National College of Ireland

Project Submission Sheet

Student Name:	Balamuralikrishna Raju		
Student ID:	23326701		
Programme:	MSc in cloud computing	**Year:**	2025 - 2026
Module:	Cloud Computing Research Project		
Lecturer:	Shaguna Gupta		
Submission Due Date:	11/12/2025		
Project Title:	Comparative Analysis of Open-Source Kubernetes Runtime Security Tools in Cloud Platforms of AKS and EKS		
Word Count:	1735		

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 references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Balamuralikrishna Raju

Date: 11/12/25

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

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

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

Your Name/Student Number	Course	Date

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

National College of Ireland

Configuration Manual

MSc Research Project
MSc Cloud Computing

Balamuralikrishna Raju
Student ID: 23326701

School of Computing
National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Balamuralikrishna Raju
Student ID:	23326701
Programme:	MSc Cloud Computing **Year:** 2025 – 2026
Module:	Cloud Computing Research Project
Lecturer:	Shaguna Gupta
Submission Due Date:	11/12/2025
Project Title:	Comparative Analysis of Open-Source Kubernetes Runtime Security Tools in Cloud Platforms of AKS and EKS
Word Count:	1735 **Page Count:** .....14.....

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: Balamuralikrishna Raju

Date: 11/12/2025.....

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):	

Configuration Manual

Balamuralikrishna Raju
23326701

Introduction

The agenda of the project is to use the open-source Kubernetes security tools like Falco, Kube Armor, Tetragon in managed Kubernetes cloud platforms like AKS and EKS to see how well these tools performs in detecting and preventing any attack into the microservice which is running in the cluster, and how much resource it consumes in the process.

Configuration setup

Initially we need to install certain tools to initialize the setup, we can use our local command prompt or virtual machine from the respective cloud to access create, access, and deploy microservices and security tools in the Kubernetes cluster.

Install AWS cli

To install run the following commands

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install

confirm the installation by running the below command
aws –version

Install kubectl – It is used to interact with the Kubernetes cluster
Run the below commands
Download the latest release with the below command
curl -LO https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl

validate the binary
curl -LO https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl.sha256

Validate the kubectl binary against the checksum file
echo "**$(**cat kubectl.sha256**)** kubectl" | sha256sum –check
If valid, the output is:
kubectl: OK

Install kubectl
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

Install Helm

We need helm tool, to deploy the open-source Kubernetes security tools in the cluster using helm charts. Depending on the OS we use, the command to install helm varies, I used ubuntu OS machine, so these are the following commands used to install helm

sudo apt-get install curl gpg apt-transport-https --yes
curl -fsSL https://packages.buildkite.com/helm-linux/helm-debian/gpgkey | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/helm.gpg] https://packages.buildkite.com/helm-linux/helm-debian/any/ any main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
sudo apt-get update
sudo apt-get install helm

Install Azure cli

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

```
ubuntu@ip-172-31-14-108:~$ kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1
ubuntu@ip-172-31-14-108:~$ aws --version
aws-cli/2.31.31 Python/3.13.9 Linux/6.14.0-1016-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-14-108:~$ helm version
version.BuildInfo{Version:"v3.19.0", GitCommit:"3d8990f0836691f0229297773f3524598f46bda6", GitTreeState:"clean", GoVersion:"go1.24.7"}
```

Creation of EKS cluster

Creation cluster is created via eksctl tool, for that need of eksctl tool is necessary

For unix
Run the below commands one by one

# for ARM systems, set ARCH to: `arm64`, `armv6` or `armv7`
ARCH=amd64
PLATFORM=$(uname -s)_$ARCH

curl -sLO "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_$PLATFORM.tar.gz"

(Optional) Verify checksum
curl -sL "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_checksums.txt" | grep $PLATFORM | sha256sum --check

tar -xzf eksctl_$PLATFORM.tar.gz -C /tmp && rm eksctl_$PLATFORM.tar.gz

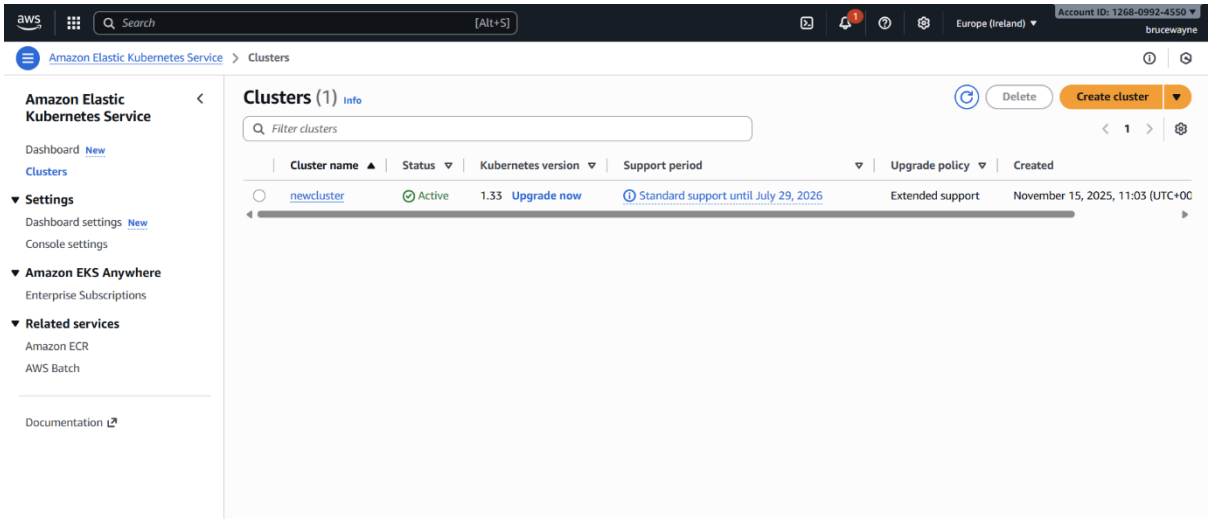
sudo install -m 0755 /tmp/eksctl /usr/local/bin && rm /tmp/eksctl

