To monitor your ECS cluster using CloudWatch, we can use CloudWatch Container Insights. This will collect metrics and logs from our ECS cluster.

Here are some CloudWatch metrics that you can use to monitor your ECS cluster:

* CPUUtilization
* MemoryUtilization
* NetworkIn
* NetworkOut
* RunningCount
* PendingCount
* StoppedCount
* DesiredCount

But we need to get alert when service goes down and what is the reason of its down so we need to find out common scenarios when the service goes down

ECS tasks might stop due to a variety of reasons. The most common reasons are:

* Essential container exited
* Failed Elastic Load Balancing (ELB) health checks
* Failed container health checks
* Unhealthy container instance
* Underlying infrastructure maintenance
* Service scaling event triggered
* **ResourceInitializationError**
* **CannotPullContainerError**
* Task stopped by user

Understanding the correlation between a stopped task and stopped reason can help reduce the effort needed to troubleshoot.

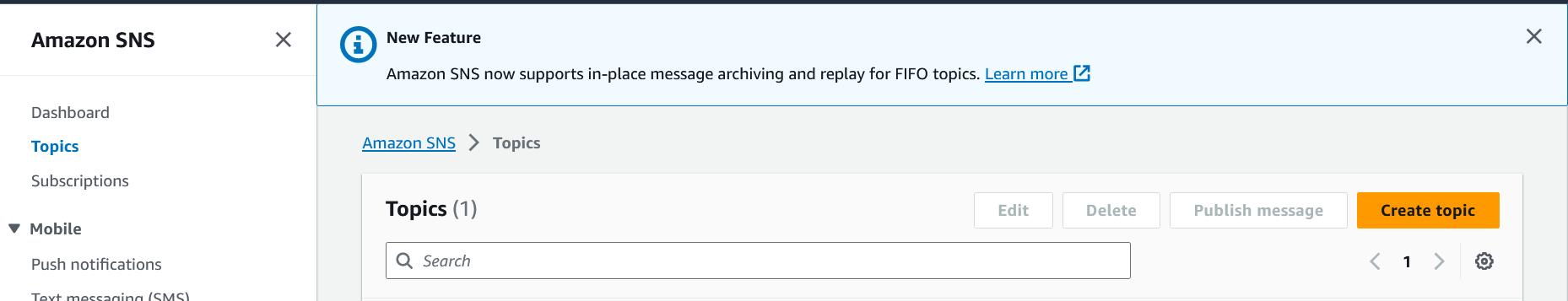
Amazon Elastic Container Service to set up email notifications for stopped tasks.

Assuming that we have a running ecs cluster

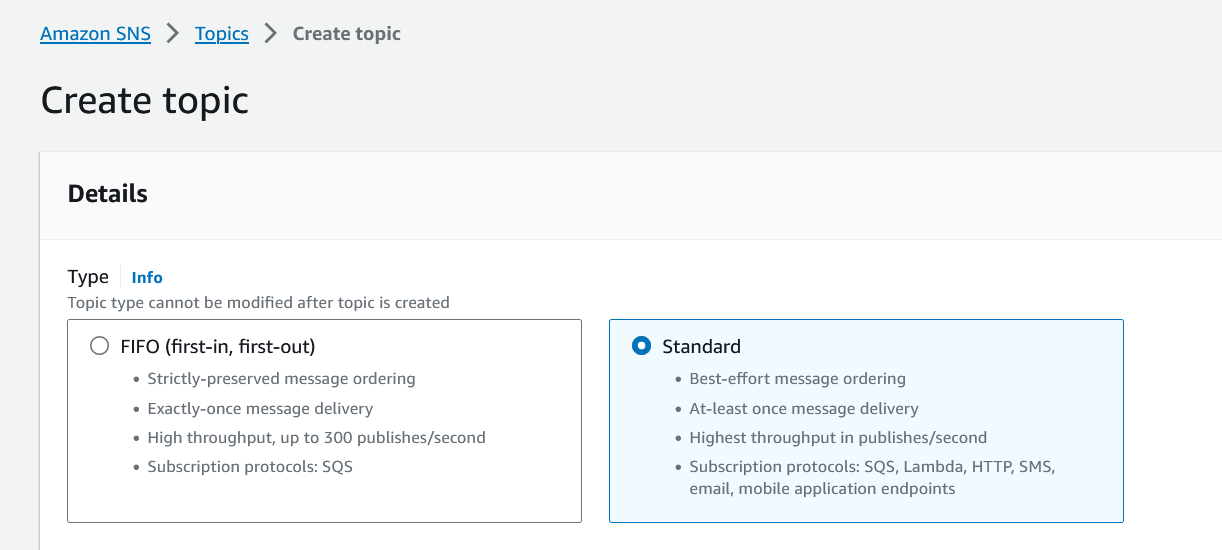
Step 1: Configure a SNS Topic.

