**Technical Documentation: On Telemetry**

**Table of Contents:**

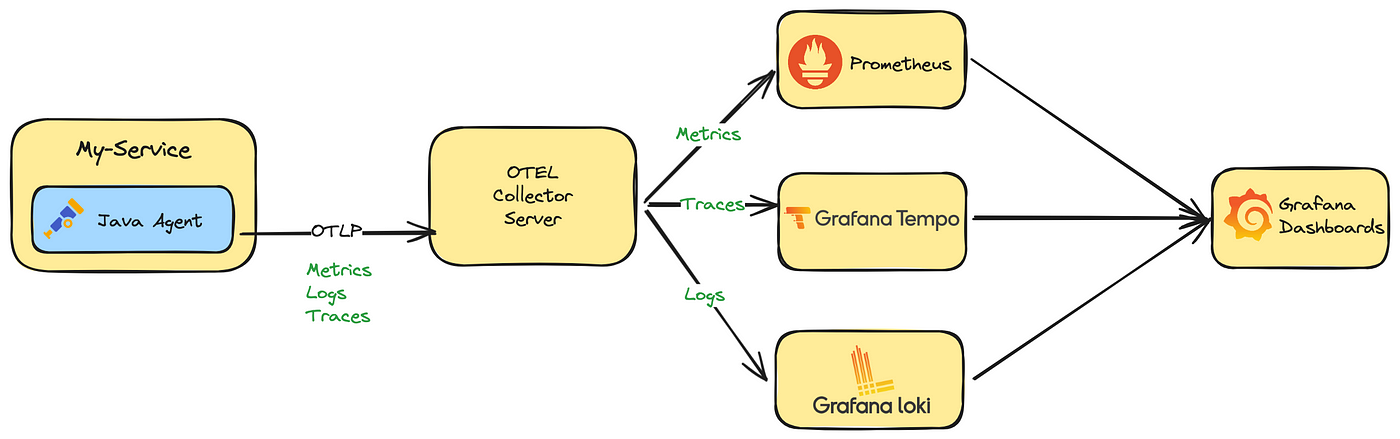
1. **Introduction**
2. **Architecture Overview**
3. **Pre-requisites**
4. **Setting Up Azure Environment**
5. **Deploying Grafana for Monitoring**
6. **Setting Up Loki for Log Aggregation**
7. **Setting Up Tempo for Tracing**
8. **Configuring Fluentd for Log Forwarding**
9. **Integration and Configuration**
10. **Security Considerations**
11. **Maintenance and Monitoring**
12. **Troubleshooting**
13. **Conclusion**

**1. Introduction**

This document outlines the steps for implementing a telemetry system using Grafana, Loki, Tempo, and Fluentd on Azure. This setup provides comprehensive monitoring, logging, and tracing capabilities for your organization's cloud-native applications deployed on Azure.

**2. Architecture Overview**

**Integration Diagram:**



**Component Diagram:**

**A diagram of a computer

Description automatically generated**

The architecture involves:

* **Grafana**: Visualization tool for metrics, logs, and traces.
* **Loki**: Log aggregation system, integrated with Grafana.
* **Tempo**: Distributed tracing system, integrated with Grafana.
* **Fluentd**: Log collector and forwarder, sending logs to Loki and other destinations.
* **Azure Services**: Azure Kubernetes Service (AKS), Azure Storage for long-term data retention and for additional insights.

**High-Level Workflow:**

1. Applications generate logs and traces.
2. Fluentd collects logs and forwards them to Loki.
3. Traces are sent directly to Tempo.
4. Grafana visualizes metrics from various data sources and integrates logs from Loki and traces from Tempo.

**3. Pre-requisites**

1. **Azure Subscription** with necessary permissions.
2. **Kubernetes Cluster** (AKS) set up on Azure.
3. **Azure CLI** and **kubectl** installed and configured.
4. **Helm** installed for managing Kubernetes packages.
5. **Azure Storage Account** for persisting logs and traces (optional but recommended).
6. **Basic understanding** of Kubernetes, Docker, and Azure services.

