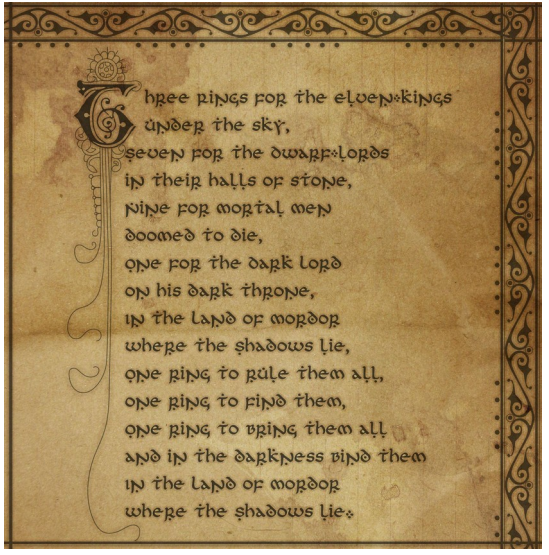
Welcome!!!

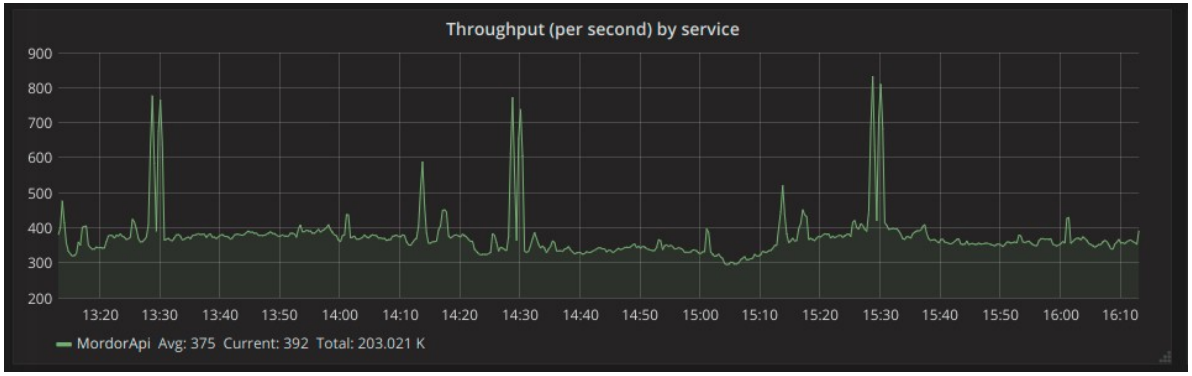
Topics :

