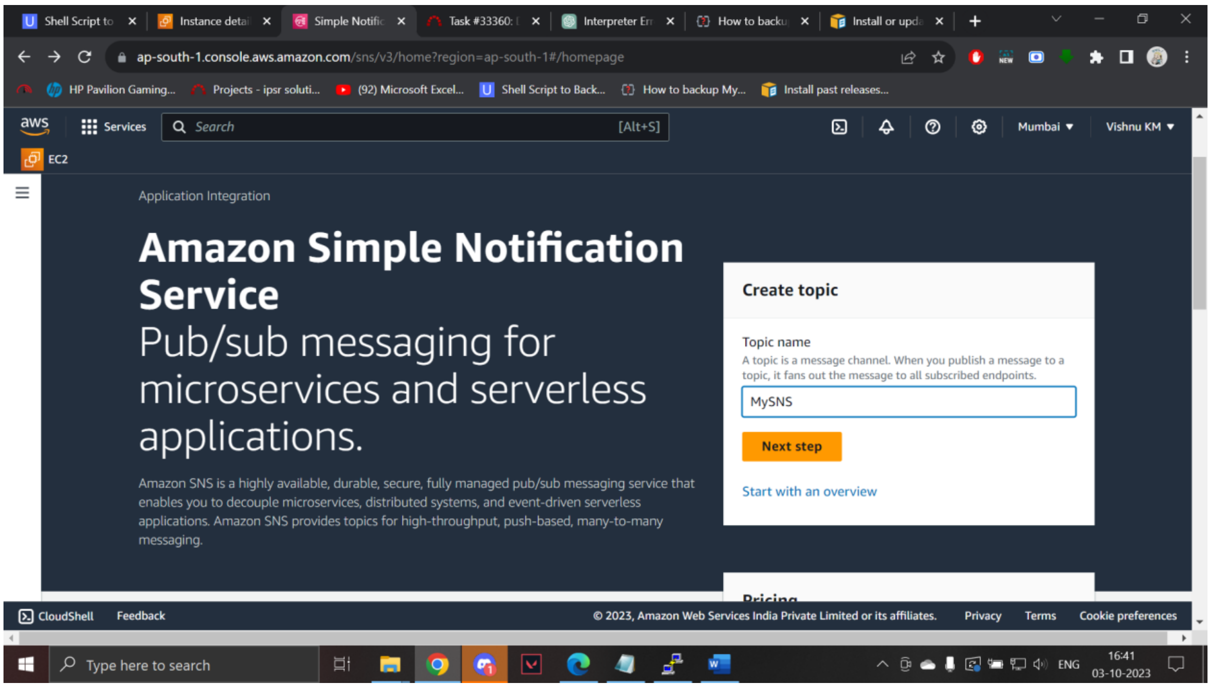
Disk space utilization notification using SNS

Amazon **Simple Notification Service** (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as *producers* and *consumers*). Publishers communicate asynchronously with subscribers by sending messages to a *topic*, which is a logical access point and communication channel. Clients can subscribe to the SNS topic and receive published messages using a supported endpoint type, such as Amazon Kinesis Data Firehose, Amazon SQS, AWS Lambda, HTTP, email, mobile push notifications, and mobile text messages (SMS).

Benefits

- **Instantaneous delivery**
  SNS is based on push-based delivery. This is the key difference between SNS and SQS. SNS is pushed once you publish the message in a topic and the message is delivered to multiple subscribers.
- **Flexible**
  SNS supports multiple endpoint types. Multiple endpoint types can receive the message over multiple transport protocols such as email, SMS, Lambda, Amazon SQS, HTTP, etc.
- **Inexpensive**
  SNS service is quite inexpensive as it is based on pay-as-you-go model, i.e., you need to pay only when you are using the resources with no up-front costs.
- **Ease of use**
  SNS service is very simple to use as Web-based AWS Management Console offers the simplicity of the point-and-click interface.
- **Simple Architecture**
  SNS is used to simplify the messaging architecture by offloading the message filtering logic from the subscribers and message routing logic from the publishers. Instead of receiving all the messages from the topic, SNS sends the message to subscriber-only of their interest.

Step 1 : Create an SNS topic



Application Integration

Amazon Simple Notification Service

Pub/sub messaging for microservices and serverless applications.

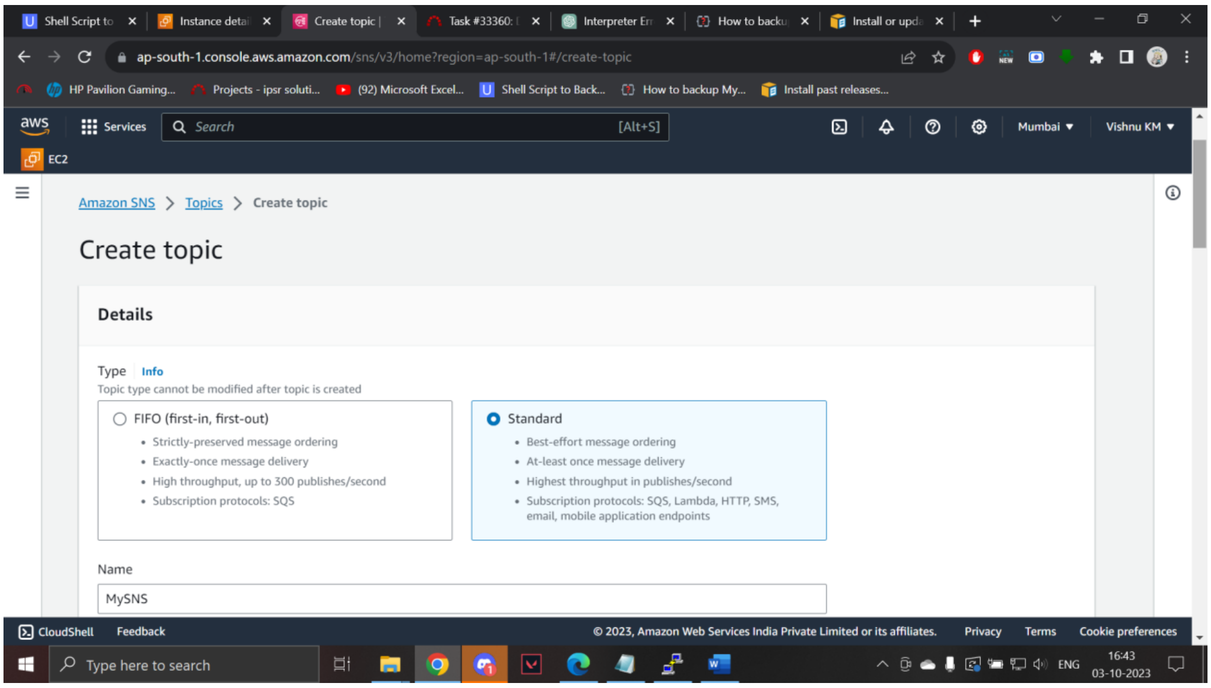
Amazon SNS is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and event-driven serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.

