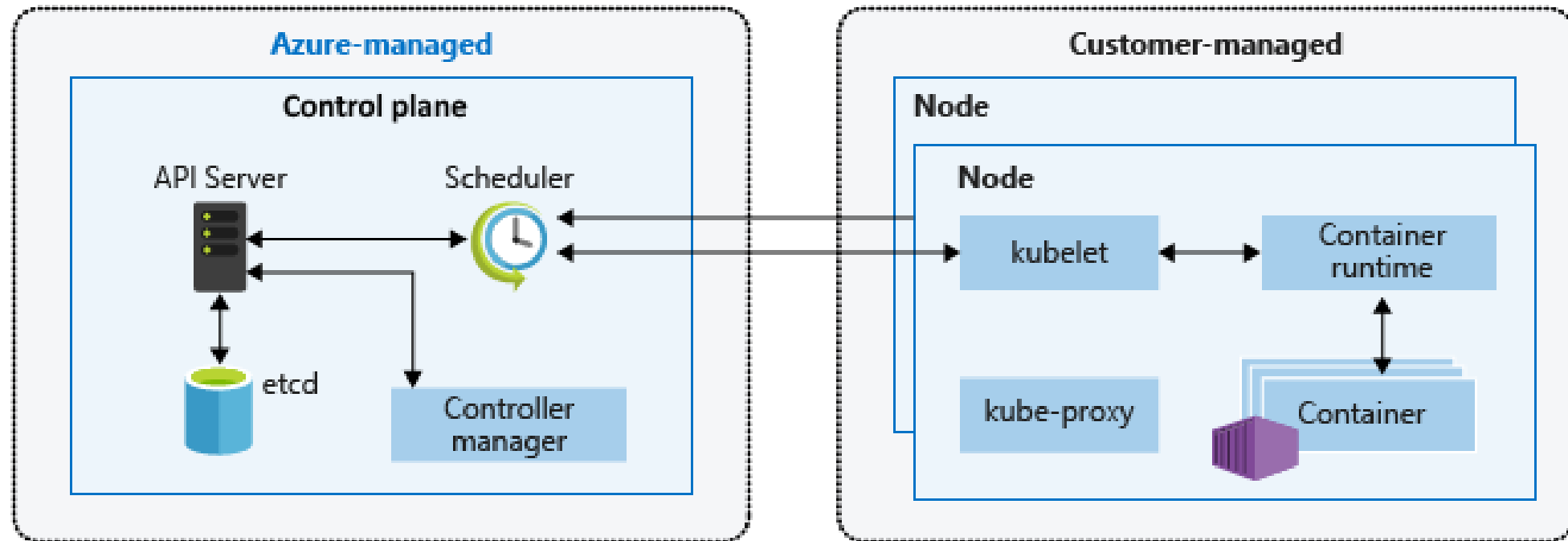
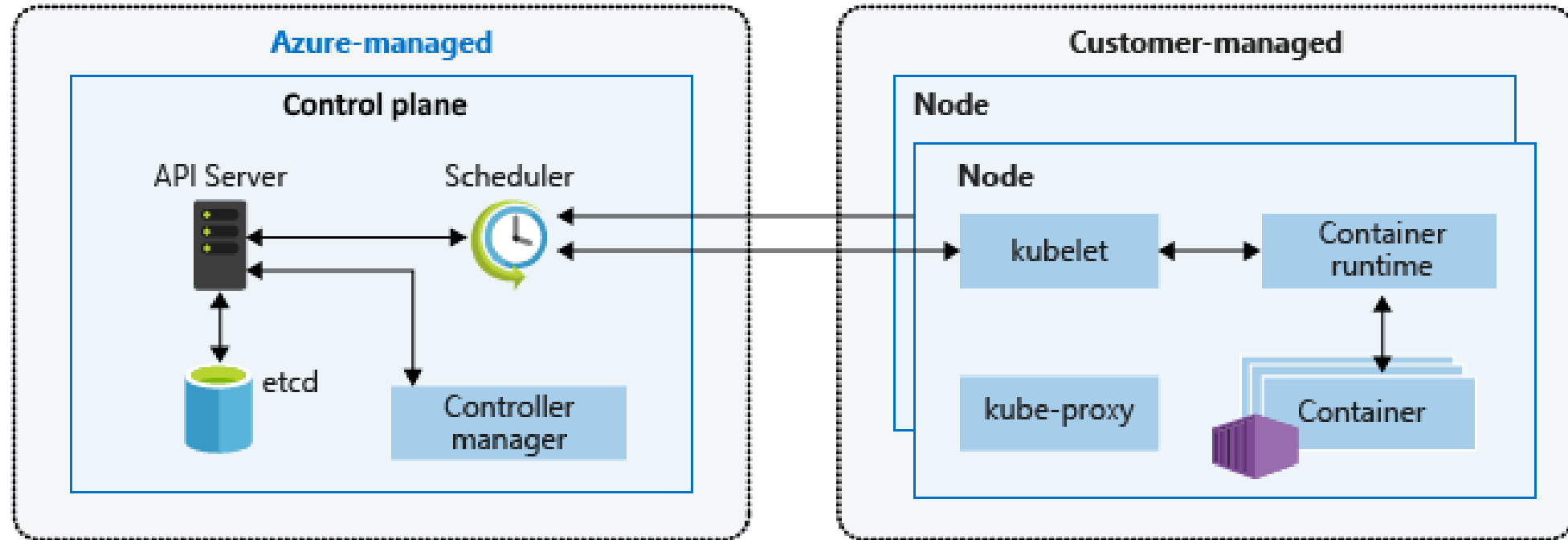


