COMMUNITY DAY

AHMEDABAD

aws COMMUNITY DAY

Service Mesh: ISTIO on EKS Cluster

Dharmang Makwana | 20th January 2024
HOD: DevOps & SRE @Panamax Inc

aws COMMUNITY DAY

01 What is Service Mesh?

02 What is ISTIO?

03 ISTIO Advantages

04 How to Design Secure End-End Encryption

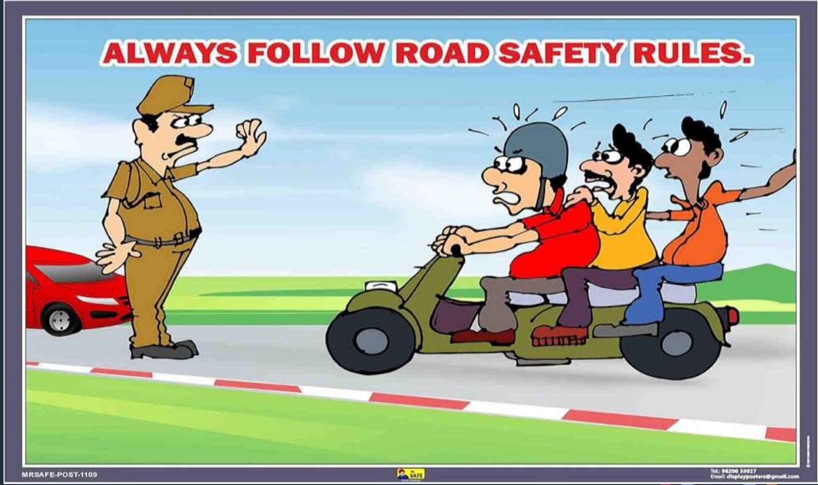
aws

COMMUNITY DAY

What is Service Mesh?

A Service Mesh is a tool for adding observability, security and reliability features to application by inserting this features at the platform layer rather than implementing it on the application layer.

aws COMMUNITY DAY



Let me make it very Simple – A Dev Job

What if I ask Development team to build below functionality?

- Traffic Management: Adding the advance traffic routing capability at the application layer like load balancing, implementation of the network policy to route the traffic

aws COMMUNITY DAY

Let me make it very Simple – A Dev Job

What if I ask Development team to build below functionality?

- Monitoring & Securing the service-to-service communication: Adding end to end encryption for service-to-service implementation and adding the monitoring capabilities

aws COMMUNITY DAY

Let me make it very Simple – A Dev Job

What if I ask Development team to build below functionality?

- Service level observability: Adding a features that allows team to monitor and analyze performance and behavior of the services in real time.

aws

COMMUNITY DAY

Focus is routed to implement additional layer

Now the problem here is, Development team is focusing on building the capability on the platform side instead of focusing on implementing core business logic.