Create topic

Topic name

A topic is a message channel. When you publish a message to a topic, it fans out the message to all subscribed endpoints.

MySNS



Amazon SNS > Topics > Create topic

Create topic

Details

Type Info

Topic type cannot be modified after topic is created

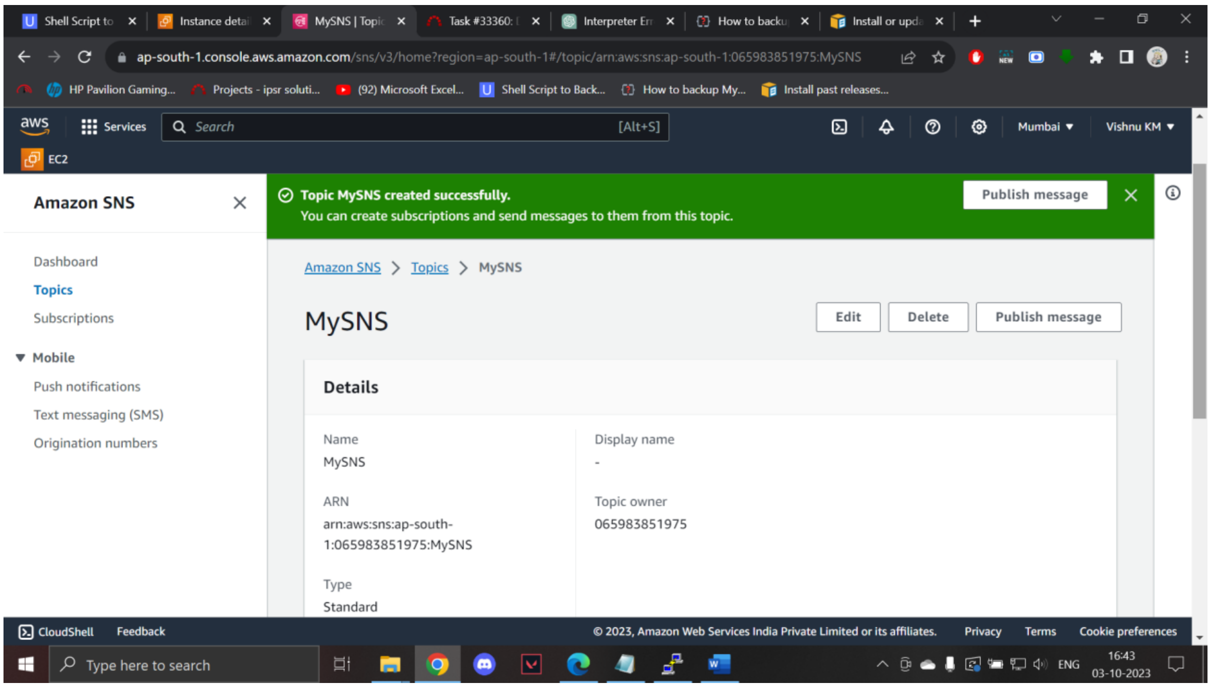
FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

[✓ 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

MySNS

We have now created our SNS topic

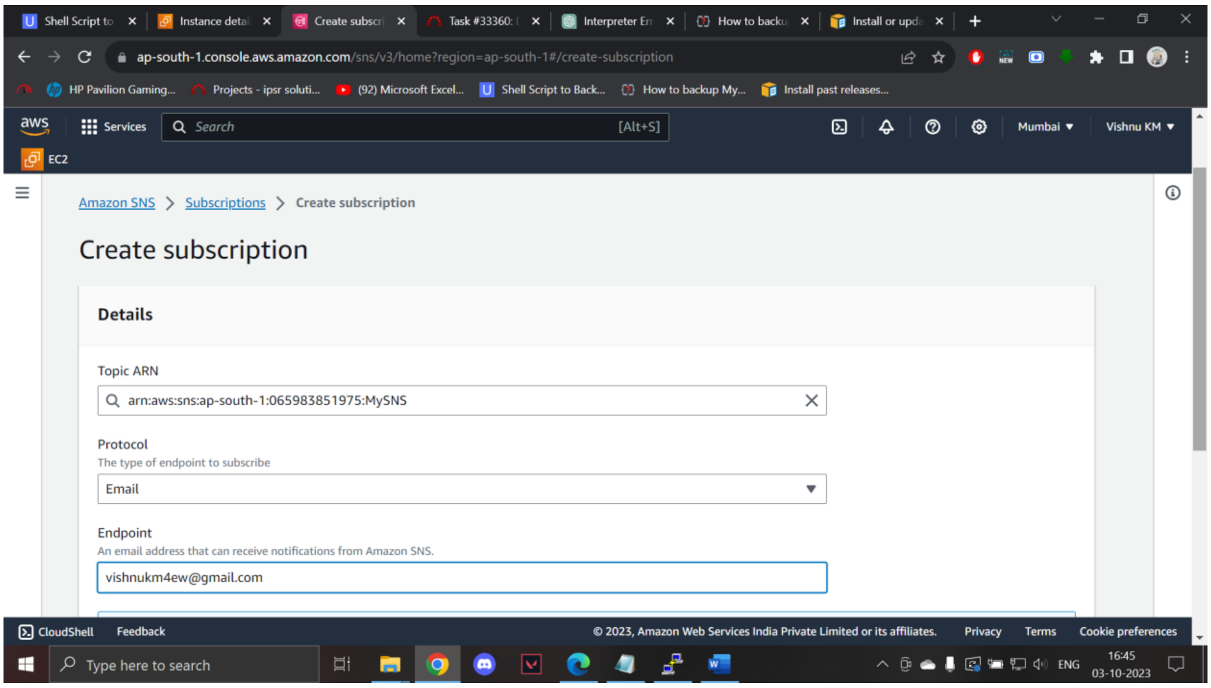

Topic MySNS created successfully.
You can create subscriptions and send messages to them from this topic.

