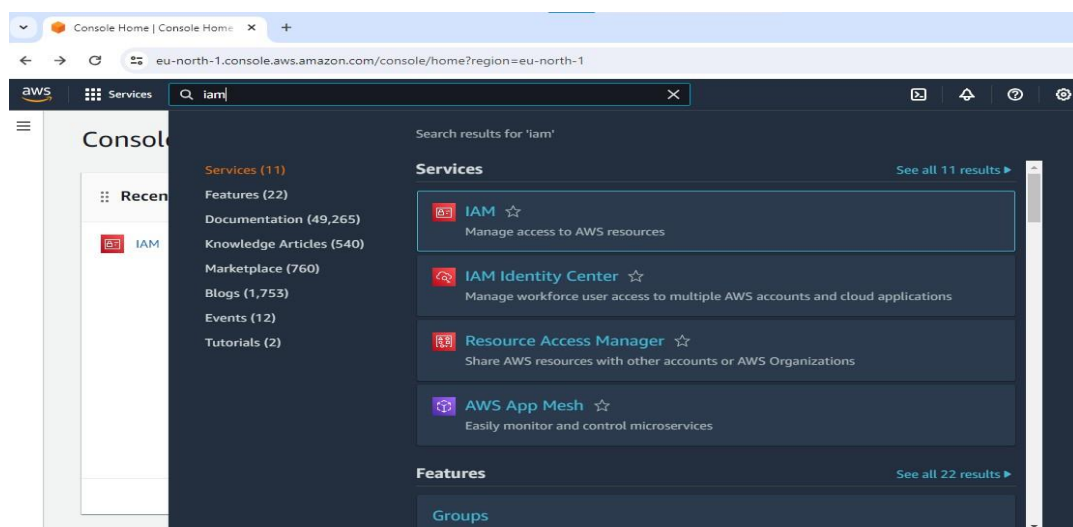
AI Use Case Explorer

Discover AI use cases, customer success stories, and expert-curated implementation plans.

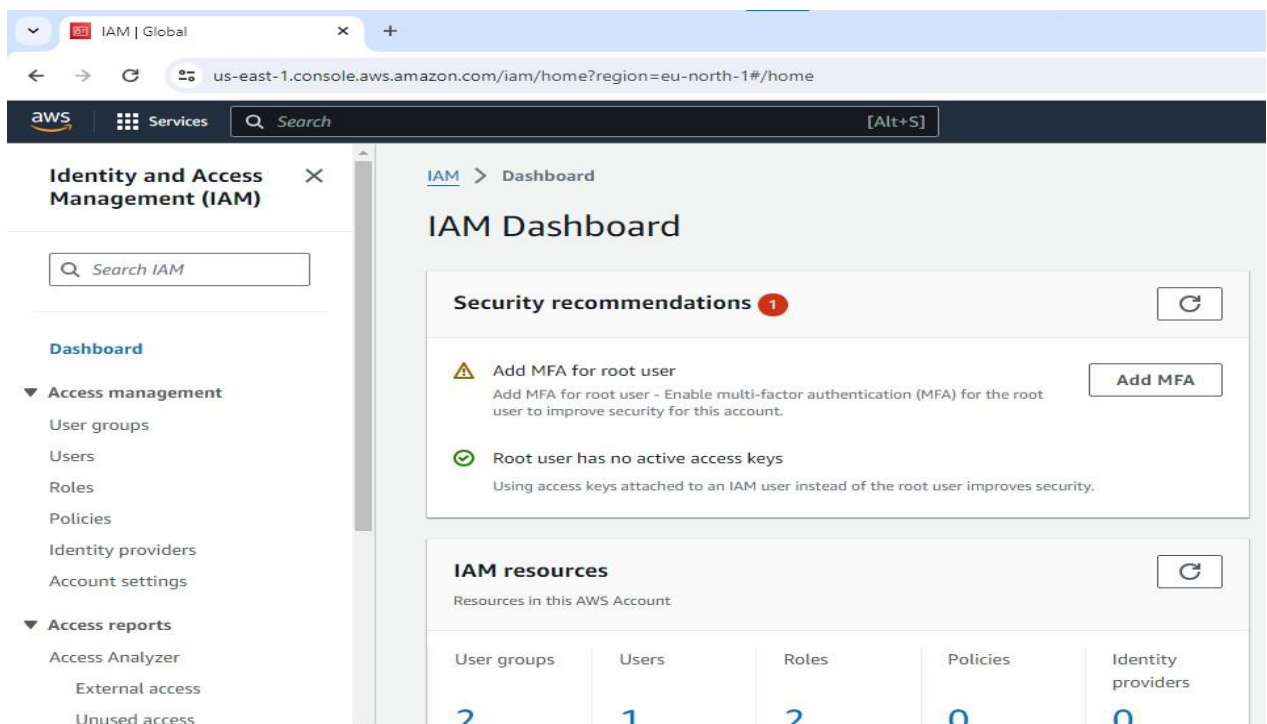
Explore now >



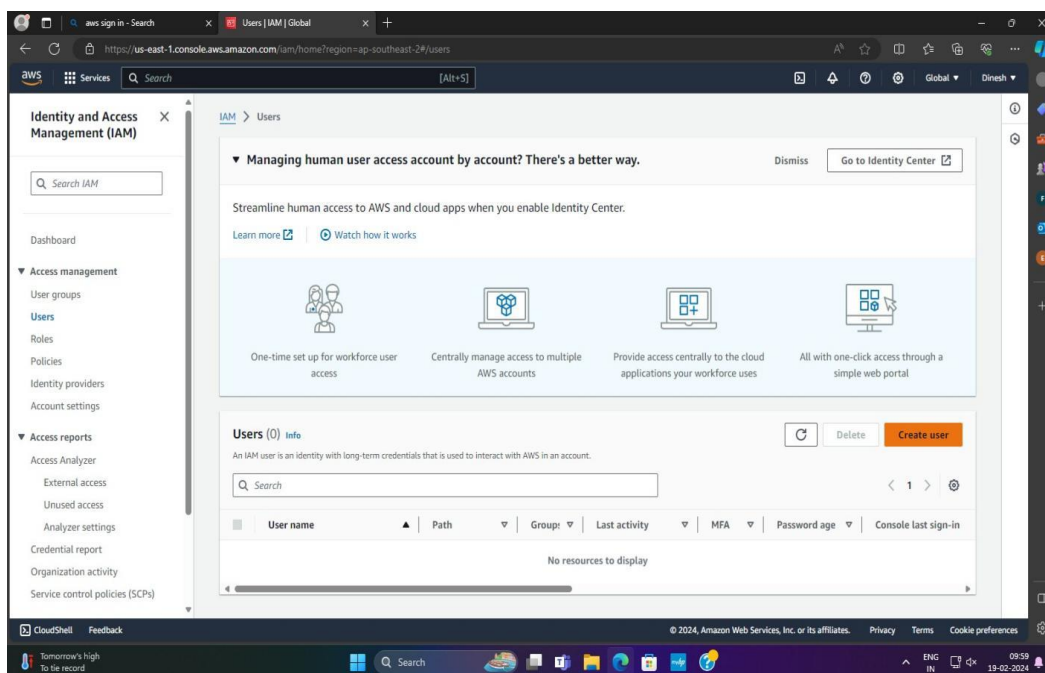
2(b). After entering into the aws management consol then search for IAM in the services.