# Kubernetes



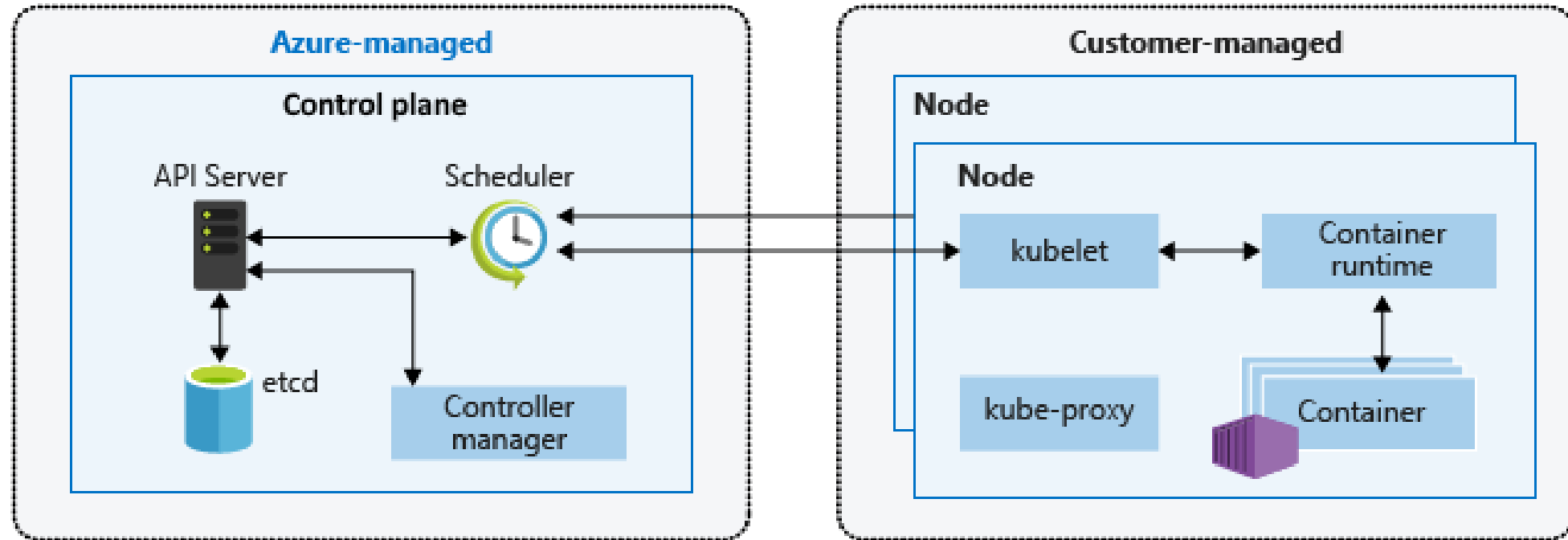
# kubernetes

- Plataforma para el manejo de aplicaciones basadas en contenedores y las reglas de almacenamiento y redes asociadas a dichas aplicaciones
- Microservicios
- Orquestación
- CI/CD



# Control plane

- Provee los servicios core de kubernetes y orquestación de las cargas de trabajo de las aplicaciones
  - Api Server. Plataforma para interacción con el API, por ejemplo kubectl
  - Etcd. Mantiene el estado de kubernetes y sus configuraciones
  - Kube-scheduler. Determina que nodos pueden correr las cargas de trabajo al crear o escalar servicios
  - Kube-controller-manager. Supervisa la operación de controladoras encargadas para la replicación de nodos y el manejo de operaciones con los mismo



