AWS Documentation

No:-	Content
1.	CloudWatch Theory
2.	Cloud Dashboard
3.	Creation of custom namespace
4.	Events

CloudWatch Theory:

1. The Notebook's Eyes: Monitoring

- **CloudWatch** has eyes everywhere! It watches your cloud house, checking the health of your servers, applications, and even your coffee machine (well, not really, but almost anything in AWS).
- It looks for things like how much memory your servers are using, how much traffic your website is getting, or if anything is acting funny (errors).

2. Magic Alarms: Alerts

- If **CloudWatch** sees something weird—like your server getting too hot or your website getting too much traffic—it rings a **magic alarm**.
- This alarm can send you a message, so you know something needs fixing before it breaks.

3. Time Machine Pages: Logs

- **CloudWatch** also has time machine pages. It writes down everything that happens so you can go back and look if something goes wrong.

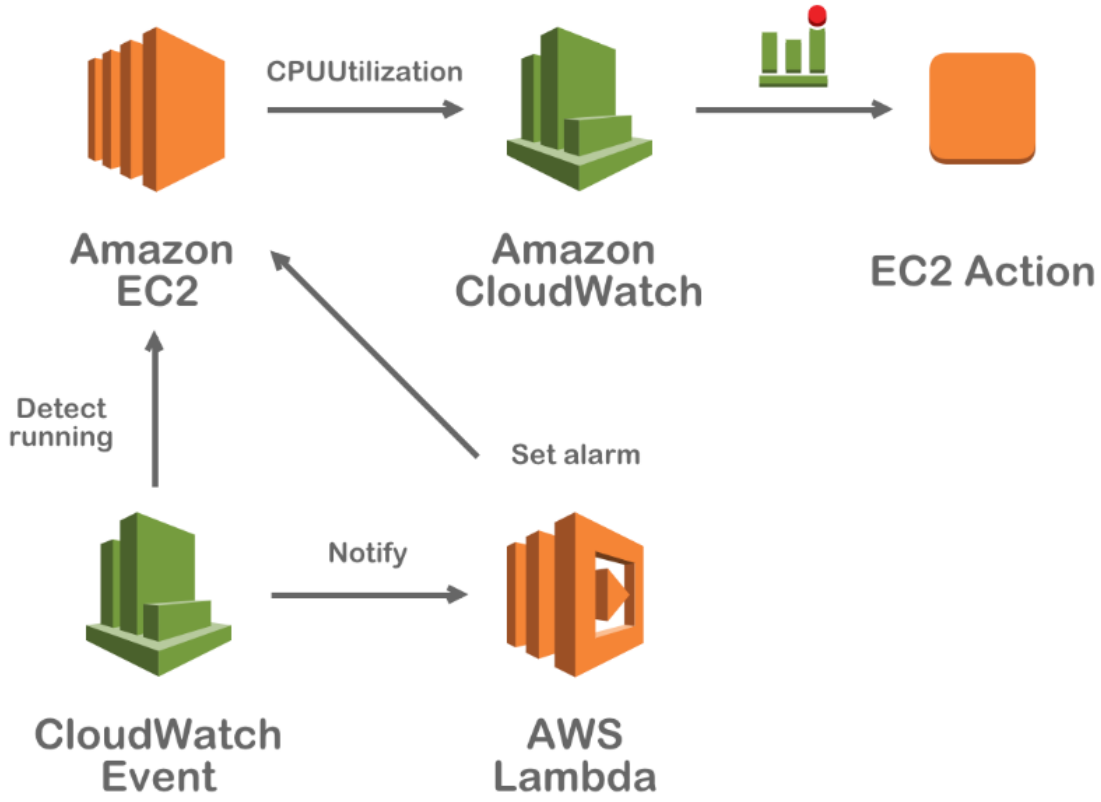
- These pages (logs) help you understand what happened in the past, like why your server stopped working at midnight.

4. Smart Insights: Metrics

- The notebook doesn't just watch and write—it's smart! **CloudWatch** can look at all the data it's collected and give you insights.
- For example, it can show you how your server's performance has changed over time or if your website's traffic is growing.

5. Custom Spells: Custom Metrics

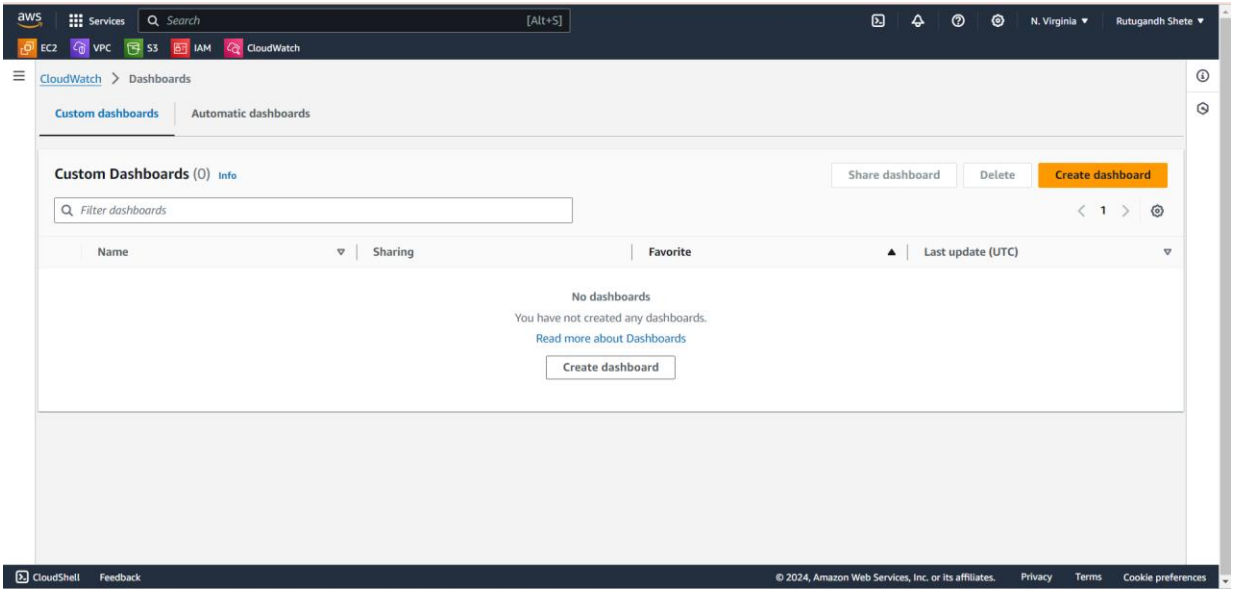
- You can teach **CloudWatch** to look for specific things that matter to you, like how many orders your online store gets every hour.
- These are called **custom metrics**—you create the spells (rules), and **CloudWatch** watches them for you.