aws

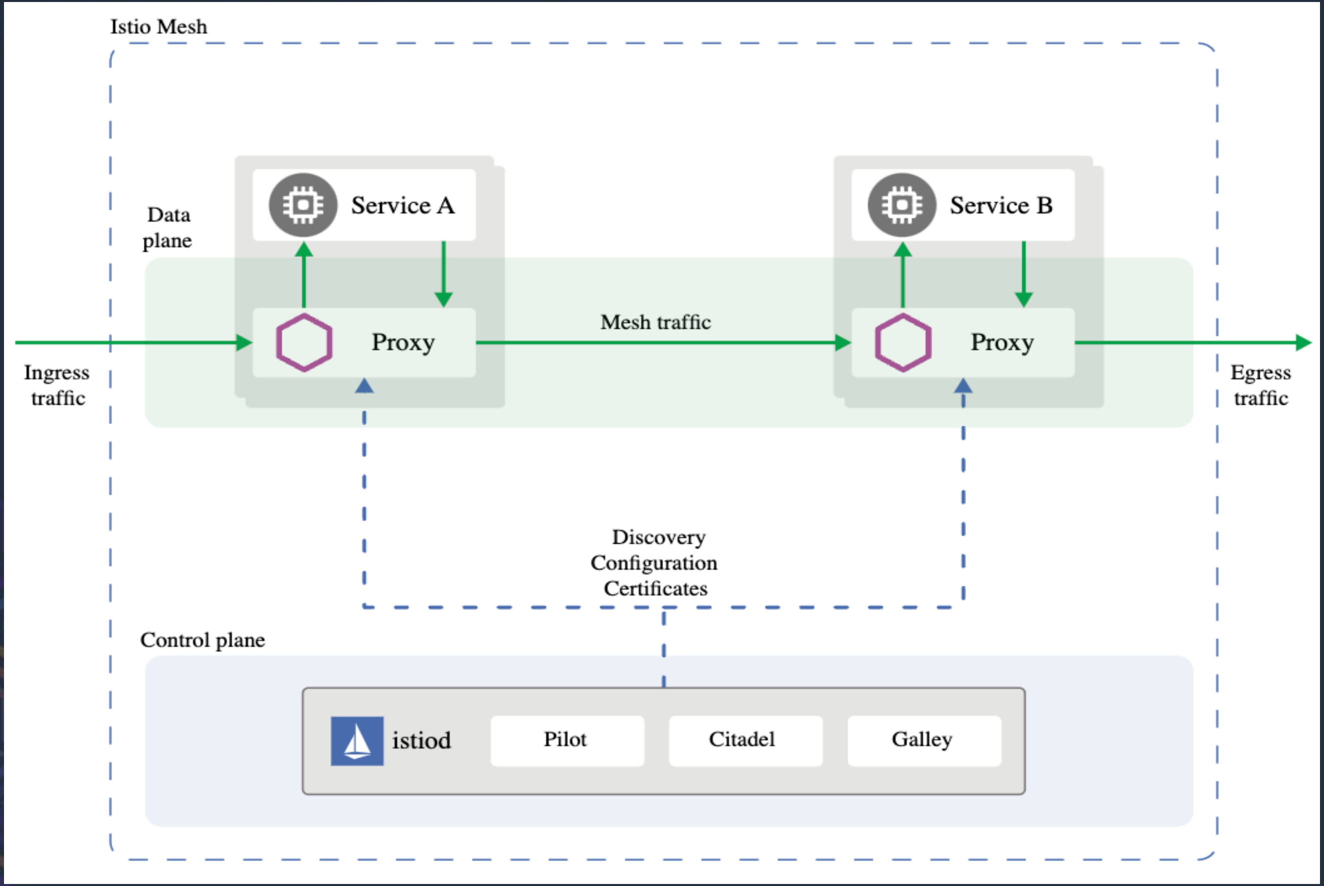
COMMUNITY DAY

What is the ISTIO?

Istio is an open-source service mesh platform that helps manage traffic between microservices within a Kubernetes cluster. It provides features such as load balancing, authentication, authorization, rate limiting, and observability.

Istio uses a sidecar proxy (Envoy) that intercepts all incoming and outgoing traffic to handle service-to-service communication. It’s designed to be platform-independent and can be used in various environments like Kubernetes, Mesos, and others.

Architecture ISTIO



aws COMMUNITY DAY

ISTIO Data Plane

The data plane consists of Envoy proxies deployed as sidecars, running alongside application instances in Kubernetes pods. The Envoy proxies manage traffic for services on the system, including managing and controlling network communication between microservices.

By deploying Envoy as a sidecar, Istio lets developers implement proxies in their application with no code changes

All application traffic flows through these Envoy proxies, which collect large amounts of data and can provide valuable insights about traffic, supporting observability.

