ackstorm

Kubernetes Training

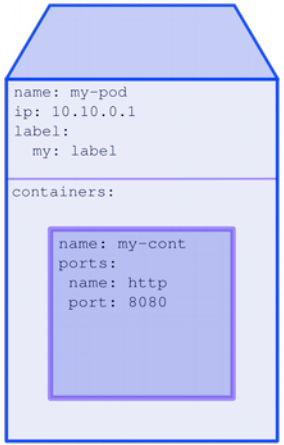
2019 / Barcelona

kubernetes

Liveness Probe

```
  containers:
  - name: my-container
    image: my-image
...
    livenessProbe:
      httpGet:
        path: /healthz
        port: 8080
        httpHeaders:
        - name: X-Custom-Header
          value: Ackstorm
      initialDelaySeconds: 5
      periodSeconds: 10
...
    livenessProbe:
      exec:
        command:
        - cat
        - /tmp/healthz
      initialDelaySeconds: 5
      periodSeconds: 5
...
    livenessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 5
      periodSeconds: 10
```

Cluster



- Kubelet performs liveness probes to ensure the container is healthy (alive).
- Unhealthy containers will force the pod, where they are running, to be killed and re-created.

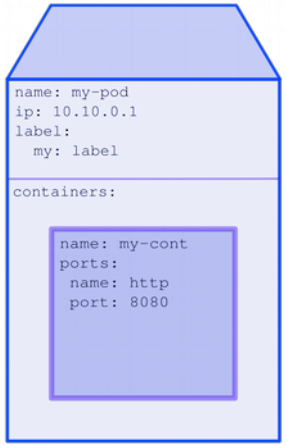
- There are three types of liveness probes:
  - Command (if returns 0)
  - TCP (if connection is established)
  - HTTP* (if returns code between 200 and 399)

* For HTTP connections, it is user's responsibility what returns the server.

Readiness Probe

```
  containers:
  - name: my-cont
    image: my-image
...
    readinessProbe:
      httpGet:
        path: /healthz
        port: 8080
        httpHeaders:
        - name: X-Custom-Header
          value: Ackstorm
      initialDelaySeconds: 5
      periodSeconds: 10
...
    readinessProbe:
      exec:
        command:
        - cat
        - /tmp/healthz
      initialDelaySeconds: 5
      periodSeconds: 5
...
    readinessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 5
      periodSeconds: 10
```

Cluster



- Kubelet performs readiness probes to ensure the container is ready to receive traffic.
- Not ready containers' IP address won't be added to service endpoints list.

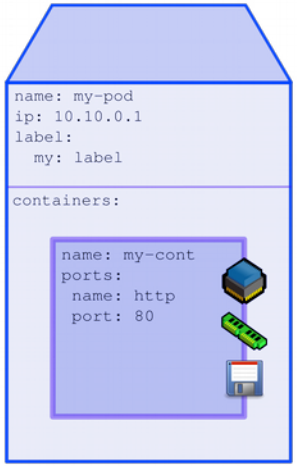
- There are three types of readiness probes:
  - Command (if returns 0)
  - TCP (if connection is established)
  - HTTP* (if returns code between 200 and 399)

* For HTTP connections, it is user's responsibility what returns the server.

Resource Requests and Limits

```
name: my-pod
containers:
- name: my-cont
  image: my-image
  ports:
  - containerPort: 80
  resources:
    requests:
      memory: "4Mi"
      cpu: "100m"
      ephemeral-storage: "2Gi"
    limits:
      memory: "32M"
      cpu: "0.1"
      ephemeral-storage: "4Gi"
```

Cluster



Kubernetes resources refer to

CPU

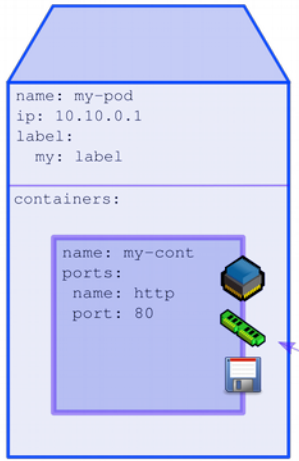
Memory

Epemeral storage (beta)

Resource Requests and Limits

```
name: my-pod
containers:
- name: my-cont
  image: my-image
  ports:
  - containerPort: 80
  resources:
    requests:
      memory: "4Mi"
      cpu: "100m"
      ephemeral-storage: "2Gi"
    limits:
      memory: "32M"
      cpu: "0.1"
      ephemeral-storage: "4Gi"
```

Cluster



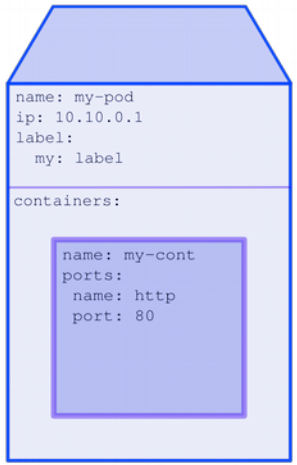
There resources are requested for each container in the pod, and...

- Resource requests will force the kernel to allocate, for a container, the amount of requested resources; regardless if it is going to use it all, or not.
- By default, if no requests are set, K8s will assign 100mCPU to each container, and no memory and storage (meaning all node resources).
  *Note: This could lead to node starvation
- Container resource requests and limits can influence the scheduler's decision of there to run the pod.
- Exceeding resource limits will force the kubelet to kill the pod.

Resources – CPU and Memory

```
name: my-pod
containers:
- name: my-cont
  image: my-image
  ports:
  - containerPort: 80
  resources:
    requests:
      memory: "4Mi"
      cpu: "100m"
    limits:
      memory: "32M"
      cpu: "0.1"
```

Cluster



- Resources refer to hardware resources, meaning CPU and Memory.

CPU

- Measured in cpu units
- Granularity of 10mCPU
- Expressed as 0.1 or 100m

Memory

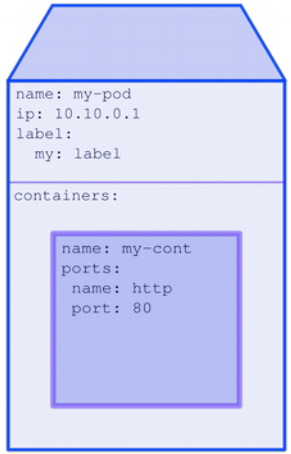
- Measured in bytes
- No granularity
- Expressed as:
  - Plain number
  - Fixed-point integer
  - Power of 10 suffixes (K, M, G, etc.)
  - Power of 2 (Ki, Mi, Gi, etc.).

- Resource requests and limits are similar to soft and hard limits, in Linux kernel
- If not specified any, by default, 0.1 CPU is going to be allocated, and no memory

Resources – Local Ephemeral Storage (beta)

```
name: my-pod
containers:
- name: db
  image: mysql
  env:
  - name: MYSQL_ROOT_PASSWORD
    value: "password"
  resources:
    requests:
      ephemeral-storage: "2Gi"
    limits:
      ephemeral-storage: "4Gi"
```

Cluster



- The storage "resource", on the node gets affected by:
  - Writable layer (Container)
  - Log usage (Container)
  - emtpyDir volumes (Pod)
- To avoid the node disk from filling in, ephemeral storage limit comes in handy
- If a Container's writable layer and logs usage exceeds its storage limit, the pod will be evicted.
- If the sum of the storage usage from all containers exceeds the total limit, the pod will be evicted.
- Measured in bytes
- No granularity
- Expressed as:
  - Plain number
  - Fixed-point integer
  - Power of 10 suffixes (K, M, G, etc.)
  - Power of 2 (Ki, Mi, Gi, etc.).

