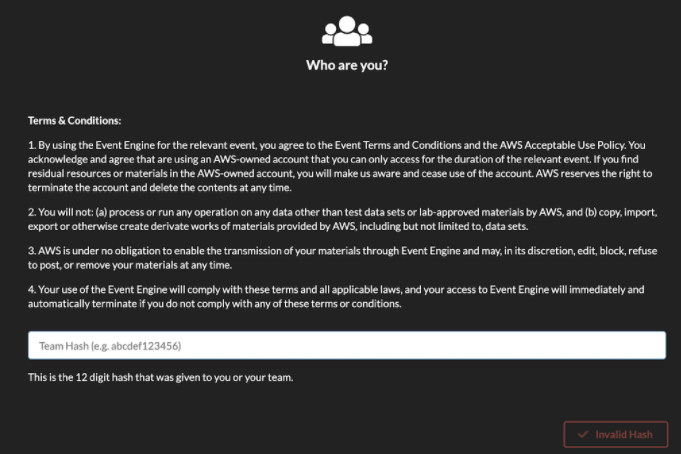
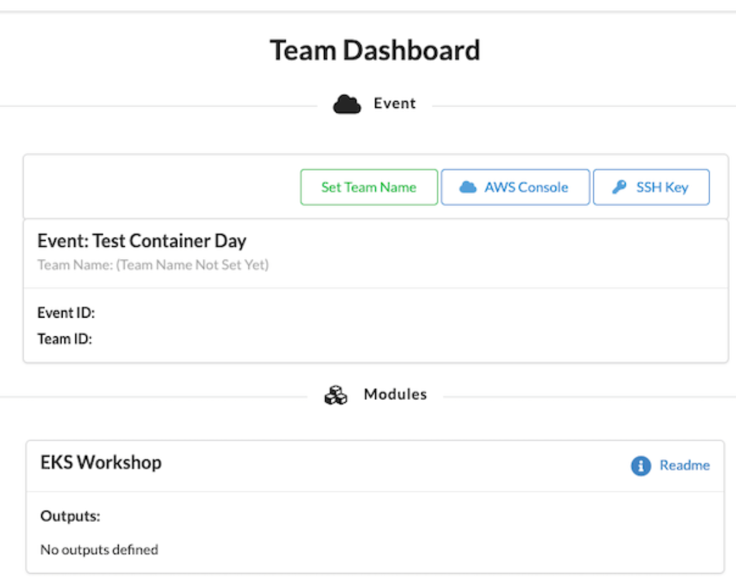
**1. Connecting to the AWS EventEngine**

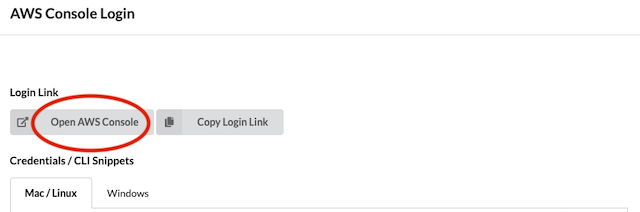
To help you get hands-on as quickly as possible, we pre-created an AWS environment for each participant. You will need the EventHash code, which will be shared by the speaker and your email address to track your unique session.

Connect to the AWS Event Engine by browsing to <https://dashboard.eventengine.run/>  
- Enter your unique hash and the click on proceed. 

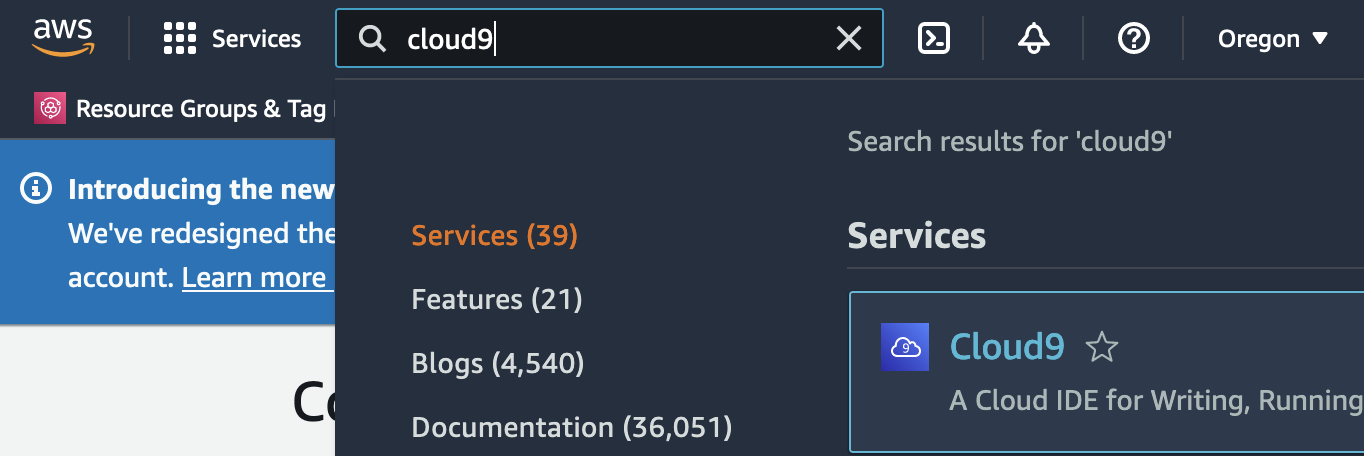
- You will see the below screen



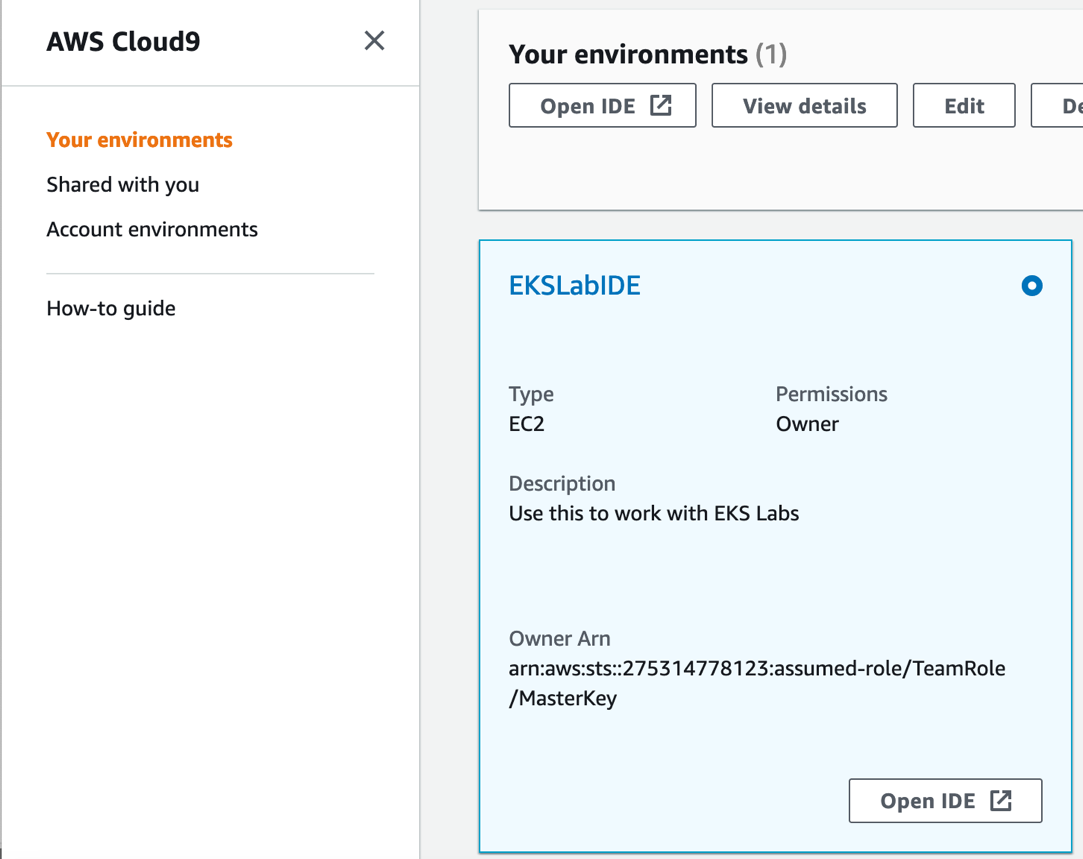
- Click on AWS Console button, and Open AWS Console button to launch the AWS Management Console. You don’t have to copy and credentials from here. If you’re previously logged in to any AWS account, you would need to first log out from there.