In the AWS console, navigate to Simple Notification Service

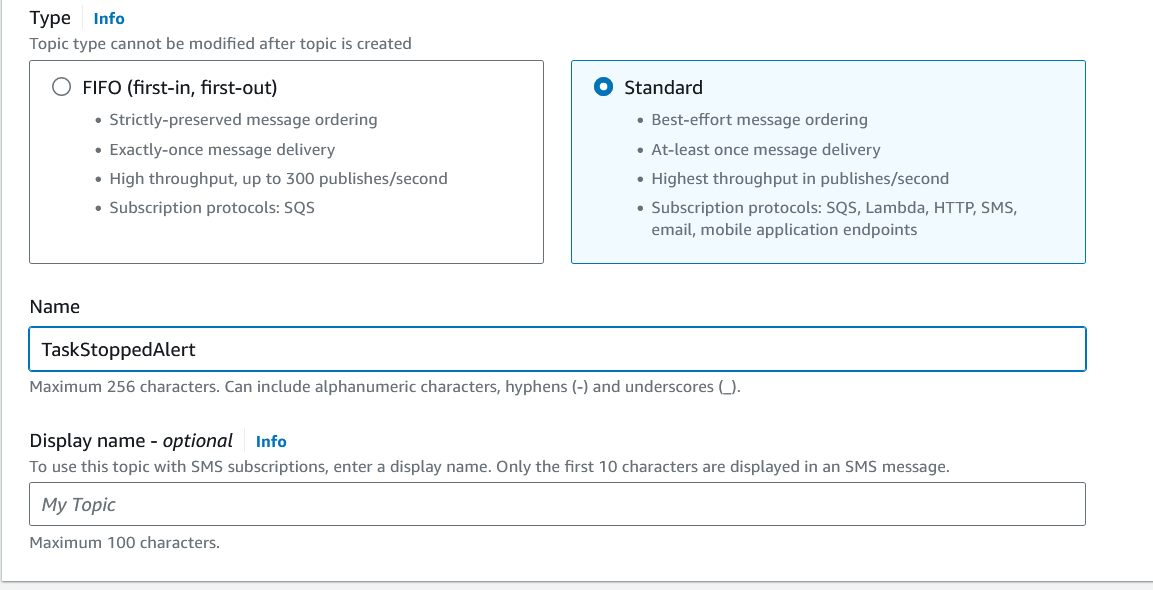
* Select create topic



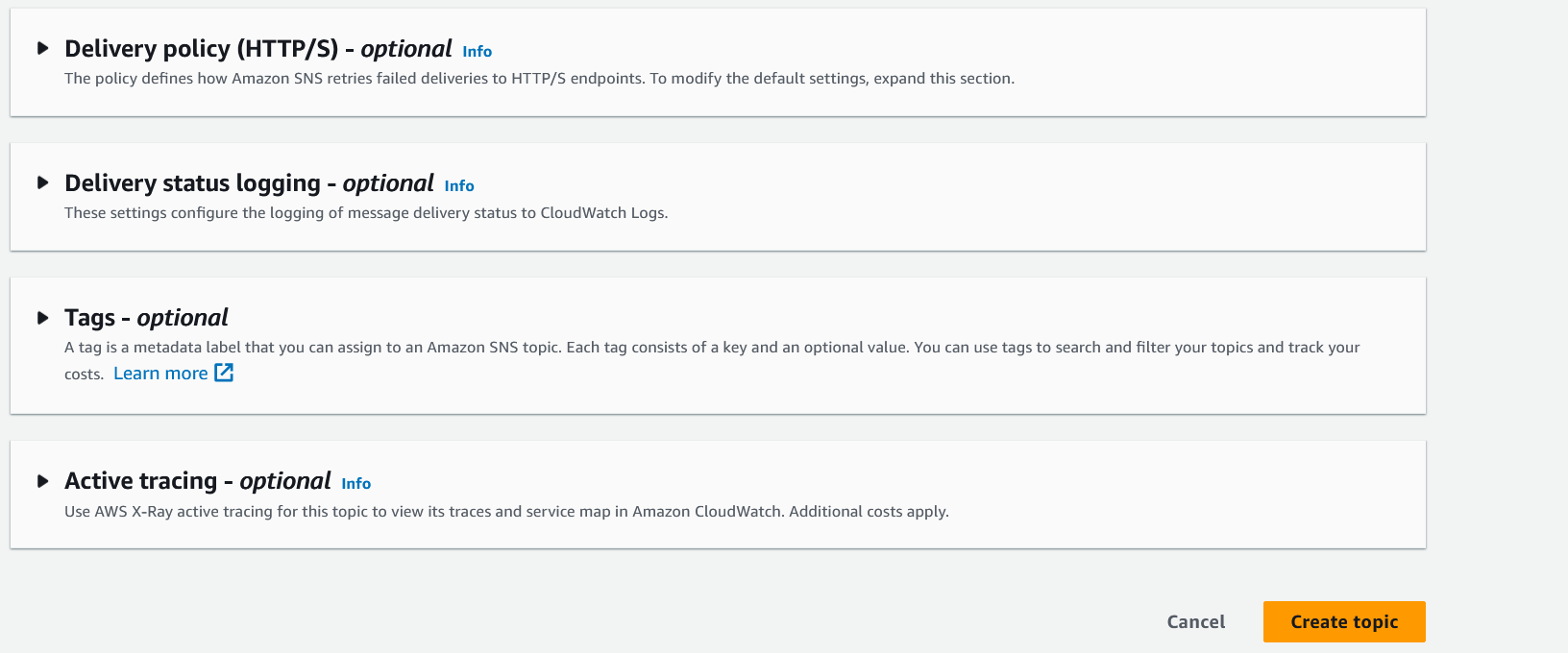
* For type, choose standard



* Name: (choose name. e.g. TaskStoppedAlert)

