Summary

- Ingress provides routing rules to manage external users' access to the services in a Kubernetes cluster via HTTPS/HTTP protocol.

- In Ingress, we can set up rules for routing traffic

- Ingress is a resource in Kubernetes for path-based routing

- An ingress controller is a kind of load balancer used for managing containerized applications

- Practical:- Creating rules for the Ingress controller

  o Adding addon in minikube for Nginix ingress controller

```
C:\Users\Vimal Daga>minikube  addons enable ingress
  - Using image k8s.gcr.io/ingress-nginx/controller:v1.0.4
  - Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
  - Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
* Verifying ingress addon...
* The 'ingress' addon is enabled

C:\Users\Vimal Daga>kubectl get pods
No resources found in default namespace.

C:\Users\Vimal Daga>kubectl get ns
NAME              STATUS   AGE
default           Active   86d
ingress-nginx     Active   64s
kube-node-lease   Active   86d
kube-public       Active   86d
kube-system       Active   86d
teama             Active   37d
testing           Active   14d
```
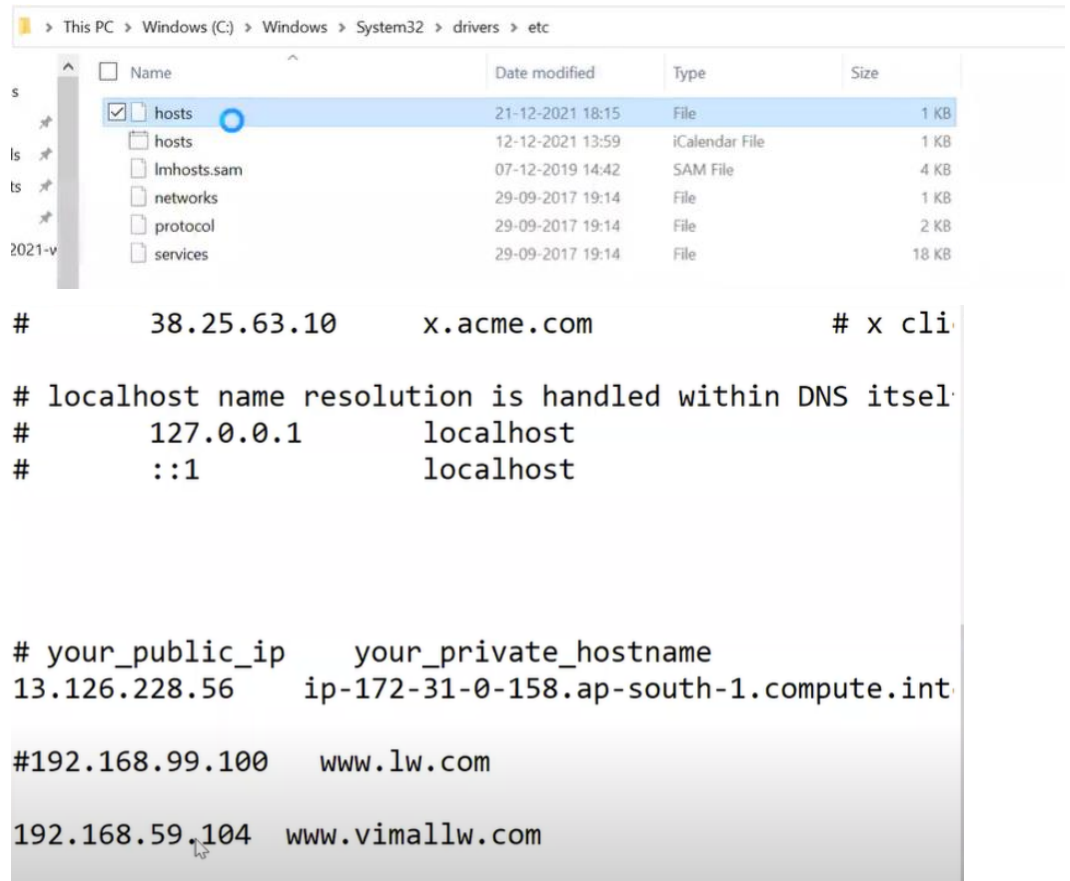
  o Creating deployment & exposing service

```
C:\Users\Vimal Daga>kubectl  create deployment mysearchd --image=httpd
deployment.apps/mysearchd created

C:\Users\Vimal Daga>kubectl expose deployment mysearchd   --type=NodePort --port=80
service/mysearchd exposed

C:\Users\Vimal Daga>kubectl get svc mysearchd
NAME        TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
mysearchd   NodePort   10.107.202.157  <none>        80:31713/TCP   16s
```
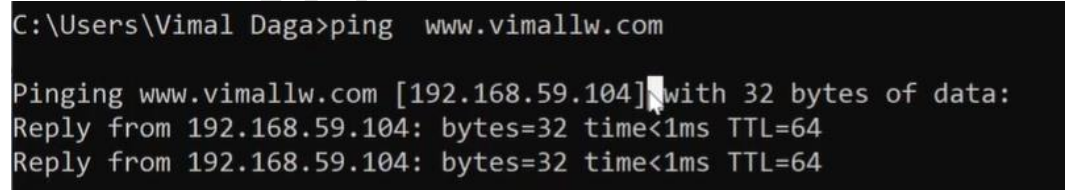
  o Setting local DNS

    ▪ Adding host

```
#         38.25.63.10        x.acme.com               # x cli

# localhost name resolution is handled within DNS itsel
#         127.0.0.1          localhost
#         ::1                localhost




# your_public_ip     your_private_hostname
13.126.228.56      ip-172-31-0-158.ap-south-1.compute.int

#192.168.99.100     www.lw.com

192.168.59.104   www.vimallw.com
```

- Pinging with the hostname

```
C:\Users\Vimal Daga>ping  www.vimallw.com

Pinging www.vimallw.com [192.168.59.104] with 32 bytes of data:
Reply from 192.168.59.104: bytes=32 time<1ms TTL=64
Reply from 192.168.59.104: bytes=32 time<1ms TTL=64
```

- o Creating ingress
  - Manifest file for ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: mywebpath
spec:
  rules:
  - host:  www.vimallw.com
    http:
      paths:
      - path: /search
        pathType: Prefix
        backend:
          service:
            name: mysearchd
            port:
              number: 80
```

```
    - path: /mail
      pathType: Prefix
      backend:
        service:        I
          name: mymaild
          port:
            number: 80
```

- Creating ingress from the manifest file

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl create -f ingress_nginx.yml
ingress.networking.k8s.io/mywebpath created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get ingress
NAME          CLASS   HOSTS             ADDRESS   PORTS   AGE
mywebpath     nginx   www.vimallw.com             80      12s
```

- Checking paths in ingress

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl describe ingress
Name:             mywebpath
Namespace:        default
Address:          localhost
Default backend:  default-http-backend:80 (<error: endpoints "default-http-backend" not found>)
Rules:
  Host             Path  Backends
  ----             ----  --------
  www.vimallw.com
                   /search   mysearchd:80 (172.17.0.6:80)
                   /mail     mymaild:80 (172.17.0.7:80)
Annotations:       <none>
Events:
  Type    Reason  Age         From              Message
```