2(c). The dash board of the IAM is displayed as follows.



2(d). Then click on the user and then under this, click on the create new user and enter the user name.



2(e). Then click on the next button. Click on the add user to the group, After that click on the create group option.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

Create group

► Set permissions boundary - optional

Cancel Previous Next

2(f). Then the create user group with the user group name. After entering the name you need to specify the permission policies as EC2 with read only access.

User group name
Enter a meaningful name to identify this group.
apce-cse
Maximum 128 characters. Use alphanumeric and '*@_@-' characters.

Permissions policies (1/912)

Filter by Type
All types 30 matches

	Policy name	Type	Use...	Description
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Provides administrative access to A
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Provides full access to Amazon EC2
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Provides read-only access to Amazi
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Policy to enable Task Autoscaling f
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Policy to enable CloudWatch Event
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Default policy for the Amazon EC2
<input type="checkbox"/>	AmazonEC2Contai...	AWS managed	None	Default policy for Amazon ECS serv
<input type="checkbox"/>	AmazonEC2FullAcc...	AWS managed	None	Provides full access to Amazon EC2
<input checked="" type="checkbox"/>	AmazonEC2ReadO...	AWS managed	None	Provides read only access to Amazc
<input type="checkbox"/>	AmazonEC2Rolefo...	AWS managed	None	Provides EC2 access to S3 bucket te
<input type="checkbox"/>	AmazonEC2Rolefo...	AWS managed	None	Provides EC2 limited access to S3 b

Cancel Create user group

2(g). Click on the create user group and you will be redirected into the review and create page .Under this click on the create user, then the user will be created.

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name apce	Console password type None	Require password reset No
-------------------	-------------------------------	------------------------------

Permissions summary

< 1 >

Name	Type	Used as
No resources		

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

3.ADD THE USER TO THE USER GROUP WITH ROLE BASED ACCESS.

3(a). To add the users into the group click on the add users

IAM > User groups > apce-cse

apce-cse

Info

Delete

Summary

Edit

User group name apce-cse	Creation time February 26, 2024, 21:19 (UTC+05:30)	ARN am:aws:iam::058264173179:group/apce-cse
-----------------------------	-------------------------------------------------------	------------------------------------------------

Users

Permissions

Access Advisor

Users in this group (0)

Remove Add users

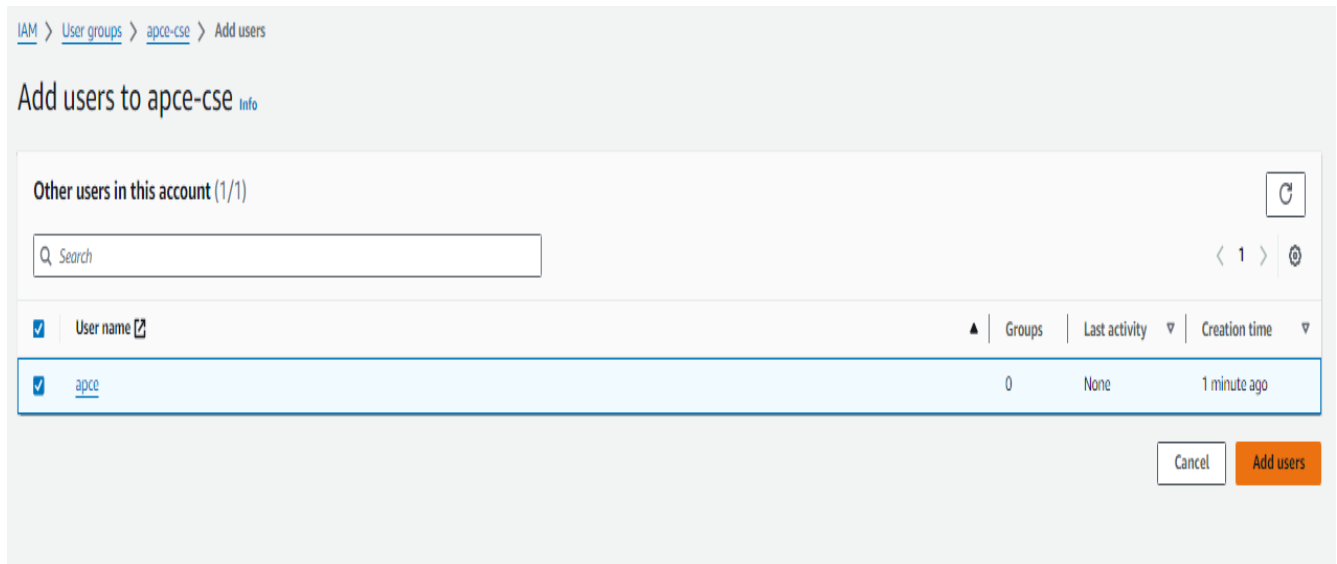
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

Q Search

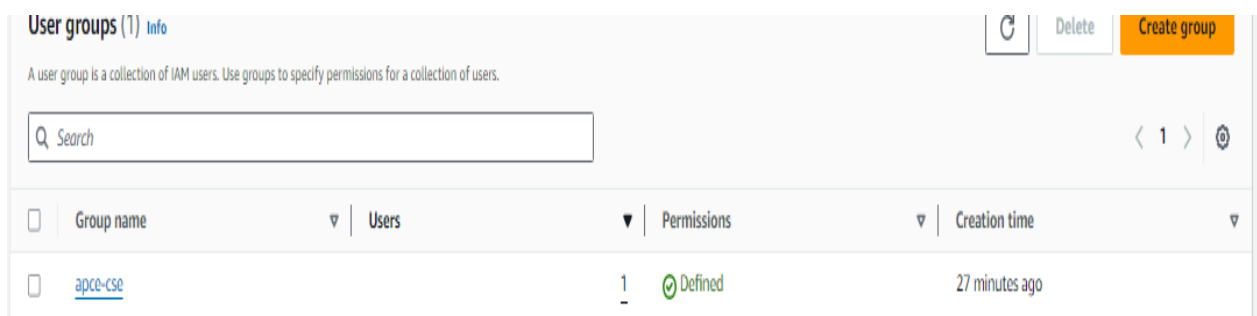
< 1 >

User name	Groups	Last activity	Creation time
No resources to display			

3(b). Then add the existing group to the user group and click on Add users option, now the user is added into the group with the role based action.



3(c) We can view the summary of the user group which is created, by clicking on the user groups on the dashboard.



RESULT:

Thus the organization was created in the amazon aws with the role based actions.

EXP NO:2

DATE:

CREATE A COST-MODEL FOR A WEB APPLICATION USING VARIOUS SERVICES AND MAKE AN ANALYSIS FOR COST-BENEFIT.