Horizontal Pod Autoscaler

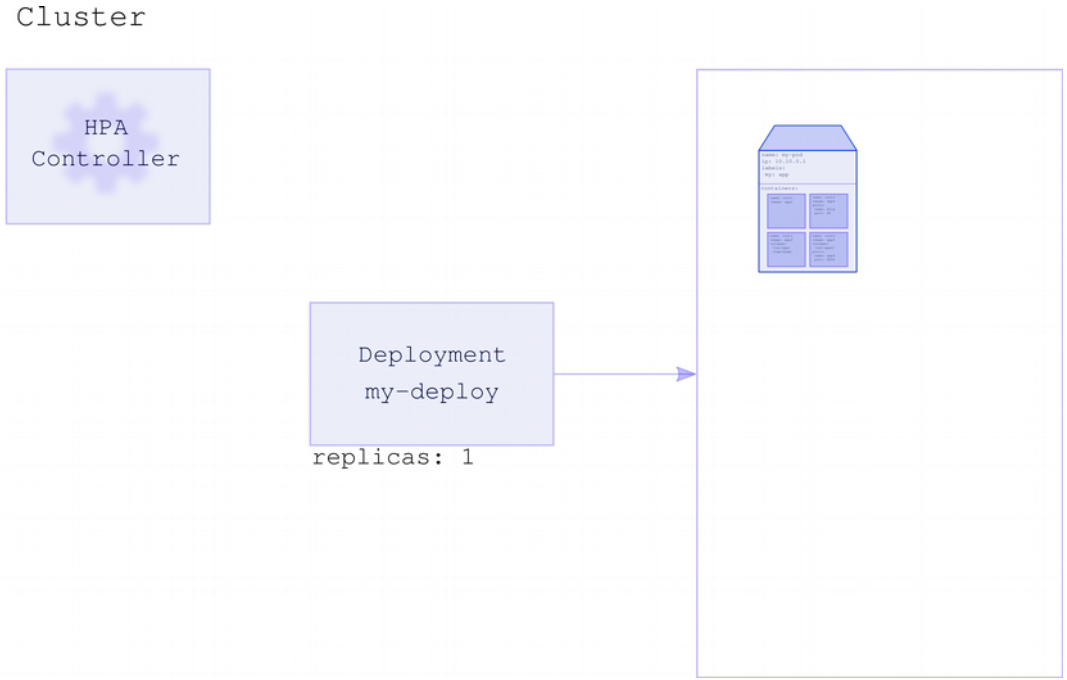
```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```

Cluster

- The Horizontal Pod Autoscaler automatically scales the number of pods in an object set, based on observed CPU utilization.
- HPA does not apply to objects that can't be scaled (e.g. DaemonSets).
- HPA is implemented as a control loop, with a period of 30 seconds
- During each period, HPA controller queries the resource utilization against the metrics server.
- Custom metrics are supported.
  Note: For scaling based on custom metrics, it is necessary to collect and send the custom metrics to the metrics server.

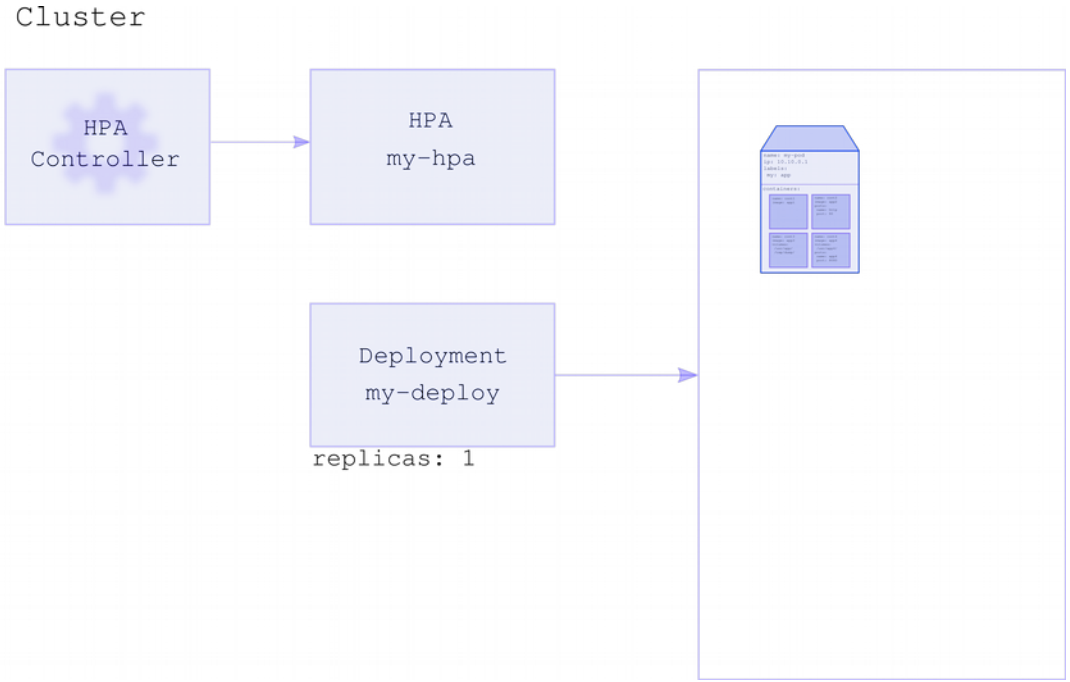
Horizontal Pod Autoscaler

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



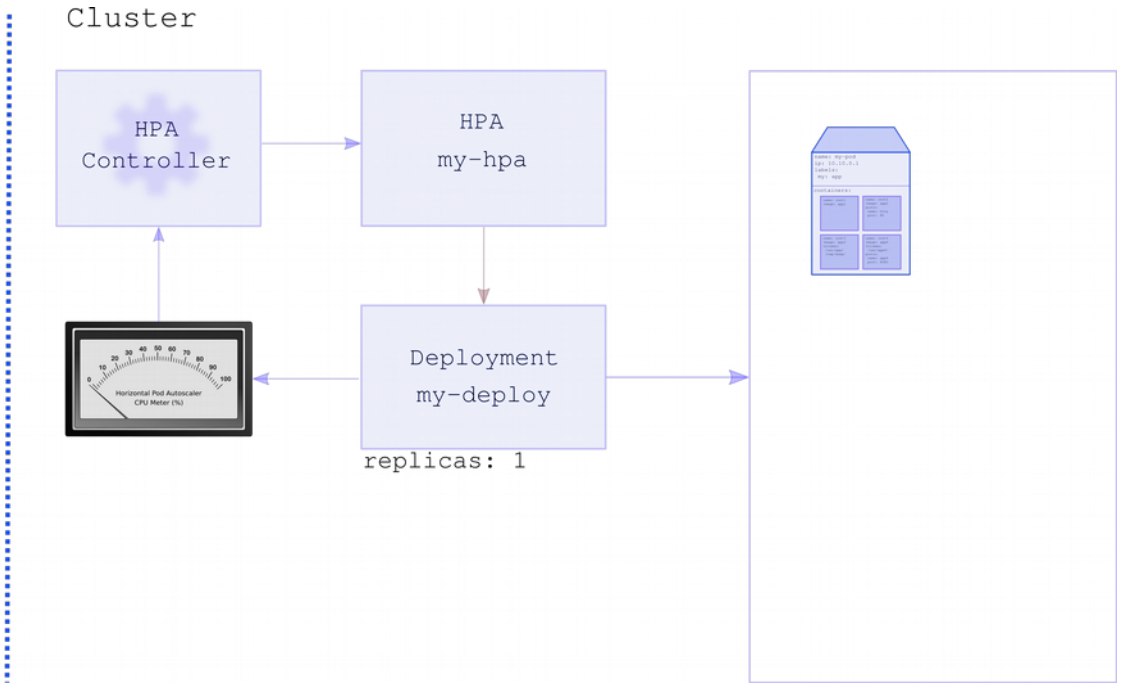
Horizontal Pod Autoscaler

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



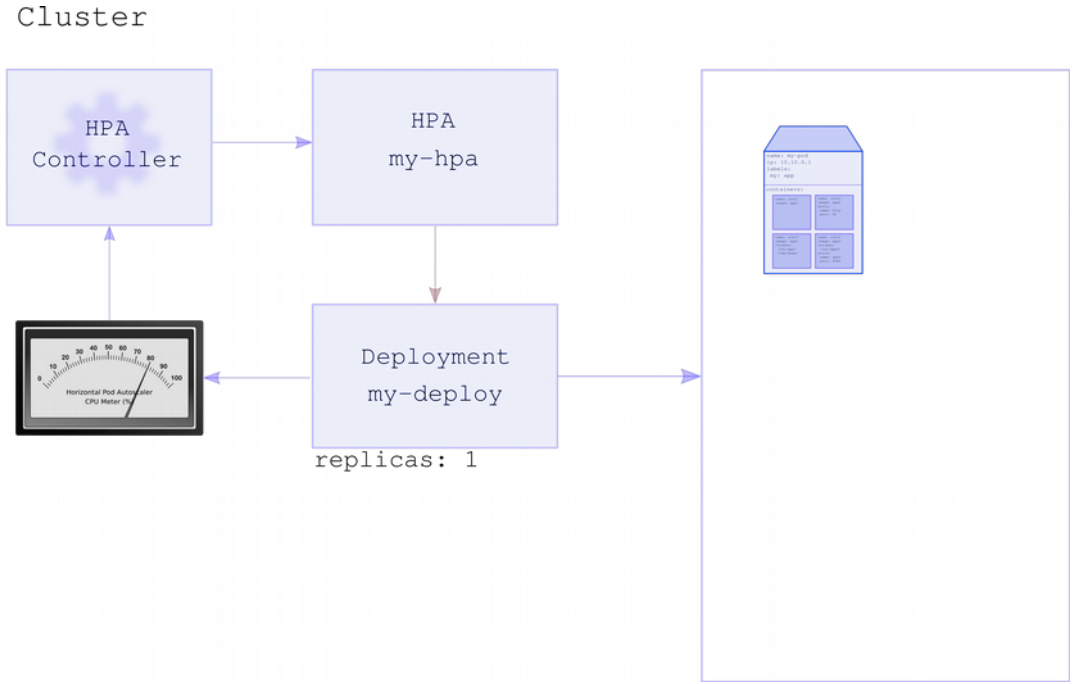
Horizontal Pod Autoscaler

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



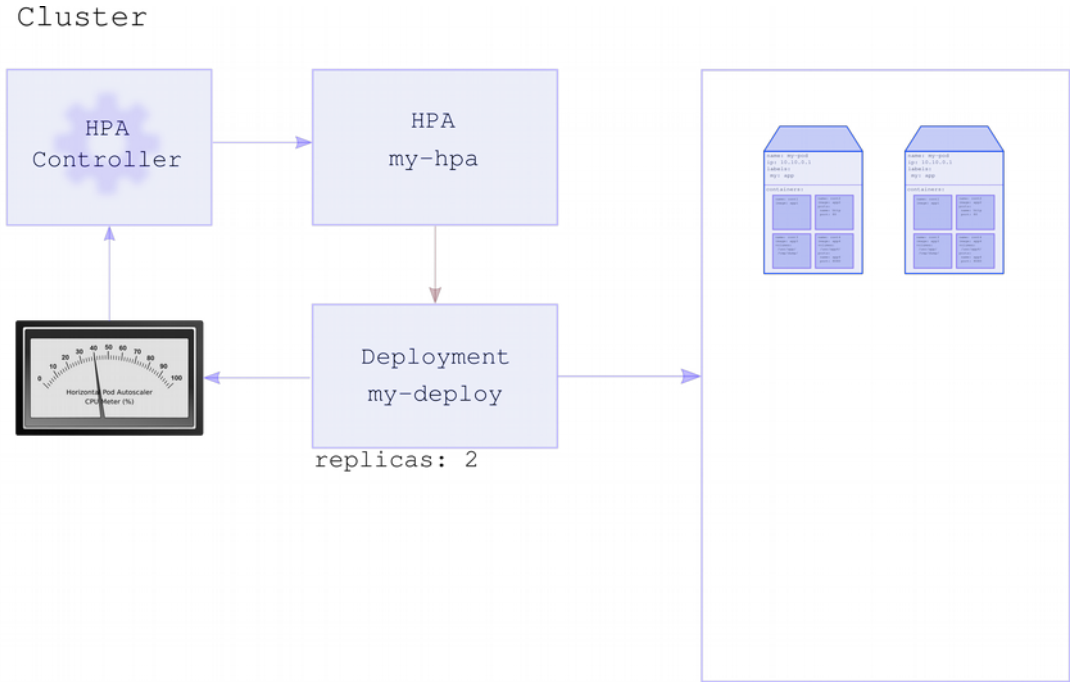
Horizontal Pod Autoscaler

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



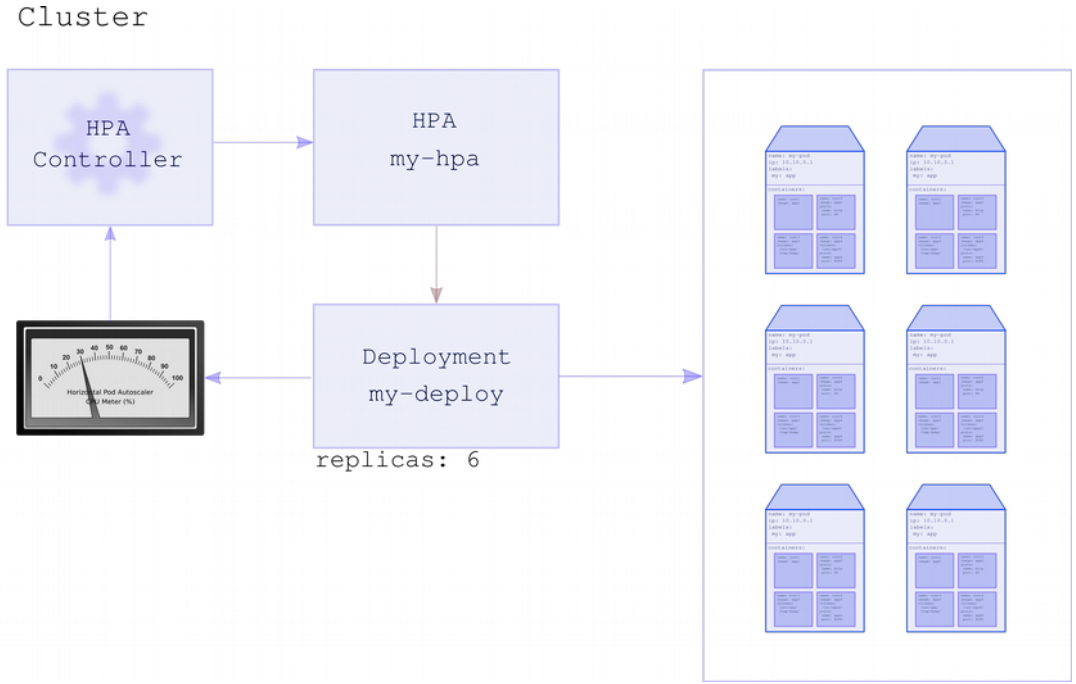
Horizontal Pod Autoscaler

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



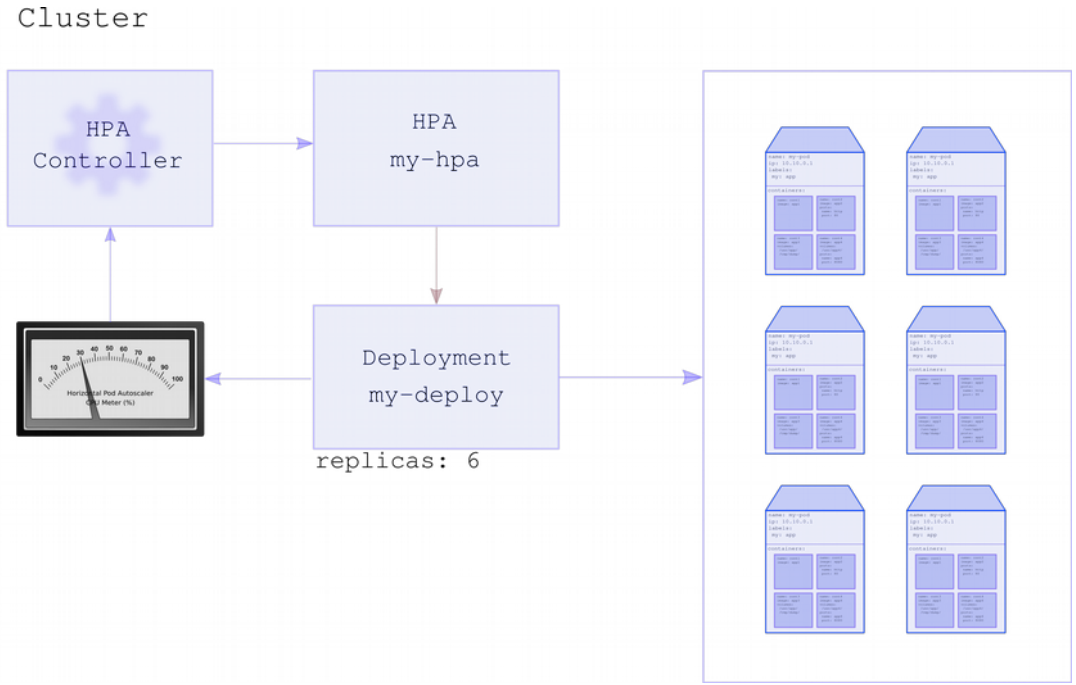
Horizontal Pod Autoscaler

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



Lab