# Nodos y pool de nodos

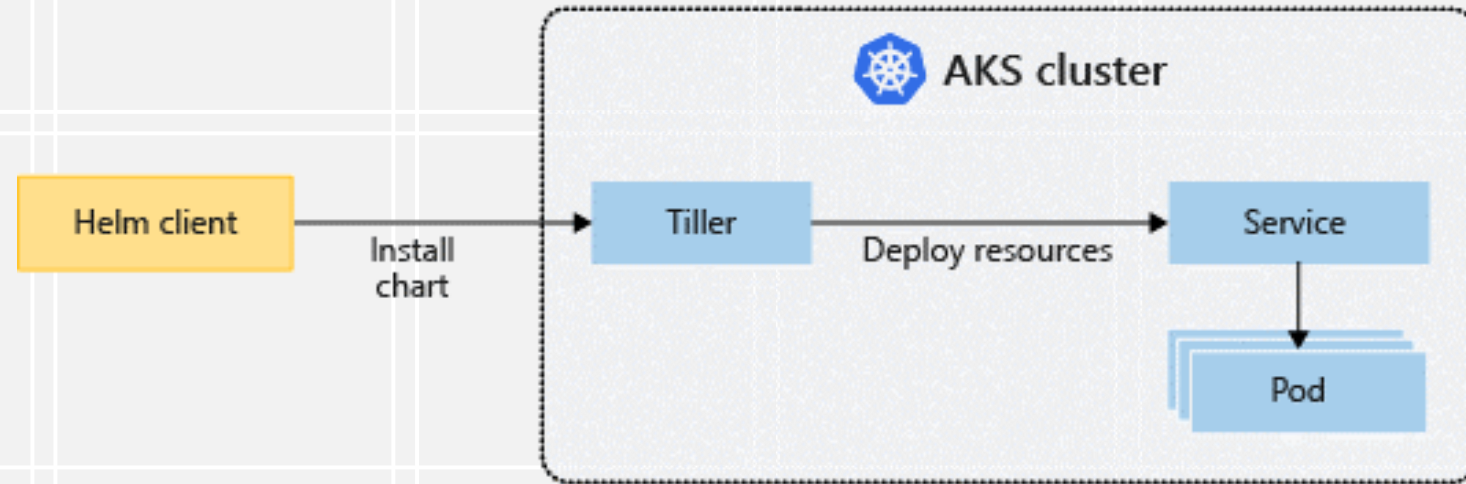
- Las aplicaciones y servicios que soportan dichas aplicaciones corren dentro de nodos, puede ser una maquina física o virtual que ejecuta el runtime de los contenedores y componentes necesarios.
  - Kubelet. Agente de kubernetes que procesa las solicitudes de orquestación
  - Kube-proxy. Se encarga de la administración de la red virtual para el nodo, rutea el trafico de red y administra las direcciones IP para los servicios y los pods
  - Container runtime. Permite la ejecución de aplicaciones basadas en contenedores

---

# Reserva de recursos

- CPU
- Memoria
  - Un kubelet Daemon se encuentra instalado en cada agente de kubernetes para orquestar la creación y terminación de un contenedor. Requiere de por lo menos 750 Mi de memoria

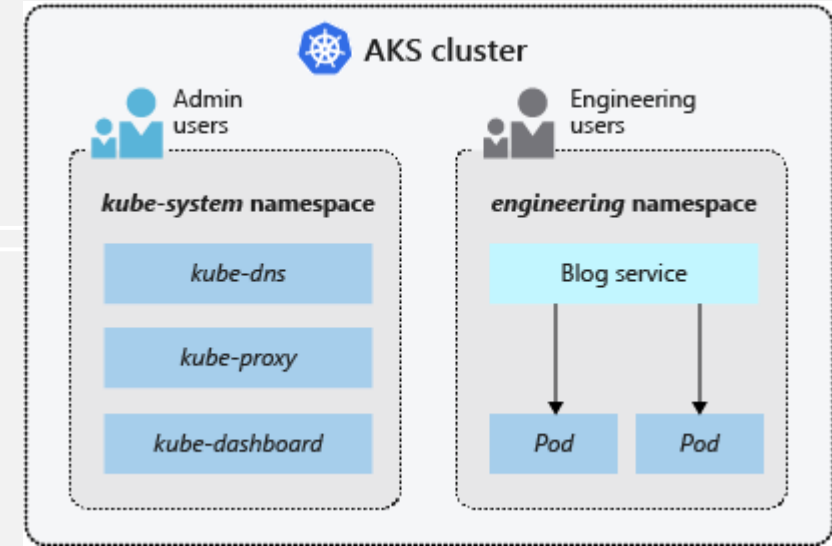
# Helm



- Package management
- Tiller
- Chart

# Namespaces

- default
- Kube-system
- Kube-public



- Los recursos como pods y deployments se agrupan en namespaces lógicamente, nos proveen una forma lógica de dividir un cluster de aks

