**Practical Exercise: Cloud-Native Observability on Kubernetes**

Training Objective

Learn how to set up and view dashboards to analyze statistics of WSO2 Micro Integrator.

High-level Steps

* Install and set up monitoring tools.
* View the dashboards to see and analyze service mediation statistics.

Prerequisites

Download and install helm. Follow the instructions given below.

Setting up Cloud Native Observability on Kubernetes

Follow the instructions given below to set up a cloud native observability solution in a Kubernetes environment.

To streamline the deployment of the cloud native observability solution in Kubernetes, the Micro Integrator provides a Helm chart via which you can deploy the solution to your Kubernetes cluster. The deployment installs the relevant products and adds the required configurations. After the installation, you can directly use the observability solution with a few additional configurations.

Prerequisites

* Set up a Kubernetes cluster. For instructions, see [Kubernetes Documentation](https://kubernetes.io/docs/home/" \t "_blank).
* Install Helm in the client machine.

Setting up the observability deployment

When you deployed the solution on a VM, you first set up the minimum deployment (with metrics monitoring capability) and then added log processing and message tracing capabilities (as add-ons). However, when you deploy on Kubernetes, you must first select the required observability capabilities and then deploy all the related technologies and configurations in one step.

Select the required deployment option from the following list and follow the instructions.

Option 1 - Metrics Monitoring

The basic observability stack allows you to view metrics by installing and configuring Prometheus and Grafana. To install it, follow the steps below:

1. Clone the [Helm repository](https://github.com/wso2/observability-ei).
2. Navigate to the home directory of the cloned repository.
3. To install the basic deployment with the wso2-observability release name, issue the following command.

helm install wso2-observability . --render-subchart-notes

1. Make changes to the default settings of the chart if required. For information about configurable parameters, see [Integration Observability - Configuration](https://github.com/wso2/observability-ei#configuration).

The above step deploys the basic deployment and displays instructions to access the dashboards. This deployment allows you to access both Prometheus and Grafana UIs and provides you with the ability to view and analyze metrics.

Option 2 - Metrics + Log Monitoring

This deployment involves deploying Prometheus, Grafana, Loki, and Fluent-bit Daemon set with all the required configurations to integrate deployed products. To install the deployment using Helm, follow the steps below:

1. Clone the [Helm repository](https://github.com/wso2/observability-ei).
2. Navigate to the home directory of the cloned repository.
3. Open the values.yaml file and set the enabled parameter to true for Loki-stack as shown in the extract below.

loki-stack:

enabled: true

1. To install the observability deployment including log processing capabilities with the wso2-observability release name, issue the following command.

helm install wso2-observability . --render-subchart-notes

1. Make changes to the default settings of the chart if required. For information about configurable parameters, see [Integration Observability - Configuration](https://github.com/wso2/observability-ei#configuration).

The above steps deploy the observability solution with log processing capabilities and display instructions to access the dashboards. With this deployment, you can access Prometheus and Grafana UIs.

Option 3 - Metrics Monitoring + Message Tracing

This involves deploying Prometheus, Grafana, and Jaeger-operator with all the required configurations to integrate deployed products. To install the deployment using Helm, follow the steps below:

1. Clone the [Helm repository](https://github.com/wso2/observability-ei).
2. Navigate to the home directory of the cloned repository.
3. Open the values.yaml file and set the enabled parameter to true for Jaeger as shown in the extract below.

jaeger:

enabled: true

1. To install the observability deployment including tracing capabilities with the wso2-observability release name, issue the following command.

helm install wso2-observability . --render-subchart-notes

1. Make changes to the default settings of the chart if required. For information about configurable parameters, see [Integration Observability - Configuration](https://github.com/wso2/observability-ei#configuration).

The above steps deploy the observability solution with tracing capabilities and displays instructions to access the dashboards. With this deployment, you are able to access Prometheus, Grafana, and Jaeger UIs.