```
name: my-hpa
maxReplicas: 6
minReplicas: 1
scaleTargetRef:
  apiVersion: extensions/v1beta1
  kind: Deployment
  name: my-deploy
targetCPUUtilizationPercentage: 80
```



Node Selector

```
name: my-pod
  ...
spec:
nodeSelector:
  schedule: here
```

Cluster

the scheduler

node

```
label:
  schedule: here
```

node

```
label:
  other: label
```

Node Selector

```
name: my-pod
  ...
spec:
nodeSelector:
  schedule: here
```

Cluster

the scheduler

node

```
label:
  schedule: here
```

node

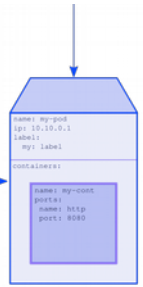
```
label:
  other: label
```

Node Selector

```
name: my-pod
  ...
spec:
nodeSelector:
  schedule: here
```

Cluster

the scheduler



node

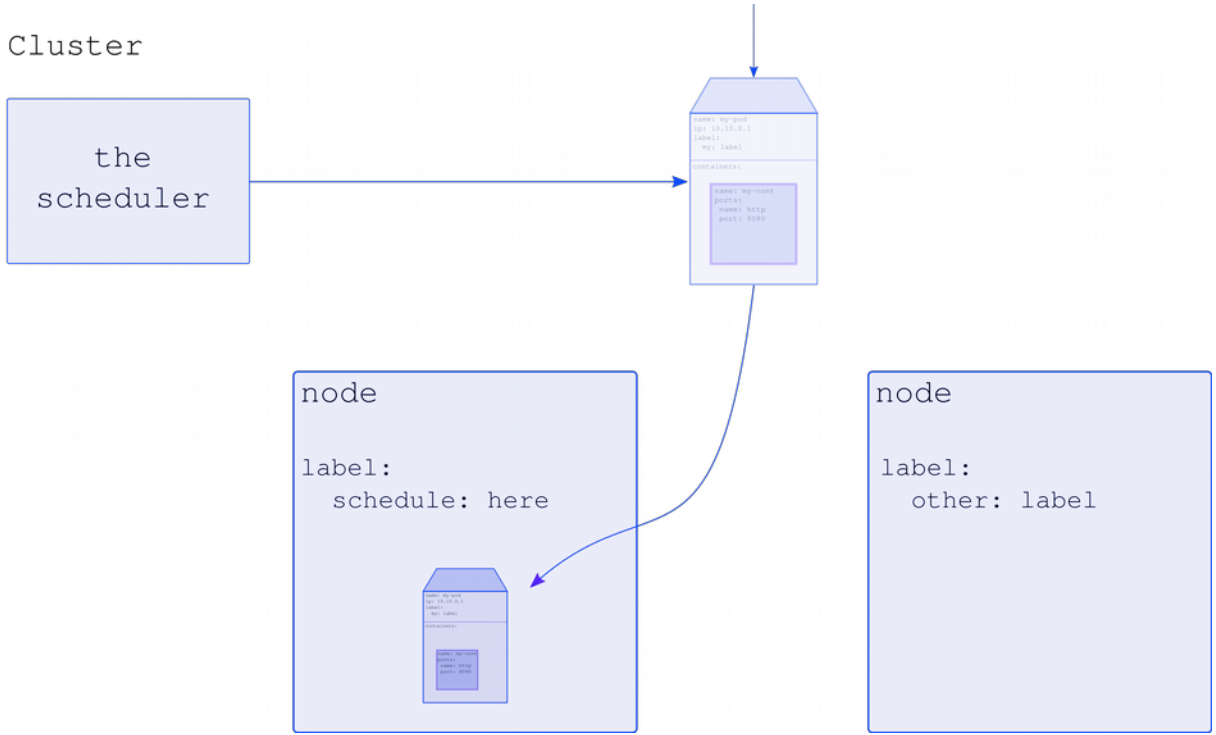
```
label:
  schedule: here
```

node

```
label:
  other: label
```

Node Selector

```
name: my-pod
  ...
spec:
nodeSelector:
  schedule: here
```



Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    nodeSelectorTerms:
    - matchExpressions:
      - key: schedule
        operator: In
        values:
        - here
```

Cluster

the scheduler

node

```
label:
  schedule: here
```

node

```
label:
  other: label
```

Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    nodeSelectorTerms:
    - matchExpressions:
      - key: schedule
        operator: In
        values:
        - here
```

Cluster

the scheduler

node

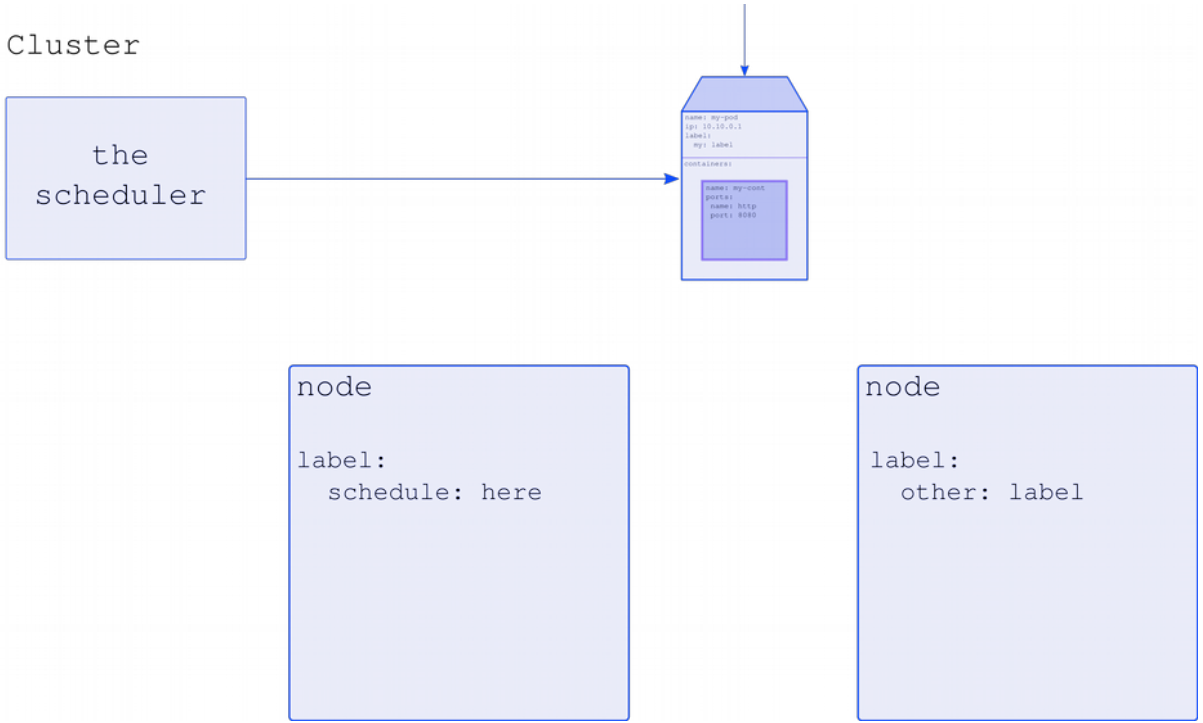
```
label:
  schedule: here
```

node

