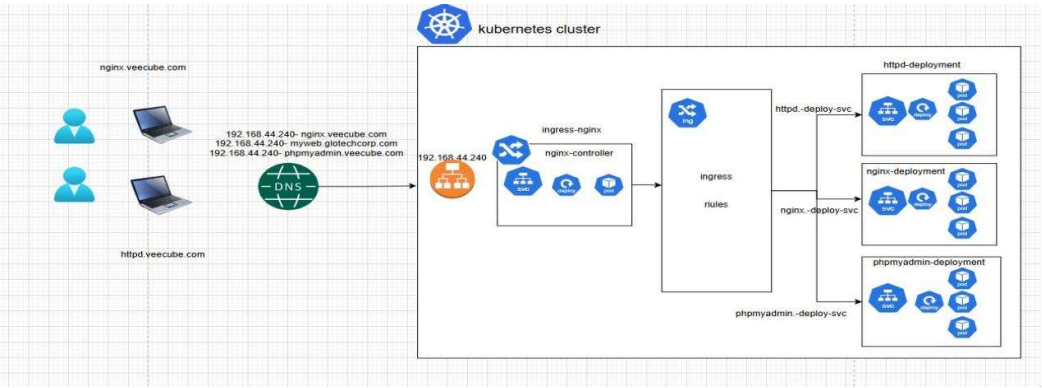
Ingress Definition:

- In Kubernetes, an Ingress is an API object that manages external access to services within a cluster, typically HTTP or HTTPS.
- It allows routing of traffic based on hostnames or URL paths to different services.

Flow chart:



Types of ingress controller:

We use nginx controller.

Ingress Controller	Proxy Technology	Key Features	Use Case
NGINX	NGINX	High customization, SSL/TLS termination	General-purpose, enterprise-level
Traefik	Traefik	Dynamic routing, auto SSL with Let's Encrypt	Microservices, automated setups
HAProxy	HAProxy	High performance, advanced load balancing	High-traffic, low-latency needs
Contour	Envoy	Advanced traffic management, observability	Cloud-native applications
Istio Ingress Gateway	Envoy	Service mesh integration, security	Service mesh environments
Kong	Kong Proxy	API management, rate limiting	API gateway, security, and analytics
Azure AGIC	Azure App Gateway	Azure-native, WAF, scalable solution	Azure-specific Kubernetes setups
AWS ALB Ingress	AWS ALB	AWS-n[illegible], service discovery	AWS-specific Kubernetes setups

1. **Create MetalLB:**

#kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.14.8/config/manifests/metallb-native.yaml

```
controlplane $ kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.14.8/config/manifests/metallb-native.yaml
namespace/metallb-system created
customresourcedefinition.apiextensions.k8s.io/bfdprofiles.metallb.io created
customresourcedefinition.apiextensions.k8s.io/bgpadvertisements.metallb.io created
customresourcedefinition.apiextensions.k8s.io/bgppeers.metallb.io created
customresourcedefinition.apiextensions.k8s.io/communities.metallb.io created
customresourcedefinition.apiextensions.k8s.io/ipaddresspools.metallb.io created
customresourcedefinition.apiextensions.k8s.io/l2advertisements.metallb.io created
customresourcedefinition.apiextensions.k8s.io/servicel2statuses.metallb.io created
serviceaccount/controller created
serviceaccount/speaker created
role.rbac.authorization.k8s.io/controller created
role.rbac.authorization.k8s.io/pod-lister created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/controller created
rolebinding.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
configmap/metallb-excludel2 created
secret/metallb-webhook-cert created
service/metallb-webhook-service created
deployment.apps/controller created
daemonset.apps/speaker created
validatingwebhookconfiguration.admissionregistration.k8s.io/metallb-webhook-configuration created
```

