GCP Prometheus + Grafana Full Setup Guide (Beginner Friendly)

---

1. Create & Configure Your GCP Project

1.1 Create a New GCP Project

1. Go to **Google Cloud Console**.
2. Click the **Project dropdown** → **New Project**.
3. Enter a name like `prometheus-monitoring-demo`.
4. Click **Create** and **switch to it**.

1.2 Set Active Project in Cloud Shell

```
gcloud config set project prometheus-monitoring-demo
```

Why: This ensures all future commands apply to this project.

Check active project:

```
gcloud config get-value project
```

---

2. Enable Required GCP APIs

Run the following:

```
gcloud services enable \
  container.googleapis.com \
  monitoring.googleapis.com \
  monitoringdata.googleapis.com \
  cloudresourcemanager.googleapis.com
```

Why these APIs?

- **container.googleapis.com** → Needed to create GKE clusters.
- **monitoring.googleapis.com** → Used for Cloud Monitoring.
- **monitoringdata.googleapis.com** → Enables Prometheus metrics ingestion.
- **cloudresourcemanager.googleapis.com** → Allows managing project resources.

---

3. Create a GKE Cluster

3.1 Create a Cluster

```
gcloud container clusters create prometheus-demo \
  --zone us-central1-a \
  --num-nodes 2
```

What happens: A 2-node Kubernetes cluster launches.

3.2 Connect kubectl to Cluster

```
gcloud container clusters get-credentials prometheus-demo --zone us-central1-
a
kubectl get nodes
```

Why: Provides Kubernetes access for deployments.

---

4. Create Kubernetes Namespaces

```
kubectl create namespace monitoring
kubectl create namespace grafana
```

Why: Organizes resources cleanly.

---

5. Install Google Managed Service for Prometheus (GMP)

Apply setup manifests:

```
kubectl apply -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/manifests/setup.yaml
kubectl apply -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/manifests/operator.yaml
```

Check pods:

```
kubectl get pods -n gmp-system
```

Why: GMP manages Prometheus scraping + metric ingestion.

---

6. Deploy Sample App for Metrics

Apply example app:

```
kubectl -n monitoring apply -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/examples/example-app.yaml
```

Check:

```
kubectl -n monitoring get pods
```

6.1 Check App Logs

```
kubectl -n monitoring logs deployment/example-app
```

Why: Confirms app is running and exposing metrics.

---

7. Configure Prometheus Scraping Using PodMonitoring

Apply PodMonitoring:

```
kubectl -n monitoring apply -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/examples/pod-monitoring.yaml
```

Check resources:

```
kubectl get podmonitoring -A
```

Why: Tells Prometheus operator which pod metrics to scrape.

---

8. Install Grafana on Kubernetes

8.1 Create Grafana Deployment File

```yaml
apiVersion: v1
kind: Service
metadata:
  name: grafana
  namespace: grafana
spec:
  type: LoadBalancer
  selector:
```

```
    app: grafana
  ports:
    - port: 3000
      targetPort: 3000
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grafana
  namespace: grafana
spec:
  replicas: 1
  selector:
    matchLabels:
      app: grafana
  template:
    metadata:
      labels:
        app: grafana
    spec:
      containers:
      - name: grafana
        image: grafana/grafana:latest
        ports:
        - containerPort: 3000
```

Apply:

```
kubectl apply -f grafana-deployment.yaml
```

Get IP:

```
kubectl -n grafana get svc grafana
```

Open in browser:

```
http://<EXTERNAL-IP>:3000
```

Default login: **admin / admin**

---

9. Connect Grafana to Managed Prometheus

9.1 Get Bearer Token

```
gcloud auth print-access-token
```

9.2 Add Prometheus Data Source in Grafana

- Go to **Configuration → Data Sources → Add Data Source → Prometheus**

- URL:

`https://monitoring.googleapis.com/v1/projects/<PROJECT_ID>/location/global/prometheus`

- Add header:

`Authorization: Bearer <TOKEN_FROM_GCLOUD>`

Why: Prometheus API is protected by Google IAM.

---

10. Query Metrics (PromQL)

10.1 In Grafana Explore

Try:

```
up
example_requests_total
rate(example_requests_total[5m])
```

10.2 Query Prometheus via curl

Instant query:

```
curl -s https://monitoring.googleapis.com/v1/projects/prometheus-monitoring-demo/location/global/prometheus/api/v1/query \
  -d 'query=up' \
  -H "Authorization: Bearer $(gcloud auth print-access-token)"
```

Range query:

```
curl -s https://monitoring.googleapis.com/v1/projects/prometheus-monitoring-demo/location/global/prometheus/api/v1/query_range \
  -d 'query=rate(example_requests_total[5m])' \
  -d 'start=2025-11-18T00:00:00Z' \
  -d 'end=2025-11-19T00:00:00Z' \
  -d 'step=60' \
  -H "Authorization: Bearer $(gcloud auth print-access-token)"
```

---

11. Getting Logs

11.1 Kubernetes Logs

`kubectl -n monitoring logs <pod-name>`

Follow logs:

```
kubectl -n monitoring logs <pod> -f
```

11.2 Cloud Logging (Centralized)

```
gcloud logging read "resource.type=k8s_container AND
resource.labels.namespace_name=monitoring" --limit 100
```

---

12. Teardown / Stop All Services (Avoid Billing)

**12.1 Delete App & Monitoring

```
kubectl -n monitoring delete -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/examples/example-app.yaml
kubectl -n monitoring delete -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/examples/pod-monitoring.yaml
kubectl delete namespace monitoring
```

12.2 Delete Grafana

```
kubectl delete namespace grafana
```

12.3 Remove GMP Operator

```
kubectl delete -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/manifests/operator.yaml
kubectl delete -f
https://raw.githubusercontent.com/GoogleCloudPlatform/prometheus-
engine/v0.15.3/manifests/setup.yaml
```

12.4 Delete GKE Cluster

```
gcloud container clusters delete prometheus-demo --zone us-central1-a --quiet
```

12.5 Disable APIs (Optional)

```
gcloud services disable container.googleapis.com monitoring.googleapis.com
monitoringdata.googleapis.com
```

12.6 Delete Project (Ultimate Cleanup)

```
gcloud projects delete prometheus-monitoring-demo
```

- Only do this if you want everything wiped completely.*