- Ingress
- Nginx Ingress Modules
- Istio Ingress
- Istio Egress

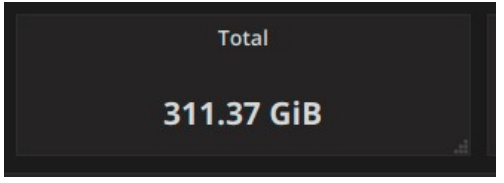


Team Mordor - K8s

```
boran@boran-HP-ZBook-15u-G3:~/workspace/hepsiburada/network$ kubectl ge
NAME                  STATUS    ROLES          AGE       VERSION
mordor-kube-agent-01    Ready     ingress,node   298d      v1.11.3
mordor-kube-agent-02    Ready     ingress,node   298d      v1.11.3
mordor-kube-agent-03    Ready     node           298d      v1.11.3
mordor-kube-agent-04    Ready     node           298d      v1.11.3
mordor-kube-agent-05    Ready     node           298d      v1.11.3
mordor-kube-agent-06    Ready     node           298d      v1.11.3
mordor-kube-agent-07    Ready     node           298d      v1.11.3
mordor-kube-agent-08    Ready     node           298d      v1.11.3
mordor-kube-agent-09    Ready     node           298d      v1.11.3
mordor-kube-agent-10    Ready     node           147d      v1.11.3
mordor-kube-agent-11    Ready     node           161d      v1.11.3
mordor-kube-agent-12    Ready     node           161d      v1.11.3
mordor-kube-agent-13    Ready     node           161d      v1.11.3
mordor-kube-agent-14    Ready     node           161d      v1.11.3
mordor-kube-agent-15    Ready     node           161d      v1.11.3
mordor-kube-agent-16    Ready     node           161d      v1.11.3
mordor-kube-agent-17    Ready     node           161d      v1.11.3
mordor-kube-agent-18    Ready     node           161d      v1.11.3
mordor-kube-agent-19    Ready     node           161d      v1.11.3
mordor-kube-agent-20    Ready     node           147d      v1.11.3
mordor-kube-agent-21    Ready     node           154d      v1.11.3
mordor-kube-agent-22    Ready     node           154d      v1.11.3
mordor-kube-agent-23    Ready     node           154d      v1.11.3
mordor-kube-agent-24    Ready     node           154d      v1.11.3
mordor-kube-agent-25    Ready     node           154d      v1.11.3
mordor-kube-agent-26    Ready     node           154d      v1.11.3
mordor-kube-agent-27    Ready     node           154d      v1.11.3
mordor-kube-agent-28    Ready     node           154d      v1.11.3
mordor-kube-agent-29    Ready     node           154d      v1.11.3
mordor-kube-agent-30    Ready     node           154d      v1.11.3
mordor-kube-agent-31    Ready     node           154d      v1.11.3
mordor-kube-agent-32    Ready     node           154d      v1.11.3
mordor-kube-agent-33    Ready     node           154d      v1.11.3
mordor-kube-agent-34    Ready     node           154d      v1.11.3
mordor-kube-agent-35    Ready     node           154d      v1.11.3
mordor-kube-agent-36    Ready     node           154d      v1.11.3
mordor-kube-agent-37    Ready     node           154d      v1.11.3
mordor-kube-agent-38    Ready     node           154d      v1.11.3
mordor-kube-master-01   Ready     master         298d      v1.11.3
mordor-kube-master-02   Ready     master         298d      v1.11.3
boran@boran-HP-ZBook-15u-G3:~/workspace/hepsiburada/network$
```