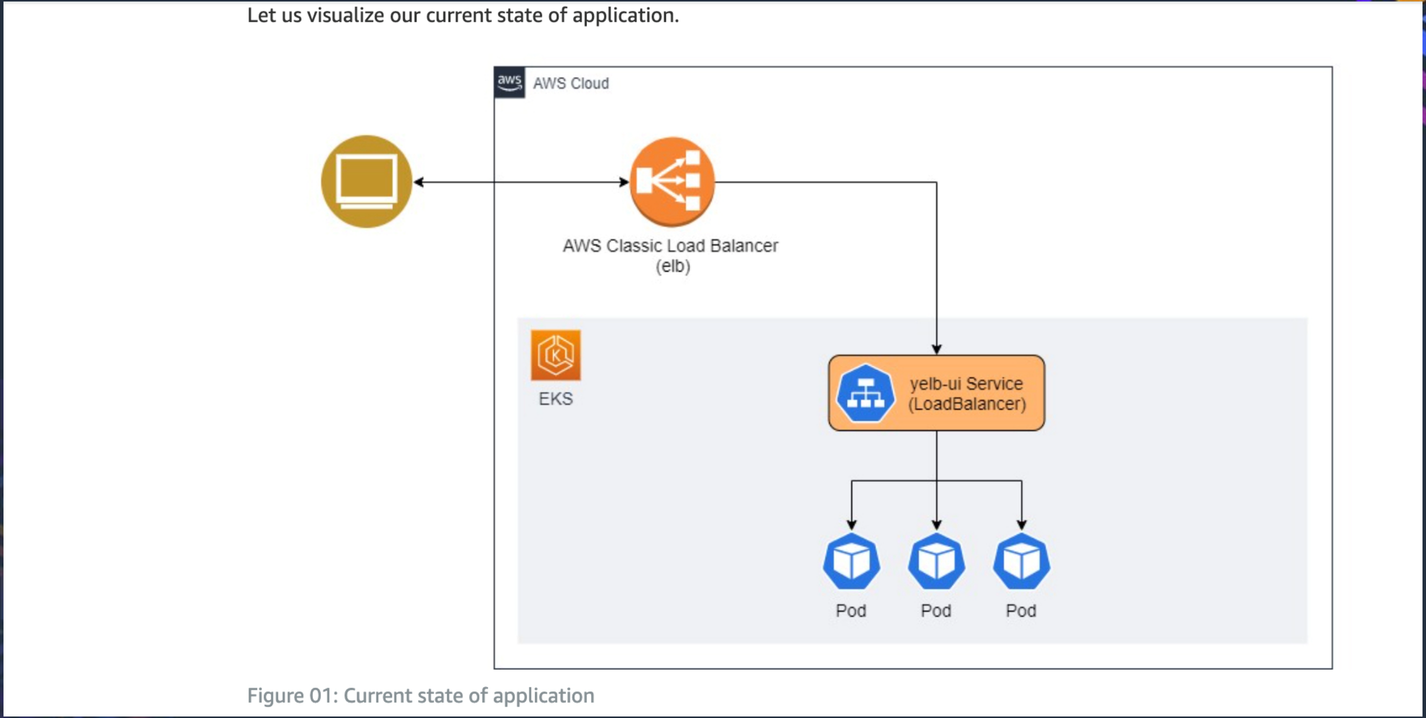
aws COMMUNITY DAY

ISTIO Control Plane

- Pilot: uses the Envoy API to communicate with the Envoy sidecar. The pilot is responsible for traffic management, routing, and service inspection.
- Citadel: provides secure communication between services by managing user authentication, certificate and credential management.
- Galley: responsible for configuration management, distribution, and processing.