MySNS

Details

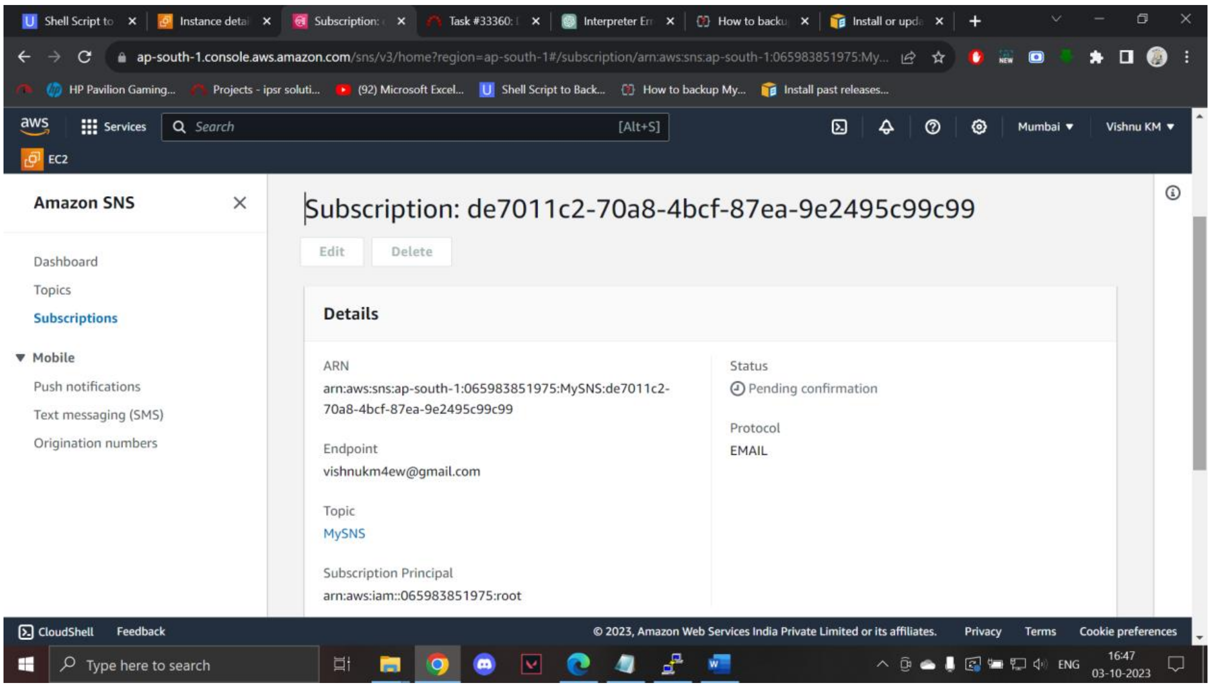
Name: MySNS
Display name: -
ARN: arn:aws:sns:ap-south-1:065983851975:MySNS
Topic owner: 065983851975
Type: Standard

Step 2 : Create a subscription policy


Create subscription

Details

Topic ARN: arn:aws:sns:ap-south-1:065983851975:MySNS
Protocol (The type of endpoint to subscribe): Email
Endpoint (An email address that can receive notifications from Amazon SNS.): vishnukm4ew@gmail.com



Subscription: de7011c2-70a8-4bcf-87ea-9e2495c99c99

Details

ARN
arn:aws:sns:ap-south-1:065983851975:MySNS:de7011c2-70a8-4bcf-87ea-9e2495c99c99

Endpoint
vishnukm4ew@gmail.com

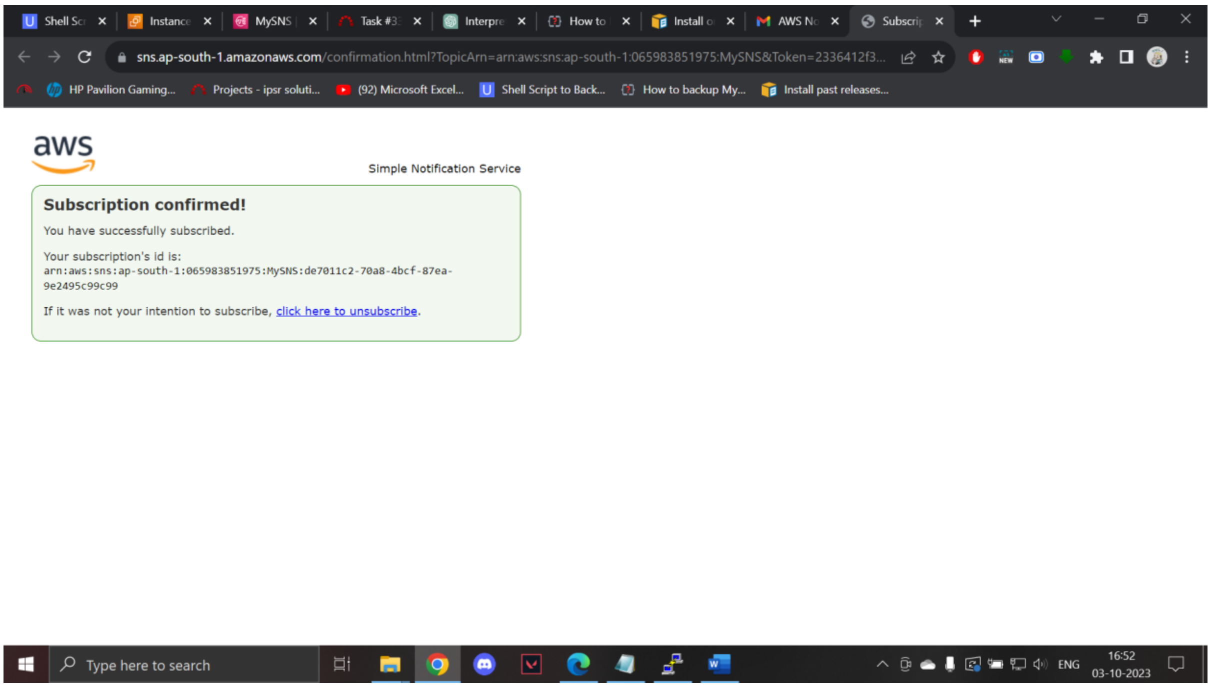
Topic
MySNS

Subscription Principal
arn:aws:iam::065983851975:root

Status
Pending confirmation

Protocol
EMAIL

Now confirm the subscription by clicking on the 'confirm subscription' mail sent to you



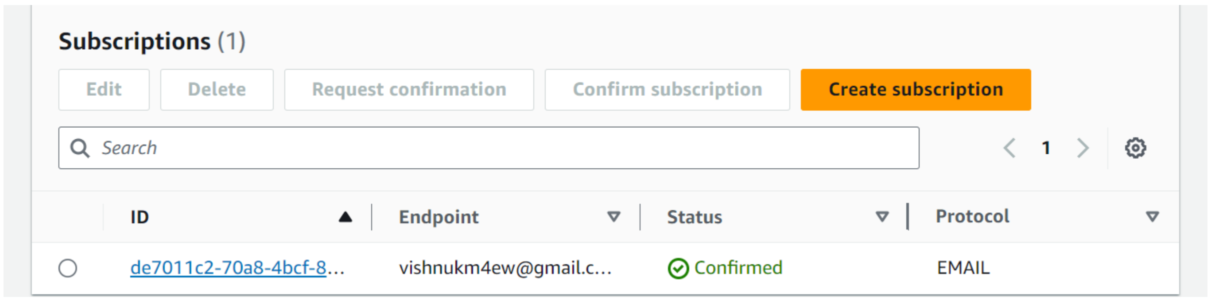
Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:ap-south-1:065983851975:MySNS:de7011c2-70a8-4bcf-87ea-9e2495c99c99