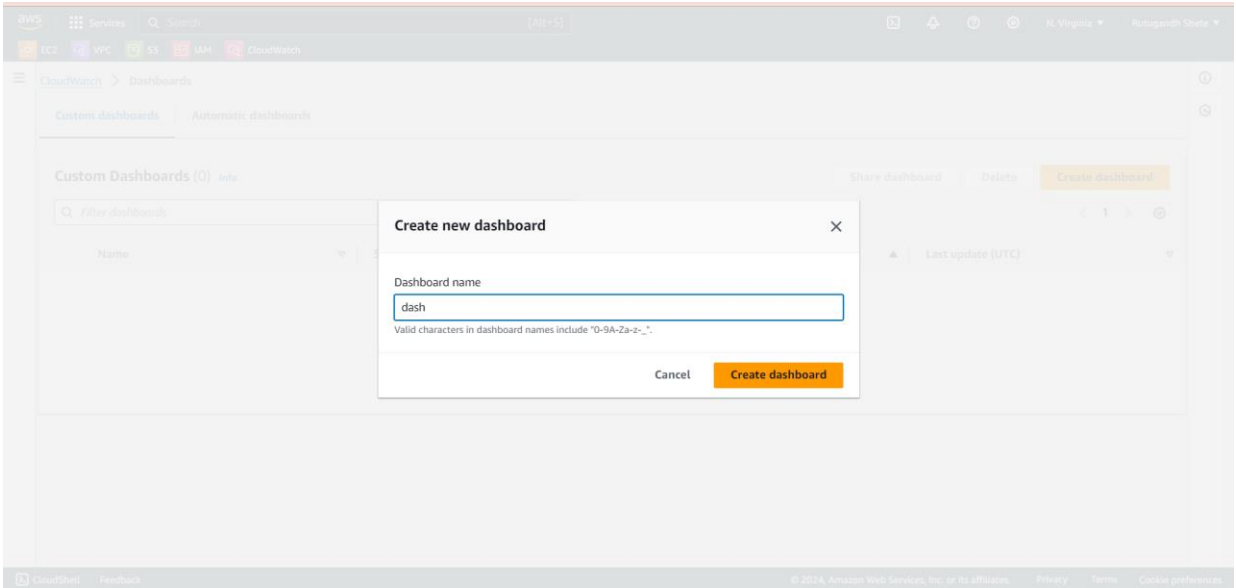
Cloud watch dashboard:

Steps:

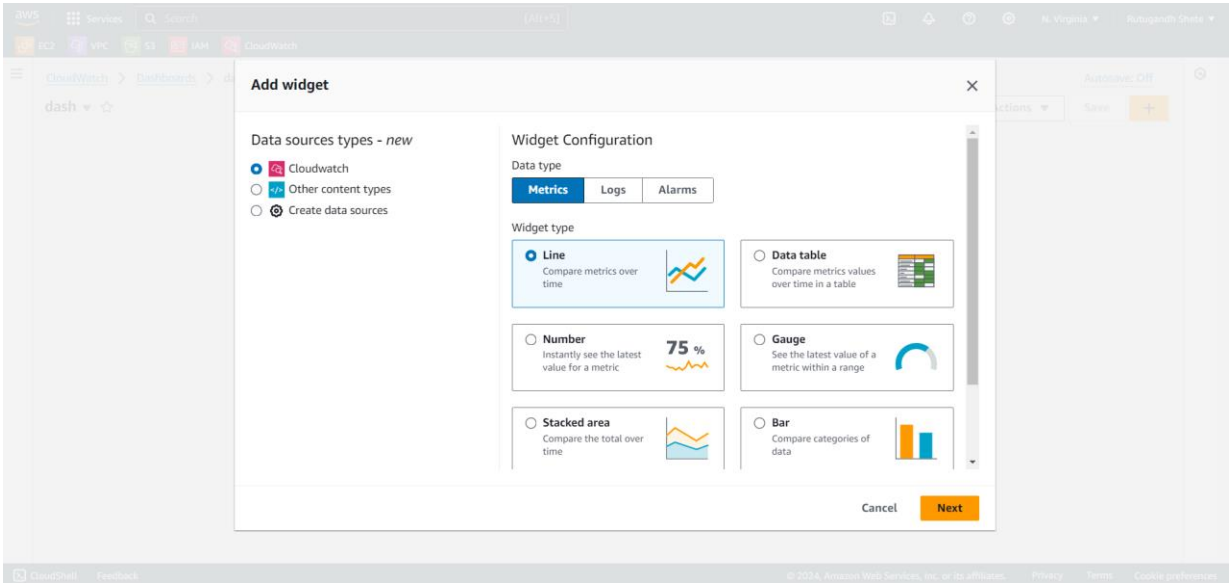
- Go to service→Create dashboard



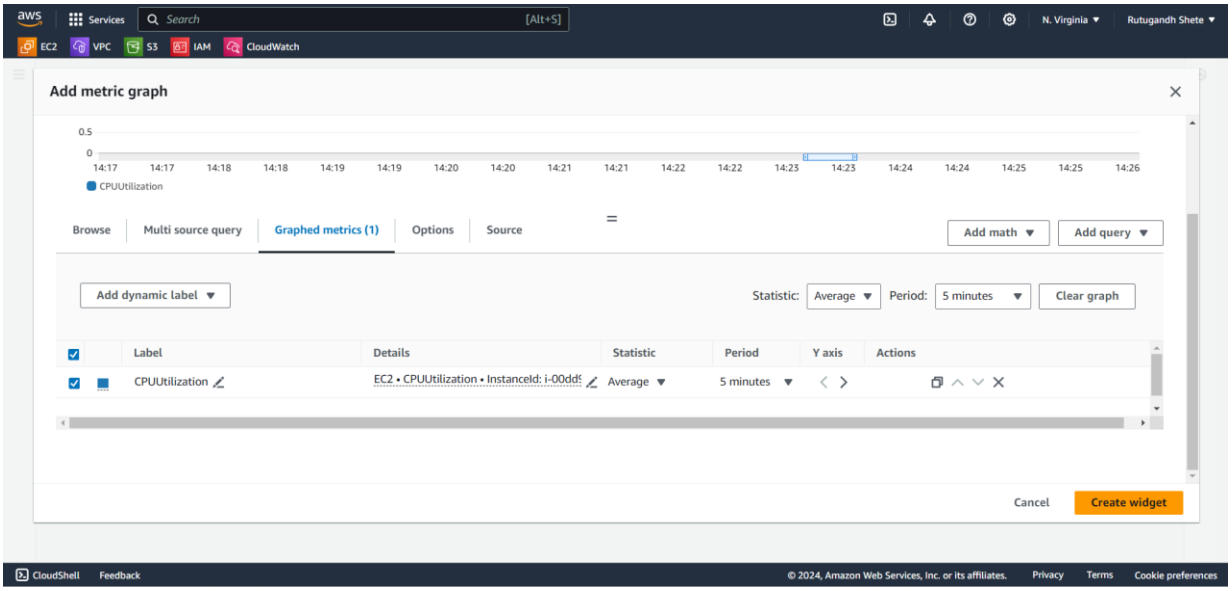
- Give name to dashboard.