COMMUNITY DAY

Insecure Architecture



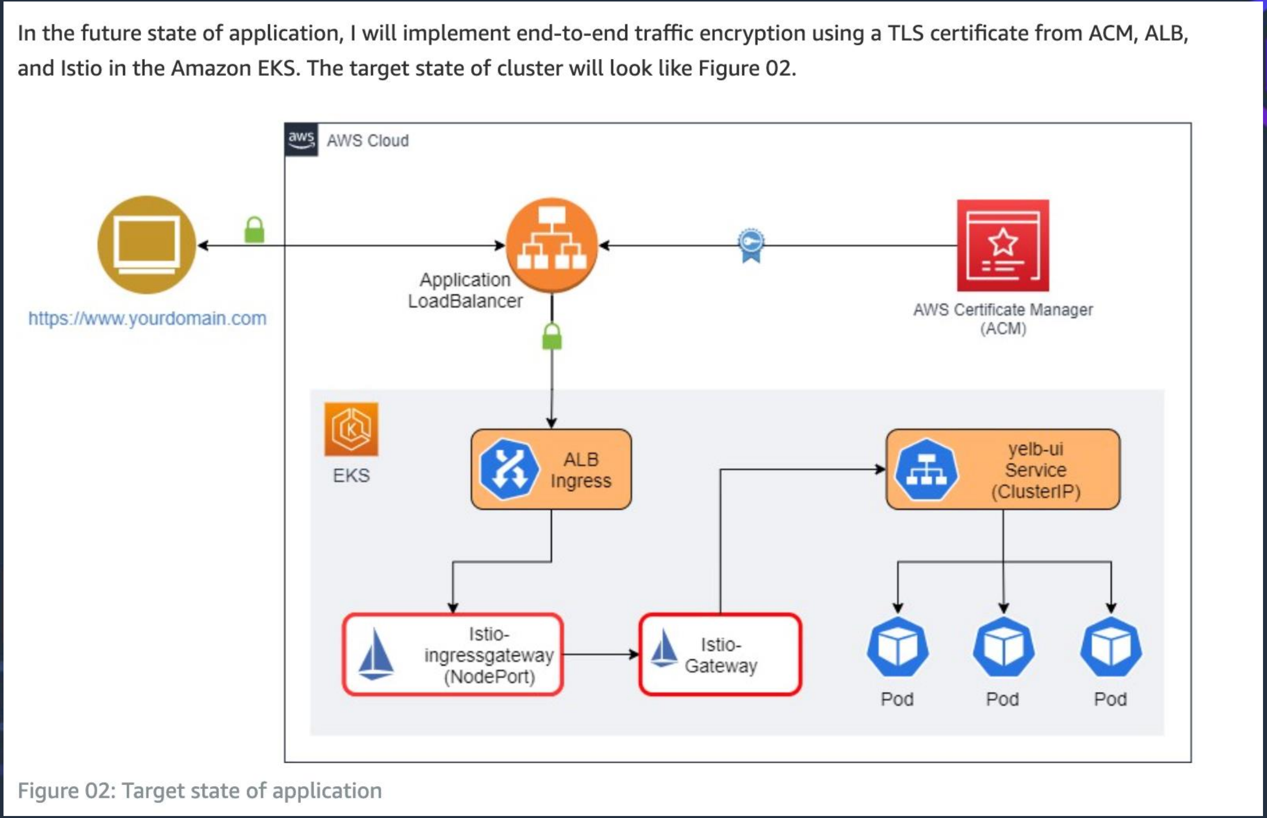
Let us visualize our current state of application.

AWS Cloud
AWS Classic Load Balancer (elb)
EKS
yelb-ui Service (LoadBalancer)
Pod
Pod
Pod

Figure 01: Current state of application

COMMUNITY DAY

secure Architecture



In the future state of application, I will implement end-to-end traffic encryption using a TLS certificate from ACM, ALB, and Istio in the Amazon EKS. The target state of cluster will look like Figure 02.