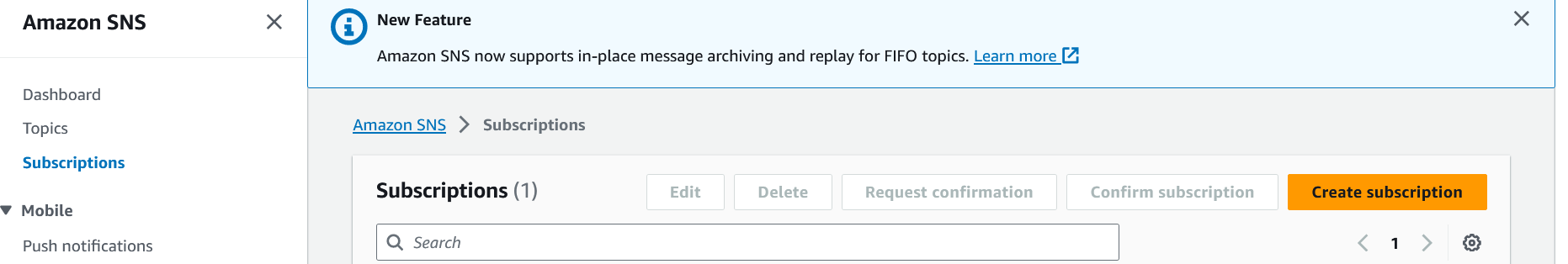


* Leave the other settings at default, scroll down, and click create topic

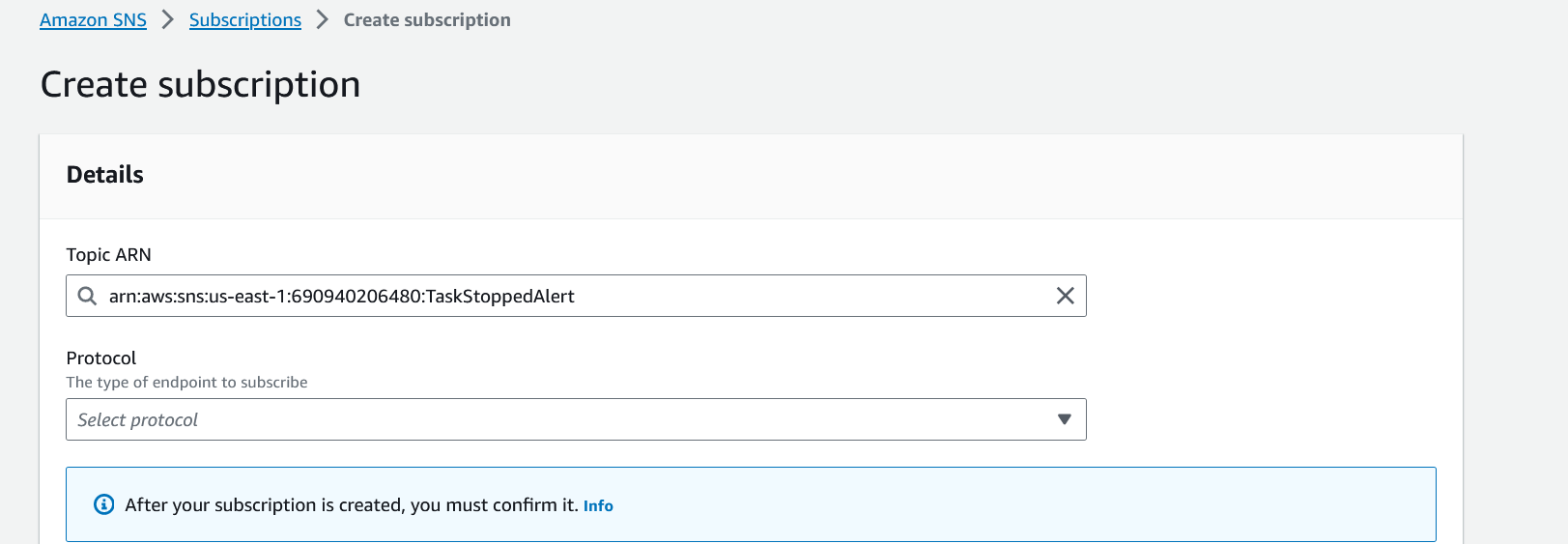


Step 2: Subscribe to the SNS topic you created.

* Select create subscription.