- You will already have an EKS cluster and Cloud9 environment. Search “Cloud9” in AWS console. Cloud9 is a managed IDE that we will use to run kubectl to interact with the EKS cluster later to simulate Kubernetes Findings.

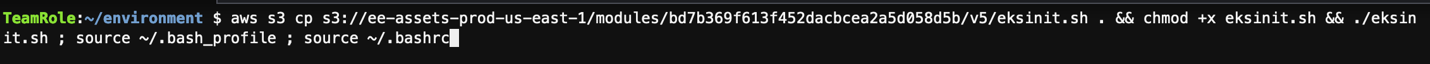


- You will be able to find EKSLabIDE cloud9 instance pre-deployed. Click Open IDE.

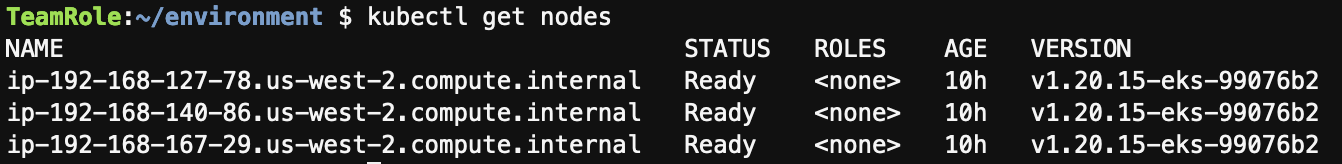


- Run a following command to install required tools.

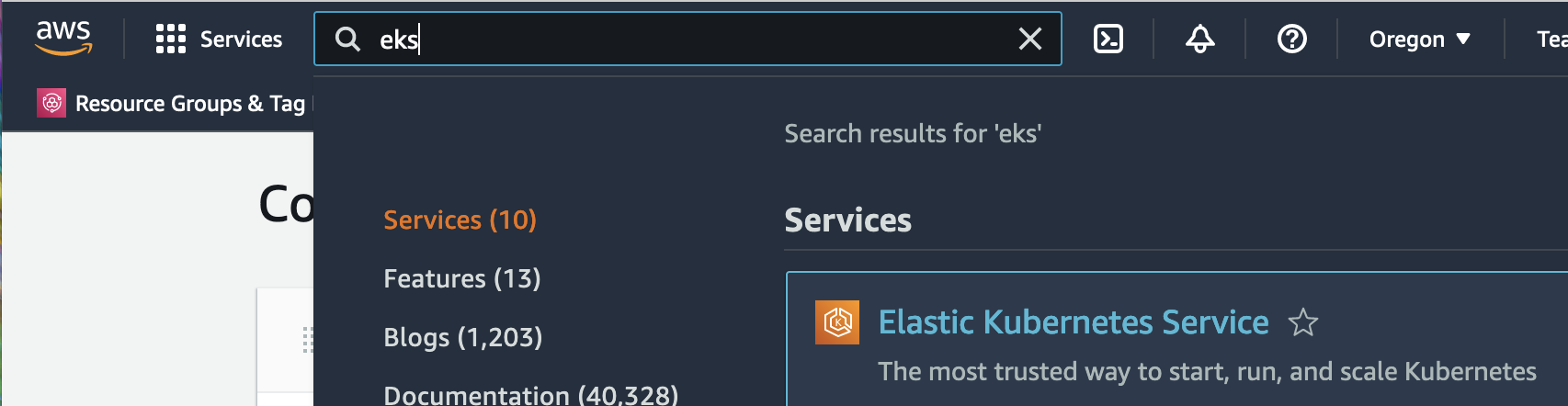
*aws s3 cp s3://ee-assets-prod-us-east-1/modules/bd7b369f613f452dacbcea2a5d058d5b/v5/eksinit.sh . && chmod +x eksinit.sh && ./eksinit.sh ; source ~/.bash\_profile ; source ~/.bashrc*

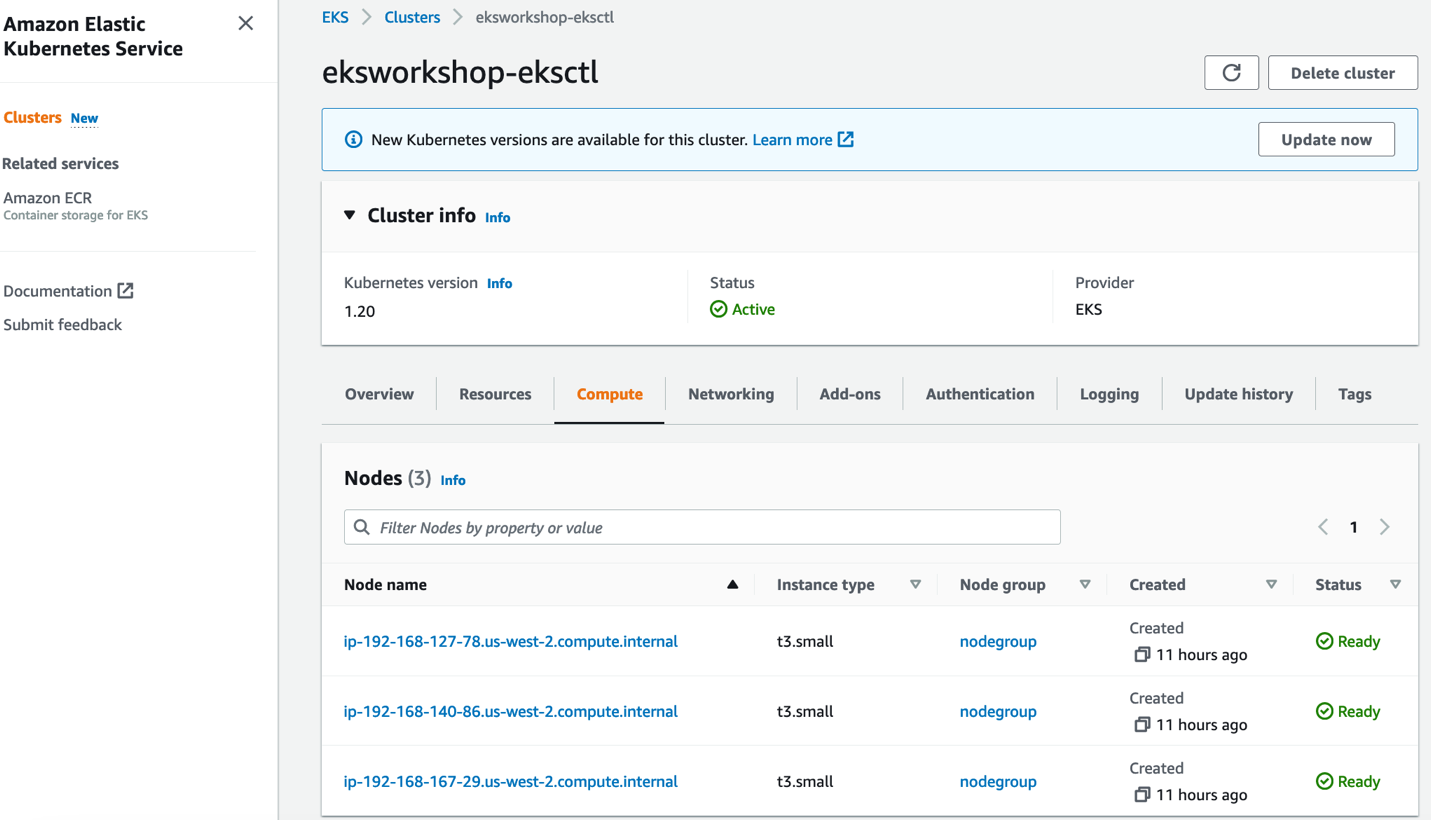


- Check if *kubectl get nodes* command works. If nodes are ready, we’re good to go.



