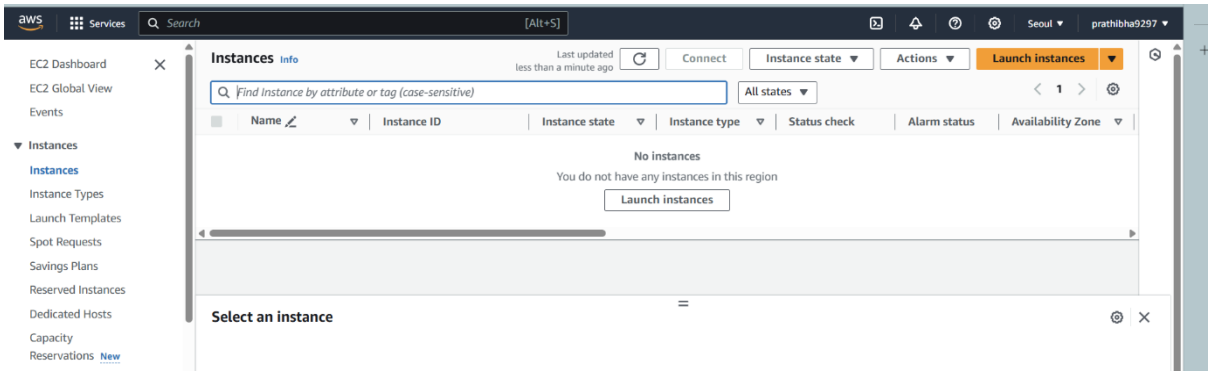
Simple Notification service (SNS)

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.

Step1: Create an instance

- Login to the AWS console
- Open seoul region and click on instances
- Click on launch instance
- Given name tag as instance-a
- Select amazon Linux in application and OS image field
- Select the key pair and launch instance as shown in below slides



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

instance-a

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Architecture	Boot mode	AMI ID	
64-bit (x86)	uefi-preferred	ami-008d41dbe16db6778	Verified provider

Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0288 USD per Hour
On-Demand Linux base pricing: 0.0144 USD per Hour
On-Demand SUSE base pricing: 0.0144 USD per Hour
On-Demand Windows base pricing: 0.019 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

