**Setting Up Azure Monitor, Prometheus, and Grafana for containers based Application Monitoring**

This guide explains how to set up **Azure Monitor**, **Prometheus**, and **Grafana** to monitor your applications and infrastructure in Azure Kubernetes Service (AKS).

**High-Level Architecture**

1. **Azure Monitor**: Provides log analytics, metrics collection, and alerts for your AKS cluster and applications.

2. **Prometheus**: Collects application metrics using a Kubernetes-compatible exporter.

3. **Grafana**: Visualizes metrics from both Azure Monitor and Prometheus.

I’m considering the AKS resource. Azure monitoring can be enabled for other resources and logs can be pushed to LAW via diagnostics settings of each resource.

**Step 1: Set Up Azure Monitor for AKS**

Azure Monitor integrates with AKS to collect cluster and application insights.

**1.1 Enable Azure Monitor in AKS**

Use the Azure CLI to enable Azure Monitor for your AKS cluster.

az aks enable-addons \

  --resource-group <RESOURCE\_GROUP> \

  --name <AKS\_CLUSTER\_NAME> \

  --addons monitoring \

  --workspace-resource-id <LOG\_ANALYTICS\_WORKSPACE\_ID>

Please fill up the variables.

**1.2 Verify Integration**

1. Navigate to the **Azure Monitor > Insights > Containers** section in the Azure Portal.

2. Confirm that your AKS cluster appears with metrics and logs.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 2: Install Prometheus in AKS**

Prometheus collects metrics from Kubernetes components (via kube-state-metrics) and application-specific exporters.

**2.1 Install kube-prometheus-stack Using Helm**

1. Add the Prometheus Helm chart repository:

helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>

helm repo update

2. Install the kube-prometheus-stack chart in your AKS cluster:

helm install prometheus-stack prometheus-community/kube-prometheus-stack \

  --namespace monitoring \

  --create-namespace

3. Verify the installation:

kubectl get pods -n monitoring

A screenshot of a computer program

Description automatically generated

usha [ ~ ]$ kubectl get pods -n monitoring

NAME READY STATUS RESTARTS AGE

alertmanager-prometheus-stack-kube-prom-alertmanager-0 2/2 Running 0 61s

prometheus-prometheus-stack-kube-prom-prometheus-0 2/2 Running 0 61s

prometheus-stack-grafana-6ffd79cf4c-b5g8f 3/3 Running 0 75s

prometheus-stack-kube-prom-operator-6749ccd648-f44bl 1/1 Running 0 75s

prometheus-stack-kube-state-metrics-65f6dd579c-lxwfp 1/1 Running 0 75s

prometheus-stack-prometheus-node-exporter-grtnm 1/1 Running 0 75s

usha [ ~ ]$

**Step 3: Install Grafana in AKS**

Grafana is included in the kube-prometheus-stack Helm chart, but it needs configuration for Azure Monitor and Prometheus as data sources.

**3.1 Access Grafana**

1. Get the Grafana service details:

kubectl get svc -n monitoring prometheus-stack-grafana

2. Forward the Grafana service port to your local machine:

kubectl port-forward svc/prometheus-stack-grafana -n monitoring 3000:80

A screen shot of a computer

Description automatically generated

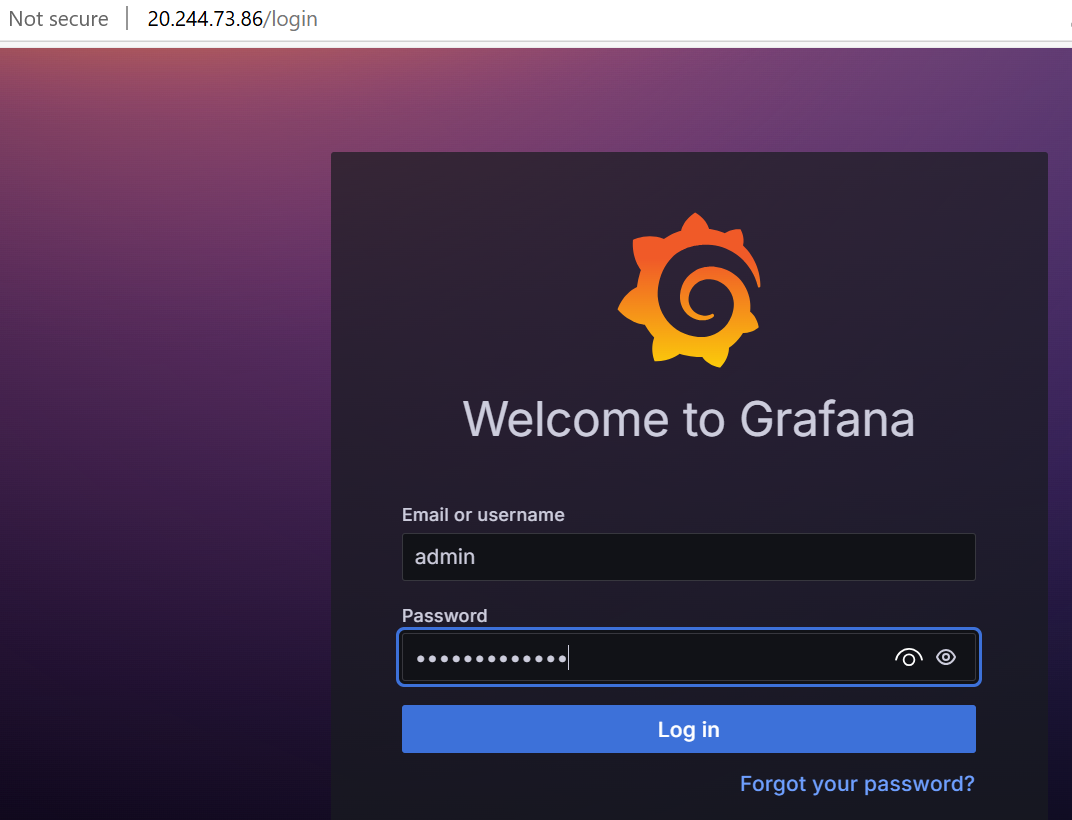
NOTE: You can expose the GRAFANA service as load balancer to get the public IP to avoid port-forwarding