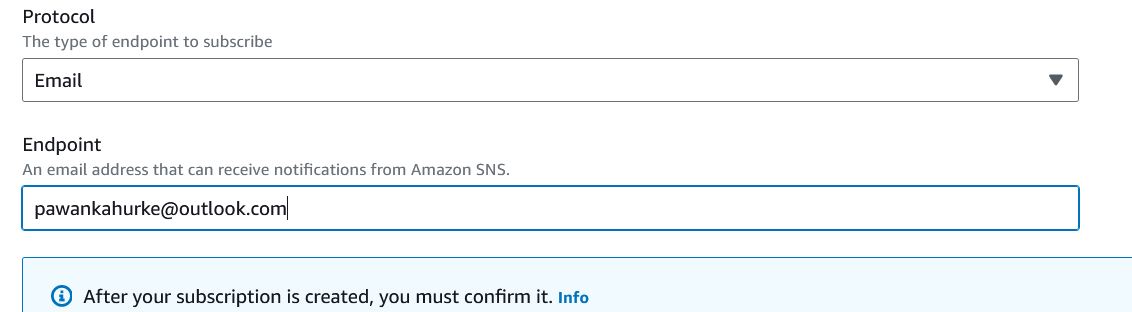
* Leave the topic ARN as default



* For protocol, select email



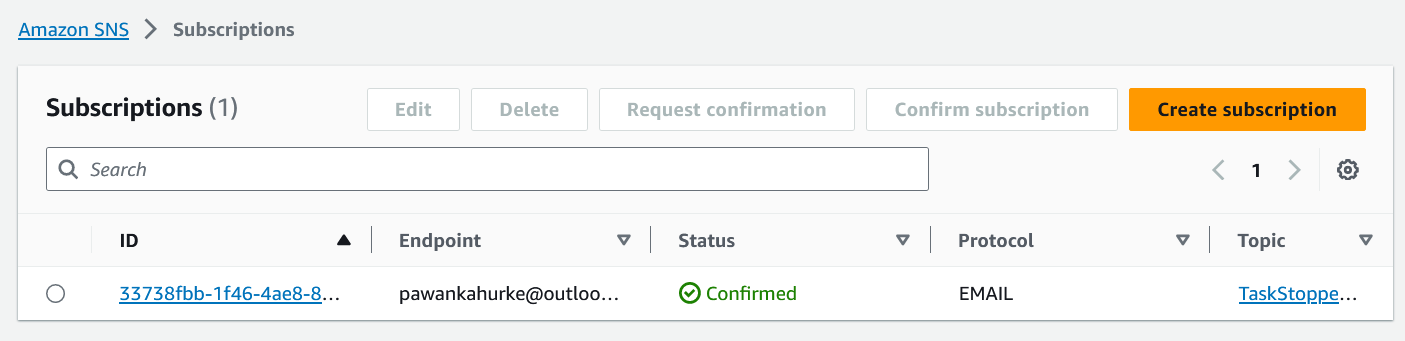
* For endpoint, enter a valid email address



* Click create subscription

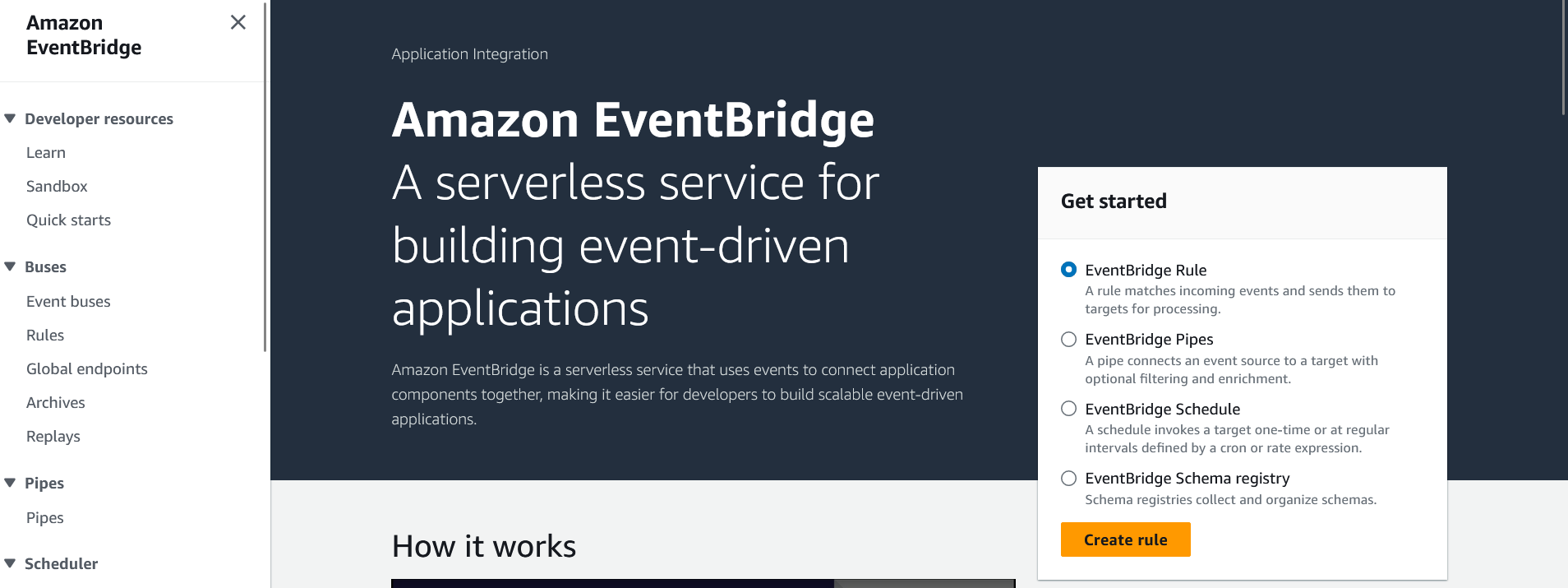
Step 3: Confirm the subscription.

