Technical Guidelines:

Set-up Minikube:

Minikube is local kubernetes which we are going to use. Following is the snip of all the commands used to start minikube over an aws ec2 instance.

```
[root@ip-172-31-10-132 ec2-user]# curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 69.2M  100 69.2M    0     0  46.3M      0  0:00:01  0:00:01 --:--:-- 46.3M
[root@ip-172-31-10-132 ec2-user]# chmod +x minikube
[root@ip-172-31-10-132 ec2-user]# sudo mv minikube /usr/local/bin/
[root@ip-172-31-10-132 ec2-user]# yum install conntrack -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package conntrack-tools.x86_64 0:1.4.4-5.amzn2.2 will be installed
--> Processing Dependency: libnetfilter_cttimeout.so.1(LIBNETFILTER_CTTIMEOUT_1.1)(64bit) for package: conntrack-tools-1.4.4-5
mzn2.2.x86_64
--> Processing Dependency: libnetfilter_cttimeout.so.1(LIBNETFILTER_CTTIMEOUT_1.0)(64bit) for package: conntrack-tools-1.4.4-5
mzn2.2.x86_64
--> Processing Dependency: libnetfilter_cthelper.so.0(LIBNETFILTER_CTHELPER_1.0)(64bit) for package: conntrack-tools-1.4.4-5.a
n2.2.x86_64
--> Processing Dependency: libnetfilter_queue.so.1()(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Processing Dependency: libnetfilter_cttimeout.so.1()(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Processing Dependency: libnetfilter_cthelper.so.0()(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Running transaction check
---> Package libnetfilter_cthelper.x86_64 0:1.0.0-10.amzn2.1 will be installed
---> Package libnetfilter_cttimeout.x86_64 0:1.0.0-6.amzn2.1 will be installed
---> Package libnetfilter_queue.x86_64 0:1.0.2-2.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                                Arch                Version                                Repository                Siz
================================================================================================================================
Installing:
```

As docker is required so I installed docker for minikube which is there in below figure.

```
[root@ip-172-31-10-132 ec2-user]# yum install docker -y
```

```
[root@ip-172-31-10-132 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-10-132 ec2-user]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2022-02-26 14:28:02 UTC; 24s ago
     Docs: https://docs.docker.com
  Process: 2781 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 2780 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 2784 (dockerd)
    Tasks: 8
   Memory: 37.8M
   CGroup: /system.slice/docker.service
           └─2784 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.205652450Z" level=info msg="Client...gr
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.246173547Z" level=warning msg="You...gh
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.246557068Z" level=warning msg="You...ic
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.247000559Z" level=info msg="Loadin...rt
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.392595245Z" level=info msg="Defaul...es
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.440904897Z" level=info msg="Loadin...ne
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.458501346Z" level=info msg="Docker...10
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.458605196Z" level=info msg="Daemon...io
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 26 14:28:02 ip-172-31-10-132.ec2.internal dockerd[2784]: time="2022-02-26T14:28:02.482863420Z" level=info msg="API li...oc
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-10-132 ec2-user]# curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 69.2M  100 69.2M    0     0  46.3M      0  0:00:01  0:00:01 --:--:-- 46.3M
```

Kubectl:

We are using *kubectl* as command line tool for running commands .So Installing kubectl using the following commands:

```
[root@ip-172-31-10-132 ec2-user]# curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.20.4/2021-04-12/bin/linux/am
4/kubectl
chmod +x ./kubectl
mkdir -p $HOME/bin
cp ./kubectl $HOME/bin/kubectl
export PATH=$HOME/bin:$PATH
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
source $HOME/.bashrc
kubectl version --short --client  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 38.3M  100 38.3M    0     0  19.1M      0  0:00:02  0:00:02 --:--:-- 19.1M
[root@ip-172-31-10-132 ec2-user]# chmod +x ./kubectl
[root@ip-172-31-10-132 ec2-user]# mkdir -p $HOME/bin
[root@ip-172-31-10-132 ec2-user]# cp ./kubectl $HOME/bin/kubectl
[root@ip-172-31-10-132 ec2-user]# export PATH=$HOME/bin:$PATH
[root@ip-172-31-10-132 ec2-user]# echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
[root@ip-172-31-10-132 ec2-user]# source $HOME/.bashrc
[root@ip-172-31-10-132 ec2-user]# kubectl version --short --client
Client Version: v1.20.4-eks-6b7464
```

