Logging

Proper logging is required for analysing the performance of the deployed applications and evaluating the efficiency of the placement algorithms.

Kubernetes Cluster Logging : Kubernetes facilitates integration of cluster-level logging architectures where logs are stored per cluster, independent of pods and containers.

We choose Grafana Loki as it’s a light weight logging framework compared to other options like ElasticSearch . So we use it together with MinIO to store the logs and use Grafana to query and visualise them.

Grafana Loki:

Installation

1. Create istio-injection enabled namespace
2. Install Helm https://helm.sh/docs/intro/install/
3. Grafana Loki stack consist of 3 components to install : Locki, Promtail and Grafana
    a. Grafana : installed through Istio plugin (https://istio.io/latest/docs/tasks/observability/metrics/using-istio-dashboard/ ,https://istio.io/latest/docs/tasks/observability/gateways/)
    b. Locki is installed with grafana/locki helm chart
        i. first the values.yaml should be updated to store the logs in MinIO

```
helm show values grafana/loki > loki-values.yaml
# update loki-values.yaml and then install loki with the updated configurations
helm install loki grafana/loki -f loki-values.yaml --namespace istio-system
helm upgrade --install --namespace <loggingNs> --set loki.serviceName=loki-gateway promtail grafana/promtail
```

    c. Add Loki as a data source in Grafana through loki-gateway

```
http://loki-gateway.logging.svc.cluster.local:80
```

> Updating loki-values.yaml (https://picluster.ricsanfre.com/docs/loki/)
>
> For Minio setup : a new repository should be created for logs.
>
> Minio url can be selection as either `minio.control-engine.svc.cluster.local:9000` to access closest one or `minio-<clustername>.control-engine.svc.cluster.local:9000` to access a particular Minio instance (if not enough space in Fog, cloud can be used to store data this way)