The Concepts of Kubernetes

- *Configuration*

Speaker: hp.huang

Outline

- Best Practices
- Managing Compute Resources for Containers
- Assigning Pods to Nodes
- Evicting Pods from Nodes
- Managing Sensitive Information
- Organizing Clusters Access
- Pod Priority and Preemption

Best Practices - General Configuration Tips

- Specify latest stable API version
- Store the configuration with version control (GitOps)
- Write the configuration file using ***YAML*** rather than JSON
- DON'T specify default value
- Put object descriptions in ***annotations***
- Use ***kubectl***
  - execute command at ***directory*** layer
  - Use ***label selectors*** for ***kubectl get, Kubectl delete***
  - *Use **kubectl run, kubectl expose***

Best Practices - Others

- DON’T use ***Naked Pods***
  - they will not be rescheduled in the event of a node failure.
- Create a Service before the workload is going to access it
- Define and use ***labels*** that identify semantic attrs. of application or Deployment
  - *{ app: myapp, tier: frontend, phase: test, deployment: v3 }*

Managing Compute Resources for Containers

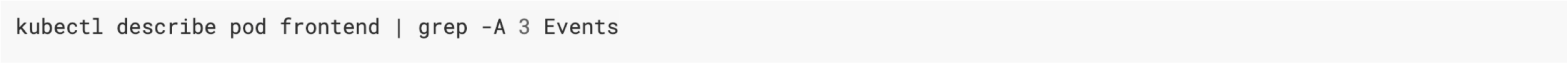
- Resource Types
  - CPU, Memory

- Resource ***requests*** *of Node*
- Resource ***limits*** *of each container*

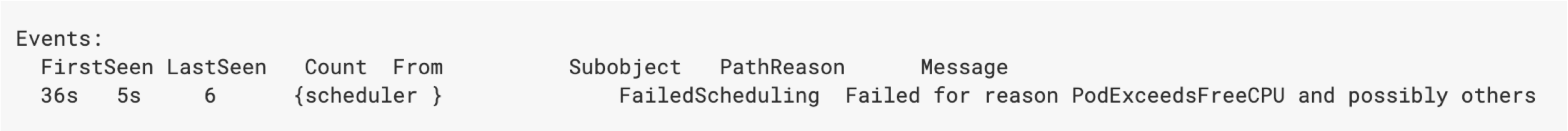
Managing Compute Resources for Containers

Troubleshooting

- Give a scenario
  - The Pods are pending with event message ***failedScheduling***



```
kubectl describe pod frontend | grep -A 3 Events
```



```
Events:
  FirstSeen LastSeen   Count  From          Subobject   PathReason      Message
  36s   5s     6      {scheduler }              FailedScheduling  Failed for reason PodExceedsFreeCPU and possibly others
```

- Actions
  1. Add more nodes to the cluster
  2. Terminate unneeded Pods to make room for pending Pods
  3. Check that the Pod is not larger than all the nodes

Assigning Pods to Nodes – Node Selector

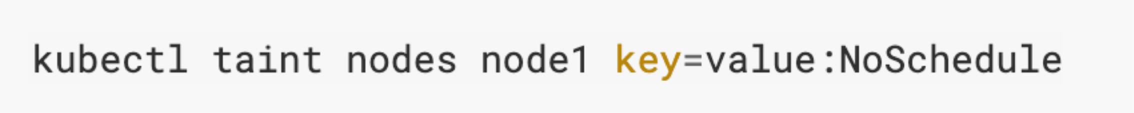
- To constrain/prefer pods to run on particular nodes
- The recommended approach – use ***label*** selectors
  - Attach ***label*** to the node
  - Add a ***nodeSelector*** filed to the pod configuration

Assigning Pods to Nodes – Use Case

- The service needs to run on specific IPs

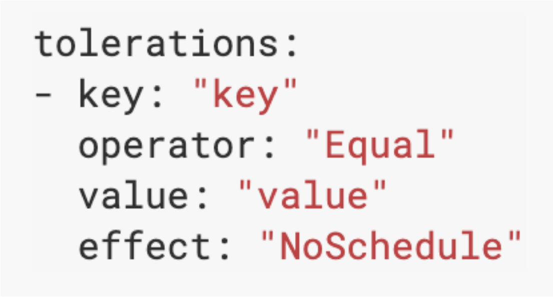
Evicting Pods from Nodes - Taints and Tolerations