* Open up the email that you entered, search for the AWS notifications email and click on the confirm subscription link provided in the email
* Verify that the subscription is confirmed

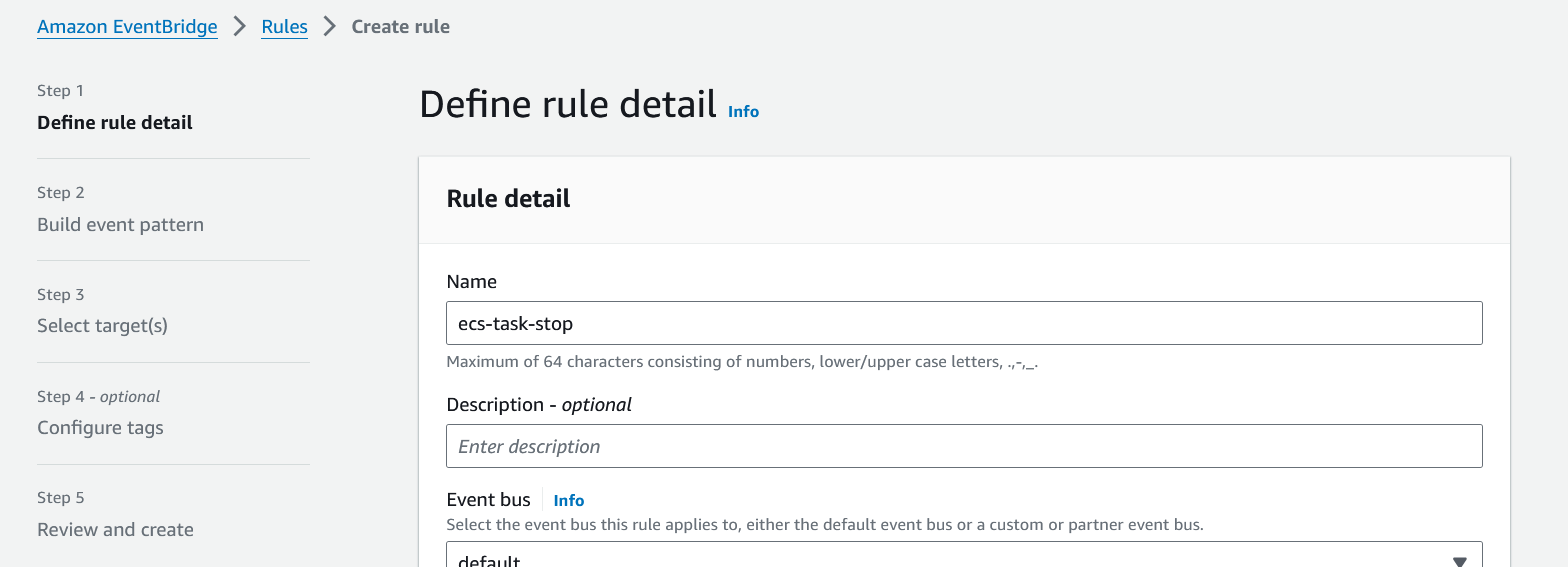


Step4: Create an Amazon EventBridge rule to trigger the SNS Topic when the state changes to stopped on an ECS Task

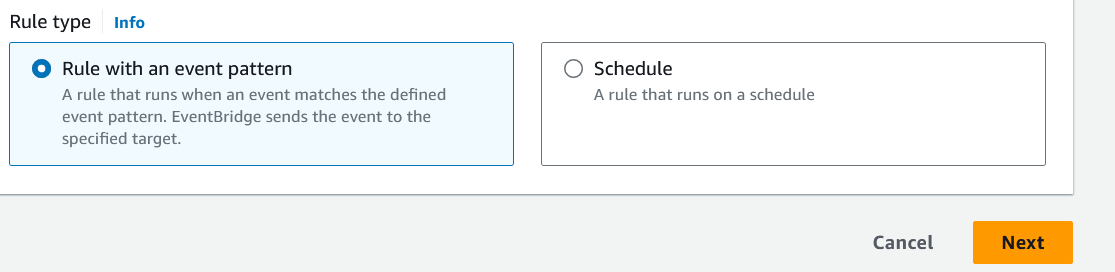
* Navigate to Amazon EventBridge in the AWS console