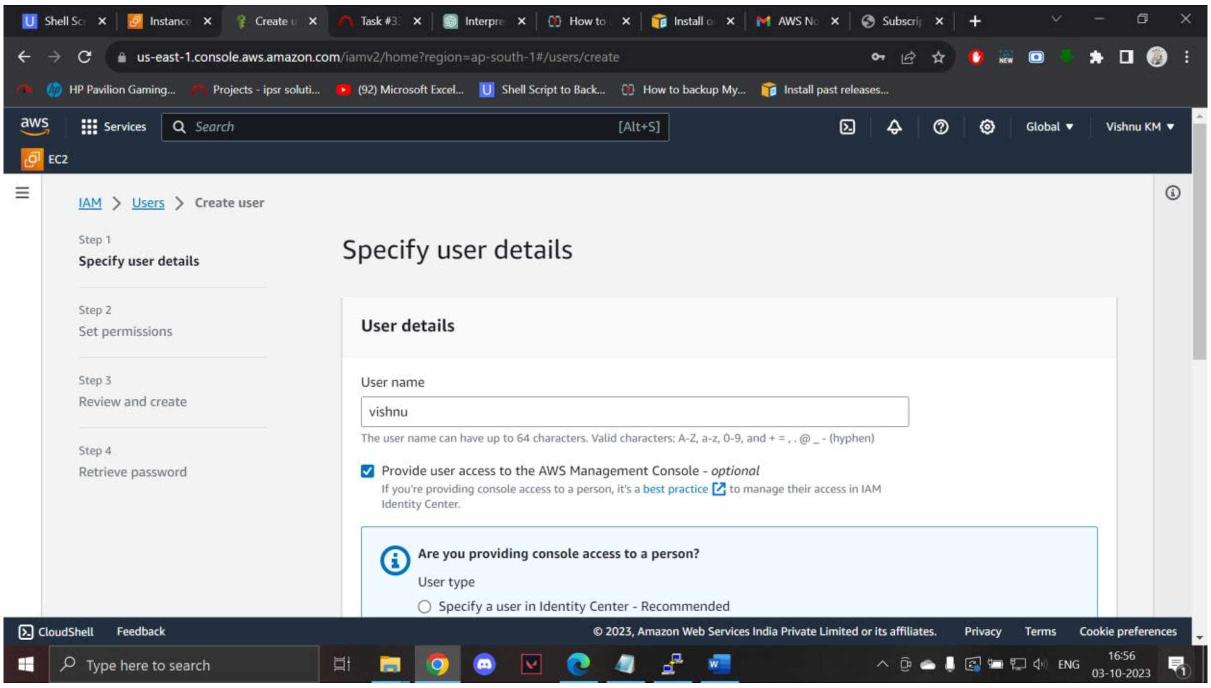
If it was not your intention to subscribe, click here to unsubscribe.



Subscriptions (1)

ID	Endpoint	Status	Protocol
de7011c2-70a8-4bcf-8…	vishnukm4ew@gmail.c…	Confirmed	EMAIL

Step 3 : Create an IAM user



us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/create

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Specify user details

User details

User name: vishnu

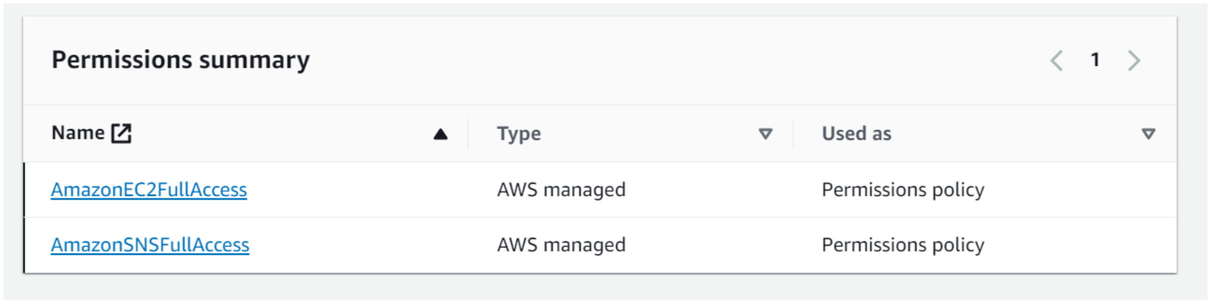
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

[✓] Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

Are you providing console access to a person?
User type
○ Specify a user in Identity Center - Recommended

© 2023, Amazon Web Services India Private Limited or its affiliates.

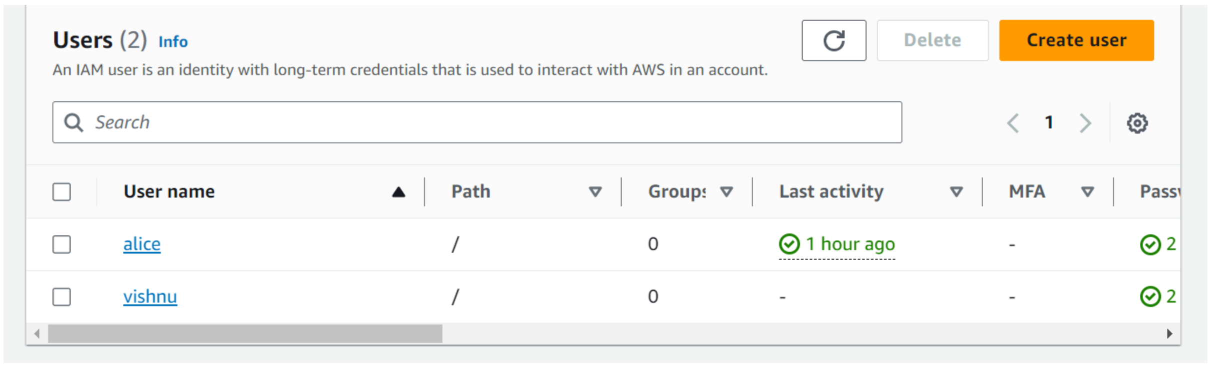
Assign it with the policies - AmazonEC2FullAccess , AmazonSNSFullAccess



Permissions summary

Name	Type	Used as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonSNSFullAccess	AWS managed	Permissions policy