1. **To check the MetalLB status:**

   #kubectl get pods –A

```
controlplane $ kubectl get pods -A
NAMESPACE           NAME                                       READY   STATUS    RESTARTS        AGE
kube-system         calico-kube-controllers-94fb6bc47-zb75s    1/1     Running   2 (6m33s ago)   26d
kube-system         canal-4bqsl                                2/2     Running   2 (6m32s ago)   26d
kube-system         canal-8px9n                                2/2     Running   2 (6m33s ago)   26d
kube-system         coredns-57888bfdc7-k8qpf                   1/1     Running   1 (6m32s ago)   26d
kube-system         coredns-57888bfdc7-ns8f8                   1/1     Running   1 (6m32s ago)   26d
kube-system         etcd-controlplane                          1/1     Running   2 (6m33s ago)   26d
kube-system         kube-apiserver-controlplane                1/1     Running   2 (6m33s ago)   26d
kube-system         kube-controller-manager-controlplane       1/1     Running   2 (6m33s ago)   26d
kube-system         kube-proxy-ldvw7                           1/1     Running   2 (6m33s ago)   26d
kube-system         kube-proxy-n7zdb                           1/1     Running   1 (6m32s ago)   26d
kube-system         kube-scheduler-controlplane                1/1     Running   2 (6m33s ago)   26d
local-path-storage  local-path-provisioner-6c5cff8948-tbl5l   1/1     Running   2 (6m33s ago)   26d
metallb-system      controller-8694df9d9b-fnppc                1/1     Running   0               3m42s
metallb-system      speaker-2trjd                              1/1     Running   0               3m42s
metallb-system      speaker-6v8bl                              1/1     Running   0               3m42s
```

2. **Create an ipaddresspool:**

   #vi ippool.yaml

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: default
  namespace: metallb-system
spec:
  addresses:
    - 192.168.1.240-192.168.1.250
```

```
controlplane $ kubectl create -f ippool.yaml
ipaddresspool.metallb.io/default created
```

3. **Deploying Nginx ingress controller**

   Copy to yaml file in the github

   This using URL enter in Google https://github.com/kubernetes/ingress-nginx/blob/controller-v1.9.6/deploy/static/provider/baremetal/deploy.yaml

   #vi ingress controller.yaml

   Change service only to modify the Load Balancer line is 365

```
  - appProtocol: https
    name: https
    port: 443
    protocol: TCP
    targetPort: https
  selector:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
  type: LoadBalancer
```

4. **Kubectl create -f ingress.yaml**

```
controlplane $ k create -f ippo.yaml
ipaddresspool.metallb.io/first-pool created
controlplane $ vi ippo.yaml
controlplane $ vi ingress.yaml
controlplane $ k create -f ingress.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
serviceaccount/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
configmap/ingress-nginx-controller created
service/ingress-nginx-controller created
service/ingress-nginx-controller-admission created
deployment.apps/ingress-nginx-controller created
job.batch/ingress-nginx-admission-create created
job.batch/ingress-nginx-admission-patch created
ingressclass.networking.k8s.io/nginx created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
controlplane $
```

5. **Now I check to Nginx install status**

```
controlplane $ k get svc -A
NAMESPACE        NAME                                 TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)
                     AGE
default          kubernetes                           ClusterIP      10.96.0.1        <none>        443/TCP
                     20d
ingress-nginx    ingress-nginx-controller             LoadBalancer   10.99.63.112     172.30.1.2    80:32628
/TCP,443:31262/TCP   9s
ingress-nginx    ingress-nginx-controller-admission   ClusterIP      10.111.253.125   <none>        443/TCP
                     9s
kube-system      kube-dns                             ClusterIP      10.96.0.10       <none>        53/UDP,5
3/TCP,9153/TCP       20d
metallb-system   metallb-webhook-service              ClusterIP      10.110.22.187    <none>        443/TCP
                     3m3s
controlplane $ k get pod -A
NAMESPACE            NAME                                   READY   STATUS      RESTARTS        AGE
ingress-nginx        ingress-nginx-admission-create-2d4wj   0/1     Completed   0               23s
ingress-nginx        ingress-nginx-admission-patch-zfkfk    0/1     Completed   0               23s
ingress-nginx        ingress-nginx-controller-d5794ff8b-hkqtt   0/1     Running     0               23s
```

6. **CREATE DEPLOYMENTS**

- kubectl create deployment httpd-deploy --image=httpd