paste the below content in a yaml file , save the file.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: newcluster          #name of the cluster which you want to set as
  region: eu-west-1      #where you want to create the cluster
  version: "1.33"       #cluster version need to specify here

vpc:
  id: vpc-0ba18fd30f1962ef4       #virtual network id(the virtual network where you gonna place eks cluster)
  subnets:
    public:
      eu-west-1c:
        id: subnet-0adaca70583a3f279  #given two public subnet id of the respective vpc to place resources in the subnet getting access from interent
      eu-west-1a:
        id: subnet-016ae80e752874c0c
    private:
      eu-west-1c:
        id: subnet-06822e5a322045f29  #given two private subnet id of the respective vpc to place resources in the subnet forbid access from interent
      eu-west-1a:
        id: subnet-05b542d1c8eafc9d7
managedNodeGroups:
  - name: workernodesofnewcluster    #name of node-group
    amiFamily: AmazonLinux2023

    instanceType: t2.large  #machine type like defining how much cpu and ram we require(based on it choose the instance type)
    minSize: 2   #minimum no of nodes
    maxSize: 4   #maximum no of nodes, to scale up the node based on load
    desiredCapacity: 2
    volumeSize: 40    #storage size of machine

    iam:
      attachPolicyARNs:
        - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
```

To apply this file, run kubectl apply -f nameofthefile.yaml

Generally, it takes 15 to 20 minutes for the creation of the Kubernetes cluster. To check whether the cluster is active or not, run the below command



To get connected to the cluster run the following command
aws eks update-kubeconfig --region us-west-1 --name newcluster

aws eks describe-cluster --name newcluster --region eu-west-1 --query "cluster.status"
It will say status is active
To see the nodes of the cluster
Run – kubectl get nodes

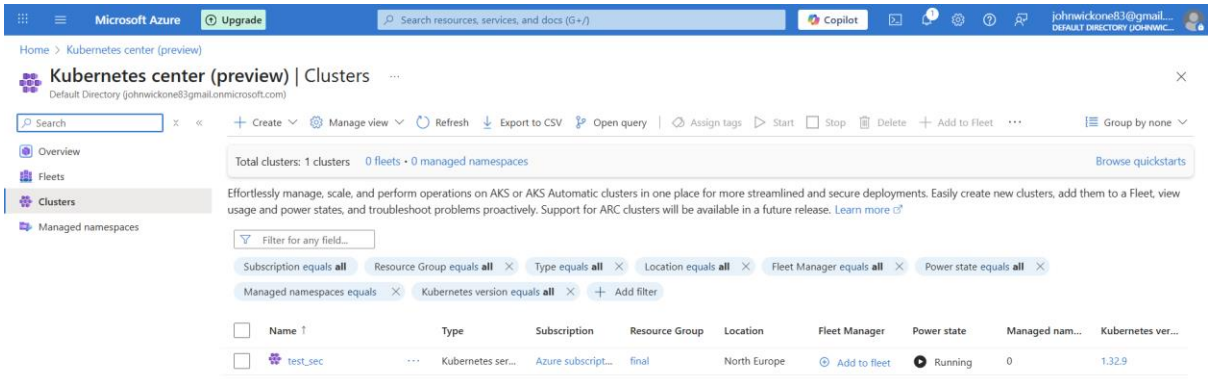
```
ubuntu@ip-172-31-14-108:~$ aws eks describe-cluster --name newcluster --region eu-west-1 --query "cluster.status"
"ACTIVE"
ubuntu@ip-172-31-14-108:~$ kubectl get nodes
NAME                                         STATUS   ROLES    AGE   VERSION
ip-172-31-13-144.eu-west-1.compute.internal   Ready    <none>   20d   v1.33.5-eks-c39b1d0
ip-172-31-17-133.eu-west-1.compute.internal   Ready    <none>   20d   v1.33.5-eks-c39b1d0
ubuntu@ip-172-31-14-108:~$
```

Create AKS cluster via azure cli

Run az login
And choose the subscription of the account

Run the following command to create the AKS cluster

```
az aks create \
  --resource-group final \
  --name test_sec \
  --location northeurope \
  --kubernetes-version 1.32.9 \
  --node-count 2 \
  --node-vm-size Standard_L2aos_v4 \
  --network-plugin azure \
  --enable-rbac \
  --generate-ssh-keys \
  --nodepool-name agentpool \
  --tier free
```



To get connected to the cluster run the following command
az aks get-credentials --resource-group final --name test_sec

To see the nodes of the created cluster run
Kubectl get nodes