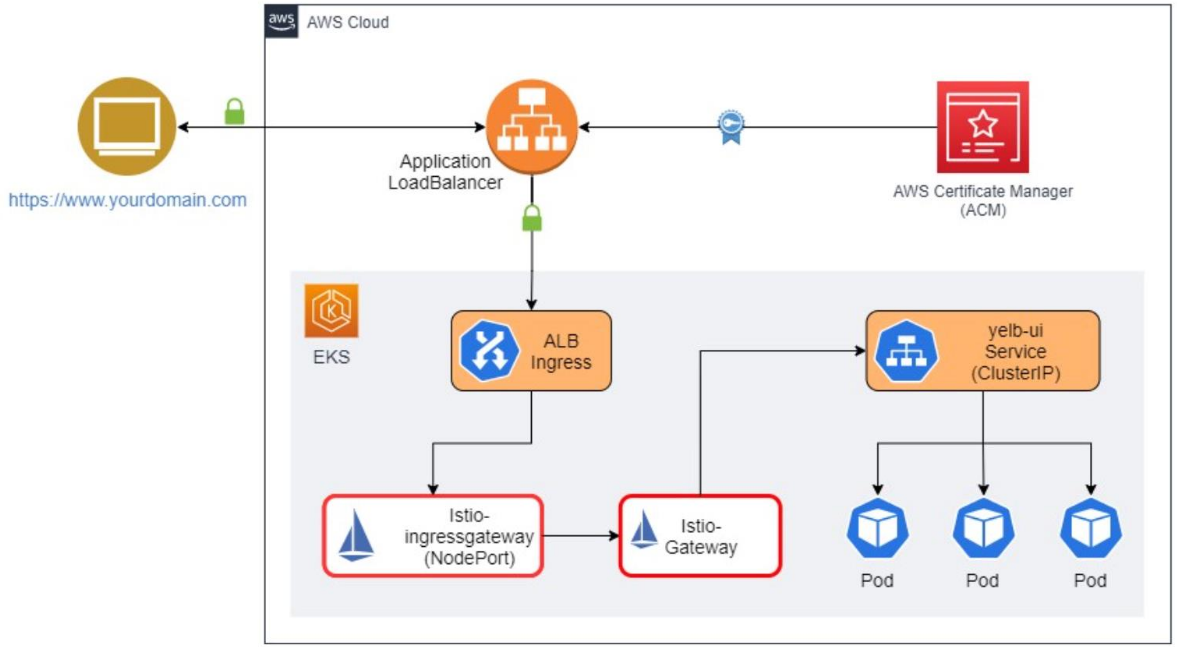

AWS Cloud
https://www.yourdomain.com
Application LoadBalancer
AWS Certificate Manager (ACM)
EKS
ALB Ingress
yelb-ui Service (ClusterIP)
Istio-ingressgateway (NodePort)
Istio-Gateway
Pod
Pod
Pod

Figure 02: Target state of application

COMMUNITY DAY

Pre-Requisite

- Working EKS Cluster
- SSL Certificate to Import on Amazon Certificate Manager
- aws-cli, openssl, helm, eksctl, kubectl installed
- Ingress Controller Deployed on EKS Cluster

aws COMMUNITY DAY

ISTIO Installation & Configuration

Step 1) Installation of ISTIO

```
# curl -L https://istio.io/downloadIstio | sh -
# cd istio-1.20.0
# export PATH=$PWD/bin:$PATH
```