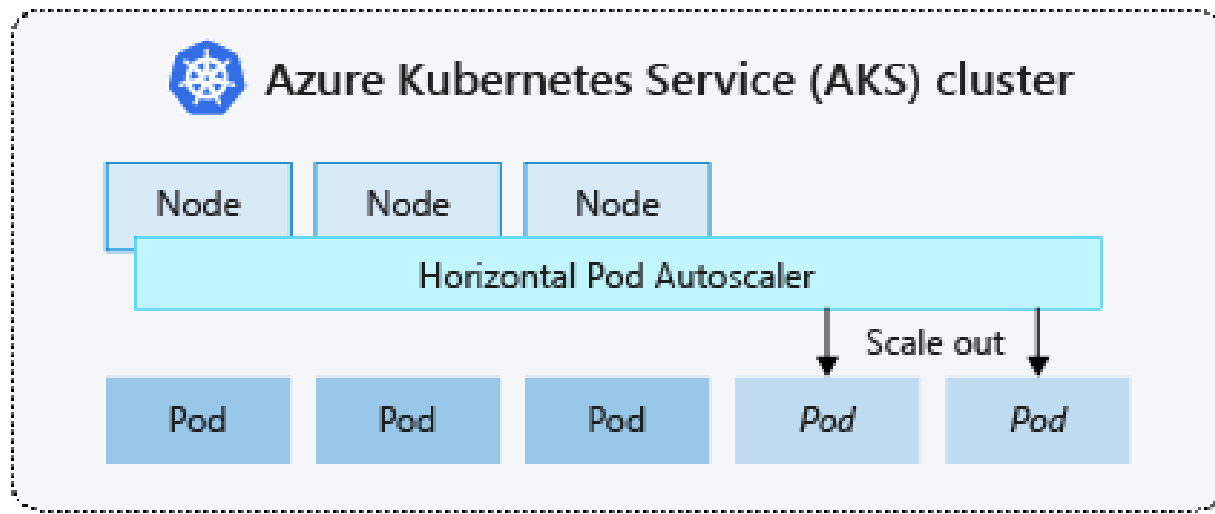
# Storage

- Volumes
- Secrets
- ConfigMaps





Scaling

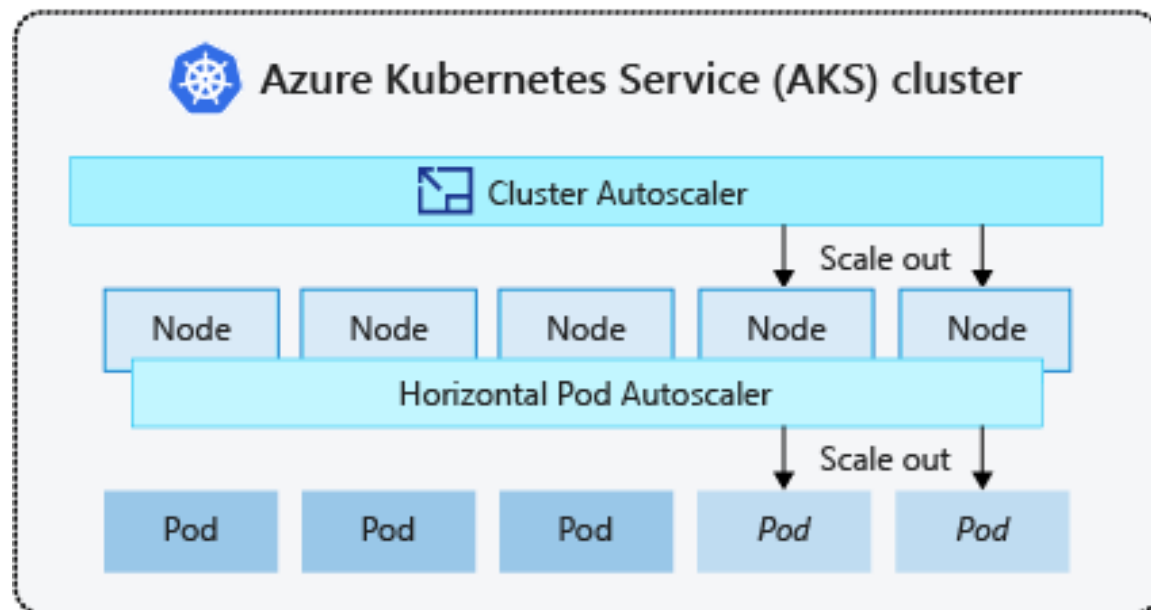


```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: azure-vote-back-hpa
spec:
  maxReplicas: 10 # define max replica count
  minReplicas: 3 # define min replica count
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: azure-vote-back
  targetCPUUtilizationPercentage: 50 # target CPU utilization

apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: azure-vote-front-hpa
spec:
  maxReplicas: 10 # define max replica count
  minReplicas: 3 # define min replica count
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: azure-vote-front
  targetCPUUtilizationPercentage: 50 # target CPU utilization
```