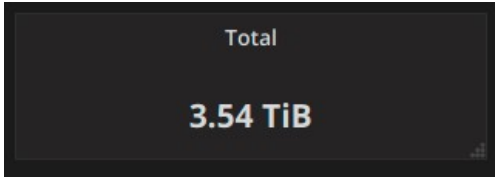


Ram

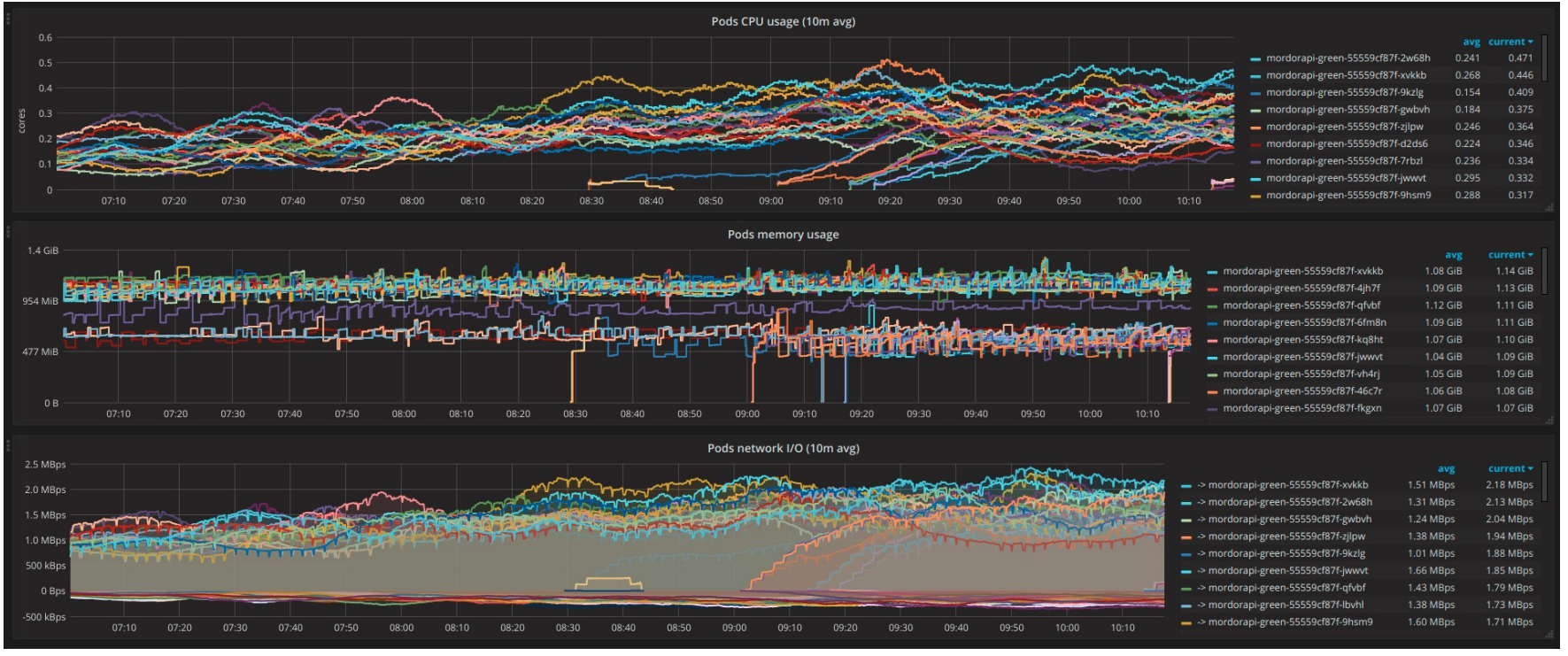


CPU

Disk



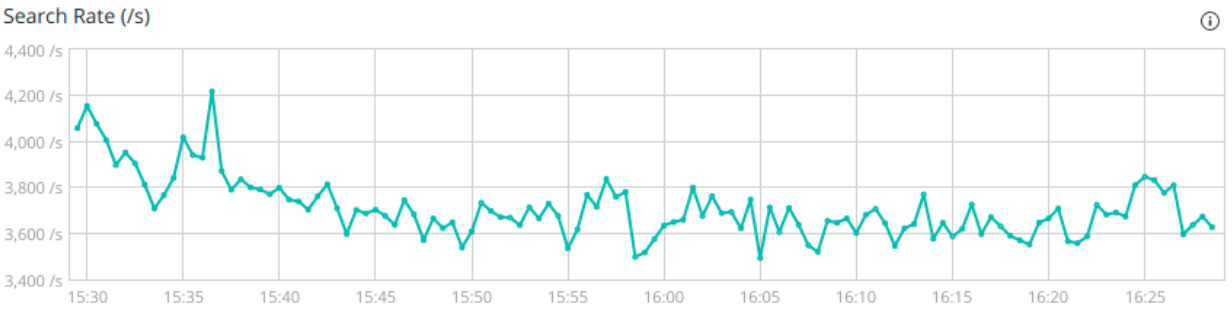
Team Mordor

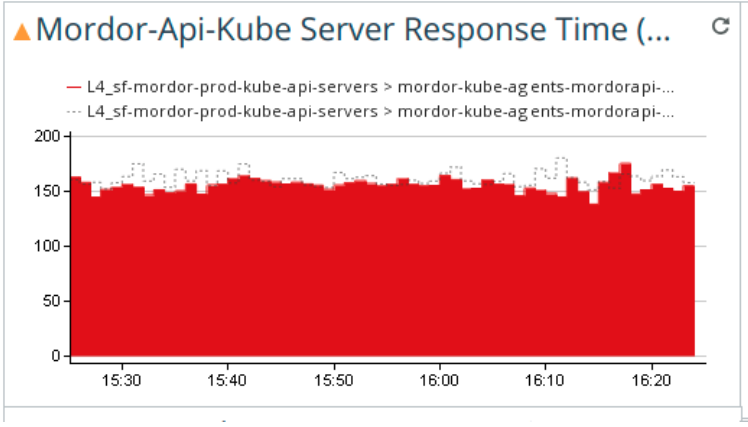

Pods CPU usage (10m avg)
cores
0.6
0.5
0.4
0.3
0.2
0.1
0
07:10 07:20 07:30 07:40 07:50 08:00 08:10 08:20 08:30 08:40 08:50 09:00 09:10 09:20 09:30 09:40 09:50 10:00 10:10
avg current
mordorapi-green-55559cf87f-2w68h 0.241 0.471
mordorapi-green-55559cf87f-xvkkb 0.268 0.446
mordorapi-green-55559cf87f-9kzlg 0.154 0.409
mordorapi-green-55559cf87f-gwbvh 0.184 0.375
mordorapi-green-55559cf87f-zjlpw 0.246 0.364
mordorapi-green-55559cf87f-d2ds6 0.224 0.346
mordorapi-green-55559cf87f-7rbzl 0.236 0.334
mordorapi-green-55559cf87f-jwwvt 0.295 0.332
mordorapi-green-55559cf87f-9hsm9 0.288 0.317
Pods memory usage
1.4 GiB
954 MiB
477 MiB
0 B
07:10 07:20 07:30 07:40 07:50 08:00 08:10 08:20 08:30 08:40 08:50 09:00 09:10 09:20 09:30 09:40 09:50 10:00 10:10
avg current
mordorapi-green-55559cf87f-xvkkb 1.08 GiB 1.14 GiB
mordorapi-green-55559cf87f-4jh7f 1.09 GiB 1.13 GiB