Our user had been created



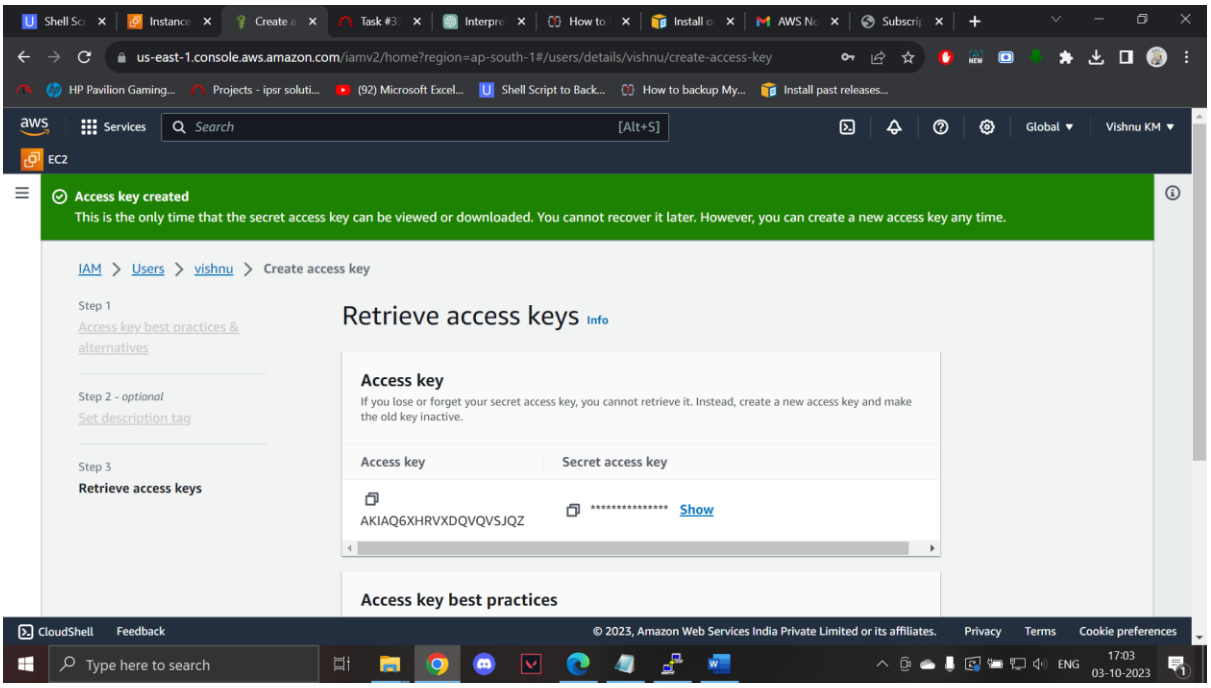
Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

	User name	Path	Groups	Last activity	MFA	Pass
	alice	/	0	1 hour ago	-	2
	vishnu	/	0	-	-	2

Setting up Access key

Click on the user and you can find an option for creating access keys



us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/vishnu/create-access-key

Access key created

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.

IAM > Users > vishnu > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAQ6XHRVXDQVQVSJQZ	*************** Show

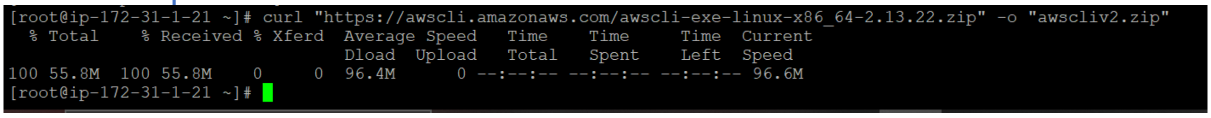
Access key best practices

© 2023, Amazon Web Services India Private Limited or its affiliates.

Step 4 : Install AWS CLI and configure it

Download the installation file

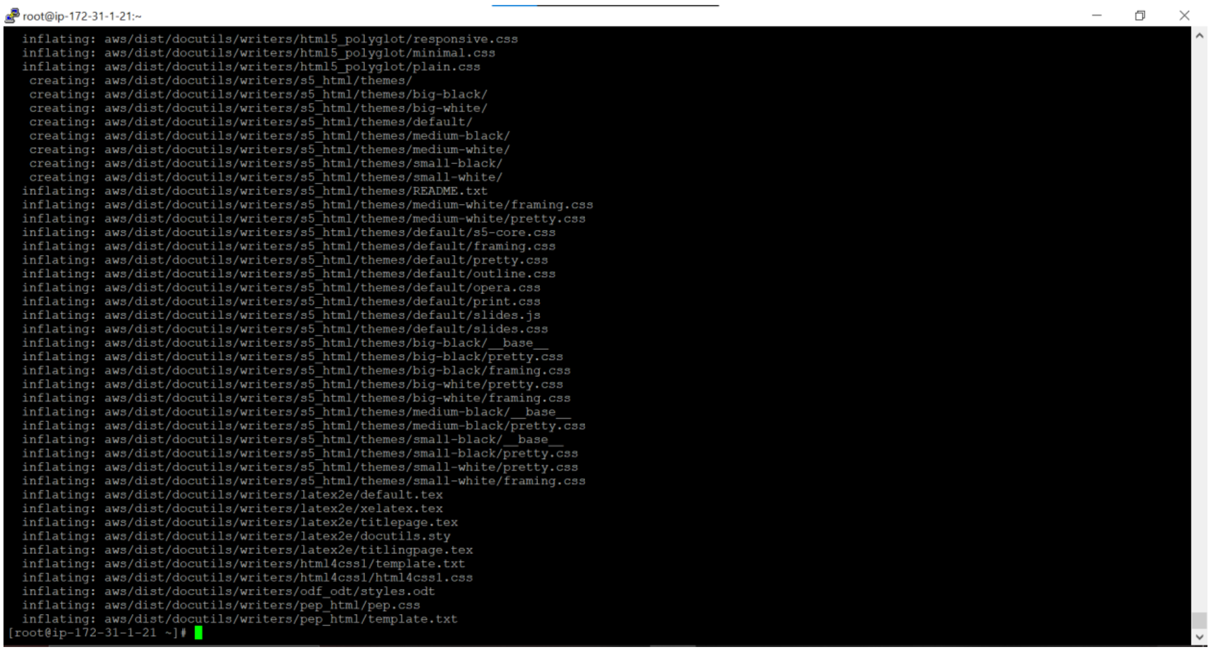
#curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"



```
[root@ip-172-31-1-21 ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.13.22.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 55.8M  100 55.8M    0     0  96.4M      0 --:--:-- --:--:-- --:--:-- 96.6M
[root@ip-172-31-1-21 ~]#
```

Unzip the installer.