```
label:
  other: label
```

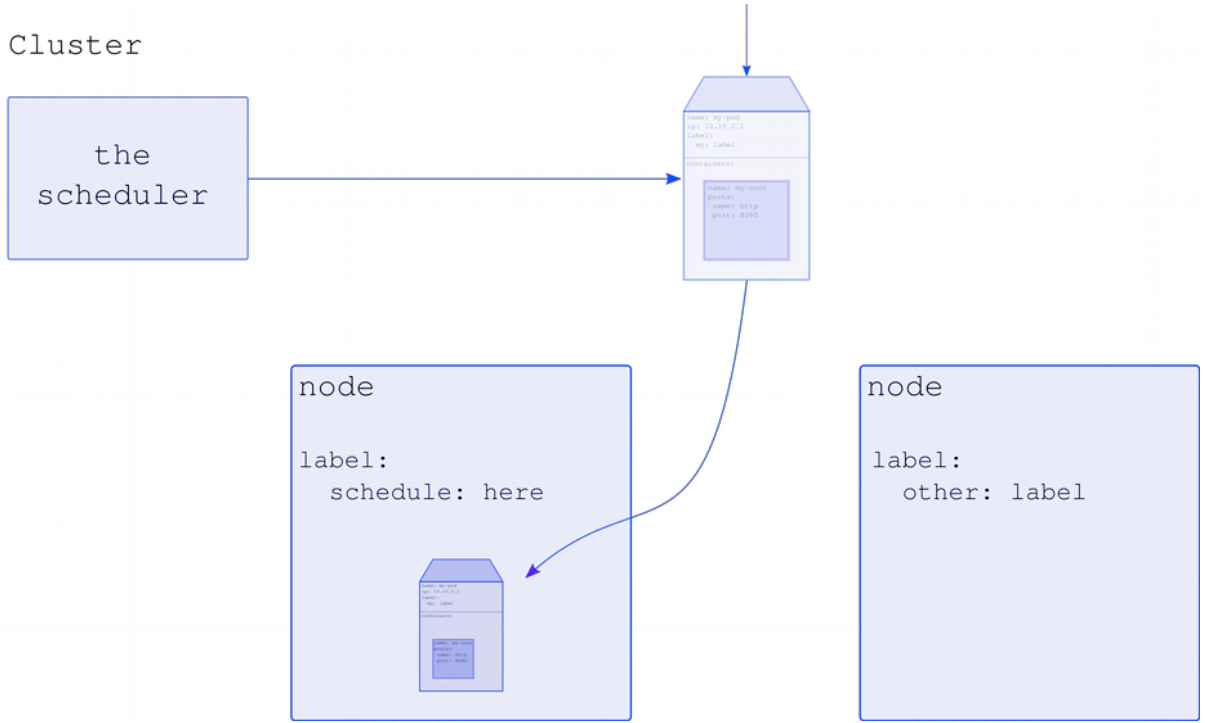
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    nodeSelectorTerms:
    - matchExpressions:
      - key: schedule
        operator: In
        values:
        - here
```



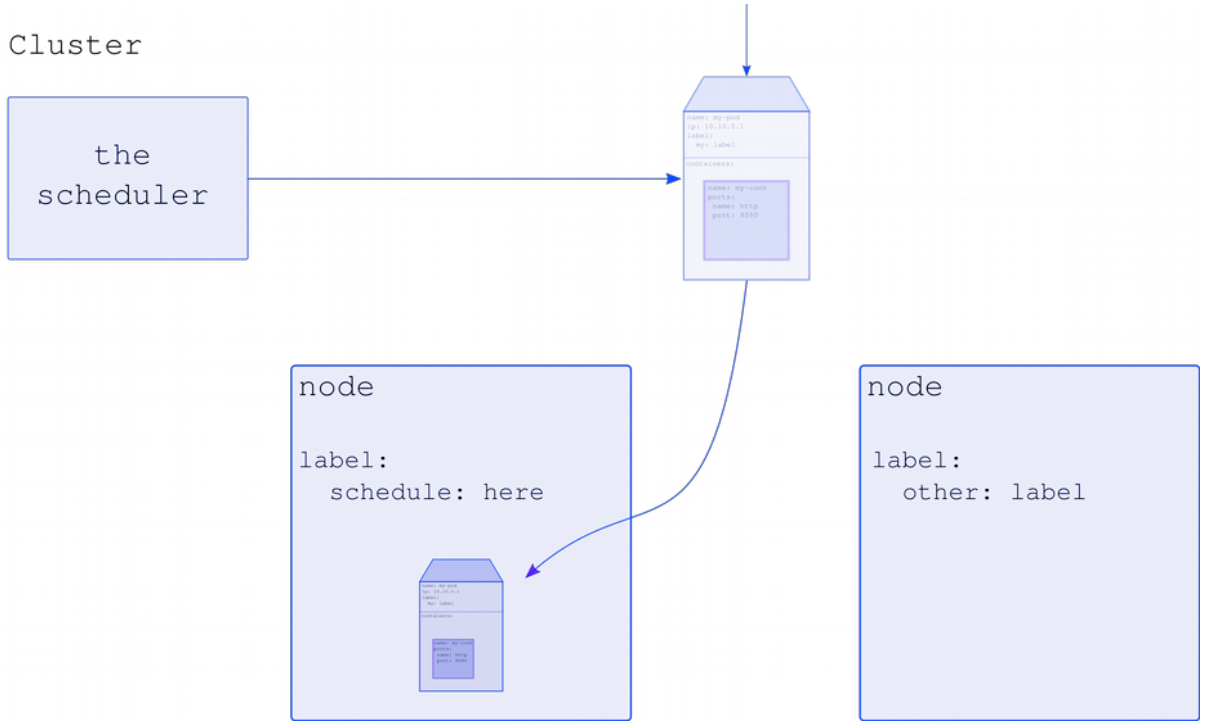
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    nodeSelectorTerms:
    - matchExpressions:
      - key: schedule
        operator: In
        values:
        - here
```



Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    nodeSelectorTerms:
    - matchExpressions:
      - key: schedule
        operator: In
        values:
        - here
```



Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```

Cluster

the scheduler

node

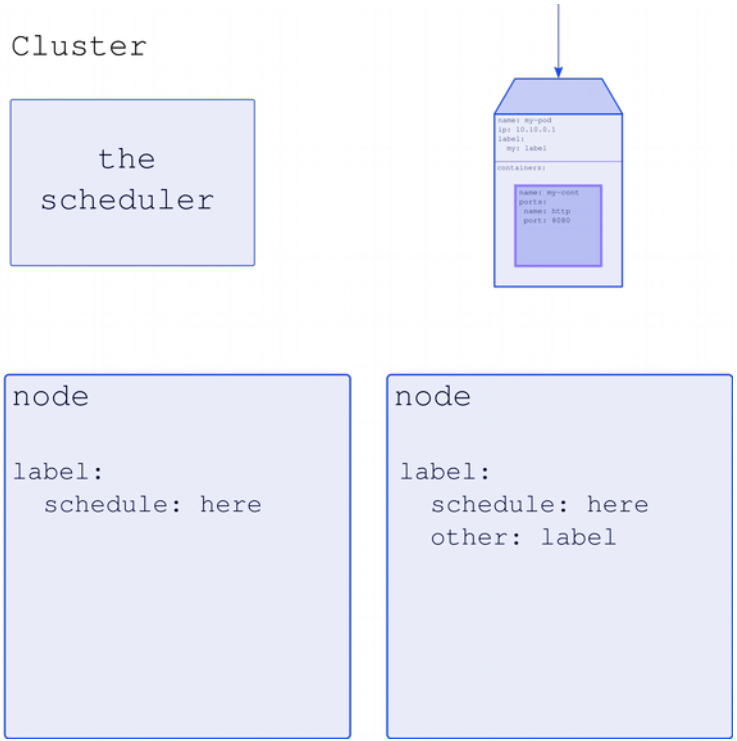
```
label:
  schedule: here
```

node

```
label:
  schedule: here
  other: label
```

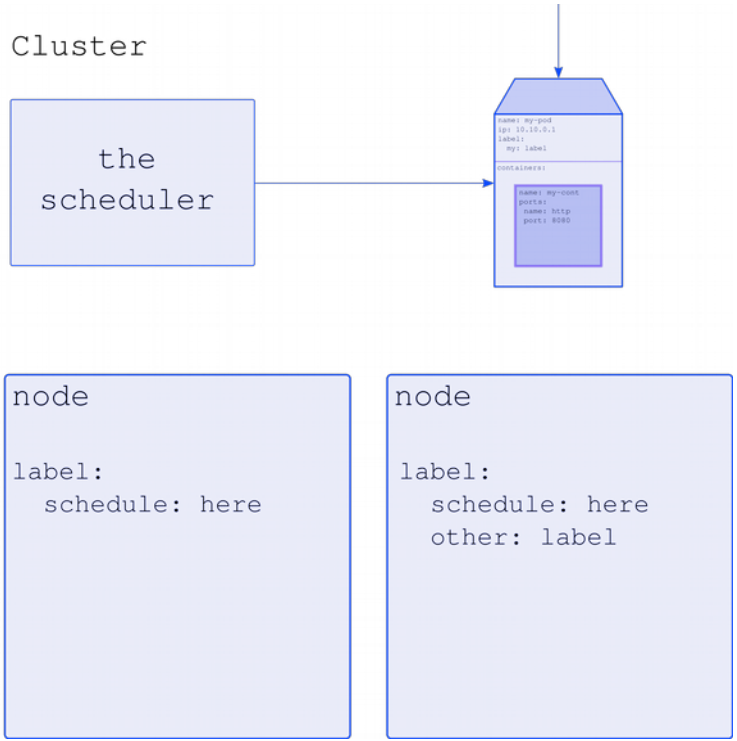
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



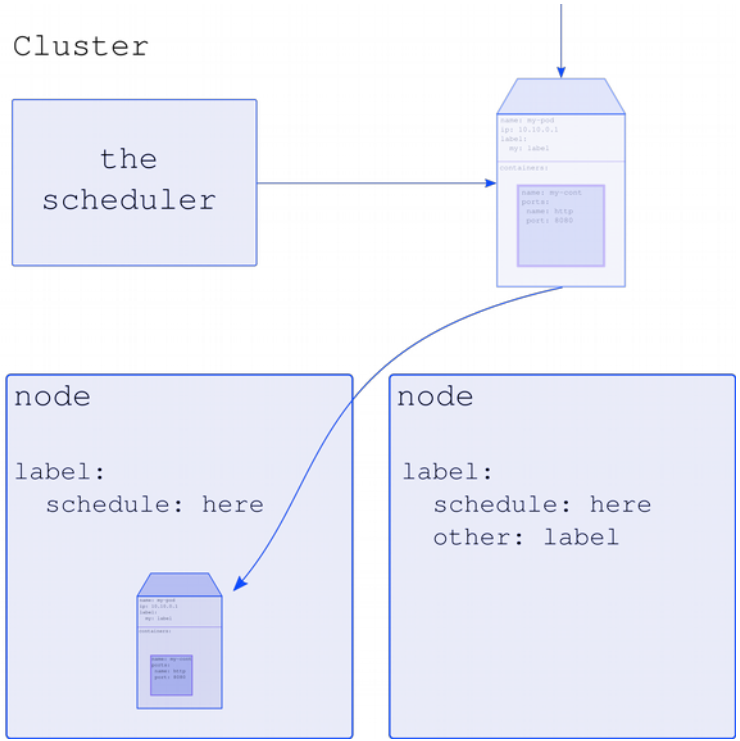
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