- Go back to AWS console and search EKS to check the overall configuration of your cluster. (eksworshop-eksctl)

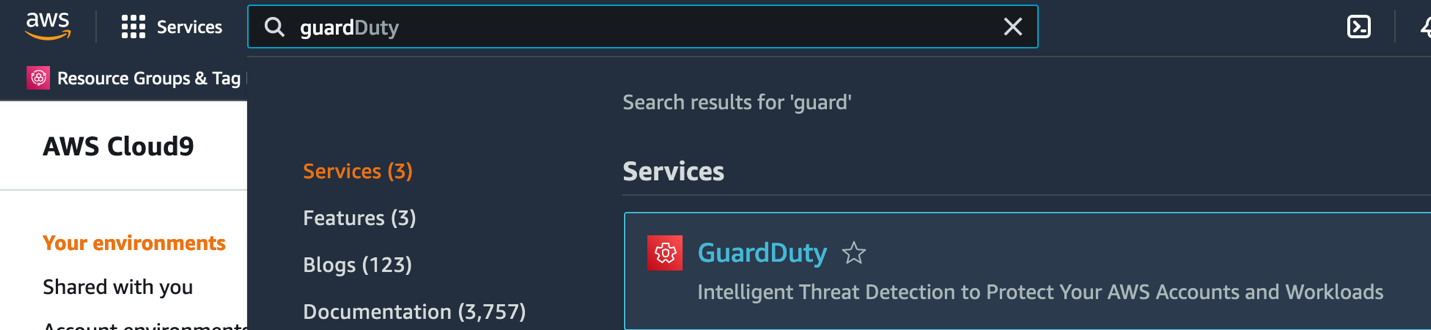


****

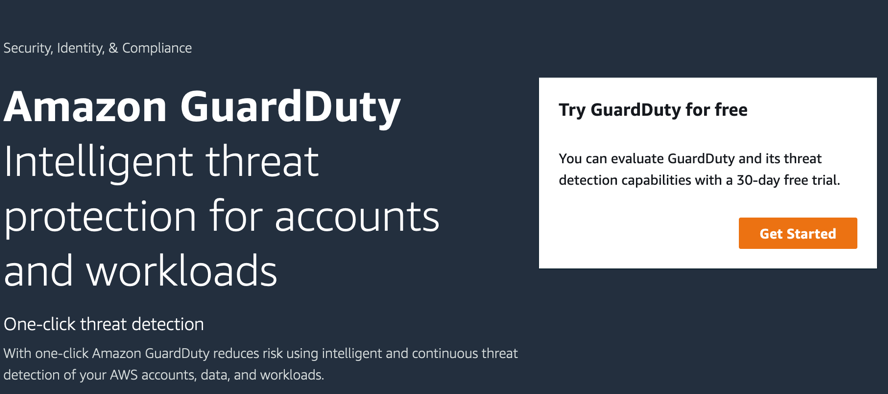
**2. Enable GuardDuty Findings on EKS**

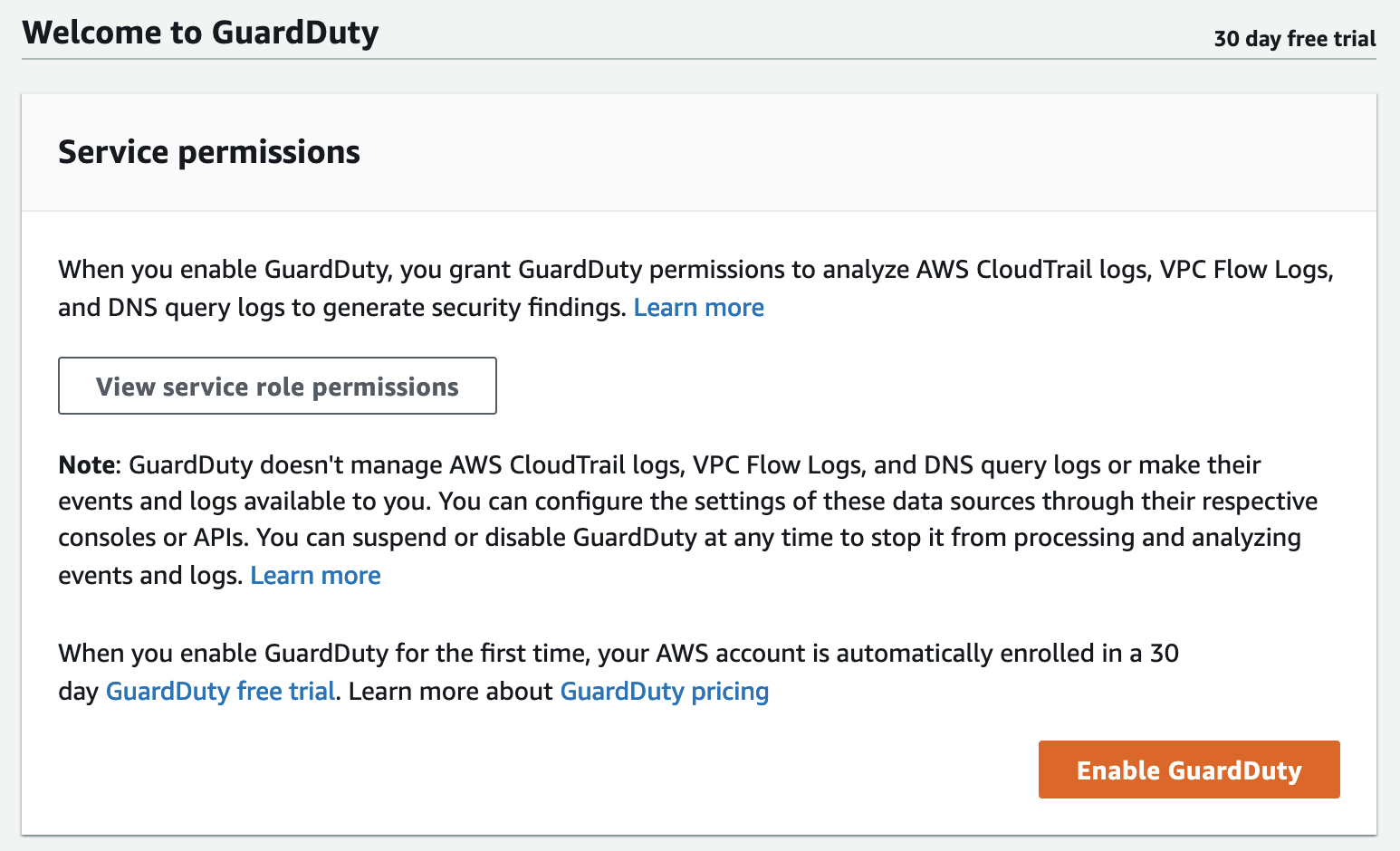
In this section, we will enable GuardDuty and Kubernetes protection.

- Search GuardDuty in AWS console



- Click get started



- Click Enable GuardDuty  


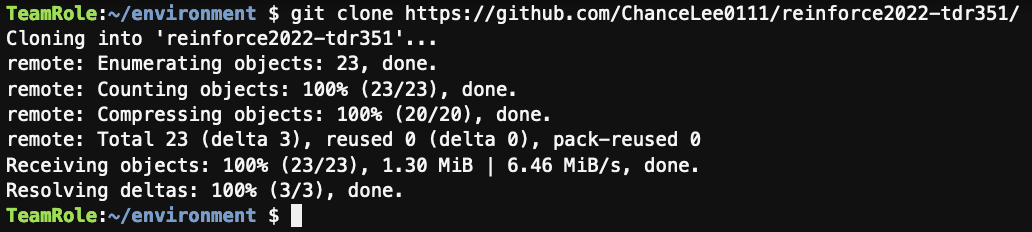
- Check if Kubernetes Protection is enabled. Go to *Findings*. Check there is no any findings available yet.



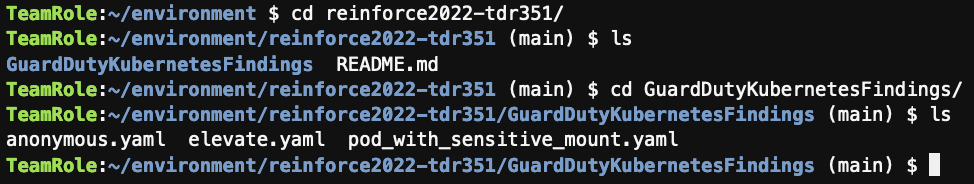
**3. Generate Kubernetes Findings on Cloud9 instance.**

In this section, we will generate various Kubernetes findings on your Amazon EKS cluster from your Cloud9 instance.