# Horizontal pod autoscaling

- Revisa las métricas de los pods cada 30 segundos



# Cluster autoscaler

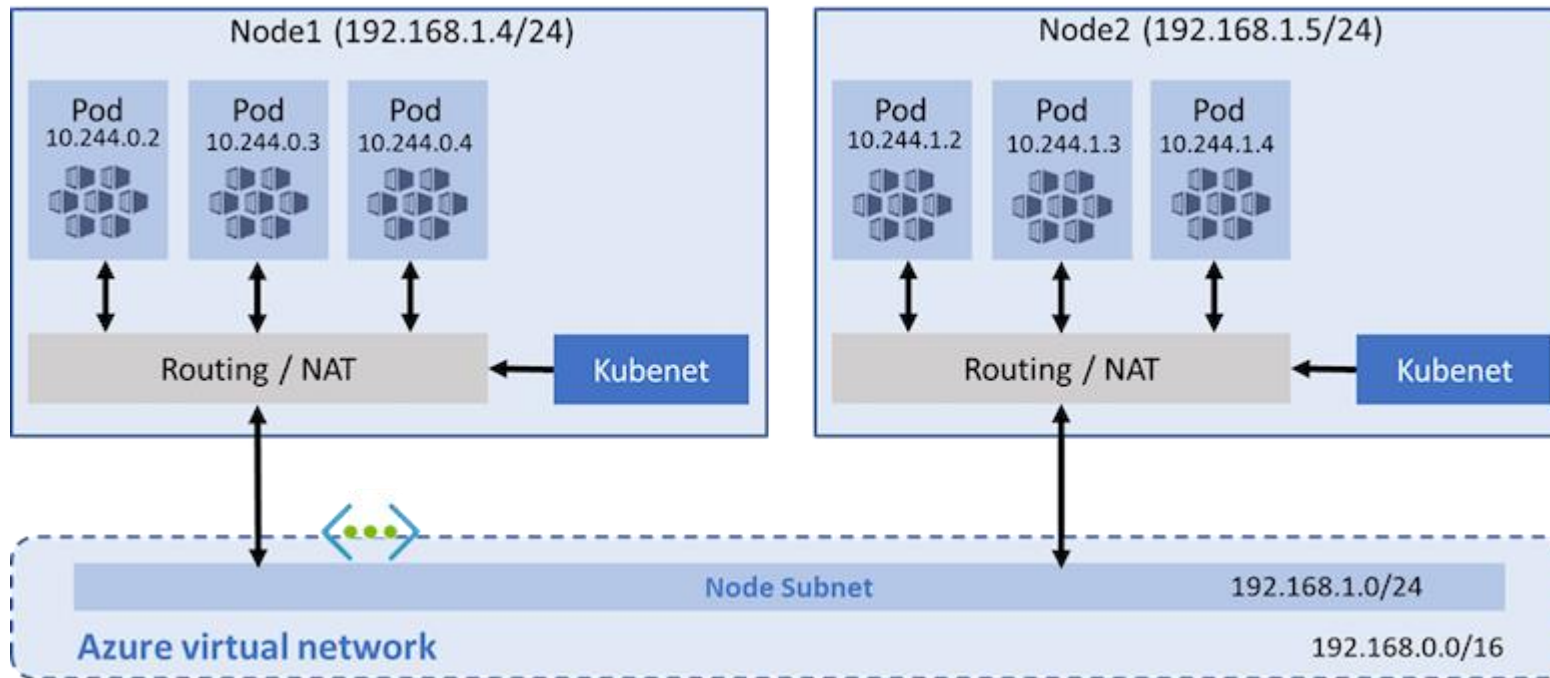
- Metrics API revisa cada 10 segundos