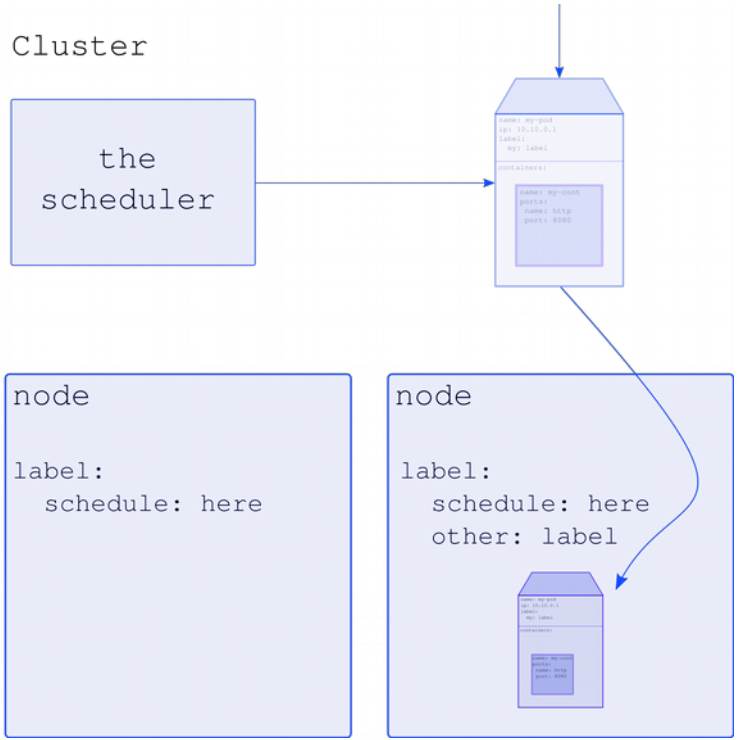
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



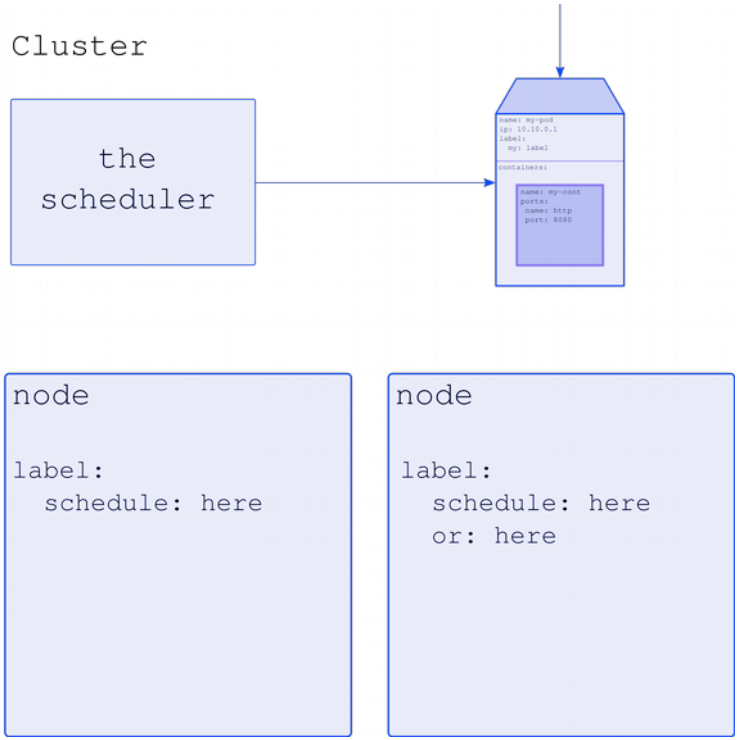
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



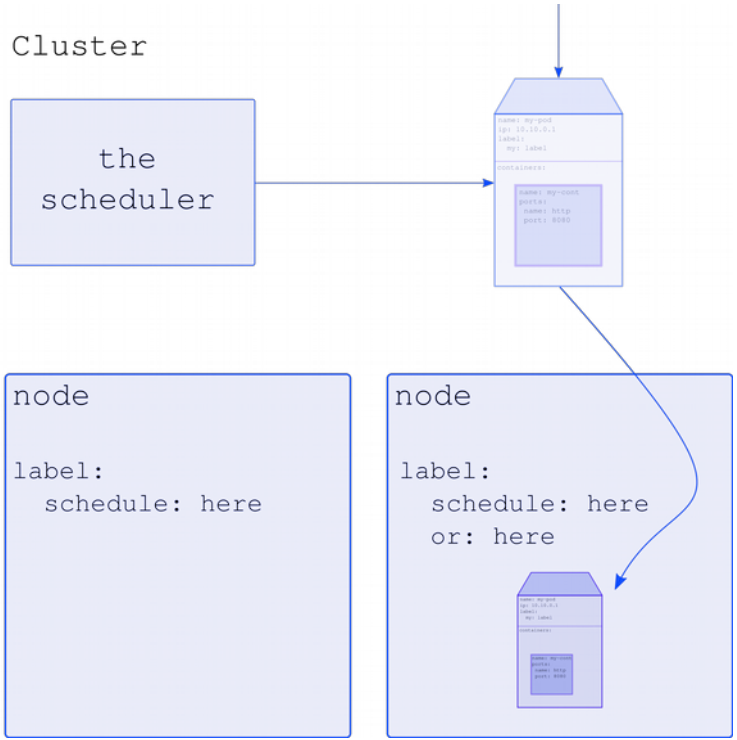
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



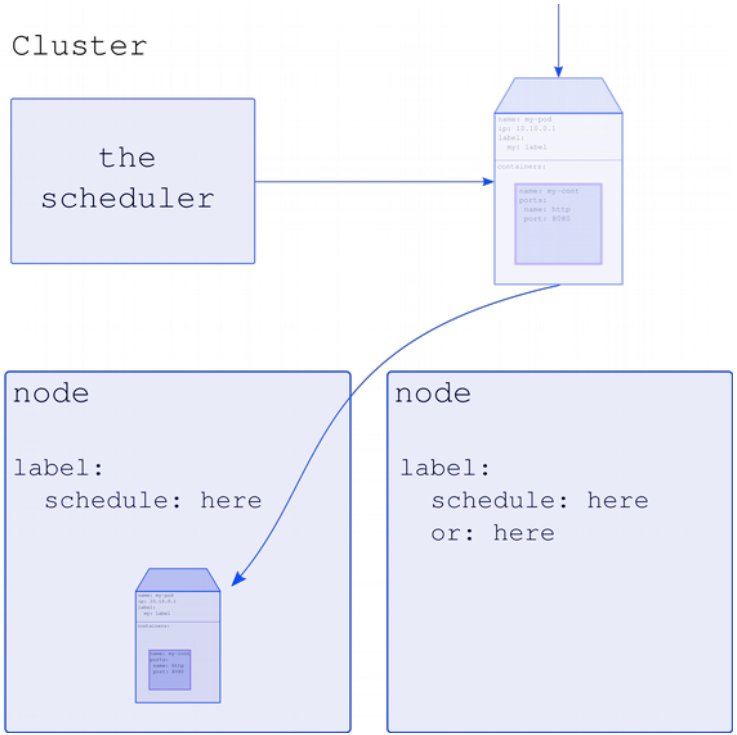
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



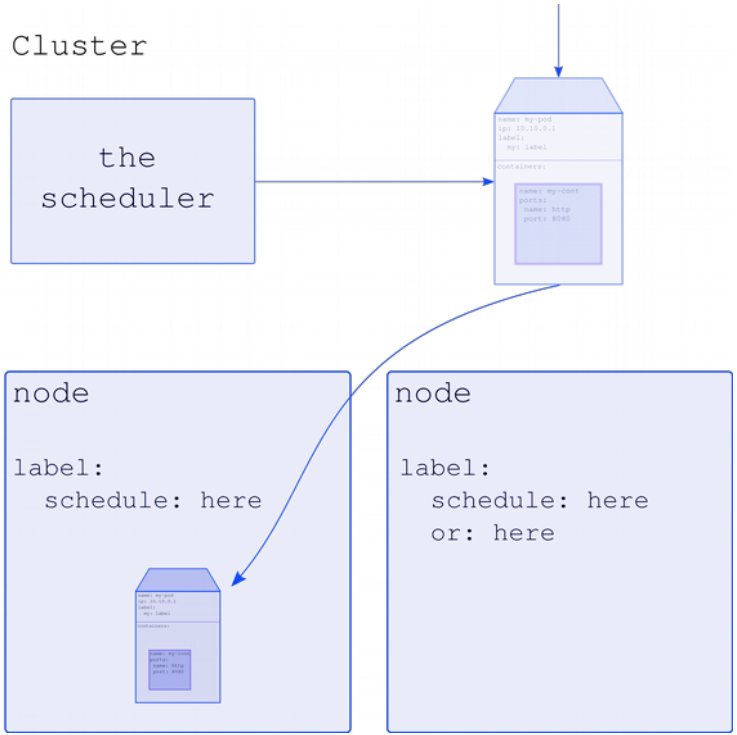
Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



Node Affinity

```
name: my-pod
  ...
spec:
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: or
          operator: In
          values:
          - here
```



Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```

Cluster

the scheduler

node

```
label:
  schedule: here
```

node

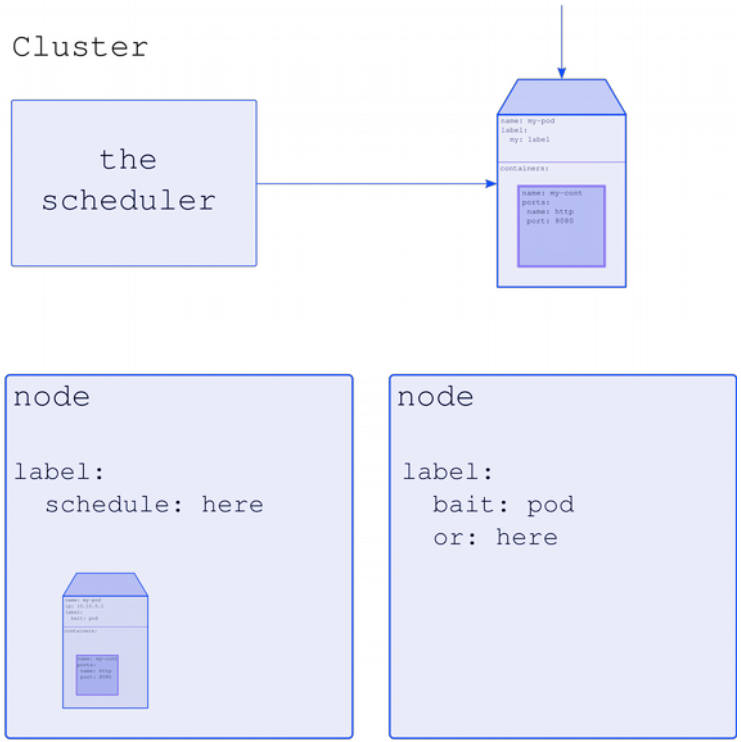
```
label:
  bait: pod
  or: here
```

Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```

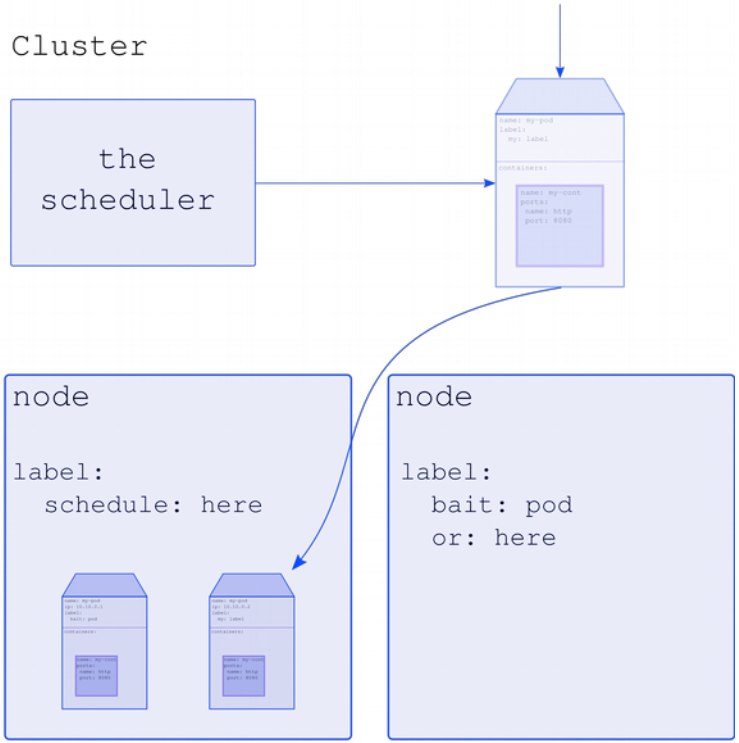
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



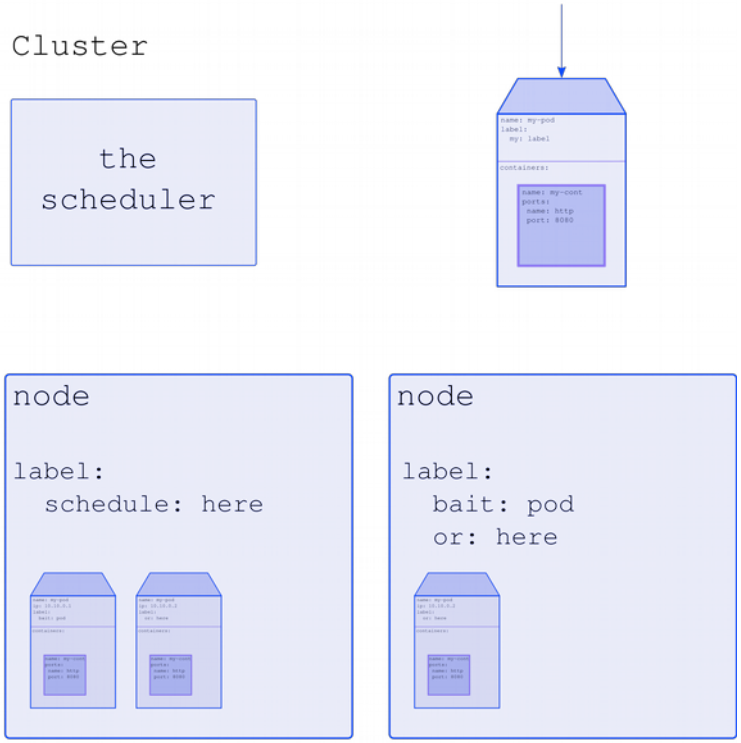
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



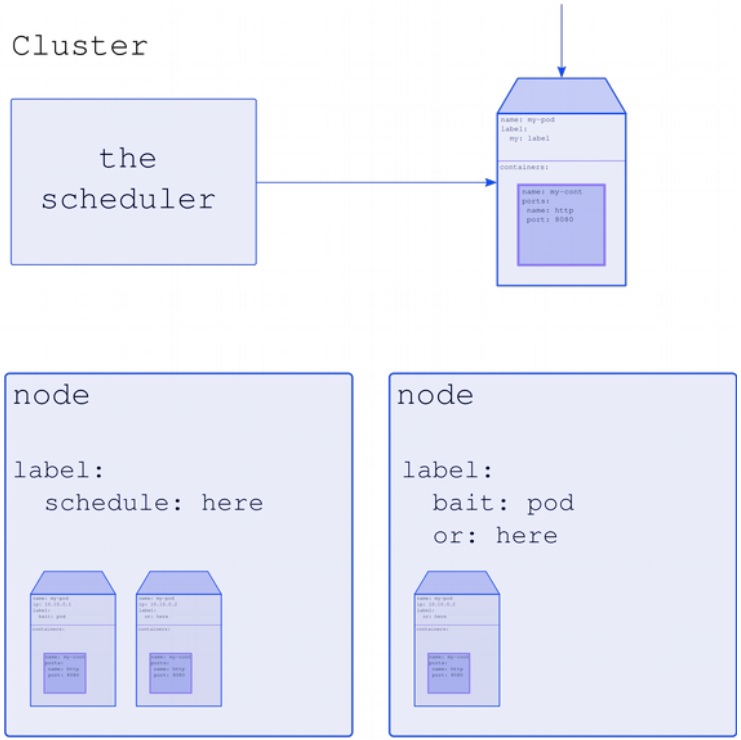
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



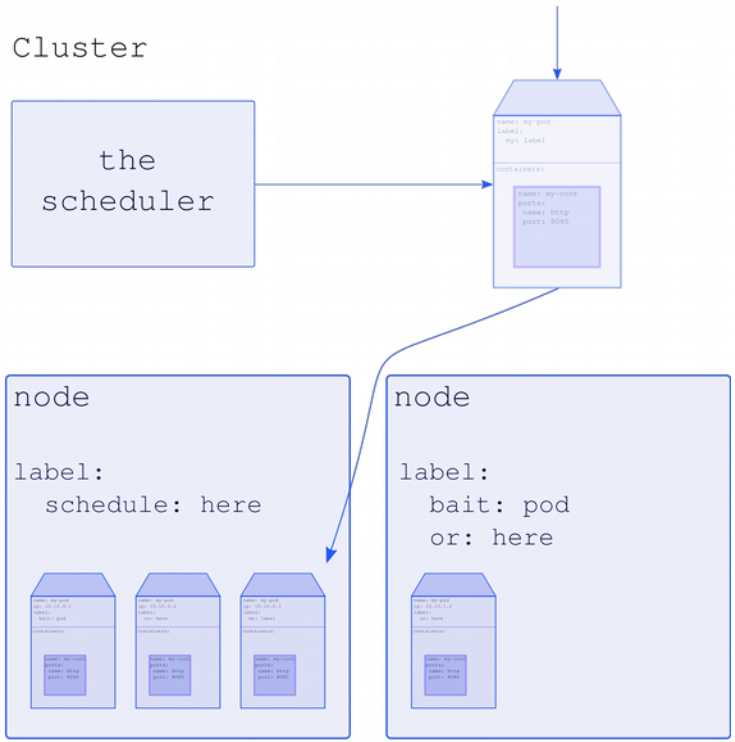
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



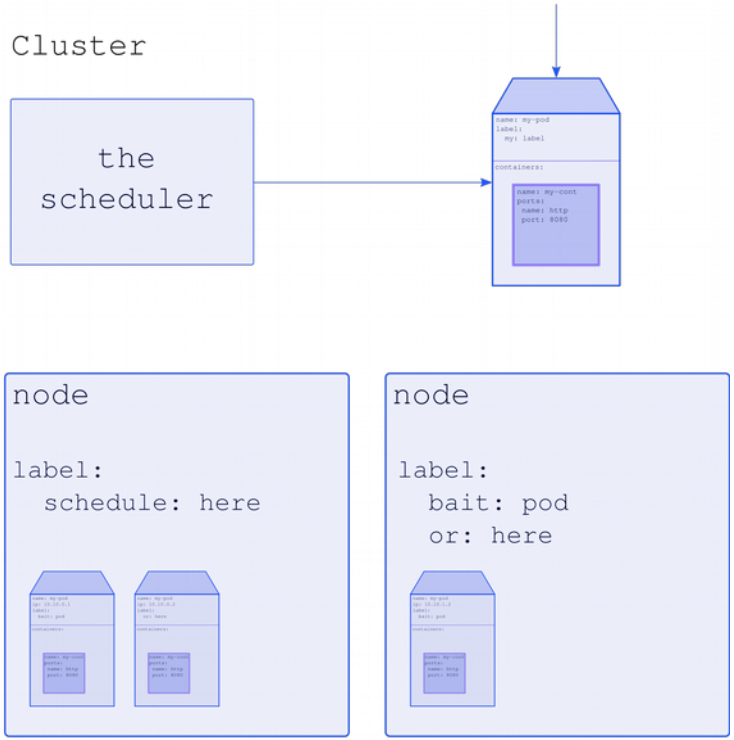
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



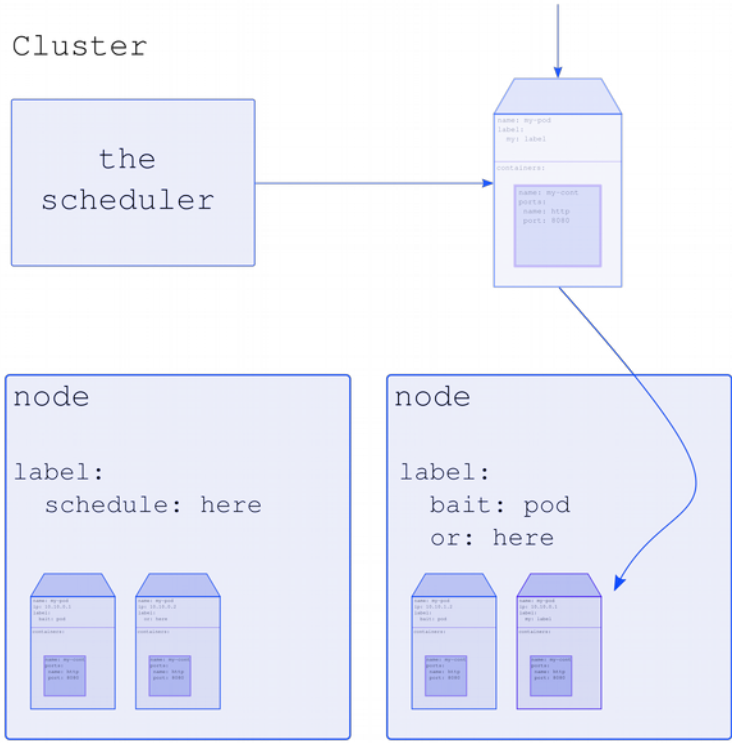
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