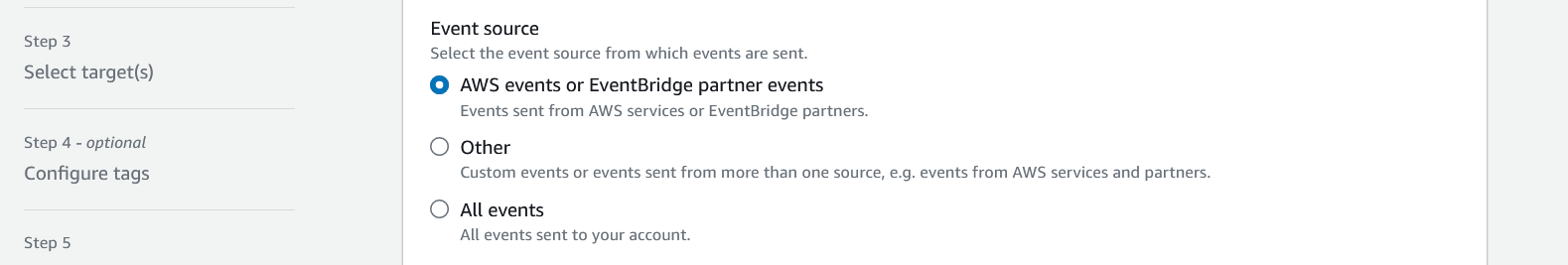
* Click create rule
* Name your rule (e.g. ecs-task-stop)



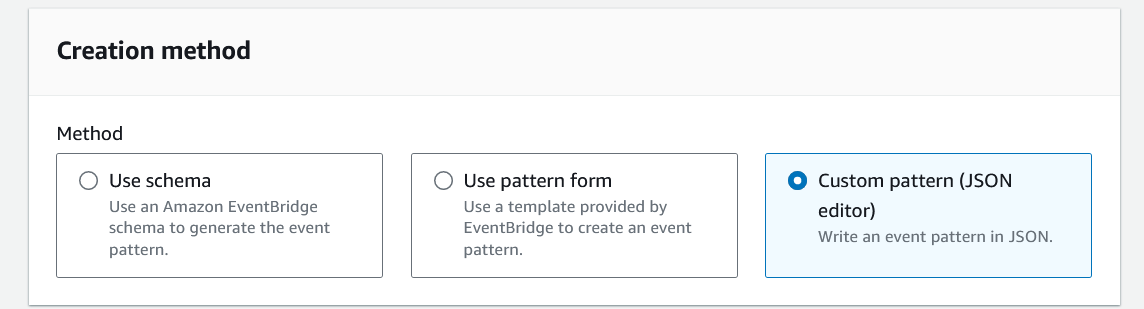
* For rule type, select rule with an event pattern and click next



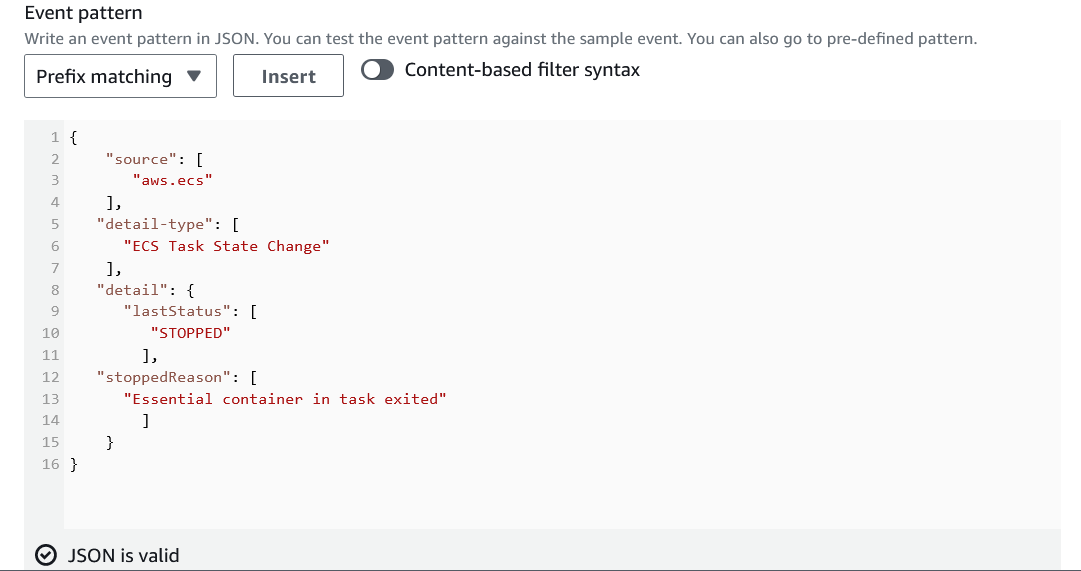
* For event source, choose AWS events or EventBridge partner events



