**Objective: Equip students with monitoring tools for EKS.**

**Tasks:**

1. Integrate EKS with AWS CloudWatch.

2. Set up Prometheus and Grafana.

3. Create dashboards and alerts.

**Documentation:**

- Importance of monitoring.

- AWS CloudWatch insights.

- Prometheus in the Kubernetes ecosystem

**Importance of Monitoring:**

Monitoring systems and applications is fundamental to ensure they are working as expected. By monitoring, you can:

1. **Detect Issues Early:** Before they impact users or escalate into bigger problems.
2. **Optimize Performance:** By identifying bottlenecks or inefficiencies in the system.
3. **Capacity Planning:** Understand system utilization and plan for future growth.
4. **Compliance and Auditing:** Ensure that systems meet required compliance standards.
5. **Cost Optimization:** Monitor and manage resources to prevent over-provisioning.

**AWS CloudWatch Insights:**

AWS CloudWatch provides real-time monitoring of AWS resources and applications. With CloudWatch, you can:

* Collect and monitor logs and metrics.
* Set alarms based on specific thresholds.
* Visualize logs and metrics in dashboards.
* Use CloudWatch Insights to analyze, visualize, and gain operational insights from logs data.

**Prometheus in the Kubernetes Ecosystem:**

Prometheus is an open-source monitoring and alerting toolkit designed for reliability and scalability. In the Kubernetes ecosystem, it's widely used due to its:

1. Flexible Query Language: Allows slicing and dicing of collected data.
2. Multi-dimensional Data Model: Uses key-value pairs.
3. Built-in Alerting: Integrates with tools like Alertmanager.
4. Ecosystem Integration: Has exporters for many services and integrations with Grafana for visualization.

**Tasks:**

**1. Integrate EKS with AWS CloudWatch:**

To integrate EKS with CloudWatch, you can use the CloudWatch Container Insights.

**Steps:**

1. Enable CloudWatch Container Insights on the EKS Cluster:

Using AWS Management Console:

* Navigate to the Amazon EKS console.
* Choose the cluster that you want to monitor.
* Choose the **Logging tab**.
* Select theContainer Insights checkbox and then choose Save changes

1. **Install CloudWatch agent and FluentD on your EKS cluster:**

Using kubectl

| kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml  kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml |
| --- |

**To install Fluentd:** [**Link here**](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs.html)

| **kubectl create configmap cluster-info \ --from-literal=cluster.name=cluster\_name \ --from-literal=logs.region=region\_name -n amazon-cloudwatch** |
| --- |

| **kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/fluentd/fluentd.yaml** |
| --- |

2. Verify the Installation:

| kubectl get pods -n amazon-cloudwatch |
| --- |

You should see pods with names starting with cloudwatch-agent and fluentd-cloudwatch.

**Task:**

### **2. Set up Prometheus and Grafana:**

**Step: install ebs driver**

| **kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.23"** |
| --- |

**Steps:**

1. **Install Prometheus using Helm:**

First, add the Prometheus community Helm repository:

Helm Install: [**Link Here**](https://helm.sh/docs/intro/install/)

| helm repo **add** prometheus-community https://prometheus-community.github.io/helm-charts helm update |
| --- |

**Now, install Prometheus:**

| **kubectl create namespace prometheus  helm repo add prometheus-community https://prometheus-community.github.io/helm-charts  helm install prometheus prometheus-community/prometheus \  --namespace prometheus \  --set alertmanager.persistentVolume.storageClass="gp2" \  --set server.persistentVolume.storageClass="gp2"** |
| --- |

| **kubectl get all -n prometheus** |
| --- |

After installation using Helm, you can port-forward Prometheus to access its UI:

| kubectl patch svc prometheus-server -n prometheus --type=json -p='[{"op": "replace", "path": "/spec/type", "value": "LoadBalancer"}]' |
| --- |

| kubectl get svc -n prometheus |
| --- |

Optional:

| kubectl port-forward -n prometheus svc/prometheus-server 8080:80 |
| --- |

1. **Install Grafana using Helm:**

**Add the Grafana Helm repository:**

| **nano /home/ubuntu/environment/grafana/grafana.yaml** |
| --- |

**Save and exit**

| **mkdir ${HOME}/environment/grafana  cat << EoF > ${HOME}/environment/grafana/grafana.yaml datasources:  datasources.yaml:  apiVersion: 1  datasources:  - name: Prometheus  type: prometheus  url: http://prometheus-server.prometheus.svc.cluster.local  access: proxy  isDefault: true EoF** |
| --- |

| helm repo **add** grafana https://grafana.github.io/helm-charts helm update |
| --- |

Install Grafana:

| kubectl create namespace grafana  helm install grafana grafana/grafana \  --namespace grafana \  --set persistence.storageClassName="gp2" \  --set persistence.enabled=true \  --set adminPassword='eks-monitoring' \  --values ${HOME}/environment/grafana/grafana.yaml \  --set service.type=LoadBalancer |
| --- |

**Task:**

**3. Create Dashboards and Alerts:**

**Access Grafana:**

After installation using Helm, you can port-forward Grafana to access its UI:

| kubectl get all -n grafana |
| --- |

Open a browser and navigate to **DNS Name**. Use **admin** as both username and password for the first login.

| kubectl get secret |
| --- |

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

**Add Prometheus as a Data Source:**

In Grafana:

1. Navigate to Configuration (gear icon) -> Data Sources.
2. Click "Add data source" and select "Prometheus".
3. Set the URL to **http://prometheus-server.monitoring.svc.cluster.local:80**
4. Click "Save & Test".

**Create Dashboards:**

You can either create custom dashboards by selecting panels and metrics of your choice, or import community-created dashboards for common applications.

**Set Up Alerts:**

In Grafana:

1. Edit a panel in your dashboard.
2. Click on the "bell" icon to open the Alert tab.
3. Set conditions for the alert and configure notifications. You can integrate with channels like Slack, Email, PagerDuty, etc.