```
C:\Users\bala>kubectl get nodes
NAME                                STATUS   ROLES    AGE    VERSION
aks-agentpool-14248651-vmss000000   Ready    <none>   3d1h   v1.32.9
aks-agentpool-14248651-vmss000001   Ready    <none>   3d1h   v1.32.9
```

Deployment of microservice and security tools – common to both AKS and EKS
Deploy Falco

To install helm repository run the following commands
helm repo add falcosecurity https://falcosecurity.github.io/charts
helm repo update

helm install --replace falco --namespace falco --create-namespace --set tty=true falcosecurity/falco

Check the Falco pods are running
kubectl get pods -n falco

Deploy a microservice
kubectl create deployment nginx --image=nginx

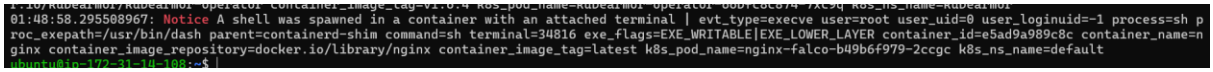
By default, the falco has certain rules, which will apply to all the pods running in the cluster, if needed, the rules can be customized to monitor specific pod or namespace
Apply the Falco rules in the Kubernetes cluster

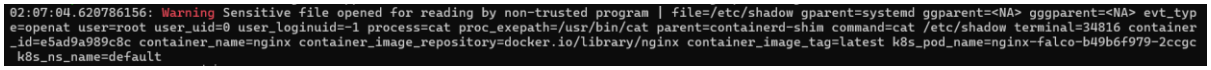
IN EKS
Set the environment variable for the pod
set POD variable
POD=$(kubectl get pod -l app=nginx -o name)

execute shell
To get into the container and execute shell commands run the below command
kubectl exec -it $(kubectl get pods --selector=app=nginx-falco -o name) – sh

To check the logs, where detection is stored, run the below command
kubectl logs -l app.kubernetes.io/name=falco -n falco -c falco | grep Notice

sensitive file access
kubectl exec -it $(kubectl get pods --selector=app=nginx-falco -o name) -- cat /etc/shadow
kubectl logs -l app.kubernetes.io/name=falco -n falco -c falco | grep Warning

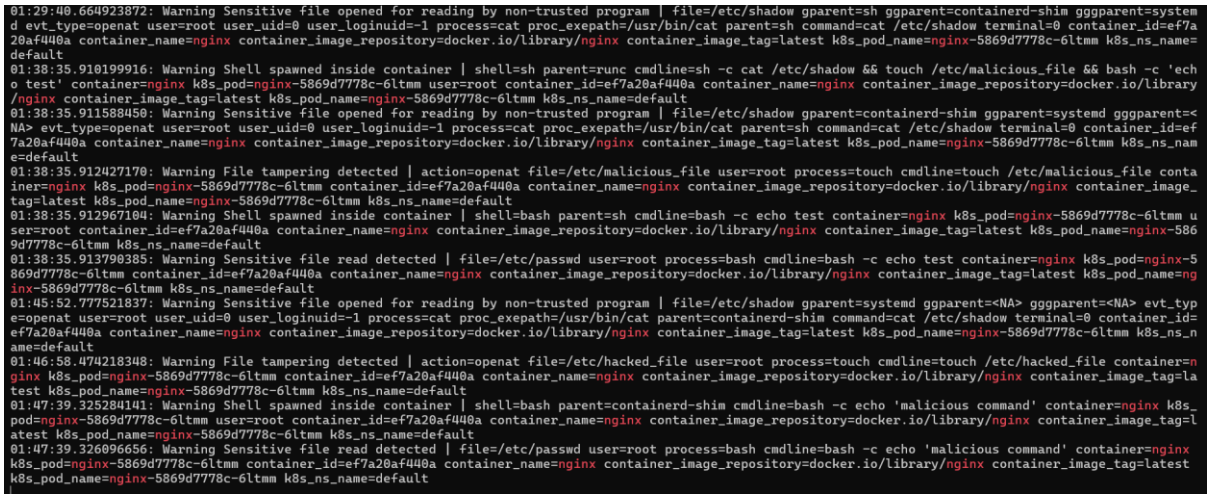




In AKS

To watch the logs continuously, in one terminal run the below command
kubectl logs -n falco falco-pjrqk -c falco -f | grep nginx

Sensitive file access
kubectl exec -n default nginx-5869d7778c-6ltmm -- cat /etc/shadow
File tampering
kubectl exec -n default nginx-5869d7778c-6ltmm -- touch /etc/hacked_file
Shell execution
kubectl exec -n default nginx-5869d7778c-6ltmm -- bash -c "echo 'malicious command'



Deployment of Kube Armor – common for both EKS and AKS

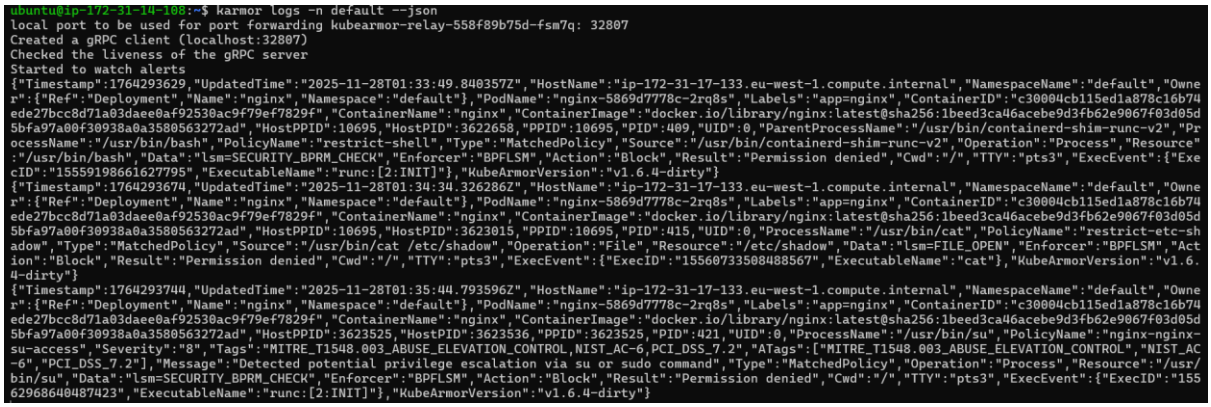
helm repo add kubearmor https://kubearmor.github.io/charts
helm repo update kubearmor
helm upgrade --install kubearmor-operator kubearmor/kubearmor-operator -n kubearmor --create-namespace
kubectl apply -f https://raw.githubusercontent.com/kubearmor/KubeArmor/main/pkg/KubeArmorOperator/config/samples/sample-config.yml

Install KArmor CLI
curl -sfL http://get.kubearmor.io/ | sudo sh -s -- -b /usr/local/bin

Deploy a test nginx application
kubectl create deployment nginx --image=nginx
Apply the Kube Armor policies by kubectl apply -f name of the file.yaml
set POD variable
POD=$(kubectl get pod -l app=nginx -o name)
In EKS

Execute shell
karmor logs -n default --json - run in a different terminal
kubectl exec -it $POD -- /bin/bash
Sensitive file access
karmor logs -n default --json - run in a different terminal
kubectl exec -it $POD -- cat /etc/shadow
Privilege escalation
karmor logs -n default --json - run in a different terminal
kubectl exec -it $POD -- su