# Redes

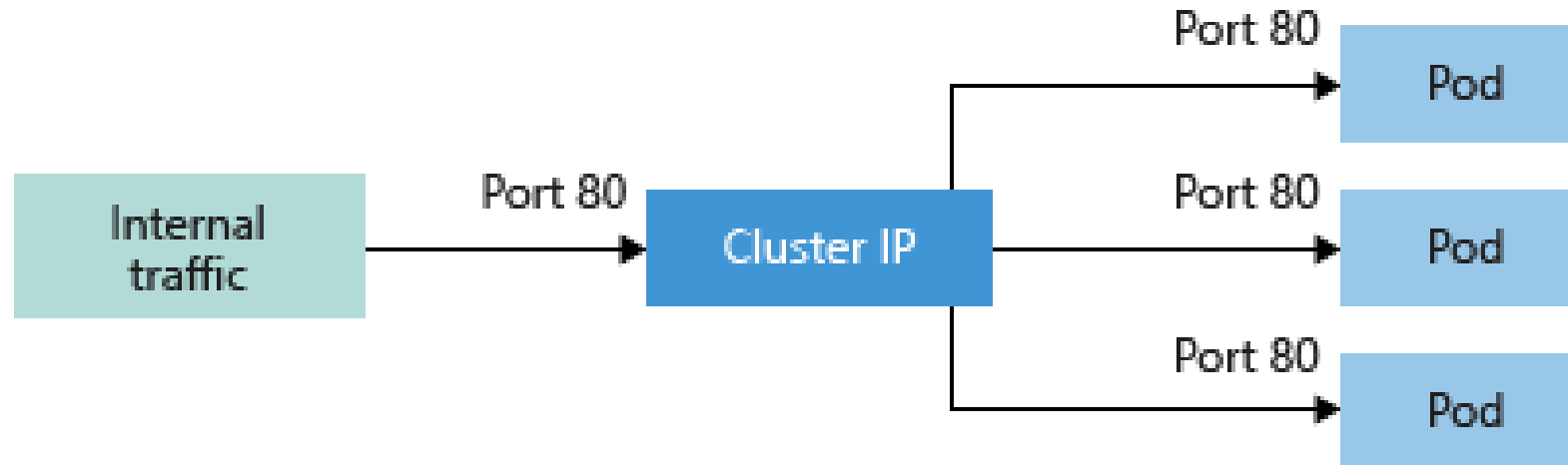
Servicios

Ingress



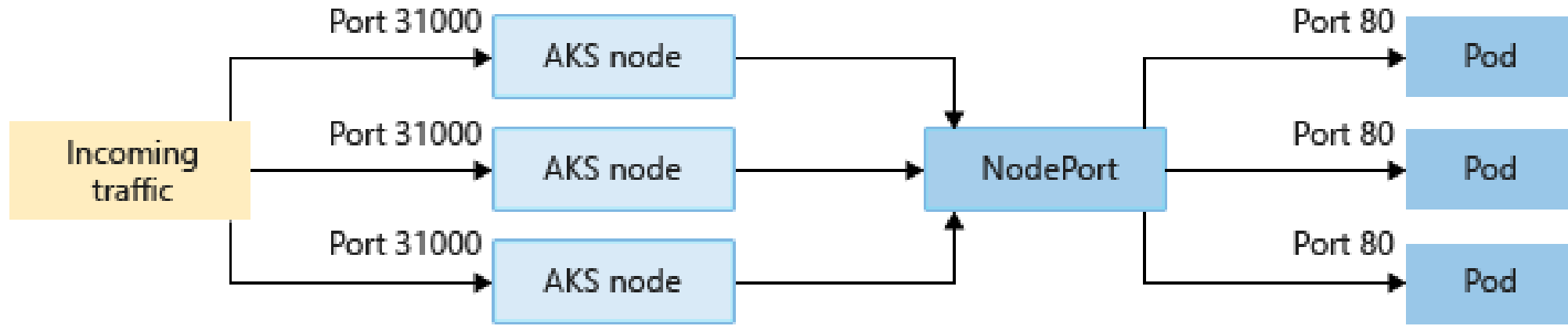
# kubernetes

- Networking. Cad ánode recibe una IP en la subred virtual.



