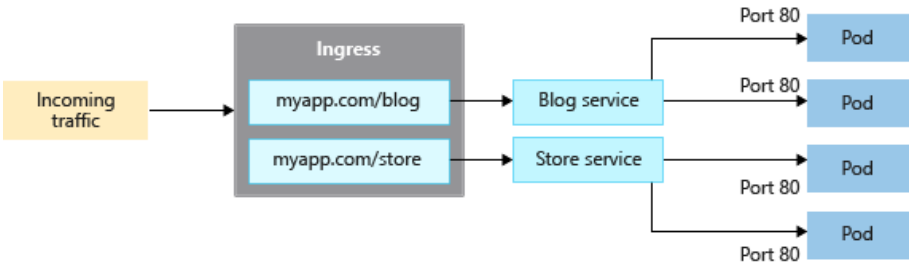
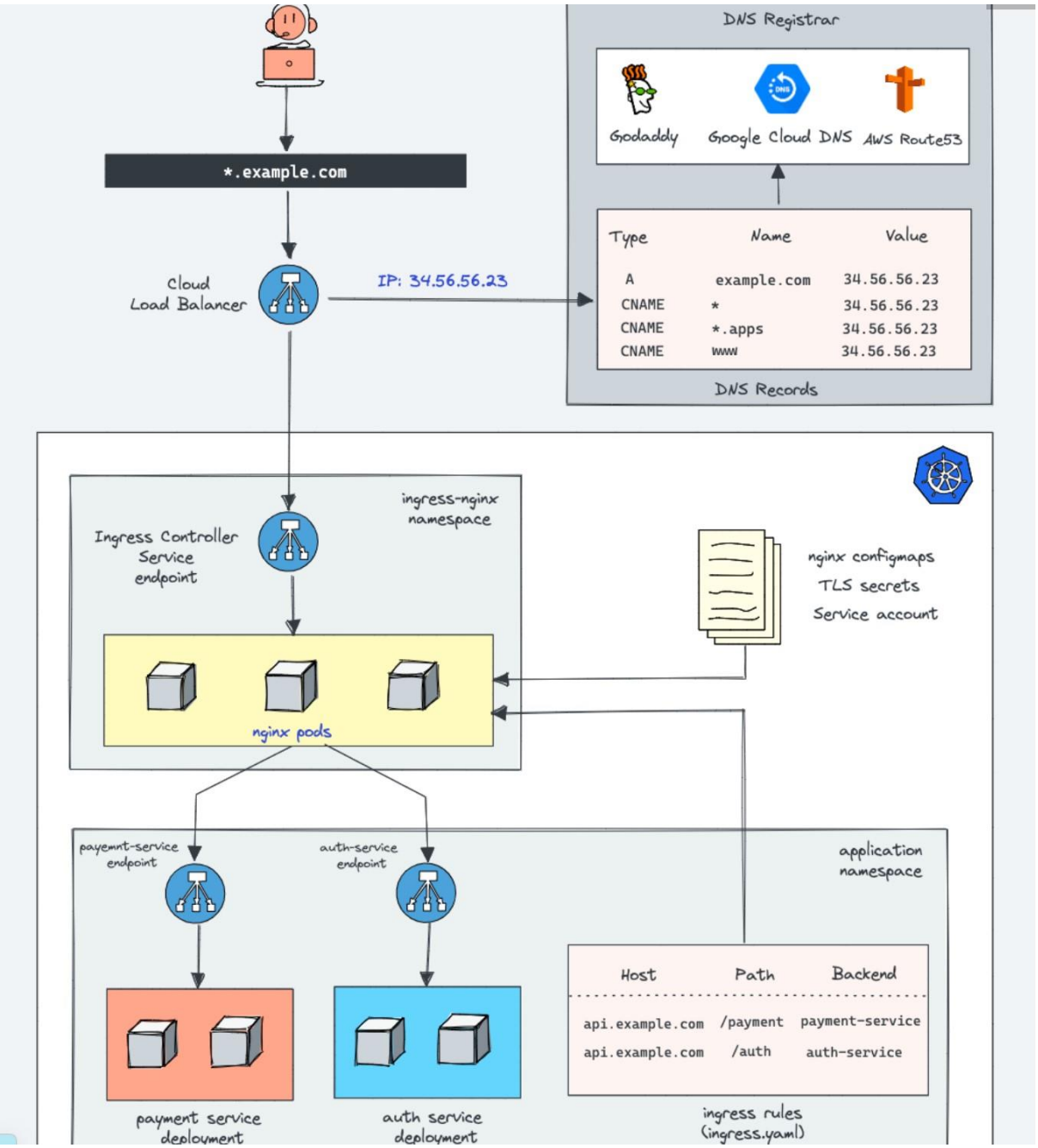
Agenda: Ingress