```
ubuntu@ip-172-31-14-108:~$ POD=$(kubectl get pod -l app=nginx -o name)
ubuntu@ip-172-31-14-108:~$ kubectl exec -it $POD -- /bin/bash
exec /bin/bash: permission denied
command terminated with exit code 255
ubuntu@ip-172-31-14-108:~$ kubectl exec -it $POD -- cat /etc/shadow
cat: /etc/shadow: Permission denied
command terminated with exit code 1
ubuntu@ip-172-31-14-108:~$ kubectl exec -it $POD -- su
exec /usr/bin/su: permission denied
command terminated with exit code 255
ubuntu@ip-172-31-14-108:~$
```



In AKS

Apply the Kube Armor policies by kubectl apply -f name of the file.yaml
In terminal one – run the following command to see the logs continuously
karmor.exe logs -n default –json

Sensitive file access
kubectl exec -n default nginx-5869d7778c-g8s99 -- cat /etc/shadow
File tampering
kubectl exec -n default nginx-5869d7778c-g8s99 -- touch /etc/blocked-file
shell execution
kubectl exec -n default nginx-5869d7778c-g8s99 -- sh -c "ls /etc"

```
azureuser@cluster-access:~$ kubectl exec -n default nginx-7b557bc76d-95rlt -- sh -c "echo test"
test
azureuser@cluster-access:~$ kubectl exec -n default nginx-7b557bc76d-95rlt -- cat /etc/shadow
cat: /etc/shadow: Permission denied
command terminated with exit code 1
azureuser@cluster-access:~$ kubectl exec -n default nginx-7b557bc76d-95rlt -- touch /etc/test-file
touch: cannot touch '/etc/test-file': Permission denied
command terminated with exit code 1
azureuser@cluster-access:~$
```

Deployment of Tetragon – common for both AKS and EKS
Only the nginx pod name changes with respect to clusters.

Install tetragon

helm repo add cilium https://helm.cilium.io
helm repo update
helm install tetragon ${EXTRA_HELM_FLAGS[@]} cilium/tetragon -n kube-system
kubectl rollout status -n kube-system ds/tetragon -w

create the tetragon detection policies for sensitive file access, file tampering, and shell execution and apply that policies, using the below command

kubectl apply -f tetragon-sensitive-file-access.yaml and similarly apply the other two policies as well.

Testing the detection
Sensitive file access
kubectl exec nginx-5869d7778c-6ltmm -n default -- sh -c "cat /etc/shadow > /dev/null"
kubectl exec nginx-5869d7778c-6ltmm -n default -- sh -c "cat /etc/passwd > /dev/null"

File tampering
kubectl exec nginx-5869d7778c-6ltmm -n default -- sh -c "echo 'test' > /etc/test-tampering.txt"
kubectl exec nginx-5869d7778c-6ltmm -n default -- sh -c "touch /bin/testfile 2>&1"

Shell execution
kubectl exec nginx-5869d7778c-6ltmm -n default -- bash -c "whoami"
kubectl exec nginx-5869d7778c-6ltmm -n default -- sh -c "id"