* For creation method, choose custom pattern (JSON editor)



* Paste in the following code in the event pattern and click next



Here in stoppedReason we need to add all the task failed scenarios

{

"source": ["aws.ecs"],

"detail-type": ["ECS Task State Change"],

"detail": {

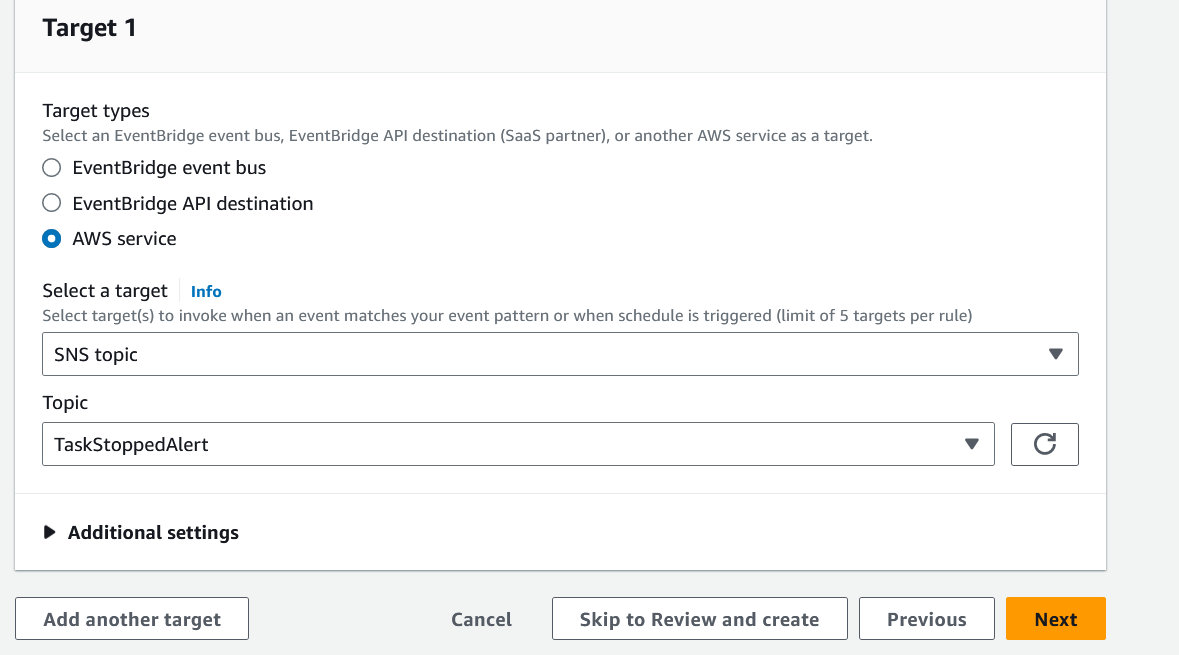
"lastStatus": ["STOPPED"],

"stoppedReason": ["Essential container in task exited", "Task stopped by user", "CannotPullContainerError", "Failed Elastic Load Balancing (ELB) health checks", "Failed container health checks", "Unhealthy container instance", "Underlying infrastructure maintenance", "Service scaling event triggered", "ResourceInitializationError"]

}

}

* For target types, select AWS service
* For target, select SNS Topic
* For topic, select the topic you created