- Go back to your Cloud9 terminal and clone git repo that has GuardDuty Kubernetes sample findings.   
*git clone https://github.com/ChanceLee0111/reinforce2022-tdr351/*

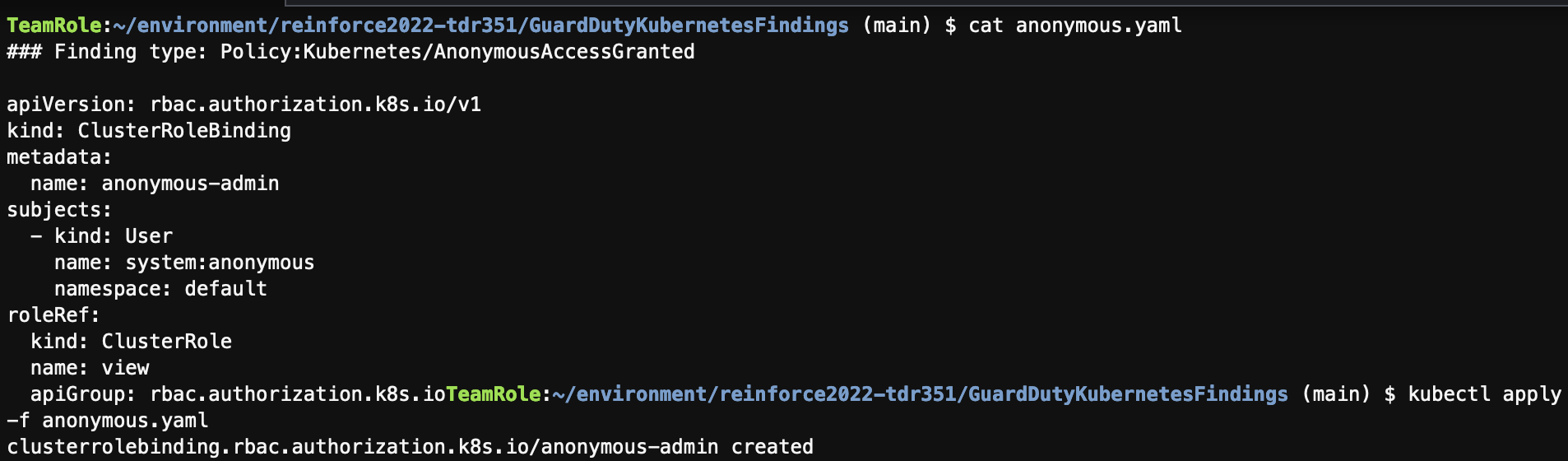


- Under GuardDutyKubernetesFindings directory, you can find 3 sample templates. (anonymous.yaml, elevate.yaml, pod\_with\_sensitive\_mount.yaml)



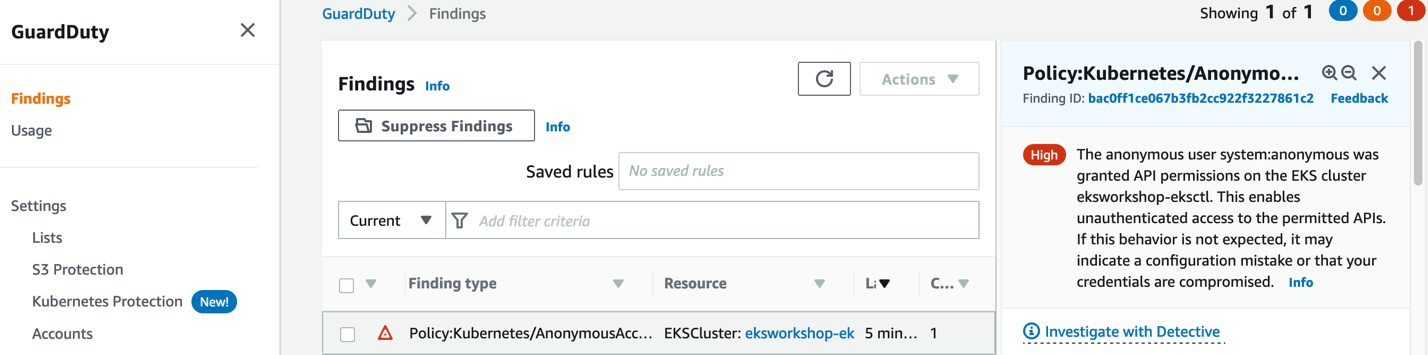
- Let’s start from the anonymous.yaml file.

$ *cat anonymous.yaml* to check the body and run kubectl apply -f anonymous.yaml

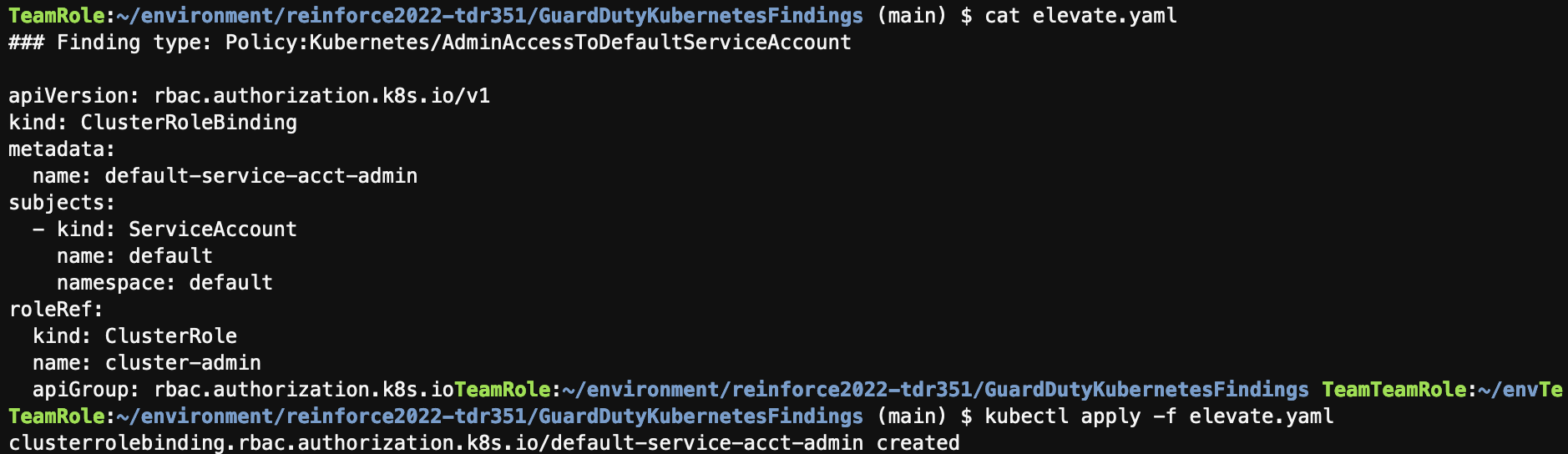


- You can check details of this finding and Remediation recommendations:  
<https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-kubernetes.html#policy-kubernetes-anonymousaccessgranted>

- Go back to AWS Console, Search GuardDuty and check the generated finding.



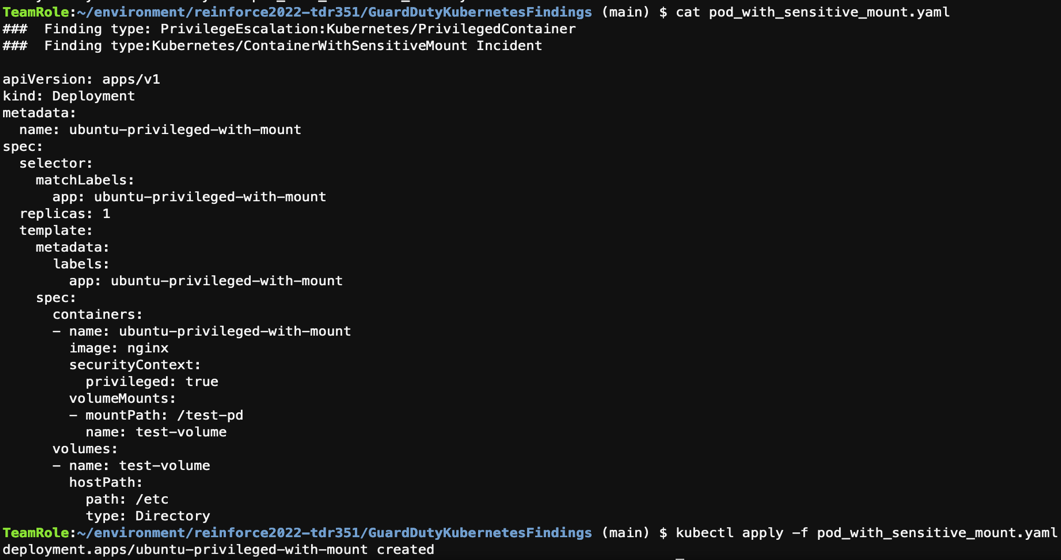
- Go back to Cloud 9 terminal and follow the same instruction for elevate.yaml.   
$ *cat elevate.yaml, kubectl apply -f elevate.yaml*