seoul

Create new key pair

Network settings Info

Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-008d41dbe16db6778

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB

Cancel | Launch instance

Review commands

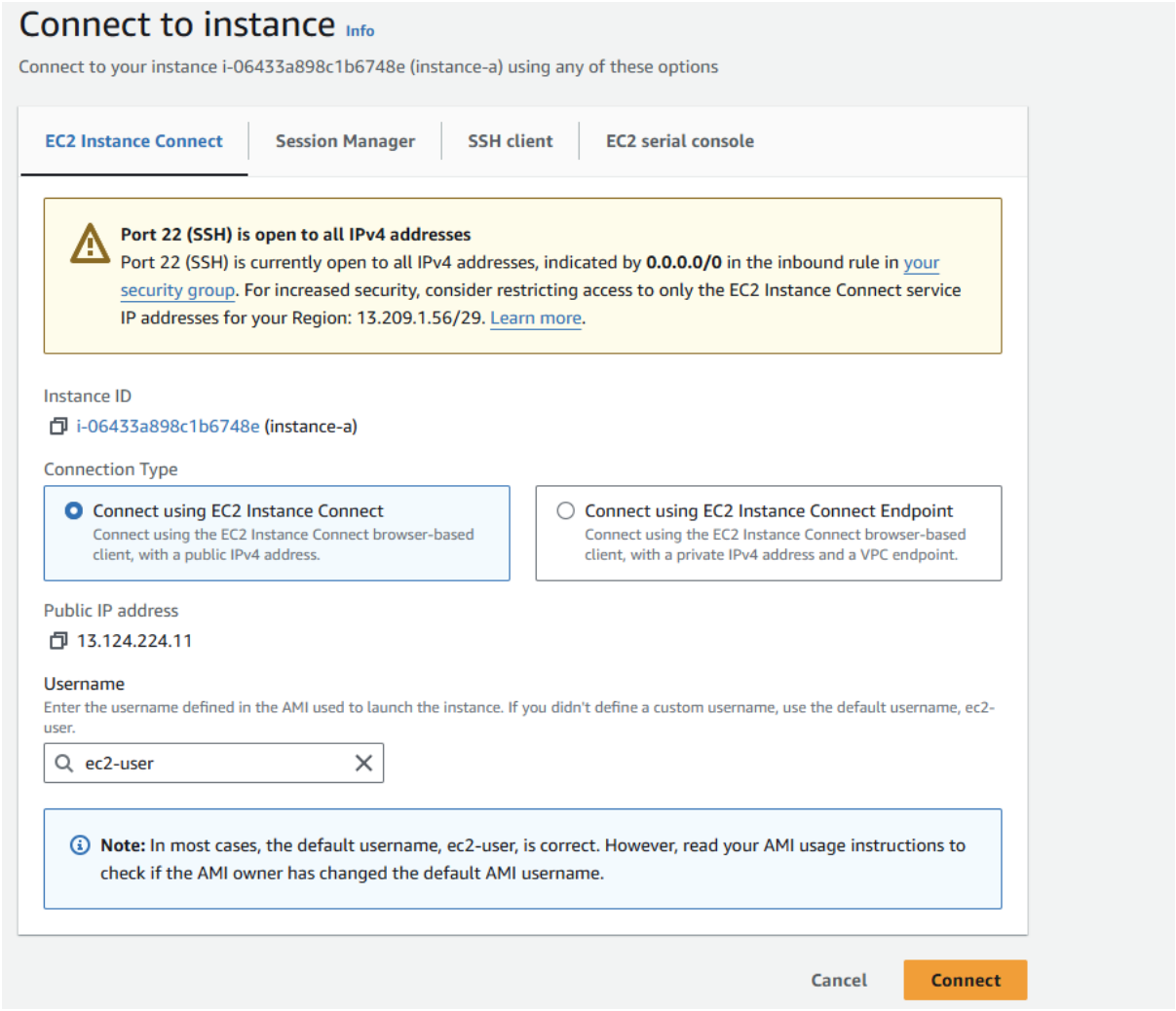
Instances (1) Info

Last updated less than a minute ago | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
instance-a	i-06433a898c1b6748e	Running	t2.micro	Initializing	View alarms	ap-northeast-2a	ec2-13-124-224-11.ap-

- Once instance state change to running click on connect and then you will get connection to the amazon Linux server.
- Change to root user using sudo -I
- For updating the server yum update -y
- For installing the stress =>yum install stress -y



Step2: Creating Simple Notification Service (SNS) topic

- Open SNS and click one create topic
- Type should be standard
- Give name tag as sns-a
- And then click on create topic

aws Services Search [Alt+S] Seoul

New Feature
Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

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.

MyTopic

Next step

Start with an overview

Pricing

Amazon SNS has no upfront costs. You pay based on the number of messages you publish, the number of messages you deliver, and any additional API calls for managing topics and

Benefits and features

Reliably deliver messages with | Automatically scale your

Amazon SNS > Topics > Create topic

Create topic

Details

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

FIFO (first-in, first-out)

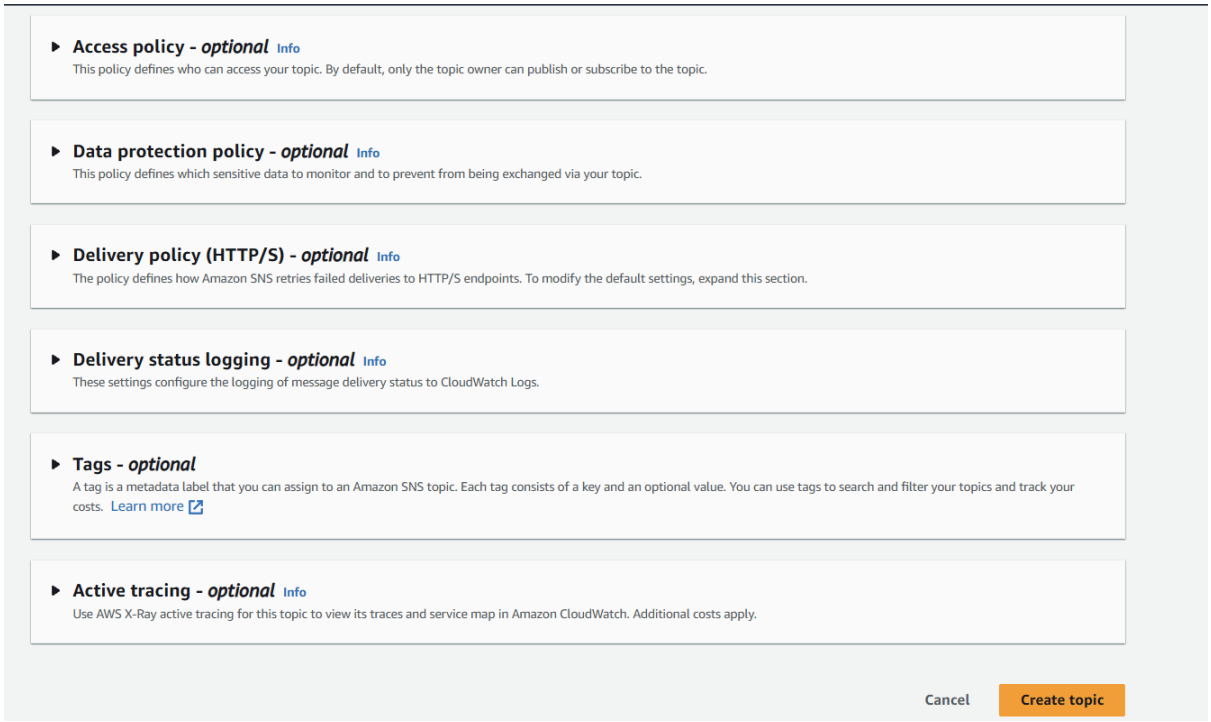
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

