```
++++++++++++++++++++
Kubernetes Monitoring
++++++++++++++++++++++

-> Prometheus is an open-source systems monitoring and alerting toolkit

-> Prometheus collects and stores its metrics as time series data

-> It provides out-of-the-box monitoring capabilities for the k8s container orchestration platform.


-> Grafana is a  analysis and monitoring tool

-> Grafana is a multi-platform open source analytics and interactive visualization web application.

-> It provides charts, graphs, and alerts for the web when connected to supported data sources.

-> Grafana allows you to query, visualize, alert on and understand your metrics no matter where they
are stored. Create, explore and share dashboards.


Note: Graphana will connect with Prometheus for data source.


########################################
How to deploy Grafana & Prometheus in K8S
########################################

-> Most Efficient way is using Helm Chart to deploy Prometheus Operator


#########################
Install Prometheus & Grafana
#########################

# Add the latest helm repository in Kubernetes
$ helm repo add stable https://charts.helm.sh/stable

# Add prometheus repo to helm
$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

# Update Helm Repo
$ helm repo update

# Search Repo
$ helm search repo prometheus-community

# install prometheus
$ helm install stable prometheus-community/kube-prometheus-stack

# Get all pods
$ kubectl get pods

Node: You should see prometheus pods running

# Check the services
$ kubectl get svc

# By default prometheus and grafana service is available within the cluster using ClusterIP, to
access them outside lets change it either NodePort or Loadbalancer.

$ kubectl edit svc stable-kube-prometheus-sta-prometheus

# Now edit the grafana service
```

```
$ kubectl edit svc stable-grafana


# Verify the service if changed to LoadBalancer
$ kubectl get svc


To access Prometheus web interface copy Loadbalancer URL and port number 9090

To access Grafana web interface copy Loadbalancer URL and port number 80

UserName: admin
Password: prom-operator


##########
ELK Stack
##########

-> The ELK Stack is a collection of three open-source products â€” Elasticsearch, Logstash, and
Kibana

-> ELK stack provides centralized logging in order to identify problems with servers or applications

-> It allows you to search all the logs in a single place


E stands for : Elastic Search --> It is used to store logs

L stands for : Log Stash --> It is used for processing logs

K stands for : Kibana --> It is an visualization tool


FileBeat : Log files

MetricBeat : Metrics

PacketBeat : Network data

HeartBeat : Uptime Monitoring


->  Filebeat collect data from the log files and sends it to logstash

-> Logstash enhances the data and sends it to Elastic search

-> Elastic search stores and indexes the data

-> Kibana displays the data stored in Elastic Search based on the request recieved


############################
ELK Installation using HELM
############################


Pre-requisites :

EKS Cluster
Nodes : 4 GB RAM
Client Machine with kubectl & helm configured

$ kubectl create ns efk
```

```
$ kubectl get ns

$ helm ls

$ helm repo add elastic https://helm.elastic.co

$ helm repo ls

$ helm show values elastic/elasticsearch >> elasticsearch.values

$ vi elasticsearch.values

-> replicas as 1 & masternodes as 1

$ helm install elasticsearch elastic/elasticsearch -f elasticsearch.values -n efk

$ helm ls -n efk

$ kubectl get all -n efk

$ helm show values elastic/kibana >> kibana.values

$ vi kibana.values

-> Change Service Type from ClusterIP to LoadBalancer

-> Change Port to 80  (default port is 5601)

$ helm install kibana elastic/kibana -f kibana.values -n efk

$ kubectl get all -n efk

$ helm install filebeat elastic/filebeat -n efk

$ helm install metricbeat elastic/metricbeat -n efk

Note: Access Kibana using Load Balancer DNS
```