## - This finding is regarding Policy:Kubernetes/AdminAccessToDefaultServiceAccount <https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-kubernetes.html#policy-kubernetes-adminaccesstodefaultserviceaccount>

[Note] This finding takes some time to be populated on the guardduty

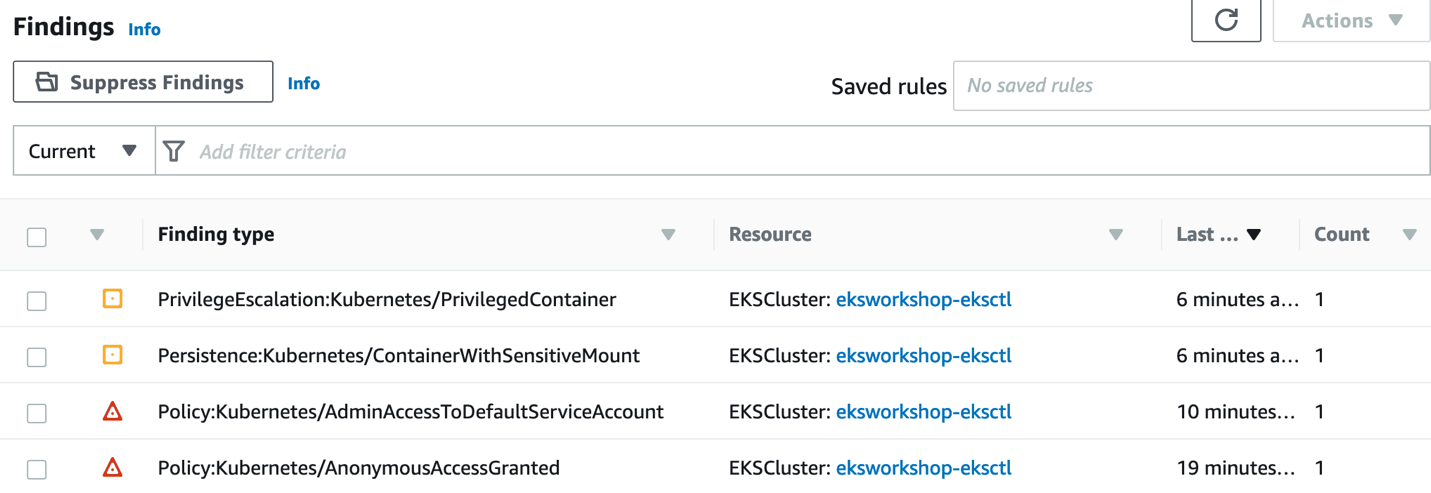
- Let’s do the same for pod\_with\_sensitive\_mount.yaml



- This yaml file generates 2 following findings.

## PrivilegeEscalation:Kubernetes/PrivilegedContainer <https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-kubernetes.html#privilegeescalation-kubernetes-privilegedcontainer>

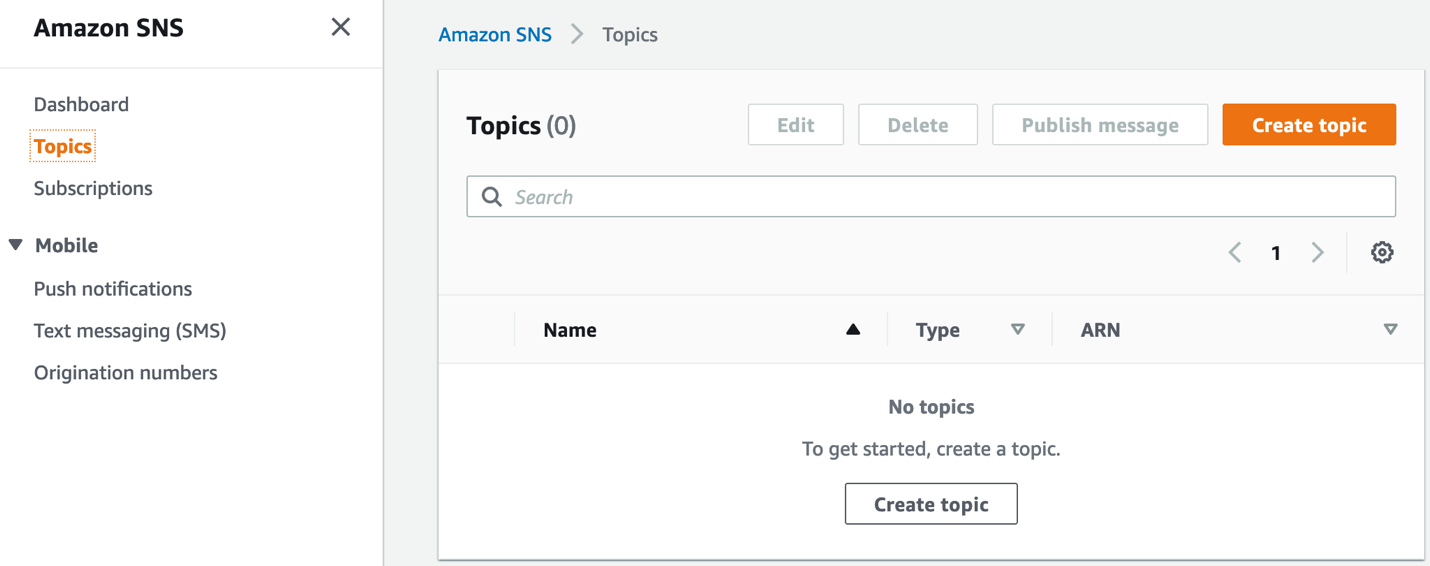
## Persistence:Kubernetes/ContainerWithSensitiveMount <https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-kubernetes.html#persistence-kubernetes-containerwithsensitivemount>