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

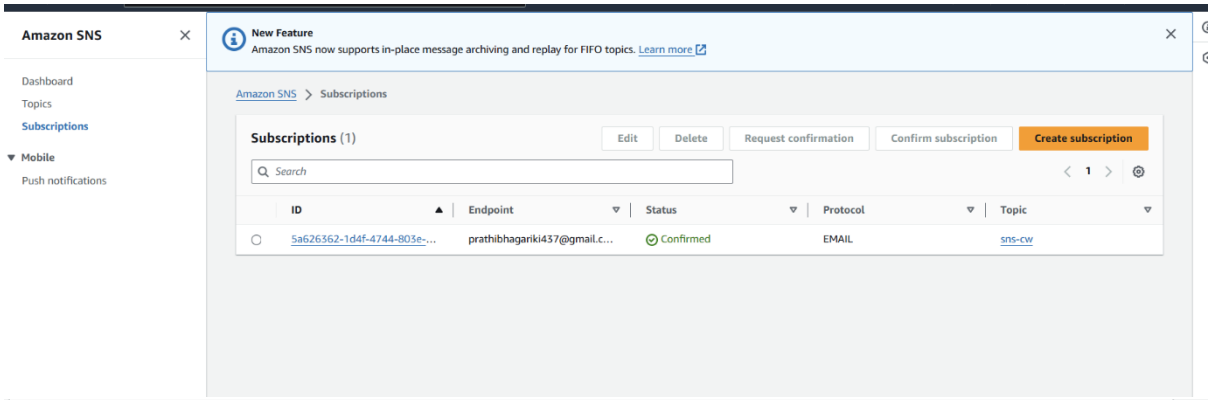
Name

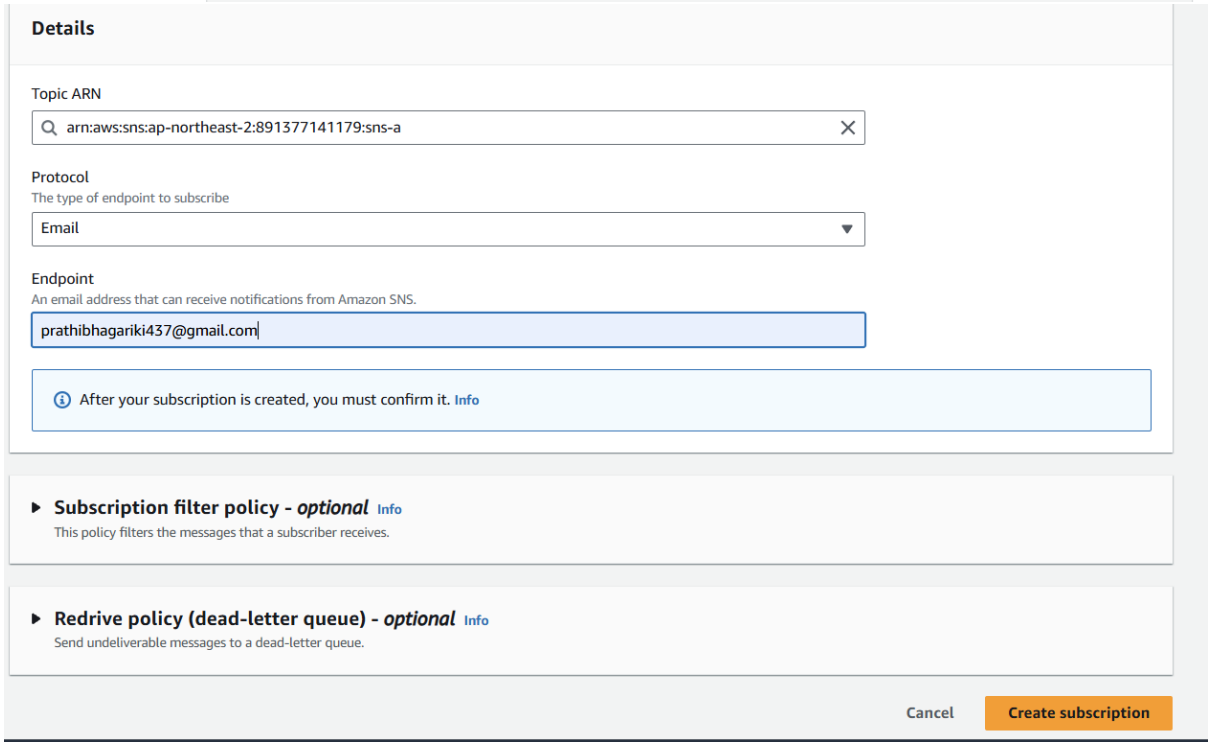
sns-a

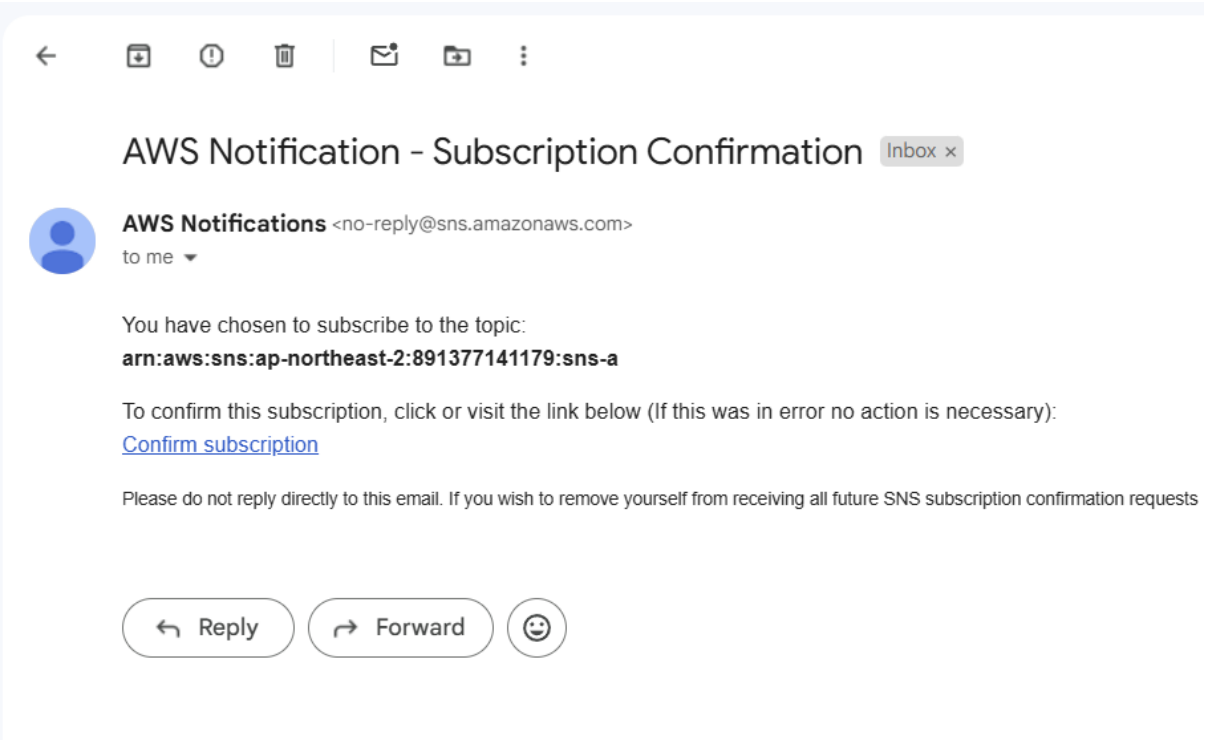


Step3: Subscription

- Open subscriptions in Amazon SNS
- Click on create subscription
- Select SNS topic ARN
- Protocol should be Email
- Enter the email address in the Endpoint field
- Click on create subscription
- Once created the subscription confirmation mail will come to the particular mail address u mentioned in subscription