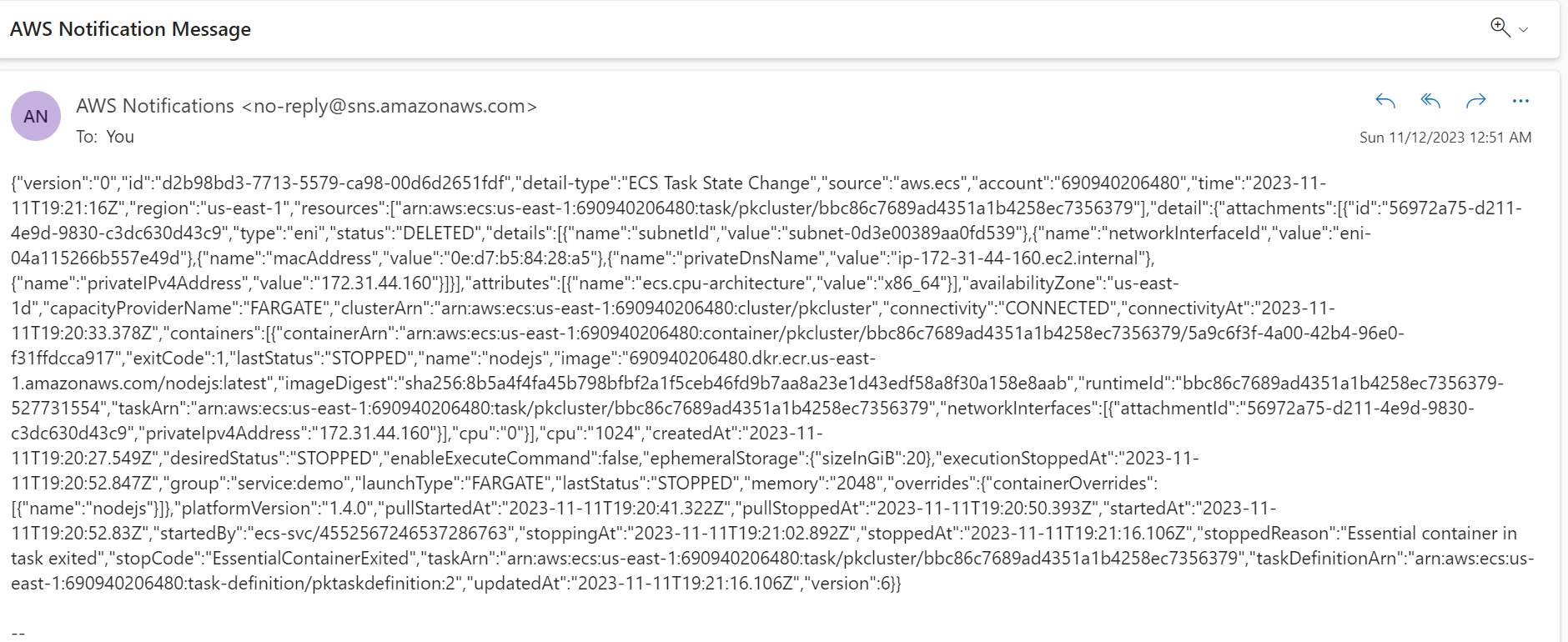


* Leave the rest default and select create rule

Step5: Test your rule

In this part I tested with multiple scenarios

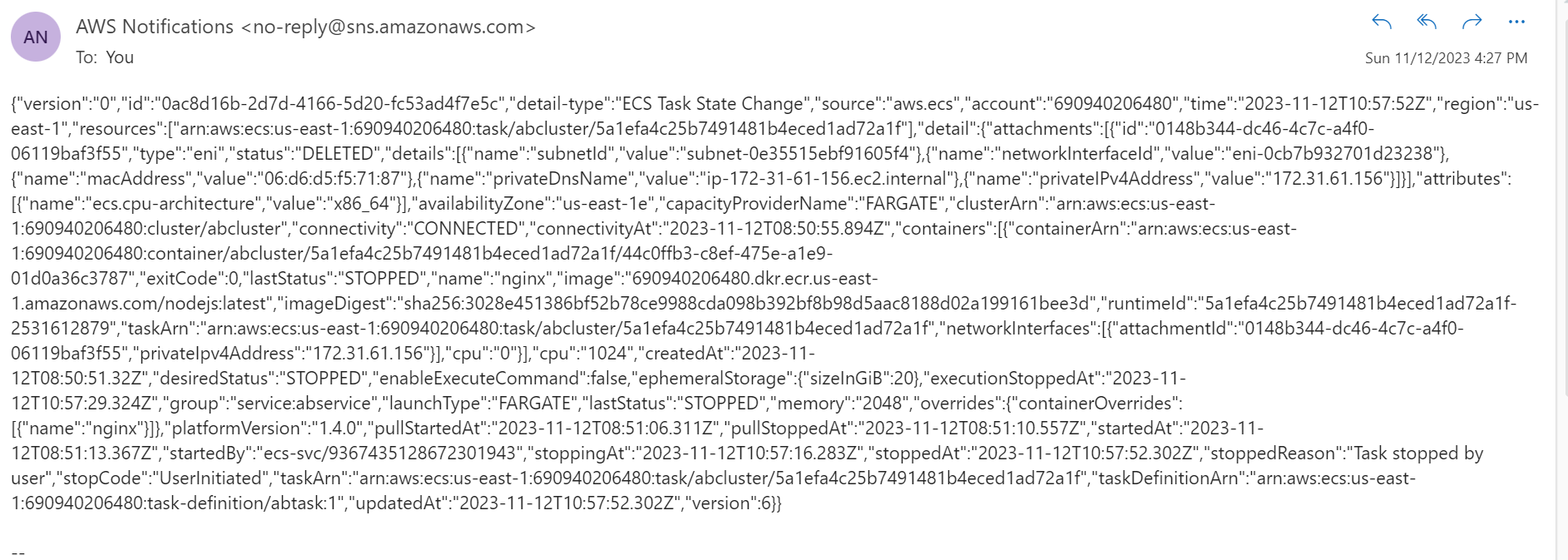
e.g I given nodejs image that doesn’t contains index.js file



In this “lastStatus":"STOPPED" "stoppedAt":"2023-11-11T19:21:16.106Z","stoppedReason":"Essential container in task exited","stopCode":"EssentialContainerExited", so we can identify the reason of stopped task

And take action on it

In second scenario I stopped the task manually in that case



"lastStatus":"STOPPED" “stoppedAt":"2023-11-12T10:57:52.302Z","stoppedReason":"Task stopped by user","stopCode":"UserInitiated",

So we can identify the cause of stopping the task and get alert when it happens.