3. Open Grafana in your browser: [http://localhost:3000](http://localhost:3000/)

• Default username: admin

• Default password: Run the following to retrieve the password:

kubectl get secret -n monitoring prometheus-stack-grafana -o jsonpath="{.data.admin-password}" | base64 --decode



**3.2 Add Prometheus as a Data Source**

1. Navigate to **Configuration > Data Sources** in Grafana.

2. Click **Add data source** and select **Prometheus**.

3. Use the following URL:

[http://prometheus-stack-prometheus.monitoring.svc.cluster.local:9090](http://prometheus-stack-prometheus.monitoring.svc.cluster.local:9090/)

A screenshot of a computer

Description automatically generated

4. Click **Save & Test**.

**Step 4: Add Azure Monitor as a Data Source in Grafana**

1. Install the Azure Monitor plugin for Grafana:

grafana-cli plugins install grafana-azure-monitor-datasource

2. Restart the Grafana pod:

kubectl delete pod -l [app.kubernetes.io/name=grafana](http://app.kubernetes.io/name=grafana) -n monitoring

3. Configure Azure Monitor in Grafana:

• Navigate to **Configuration > Data Sources > Add Data Source**.

• Select **Azure Monitor**.

• Provide the following details:

• **Tenant ID**: Found in Azure AD under **Properties**.

• **Client ID** and **Client Secret**: From an Azure AD service principal.

• **Subscription ID**: Your Azure subscription.