- Click on confirm subscription in email
- The following slides will showcase the subscription process

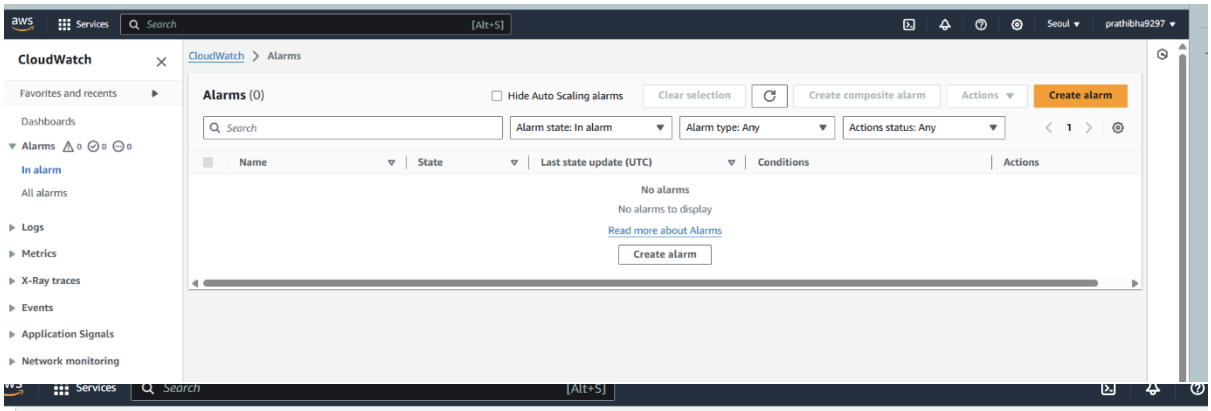

Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
New Feature
Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more
Amazon SNS > Subscriptions
Subscriptions (1)
Edit
Delete
Request confirmation
Confirm subscription
Create subscription
Search
ID
Endpoint
Status
Protocol
Topic
5a626362-1d4f-4744-803e-...
prathibhagariki437@gmail.c...
Confirmed
EMAIL
sns-cw

