



Workshop

Istio mixer adapter for WSO2 API Manager

Requisitos	3
Instalar o Istio	4
Instalar o API Manager e o API Manager Analytics no Istio	5
Instalar o WSO2 Istio Mixer Adapter	7
Deploy do Microserviço no Istio	8
Aplicar o API Management para os Microserviços	9
Anexo	11

Requisitos

Github: <https://github.com/wso2/istio-apim>

- Download e Instalação do Istio: <https://istio.io/docs/setup/#downloading-the-release>
- Configuração do diretório "bin" do Istio
- Configuração do Docker Desktop - <https://istio.io/docs/setup/platform-setup/docker/>
- Instalar o Docker - Docker for Desktop e Kubernetes habilitados (kubectl). Pode utilizar o Minikube se preferir. Este workshop utilizamos o Docker for Desktop
- Executar: docker system prune
- Verificar as configurações de recursos do Docker - Processador e memória - seguir a recomendação da WSO2

Instalar o Istio

1. Instalar o Istio de acordo com o site: <https://istio.io/docs/setup/install/kubernetes/>

```
cd istio-1.3.4
```

```
for i in install/kubernetes/helm/istio-init/files/crd*.yaml; do kubectl  
apply -f $i; done
```

```
kubectl apply -f install/kubernetes/istio-demo.yaml
```

2. Verificar a Instalação

```
kubectl get svc -n istio-system
```

3. Verificar se os Pods do Kubernetes estão em execução - "running"

```
kubectl get pods -n istio-system
```

Instalar o API Manager e o API Manager Analytics no Istio

1. Download do API Manager o API Manager Analytics através do site:
<https://github.com/wso2/istio-apim/releases/tag/1.0>
 - a. Copiar para a pasta específica do projeto/workshop e descompactar o ZIP o arquivo baixado.
2. Deploy dos artefatos do API Manager Analytics no Kubernetes
 - a. Ir para a pasta do API Manager
`cd workshop-istio-adapter/wso2am-istio-1.0`
 - b. Executar o comando para deploy
`kubectl apply -f install/analytics/k8s-artifacts/`

Saída esperada:

```
> kubectl apply -f install/analytics/k8s-artifacts/  
namespace/wso2 created  
deployment.apps/wso2apim-with-analytics-apim-analytics-deployment created  
service/wso2apim-with-analytics-apim-analytics-service created
```

3. Deploy os Configmaps do API Manager no Kubernetes
 - a. Executar os comandos para deploy

```
kubectl create configmap apim-conf  
--from-file=./install/api-manager/resources/conf/ -n wso2
```

```
kubectl create configmap apim-lifecycles  
--from-file=./install/api-manager/resources/lifecycles/ -n wso2
```

Saída esperada dos dois comandos:

```
> kubectl create configmap apim-conf --from-file=./install/api-manager/resources/conf/ -n wso2  
configmap/apim-conf created  
  
~/Documents/Workshop/workshop-istio/wso2am-istio-1.0  
> kubectl create configmap apim-lifecycles --from-file=./install/api-manager/resources/lifecycles/ -n wso2  
configmap/apim-lifecycles created
```

4. Deploy dos artefatos do K8s do API Manager no Kubernetes
 - a. Executar o comando para deploy
`kubectl apply -f install/api-manager/k8s-artifacts/`

Saída esperada:

```
> kubectl apply -f install/api-manager/k8s-artifacts/  
namespace/wso2 unchanged  
serviceaccount/wso2svc-account created  
clusterrole.rbac.authorization.k8s.io/crd-deploy created  
clusterrolebinding.rbac.authorization.k8s.io/crd-deploy-binding created  
deployment.apps/wso2apim-with-analytics-apim created  
service/wso2apim-with-analytics-apim-service created
```

5. Acessar os portais do API Manager

a. Adicionar o K8s Node IP ao /etc/hosts

<K8s_node_ip> wso2apim

Ex: 127.0.0.1 wso2apim

Publisher: <https://wso2apim:32001/publisher>

Store: <https://wso2apim:32001/store>

Admin: <https://wso2apim:32001/admin>

Instalar o WSO2 Istio Mixer Adapter

1. Criar um segredo K8s (secret) no namespace istio-system para o certificado público do WSO2 API Manager

- a. Executar o comando:

```
kubectl create secret generic server-cert  
--from-file=./install/adapter-artifacts/server.pem -n istio-system
```

Saída esperada:

```
> kubectl create secret generic server-cert --from-file=./install/adapter-artifacts/server.pem -n istio-system  
secret/server-cert created
```

2. Deploy do wso2-adapter como um serviço do cluster

- a. Executar o comando:

```
kubectl apply -f install/adapter-artifacts/
```

Saída esperada:

```
> kubectl apply -f install/adapter-artifacts/  
attributemanifest.config.istio.io/istio-proxy created  
attributemanifest.config.istio.io/kubernetes configured  
template.config.istio.io/authorization created  
template.config.istio.io/metric created  
secret/wso2server-secret created  
service/wso2adapterservice created  
deployment.apps/wso2adapter created  
handler.config.istio.io/wso2-handler created  
instance.config.istio.io/wso2-authorization created  
instance.config.istio.io/wso2-metrics created  
adapter.config.istio.io/wso2 created
```