Deploying the application:

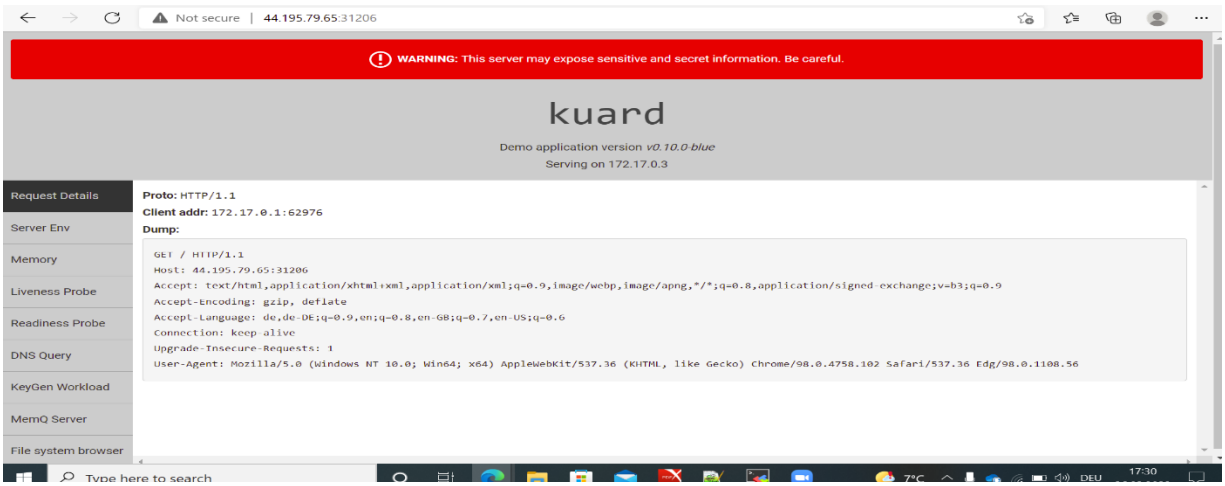
After setting up cluster and cli tools I am deploying the demo kuard app over the minikube using the following command which results in initializing a pod name kuard.

```
[root@ip-172-31-10-132 ~]# kubectl run --restart=Never --image=gcr.io/kuar-demo/kuard-amd64:blue kuard
pod/kuard created
[root@ip-172-31-10-132 ~]#
```

Expose pod: For exposing the pod outside the cluster I am exposing it using nodeport service after which we can access it over internet.

```
[root@ip-172-31-10-132 ~]# kubectl expose pod kuard --port 8080 --type NodePort
service/kuard exposed
[root@ip-172-31-10-132 ~]# kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kuard        NodePort    10.96.232.221   <none>        8080:31206/TCP   5s
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          119m
[root@ip-172-31-10-132 ~]#
```

The figure demonstrating the exposed service is attached below:



Helm Set-up:

We are going to use helm charts to setup Prometheus and Grafana so setting up helm is first priority.

```
[root@ip-172-31-10-132 ~]# curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 > get_helm.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11156  100 11156    0     0   647k      0 --:--:-- --:--:-- --:--:--  680k
[root@ip-172-31-10-132 ~]# chmod 700 get_helm.sh
[root@ip-172-31-10-132 ~]# ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.8.0-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
[root@ip-172-31-10-132 ~]# helm version --short
v3.8.0+gd141386
[root@ip-172-31-10-132 ~]#
```

Above fig. consists of all the steps for helm.

