Kubernetes Deployment

1. Kubernetes Cluster Setup (AWS EKS)

guide to provision an EKS cluster:

1. Configure AWS CLI and install `eksctl`, `kubectl`.

2. Create EKS Cluster:

eksctl create cluster --name --region --nodes 2 --node-type t3.small

3. Validate cluster and node status using `kubectl get nodes`.

4. Upgrade from Kubernetes v1.27 to v1.28:

- eksctl upgrade cluster --name --version 1.28 --approve

- Create new nodegroup if upgrade fails.

- Drain and delete older nodegroup safely using `kubectl drain` and `eksctl delete nodegroup`.

2. Deploy NGINX Web + MySQL Application

1. Create app namespace and deployment YAMLs:

- Deploy NGINX using `nginx-deployment.yml`

- Deploy MySQL using `mysql-deployment.yml`

2. Validate using:

- kubectl get pods -n app-demo

- kubectl get svc -n app-demo

3. Troubleshooting:

- ImagePullBackOff: check image repo.

- CrashLoopBackOff: validate config/env.

3. Monitoring with Prometheus + Grafana

1. Create namespace: `kubectl create ns monitoring`

2. Apply configs:

- prometheus-config.yml (scrape settings)

- prometheus-deploy.yml (Deployment + Service)

- grafana-deploy.yml (Expose via NodePort)

- node-exporter-deploy.yml & kube-state-metrics.yml

3. Access via port-forward:

kubectl port-forward svc/grafana-service 3000:3000 -n monitoring

4. Grafana login: admin/admin ® Add Prometheus data source.

5. Validate dashboards and target status.

Troubleshooting Summary

- Prometheus no target:

® Missing annotations or relabel\_configs mismatch.

® Wrong port name or missing `metrics` in containerPort.

- Grafana shows N/A:

® Ensure targets are scraping data (via Prometheus > Targets).

® Restart Grafana and Prometheus to reload config.

- Permission Error:

® Grant clusterRole to serviceAccount used by Prometheus to list pods.

Workaround:

- Patch annotations to node-exporter DaemonSet using `kubectl patch`

- Validate relabel\_configs for matching labels and ports

# Monitoring Setup Summary: Kubernetes, Prometheus, Grafana

## Summaries

### Kubernetes Cluster Setup

* - Deployed a Kubernetes cluster (2 worker nodes).
* - Installed required tools: kubectl, container runtime, CNI plugin.
* - Verified all nodes are in Ready state using `kubectl get nodes`.

### Application + Monitoring Stack Deployment

* - Deployed MySQL and NGINX apps in the cluster.
* - Deployed Prometheus using `prometheus-combined-deploy.yml`.
* - Deployed node-exporter, kube-state-metrics, and nginx-exporter with correct ports and annotations.
* - Created ConfigMap for Prometheus scrape\_configs including targets: node-exporter, kube-state-metrics, kubernetes-pods.

### Grafana Visualization

* - Deployed Grafana using `grafana-deploy.yml`.
* - Configured Prometheus as datasource.
* - Imported dashboards: Kubernetes Cluster (Prometheus), Node Exporter Full, and NGINX Exporter.
* - Verified Grafana displayed data successfully for node-exporter, kube-state-metrics, and application metrics.

## Troubleshooting Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Symptoms | Root Cause | Workaround |
| No data in Grafana | Prometheus target not scraped | Missing pod annotations or relabel\_configs in prometheus.yml | Added `prometheus.io/scrape`, `port`, and `path` annotations to pods |
| Prometheus shows 'no target' | ServiceAccount does not have permission to list pods | Default service account lacks RBAC permissions | Applied `prometheus-rbac.yml` to grant correct roles |
| ConfigMap failed to apply | Invalid YAML format | Used wrong indentation or invalid field (`scrape\_configs` inside metadata) | Moved `scrape\_configs` into data.prometheus.yml properly |
| Kube-state-metrics ImagePullBackOff | Container image not found | Incorrect image path or version | Updated image to `k8s.gcr.io/kube-state-metrics/kube-state-metrics:v2.9.2` |

## Kubernetes Upgrade: v1.27 to v1.28

* - Backup current cluster configuration and etcd snapshots.
* - Drain and cordon all worker nodes: `kubectl drain <node> --ignore-daemonsets`.
* - Upgrade kubeadm on the control plane node: `apt install -y kubeadm=1.28.x`.
* - Run `kubeadm upgrade plan` to verify upgrade path.
* - Apply the upgrade: `kubeadm upgrade apply v1.28.x`.
* - Upgrade kubelet and kubectl on control plane node: `apt install -y kubelet=1.28.x kubectl=1.28.x`.
* - Restart kubelet: `systemctl restart kubelet`.
* - Repeat upgrade steps on each worker node (kubeadm, kubelet, kubectl).
* - Uncordon all nodes: `kubectl uncordon <node>`.