mordorapi-green-55559cf87f-qfvbf 1.12 GiB 1.11 GiB
mordorapi-green-55559cf87f-6fm8n 1.09 GiB 1.11 GiB
mordorapi-green-55559cf87f-kq8ht 1.07 GiB 1.10 GiB
mordorapi-green-55559cf87f-jwwvt 1.04 GiB 1.09 GiB
mordorapi-green-55559cf87f-vh4rj 1.05 GiB 1.09 GiB
mordorapi-green-55559cf87f-46c7r 1.06 GiB 1.08 GiB
mordorapi-green-55559cf87f-fkgxn 1.07 GiB 1.07 GiB
Pods network I/O (10m avg)
2.5 MBps
2.0 MBps
1.5 MBps
1.0 MBps
500 kBps
0 Bps
-500 kBps
07:10 07:20 07:30 07:40 07:50 08:00 08:10 08:20 08:30 08:40 08:50 09:00 09:10 09:20 09:30 09:40 09:50 10:00 10:10
avg current
-> mordorapi-green-55559cf87f-xvkkb 1.51 MBps 2.18 MBps
-> mordorapi-green-55559cf87f-2w68h 1.31 MBps 2.13 MBps
-> mordorapi-green-55559cf87f-gwbvh 1.24 MBps 2.04 MBps
-> mordorapi-green-55559cf87f-zjlpw 1.38 MBps 1.94 MBps
-> mordorapi-green-55559cf87f-9kzlg 1.01 MBps 1.88 MBps
-> mordorapi-green-55559cf87f-jwwvt 1.66 MBps 1.85 MBps
-> mordorapi-green-55559cf87f-qfvbf 1.43 MBps 1.79 MBps
-> mordorapi-green-55559cf87f-lbvhl 1.38 MBps 1.73 MBps
-> mordorapi-green-55559cf87f-9hsm9 1.60 MBps 1.71 MBps