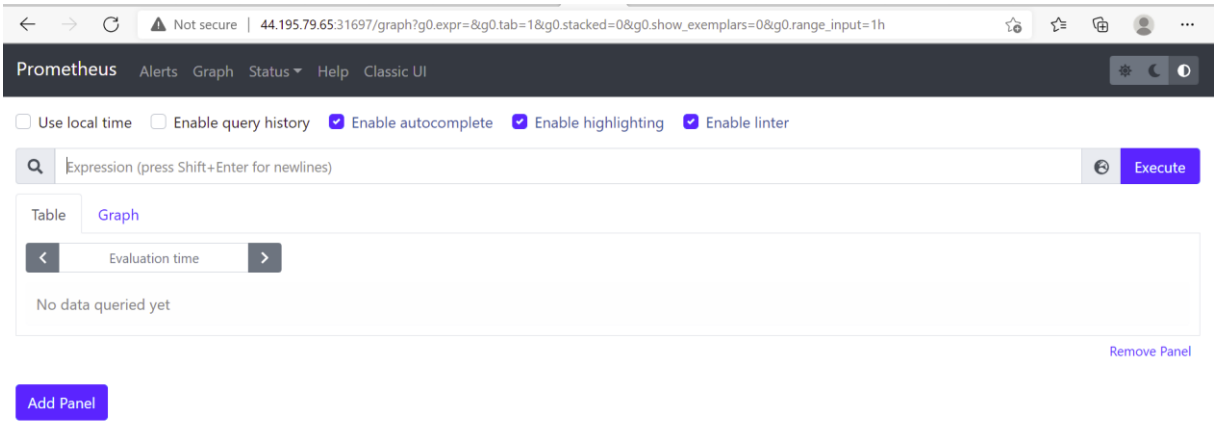
Installing Prometheus:

We are using Prometheus for exporting the metrics of cluster. Using following steps, we are setting up Prometheus along with exporter and alert manager.

```
[root@ip-172-31-10-132 ~]# helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" already exists with the same configuration, skipping
[root@ip-172-31-10-132 ~]# helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" already exists with the same configuration, skipping
[root@ip-172-31-10-132 ~]# helm install prometheus prometheus-community/prometheus
NAME: prometheus
LAST DEPLOYED: Sat Feb 26 16:50:36 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.default.svc.cluster.local


Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app=prometheus,component=server" -o jsonpath="{.items[0].metadata
.name}")
  kubectl --namespace default port-forward $POD_NAME 9090


The Prometheus alertmanager can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-alertmanager.default.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app=prometheus,component=alertmanager" -o jsonpath="{.items[0].me
tadata.name}")
  kubectl --namespace default port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been moved to a global property.  #####
```

Prometheus Dashboard:

After setting up and exposing Prometheus using node port following dashboard is there:



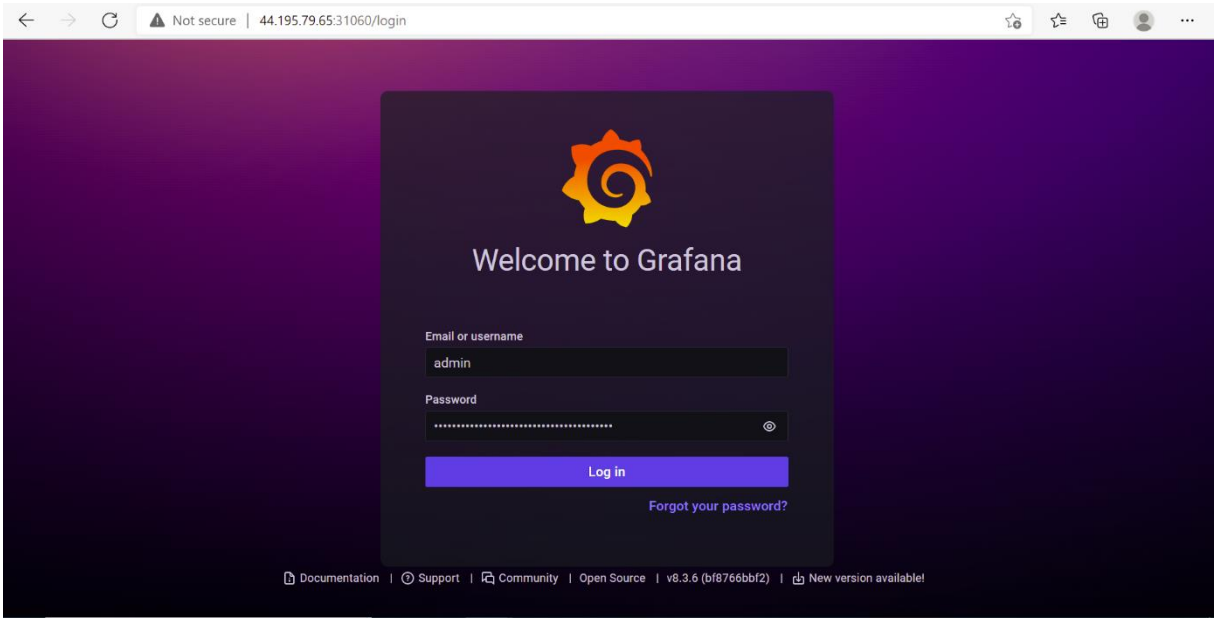
Grafana setup:

