Sesión 3 - Scheduling en Kubernetes

MitoCode Network

Por: Juan Carlos Salvador García

MitoCode

AGENDA

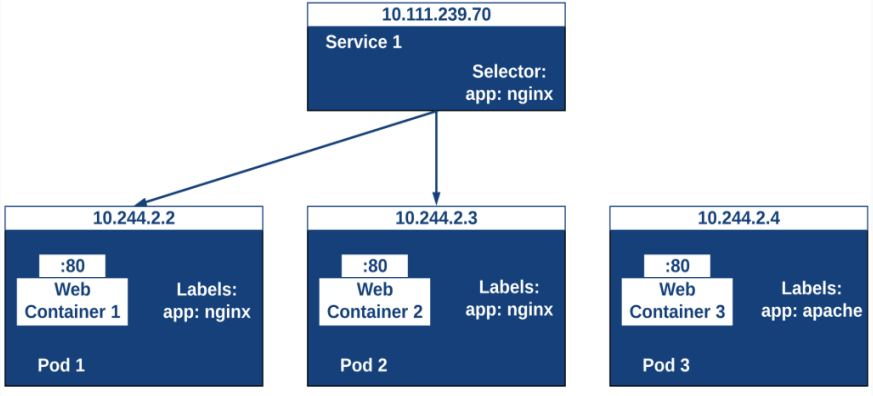
1. Labels y Selectors
2. Taints y Tolerations
3. Node Selectors
4. Requerimiento de Recursos


¿Cómo se asignan los nodos a los Pods?
NODENAME
MC
MitoCode

LABELS Y SELECTORS

¿Qué son?

Son valores de tipo key/value que te permiten agrupar un conjunto de recursos de kubernetes por medio de una o varias etiquetas.



kubectl get pods -show-labels

kubectl get pods -l app=code

MitoCode

LABELS Y SELECTORS

Despliegue Declarativo:

kubectl apply -f deployment.yaml

Despliegue Imperativo:

kubectl label pods nginx owner=mitocode

deploymentyaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
    env: certification
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
      env: certification
  template:
    metadata:
      labels:
        app: nginx
        env: certification
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

MitoCode

TAINTS Y TOLERATIONS

¿Qué son?

Estas funcionalidades permiten asegurar que no se ejecuten Pods en ciertos nodos. Los taints son aplicables hacía los nodos y los tolerations hacía los pods.

Taints

kubectl taint nodes node-name key=value:taint-effect

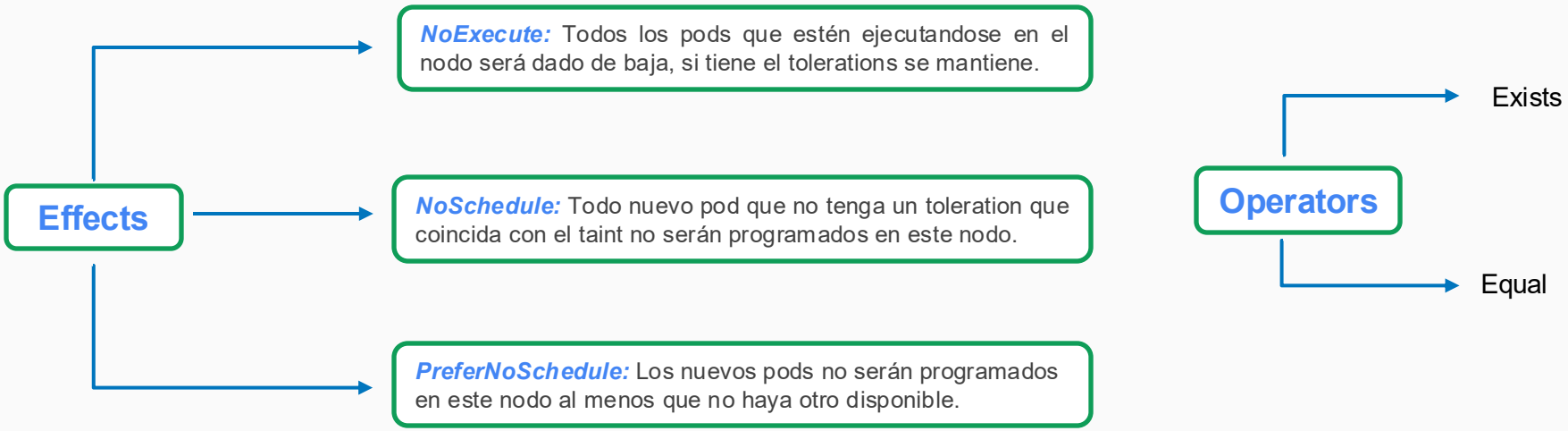
kubectl taint nodes node-name key=value:taint-effect-

Tolerations

```yaml
tolerations:
- key: "example-key"
  operator: "Exists"
  effect: "NoSchedule"
```

MitoCode

TAINTS Y TOLERATIONS



MitoCode

Node Selectors

Node Affinity

nodeAffinity.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: with-node-affinity
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: topology.kubernetes.io/zone
            operator: In
            values:
            - antarctica-east1
            - antarctica-west1
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: another-node-label-key
            operator: In
            values:
            - another-node-label-value
  containers:
  - name: with-node-affinity
    image: registry.k8s.io/pause:2.0
```

Tipos de Node Affinity

requiredDuringSchedulingIgnoredDuringExecution

preferredDuringSchedulingIgnoredDuringExecution

Operators

In, NotIn, Exists, DoesNotExist

MitoCode

Requerimientos de Recursos

Para iniciar el pod requerira 64mi de memoria y 250m de cpu del nodo al que sea asignado.

requestLimit.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image: images.my-company.example/app:v4
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

El pod será reiniciado si es que supera alguno de los valores definidos.