



coremain



Formación Kubernetes

## Índice



- 01 Introducción
- 02 Apartado teórico: Infraestructura
- 03 Apartado teórico: Recursos
- 04 Apartado práctico: Comandos básicos
- 05 Apartado práctico: Demostración
- 06 Prácticas propuestas





coremain

# Introducción



## ¿Qué es Kubernetes?

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa. Tiene un ecosistema grande y en rápido crecimiento. El soporte, las herramientas y los servicios para Kubernetes están ampliamente disponibles.

- ✓ Clúster
- ✓ Orquestador de contenedores



## ¿Por qué Kubernetes?

- ✓ Facilita la escalabilidad (tanto vertical como horizontal) y la alta disponibilidad
- ✓ CI/CD friendly
- ✓ Modular y personalizable
- ✓ Lenguaje declarativo (YAML)



## Recursos necesarios

- ✓ Repositorio GitHub
- ✓ Kubectl
- ✓ Plataforma k8s local: Docker desktop, minikube.





coremain

# Apartado teórico



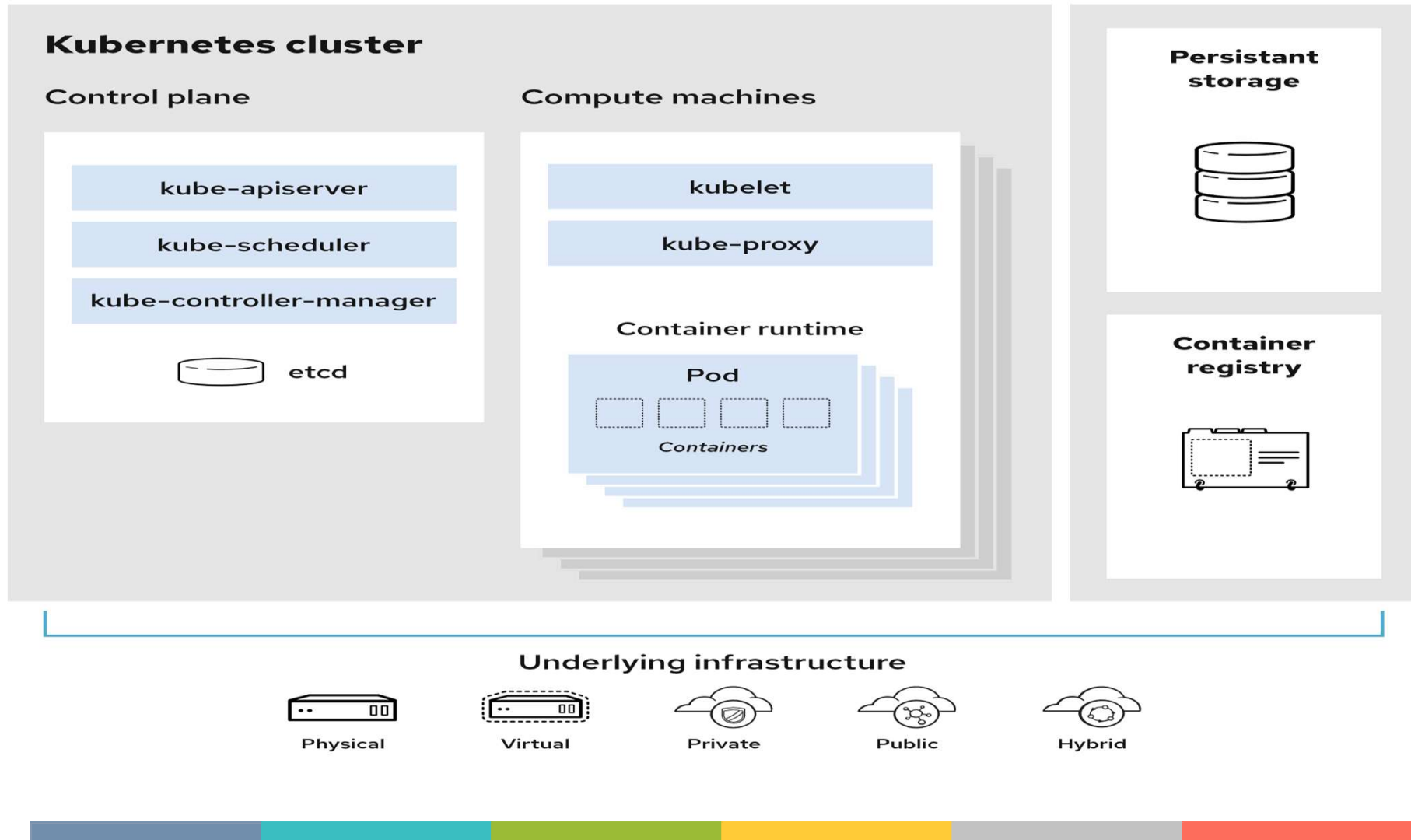


# Infraestructura





## Diagrama Kubernetes



## Nodo Master

Contiene los servicios que controlan el clúster:

- ✓ API Server: Permite a los usuarios interactuar con el clúster mediante herramientas como kubectl o kubeadm o peticiones REST. Herramienta declarativa, permite configurar el estado deseado.
- ✓ Scheduler: Controla todos los nodos y sus recursos. A la hora de ejecutar una carga de trabajo, Scheduler decide en qué nodo se ejecuta.

Las cargas de trabajo pueden tener afinidad a nodos y políticas.



Nodo Master

Contiene los servicios que controlan el clúster:

- ✓ Controller Manager: Gestor de controladores del clúster.
  - Controlador: procesos que monitorizan el estado del clúster y aplican los cambios necesarios para alcanzar el estado deseado.
- ✓ ETCD: Almacenamiento clave-valor de configuración