AIM:

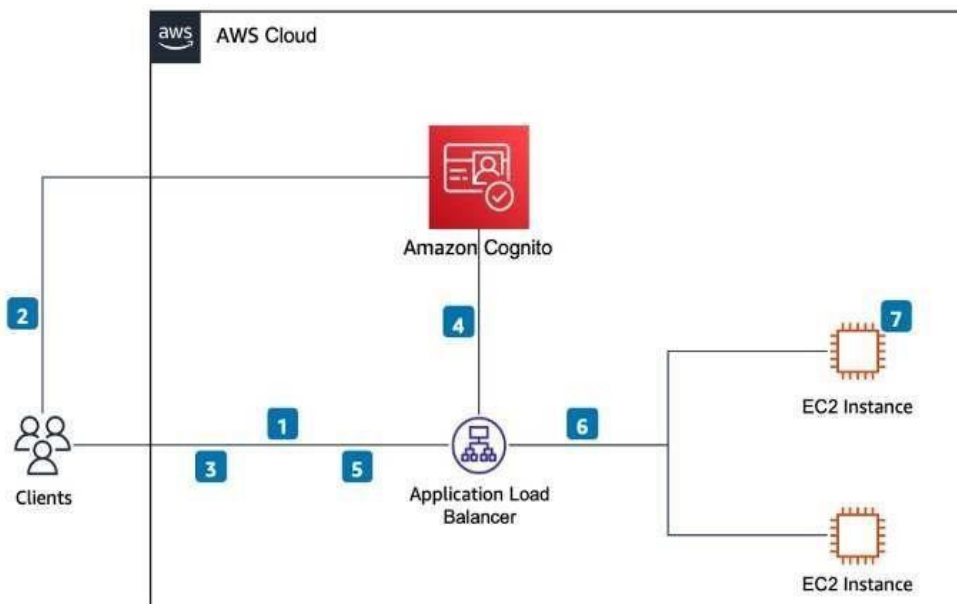
To create a Cost-model for a web application using various services and make a analysis for Cost-benefit.

PROCEDURE:

Creating a cost-model for a web application in AWS involves estimating the costs of using various AWS services for the application. Here's a general process to create a cost-model and do cost-benefit analysis:

1. Identify the AWS services used by the web application:

Some common services used by web applications include Amazon S3, Amazon EC2, Amazon RDS, Amazon API Gateway, AWS Lambda, Amazon DynamoDB, Amazon CloudFront, and Amazon SNS.



2. Estimate the costs of each service:

You can use the AWS Pricing Calculator to estimate the costs of each service. The pricing calculator allows you to enter the specifics of your usage, such as the number of instances, storage size, and data transfer.

3. **Create a cost-model:** Once you have estimated the costs of each service, you can create a costmodel that summarizes the total costs. You can use a spreadsheet or a cloud cost management tool to create the cost-model.
4. **Do cost-benefit analysis:** after creating the cost-model, you can do a cost-benefit analysis to determine if the benefits of using AWS services outweigh the costs. You can compare the costs of using AWS services to the costs of running the application on-premises or using a different cloud provider.

Python code:

```
import boto3

session = boto3.Session(
    aws_access_key_id='YOUR_ACCESS_KEY',
    aws_secret_access_key='YOUR_SECRET_KEY',
    region_name='us-east-1'
)

# Create a Cost Explorer client
cost_explorer = session.client('ce')

Model time_period = { 'TimeUnit':
    'MONTHS', 'Start': '2022-01-01',
    'End': '2022-12-31'
}

model granularity = 'DAILY'

model metrics = ['BlendedCost',
    'UsageQuantity']
```

```

model group_by = [{ 'Type':
'DIMENSION', 'Key':
'SERVICE' }] # Get the cost and usage
data response =
cost_explorer.get_cost_and_usage (
    TimePeriod=time_period,
    Granularity=granularity,
    Metrics=metrics, GroupBy=group_by
)# Print the cost and usage data
print(response) Output:
{

```

```

    'ResultsByTime': [
        {
            'TimePeriod': {
                'Start': '2022-01-01',
                'End': '2022-12-31',
                'TimeUnit': 'MONTHS'
            },
            'Groups': [
                {
                    'Keys': [
                        'AmazonEC2'
                    ],
                    'Metrics': { 'BlendedCost':
                        {
                            'Amount': '1234.56',
                            'Unit': 'USD'
                        },
                    'UsageQuantity': {
                        'Amount': '1000.0',
                        'Unit': 'Hours'
                    }
                }
            ]
        }
    ]
}

```

```

        }
    }
},
{
    'Keys': [
        'AWSLambda'
    ],
    'Metrics': { 'BlendedCost':
        {
            'Amount': '789.0',
            'Unit': 'USD'
        },
        'UsageQuantity': {
            'Amount': '5000000',
            'Unit': 'requests'
        }
    }
}
]
}
],
'ResponseMetadata': {
    'RequestId': 'abcdefg-1234-5678-
90ab-cdefghijkl',
    'HTTPStatusCode': 200,
    'HTTPHeaders': {
        'content-type':
        'text/xml;charset=UTF-8',
        'content-length': '1234',
        'date': 'Tue, 15 Feb 2022
12:34:56 GMT'
    }
}
}

```

```
},  
'RetryAttempts': 0
```

Result

Thus, Cost-model for a web application using various services created and analysis was implemented successfully.

EXP NO:3
DATE:

CREATE ALERTS FOR USAGE OF CLOUD RESOURCES

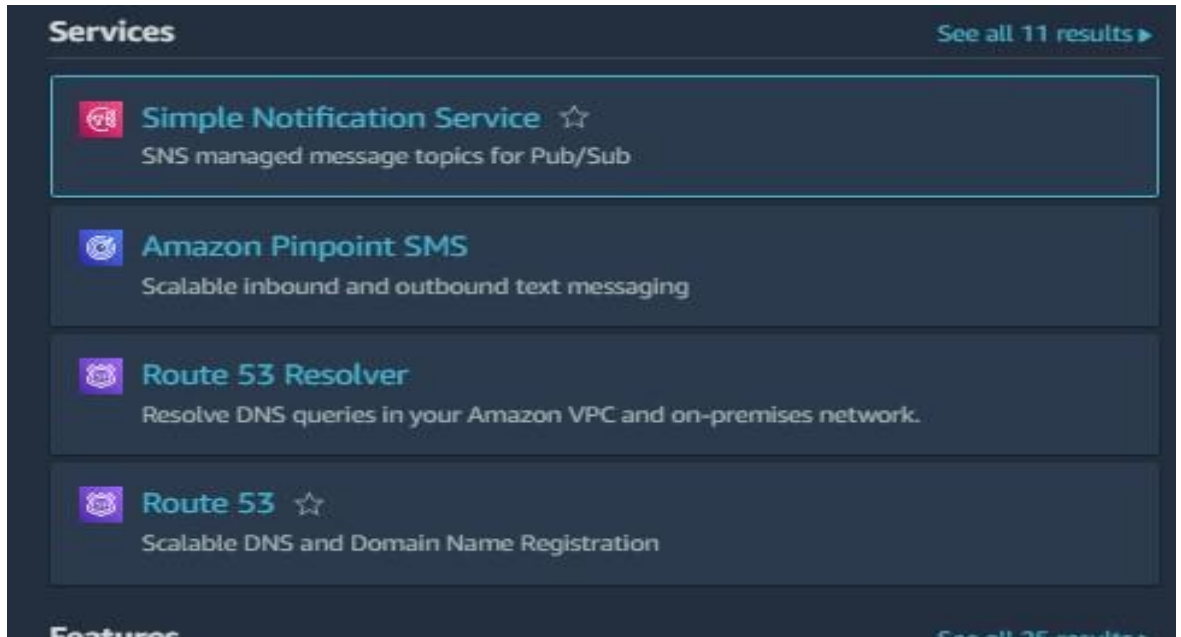
AIM:

To create alerts for usage of cloud resources.

PROCEDURE:

Step-(1): Login into the aws management console.

Step-(2): Search for the sns on the services.



Step-(3): Under the sns click on the create topic with standard type.