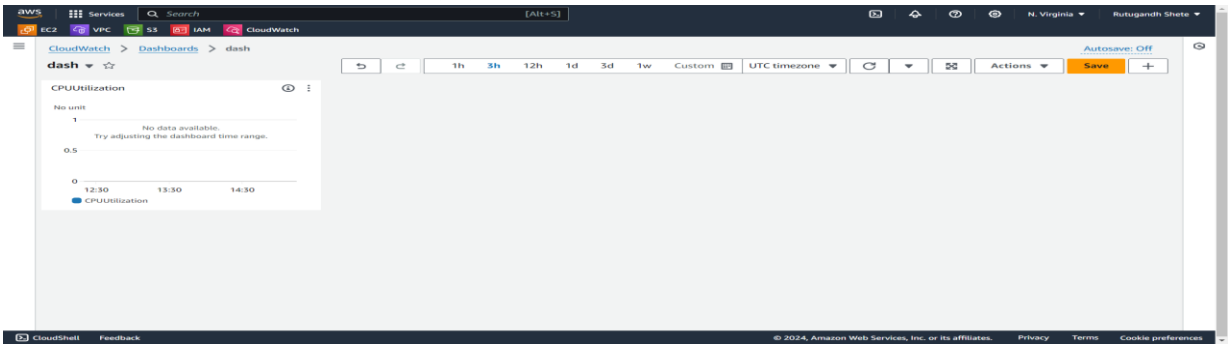
- We can see widget configuration →by choosing data type we can see the results in different forms like into different widget type like line, bars etc.



- Create widget, **Customize Display**: Adjust colors, labels, or thresholds to make the widget clear and useful.



- We can After configuring the widget, click **Add to dashboard**.
  You can drag and drop widgets to rearrange them on your dashboard.
  Click **Save dashboard** to keep your changes



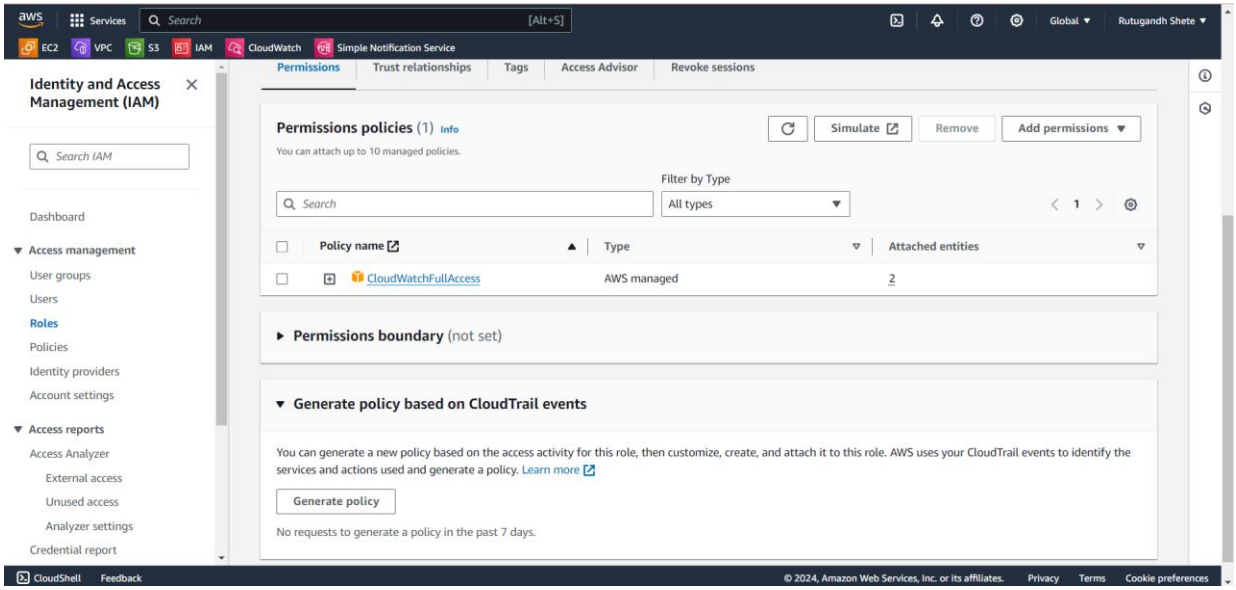
Creation of Custom Namespace:

We create custom namespace because if we want to monitor without default namespace then we use custom namespace.

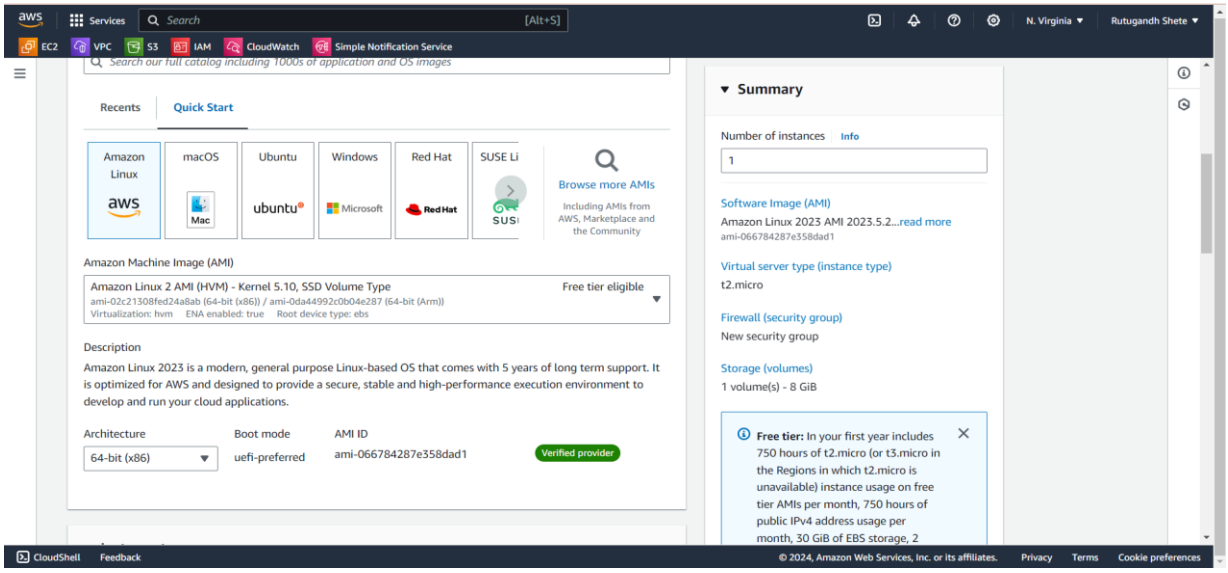
Real Example: - If you have an application that processes customer orders, you could create a custom namespace called MyApp/OrderProcessing and publish metrics like OrderProcessingTime, FailedOrders, or SuccessfulOrders.

