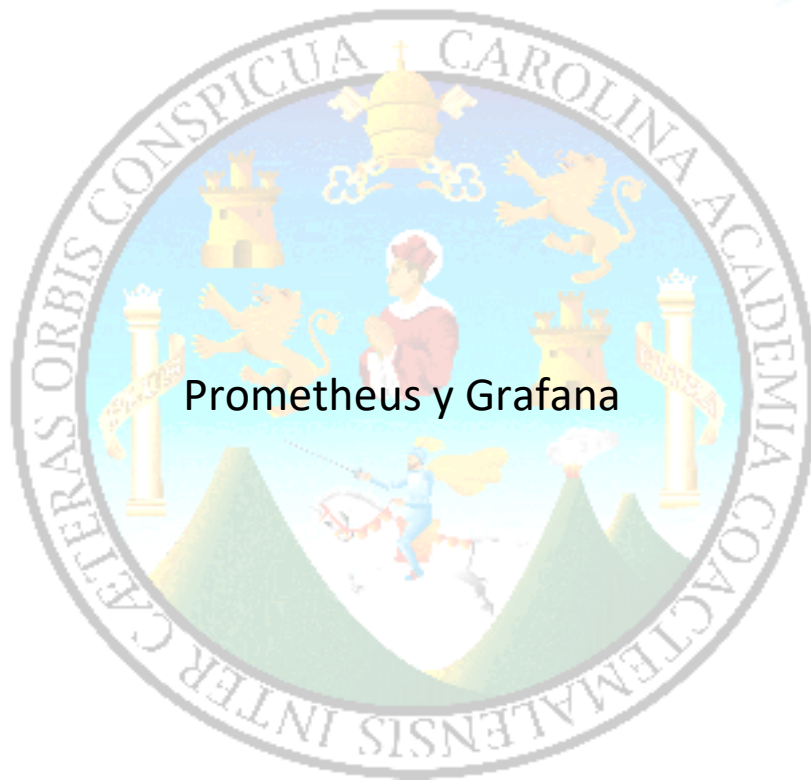


Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Sistemas Operativos 2
Sección O



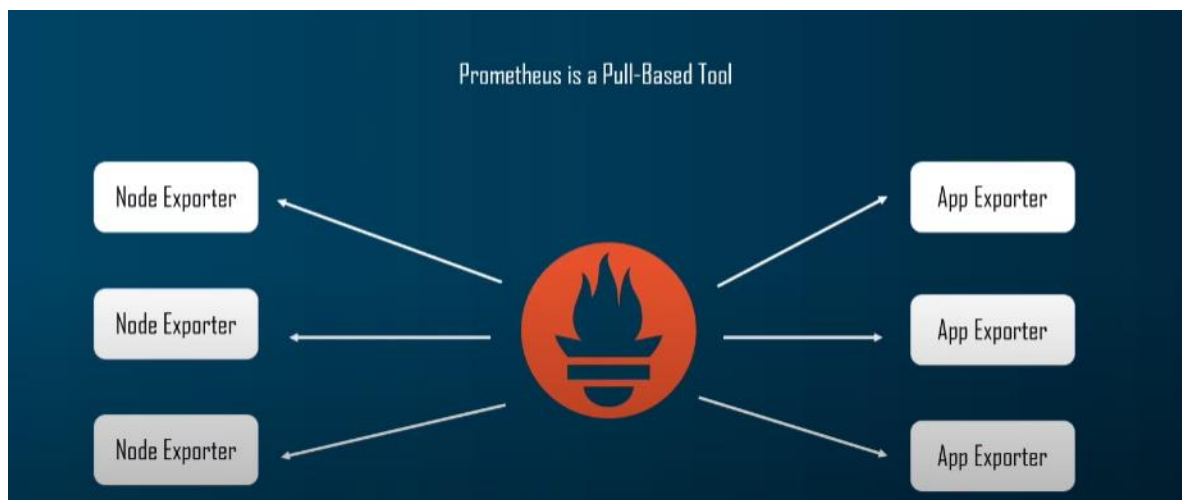
Prometheus y Grafana

Grupo 26

Prometheus

Prometheus es un conjunto de herramientas de monitoreo y alerta de sistemas de código abierto construido originalmente en SoundCloud. Desde su inicio en 2012, muchas empresas y organizaciones han adoptado Prometheus, y el proyecto cuenta con una comunidad de usuarios y desarrolladores muy activa. Ahora es un proyecto de código abierto independiente y se mantiene independientemente de cualquier empresa. Para enfatizar esto y aclarar la estructura de gobierno del proyecto, Prometheus se unió a la *Cloud Native Computing Foundation* en 2016 como el segundo proyecto alojado, después de Kubernetes.

Prometheus recopila y almacena sus métricas como datos de series de tiempo, es decir, la información de las métricas se almacena con la marca de tiempo en la que se registró, junto con pares opcionales de clave-valor denominados etiquetas.