**4. Setting Up Azure Environment**

1. **Create AKS Cluster**:

bash

Copy code

az aks create --resource-group <resource-group> --name <cluster-name> --node-count 3 --enable-addons monitoring --generate-ssh-keys

1. **Create Azure Storage Account** (for persistent storage):

bash

Copy code

az storage account create --name <storage-account-name> --resource-group <resource-group> --location <location> --sku Standard\_LRS

1. **Create Azure Container Registry (ACR)** (optional, for custom images):

bash

Copy code

az acr create --resource-group <resource-group> --name <acr-name> --sku Basic

**5. Deploying Grafana for Monitoring**

1. **Add the Grafana Helm repository**:

bash

Copy code

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

helm repo update

1. **Install Grafana using Helm**:

bash

Copy code

helm install grafana grafana/grafana --namespace monitoring --create-namespace

1. **Expose Grafana via a LoadBalancer or Ingress**:

bash

Copy code

kubectl expose deployment grafana --type=LoadBalancer --name=grafana-service --namespace monitoring

1. **Access Grafana**:
   * Get the external IP using kubectl get services -n monitoring.
   * Default credentials: admin/admin (change it after first login).

**6. Setting Up Loki for Log Aggregation**

1. **Add the Loki Helm repository**:

bash

Copy code

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

helm repo update

1. **Install Loki using Helm**:

bash

Copy code

helm install loki grafana/loki-stack --namespace monitoring --set grafana.enabled=false,promtail.enabled=true

1. **Configure Persistent Storage** (optional):
   * Edit values to include Azure Storage:

yaml

Copy code

loki:

storage:

type: azure

azure:

account\_name: <storage-account-name>

account\_key: <storage-account-key>

container\_name: loki

**7. Setting Up Tempo for Tracing**

1. **Install Tempo using Helm**:

bash

Copy code

helm install tempo grafana/tempo --namespace monitoring

1. **Configure Persistent Storage for Traces** (optional):
   * Edit values to include Azure Storage:

yaml

Copy code

tempo:

storage:

trace:

backend: azure

azure:

account\_name: <storage-account-name>

account\_key: <storage-account-key>

container\_name: tempo

**8. Configuring Fluentd for Log Forwarding**

1. **Deploy Fluentd as DaemonSet**:

bash

Copy code

kubectl apply -f https://raw.githubusercontent.com/fluent/fluentd-kubernetes-daemonset/master/fluentd-daemonset-elasticsearch-rbac.yaml

1. **Configure Fluentd to Send Logs to Loki**:
   * Edit the Fluentd config to include Loki output plugin:

yaml

Copy code

<match \*\*>

@type loki

url "http://<loki-service>:3100/loki/api/v1/push"

labels { "job": "fluentd" }

line\_format "json"

</match>

**9. Integration and Configuration**

1. **Grafana Data Sources**:
   * **Add Loki**:
     + In Grafana, go to Configuration > Data Sources > Add data source > Loki.
   * **Add Tempo**:
     + Go to Configuration > Data Sources > Add data source > Tempo.
2. **Dashboard Configuration**:
   * Import or create dashboards to visualize logs and traces.
   * Use pre-built dashboards available on Grafana.com or customize according to your requirements.

**10. Security Considerations**

1. **Secure Access**:
   * Implement **OAuth2** or **Azure AD** for Grafana authentication.
   * Use **RBAC** in Kubernetes to control access to resources.
   * Enable **SSL/TLS** for secure communication.
2. **Network Policies**:
   * Implement Kubernetes Network Policies to restrict traffic between namespaces and components.
3. **Data Encryption**:
   * Ensure that logs and traces stored in Azure Storage are encrypted.

**11. Maintenance and Monitoring**

1. **Resource Monitoring**:
   * Use Azure Monitor and Grafana dashboards to monitor resource usage.
   * Set up alerts for critical thresholds.
2. **Data Retention**:
   * Configure retention policies in Loki and Tempo to manage storage consumption.
   * Periodically clean up old logs and traces.
3. **Backup and Recovery**:
   * Regularly back up Grafana, Loki, and Tempo configurations.
   * Implement disaster recovery strategies for Azure Storage.

**12. Troubleshooting**

1. **Common Issues**:
   * **Grafana not loading data**: Check data source configurations and network connectivity.
   * **Fluentd logs not appearing in Loki**: Verify Fluentd config and Loki endpoint.
   * **Tempo traces missing**: Ensure that the application is correctly instrumented with tracing libraries.
2. **Logs and Diagnostics**:
   * Access pod logs using OpenLens.
   * (Initially) Use Azure Monitor logs for additional diagnostics.

**13. Conclusion**

This telemetry implementation using Grafana, Loki, Tempo, and Fluentd on Azure provides a robust monitoring and observability stack for cloud-native applications. By following the steps in this documentation, your organization can achieve end-to-end visibility into application performance, logs, and traces, ensuring better reliability, security, and operational efficiency.

Open Telemetry installation following below helm.

helm install my-otel-collector open-telemetry/opentelemetry-collector -f values.yaml