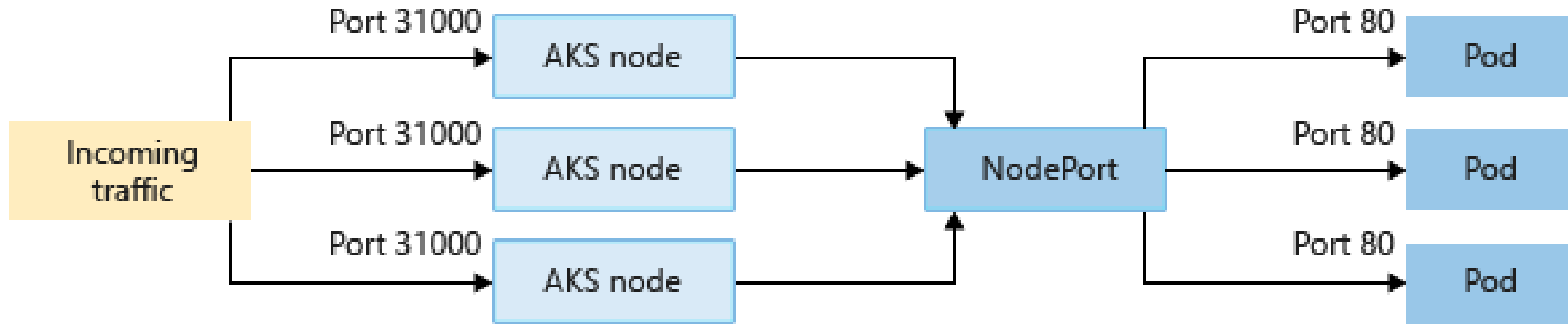
## Servicios

- Cluster IP. Crea una IP interna para su uso dentro del cluster de kubernetes. Bueno para cargas de trabajo que no deseamos exponer externamente



# Servicios

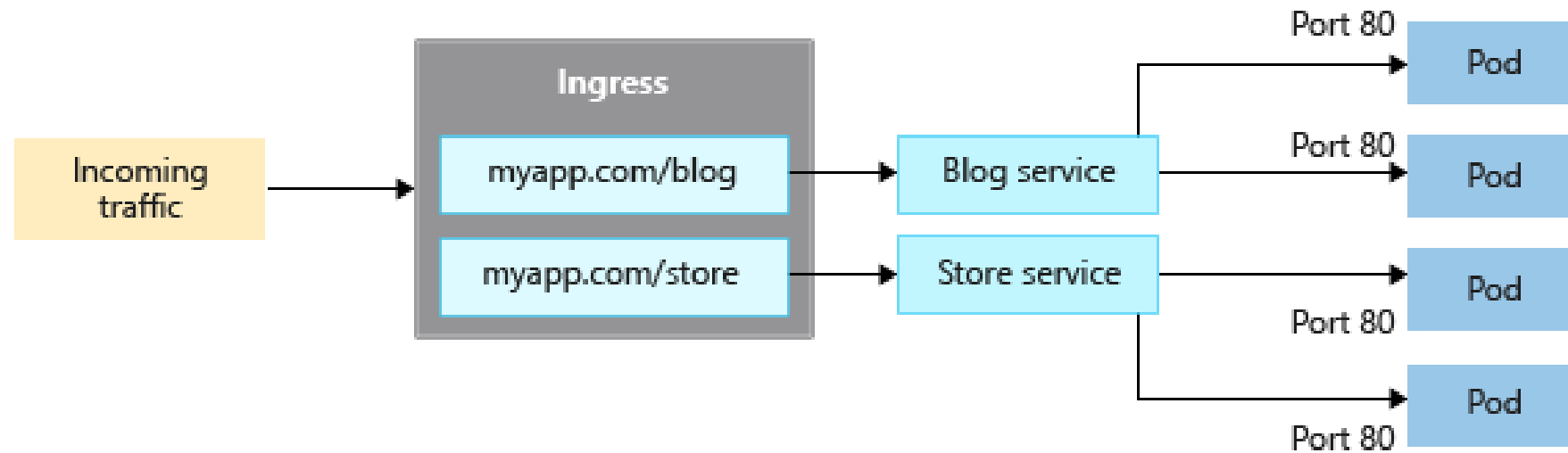
- Node Port. Crea el mapeo de un puerto que permite acceder al servicio a través de la IP del nodo y el puerto