- Go to AWS GuardDuty AWS Console to check findings  


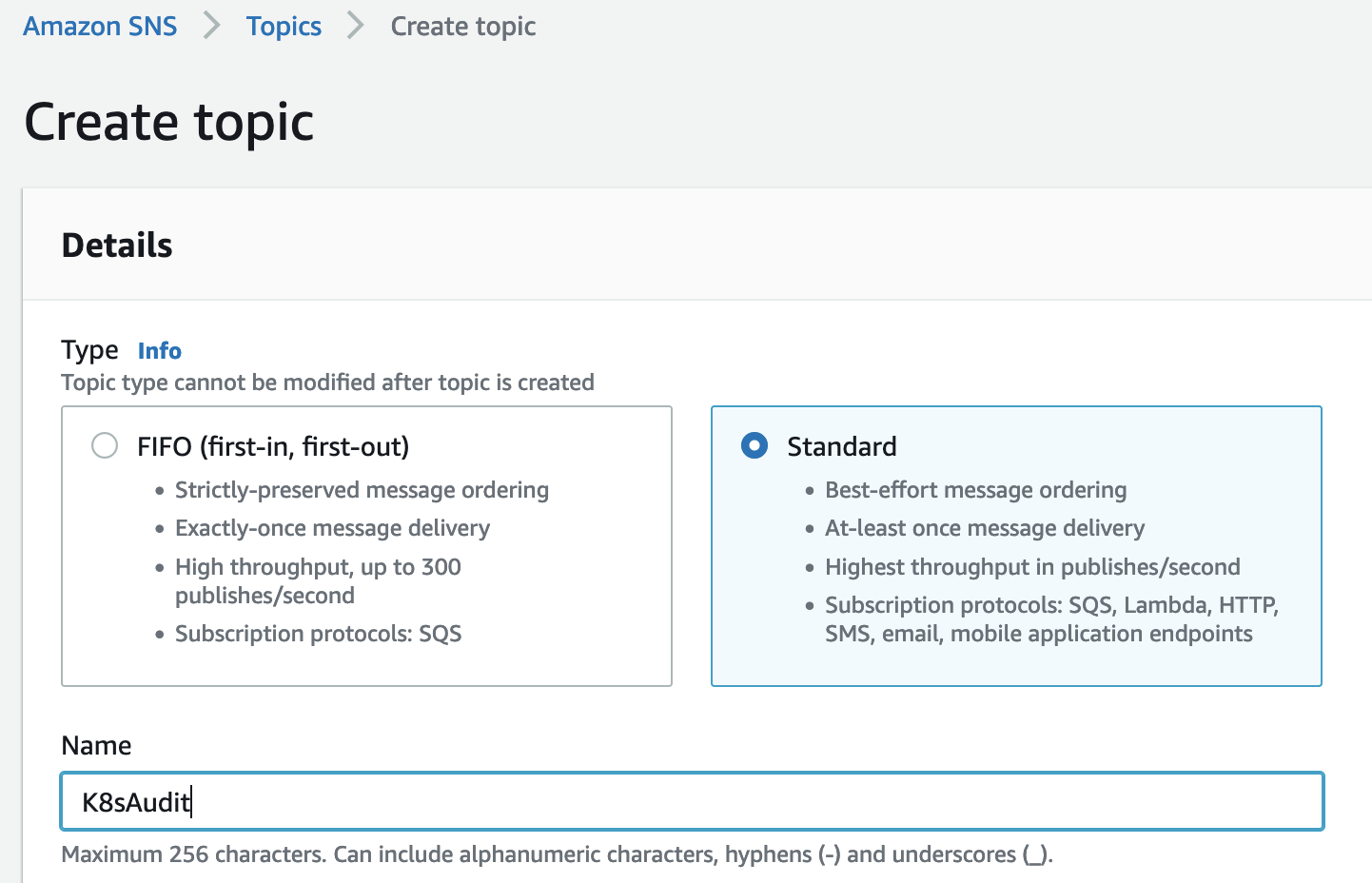
**5. Automated incident response**

In this section, we will create an EventBridge rule to filter specific GuardDuty Kubernetes events and send messages to Amazon SNS topic which has an email subscription to get an automated email notification. To that, let’s first create the Amazon SNS topic and subscribe it with your email address.

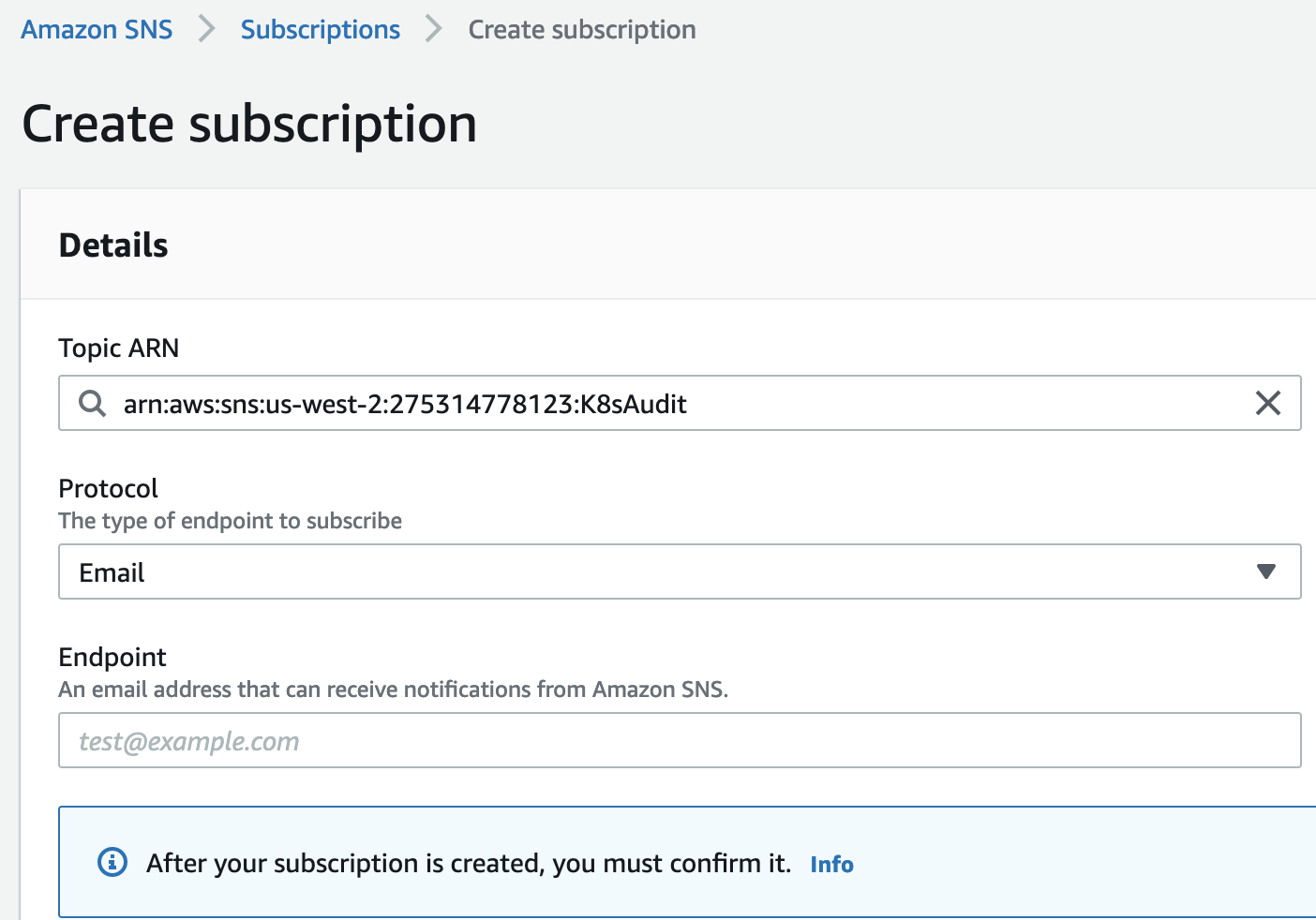
- Search SNS on AWS Console, go to Topics and click create topic.



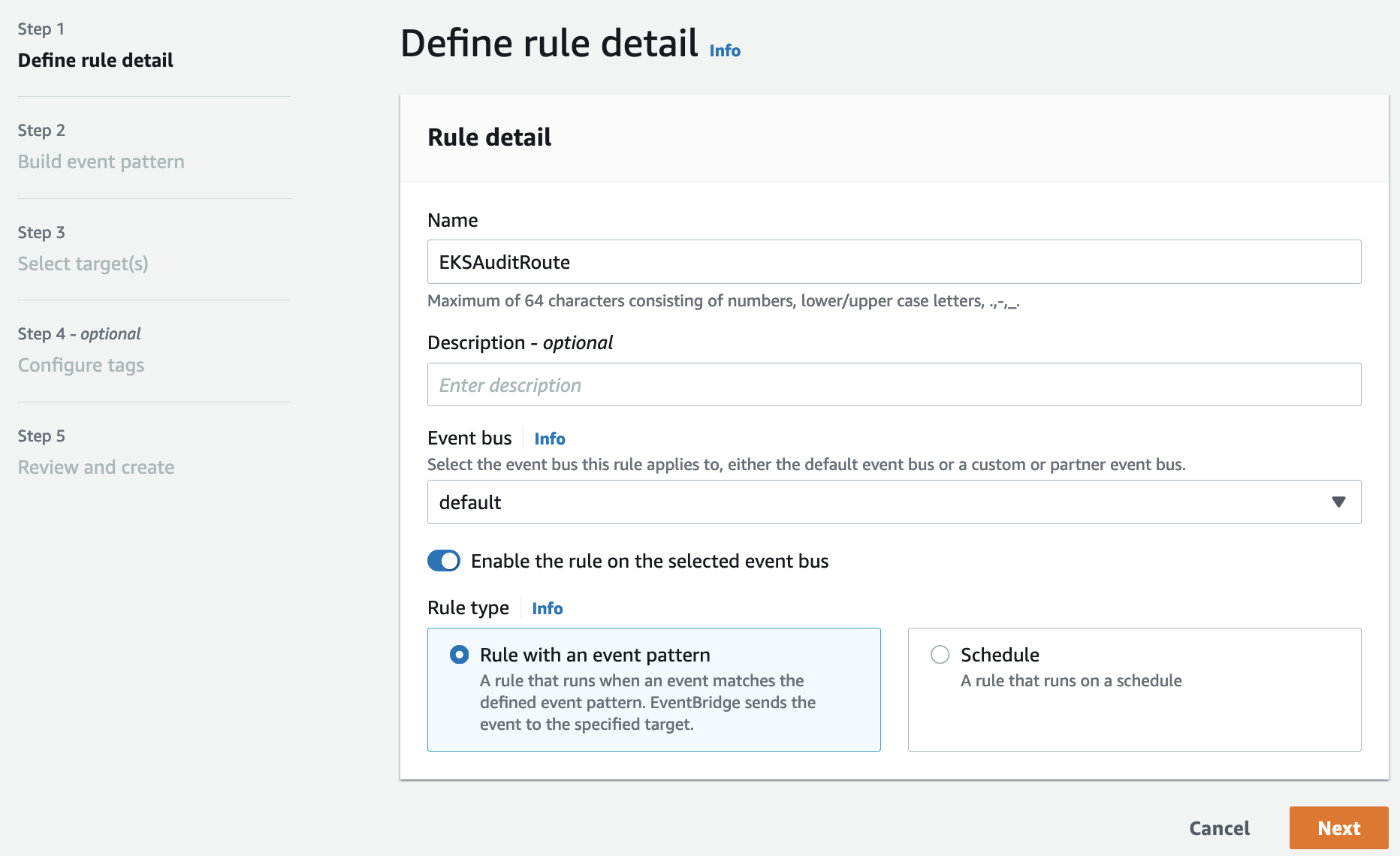
- Select a Standard type, name it as K8sAudit and keep everything as default and create it.