This deployment installs Jaeger-Operator. To install the Jaeger deployment, follow the steps in [Jaeger Operator documentation - Creating a new instance](https://github.com/jaegertracing/helm-charts/tree/master/charts/jaeger-operator#creating-a-new-jaeger-instance) and deploy the preferred Jaeger deployment.

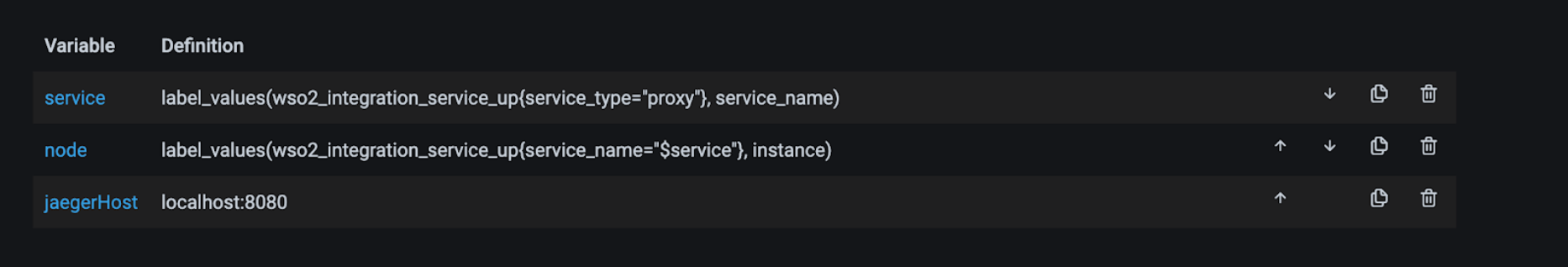
**Note**

* There are some limitations because the Jaeger client, by default, uses a UDP sender as mentioned in [the Jaeger documentation](https://www.jaegertracing.io/docs/1.22/client-libraries/). If the payload size exceeds 65 KB, spans might get lost in the Jaeger console.
* Jaeger [sampler types](https://www.jaegertracing.io/docs/1.22/sampling/) can also play a major role in tracing. Depending on the TPS, the sampler type should be carefully chosen.
* Be sure to check the performance tests and scaling requirements before including tracing in production deployments. For details on how to achieve better performance, see the [Jaeger performance tuning guide](https://www.jaegertracing.io/docs/1.22/performance-tuning/).

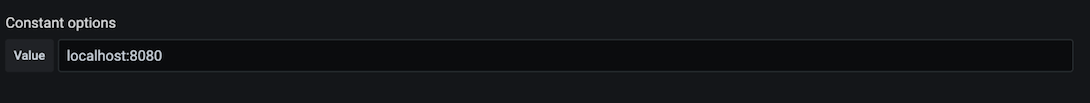
**CONFIGURING GRAFANA TO VISUALIZE TRACING INFORMATION**

The Helm chart configures the Jaeger data source automatically. Therefore, unlike in Setting up [Cloud Native Observability in a Virtual Machine](https://apim.docs.wso2.com/en/4.1.0/observe/micro-integrator/setting-up-cloud-native-observability-on-a-vm), it is not required to add it manually. However, to configure the links into Jaeger UI from the service-level dashboards, you need to perform the following steps:

1. Access Grafana via localhost:3000 and sign in.
2. Navigate to the settings section of the service level dashboard by clicking the cog wheel icon in the top right corner.
3. Click **Variable**. This opens the following view.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/monitoring-dashboard/variables.png)

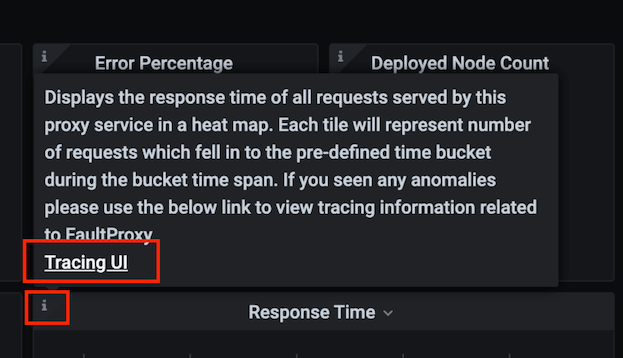
1. Edit the JaegerHost variable and provide your Jaeger query component hostname and port in the host:port syntax as shown below.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/monitoring-dashboard/constant-options.png)

1. Click **Save**

You need to perform the above steps for all the service-level dashboards (i.e., Proxy Service dashboard, API Service Dashboard, and Inbound Endpoint dashboard).