Create topic

Details

Type **Info**
Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)

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

☒ Standard

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

Name

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

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

Maximum 100 characters.

Step-(4):Enter the topic name and click create to proceed further,then the topic is created.

The screenshot shows the AWS SNS console for a topic named 'topic1'. At the top, there are buttons for 'Edit', 'Delete', and 'Publish message'. Below this is a 'Details' section with a table containing the following information:

Name	topic1
Display name	-
ARN	arn:aws:sns:us-east-1:058264173179:topic1
Type	Standard
Topic owner	058264173179

Below the details section are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery policy (HTTP/S)', 'Delivery status logging', 'Encryption', 'Tags', and 'Integrations'. The 'Subscriptions' tab is active, showing 'Subscriptions (0)'. There are buttons for 'Edit', 'Delete', 'Request confirmation', 'Confirm subscription', and 'Create subscription'. A search bar is present with the text 'Search'. Below the search bar is a table with columns: ID, Endpoint, Status, and Protocol. The table is empty, and a message states: 'No subscriptions found. You don't have any subscriptions to this topic.' There is a 'Create subscription' button at the bottom of the table.

Step-(5):Then the next step is to create the subscription.To do this click on the create subscription.

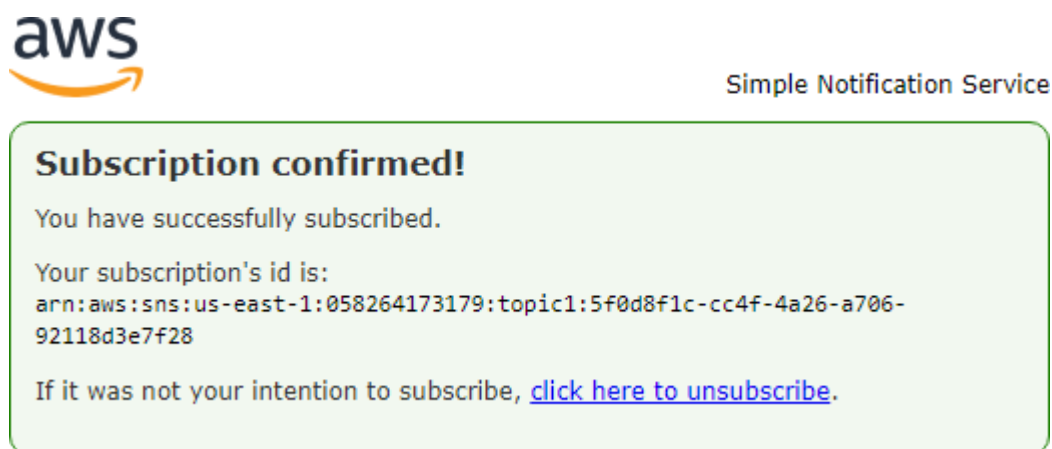
The screenshot shows the AWS SNS console 'Create subscription' form. The 'Topic ARN' field is populated with 'arn:aws:sns:ap-southeast-2:767398037480:ex_2'. The 'Protocol' dropdown is set to 'Email'. The 'Endpoint' field is populated with 'dineshjagan06@gmail.com'. Below the endpoint field is a blue box with an information icon and the text: 'After your subscription is created, you must confirm it. Info'. There are two optional sections: 'Subscription filter policy - optional Info' and 'Redrive policy (dead-letter queue) - optional Info'. At the bottom right, there are 'Cancel' and 'Create subscription' buttons. The browser address bar shows the URL: 'https://ap-southeast-2.console.aws.amazon.com/sns/v3/home?region=ap-southeast-2#/create-subscription'. The Windows taskbar at the bottom shows the date and time as 19-02-2024, 09:52.

Select the protocol as E-mail and then enter the root e-mail id, then click on the create subscription option.

The conformation mail will be sent after clicking on the create subscription button. Now go to your mail box and check the mail of aws, also click on the confirm subscription Button.



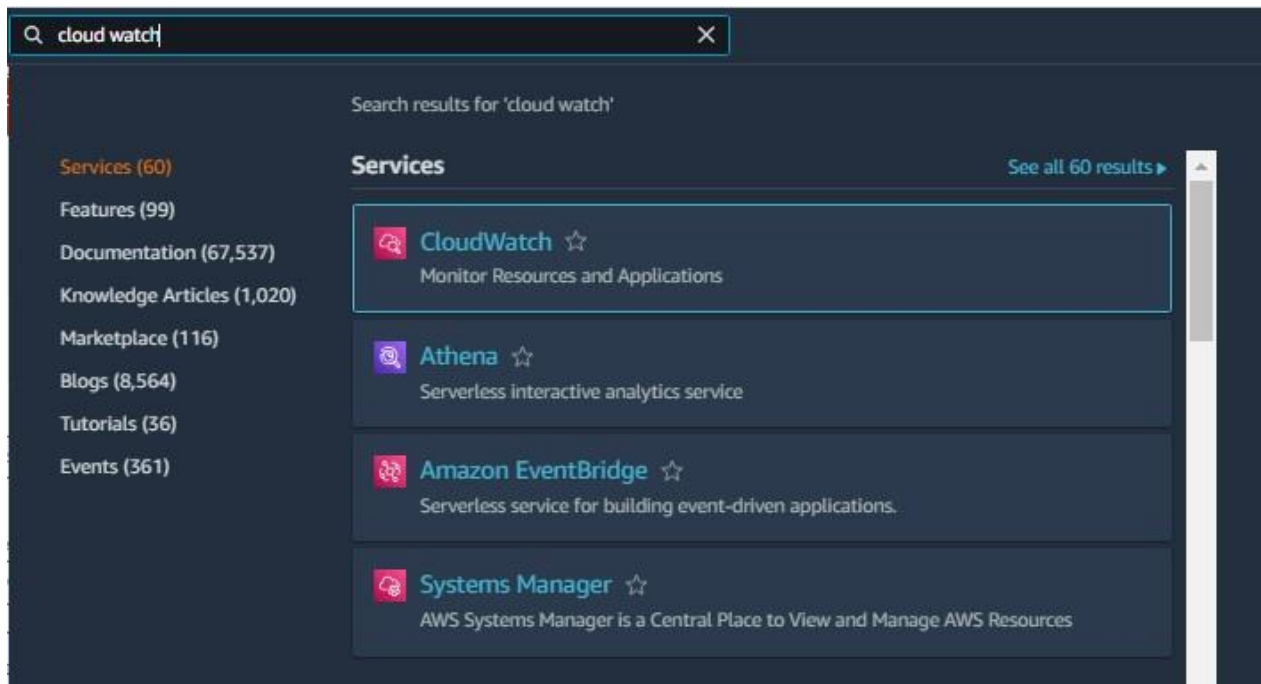
After confirm subscription, We can able to see the following message.