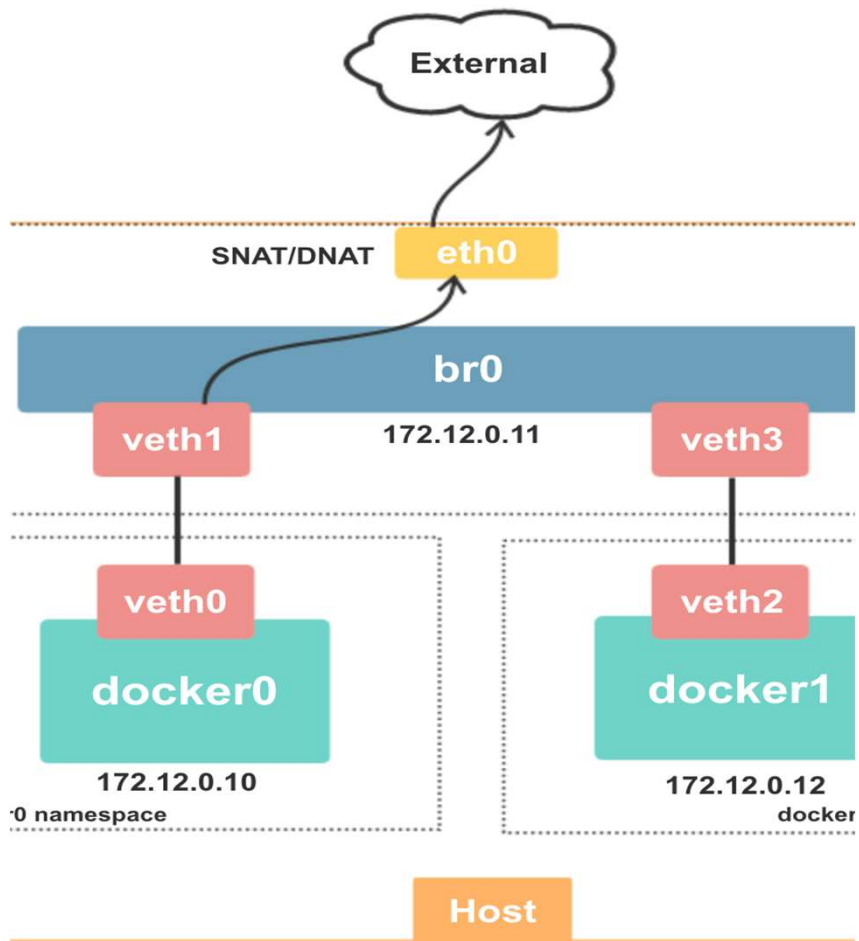
Nodo(s) Worker

Aquí se ejecutan nuestras cargas de trabajo

- ✓ Kubelet: Se registra como nodo contra Scheduler. Ejecuta las cargas de trabajo en Pods.
- ✓ Kube-proxy: Enruta el tráfico de red hacia los Pods.
- ✓ Container runtime: Software que se encarga de ejecutar los contenedores.



## Networking



Pod: Host lógico/virtual que puede tener uno o mas contenedores.

- ✓ Comunicación dentro del mismo pod
- ✓ Comunicación pod a pod mismo nodo
- ✓ Comunicación pod a pod distinto nodo