- Ingress Controllers and Alternatives
- Setting up Ingress Locally
- Creating the Ingress Config

Ingress Controller

- Ingress exposes **HTTP and HTTPS** routes from **outside the cluster** to services within the cluster. Traffic routing is controlled by **rules** defined on the Ingress resource.
- ***Ingress controllers*** work **at layer 7** (unlike LoadBalancer Service which works at layer 4), and can use more **intelligent rules** to distribute application traffic.
- Ingress actually acts as a **reverse proxy** to bring traffic into the cluster, then uses internal service routing to get the traffic where it is going.
- An Ingress may be configured to give Services externally-reachable URLs, load balance traffic, terminate SSL / TLS, and offer name-based virtual hosting.





In order for the Ingress resource to work, the cluster must have an ingress controller running.

(List of Ingress Controllers: https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/)

Ingress with NGINX Ingress Controller as a reverse proxy and load balancer

(https://kubernetes.github.io/ingress-nginx/deploy/).

Step1: Install NGINX Ingress controller

For Docker Desktop:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.0/deploy/static/provider/cloud/deploy.yaml
```

OR

For Minikube:

```
minikube addons enable ingress
```

Step2: Create a YAMLs file as below

nginx.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mynginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mynginx
  template:
    metadata:
      labels:
        app: mynginx
    spec:
      containers:
      - image: nginx
        name: mynginx
---

apiVersion: v1
kind: Service
metadata:
  name: mynginx-cip
spec:
  type: ClusterIP
  ports:
  - port: 8090
    protocol: TCP
    targetPort: 80
  selector:
    app: mynginx
```

kubectl apply -f nginx.yaml

httpd.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: myhttpd
  name: myhttpd
spec:
  replicas: 1
  selector:
    matchLabels:
      app: myhttpd
  template:
    metadata:
      labels:
        app: myhttpd
    spec:
      containers:
      - image: httpd
        name: myhttpd
---
apiVersion: v1
kind: Service
metadata:
  name: myhttpd-cip
spec:
  type: ClusterIP
  ports:
  - port: 8090
    protocol: TCP
    targetPort: 80
  selector:
    app: myhttpd
```

kubectl apply -f httpd.yaml

Ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
```

```
 name: my-ingress
 annotations:
  nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 ingressClassName: nginx
 defaultBackend:
  service:
   name: mynginx-cip
   port:
    number: 8090
 rules:
 - host: demo.mydomain.com
  http:
   paths:
   - path: /nginx
    pathType: Prefix
    backend:
     service:
      name: mynginx-cip
      port:
       number: 8090
   - path: /httpd
    pathType: Prefix
    backend:
     service:
      name: myhttpd-cip
      port:
       number: 8090
 - host: demo2.mydomain.com
  http:
   paths:
   - path: /A
    pathType: Prefix
    backend:
     service:
      name: A-cip
      port:
       number: 8090
```

```
  - path: /B
    pathType: Prefix
    backend:
      service:
        name: B-cip
        port:
          number: 8090
```

kubectl apply -f Ingress.yaml

Step3: Test the Ingress

Note: Stop IIS or Apache or Tomcat or any other service running on your machine on Port 80.

minikube ip

curl --header 'Host: demo.mydomain.com' http://192.168.49.2:80/nginx

curl --header 'Host: demo.mydomain.com' http://192.168.49.2:80/httpd

curl --header 'Host: demo.mydomain.com' http://192.168.49.2:80

Note: In the above URL: 192.168.49.2 is the Minikube IP Address.

Docker Desktop

curl --header "Host: demo.mydomain.com" http://localhost:80/nginx

curl --header "Host: demo.mydomain.com" http://localhost:80/httpd

curl --header "Host: demo.mydomain.com" http://locahost:80

OR

#For Mac / Linux do the following

sudo nano /etc/hosts

#For Windows do the following

Open Notepad as Administrator and Open File: c:\windows\system32\drivers\etc\hosts

Edit file and add below line and save.

192.168.49.2 demo.mydomain.com #Use this for Minikube

127.0.0.1 demo.mydomain.com #Use this for Docker Desktop

Step4: Open in browser following URL's (**may require to wait for couple of minutes**). Also stop other WebService Services running on Port 80.

Test using the below URL in browser:

curl http://demo.mydomain.com

curl http://demo.mydomain.com/nginx

curl http://demo.mydomain.com/httpd

To View the Logs and the POD to which the traffic is forward:

kubectl get pods **-n ingress-nginx**

kubectl logs **-n ingress-nginx** pod/ingress-nginx-controller-XXXXXXX-xxxx