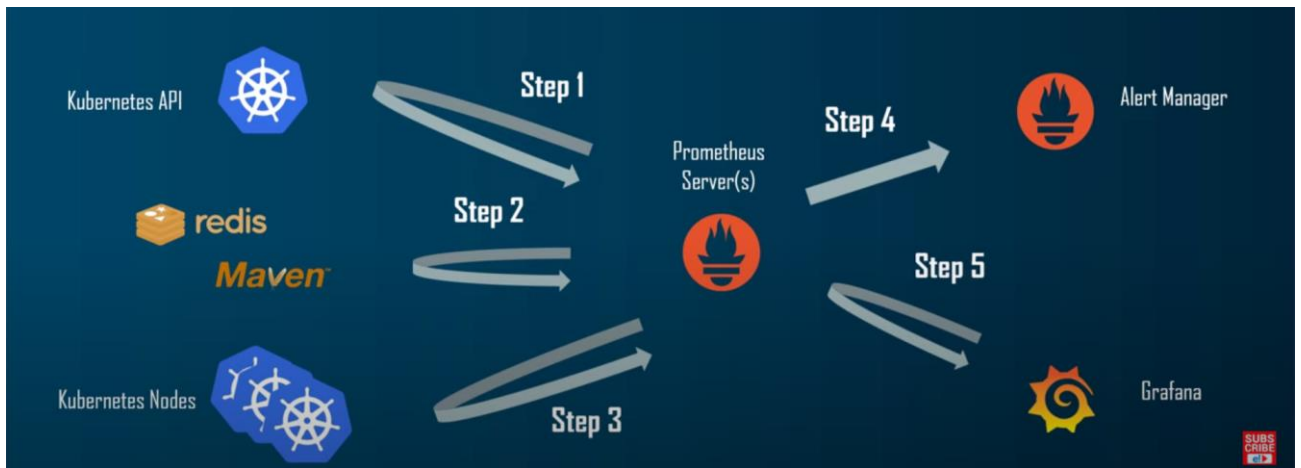


Grafana

Grafana permite consultar, visualizar, alertar y comprender sus métricas sin importar dónde estén almacenadas. Cree, explore y comparta hermosos cuadros de mando con su equipo y fomente una cultura basada en datos.



Arquitectura del monitoreo en Kubernetes



Instalación y configuración

1. Utilizando archivos de manifiesto.
 - a. Se configura los pods.

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: prometheus
rules:
- apiGroups: [""]
  resources:
    - nodes
    - nodes/metrics
    - services
    - endpoints
    - pods
  verbs: ["get", "list", "watch"]
- apiGroups: [""]
  resources:
    - configmaps
  verbs: ["get"]
- apiGroups:
    - networking.k8s.io
  resources:
    - ingresses
  verbs: ["get", "list", "watch"]
- nonResourceURLs: ["/metrics"]
  verbs: ["get"]
```

- b. Se especifica un servicio para acceder a Prometheus

```
apiVersion: v1      You, seconds ag
kind: Service
metadata:
  name: prometheus
  labels:
    app: prometheus
spec:
  ports:
    - name: web
      port: 9090
      targetPort: web
  selector:
    app: prometheus
  sessionAffinity: ClientIP
```

- c. Configuración de para el monitoreo

```
You, seconds ago | 1 author (You)
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: prometheus-self
  labels:
    app: prometheus
spec:
  endpoints:
    - interval: 30s
      port: web
  selector:
    matchLabels:
      app: prometheus
```

2. Ejecución de comandos se estará utilizando Helm.

- a. Se debe de crear un *namespace* para los pods de Prometheus.

```
kubectl create namespace prometheus
```

- b. Instalar el monitor de Helm por medio del repositorio.

```
helm repo add monitor https://charts.helm.sh/stable
```

- c. Actualizar repositorio

```
helm repo update
```

- d. instalar prometheus operator

```
helm install prometheus monitor/prometheus-operator --namespace prometheus
```

- e. Ver los pods de Prometheus

```
kubectl get pods -n Prometheus
```

- f. Iniciar Prometheus

```
kubectl port-forward -n prometheus [select name pod] 9090
```

- g. Obtener las credenciales para Grafana (admin-password, admin-user)

```
kubectl get secret --namespace prometheus prometheus-grafana -o yaml
```

- h. Decodificar las credenciales (aplica para Linux, se puede utilizar cualquier decodificador de base 64).

```
echo "[user]" | base64 --decode
```

```
echo "[password]" | base64 --decode
```

- i. Iniciar Grafana

```
kubectl port-forward -n prometheus [select name pod] 3000
```

Dashboard de Grafana



Se selecciona el *datasource* Prometheus y la instancia que se utiliza.

En esta captura se muestra el monitoreo de los Nodos que hay en Kubernetes.