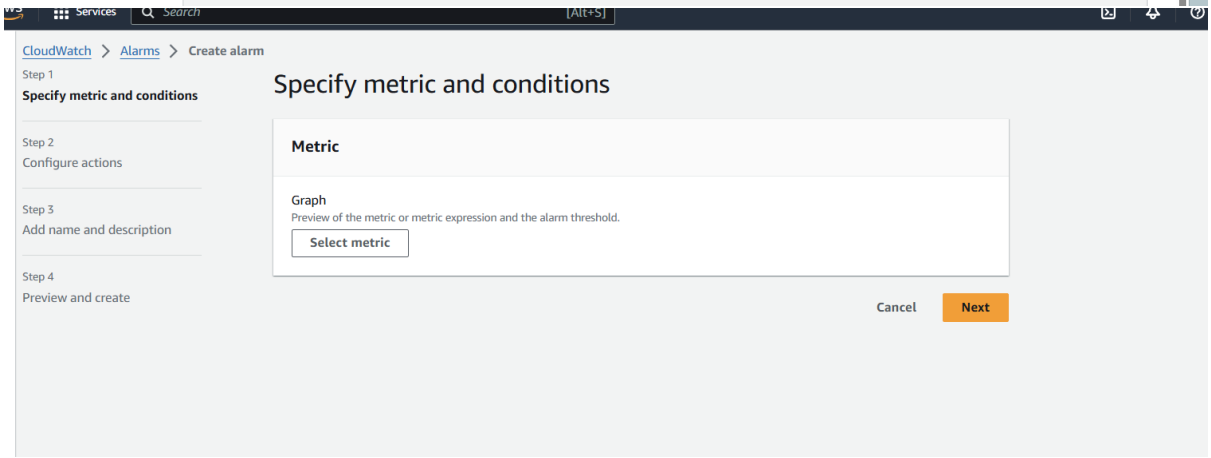

Details
Topic ARN
arn:aws:sns:ap-northeast-2:891377141179:sns-a
Protocol
The type of endpoint to subscribe
Email
Endpoint
An email address that can receive notifications from Amazon SNS.
prathibhagariki437@gmail.com
After your subscription is created, you must confirm it. Info
Subscription filter policy - optional Info
This policy filters the messages that a subscriber receives.
Redrive policy (dead-letter queue) - optional Info
Send undeliverable messages to a dead-letter queue.
Cancel
Create subscription