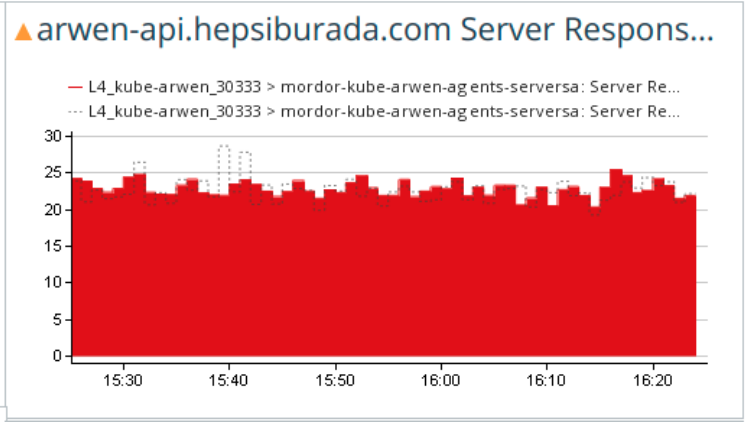
Team Mordor

Nodes: **27** Indices: **42** Memory: **175.3 GB / 428.1 GB** Total Shards: **221** Unassigned Shards: **0** Documents: **106,095,933** Data: **133.8 GB**

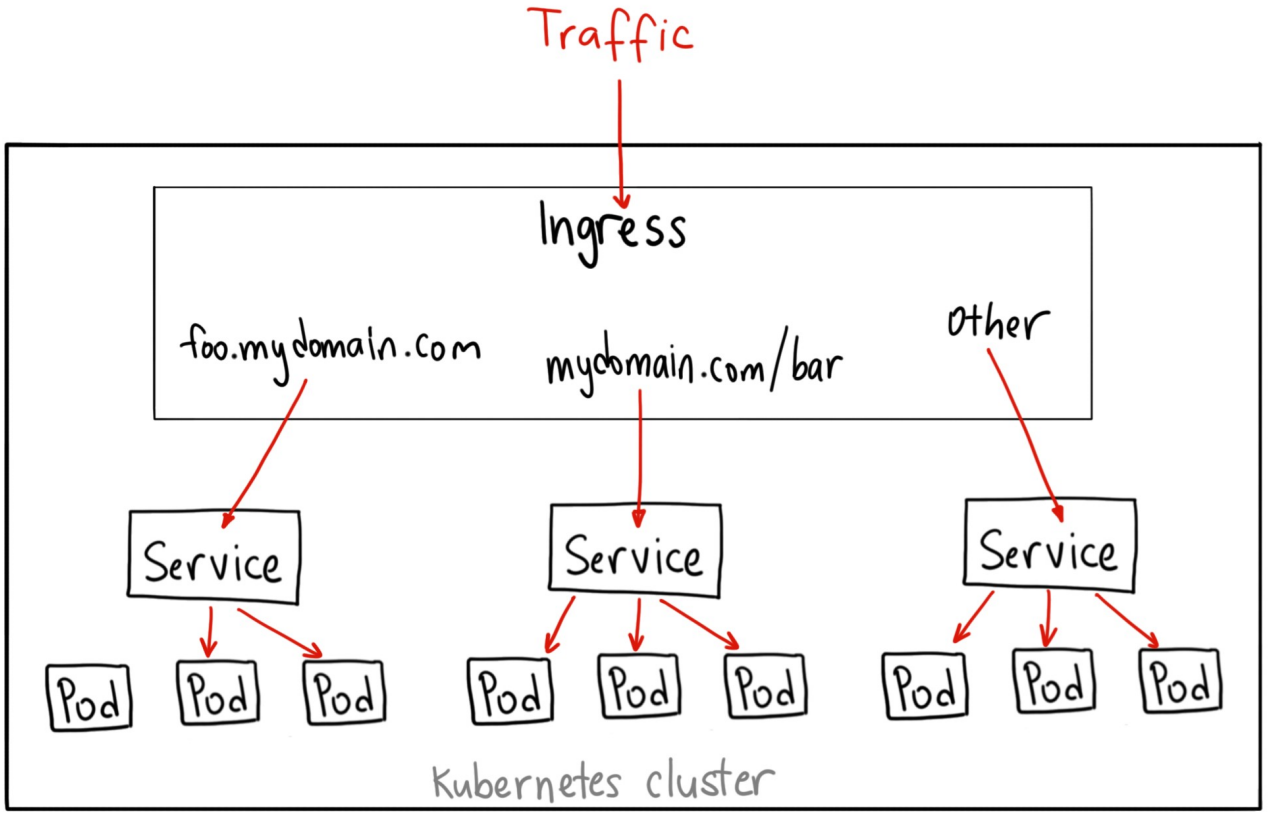
Nodes: **27** Indices: **35** Memory: **168.1 GB / 428.1 GB** Total Shards: **233** Unassigned Shards: **0** Documents: **206,510,695** Data: **193.5 GB**