Steps:

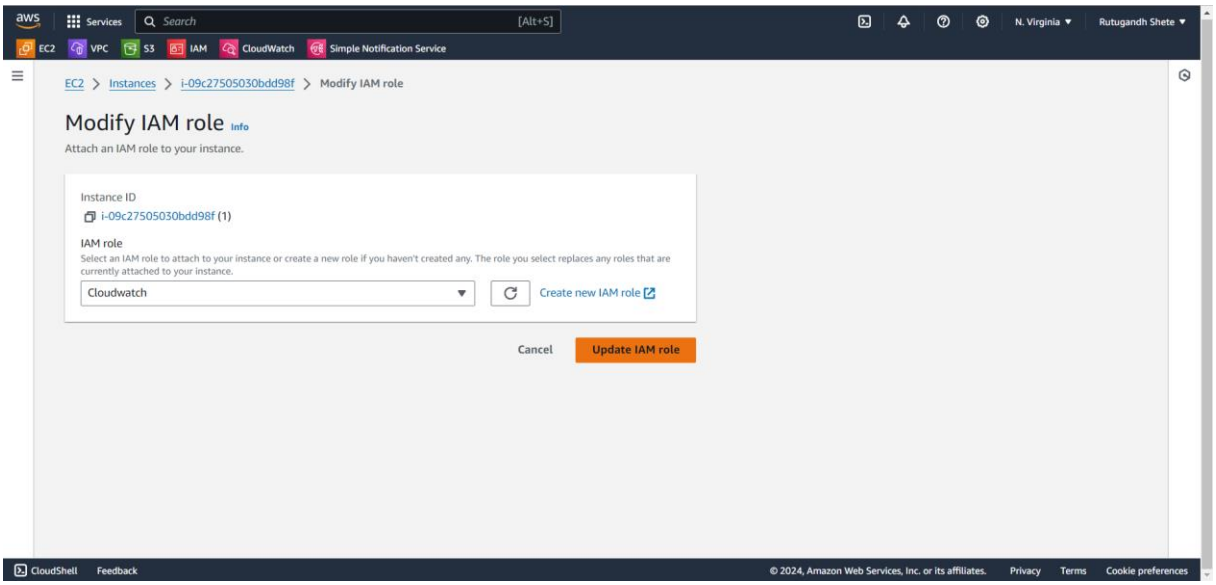
- Create IAM role with permissions “CloudwatchFullaccess”



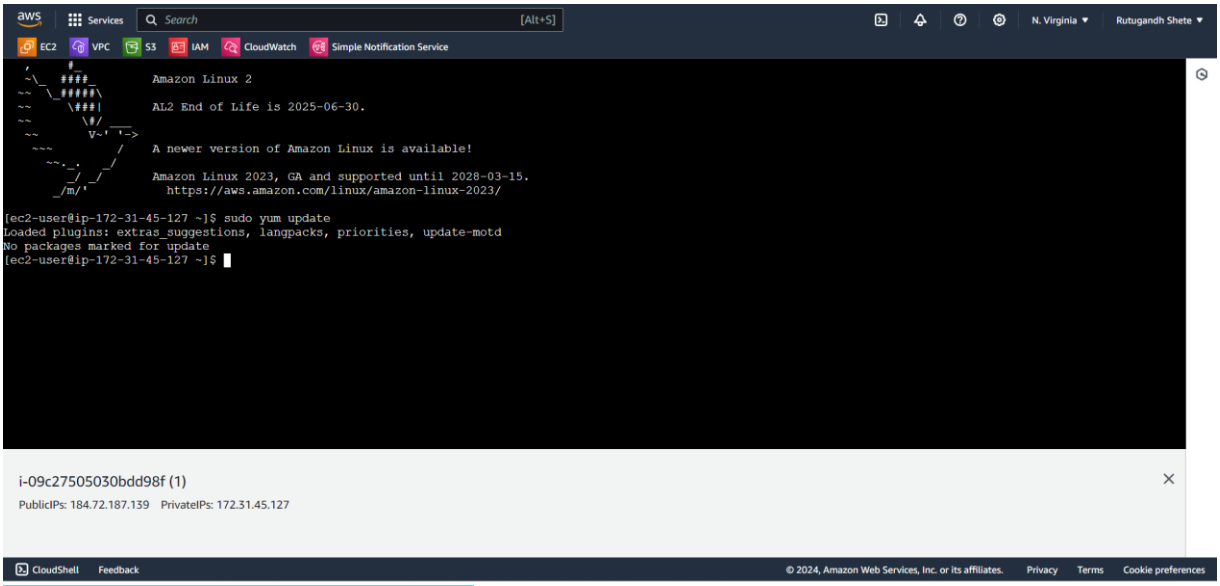
- Create instance but select AMI machine “Amazon machine 2”



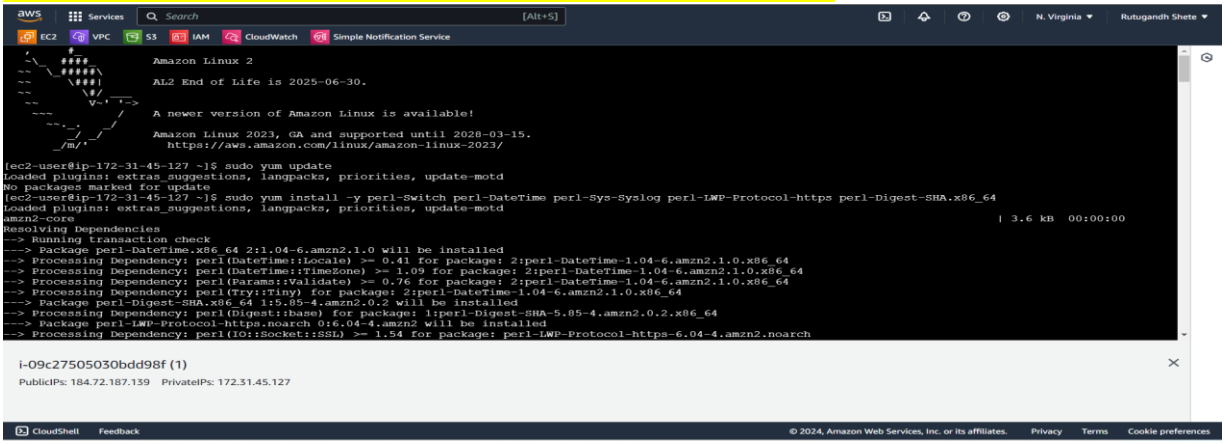
- Attach IAM role to instance.