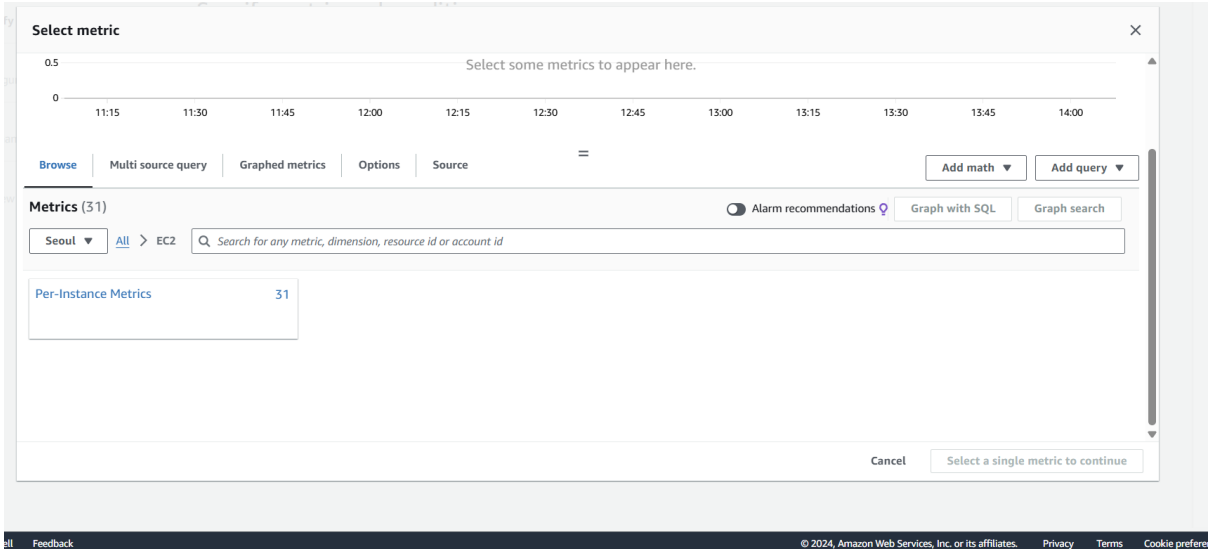


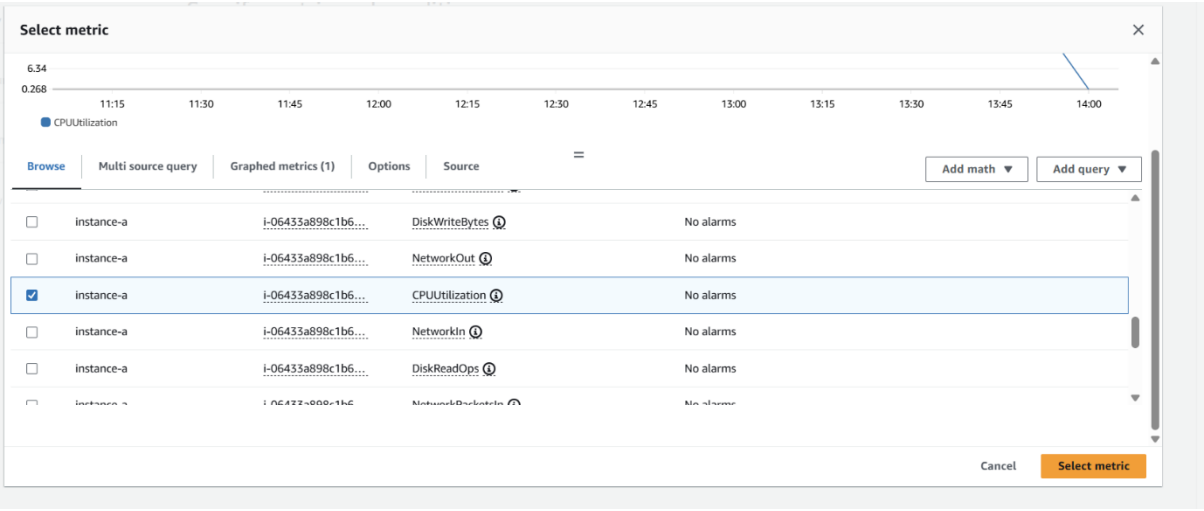
Step4: Creating alarm in CloudWatch

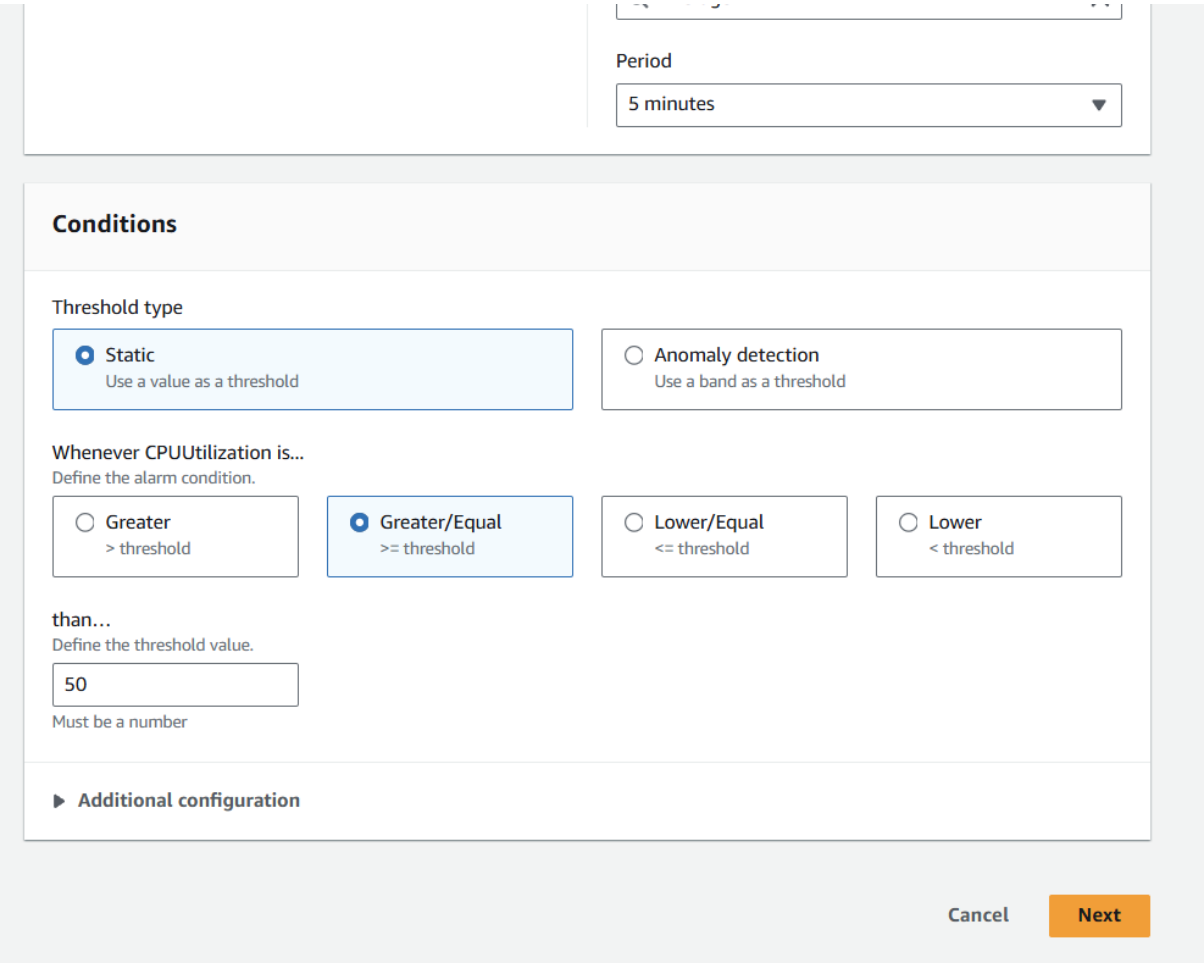
- Open the CloudWatch service and go to in alarm
- Click on create alarm
- Click on select metric
- Click on Select per instance metrics and select the CPU utilization for instance-a and click on select metric
- In conditions field select threshold type should be static and threshold value should be greater than or equal to 50
- Give name tag as sns-a-notification and click on create alarm as shown in below slides


aws
Services
Search
[Alt+S]
Seoul
prathibha9297
CloudWatch
Favorites and recents
Dashboards
Alarms
In alarm
All alarms
Logs
Metrics
X-Ray traces
Events
Application Signals
Network monitoring
CloudWatch > Alarms
Alarms (0)
Hide Auto Scaling alarms
Clear selection
Create composite alarm
Actions
Create alarm
Search
Alarm state: In alarm
Alarm type: Any
Actions status: Any
Name
State
Last state update (UTC)
Conditions
Actions
No alarms
No alarms to display
Read more about Alarms
Create alarm