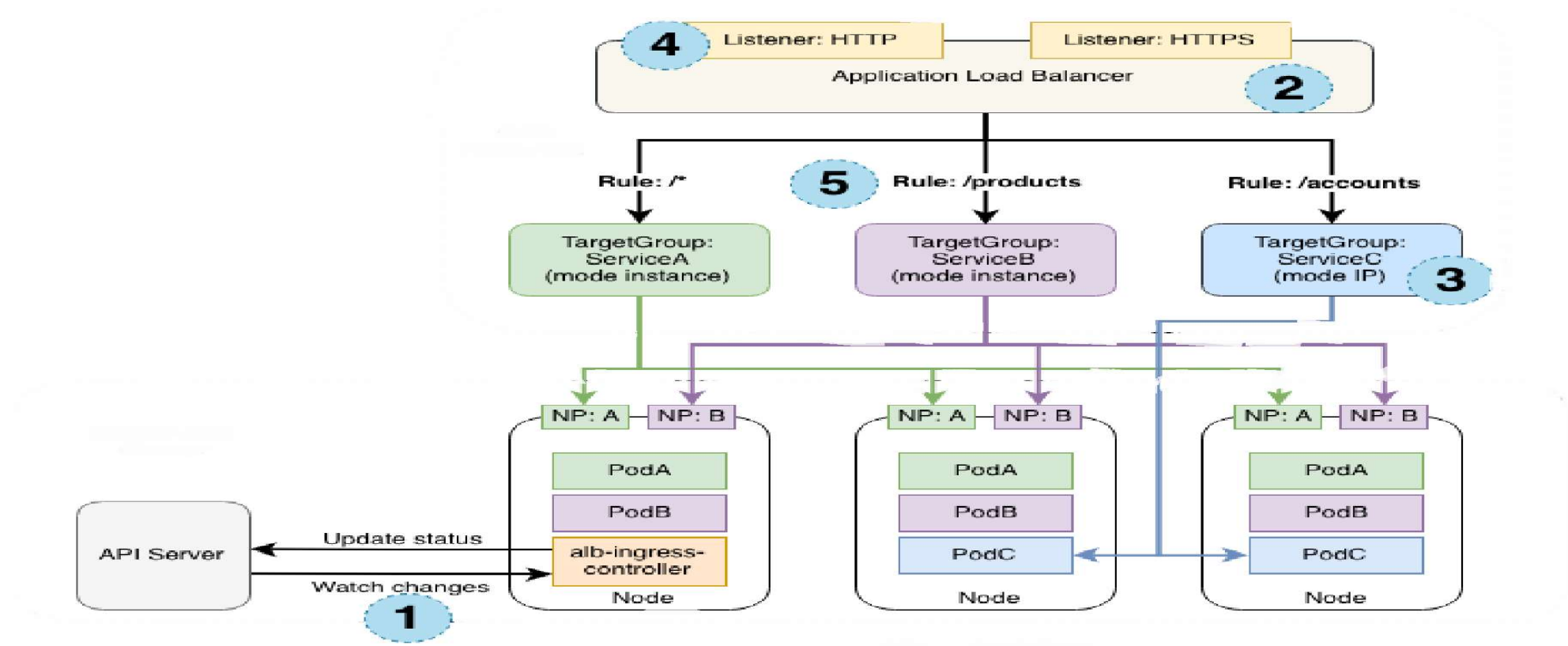
## Concepto Service

El Service es un recurso que adquiere una IP fija y enruta dinámicamente contra los pods.

- ✓ ClusterIP: Servicio orientado al tráfico dentro del clúster.
- ✓ NodePort (30000–32767): Servicio expuesto a Internet a través de un puerto de los nodos.
- ✓ LoadBalancer: Servicio expuesto a Internet a través de una IP pública.



## Ingress



## Demo

Demostración del funcionamiento del networking Kubernetes utilizando los diferentes tipos de Service:

- ✓ ClusterIP
- ✓ LoadBalancer
- ✓ Ingress.







# Recursos



## Recursos

- ✓ Namespace: Clúster virtual
- ✓ Pod: Uno o más contenedores agrupados en un host lógico.
- ✓ Deployment: Componente que actualiza y persiste uno o más pods.
- ✓ Service: Recurso que adquiere una IP fija y enruta dinámicamente para dar conectividad a los Pods.
- ✓ DaemonSet: Ejecuta un pod en cada nodo
- ✓ StatefulSet: Destinado a aplicaciones stateful (con estado fijo)



## Recursos II

- ✓ Subcomponente. Maneja réplicas

```
[kubeadm@server1 manifest]$ kubectl get deployments.apps
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
deployment-nginx    2/2     2             2           9m32s
[kubeadm@server1 manifest]$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/deployment-nginx-84f7d65dcf-294q8  1/1     Running   0          9m41s
pod/deployment-nginx-84f7d65dcf-dqp7h  1/1     Running   0          9m41s
pod/test                             1/1     Running   2          20h