4. Click **Save & Test** to verify the connection.

**Step 5: Configure Dashboards**

**5.1 Import Pre-Built Dashboards**

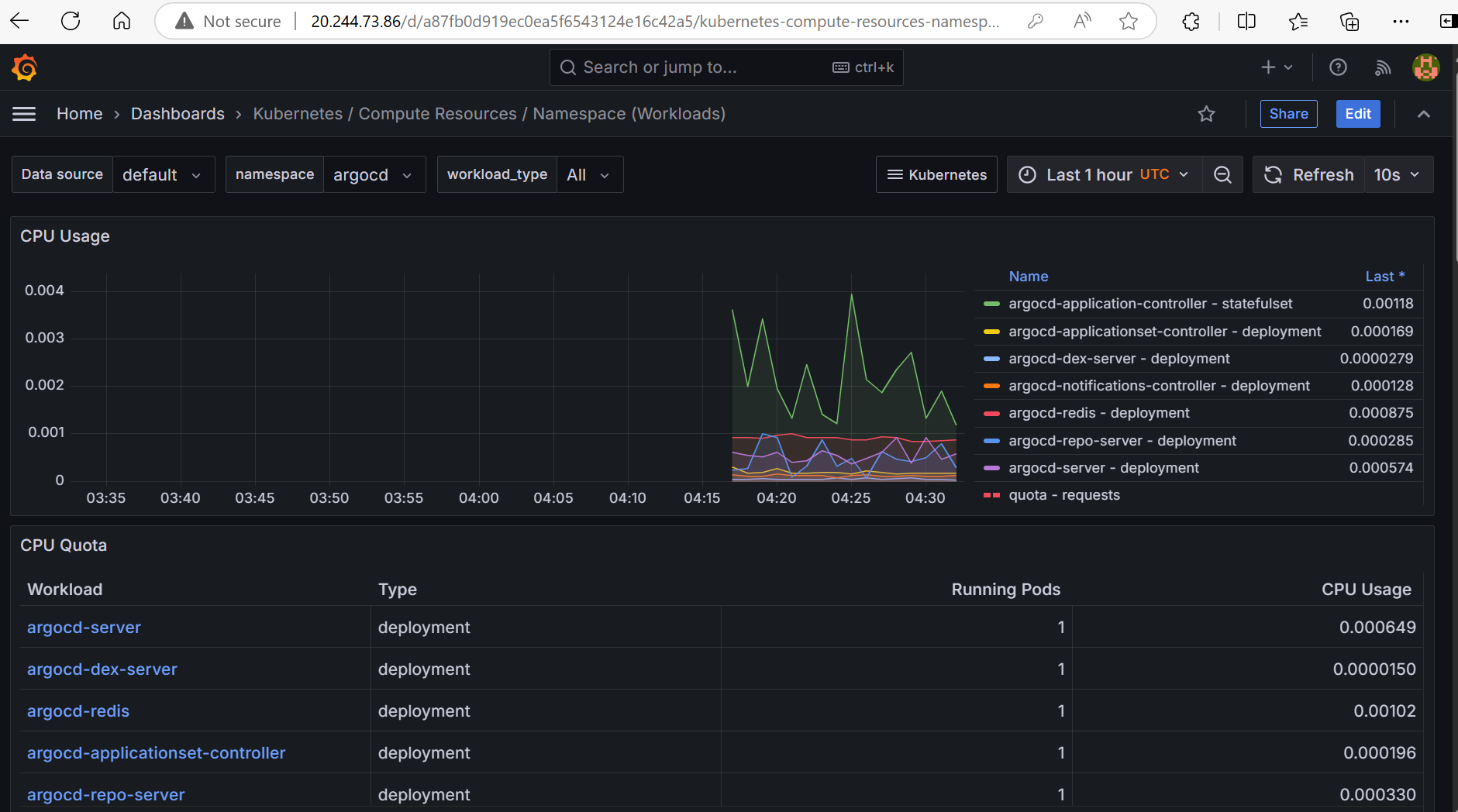
1. Navigate to **Create > Import** in Grafana.

2. Use the following dashboard IDs:

• **Prometheus Kubernetes Dashboard**: ID 315

• **Azure Monitor Insights Dashboard**: Search for Azure-specific dashboards.

3. Customize the dashboards to display AKS metrics, Prometheus application metrics, and Azure Monitor insights.



**Step 6: Enable Alerts**

**6.1 Azure Monitor Alerts**

1. **Navigate to Azure Monitor Alerts:**
   * Go to the Azure portal.
   * Search for and select "Monitor".
   * In the Monitor service, go to "Alerts" and then click on "New alert rule".
2. **Configure Alert Rule:**
   * Select the AKS resource you want to monitor from the list of available resources.
   * Under "Condition", choose the metric you want to create an alert for (e.g., CPU usage, memory pressure, pod status).
   * Set the threshold values that will trigger the alert (e.g., CPU usage > 80% for 5 minutes).
   * Configure the other settings such as the aggregation type and the period of time over which the data is collected and compared to the threshold.
3. **Set up Action Group:**
   * Create or select an existing action group to define what actions should be taken when the alert is triggered (e.g., sending an email, triggering a webhook, etc.).
4. **Define Alert Details:**
   * Provide a meaningful name for the alert rule and, if necessary, a description.
   * Set the severity level according to the importance of the alert.
5. **Create Alert Rule:**
   * Review the alert rule configuration and click "Create alert rule" to activate it.

**6.2 Prometheus Alerts**

1. Edit the Prometheus alertmanager configuration:

alerting:

  alertmanagers:

    - static\_configs:

        - targets:

          - 'alertmanager.monitoring.svc.cluster.local:9093'

2. Add alert rules in prometheus-rules.yaml and apply them:

apiVersion: monitoring.coreos.com/v1

kind: PrometheusRule

metadata:

name: example-alert

namespace: monitoring

spec:

groups:

- name: example

rules:

- alert: HighCpuUsage

expr: process\_cpu\_seconds\_total{job="your-job"} > 0.75

for: 5m

labels:

severity: critical

annotations:

summary: High CPU usage detected

description: "CPU usage is above 75% for more than 5 minutes."

kubectl apply -f prometheus-rules.yaml

**Outcome**

• **Azure Monitor** tracks AKS cluster performance and application logs.

• **Prometheus** collects application-specific metrics.

• **Grafana** visualizes metrics from both Azure Monitor and Prometheus.

This setup provides a robust monitoring and observability solution for your AKS cluster and applications. Let me know if you need further assistance!