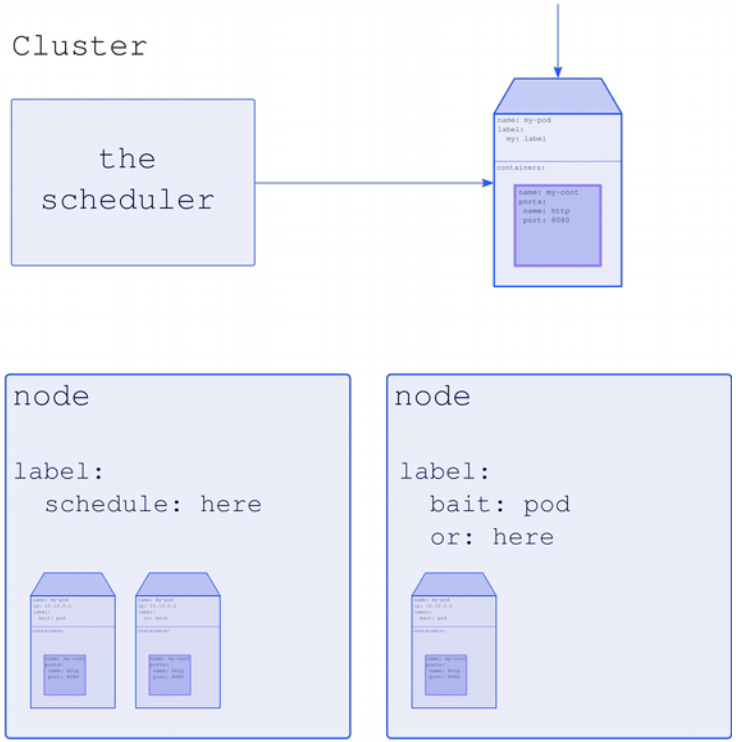
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: bait
          operator: In
          values:
          - pod
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



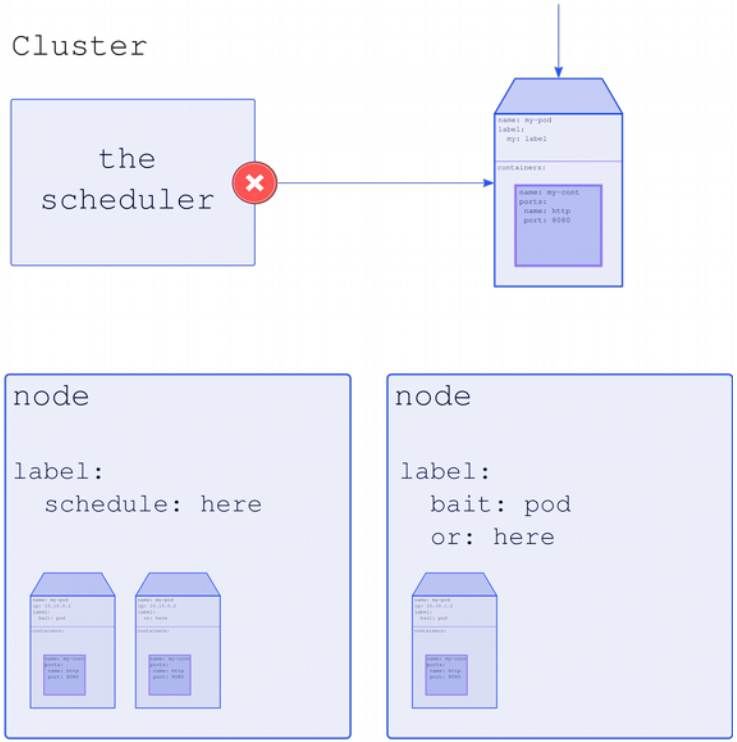
Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



Pod inter-Affinity/Anti-Affinity

```
name: my-pod
  ...
spec:
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: schedule
          operator: In
          values:
          - here
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: or
            operator: In
            values:
            - here
        topologyKey: kubernetes.io/hostname
```



Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```

Cluster

the scheduler

node

```
taints:
  node=not-ready:NoExecute
  without=gpu:NoSchedule
```

node

```
taints:
  k1:v1:PreferNoSchedule
```

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```

Cluster

the scheduler

node

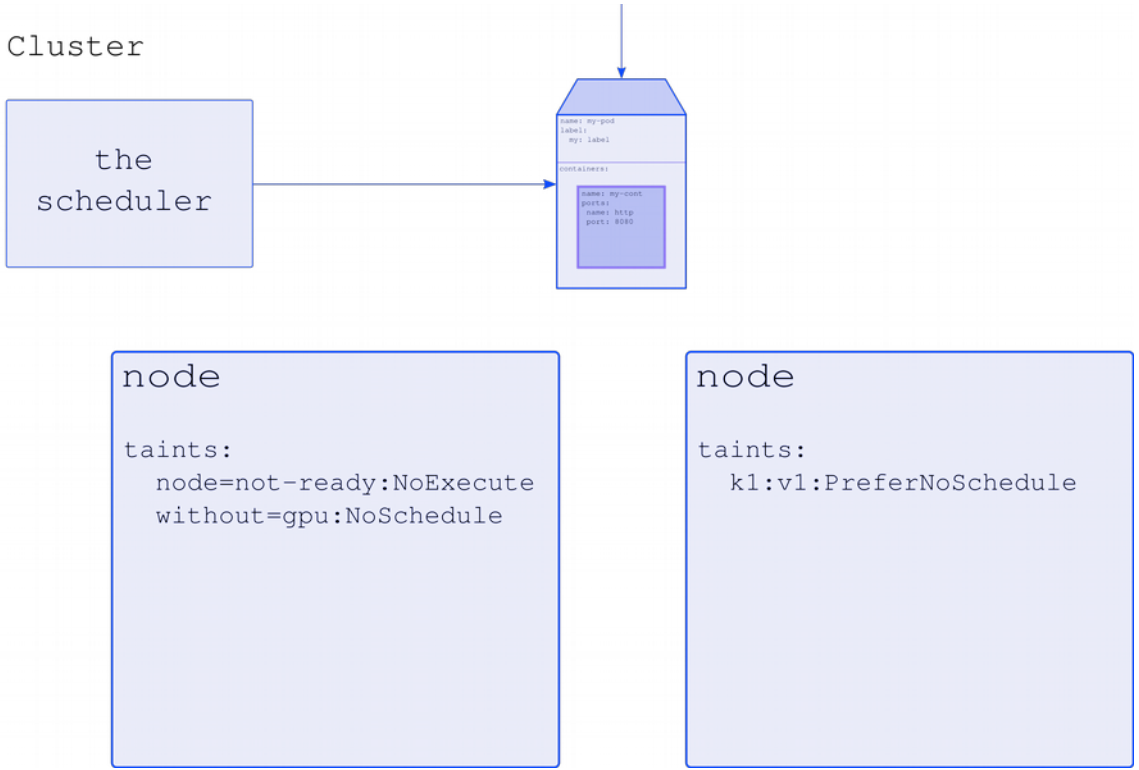
```
taints:
  node=not-ready:NoExecute
  without=gpu:NoSchedule
```

node

```
taints:
  k1:v1:PreferNoSchedule
```

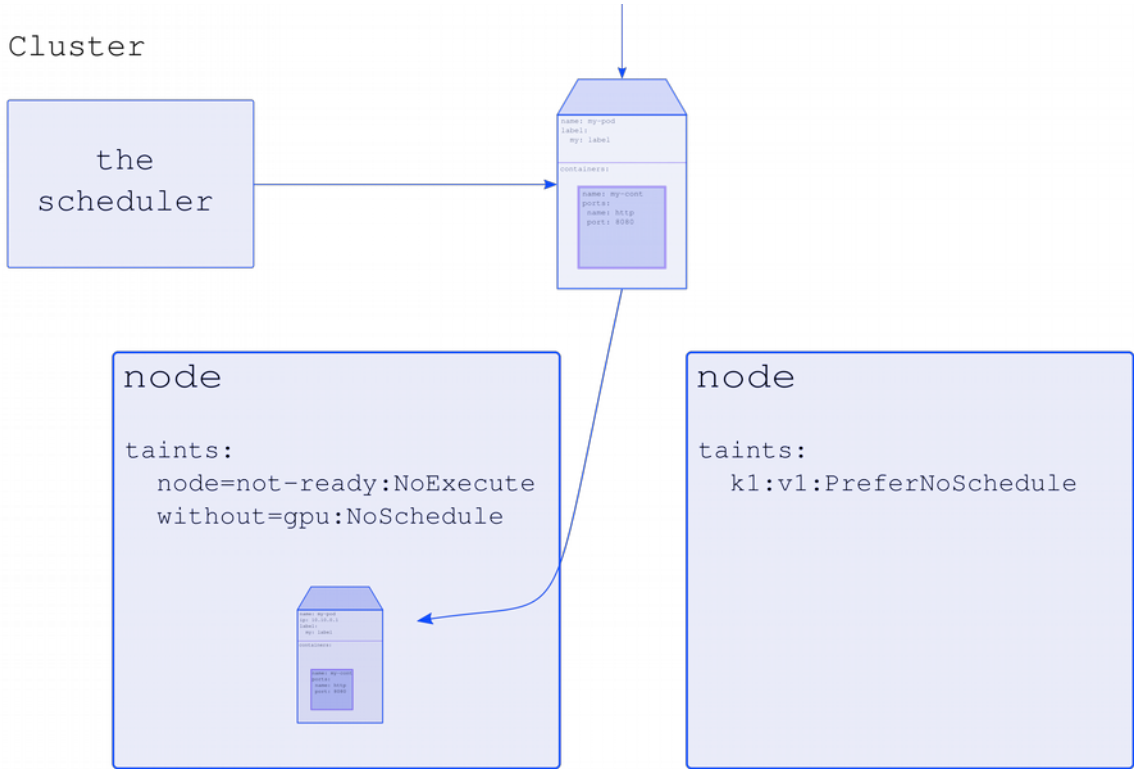
Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```