NAME                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP     10.96.0.1    <none>        443/TCP    42h

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/deployment-nginx    2/2     2             2           9m41s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/deployment-nginx-84f7d65dcf  2         2         2       9m41s
[kubeadm@server1 manifest]$
```

### Recursos III

- ✓ Job: Ejecuta n veces un Pod hasta que termina.
- ✓ CronJob: Ejecución de Jobs en una fecha o intervalo temporal específico.
- ✓ ConfigMap: Ejecución de Jobs en una fecha o intervalo temporal específico.
- ✓ Secret: Almacena datos sensibles (tokens, claves...) que se pueden emplear como variables de entorno o ficheros que pueden montarse como volúmenes.
- ✓ HPA: Horizontal Pod Autoscaler. Aumenta o disminuye el número de replicas de un Pod.
- ✓ Service: Recurso que adquiere una IP fija y enruta dinámicamente contra los Pods.
- ✓ Network Policy: Aplica restricciones a la comunicación de red entre Pods.
- ✓ Persistent Volume: Almacenamiento del clúster.
- ✓ Persistent Volume Claim: Reserva de espacio de disco para asignar a un Pod.
- ✓ Storage Class: Tipo de almacenamiento que podemos usar.



Test

<https://docs.google.com/forms/d/1ru0NrUdruoQUlli9FySNmfN59OeefsKQ0cv45B5dc84/edit>





coremain

Apartado práctico







# Comandos básicos



## Contextos

Para conectar a los clústeres Kubernetes, existe un fichero llamado kubeconfig. En el se configura la conexión y autenticación contra cada uno de ellos. Cada conexión a un clúster se le denomina contexto.