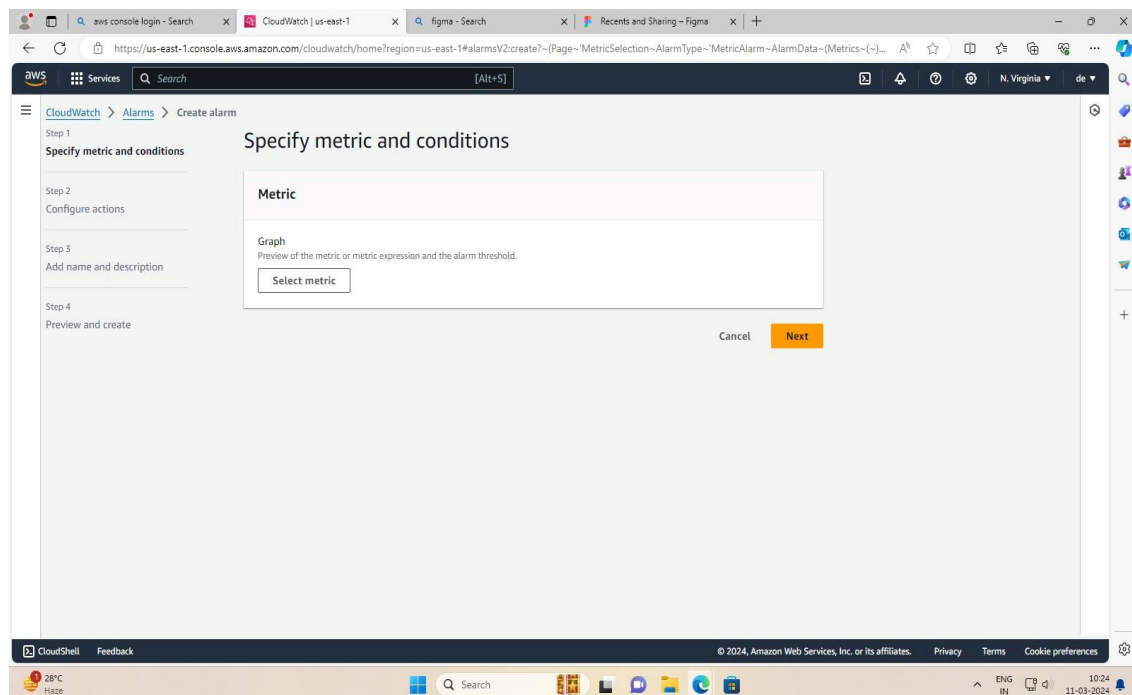
Now the topic has been successfully created with an subscription id as given below

Details	
ARN arn:aws:sns:us-east-1:058264173179:topic1:5f0d8f1c-cc4f-4a26-a706-92118d3e7f28	Status Confirmed
Endpoint pdhanushkumar96@gmail.com	Protocol EMAIL
Topic topic1	
Subscription Principal arn:aws:iam::058264173179:root	

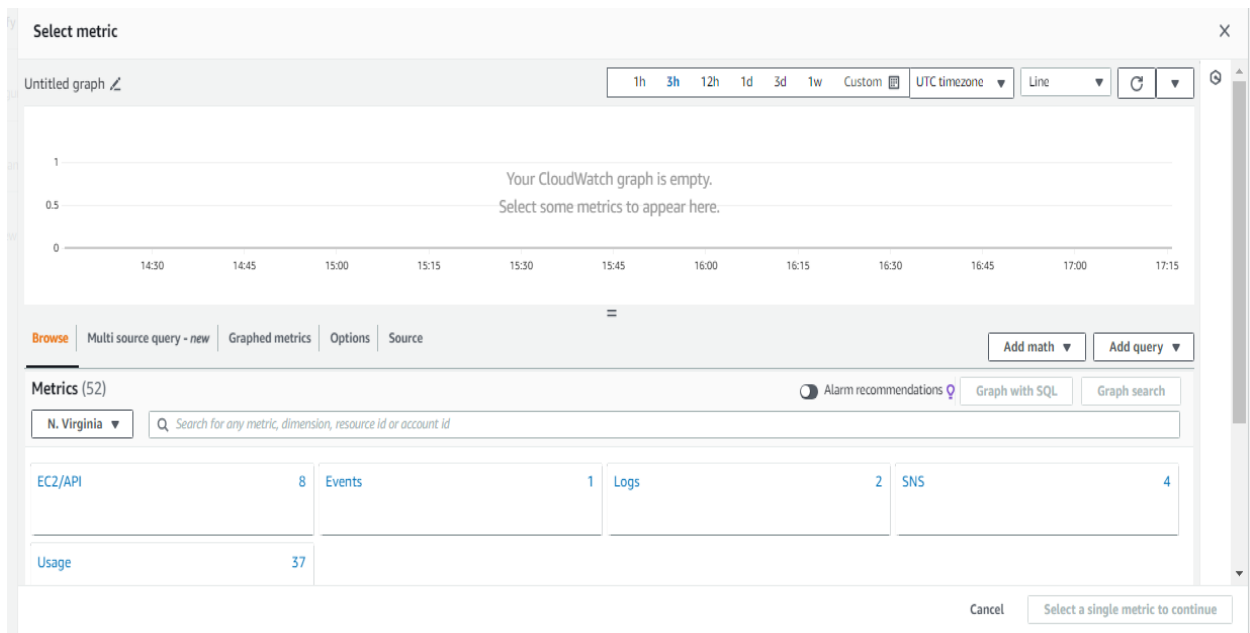
Step-(6): Search for the cloud watch in the services for creating the alarms for alert the billing of the cloud resource usage.



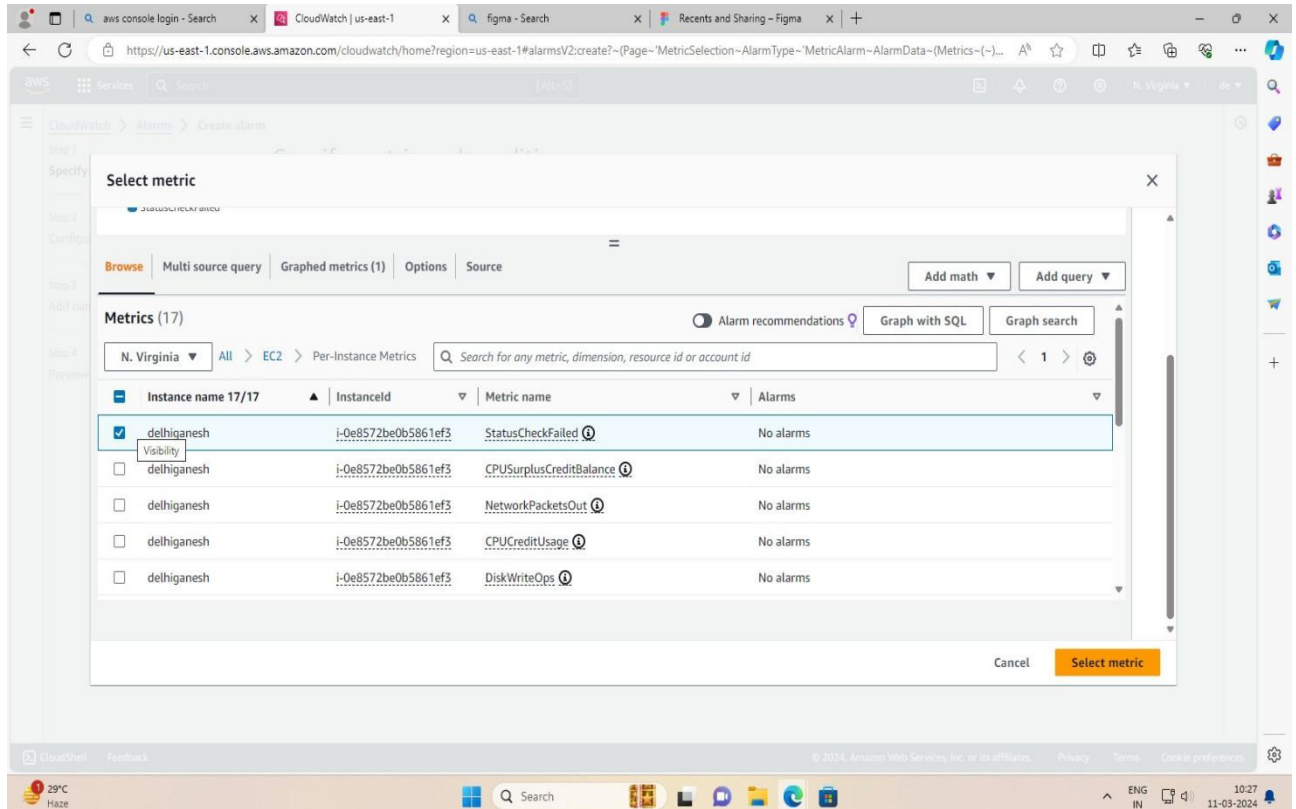
Step-(7): Click on the create alarm option in the cloud watch. Specify the conditions and metric for the alarm.