```
azureuser@cluster-access:~$ kubectl logs tetragon-477cv -n tetragon -c export-stdout --tail=500 | grep "detect-file-tampering" | tail -5
{"process_kprobe":{"process":{"exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjQ1MzgzMjY5OTQyNTgyOjcyODQ0Nw==", "pid":728447, "uid":0, "cwd":"/", "b
inary":"/usr/bin/sh", "arguments":"-c \"echo 'test' > /etc/testfile\"", "flags":"execve rootcwd clone", "start_time":"2025-12-03T12:54:00.811775380Z", "auid
":4294967295, "pod":{"namespace":"default", "name":"nginx-5869d7778c-6ltmm", "uid":"8ae7f498-dd2b-4a04-928a-dbf5ea3f4008", "container":{"id":"containerd://e
f7a20af440ae16f9897e2029c7eac92dbaf19352bed6759fe35cde22aa74f20", "name":"nginx", "image":{"id":"docker.io/library/nginx@sha256:553f64aecdc31b5bf944521731cd
70e35da4faed96b2b7548a3d8e2598c52a42", "name":"docker.io/library/nginx:latest"}, "start_time":"2025-12-03T00:43:46Z", "pid":134, "security_context":{}}, "po
d_labels":{"app":"nginx", "pod-template-hash":"5869d7778c"}, "workload":"nginx", "workload_kind":"Deployment"}, "docker":"ef7a20af440ae16f9897e2029c7eac9",
"parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjE1NjQ4NjAwMDAwMDA6Mjk4NDE=", "refcnt":1, "tid":728447, "in_init_tree":false}, "parent":{"exe
c_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjE1NjQ4NjAwMDAwMDA6Mjk4NDE=", "pid":29841, "uid":0, "cwd":"/run/containerd/io.containerd.runtime.v2.task
/k8s.io/a7ce165d04d5dff52f5986a8215a8b47bf5f65f126aeeafa5e02316d9d580734", "binary":"/usr/bin/containerd-shim-runc-v2", "arguments":"-namespace k8s.io -id a
7ce165d04d5dff52f5986a8215a8b47bf5f65f126aeeafa5e02316d9d580734 -address /run/containerd/containerd.sock", "flags":"procFS", "start_time":"2025-12-03T00:43:
42.401832708Z", "auid":4294967295, "parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjM0MDAwMDAwMDox", "tid":29841, "in_init_tree":false}, "fun
ction_name":"security_file_permission", "args":[{"file_arg":{"path":"/etc/testfile", "permission":"-rw-r--r--"}}, {"int_arg":2}], "action":"KPROBE_ACTION_PO
ST", "policy_name":"detect-file-tampering", "return_action":"KPROBE_ACTION_POST"}, "node_name":"aks-agentpool-14248651-vmss000001", "time":"2025-12-03T12:54
:00.812553474Z", "node_labels":{"agentpool":"agentpool", "beta.kubernetes.io/arch":"amd64", "beta.kubernetes.io/instance-type":"Standard_L2aos_v4", "beta.ku
bernetes.io/os":"linux", "failure-domain.beta.kubernetes.io/region":"northeurope", "failure-domain.beta.kubernetes.io/zone":"northeurope-2", "kubernetes.azu
re.com/agentpool":"agentpool", "kubernetes.azure.com/azure-cni-overlay":"true", "kubernetes.azure.com/cluster":"MC_final_test_sec_northeurope", "kubernetes.
azure.com/consolidated-additional-properties":"e0080963-cfdc-11f0-aace-5e38850148e8", "kubernetes.azure.com/kubelet-identity-client-id":"82105cbd-ca27-41fe-
9f30-ab604fc33298", "kubernetes.azure.com/kubelet-serving-ca":"cluster", "kubernetes.azure.com/localdns-state":"disabled", "kubernetes.azure.com/mode":"syst
em", "kubernetes.azure.com/network-name":"final-vnet", "kubernetes.azure.com/network-resourcegroup":"final", "kubernetes.azure.com/network-stateless-cni":"f
alse", "kubernetes.azure.com/network-subnet":"default", "kubernetes.azure.com/network-subscription":"8a24b207-6e52-4fb2-967a-773f648eed1a", "kubernetes.azur
e.com/node-image-version":"AKSUbuntu-2204gen2containerd-202511.07.0", "kubernetes.azure.com/nodenetwork-vnetguid":"b035f07e-f0ea-492c-9e81-28adc465ee8d", "k
ubernetes.azure.com/nodepool-type":"VirtualMachineScaleSets", "kubernetes.azure.com/os-sku":"Ubuntu", "kubernetes.azure.com/os-sku-effective":"Ubuntu2204",
"kubernetes.azure.com/os-sku-requested":"Ubuntu", "kubernetes.azure.com/podnetwork-type":"overlay", "kubernetes.azure.com/role":"agent", "kubernetes.azure.c
om/sku-cpu":"2", "kubernetes.azure.com/sku-memory":"16384", "kubernetes.io/arch":"amd64", "kubernetes.io/hostname":"aks-agentpool-14248651-vmss000001", "kub
