



# kubernetes

---

*Kubernetes: Application on Istio*

## *KUBERNETES : MicroServices*

---

- With the Istio mesh in place and **configured to inject sidecar** Pods, we can create an application manifest with *specifications* for our Service and Deployment objects.
- We will create the Manifest File.
- Deploy the Manifest File.

## *KUBERNETES : MicroServices*

---

- To control access to a cluster and routing to Services, Kubernetes uses **Ingress Resources** and **Controllers**.
- **Istio** uses a different set of objects to achieve similar ends, though with some important differences. Instead of using a Controller to load balance traffic, the **Istio mesh** uses a **Gateway**, which functions as a load balancer that handles incoming and outgoing HTTP/TCP connections.
- The **Gateway** then allows for monitoring and routing rules to be applied to traffic entering the mesh. Specifically, the configuration that determines traffic routing is defined as a Virtual Service. Each Virtual Service includes routing rules that match criteria with a **specific protocol and destination**.
- To allow **external traffic** into our **mesh** and configure routing to our NodeJS app, we will need to create an **Istio Gateway** and **Virtual Service**.

## *KUBERNETES : MicroServices*

---

- First we will define the **Gateway Manifest** file.
- Deploy the **Istio Object Manifest**.
- Once you have created your application **Service** and **Deployment objects**, along with a **Gateway** and **Virtual Service**, you will be able to generate some requests to your application and look at the associated data in your **Istio Grafana dashboards**.
- First, however, you will need to configure Istio to expose the **Grafana addon** so that you can access the **dashboards** in your browser.
- Create **Manifest for a Gateway and Virtual Service** so that we can expose the **Grafana addon**.

## *KUBERNETES : MicroServices*

---

- Create your Grafana resources:
- Verify Gateway in istio-system NameSpace.  
`kubectl get gateway -n istio-system`  
`kubectl get virtualservice -n istio-system`
- Create Application on Cluster.
- Get Application Pods.
- Describe Application Pods.
- Create Application Gateway and Virtual Machine.

## *KUBERNETES : MicroServices*

---

- Verify Application on Load Balancer on Port 80.
- Verify Grafana Dashboard on Port 15031.

*Will see you in Next Lecture...*

---

*Thank you!*

A close-up photograph of a hand holding a black marker, writing the words 'Thank you!' in a cursive script on a white surface. The hand is positioned on the right side of the frame, with the marker tip touching the paper. The background is plain white.

*See you in next lecture ...*