Step 2) Configuration of the ISTIO Profile

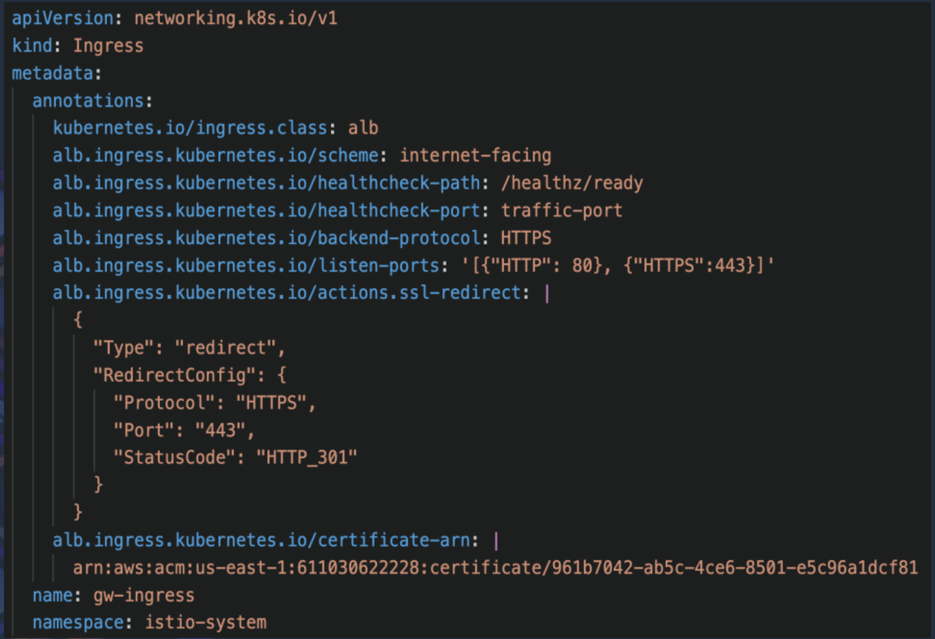
```
# istioctl install \
        --set profile=demo \
        --set values.gateways.istio-ingressgateway.type=NodePort
```

Step 3) Add a namespace label to instruct Istio to automatically inject Envoy sidecar proxies when you deploy your application later (awscommunity is the name of the namespace)

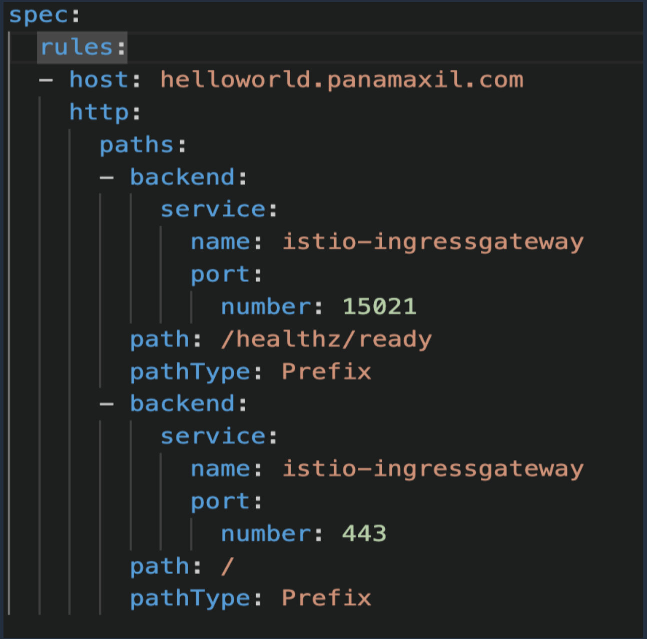
```
# kubectl label namespace awscommunity istio-injection=enabled
```