ernetes.io/os":"linux", "node.kubernetes.io/instance-type":"Standard_L2aos_v4", "topology.disk.csi.azure.com/zone":"northeurope-2", "topology.kubernetes.io/
region":"northeurope", "topology.kubernetes.io/zone":"northeurope-2"}}
{"process_kprobe":{"process":{"exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjQ1MzgzMjY5OTQyNTgyOjcyODQ0Nw==", "pid":728447, "uid":0, "cwd":"/", "b
inary":"/usr/bin/sh", "arguments":"-c \"echo 'test' > /etc/testfile\"", "flags":"execve rootcwd clone", "start_time":"2025-12-03T12:54:00.811775380Z", "auid
":4294967295, "pod":{"namespace":"default", "name":"nginx-5869d7778c-6ltmm", "uid":"8ae7f498-dd2b-4a04-928a-dbf5ea3f4008", "container":{"id":"containerd://e
f7a20af440ae16f9897e2029c7eac92dbaf19352bed6759fe35cde22aa74f20", "name":"nginx", "image":{"id":"docker.io/library/nginx@sha256:553f64aecdc31b5bf944521731cd
70e35da4faed96b2b7548a3d8e2598c52a42", "name":"docker.io/library/nginx:latest"}, "start_time":"2025-12-03T00:43:46Z", "pid":134, "security_context":{}}, "po
d_labels":{"app":"nginx", "pod-template-hash":"5869d7778c"}, "workload":"nginx", "workload_kind":"Deployment"}, "docker":"ef7a20af440ae16f9897e2029c7eac9",
"parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjE1NjQ4NjAwMDAwMDA6Mjk4NDE=", "refcnt":1, "tid":728447, "in_init_tree":false}, "parent":{"exe
c_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjE1NjQ4NjAwMDAwMDA6Mjk4NDE=", "pid":29841, "uid":0, "cwd":"/run/containerd/io.containerd.runtime.v2.task
/k8s.io/a7ce165d04d5dff52f5986a8215a8b47bf5f65f126aeeafa5e02316d9d580734", "binary":"/usr/bin/containerd-shim-runc-v2", "arguments":"-namespace k8s.io -id a
7ce165d04d5dff52f5986a8215a8b47bf5f65f126aeeafa5e02316d9d580734 -address /run/containerd/containerd.sock", "flags":"procFS", "start_time":"2025-12-03T00:43:
42.401832708Z", "auid":4294967295, "parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjM0MDAwMDAwMDox", "tid":29841, "in_init_tree":false}, "fun
ction_name":"security_file_permission", "args":[{"file_arg":{"path":"/etc/testfile", "permission":"-rw-r--r--"}}, {"int_arg":2}], "action":"KPROBE_ACTION_PO
ST", "policy_name":"detect-file-tampering", "return_action":"KPROBE_ACTION_POST"}, "node_name":"aks-agentpool-14248651-vmss000001", "time":"2025-12-03T12:54
azureuser@cluster-access:~$  kubectl logs tetragon-477cv -n tetragon -c export-stdout --tail=500 | grep '"binary":"/usr/bin/bash"' | tail -3
{"process_exec":{"process":{"exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjQ4MzQyMTc0MzQ5MDUxOjc3NTg5Mg==", "pid":775892, "uid":0, "cwd":"/", "bin
ary":"/usr/bin/whoami", "flags":"execve rootcwd", "start_time":"2025-12-03T13:43:19.716182030Z", "auid":4294967295, "pod":{"namespace":"default", "name":"ng
inx-5869d7778c-6ltmm", "uid":"8ae7f498-dd2b-4a04-928a-dbf5ea3f4008", "container":{"id":"containerd://ef7a20af440ae16f9897e2029c7eac92dbaf19352bed6759fe35cde
22aa74f20", "name":"nginx", "image":{"id":"docker.io/library/nginx@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42", "name":"docker.
io/library/nginx:latest"}, "start_time":"2025-12-03T00:43:46Z", "pid":195, "security_context":{}}, "pod_labels":{"app":"nginx", "pod-template-hash":"5869d77
78c"}, "workload":"nginx", "workload_kind":"Deployment"}, "docker":"ef7a20af440ae16f9897e2029c7eac9", "parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNz
MDAwMDAxOjQ4MzQyMTcyNTkwNzg1Ojc3NTg5Mg==", "tid":775892, "in_init_tree":false}, "parent":{"exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjQ4MzQyMTc
yNTkwNzg1Ojc3NTg5Mg==", "pid":775892, "uid":0, "cwd":"/", "binary":"/usr/bin/bash", "arguments":"-c whoami", "flags":"execve rootcwd clone", "start_time":"2
025-12-03T13:43:19.714423723Z", "auid":4294967295, "pod":{"namespace":"default", "name":"nginx-5869d7778c-6ltmm", "uid":"8ae7f498-dd2b-4a04-928a-dbf5ea3f400
8", "container":{"id":"containerd://ef7a20af440ae16f9897e2029c7eac92dbaf19352bed6759fe35cde22aa74f20", "name":"nginx", "image":{"id":"docker.io/library/ngin
