

# Sesion 4 – Workloads en Kubernetes

MitoCode Network

Por: Juan Carlos Salvador García

# AGENDA

1 Objetos en Kubernetes: Daemonsets

2 Static Pods

3 Rolling Updates y Rollbacks

4 Multicontainers

5 Comandos y Variables de Entorno

6 Configmaps

7 Secrets

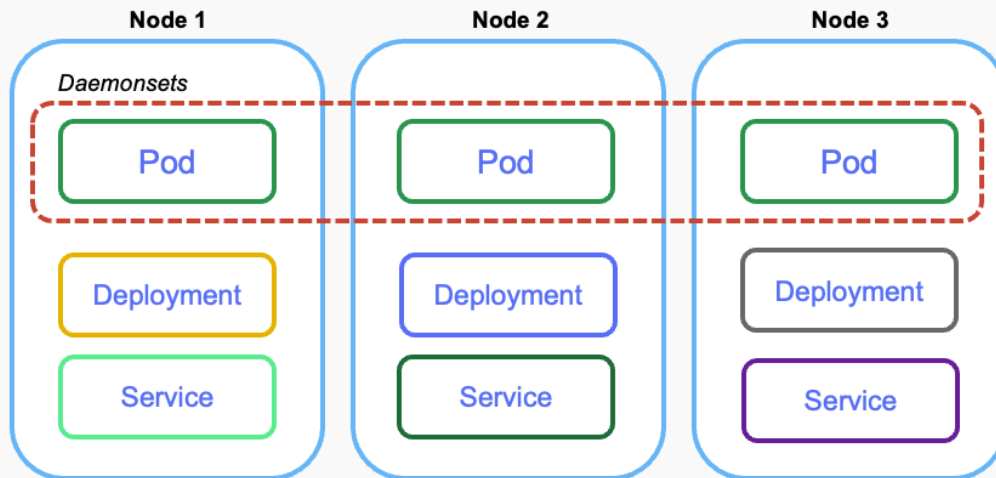


# Daemonsets

Se encarga de asegurar que un pod sea desplegado en cada nodo que se encuentre disponible dentro del cluster.

## Casos de Uso

- ✓ Monitoreo de nodos.
- ✓ Agentes para logs.



# Despliegue de Daemonsets

## Despliegue Imperativo

kubectl apply -f daemonsets.yaml

kubectl get daemonsets

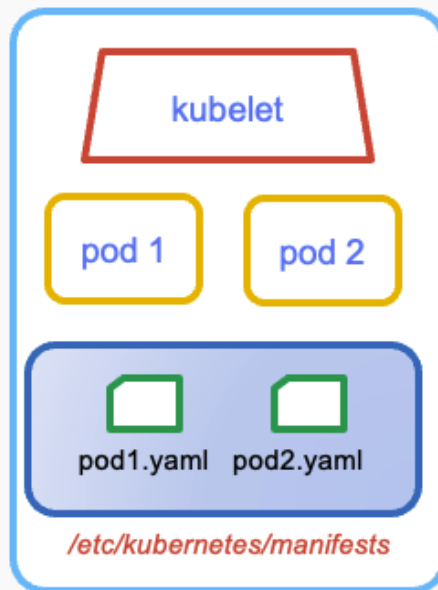
kubectl describe daemonsets dms

```
daemonsets.yaml
1  apiVersion: apps/v1
2  kind: DaemonSet
3  metadata:
4    name: fluentd-elasticsearch
5    namespace: kube-system
6  spec:
7    selector:
8      matchLabels:
9        name: fluentd-elasticsearch
10   template:
11     metadata:
12       labels:
13         name: fluentd-elasticsearch
14     spec:
15       containers:
16       - name: fluentd-elasticsearch
17         image: quay.io/fluentd_elasticsearch/fluentd:v2.5.2
18
```

# Static Pods

Es una característica que te permite crear Pods sin necesidad de usar el kube api-server y solo delegando la responsabilidad al kubelet del nodo asignado.

## Node 01

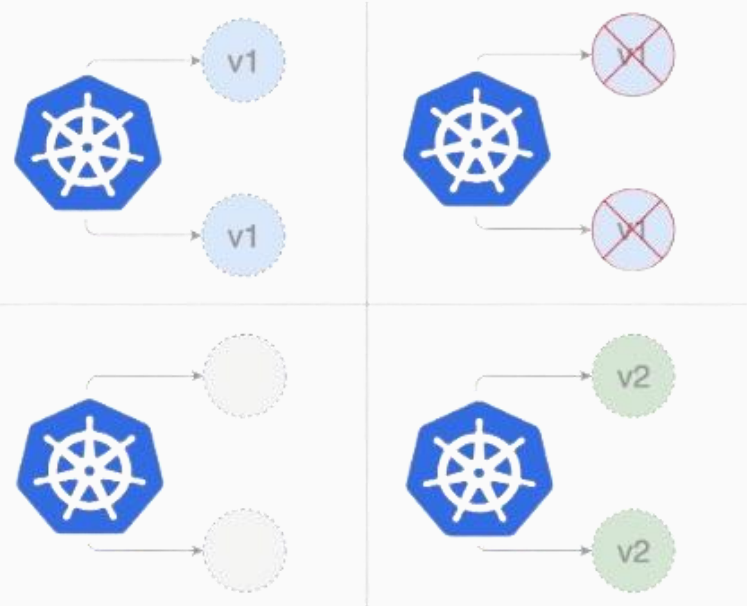


# Rolling Update y Rollbacks

## Rolling Update



Permite realizar una actualización progresiva de los pods.



## Rollbacks



Permiten volver a una versión anterior de la aplicación.

# Comandos Rolling Update y Rollbacks

```
kubectl set image deployment/my-deploy nginx=nginx:1.2
```

```
kubectl rollout status deployment/my-deploy
```

```
kubectl rollout history deployment/my-deploy
```

```
kubectl rollout undo deployment/my-deploy
```

# Multicontainer Pods

## Patrón Sidecar



Es un Pod donde se alojan 2 contenedores, el cual uno se encarga de la aplicación y el otro proporciona funcionalidades auxiliares.



El segundo contenedor opera de manera asincrónica y no suele estar expuesto públicamente.



Ambos contenedores comparten el mismo ciclo de vida.

## Casos de Uso



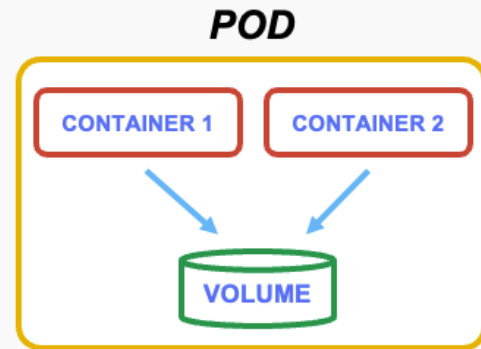
Recolección de logs.



Servicio de Proxy.



Cache de datos.





# Multicontainer Pods

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: myapp-deployment
5 spec:
6   replicas: 3
7   selector:
8     matchLabels:
9       app: myapp
10  template:
11    metadata:
12      labels:
13        app: myapp
14    spec:
15      containers:
16
17      - name: myapp-container
18        image: myapp:latest
19        ports:
20          - containerPort: 8080
21            name: myapp-port
22
23      - name: logging-sidecar
24        image: logging-sidecar:latest
25        volumeMounts:
26          - name: logs-volume
27            mountPath: /var/log/myapp
28
29    volumes:
30      - name: logs-volume
31        emptyDir: {}
```

→ *Aplicación principal*

→ *Lectura de logs*