```
controlplane $ kubectl create deployment httpd-deploy --image=httpd --replicas 3
deployment.apps/httpd-deploy created
```

7. kubectl create deployment phpmyadmin-deploy --image=phpMyAdmin

```
controlplane $ kubectl create deployment php-deploy --image=phpmyadmin --replicas 3
deployment.apps/php-deploy created
controlplane $
```

Create Services

- kubectl expose deployment httpd-deploy --name httpd-deploy-svc --type=NodePort --port 80 --target-port 80

```
controlplane $ kubectl expose deployment httpd-deploy --name httpd-deploy-svc --type=NodePort --port 80 --target-port 80
service/httpd-deploy-svc exposed
```

- kubectl expose deployment phpmyadmin-deploy --name phpmyadmin-deploy-svc --type NodePort --port 8080 --target-port 80

```
controlplane $ kubectl expose deployment phpmyadmin-deploy --name phpmyadmin-deploy-svc --type NodePort --port 8080 --target-port 80
service/phpmyadmin-deploy-svc exposed
```

Now to see the service created status

```
controlplane $ k get svc
NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
httpd-deploy-svc      NodePort    10.98.170.49     <none>        80:31392/TCP     69s
kubernetes            ClusterIP   10.96.0.1        <none>        443/TCP          20d
phpmyadmin-deploy-svc NodePort    10.105.198.152   <none>        8080:30902/TCP   44s
```

Types of Ingress:

- PATH Based Routing
- Name Based Routing

Name Based Routing:

vi namebased.yaml

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: namebased-kubernetes-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
  - host: "php.veecube.com"
    http:
      paths:
      - pathType: Prefix
        path: "/"
        backend:
          service:
            name: phpmyadmin-deploy-svc
            port:
              number: 8080
  - host: "httpd.veecube.com"
    http:
      paths:
      - pathType: Prefix
        path: "/"
        backend:
          service:
            name: httpd-deploy-svc
```

And seeing the Yaml file status

```
controlplane $ vi namebased.yaml
controlplane $ k create -f namebased.yaml
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/namebased-kubernetes-ingress created
controlplane $
```

Path Based Routing:

- vi pathbased.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: pathbased-kubernetes-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - host: "web.veecube.com"
    http:
      paths:
      - path: /httpd
        pathType: Prefix
        backend:
          service:
            name: httpd-deploy-svc
            port:
              number: 80
      - path: /php
        pathType: Prefix
        backend:
          service:
            name: phpmyadmin-deploy-svc
            port:
              number: 8080
"pathbased.yaml" 27L, 590C
```

And seeing the yaml file status

```
controlplane $ k create -f pathbased.yaml
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/pathbased-kubernetes-ingress created
controlplane $ k get pod
NAME                                 READY   STATUS    RESTARTS   AGE
httpd-deploy-7b6db9884c-x2lvb        1/1     Running   0          6m5s
phpmyadmin-deploy-58958656f4-ghblk   1/1     Running   0          5m50s
```

Now I check ingress controller external is and use that it in the /etc/hosts

```
controlplane $ k get svc -A
NAMESPACE       NAME                                 TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
default         httpd-deploy-svc                     NodePort       10.96.44.131     <none>        80:30860/TCP                 25m
default         kubernetes                           ClusterIP      10.96.0.1        <none>        443/TCP                      20d
default         phpmyadmin-deploy-svc                NodePort       10.102.196.136   <none>        8080:30948/TCP               24m
ingress-nginx   ingress-nginx-controller             LoadBalancer   10.101.141.228   172.30.1.2    80:32395/TCP,443:32732/TCP   26m
ingress-nginx   ingress-nginx-controller-admission   ClusterIP      10.107.177.138   <none>        443/TCP                      26m
kube-system     kube-dns                             ClusterIP      10.96.0.10       <none>        53/UDP,53/TCP,9153/TCP       20d
metallb-system  metallb-webhook-service              ClusterIP      10.103.187.31    <none>        443/TCP                      29m
controlplane $
```