Step-(8): Click on next and under the select metric click on the sns.



Step-(9): Then the notification option for the alarm.



After this select your metric for the alarm as shown above.

Step-(10): Set the threshold value as 500 or 1 (“as you wish”) and click on the greater than or equal option.

Period: 5 minutes

Conditions

Threshold type

- ☒ Static: Use a value as a threshold
- ☐ Anomaly detection: Use a band as a threshold

Whenever StatusCheckFailed is... Define the alarm condition.

- ☐ Greater: > threshold
- ☒ Greater/Equal: >= threshold
- ☐ Lower/Equal: <= threshold
- ☐ Lower: < threshold

than... Define the threshold value.

1

Must be a number

Additional configuration

Cancel Next

Step-(11): Then click on the next and choose the topic for sending the notification.

Alarm recommendations available

Turn on Recommendations to pre-populate the wizard with the recommended alarms.

CloudWatch > Alarms > Create alarm

Step 1: Specify metric and conditions

Step 2: **Configure actions**

Step 3: Add name and description

Step 4: Preview and create

Configure actions

Notification

Alarm state trigger: Define the alarm state that will trigger this action.

- ☒ In alarm: The metric or expression is outside of the defined threshold.
- ☐ OK: The metric or expression is within the defined threshold.
- ☐ Insufficient data: The alarm has just started or not enough data is available.

Remove

Send a notification to the following SNS topic: Define the SNS (Simple Notification Service) topic that will receive the notification.

- ☒ Select an existing SNS topic
- ☐ Create new topic
- ☐ Use topic ARN to notify other accounts

Send a notification to...

Select an SNS topic

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Add notification

Step-(12): After configuring click on the create alarm.

The screenshot shows the 'Create Alarm' wizard in the AWS console. It is at Step 2: 'Configure actions'. The 'Actions' section shows a notification action: 'When In alarm, send a notification to "topic1"'. There is an 'Edit' button next to the step title. Below this is Step 3: 'Add name and description', which shows a name 'billing alarm' and a description '-'. At the bottom are 'Cancel', 'Previous', and 'Create alarm' buttons.

Additional configuration

Step 2: Configure actions Edit

Actions

Notification
When In alarm, send a notification to "topic1"

Step 3: Add name and description Edit

Name and description

Name
billing alarm

Description
-

Cancel Previous **Create alarm**

Step-(13): Now the alarm can be created as shown below.

This screenshot shows the same 'Create Alarm' wizard as in Step 12, but within a browser window. The browser tabs include 'aws console login - Search', 'CloudWatch | us-east-1', 'Figma - Search', and 'Recents and Sharing - Figma'. The URL is 'https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?-(Page=Preview~AlarmType=MetricAlarm~AlarmData=(Namespace=AWS2...'. The wizard content is identical to the previous screenshot, showing the 'Configure actions' step with a notification to 'delhi' and the 'Add name and description' step with the name 'example'. The bottom of the browser shows the Windows taskbar with the date '11-03-2024' and time '10:29'.

aws console login - Search CloudWatch | us-east-1 Figma - Search Recents and Sharing - Figma

https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?-(Page=Preview~AlarmType=MetricAlarm~AlarmData=(Namespace=AWS2...

Additional configuration

Step 2: Configure actions Edit

Actions

Notification
When In alarm, send a notification to "delhi"

Notification
When In alarm, send a notification to "delhi"

Step 3: Add name and description Edit

Name and description

Name
example

Description
-

Cancel Previous **Create alarm**

CloudShell Feedback

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

Breaking news
Unfolding now

Search

ENG IN 10:29 11-03-2024

RESULT:

Thus the alerts for usage of cloud resources was created successfully.

EXP NO:4
DATE:

CREATE BILLING ALERTS FOR YOUR CLOUD ORGANIZATION

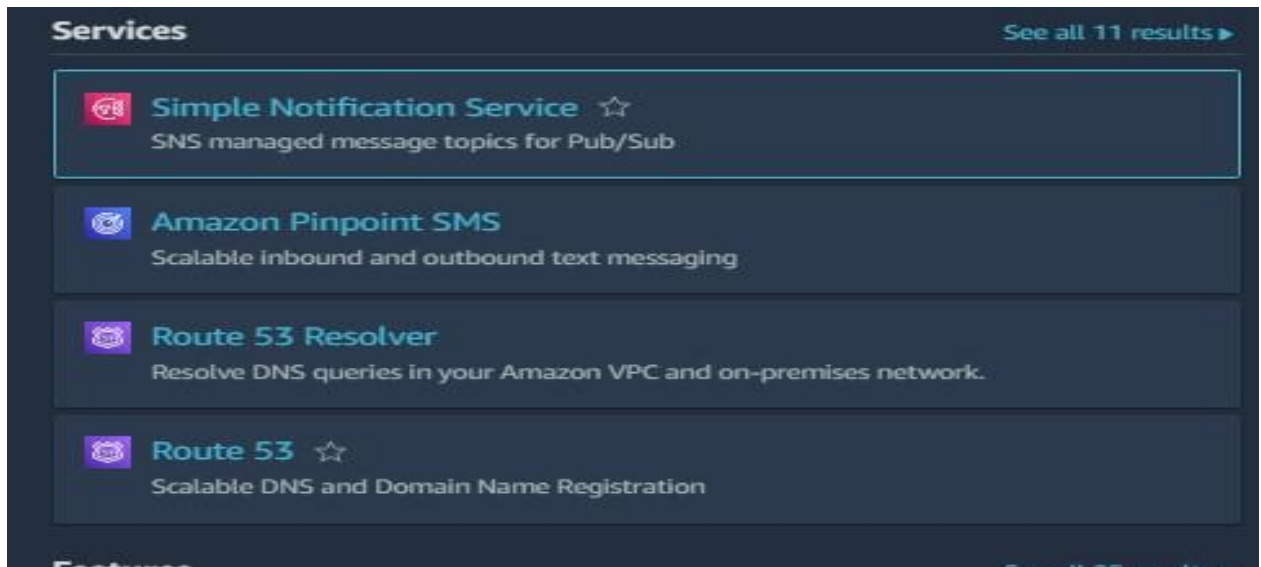
AIM:

To create billing alerts for the cloud organization.