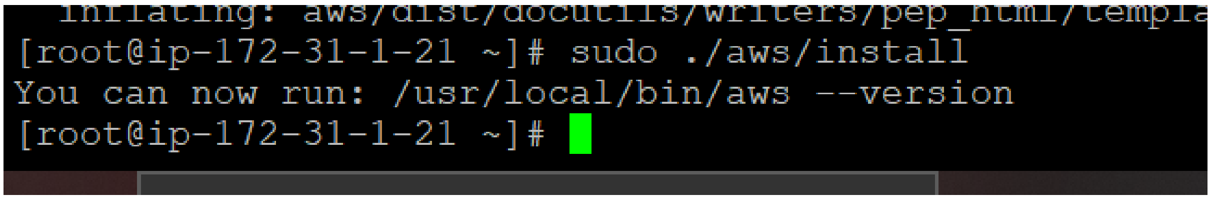
#yum install unzip -y
#unzip awscliv2.zip



```
root@ip-172-31-1-21:~
  inflating: aws/dist/docutils/writers/html5_polyglot/responsive.css
  inflating: aws/dist/docutils/writers/html5_polyglot/minimal.css
  inflating: aws/dist/docutils/writers/html5_polyglot/plain.css
   creating: aws/dist/docutils/writers/s5_html/themes/
   creating: aws/dist/docutils/writers/s5_html/themes/big-black/
   creating: aws/dist/docutils/writers/s5_html/themes/big-white/
   creating: aws/dist/docutils/writers/s5_html/themes/default/
   creating: aws/dist/docutils/writers/s5_html/themes/medium-black/
   creating: aws/dist/docutils/writers/s5_html/themes/medium-white/
   creating: aws/dist/docutils/writers/s5_html/themes/small-black/
   creating: aws/dist/docutils/writers/s5_html/themes/small-white/
  inflating: aws/dist/docutils/writers/s5_html/themes/README.txt
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-white/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-white/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/s5-core.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/outline.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/opera.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/print.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/slides.js
  inflating: aws/dist/docutils/writers/s5_html/themes/default/slides.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-black/__base__
  inflating: aws/dist/docutils/writers/s5_html/themes/big-black/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-black/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-white/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-white/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-black/__base__
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-black/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/small-black/__base__
  inflating: aws/dist/docutils/writers/s5_html/themes/small-black/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/small-white/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/small-white/framing.css
  inflating: aws/dist/docutils/writers/latex2e/default.tex
  inflating: aws/dist/docutils/writers/latex2e/xelatex.tex
  inflating: aws/dist/docutils/writers/latex2e/titlepage.tex
  inflating: aws/dist/docutils/writers/latex2e/docutils.sty
  inflating: aws/dist/docutils/writers/latex2e/titlingpage.tex
  inflating: aws/dist/docutils/writers/html4css1/template.txt
  inflating: aws/dist/docutils/writers/html4css1/html4css1.css
  inflating: aws/dist/docutils/writers/odf_odt/styles.odt
  inflating: aws/dist/docutils/writers/pep_html/pep.css
  inflating: aws/dist/docutils/writers/pep_html/template.txt
[root@ip-172-31-1-21 ~]#
```

Run the install program.

#sudo ./aws/install



```
  inflating: aws/dist/docutils/writers/pep_html/templa
[root@ip-172-31-1-21 ~]# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
[root@ip-172-31-1-21 ~]#
```

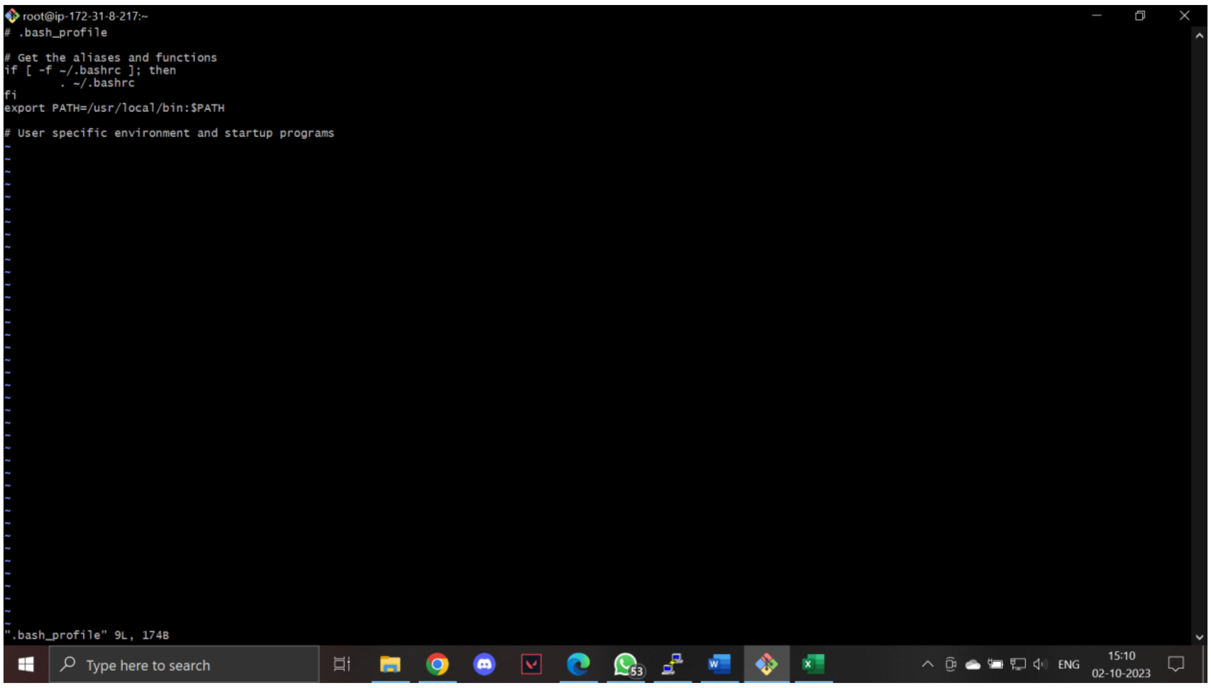
#vim .bash_profile

Add the following lines

export PATH=/usr/local/bin:$PATH

By doing this, you're telling your shell to look in "/usr/local/bin" first when searching for executable files, and then it will search in the other directories listed in the original **PATH** variable.

This can be useful if you have custom software or scripts installed in "/usr/local/bin" that you want to use without specifying the full path every time you run them. Just be aware that if you have similarly named commands in other directories in your original **PATH**, the one in "/usr/local/bin" will take precedence.