aws
COMMUNITY DAY

Ingress Controller Deployment



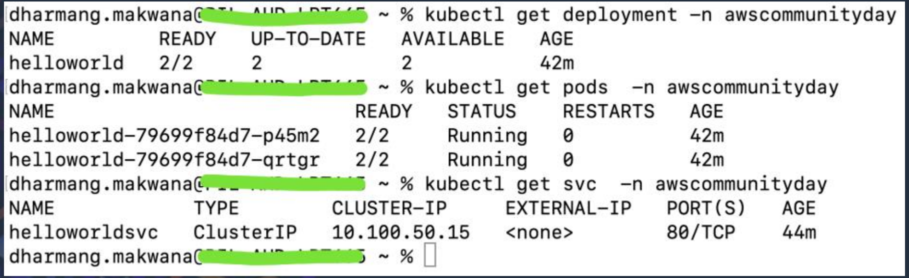
```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/healthcheck-path: /healthz/ready
    alb.ingress.kubernetes.io/healthcheck-port: traffic-port
    alb.ingress.kubernetes.io/backend-protocol: HTTPS
    alb.ingress.kubernetes.io/listen-ports: '[{"HTTP": 80}, {"HTTPS":443}]'
    alb.ingress.kubernetes.io/actions.ssl-redirect: |
      {
        "Type": "redirect",
        "RedirectConfig": {
          "Protocol": "HTTPS",
          "Port": "443",
          "StatusCode": "HTTP_301"
        }
      }
    alb.ingress.kubernetes.io/certificate-arn: |
      arn:aws:acm:us-east-1:611030622228:certificate/961b7042-ab5c-4ce6-8501-e5c96a1dcf81
  name: gw-ingress
  namespace: istio-system
```



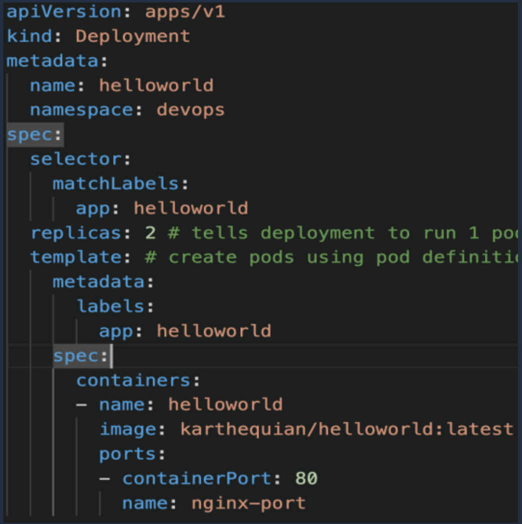
```yaml
spec:
  rules:
  - host: helloworld.panamaxil.com
    http:
      paths:
      - backend:
          service:
            name: istio-ingressgateway
            port:
              number: 15021
        path: /healthz/ready
        pathType: Prefix
      - backend:
          service:
            name: istio-ingressgateway
            port:
              number: 443
        path: /
        pathType: Prefix
```

Application Deployment

- For the Application Deployment we are using very simple application called helloworld.
- We are creating a simple deployment file and cluster IP service for the same.



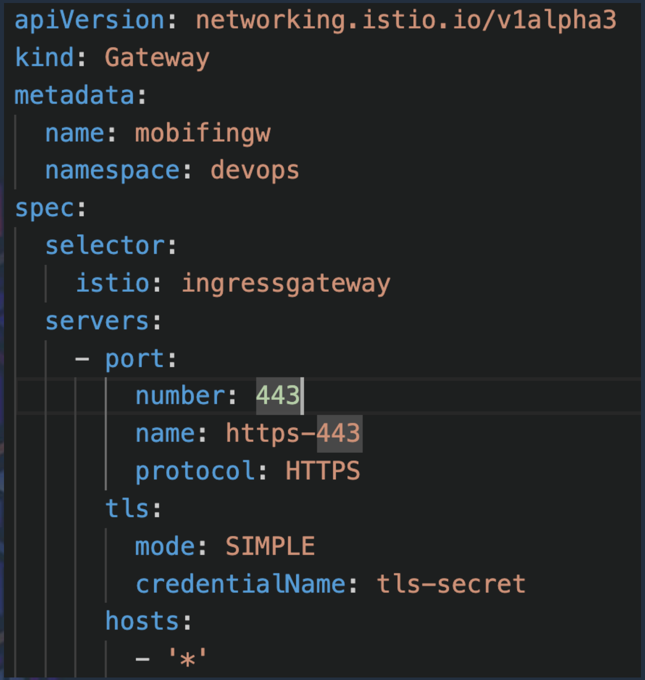
```
dharmang.makwana@[redacted] ~ % kubectl get deployment -n awscommunityday
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
helloworld   2/2     2            2           42m
dharmang.makwana@[redacted] ~ % kubectl get pods  -n awscommunityday
NAME                          READY   STATUS    RESTARTS   AGE
helloworld-79699f84d7-p45m2   2/2     Running   0          42m
helloworld-79699f84d7-qrtgr   2/2     Running   0          42m
dharmang.makwana@[redacted] ~ % kubectl get svc  -n awscommunityday
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
helloworldsvc   ClusterIP   10.100.50.15   <none>        80/TCP    44m
dharmang.makwana@[redacted] ~ %
```