# Servicios

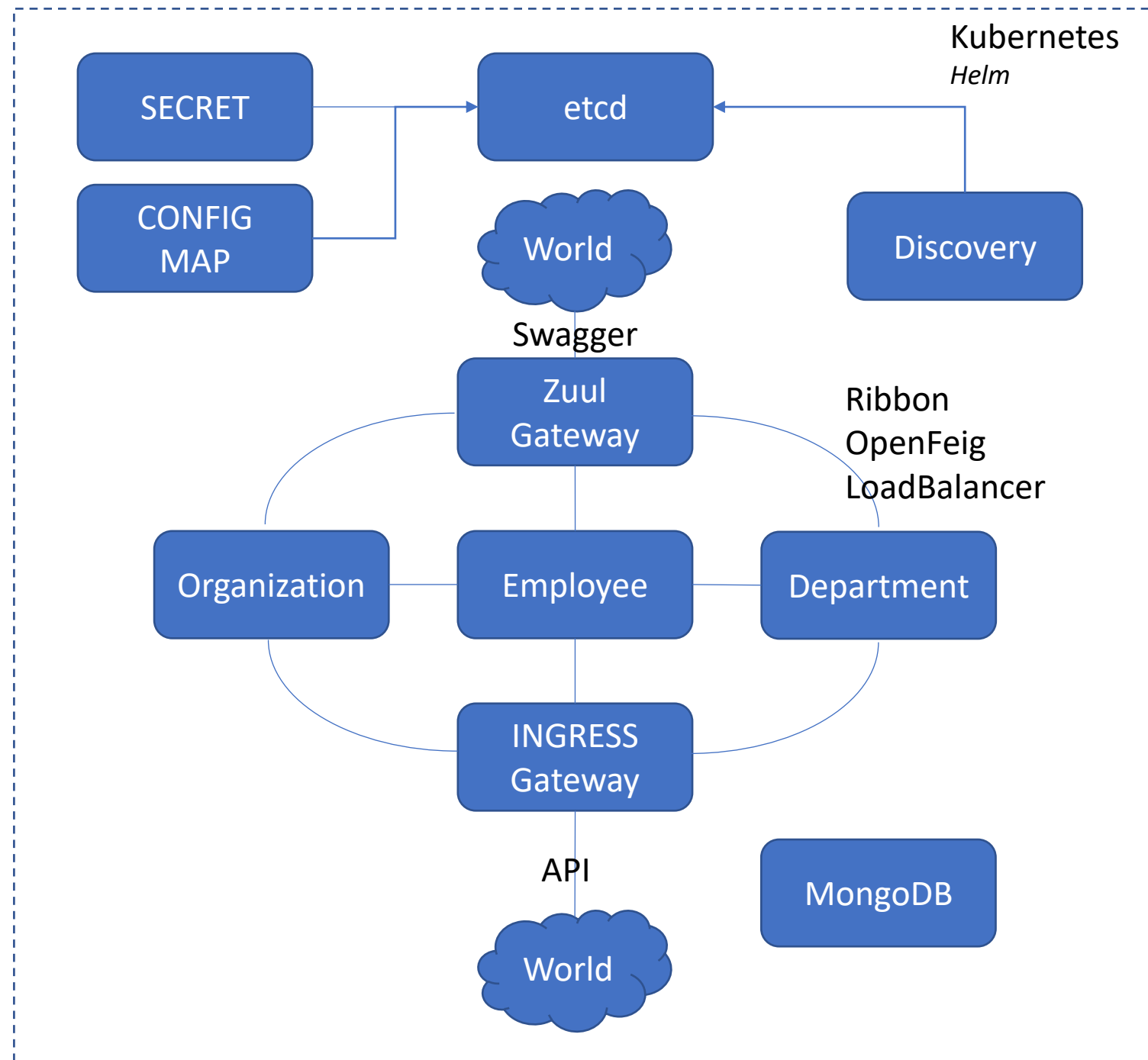
- Load Balancer. Crea un load balancer con acceso externo, al que le configura una IP externa para el acceso de los servicios en el pod






# Ingress controller

- API Gateway
- SSL/TLS Termination



- 
- `kubectl apply -f https://git.io/mi.yaml`
  - `kubectl get service -o wide`
  - `kubectl get pods -o wide`
  - `kubectl get deployment my-dep`
  - `kubectl describe pods my-pod`
  - `kubectl logs --tail 200 my-pod`
  - `kubectl port-forward my-pod 5000:6000`
  - `kubectl get pod department-7b468c7685-m9hbx -o yaml`
- 