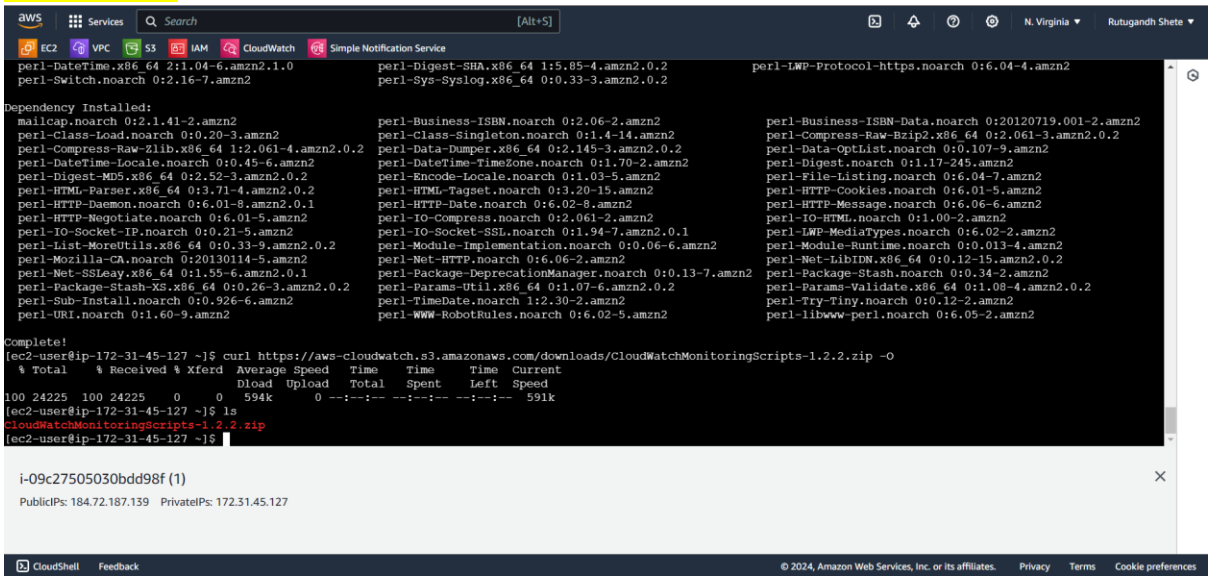
- Connet instance to terminal.



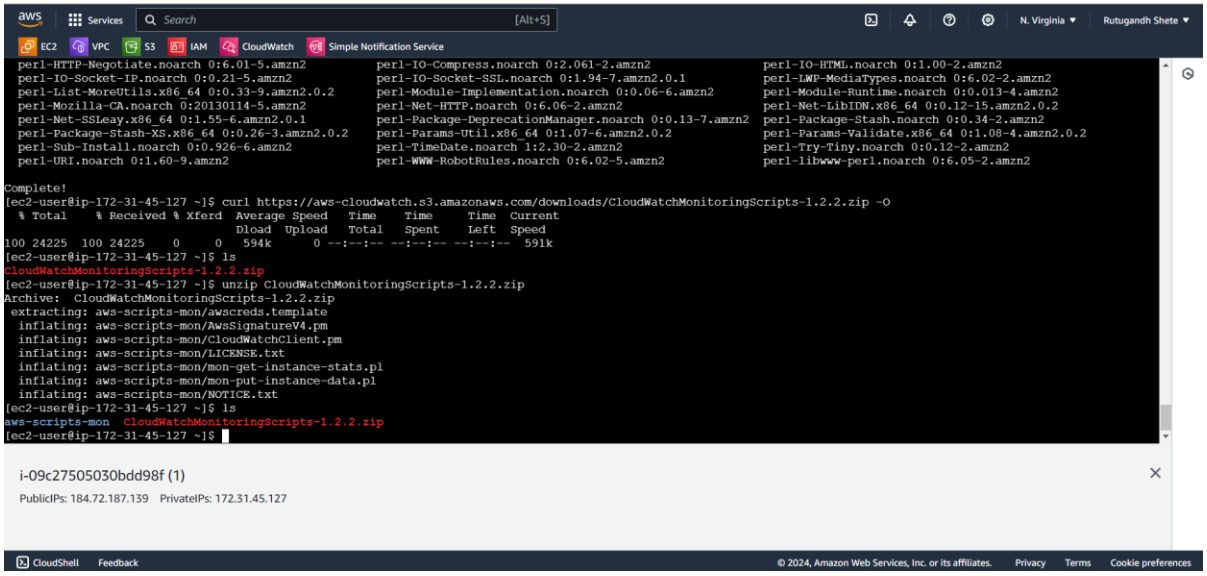
- **Install Required Perl Packages:** These packages are necessary to run the monitoring scripts. “sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86_64”.



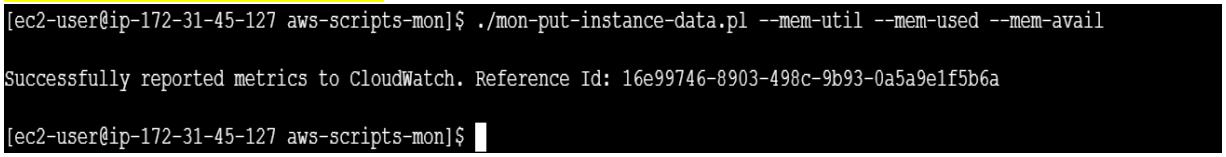
- You download a zip file containing the CloudWatch monitoring scripts, unzip it, and navigate to the folder where the scripts are stored. “curl https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip -O”