We are going to use Grafana as UI for metrics exported from Prometheus. We are using helm chart to install Grafana. Following commands are used to setup the Grafana in minikube:

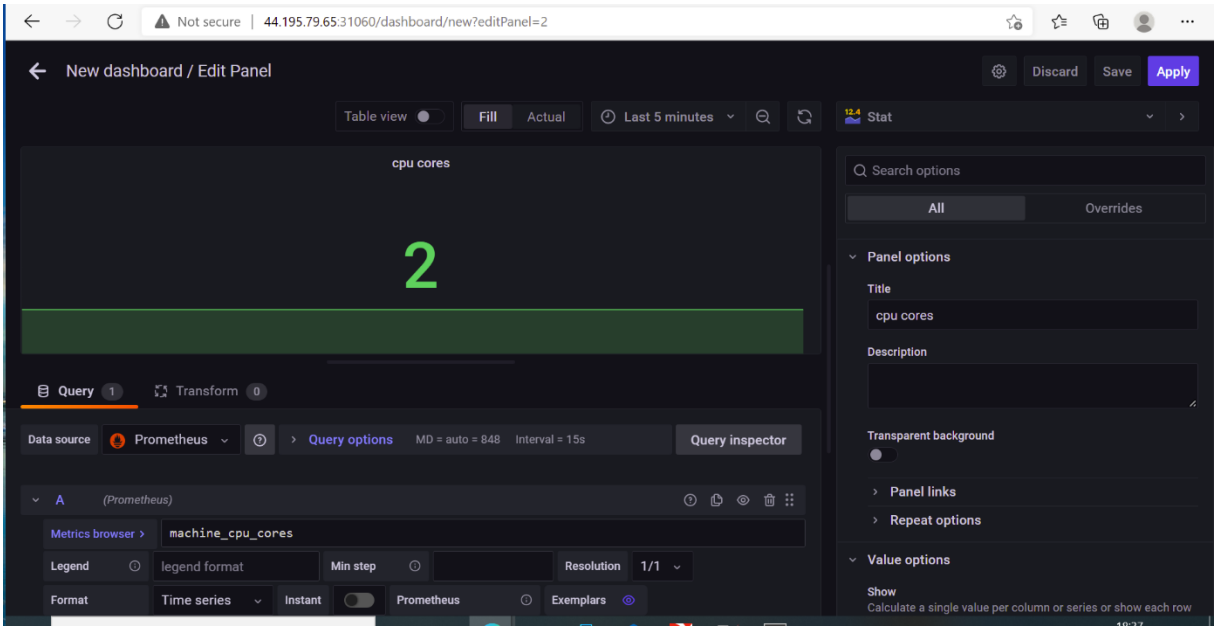
```
[root@ip-172-31-10-132 ~]# helm repo add grafana https://grafana.github.io/helm-charts
"grafana" already exists with the same configuration, skipping
[root@ip-172-31-10-132 ~]# helm install grafana grafana/grafana
W0226 16:57:54.228969   29751 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W0226 16:57:54.230757   29751 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W0226 16:57:54.276830   29751 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W0226 16:57:54.277342   29751 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
NAME: grafana
LAST DEPLOYED: Sat Feb 26 16:57:53 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

   kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:

   grafana.default.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:

     export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=graf
ana" -o jsonpath="{.items[0].metadata.name}")
     kubectl --namespace default port-forward $POD_NAME 3000

3. Login with the password from step 1 and the username: admin
#################################################################################
```

Grafana Dashboard:

Once we expose Grafana over internet following dashboard is there.



Metrics in grafana: We can setup a dashboard in which we can define metrics so that grafana will fetch the metrics from prometheus .Following snap is for query resulting in number of cpu cores.



Alerting in grafana: we can create any of the alerting methods for threshold values such as mail, slack notifications, and telegram bots.