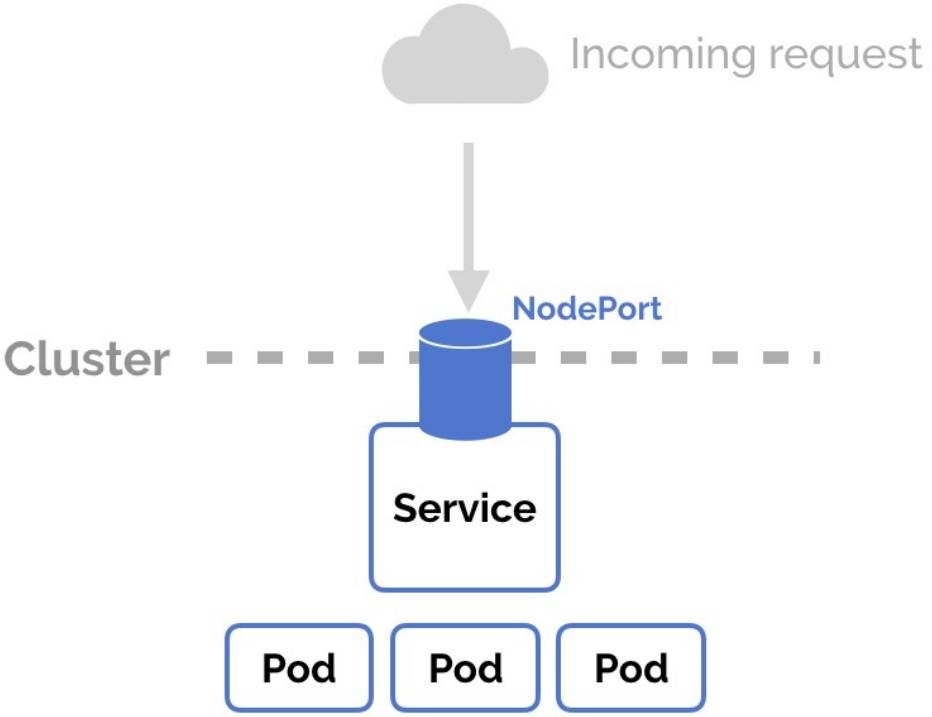
What is the Ingress ?