Services
Search
[Alt+S]
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
Specify metric and conditions
Metric
Graph
Preview of the metric or metric expression and the alarm threshold.
Select metric
Cancel
Next


Select metric
0.5
0
Select some metrics to appear here.
11:15
11:30
11:45
12:00
12:15
12:30
12:45
13:00
13:15
13:30
13:45
14:00
Browse
Multi source query
Graphed metrics
Options
Source
Add math
Add query
Metrics (31)
Alarm recommendations
Graph with SQL
Graph search
Seoul
All > EC2
Search for any metric, dimension, resource id or account id
Per-Instance Metrics
31
Cancel
Select a single metric to continue
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie prefere


Select metric
6.34
0.268
11:15
11:30
11:45
12:00
12:15
12:30
12:45
13:00
13:15
13:30
13:45
14:00
CPUUtilization
Browse
Multi source query
Graphed metrics (1)
Options
Source
Add math
Add query
instance-a
i-06433a898c1b6...
DiskWriteBytes
No alarms
instance-a
i-06433a898c1b6...
NetworkOut
No alarms
instance-a
i-06433a898c1b6...
CPUUtilization
No alarms
instance-a
i-06433a898c1b6...
NetworkIn
No alarms
instance-a
i-06433a898c1b6...
DiskReadOps
No alarms
Cancel
Select metric


Period
5 minutes
Conditions
Threshold type
Static
Use a value as a threshold
Anomaly detection
Use a band as a threshold
Whenever CPUUtilization is...
Define the alarm condition.
Greater
> threshold
Greater/Equal
>= threshold
Lower/Equal
<= threshold
Lower
< threshold
than...
Define the threshold value.
50
Must be a number
Additional configuration
Cancel
Next

alarm

Add name and description

Name and description

Alarm name

sns-a-notification

Alarm description - *optional* View formatting guidelines

Edit | Preview

 # This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

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

▶ Additional configuration

Step 2: Configure actions

Edit

Actions

Notification
When In alarm, send a notification to "sns-a"