- unzip CloudWatchMonitoringScripts-1.2.2.zip



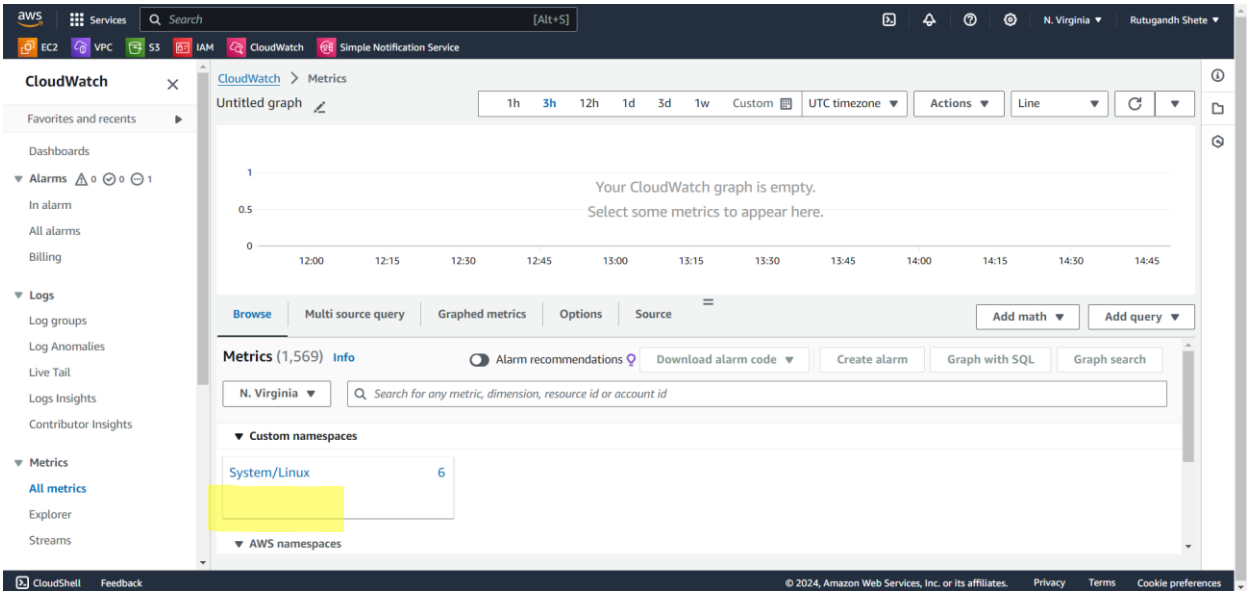
- cd into unzipped file
- You run a specific script (mon-put-instance-data.pl) to collect memory usage data (like memory utilization, used memory, and available memory) and send this data to CloudWatch. “./mon-put-instance-data.pl --mem-util --mem-used --mem-avail”



- You can set up a cron job to run this script every 5 minutes, so the memory and disk usage data is regularly sent to CloudWatch. “crontab -e” */5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl — mem-used-incl-cache-buff — mem-util — disk-space-util — disk-path=/ — from-cron

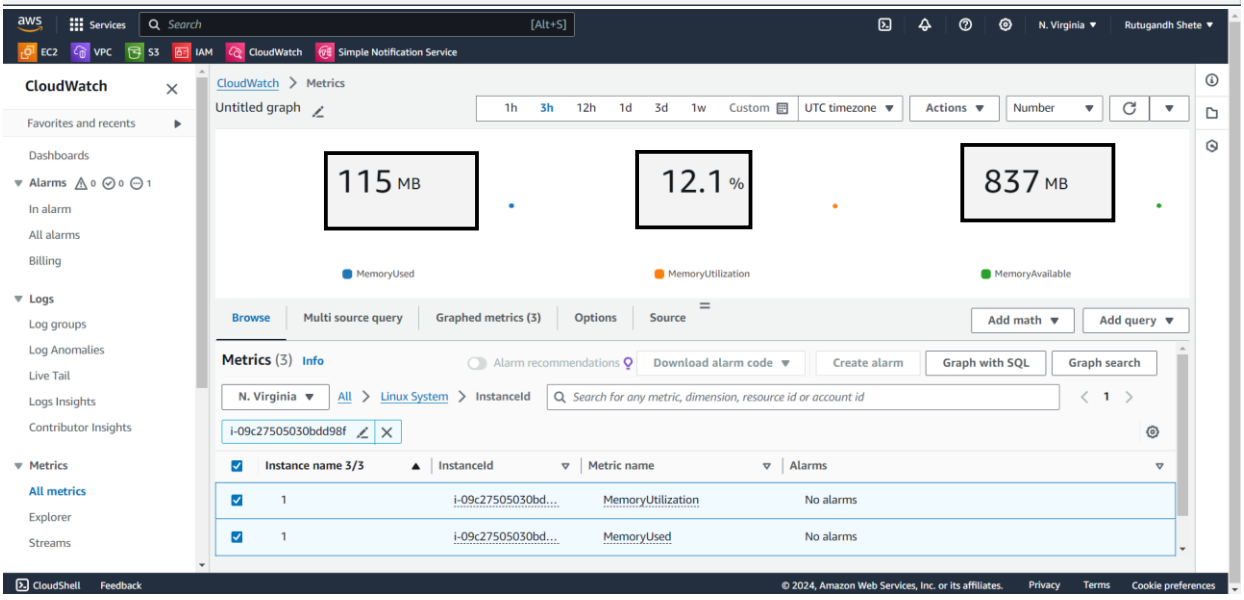
```
[ec2-user@ip-172-31-45-127 aws-scripts-mon]$ crontab -e
no crontab for ec2-user - using an empty one
crontab: installing new crontab
[ec2-user@ip-172-31-45-127 aws-scripts-mon]$ crontab -l
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl — mem-used-incl-cache-buff — mem-util — disk-space-util — disk-path=/ — from-cron
[ec2-user@ip-172-31-45-127 aws-scripts-mon]$
```

- Go to CloudWatch → Metrics → Select `All Metrics` →Select `Custom Namespaces` → Select `System/ Linux` Select `Instance Id.



- Copy instance ID see the metrics that are available

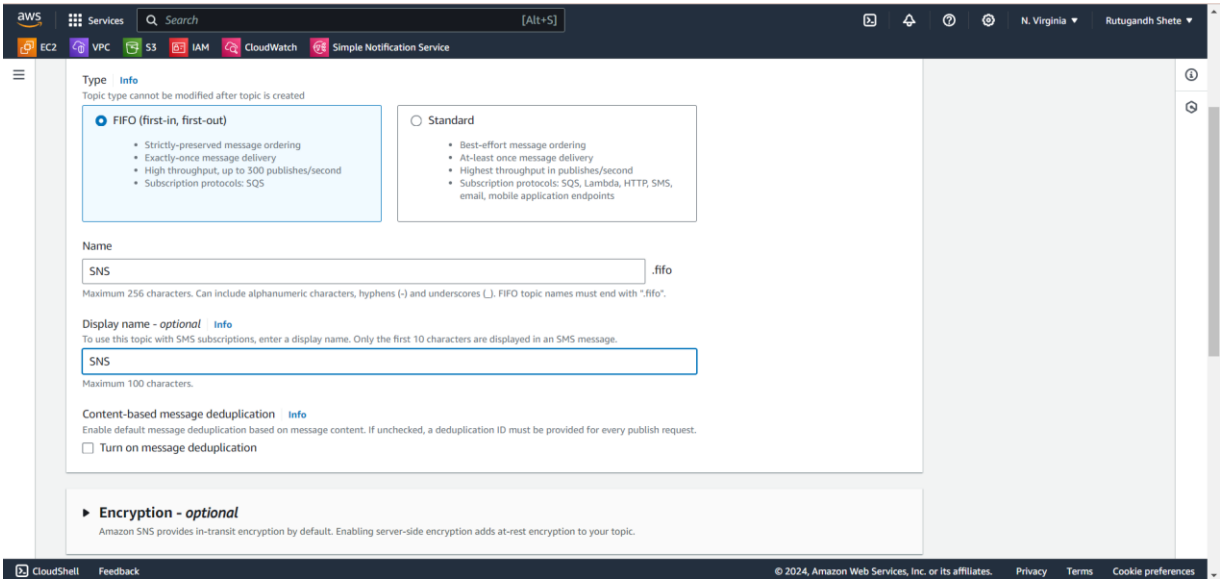
Instance name 3/3	InstanceId	Metric name	Alarms
1	i-09c2750...	MemoryUtilization	No alarms
1	i-09c27505030bd...	MemoryUsed	No alarms
1	i-09c27505030bd...	MemoryAvailable	No alarms