```
controlplane $ k get svc -n ingress-nginx
NAME                                 TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
ingress-nginx-controller             LoadBalancer   10.101.141.228   172.30.1.2    80:32395/TCP,443:32732/TCP   39m
ingress-nginx-controller-admission   ClusterIP      10.107.177.138   <none>        443/TCP                      39m
```

To enter the configuration file in /etc/hosts

```
172.30.1.2 httpd.veecube.com
172.30.1.2 web.veecube.com
172.30.1.2 php.veecube.com
```

To check the Name based output status

```
controlplane $ curl httpd.veecube.com
<html><body><h1>It works!</h1></body></html>
controlplane $ curl php.veecube.com
<!doctype html>
<html lang="en" dir="ltr">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="referrer" content="no-referrer">
  <meta name="robots" content="noindex,nofollow,notranslate">
  <meta name="google" content="notranslate">
  <style id="cfs-style">html{display: none;}</style>
  <link rel="icon" href="favicon.ico" type="image/x-icon">
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
  <link rel="stylesheet" type="text/css" href="./themes/pmahomme/jquery/jquery-ui.css">
  <link rel="stylesheet" type="text/css" href="js/vendor/codemirror/lib/codemirror.css?v=5.2.1">
  <link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/hint/show-hint.css?v=5.2.1">
  <link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/lint/lint.css?v=5.2.1">
  <link rel="stylesheet" type="text/css" href="./themes/pmahomme/css/theme.css?v=5.2.1">
  <title>phpMyAdmin</title>
    <script data-cfasync="false" type="text/javascript" src="js/vendor/jquery/jquery.min.js?v=5.2.1"></scrip
t>
  <script data-cfasync="false" type="text/javascript" src="js/vendor/jquery/jquery-migrate.min.js?v=5.2.1"><
/script>
  <script data-cfasync="false" type="text/javascript" src="js/vendor/sprintf.js?v=5.2.1"></script>
```

To check the Path based output status

```
controlplane $ curl httpd.veecube.com
<html><body><h1>It works!</h1></body></html>
controlplane $ curl php.veecube.com
<!doctype html>
<html lang="en" dir="ltr">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="referrer" content="no-referrer">
  <meta name="robots" content="noindex,nofollow,notranslate">
  <meta name="google" content="notranslate">
  <style id="cfs-style">html{display: none;}</style>
  <link rel="icon" href="favicon.ico" type="image/x-icon">
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
  <link rel="stylesheet" type="text/css" href="./themes/pmahomme/jquery/jquery-ui.css">
  <link rel="stylesheet" type="text/css" href="js/vendor/codemirror/lib/codemirror.css?v=5.2.1">
  <link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/hint/show-hint.css?v=5.2.1">
  <link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/lint/lint.css?v=5.2.1">
  <link rel="stylesheet" type="text/css" href="./themes/pmahomme/css/theme.css?v=5.2.1">
  <title>phpMyAdmin</title>
    <script data-cfasync="false" type="text/javascript" src="js/vendor/jquery/jquery.min.js?v=5.2.1"></scrip
t>
  <script data-cfasync="false" type="text/javascript" src="js/vendor/jquery/jquery-migrate.min.js?v=5.2.1"><
/script>
  <script data-cfasync="false" type="text/javascript" src="js/vendor/sprintf.js?v=5.2.1"></script>
```

To seeing service status

```
NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
httpd-deploy-svc       NodePort    10.96.44.131     <none>        80:30860/TCP     9m1s
kubernetes             ClusterIP   10.96.0.1        <none>        443/TCP          20d
phpmyadmin-deploy-svc  NodePort    10.102.196.136   <none>        8080:30948/TCP   8m48s
```

To check the Output to access from URL use to Traffic port

1. **Apache Output status**

It works!

2. **Phpmyadmin Output status**