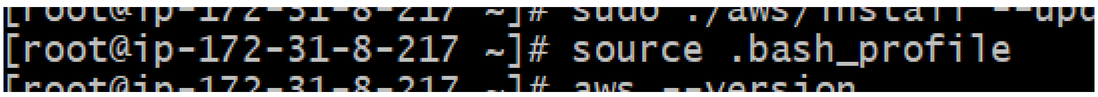


```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
export PATH=/usr/local/bin:$PATH

# User specific environment and startup programs
~
~
~
".bash_profile" 9L, 174B
```

#source .bash_profile

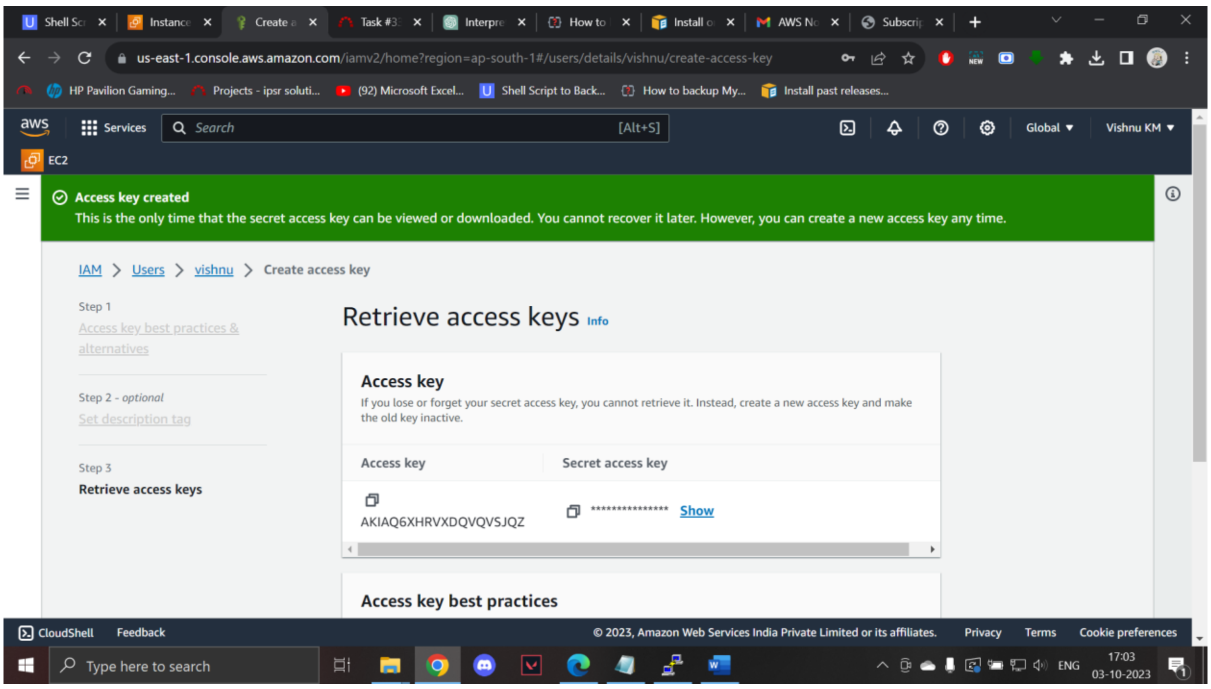
After running **source .bash_profile**, any changes you've made to your **.bash_profile** file will take effect immediately in your current terminal session. This can include setting environment variables, modifying your **PATH**, defining new aliases, or performing other customizations to your shell environment.

It's a handy command to use when you want to test changes to your shell configuration without having to close and reopen your terminal session.



```
[root@ip-172-31-8-217 ~]# sudo ./aws/install --upd
[root@ip-172-31-8-217 ~]# source .bash_profile
[root@ip-172-31-8-217 ~]# aws --version
```

We can find the access key id and the secret access key from the csv file once we create an access key



Access key created
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.

IAM > Users > vishnu > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

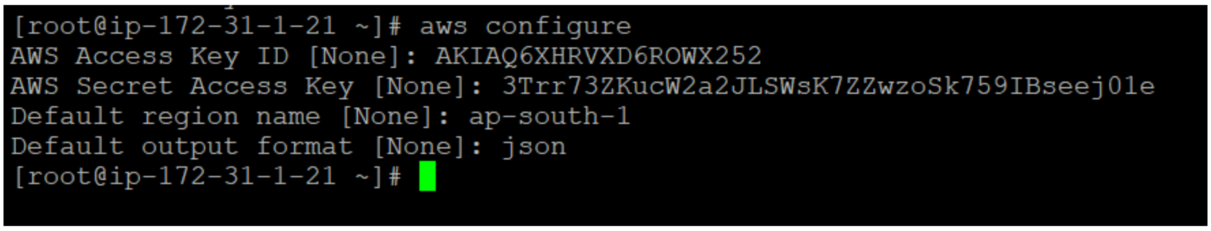
Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAQ6XHRVXDQVQVSJQZ	*************** Show

Access key best practices

© 2023, Amazon Web Services India Private Limited or its affiliates.

#aws configure



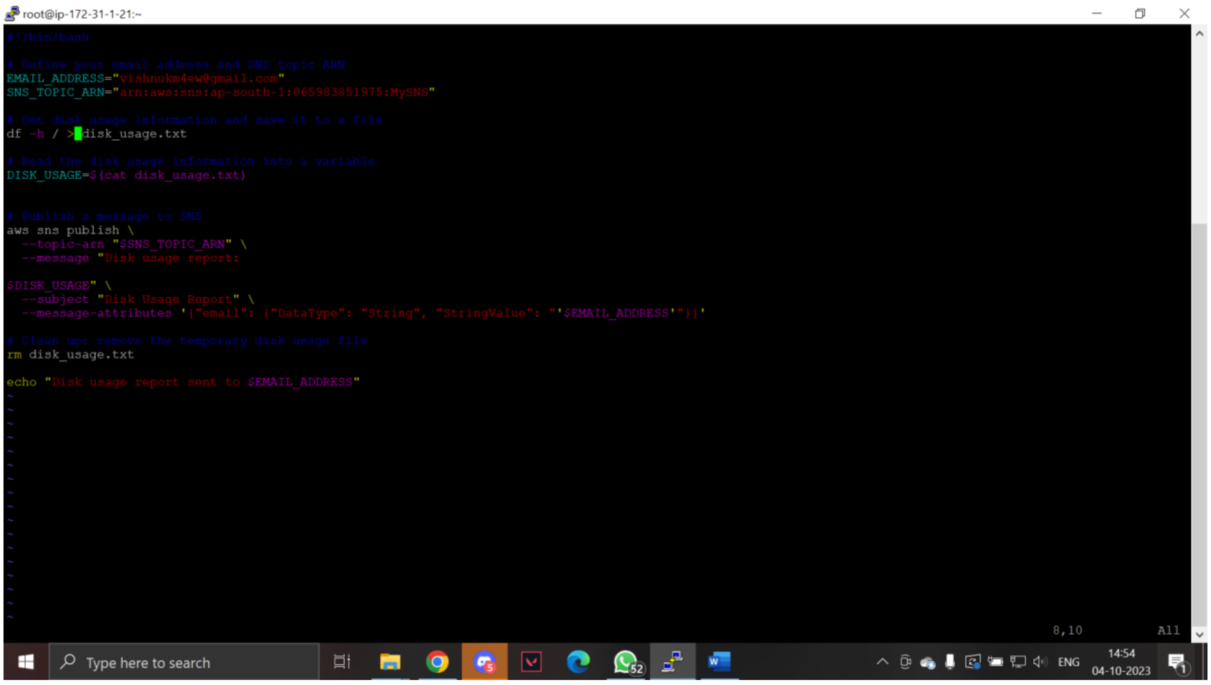
```
[root@ip-172-31-1-21 ~]# aws configure
AWS Access Key ID [None]: AKIAQ6XHRVXD6ROWX252
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
[root@ip-172-31-1-21 ~]#
```