Once Grafana is successfully configured to visualize statistics, you should be correctly redirected to the Jaeger UI from the Response Time widget of each service level dashboard as shown below.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/monitoring-dashboard/jaeger-ui.png)

Option 4 - Metrics + Logs + Message Tracing

To install the cloud native observability solution with logging and tracing capabilities in your Kubernetes cluster, follow the steps below:

1. Clone the [Helm repository](https://github.com/wso2/observability-ei).
2. Navigate to the home directory of the cloned repository.
3. Open the values.yaml file and set the enabled parameter to true for both Loki-stack and Jaeger as shown in the extract below.

loki-stack:

enabled: true

jaeger:

enabled: true

1. To install the complete deployment with the wso2-observability release name, issue the following command.

helm install wso2-observability . --render-subchart-notes

1. Make changes to the default settings of the chart if required. For information about configurable parameters, see [Integration Observability - Configuration](https://github.com/wso2/observability-ei#configuration).

The above step deploys the complete deployment and displays instructions to access the dashboards. This deployment allows you to access Prometheus, Grafana, and Jaeger UIs.

Setting up the Micro Integrator deployment

To integrate with the observability deployment, you are required to perform the following three main tasks in containers:

Enabling observability for the Micro Integrator

* **Enabling the statistics publishing handler**

Add the following lines in the <PATH>/deployment.tomlfile in the Kubernetes project *before* creating your micro integrator image.

[[synapse\_handlers]]

name="MetricHandler"

class="org.wso2.micro.integrator.observability.metric.handler.MetricHandler"

For more information about the Micro Integrator Kubernetes development flow, see [MI Kubernetes guide](https://apim.docs.wso2.com/en/4.1.0/install-and-setup/setup/mi-setup/deployment/kubernetes_deployment_patterns/).

* **Enabling the metrics endpoint**

Set an environment variable in the Kubernetes resource definition. You can either add that at the time of creating the project using the wizard. Alternatively, you can open the /integration\_cr.yaml file in the Kubernetes project and add the following under the spec tag.

env:

- name: "JAVA\_OPTS"

value: "-DenablePrometheusApi=true"

* **Enabling discovery for Prometheus**

This allows Prometheus to discover Micro Integrator targets through service discovery methods. To achieve this, set the following pod level annotations to the Micro Integrator pod.

* + prometheus.io.wso2/path: /metric-service/metrics
  + prometheus.io.wso2/port: "9201"
  + prometheus.io.wso2/scrape: "true"

Configuring the Micro Integrator to publish logs

**Tip**

This step is only required if you have log processing capabilities in your observability deployment.

Once the above tasks are completed, the container that is being deployed through the integration Kubernetes resource emits metric data, and the Observability deployment can discover and start without further configuration.

**Configuring pods to parse logs through Fluent-bit**

To do this, set the following pod level annotation to the Micro Integrator pod.

fluentbit.io/parser: wso2

Configuring the Micro Integrator to publish tracing information

**Tip**

This step is only required if you have message tracing capabilities in your observability deployment.

To configure the Micro Integrator to publish tracing information, add the following lines to the deployment.toml file in the Kubernetes project *before* creating your micro integrator container image.

[mediation]

flow.statistics.capture\_all= true

stat.tracer.collect\_payloads= true

stat.tracer.collect\_mediation\_properties= true

[opentracing]

enable = true

jaeger.sampler.manager\_host="hostname"

jaeger.sender.agent\_host="hostname"

**Tip**

Enter the host name of your Jaeger service as the value for manager\_host and agent\_host parameters.

These settings enable the tracing data instrumentation and publishing to a Jaeger instance.