PROCEDURE:

Step-(1): Login into the aws management console.

Step-(2): Search for the sns on the services.



Step-(3): Under the sns click on the create topic with standard type.

A screenshot of the 'Create topic' form in the AWS Management Console. The form has a 'Details' section with two tabs: 'Type' and 'Info'. Under the 'Type' tab, there are two radio button options: 'FIFO (first-in, first-out)' and 'Standard'. The 'Standard' option is selected. Below the type selection, there is a 'Name' field with the text 'topic1' entered. A note below the name field states: 'Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).'. There is also a 'Display name - optional' field with the text 'My Topic' entered. A note below the display name field states: 'To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. Maximum 100 characters.'

Step-(4): Enter the topic and click create to proceed further, then the topic is created.

The screenshot shows the AWS SNS console for a topic named 'topic1'. At the top, there are buttons for 'Edit', 'Delete', and 'Publish message'. Below this is a 'Details' section with a table containing the following information:

Name	topic1	Display name	-
ARN	arn:aws:sns:us-east-1:058264173179:topic1	Topic owner	058264173179
Type	Standard		

Below the details is a horizontal menu with tabs: 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery policy (HTTP/S)', 'Delivery status logging', 'Encryption', 'Tags', and 'Integrations'. The 'Subscriptions' tab is active, showing 'Subscriptions (0)'. There are buttons for 'Edit', 'Delete', 'Request confirmation', 'Confirm subscription', and a prominent orange 'Create subscription' button. A search bar with the placeholder 'Search' is present. Below the search bar is a table header with columns: 'ID', 'Endpoint', 'Status', and 'Protocol'. The table body is empty, with a message stating 'No subscriptions found. You don't have any subscriptions to this topic.' and a 'Create subscription' button at the bottom.

Step-(5): Then the next step is to create the subscription. To do this click on the create subscription.

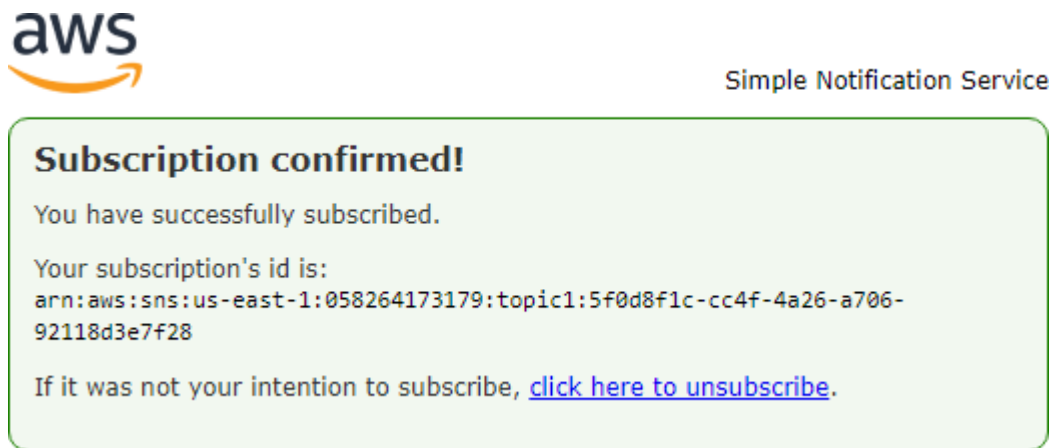
The screenshot shows the 'Create subscription' page in the AWS SNS console. It has a 'Details' section with a 'Topic ARN' field containing 'MyTopic' and a 'Protocol' dropdown menu set to 'Select protocol'. Below this is a blue information box that says: 'After your subscription is created, you must confirm it. Info'. There are two expandable sections: 'Subscription filter policy - optional Info' (with a note 'This policy filters the messages that a subscriber receives.') and 'Redrive policy (dead-letter queue) - optional Info' (with a note 'Send undeliverable messages to a dead-letter queue.'). At the bottom right, there are 'Cancel' and 'Create subscription' buttons.

select the protocol as E-mail and then enter the root e-mail id, then click on the create subscription option.

The conformation mail will be send after clicking on the create subscriptio. Go to your mail box and check the mail of aws, also click on the confirm subscription.



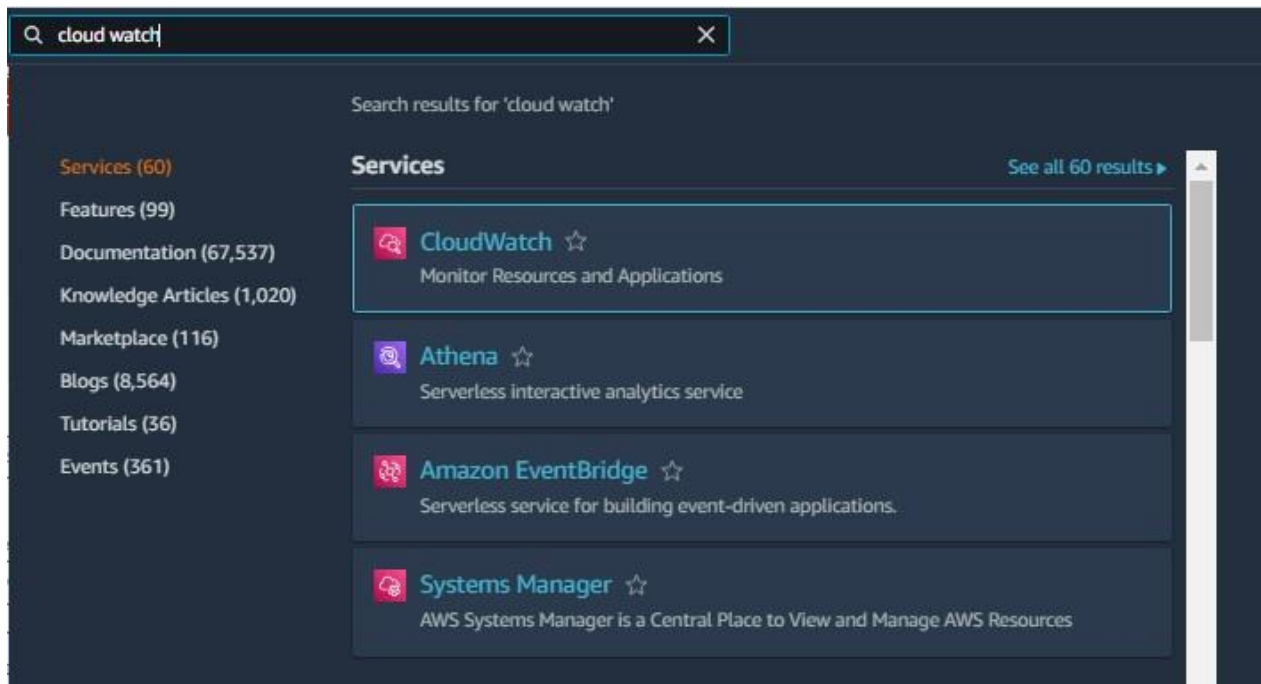
after confirm subscription, we can able to see the following message.