- To ensure that pods are not scheduled onto inappropriate nodes
- **Taint**
  - *kubectl taint nodes <node_name> <key>=<value>:<Effect>*
  - for example:

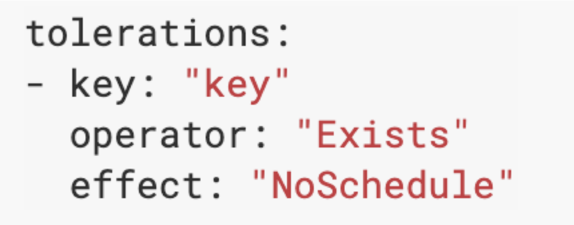


```
kubectl taint nodes node1 key=value:NoSchedule
```

- **Toleration**



```
tolerations:
- key: "key"
  operator: "Equal"
  value: "value"
  effect: "NoSchedule"
```



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

- **Effect**
  - _NoSchedule_
  - _RreferNoSchedule_
  - _NoExecute_*

Evicting Pods from Nodes – Use Case

- Dedicated Nodes*
  - *kubectl taint nodes nodename* ***dedicated=groupName****:NoSchedule*

- Nodes with Special Hardware
  - *kubectl taint nodes nodename* ***special=true****:NoSchedule*
  - *kubectl taint nodes nodename* ***special=true****:PreferNoSchedule*

*The pods with the tolerations will then be allowed to use the tainted (dedicated) nodes as well as any other nodes in the cluster

Managing Sensitive Information - Secrets

- Creating Secrets
  - Creating Secrets Using ***kubectl***
  - Creating Secrets Manually
  - Creating Secrets Using ***Generator***

- Using Secrets
  - Using Secrets as ***Files*** from a Pod
  - Using Secrets as ***Environment Variables***

Organizing Clusters Access - kubeconfig

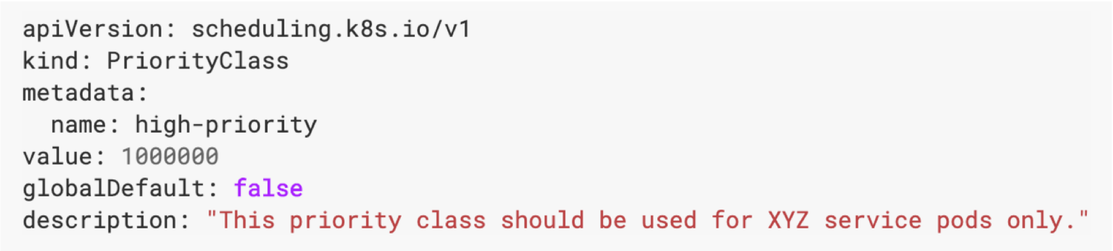
- Organize information about ***clusters***, ***users***, ***namespaces***, and ***authentication mechanisms***.

- The consist of Context
  - cluster
  - namespace
  - user

- Partition & Merging
  - *https://github.com/hp-huang-tw/kubernetes/tree/master/config-exercise*

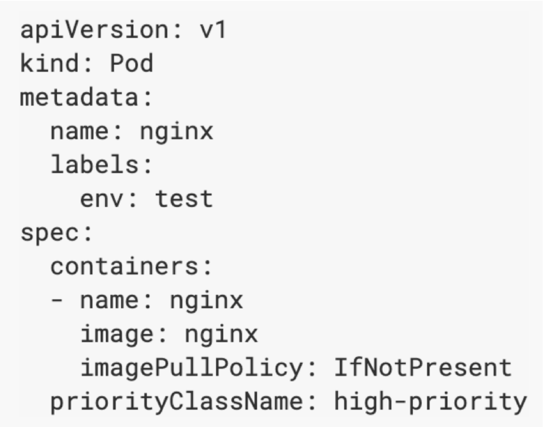
Pod Priority and Preemption

- **Priority**
  - The Pod can has priority. [Stable from kebernetes 1.14]
  - ***PriorityClass***



```yaml
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high-priority
value: 1000000
globalDefault: false
description: "This priority class should be used for XYZ service pods only."
```

  - ***PriorityClassName***



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  priorityClassName: high-priority
```

- **Preemption**
  - Give an example
    - There’s a pending Pod P which priority is 1000, there’re Pods Q_{1-n} which priority is 1 and is running on Nodes N_{1-m}.

Reference

- https://kubernetes.io/docs/concepts/configuration/