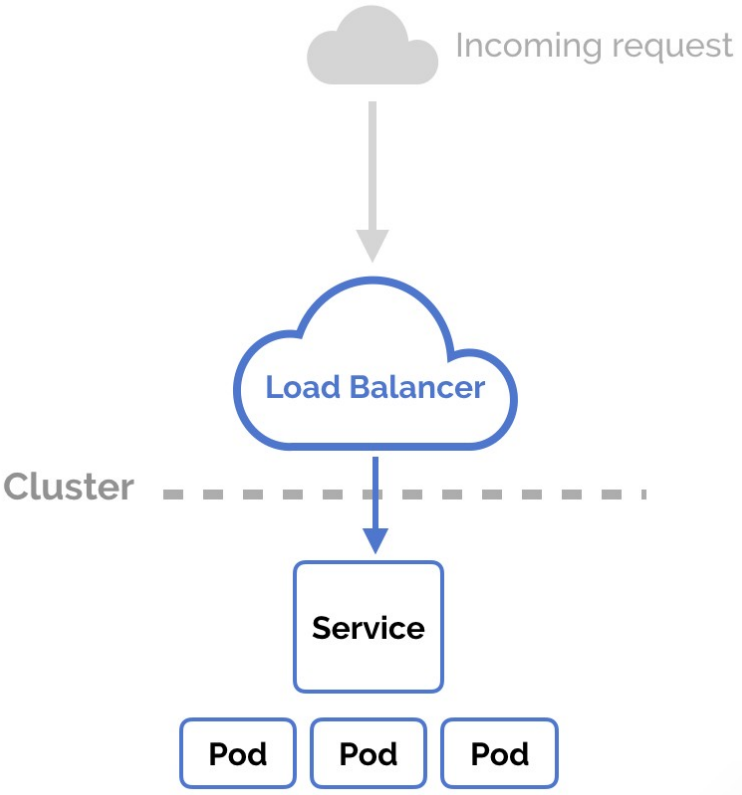
Kubernetes Ingress vs LoadBalancer vs NodePort

- What are the differences ?

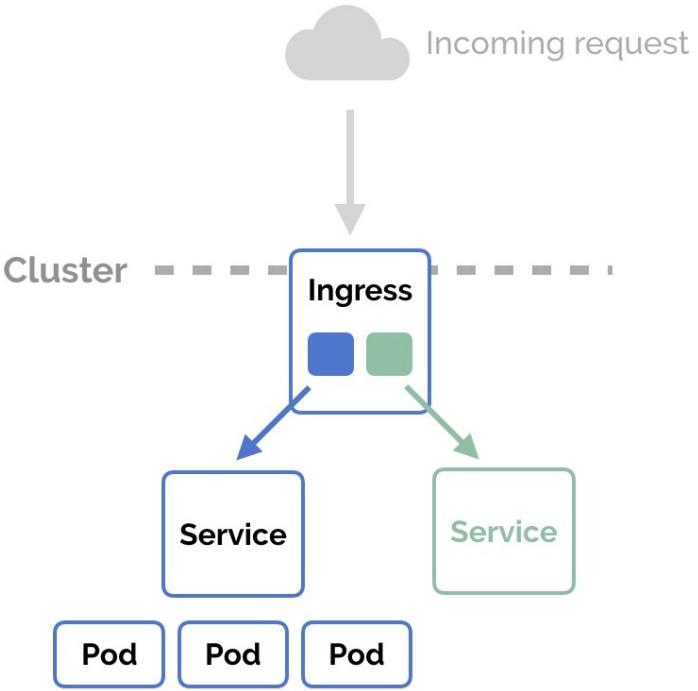
NodePort



LoadBalancer



Ingress


Incoming request
Cluster
Ingress
Service
Service
Pod
Pod
Pod

Scaling Ingress

- Daemon Set or Replicas
- Helm Carts or yamls

Nginx Ingress Modules

- community ingress
- nginxinc ingress

- Comparing two ingresses

Kubernetes Ingress

- Annotations

- Rate Limiting
- Canary Config Release
- Server snippet → Custom configs

Lets Encrypt with NGINX

- What is Let's Encrypt
- cert-manager

- kube-lego vs cert-manager

Cert Manager

- What is ingress-shim ?
- CRD’s
- Still not Prod-Ready (pre-1.0)

Lets Encrypt with NGINX

- Handzone!