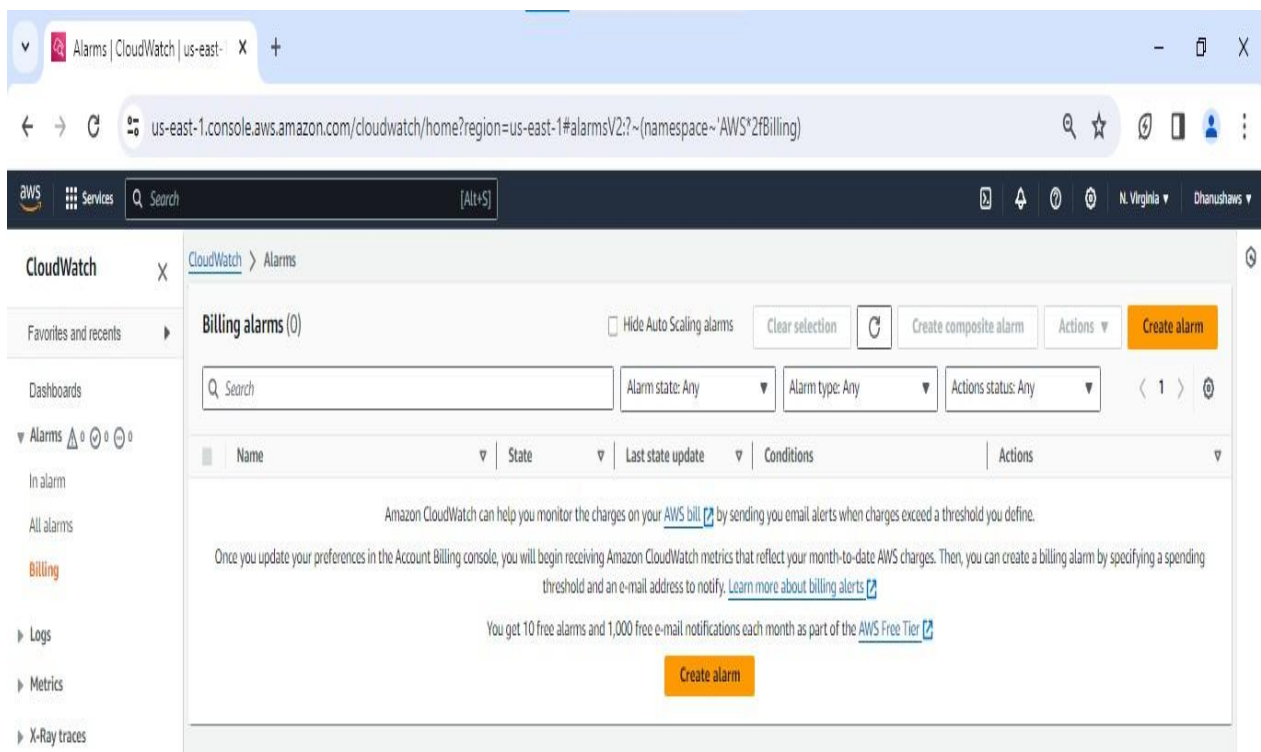
Now the topic can be successfully created as bellow

Details	
ARN arn:aws:sns:us-east-1:058264173179:topic1:5f0d8f1c-cc4f-4a26-a706-92118d3e7f28	Status Confirmed
Endpoint pdhanushkumar96@gmail.com	Protocol EMAIL
Topic topic1	
Subscription Principal arn:aws:iam::058264173179:root	

Step-(6): Search for the cloud in the services for creating the alarms for alert the billing of the cloud resource usage.



Step-(7): In the cloud watch click on the billing alarm and then click on the create new alarm.



Step-(8): Then select the metric for the billing alarm as shown below.

The screenshot shows the AWS CloudWatch console in the 'us-east-1' region. The breadcrumb trail is 'CloudWatch > Alarms > Create alarm'. The left sidebar shows four steps: 'Step 1: Specify metric and conditions' (active), 'Step 2: Configure actions', 'Step 3: Add name and description', and 'Step 4: Preview and create'. The main area is titled 'Specify metric and conditions'. It features a 'Metric' input field, a 'Graph' preview area with the text 'Preview of the metric or metric expression and the alarm threshold.', and a 'Select metric' button. At the bottom right are 'Cancel' and 'Next' buttons.

Step-(9): Set USD as 10 dollars and specify the greater than or equal option.

The screenshot shows the 'Configure actions' page in the AWS CloudWatch console. The section is titled 'Notification'. Under 'Alarm state trigger', there are three radio button options: 'In alarm' (selected), 'OK', and 'Insufficient data'. Each option has a description of the trigger condition. A 'Remove' button is located to the right of these options. Below this, the text says 'Send a notification to the following SNS topic' and 'Define the SNS (Simple Notification Service) topic that will receive the notification.' There are three radio button options: 'Select an existing SNS topic' (selected), 'Create new topic', and 'Use topic ARN to notify other accounts'. Under 'Send a notification to...', there is a search box containing 'topic1' and a close button. Below the search box, it says 'Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.' At the bottom, there is an 'Email (endpoints)' section showing 'pdhanushkumar96@gmail.com' with a link to 'View in SNS Console', and an 'Add notification' button.

Step-(10): Describe the name of the alarm and then click on next.

Add name and description

Name and description

Alarm name

Alarm description - optional [View formatting guidelines](#)

Edit | Preview

if the usd goes > 10 dollars then it sends a mail to my Id

Up to 1024 characters (58/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel Previous Next

Click on next.

Step-(11): Now the billing alarm is created as shown in below.

CloudWatch > Alarms

Alarms (1)

☐ Hide Auto Scaling alarms

Clear selection ↺ Create composite alarm Actions Create alarm

Alarm state: OK Alarm type: Any Actions status: Any < 1 > ⚙

<input type="checkbox"/>	Name	State	Last state update	Conditions	Actions
<input type="checkbox"/>	billing alert	OK	2024-02-27 00:53:57	CallCount > 10 for 1 datapoints within 5 minutes	Actions enabled

RESULT

Thus the billing alert is created for the cloud organization successfully.

EXP NO:5

DATE:

**COMPARE CLOUD COST FOR A SIMPLE
WEB APPLICATION ACROSS AWS,
AZURE AND GCP AND SUGGEST THE
BEST ONE.**

AIM:

To compare Cloud cost for a simple web application across AWS, Azure and GCP and suggest the best one.

OBSERVATION:

1. **AWS:** AWS offers a rich array of tools, including databases, analytics, management, IoT, security, and enterprise applications. AWS introduced per-second billing in 2017 for EC2 Linux-based instances and EBS volumes.
2. **Azure:** Azure has slightly surpassed AWS in the percentage of enterprises using it. Azure also offers various services for enterprises, and Microsoft's longstanding relationship with this segment makes it an easy choice for some customers. While Azure is the most expensive choice for general-purpose instances, it's one of the most cost-effective alternatives to compute-optimized instances.
3. **Google Cloud Platform (GCP):** GCP stands out thanks to its almost limitless internal research and expense. GCP is different due to its role in developing various open-source technologies. Google Cloud is much cheaper than AWS and Azure for computing optimized cloud-based instances.

The best platform depends on your specific needs and requirements. If you need a wide array of tools and services, AWS might be the best choice. If you're looking for enterprise services and have a longstanding relationship with Microsoft, Azure could be your best bet.

CONCLUSION:

If you prioritize innovation and open-source technologies, GCP could be the right choice. For compute optimized instances, GCP seems to be the most cost-effective. However, it's essential to understand your requirements fully before making a decision

RESULT:

Thus, the comparison for Cloud cost for a simple web application across AWS, Azure and GCP were implemented successfully.