- Create Subscription, choose Email as a protocol and add your email address as an endpoint. Keep everything else as default. Log in to your email that you specify and confirm the subscription to the SNS topic.

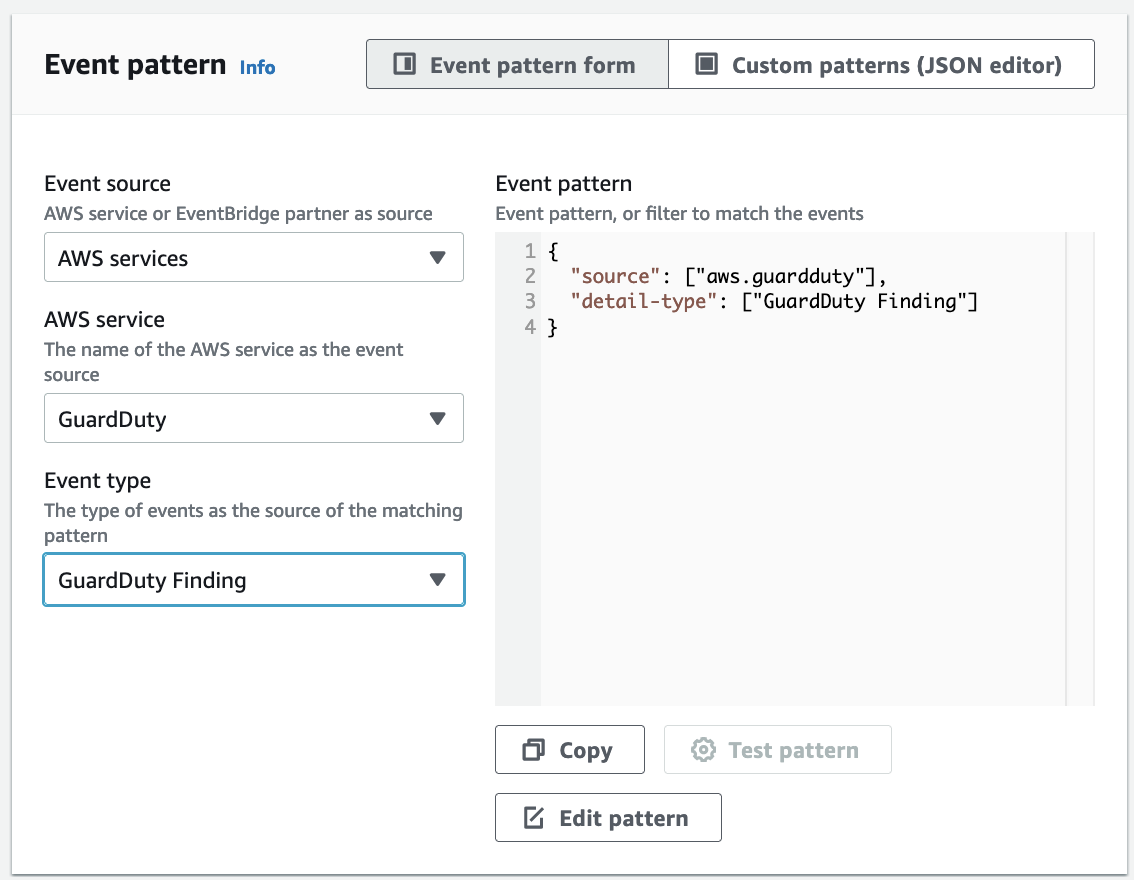


- Let’s create an eventbridge rule to catch Guardduty Kubernetes findings gand route messages to the SNS topic. Go to the Amazon EventBridge Console and click Create rule. Name it as EKSAuditRoute and keep everything as default.



- Keep everything as default on Event source and Sample event, scroll down to the Event Pattern.

* Event source : AWS services
* AWS service : GuardDuty
* Event type : GuardDuty Finding



- The default event pattern matches all the Guardduty Finding. Let’s customize it as following. This event pattern will match one of described types. Click Edit pattern and add the following pattern. You can find more information of Event Pattern rules.