Ubicación del kubeconfig: C:\Users\Xacobe\.kube\config

- ✓ Kubectl config get-contexts: Listar los contextos disponibles.
- ✓ Kubectl config use-context <contexto>: Utilizar un contexto determinado.
- ✓ Kubectl config current-context: Mostrar el contexto actual.





## Básicos

- ✓ `Kubectl get <tipo_recurso>`: Listar todos los recursos de un determinado tipo.
- ✓ `Kubectl describe <tipo_recurso><nombre_recurso>`: Mostrar detalles de un determinado recurso.
- ✓ `Kubectl apply -f <archivo_yaml>`: Aplicar un manifiesto YAML.
- ✓ `Kubectl delete <tipo_recurso><recurso>` o `<archivo_yaml>`: Borrar un determinado recurso, o todos los recursos definidos en un YAML.
- ✓ Modificador `-n`: Permite seleccionar un namespace.
- ✓ `Kubectl edit <tipo_recurso><recurso>`: Editar un recurso.
- ✓ `Kubectl logs <pod>`: Mostrar los logs del contenedor que corre en un Pod.





# Ejemplo práctico



## Servicios

Wordpress



Mysql



phpMyAdmin



Ingress

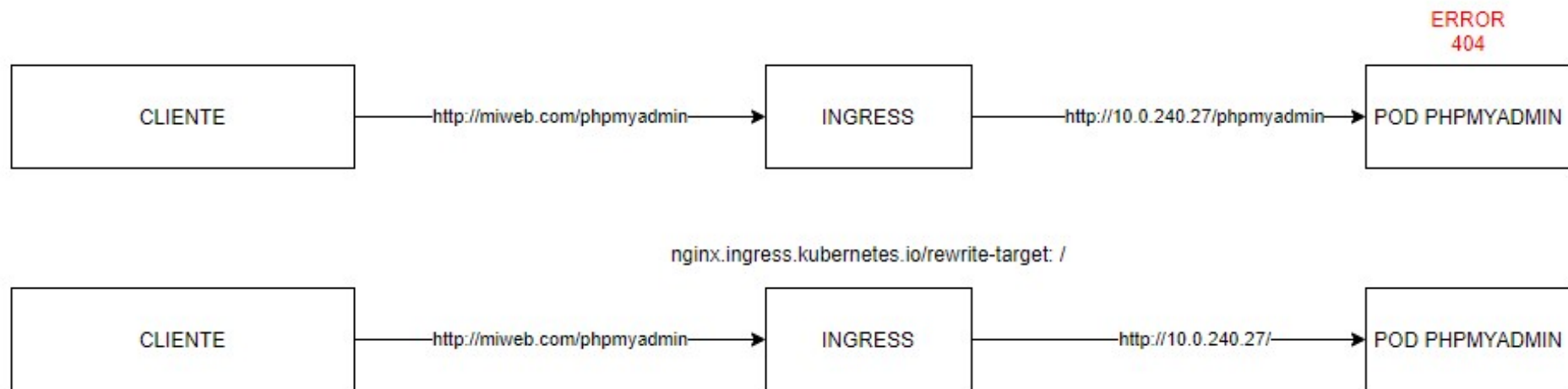


CertManager



## Ingress

Error al acceder a phpmyadmin debido a la configuración de Ingress NGINX





# Prácticas propuestas





# Gracias por asistir!



+34 902 90 55 90



coremain@coremain.com



[www.coremain.com](http://www.coremain.com)



[www.facebook.com/CoremainTIC/](https://www.facebook.com/CoremainTIC/)



[www.linkedin.com/company/coremain](https://www.linkedin.com/company/coremain)



[www.twitter.com/CoremainTIC](https://www.twitter.com/CoremainTIC)