x@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42", "name":"docker.io/library/nginx:latest"}, "start_time":"2025-12-03T00:43:46Z", "
pid":195, "security_context":{}}, "pod_labels":{"app":"nginx", "pod-template-hash":"5869d7778c"}, "workload":"nginx", "workload_kind":"Deployment"}, "docker
":"ef7a20af440ae16f9897e2029c7eac9", "parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjE1NjQ4NjAwMDAwMDA6Mjk4NDE=", "tid":775892, "in_init_tre
e":false}}, "node_name":"aks-agentpool-14248651-vmss000001", "time":"2025-12-03T13:43:19.716181819Z", "node_labels":{"agentpool":"agentpool", "beta.kubernet
es.io/arch":"amd64", "beta.kubernetes.io/instance-type":"Standard_L2aos_v4", "beta.kubernetes.io/os":"linux", "failure-domain.beta.kubernetes.io/region":"no
rtheurope", "failure-domain.beta.kubernetes.io/zone":"northeurope-2", "kubernetes.azure.com/agentpool":"agentpool", "kubernetes.azure.com/azure-cni-overlay"
:"true", "kubernetes.azure.com/cluster":"MC_final_test_sec_northeurope", "kubernetes.azure.com/consolidated-additional-properties":"e0080963-cfdc-11f0-aace-
5e38850148e8", "kubernetes.azure.com/kubelet-identity-client-id":"82105cbd-ca27-41fe-9f30-ab604fc33298", "kubernetes.azure.com/kubelet-serving-ca":"cluster"
, "kubernetes.azure.com/localdns-state":"disabled", "kubernetes.azure.com/mode":"system", "kubernetes.azure.com/network-name":"final-vnet", "kubernetes.azur
e.com/network-resourcegroup":"final", "kubernetes.azure.com/network-stateless-cni":"false", "kubernetes.azure.com/network-subnet":"default", "kubernetes.azu
re.com/network-subscription":"8a24b207-6e52-4fb2-967a-773f648eed1a", "kubernetes.azure.com/node-image-version":"AKSUbuntu-2204gen2containerd-202511.07.0", "
kubernetes.azure.com/nodenetwork-vnetguid":"b035f07e-f0ea-492c-9e81-28adc465ee8d", "kubernetes.azure.com/nodepool-type":"VirtualMachineScaleSets", "kubernet
es.azure.com/os-sku":"Ubuntu", "kubernetes.azure.com/os-sku-effective":"Ubuntu2204", "kubernetes.azure.com/os-sku-requested":"Ubuntu", "kubernetes.azure.com
/podnetwork-type":"overlay", "kubernetes.azure.com/role":"agent", "kubernetes.azure.com/sku-cpu":"2", "kubernetes.azure.com/sku-memory":"16384", "kubernetes
.io/arch":"amd64", "kubernetes.io/hostname":"aks-agentpool-14248651-vmss000001", "kubernetes.io/os":"linux", "node.kubernetes.io/instance-type":"Standard_L2
aos_v4", "topology.disk.csi.azure.com/zone":"northeurope-2", "topology.kubernetes.io/region":"northeurope", "topology.kubernetes.io/zone":"northeurope-2"}}
{"process_kprobe":{"process":{"exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjQ4MzQyMTc0MzQ5MDUxOjc3NTg5Mg==", "pid":775892, "uid":0, "cwd":"/", "b
inary":"/usr/bin/whoami", "flags":"execve rootcwd", "start_time":"2025-12-03T13:43:19.716182030Z", "auid":4294967295, "pod":{"namespace":"default", "name":"
nginx-5869d7778c-6ltmm", "uid":"8ae7f498-dd2b-4a04-928a-dbf5ea3f4008", "container":{"id":"containerd://ef7a20af440ae16f9897e2029c7eac92dbaf19352bed6759fe35c
de22aa74f20", "name":"nginx", "image":{"id":"docker.io/library/nginx@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42", "name":"docke
r.io/library/nginx:latest"}, "start_time":"2025-12-03T00:43:46Z", "pid":195, "security_context":{}}, "pod_labels":{"app":"nginx", "pod-template-hash":"5869d
7778c"}, "workload":"nginx", "workload_kind":"Deployment"}, "docker":"ef7a20af440ae16f9897e2029c7eac9", "parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bX
NzMDAwMDAxOjQ4MzQyMTcyNTkwNzg1Ojc3NTg5Mg==", "refcnt":1, "tid":775892, "in_init_tree":false}, "parent":{"exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwM
DAxOjQ4MzQyMTcyNTkwNzg1Ojc3NTg5Mg==", "pid":775892, "uid":0, "cwd":"/", "binary":"/usr/bin/bash", "arguments":"-c whoami", "flags":"execve rootcwd clone", "
start_time":"2025-12-03T13:43:19.714423723Z", "auid":4294967295, "pod":{"namespace":"default", "name":"nginx-5869d7778c-6ltmm", "uid":"8ae7f498-dd2b-4a04-92