Step 3: Add name and description

Edit

Name and description

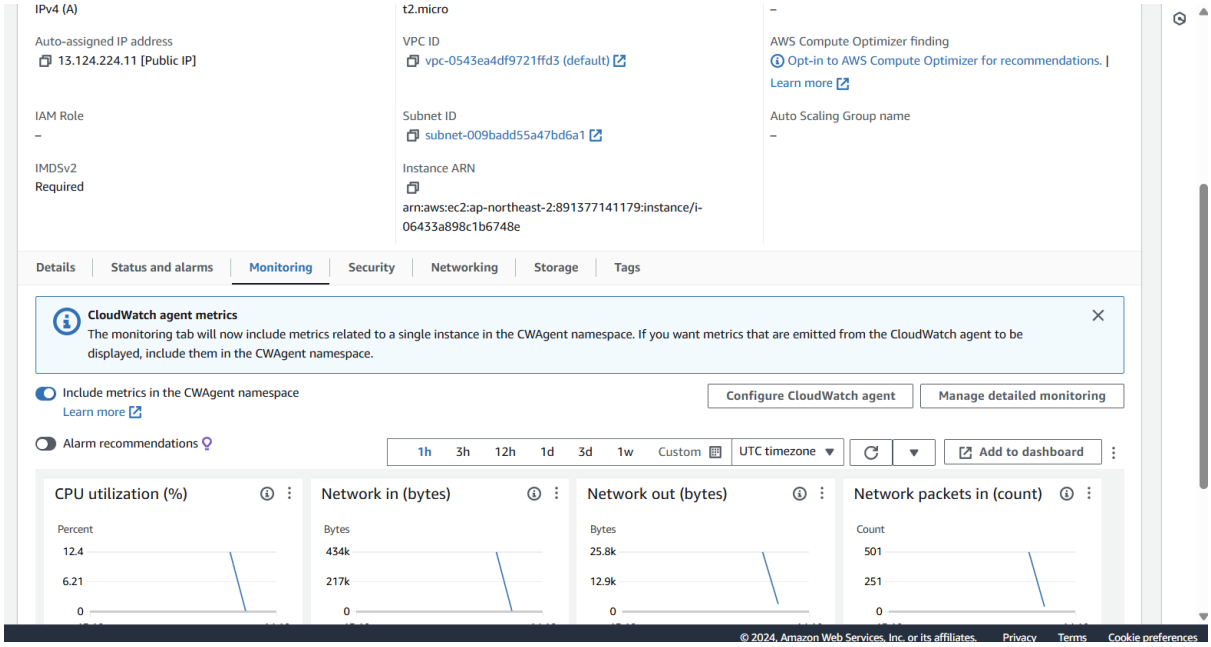
Name
sns-a-notification

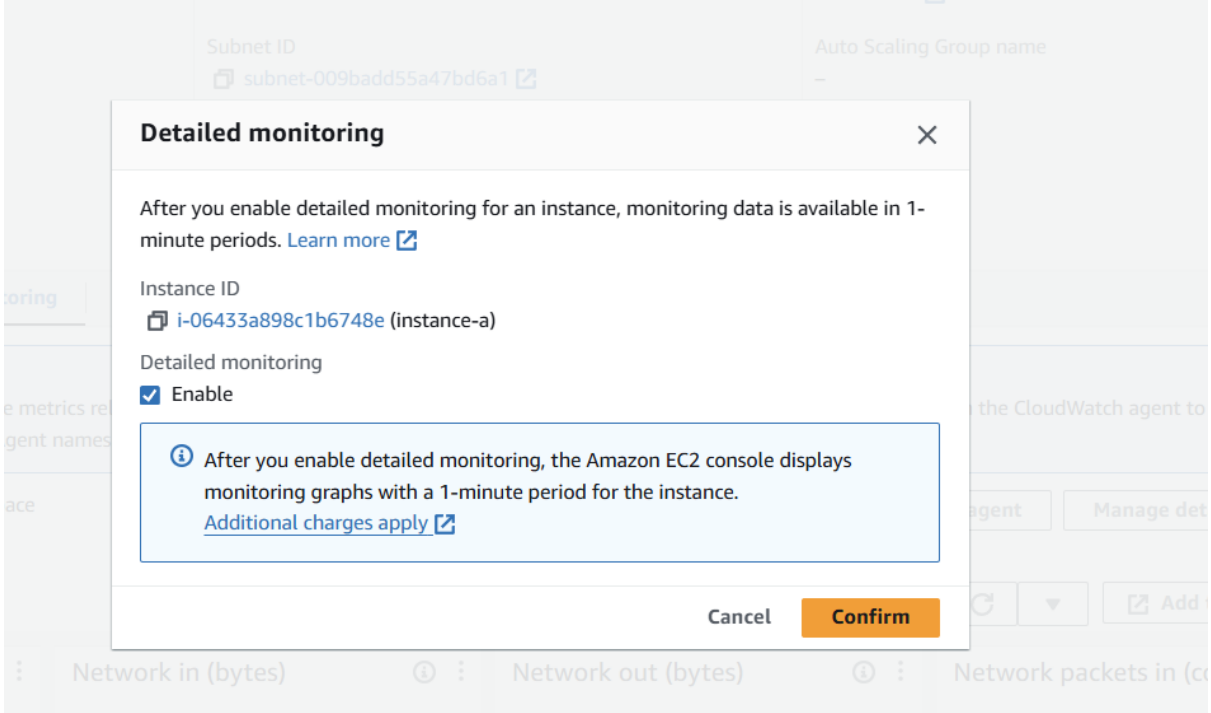
Description
-

Cancel | Previous | Create alarm

Step5: Enable the detailed monitoring for instance-a

- Once alarm cleared go to instance-a and click on monitoring
- Click on managed detailed monitoring and click on enable and then hit on confirm





- Go to AWS Linux server and give stress command => stress –cpu 10 -v –timeout 400s
- Once given stress to the instance-a the CPU utilization should be more than 50 percentage
- Then we will get notification to the email as shown in below slides.

aws Services Search [Alt+S]

```
[root@ip-172-31-0-10 ~]# stress --cpu 10 -v --timeout 400s
stress: info: [26179] dispatching hogs: 10 cpu, 0 io, 0 vm, 0 hdd
stress: dbug: [26179] using backoff sleep of 30000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 10 [26180] forked
stress: dbug: [26179] using backoff sleep of 27000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 9 [26181] forked
stress: dbug: [26179] using backoff sleep of 24000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 8 [26182] forked
stress: dbug: [26179] using backoff sleep of 21000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 7 [26183] forked
stress: dbug: [26179] using backoff sleep of 18000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 6 [26184] forked
stress: dbug: [26179] using backoff sleep of 15000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 5 [26185] forked
stress: dbug: [26179] using backoff sleep of 12000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 4 [26186] forked
stress: dbug: [26179] using backoff sleep of 9000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 3 [26187] forked
stress: dbug: [26179] using backoff sleep of 6000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 2 [26188] forked
stress: dbug: [26179] using backoff sleep of 3000us
stress: dbug: [26179] setting timeout to 400s
stress: dbug: [26179] --> hogcpu worker 1 [26189] forked
```

i-06433a898c1b6748e (instance-a)

PublicIPs: 13.124.224.11 PrivateIPs: 172.31.0.10

Output:

Search mail

ALARM: "sns-a-notification" in Asia Pacific (Seoul) Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 19:45 (0 minutes ago)
to me

You are receiving this email because your Amazon CloudWatch Alarm "sns-a-notification" in the Asia Pacific (Seoul) region has entered the ALARM state, becau
1 out of the last 1 datapoints [50.84180790960456 (21/08/24 14:10:00)] was greater than or equal to the threshold (50.0) (minimum 1 datapoint for OK -> ALARM
"Wednesday 21 August, 2024 14:15:38 UTC".

View this alarm in the AWS Management Console:
https://ap-northeast-2.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-northeast-2#alarmsV2:alarm/sns-a-notification

Alarm Details:
- Name: sns-a-notification
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [50.84180790960456 (21/08/24 14:10:00)] was greater than or equal to the thres
datapoint for OK -> ALARM transition).
- Timestamp: Wednesday 21 August, 2024 14:15:38 UTC
- AWS Account: 891377141179
- Alarm Arn: arn:aws:cloudwatch:ap-northeast-2:891377141179:alarm:sns-a-notification

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 50.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [InstanceId = i-06433a898c1b6748e]