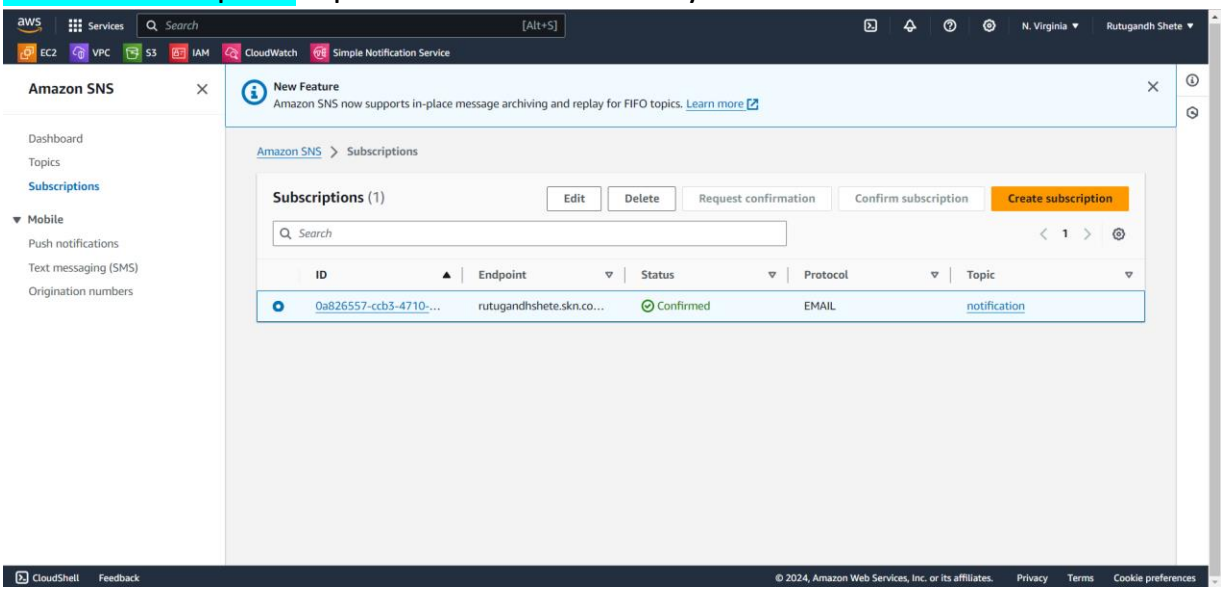
Simple notification service:

Steps:

- Create topic→Name→description



- Create subscription→provide Protocol→add your Mail ID

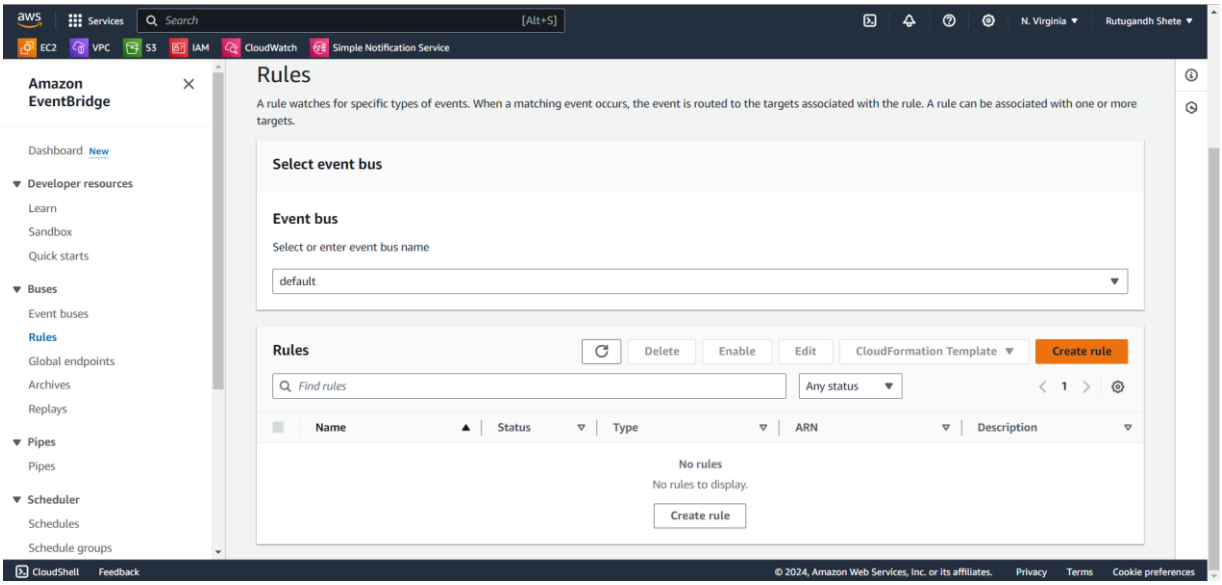


Events:

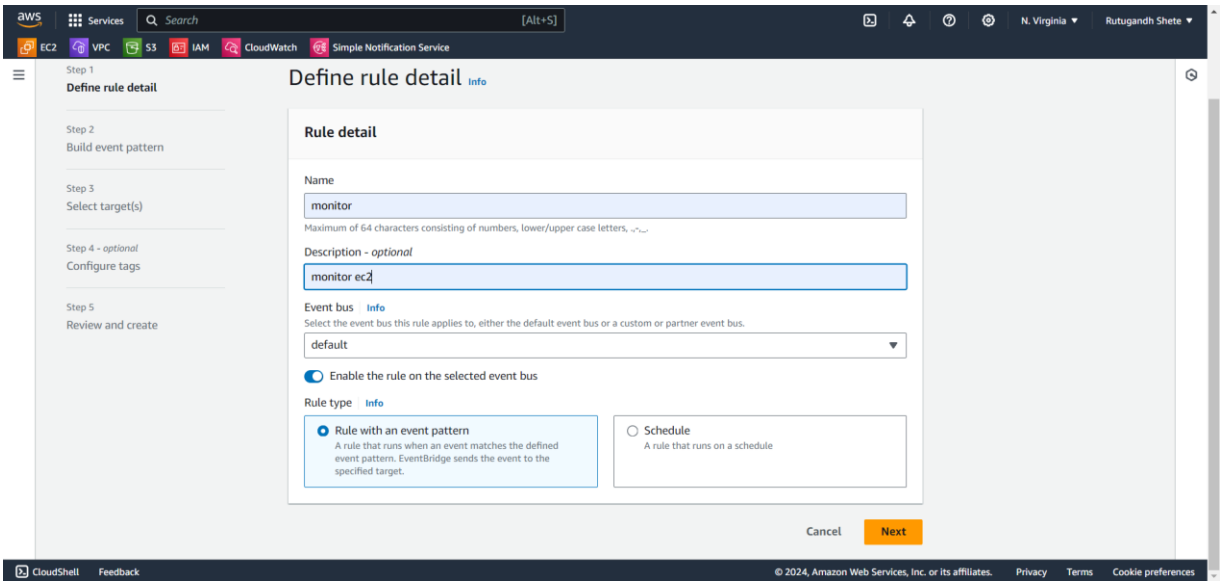
In AWS CloudWatch, **events** refer to actions or changes in your AWS environment that CloudWatch can detect and respond to. CloudWatch Events (now known as Amazon EventBridge) enables you to track these events and automatically trigger responses, such as invoking AWS Lambda functions, starting an EC2 instance, or sending notifications.

Steps:

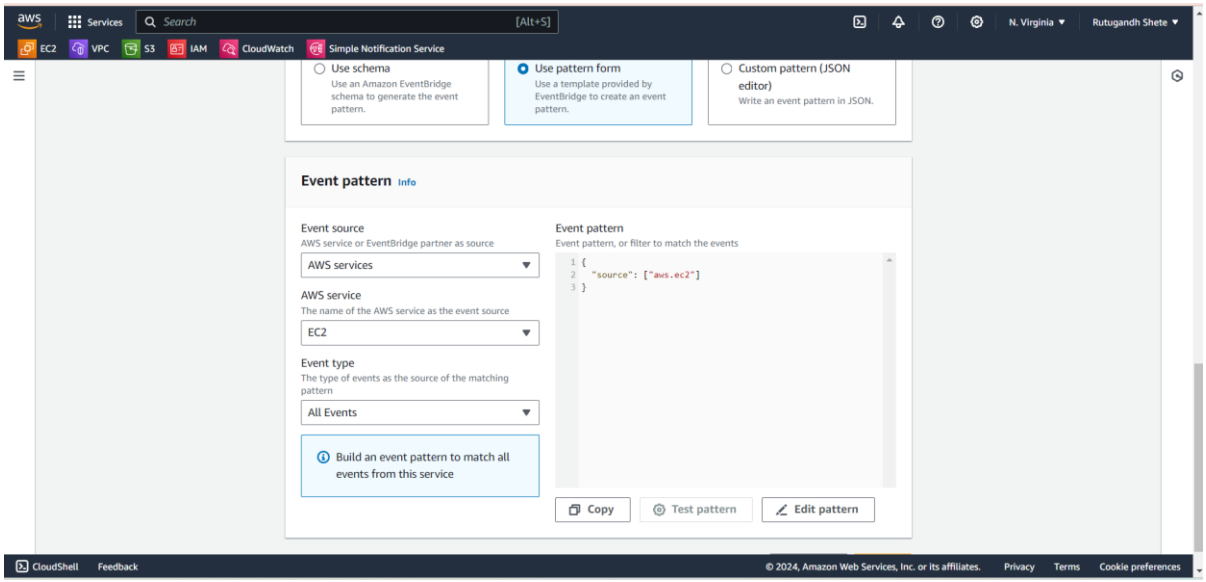
- Go to Events→Create rule



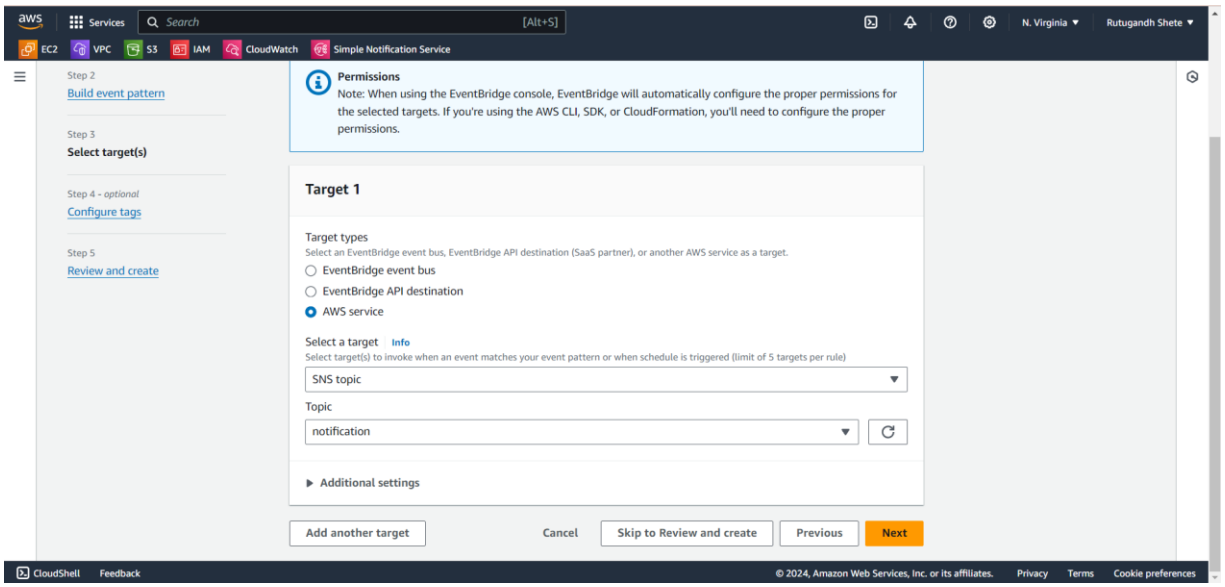
- Define rule details.



- In Event pattern→select event source→select Aws service→event type.



- Add target 1→SNS service→topic that we have created.



- Event will get triggered when we perform certain actions on EC2→we will get notified with SNS service through Mail. Terminate Instance.