Istio Gateway

- Unlike Kubernetes Ingress Resources, does not include any traffic routing configuration
- Traffic routing for ingress traffic is instead configured using Istio routing rules

- Deploy the App ‘http-bin’

The Istio “Gateway” Type

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: httpbin-gateway
spec:
  selector:
    istio: ingressgateway
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "*"
```

Exposing Ports via Gateway

- Expose port “80” via using Gateway
- kubectl get pod -n istio-system -l istio=ingressgateway -o name
- kubectl port-forward --address 0.0.0.0 -n istio-system istio-ingressgateway-774d77cb7c-j5gwf 15000
- curl localhost:15000/listeners
- curl http://localhost:15000/config_dump

Recap

- The Gateway: Istio Gateway’s are responsible for opening ports on relevant Istio gateway pods and receiving traffic for hosts. That’s it. However, they’re the critical link between received traffic and routing.

- The VirtualService: Istio VirtualService’s are what get “attached” to Gateways and are responsible defining the routes the gateway should implement. You can have multiple VirtualServices attached to Gateways. But not for the same domain.

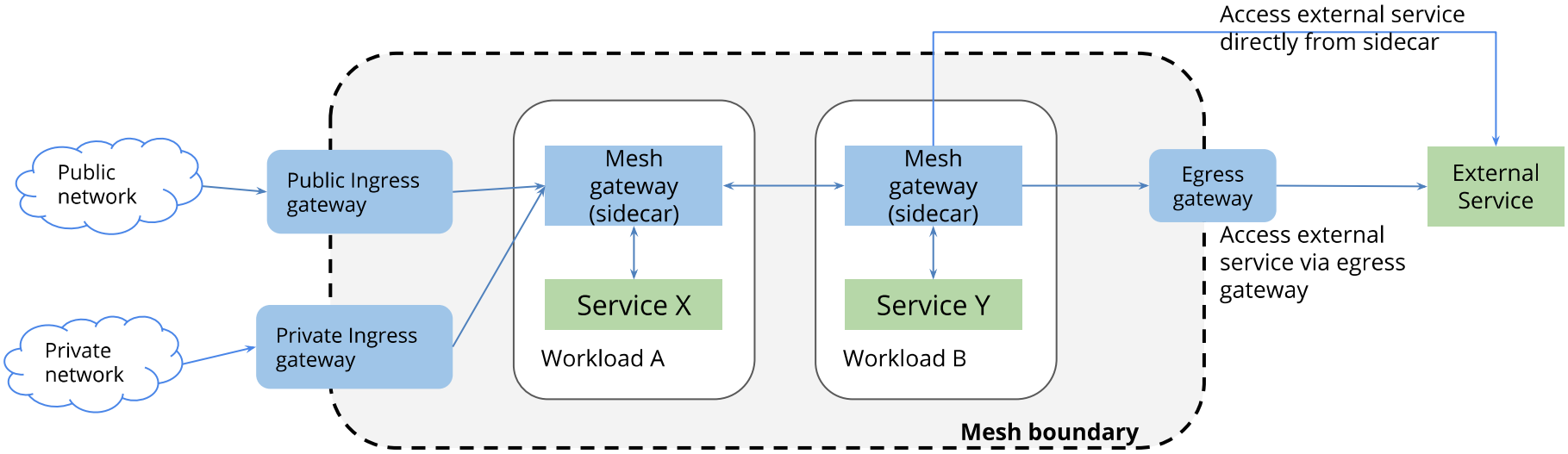
Istio Egress

- ServiceEntry ????
- ServiceEntry enables adding additional entries into Istio’s internal service registry
- location: MESH_INTERNAL or MESH_EXTERNAL

Istio Egress

- Example: ->

Istio Egress



Lets Encrypt with Istio

- Istio Gateway with cert-manager

Thanks :)

- https://github.com/boranx
- https://www.linkedin.com/in/boranseref/