<https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-event-patterns.html>

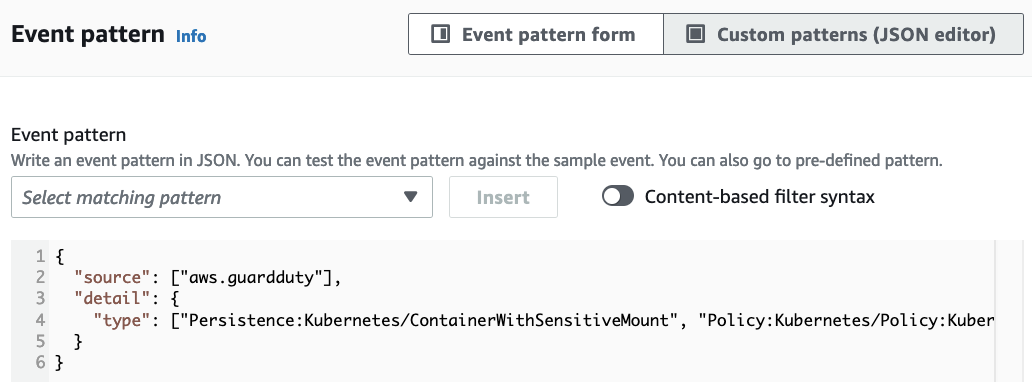
{

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

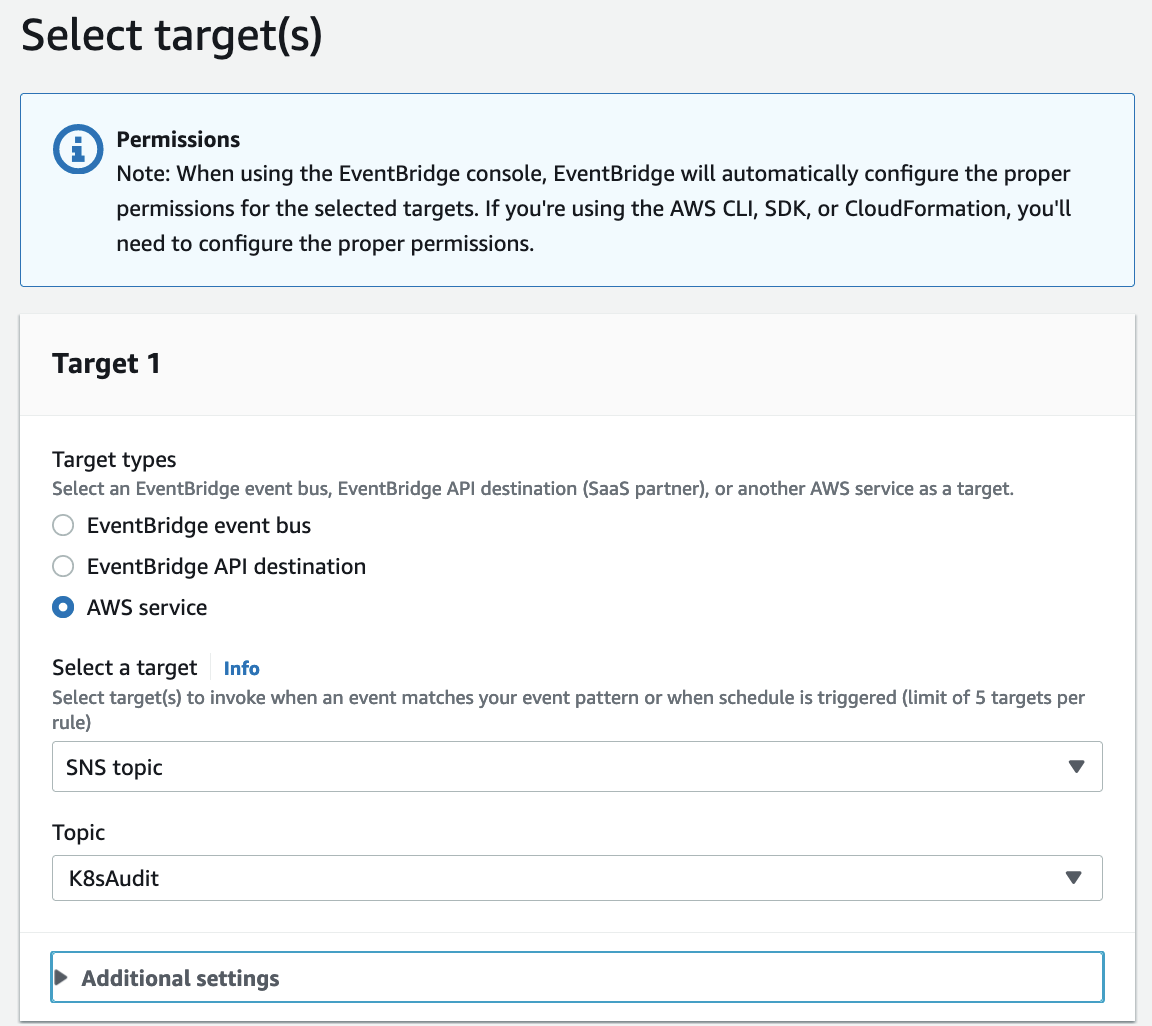
"detail": {

"type": ["Persistence:Kubernetes/ContainerWithSensitiveMount", "Policy:Kubernetes/Policy:Kubernetes/AdminAccessToDefaultServiceAccount", "Policy:Kubernetes/AnonymousAccessGranted", "PrivilegeEscalation:Kubernetes/PrivilegedContainer", "Execution:Kubernetes/ExecInKubeSystemPod"]

}



- Click next, select Target types : AWS service, Select a target : SNS topic, select a topic created from the last section and click Next. Skip Tags. Review and Create rule.



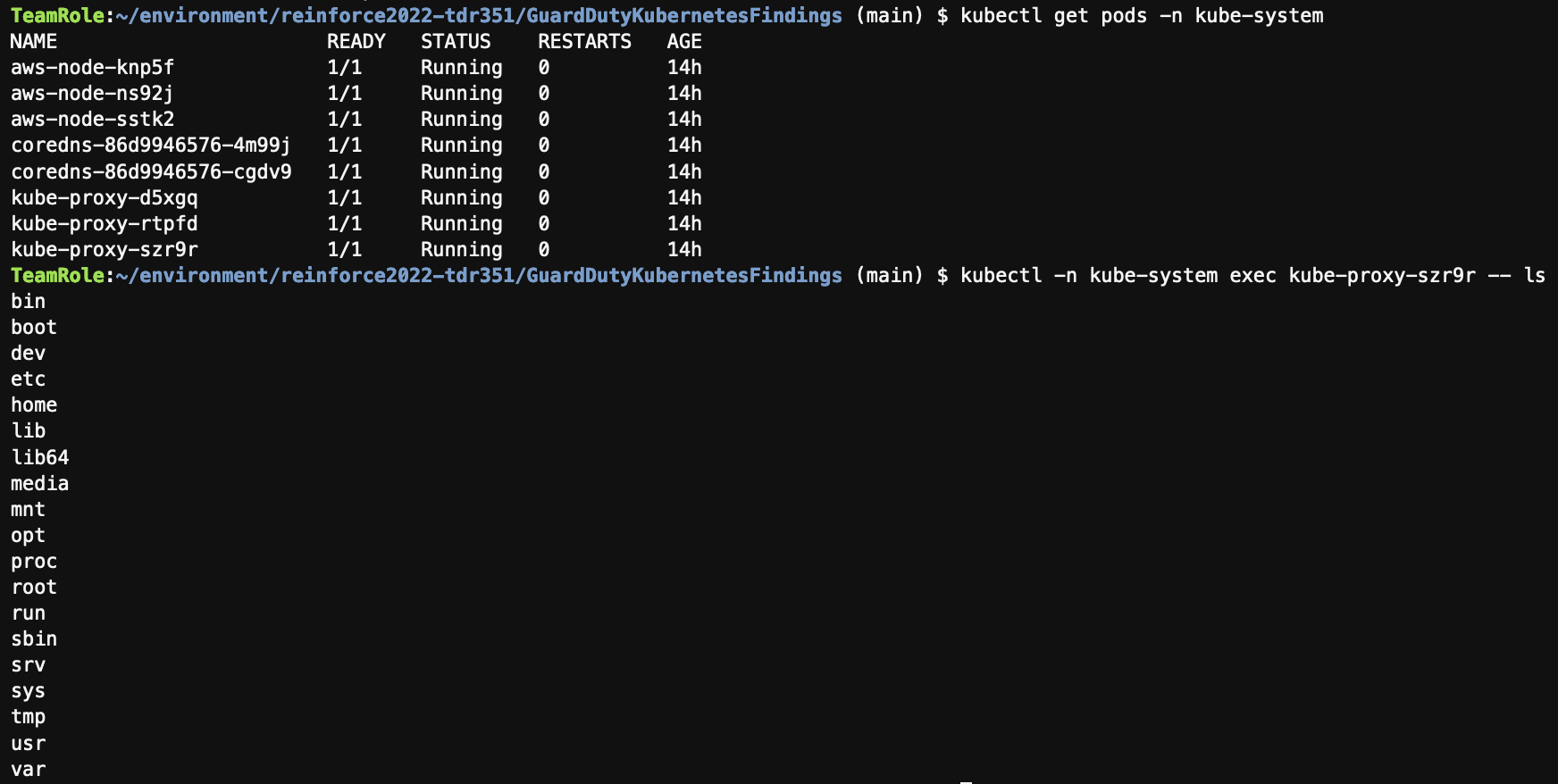
- Now, let’s generate relevant findings on your cloud9 terminal to check if the EventBridge rule works to route events from GuardDuty to the target, SNS.

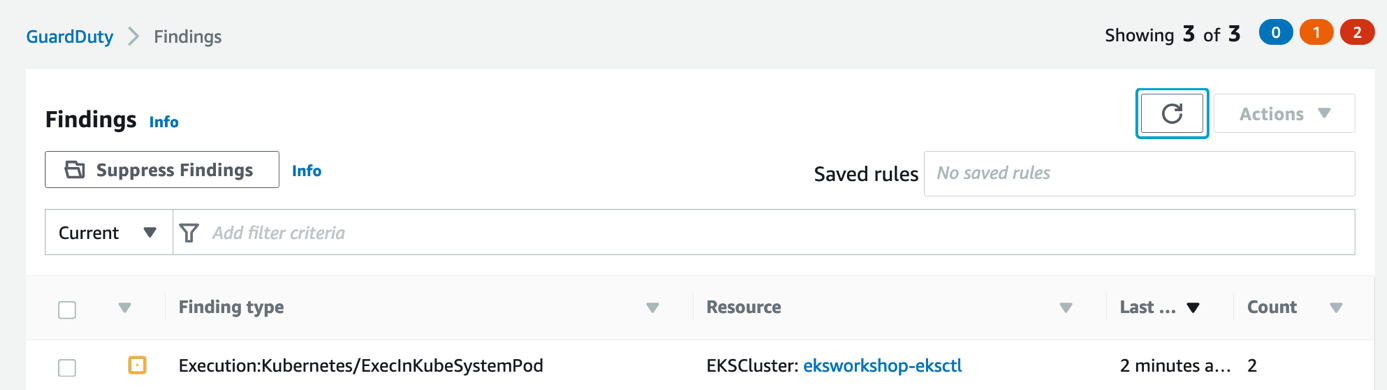
- Go back to Cloud9 terminal, and let’s generate a new finding, *Execution:Kubernetes/ExecInKubeSystemPod*

<https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-kubernetes.html#execution-kubernetes-execinkubesystempod>

*$ kubectl -n kube-system get pods*

*$ kubectl -n kube-system exec kube-proxy-xxxxx (an exact name varies) -- ls*

**

- Go back to GuardDuty console to check if a finding is generated. Once it is generated, you will receive an   


- You will receive an email (a subscription on the SNS topic) shortly. You can try re-generating other finding types by modifying the name of resource in the yaml files.

[Note] Finding changes that will create a CloudWatch event include newly generated findings or newly aggregated findings. Events are emitted on a best effort basis. For the testing purpose, you may archive previous findings on the GuardDuty and re-generate findings.

<https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings_cloudwatch.html>