8a-dbf5ea3f4008", "container":{"id":"containerd://ef7a20af440ae16f9897e2029c7eac92dbaf19352bed6759fe35cde22aa74f20", "name":"nginx", "image":{"id":"docker.i
o/library/nginx@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42", "name":"docker.io/library/nginx:latest"}, "start_time":"2025-12-03
T00:43:46Z", "pid":195, "security_context":{}}, "pod_labels":{"app":"nginx", "pod-template-hash":"5869d7778c"}, "workload":"nginx", "workload_kind":"Deploym
ent"}, "docker":"ef7a20af440ae16f9897e2029c7eac9", "parent_exec_id":"YWtzLWFnZW50cG9vbC0xNDI0ODY1MS12bXNzMDAwMDAxOjE1NjQ4NjAwMDAwMDA6Mjk4NDE=", "tid":775892
```

Prometheus and Grafana deployment in EKS and AKS

Add the Prometheus helm repository, for that run the below command
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo update

create an exclusive namespace in the cluster for setting up the monitoring tools
kubectl create namespace monitoring

Install prometheus
helm install prometheus prometheus-community/prometheus -n monitoring
For EKS, installed Prometheus using the below command, where the PVC claim is disabled.
To accomplish customized promothesus-values.yaml is used

helm install prometheus prometheus-community/prometheus -n monitoring -f C:/Users/bala/prometheus-eks-values.yaml

Verify promtheus pod are up and running
kubectl get pods -n monitoring
kubectl get svc -n monitoring

Deploy Grafana
Add the Grafana helm repository

helm repo add grafana https://grafana.github.io/helm-charts
helm repo update

create a grafana configuration file with username and password, and pointing prometheus as data source in the file and install that grafana with that file.

First generate a password using a openssl
openssl rand -base64 32
Note down the password, and place it in the YAML file.
Create a file named grafana-values.yaml and grafana-eks-values.yaml for AKS and EKS respectively
Install grafana using the file we generated, which is stored in my local. Already the both the cluster kubeconfig file is loaded in my local machine, so, it will have access to the respective clusters.

Run the below command for AKS
helm install grafana grafana/grafana -n monitoring -f C:/Users/bala/grafana-values.yaml

For EKS run the below command
helm install grafana grafana/grafana -n monitoring -f C:/Users/bala/grafana-eks-values.yaml

verify the grafana installation
kubectl get pods -n monitoring | grep grafana
kubectl get svc grafana -n monitoring

Access the grafana UI by running the below command
Run the below command from local command prompt to port-forward to the cluster, port 3000 is from local machine to the port 80, where the grafana running
kubectl port-forward -n monitoring svc/grafana 3000:80

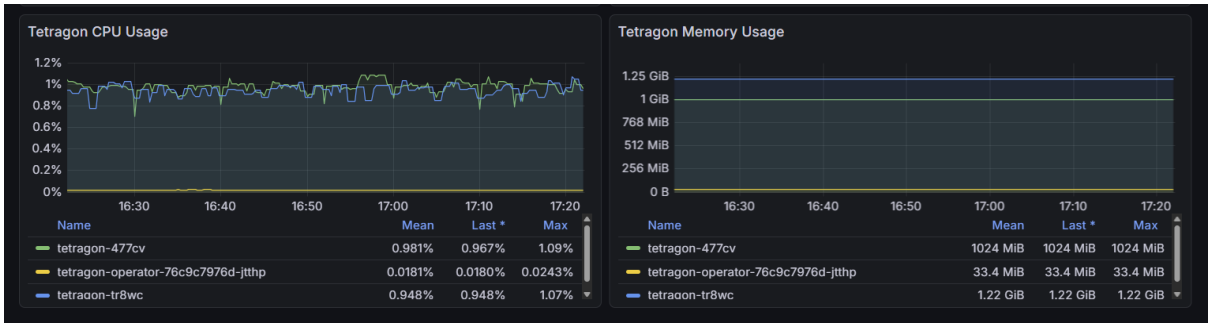
Access the grafana by running the below command in browser
http://localhost:3000

For EKS use the below login credentials
Username: admin
Password:X1LomNCQMwTMfBiXgyx5WBIHw7LAshNa6QTyzTi

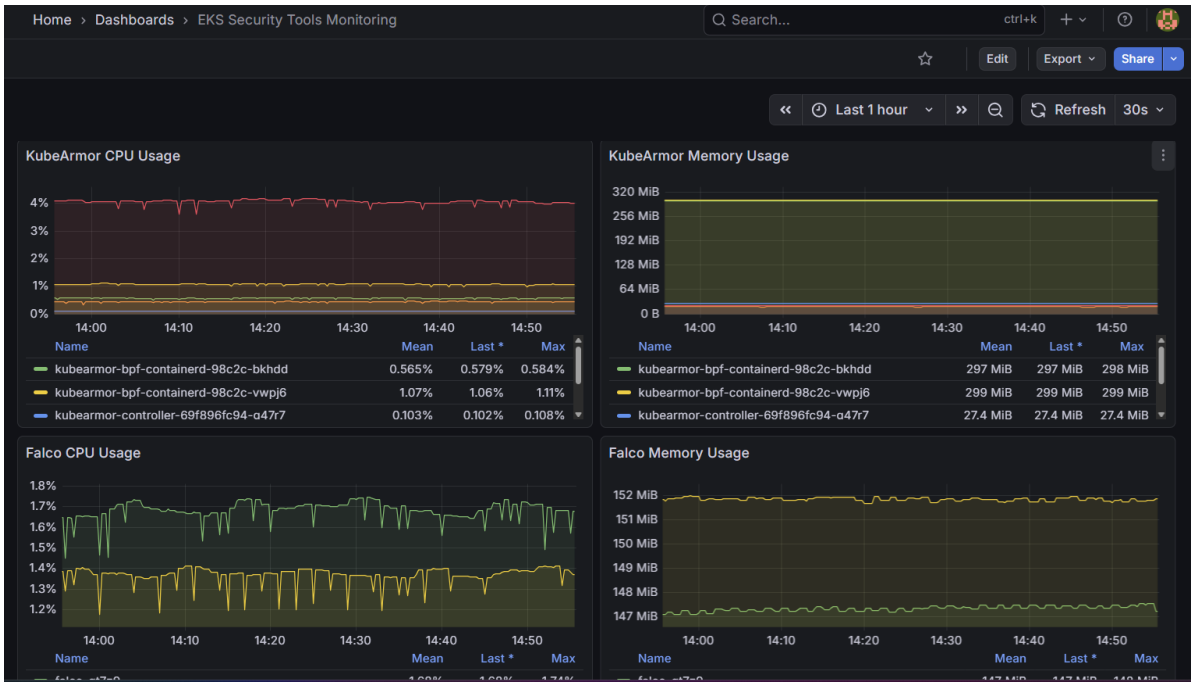
To view the resource utilization of the respective security tools in each cluster, on grafana dashboard click, import dashboard and load the already configured JSON files like C:\Users\bala\security-tools-dashboard-eks.json and C:\Users\bala\security-tools-dashboard.json` for EKS and AKS respectively.

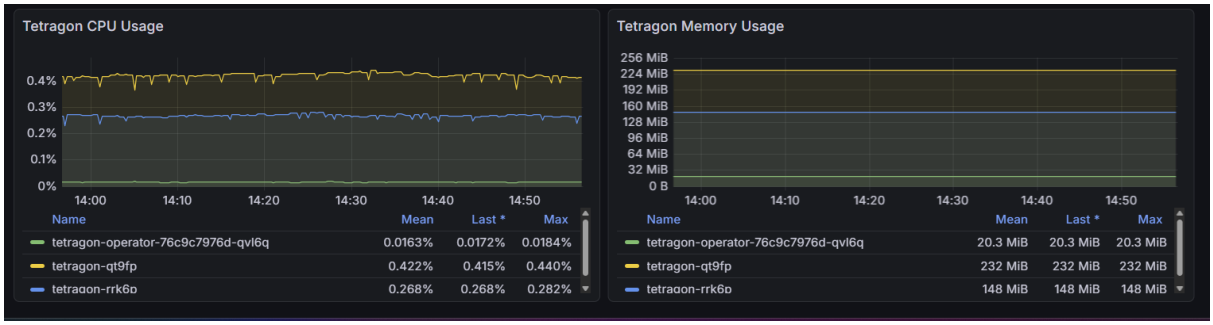
Now the dashboard will reflect the resource utilization of security tools like Falco, Kube Armor, and tetragon in the UI.

Security tools resource utilization in AKS cluster



Security tools resource utilization in EKS cluster





References:

For creating AWS EKS cluster - https://docs.aws.amazon.com/eks/latest/userguide/clusters.html

For creating Azure AKS cluster – https://learn.microsoft.com/en-us/azure/aks/what-is-aks

For Kube Armor implementation - https://docs.kubearmor.io/kubearmor/quick-links/deployment_guide

For Tetragon implementation - https://tetragon.io/docs/getting-started/install-k8s/

For Falco implementation – https://falco.org/docs/getting-started/falco-kubernetes-quickstart/