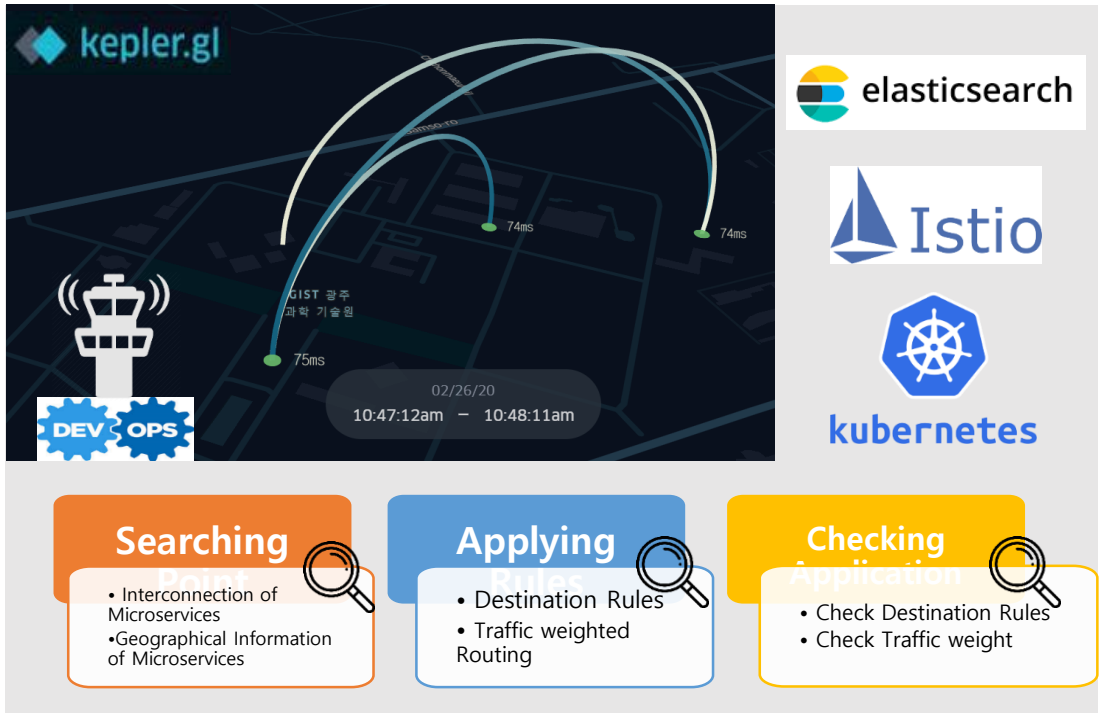
Visualizing Cloud-Native AI+X Applications employing Service Mesh

Overview



서비스 메쉬를 활용한 클라우드 네이티브 AI+X 어플리케이션 시공간적 가시화

Prerequisite (1/3)

Insall Istio on your kubernetes cluster

Proceed with the premise that you have a cluster running a compatible version of Kubernetes.

Please refer to the following site: https://istio.io/docs/setup/getting-started/

Download Istio

```
$ curl -L https://istio.io/downloadIstio | sh -
$ cd istio-1.6.0
$ export PATH=$PWD/bin:$PATH
```

Install Istio

```
$ istioctl manifest apply --set profile=demo
$ kubectl label namespace default istio-injection=enabled //Set to namespace where envoy is inserted.
```

Deploy the sample application

```
$ kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml
$ kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml
```

Prerequisite (2/3)

Insall EFK (Elasticsearch + Fluentd + Kibana)

For logging-stack.yaml file,

please refer to the following site: https://istio.io/docs/tasks/observability/mixer/logs/fluentd/

```
$ kubectl apply -f logging-stack.yaml
```

View the new logs

1. Navigate to the Kibana UI and click the "Set up index patterns" in the top right.
2. Use * as the index pattern, and click "Next step.".
3. Select @timestamp as the Time Filter field name, and click "Create index pattern."
4. Now click "Discover" on the left menu, and start exploring the logs generated

Prerequisite (3/3)

Insall Python and packages

Configure the environment for implementing visualizations.

Install Python

```
$ sudo apt-get install python3
$ sudo apt-get install python3-pip
```

Install packages (Elasticsearch, Kubernetes, Keplergl, Pandas, Flask)

```
$ pip3 install elasticsearch
$ pip3 install kubernetes
$ pip3 install keplergl
$ pip3 install pandas
$ pip3 install -U Flask
```

Configure Istio

Configuration for access logentry instances

Create a logentry consisting of information such as the picture.

```
source: source.name | "unknown"
srcnamespace: source.namespace | "unknown"
destination: destination.name | "unknown"
desnamespace: destination.namespace | "unknown"
latency: response.duration | "0ms"
```

```
$ kubectl apply -f fluent.yaml
```

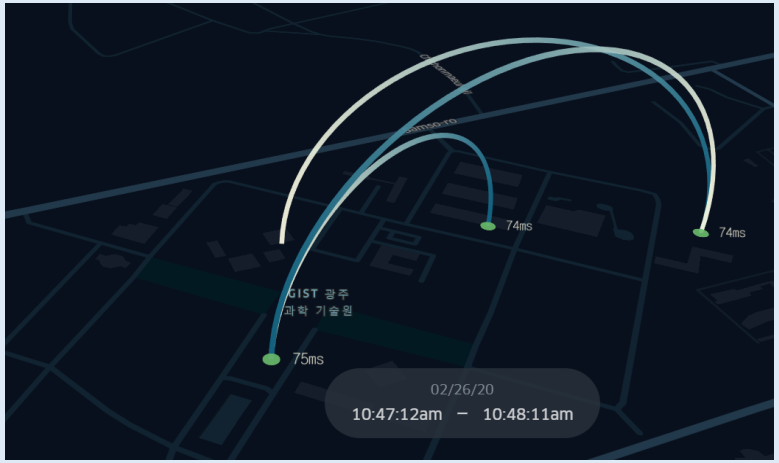
Run Visualization

Run python file for visualization

Interconnectivity between micro-services obtained through the istio has been visualized through keplergl.

```
$ python to_kepler.py
```

http://localhost:8080/{ namespace_name }



Kiali Dashboard for Kubernetes Service Graph

```
$ export PATH=$PATH:$HOME/.istioctl/bin

$ istioctl  dashboard kiali
```

Login (username: admin / password: admin) -> Graph (namespace: default / Versioned app graph)

If you want to check traffic status in real time, you must connect *http://210.125.84.90/productage* to generate traffic.

Kiali dashboard ->