Deploy do Microserviço no Istio

1. Habilitar o "istio sidecar injection" no namespace default. Se não estiver habilitado ainda.

- a. Executar o comando:

```
kubectl label namespace default istio-injection=enabled
```

Saída esperada:

```
> kubectl label namespace default istio-injection=enabled
namespace/default labeled
```

2. Deploy do serviço de exemplo HTTPBIN.

- a. Executar o comando:

```
kubectl create -f samples/httpbin/httpbin.yaml
```

Saída esperada:

```
> kubectl create -f samples/httpbin/httpbin.yaml
service/httpbin created
deployment.apps/httpbin created
```

3. Export o serviço HTTPBIN via ingress istio gateway.

- a. Executar o comando:

```
kubectl create -f samples/httpbin/httpbin-gw.yaml
```

Saída esperada:

```
> kubectl create -f samples/httpbin/httpbin-gw.yaml
gateway.networking.istio.io/httpbin-gateway created
virtualservice.networking.istio.io/httpbin created
```

4. Chamar o serviço HTTPBIN. Acessar o serviço httpbin via Istio ingress gateway.

- a. Usar o Postman:

<http://localhost/headers>

<http://localhost/ip>

Aplicar o API Management para os Microserviços

1. Criar e publicar uma API no WSO2 API Manager Publisher
 - a. Logar no WSO2 API Manager Publisher
 - b. Criar a API de acordo com os dados abaixo:

API Name : HttpbinAPI

API Context : /httpbin

API Version : 1.0.0

Production Endpoint : <http://httpbin.default.svc.cluster.local>

Verifique se está usando a URL correta do seu endpoint:

`http://<service_name>.<namespace_of_the_service>.svc.cluster.local`

Recurso	Método	Escopo	Escopo - Função
/ip	GET	scope_ip	admin
/headers	GET	scope_headers	admin
/delay/{delay}	GET		
/status/{status_code}	GET		

Quando criar uma API, o WSO2 API Manager automaticamente cria e implanta os recursos do Istio para a API.

*** Selecionar um ou mais planos de subscrição da API

2. Acessar o serviço utilizando um token JWT
 - a. Acessar o WSO2 API Manager Store.
 - b. Criar uma aplicação - Selecionar o tipo de Token como JWT.
 - c. Subscrever a API (httpbinAPI) e utilizar a aplicação que você criou.
 - d. Selecione os escopos relevantes e gerar um token de acesso.
 - e. Chamar a API fornecendo no cabeçalho da autorização. Utilizar a coleção do Postman.
3. Acessar o Analytics para "Business Insights". Acesse na Store e no Publisher e veja as análises.

4. Veja no Anexo como acessar o Grafana para telemetria;
5. Veja no Anexo como monitorar o WSO2 Adapter.

6. Remover a instalação do API Manager

```
kubectl delete -f samples/httpbin/
kubectl delete -f install/adapter-artifacts/
kubectl delete secrets server-cert -n istio-system
kubectl delete -f install/analytics/k8s-artifacts/
kubectl delete -f install/api-manager/k8s-artifacts/
kubectl delete configmap apim-conf -n wso2
kubectl delete configmap apim-lifecycles -n wso2
```

7. Remover a instalação do Istio

```
for i in install/kubernetes/helm/istio-init/files/crd*.yaml; do kubectl
delete -f $i; done
```

Anexo

Set WSO2 namespace as default:

```
kubectl config set-context --current --namespace=wso2
kubectl config set-context --current --namespace=default
```

Serviço HTTPBin

<https://httpbin.org/#/>

Troubleshooting Guide

Recuperar o nome do pod do wso2adapter:

```
kubectl get pods -n istio-system -l app=wso2adapter
```

Com a saída do comando acima use para mostrar o log do wso2adapter:

```
kubectl logs -f <pod_name> -n istio-system
```

Grafana

Iniciar o roteamento do Grafana para porta 3000

```
kubectl -n istio-system port-forward $(kubectl -n istio-system
get pod -l app=grafana -o jsonpath='{.items[0].metadata.name}')
3000:3000 &
```

Acesso ao Mesh dashboard:

<http://localhost:3000/d/G8wLrJIZk/istio-mesh-dashboard?orgId=1>

Acesso ao Service Dashboard:

http://localhost:3000/d/LJ_uJAvmk/istio-service-dashboard?orgId=1

=1

Acesso ao Istio Dashboard:

<http://localhost:3000/d/UbsSZTDik/istio-workload-dashboard?orgId=1>

d=1

Istio Telemetry

<https://istio.io/docs/tasks/telemetry/>