Step 5 : Write a script file to display the disk usage notification as mail

```
#vim diskusage.sh
```

Add the following lines

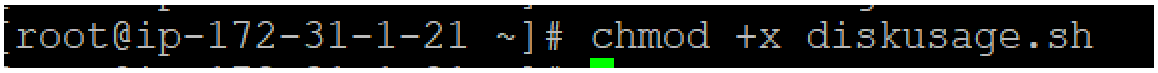
```
#!/bin/bash

# Define your email address and SNS topic ARN
EMAIL_ADDRESS="vishnukm4ew@gmail.com"
SNS_TOPIC_ARN="arn:aws:sns:ap-south-1:065983851975:MySNS"

# Get disk usage information and save it to a file
df -h /> disk_usage.txt

# Read the disk usage information into a variable
DISK_USAGE=$(cat disk_usage.txt)


# Publish a message to SNS
aws sns publish \
  --topic-arn "$SNS_TOPIC_ARN" \
  --message "Disk usage report:

$DISK_USAGE" \
  --subject "Disk Usage Report" \
  --message-attributes '{"email": {"DataType": "String", "StringValue":
"'$EMAIL_ADDRESS'"}}'

# Clean up: remove the temporary disk usage file
rm disk_usage.txt

echo "Disk usage report sent to $EMAIL_ADDRESS"
```



```
#!/bin/bash

# Define your email address and SNS topic ARN
EMAIL_ADDRESS="vishnukm4ew@gmail.com"
SNS_TOPIC_ARN="arn:aws:sns:ap-south-1:065983851975:MySNS"

# Get disk usage information and save it to a file
df -h / > disk_usage.txt

# Read the disk usage information into a variable
DISK_USAGE=$(cat disk_usage.txt)


# Publish a message to SNS
aws sns publish \
  --topic-arn "$SNS_TOPIC_ARN" \
  --message "Disk usage report:

$DISK_USAGE" \
  --subject "Disk Usage Report" \
  --message-attributes '{"email": {"DataType": "String", "StringValue": "'$EMAIL_ADDRESS'"}}'

# Clean up: remove the temporary disk usage file
rm disk_usage.txt

echo "Disk usage report sent to $EMAIL_ADDRESS"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                    8,10          All
```

Step 6 : Give execute permission to the script file

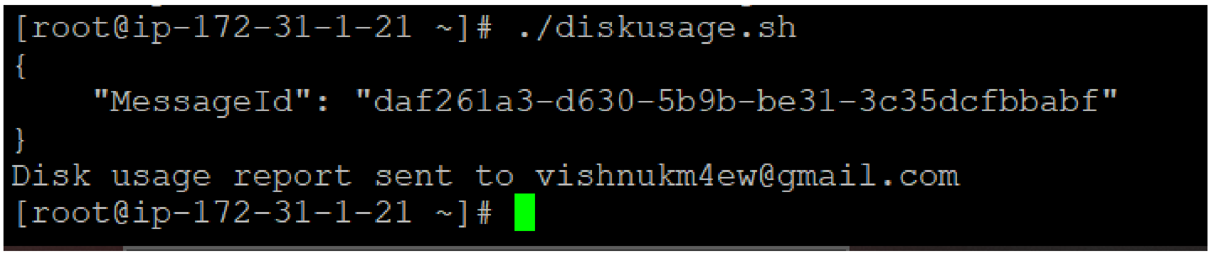
#chmod +x diskusage.sh



```
[root@ip-172-31-1-21 ~]# chmod +x diskusage.sh
```

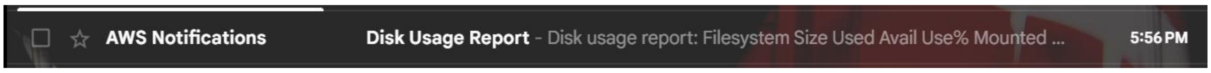
Step 7 : You can now run the script

#./ diskusage.sh

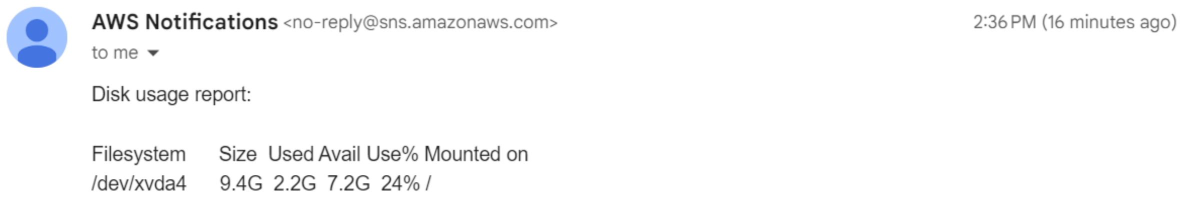


```
[root@ip-172-31-1-21 ~]# ./diskusage.sh
{
    "MessageId": "daf261a3-d630-5b9b-be31-3c35dcfbbabf"
}
Disk usage report sent to vishnukm4ew@gmail.com
[root@ip-172-31-1-21 ~]#
```

You will get the mail as shown



AWS Notifications   **Disk Usage Report** - Disk usage report: Filesystem Size Used Avail Use% Mounted ...   5:56 PM



AWS Notifications <no-reply@sns.amazonaws.com> 2:36 PM (16 minutes ago)
to me

Disk usage report:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda4      9.4G  2.2G  7.2G  24% /
```

Note : I used alice from previous task (s3 backup scripting) and just assigned alice with the 2 new policies – snsfullacess and ec2fullaccess