metalLB setting

MetalLB deployment

Deploy MetalLB:

1. Apply the manifest:

   ```
   kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.8.1/manifests/metallb.yaml
   ```

2. Allocate a pool of addresses on your local network for MetalLB to use. You need at least one address for the Istio Gateway. This example assumes addresses 10.0.0.100-10.0.0.110. You must modify these addresses based on your environment.

   ```
   cat <<EOF | kubectl apply -f -
   apiVersion: v1
   kind: ConfigMap
   metadata:
     namespace: metallb-system
     name: config
   data:
     config: |
       address-pools:
       - name: default
         protocol: layer2
         addresses:
         - 10.0.0.100-10.0.0.110
   EOF
   ```

Ensure that MetalLB works as expected (optional):

1. Create a dummy service:

   ```
   kubectl create service loadbalancer nginx --tcp=80:80
   ```

2. Ensure that MetalLB has allocated an IP address for the service:

   ```
   kubectl describe service nginx
   ```

3. Check the corresponding MetalLB logs:

   ```
   kubectl logs -n metallb-system -l component=controller
   ```

4. Create a pod that will be exposed with the service:

```
kubectl run nginx --image nginx --restart=Never -l app=nginx
```

5. Ensure that MetalLB has assigned a node to announce the allocated IP address:

```
kubectl describe service nginx
```

6. Check the corresponding MetalLB logs:

```
kubectl logs -n metallb-system -l component=speaker
```

7. Check that MetalLB responds to ARP requests for the allocated IP address:

```
arping -I eth0 10.0.0.101
```

8. Check the corresponding MetalLB logs:

```
kubectl logs -n metallb-system -l component=speaker
```

9. Verify that everything works as expected:

```
curl http://10.0.0.101
```

10. Clean up:

```
kubectl delete service nginx
kubectl delete pod nginx
```

To expose Kubeflow with a LoadBalancer Service, just change the type of the istio-ingressgateway Service to LoadBalancer.

```
kubectl patch service -n istio-system istio-ingressgateway -p '{"spec": {"type": "LoadBalancer"}}'
```

After that, get the LoadBalancer's IP or Hostname from its status and create the necessary certificate.

Create the Certificate with cert-manager:

After applying the above Certificate, cert-manager will generate the TLS certificate inside the istio-ingressgateway-certs secrets. The istio-ingressgateway-certs secret is mounted on the istio-ingressgateway deployment and used to serve HTTPS.

Navigate to `https://<LoadBalancer Address>/` and start using Kubeflow.