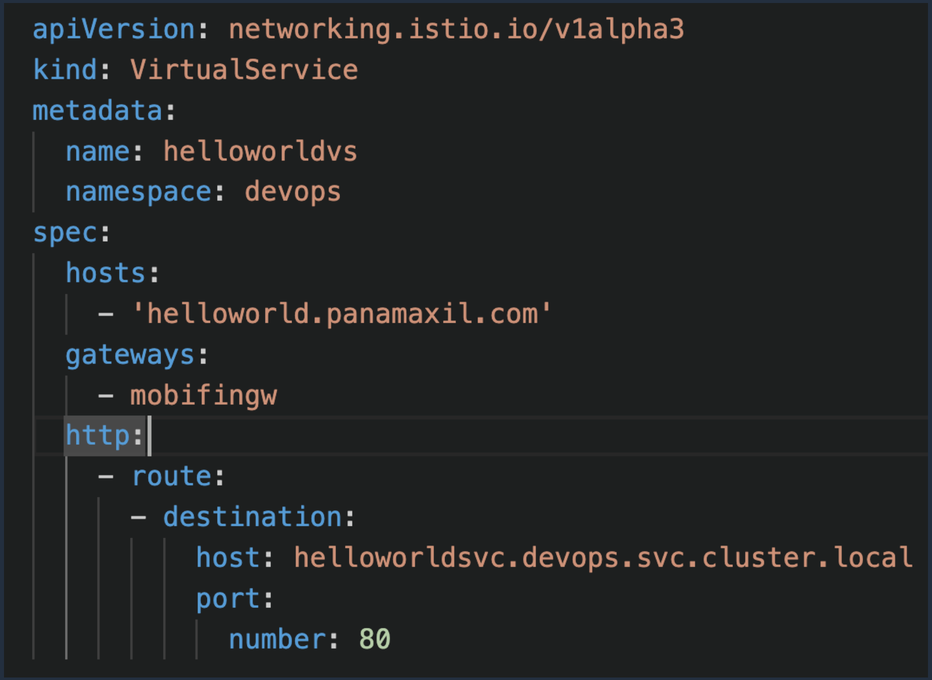
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: helloworld
  namespace: devops
spec:
  selector:
    matchLabels:
      app: helloworld
  replicas: 2 # tells deployment to run 1 po
  template: # create pods using pod definiti
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
      - name: helloworld
        image: karthequian/helloworld:latest
        ports:
        - containerPort: 80
          name: nginx-port
```

aws COMMUNITY DAY

Gate Way and Virtual Service Creation



```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: mobifingw
  namespace: devops
spec:
  selector:
    istio: ingressgateway
  servers:
    - port:
        number: 443
        name: https-443
        protocol: HTTPS
      tls:
        mode: SIMPLE
        credentialName: tls-secret
      hosts:
        - '*'
```



```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: helloworldvs
  namespace: devops
spec:
  hosts:
    - 'helloworld.panamaxil.com'
  gateways:
    - mobifingw
  http:
    - route:
      - destination:
          host: helloworldsvc.devops.svc.cluster.local
          port:
            number: 80
```

aws

COMMUNITY DAY

dharmangroy@gmail.com

https://github.com/dharmangmakwana/Istio-eks.git

+91 990 955 8715

https://www.linkedin.com/in/dharmang-makwana-2318a655/