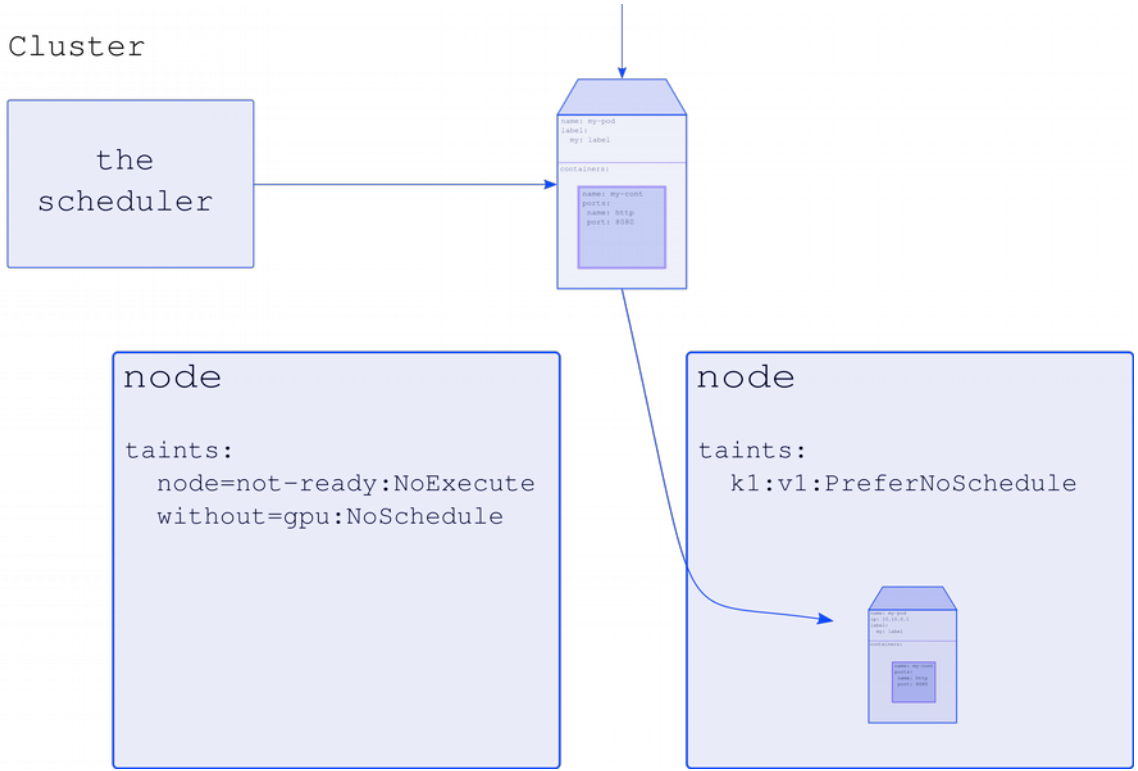
Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```



Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```



Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

taints:

node

taints:

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

taints:
  node:not-ready:NoExecute
  k1:v1:NoSchedule
  k2:v2:NoSchedule

node

taints:
  node:not-ready:NoSchedule
  key1:value1:NoExecute
  k2:v2:NoSchedule

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

```
taints:
  node:not-ready:NoExecute
  k1:v1:NoSchedule
  k2:v2:NoSchedule
```

node

```
taints:
  node:not-ready:NoSchedule
  key1:value1:NoExecute
  k2:v2:NoSchedule
```

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

t>3600

node

```
taints:
  node:not-ready:NoExecute
  k1:v1:NoSchedule
  k2:v2:NoSchedule
```

node

```
taints:
  node:not-ready:NoSchedule
  key1:value1:NoExecute
  k2:v2:NoSchedule
```

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

taints:

node

taints:

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

taints:
node:not-ready:NoExecute
k1:v1:NoSchedule

node

taints:
node:not-ready:NoSchedule
k2:v2:NoSchedule

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

taints:
node:not-ready:NoExecute
k1:v1:NoSchedule

node

taints:
node:not-ready:NoSchedule
k2:v2:NoSchedule

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

node

taints:
node:not-ready:NoExecute
k1:v1:NoSchedule

node

taints:
node:not-ready:NoSchedule
k2:v2:NoSchedule

Taints and Tolerations

```
name: my-pod
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: not-ready
  effect: NoExecute
  tolerationSeconds: 3600

- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule

- key: k2
  operator: Equal
  value: v2
  effect: NoSchedule
```

Cluster

the scheduler

t>3600

node

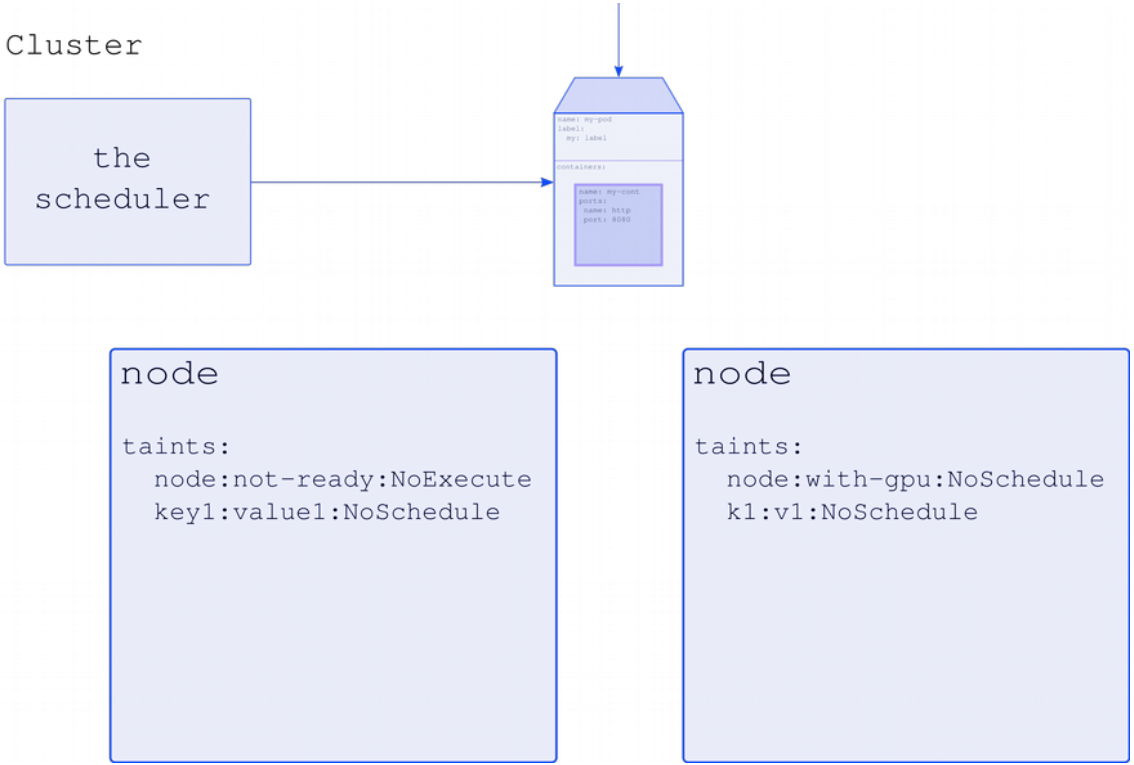
taints:
node:not-ready:NoExecute
k1:v1:NoSchedule

node

taints:
node:not-ready:NoSchedule
k2:v2:NoSchedule

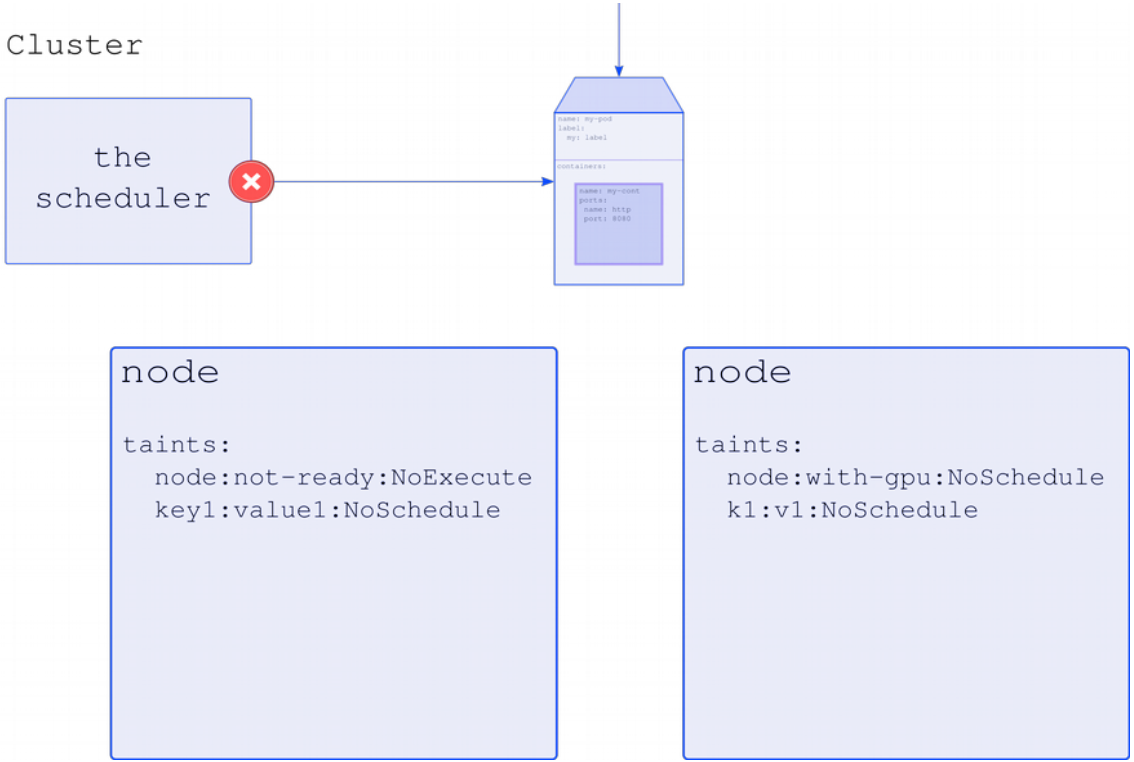
Taints and Tolerations

```
name: my-deploy
replicas: 1
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: with-gpu
  effect: NoSchedule
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```



Taints and Tolerations

```
name: my-deploy
replicas: 1
  ...
spec:
tolerations:
- key: node
  operator: Equal
  value: with-gpu
  effect: NoSchedule
- key: key1
  operator: Equal
  value: value1
  effect: NoSchedule
```



Network Policies

A network policy is a specification of how groups of pods are allowed to communicate with each other and other network endpoints. NetworkPolicy resources use labels to select pods and define rules which specify what traffic is allowed to the selected pods.

By default, pods are non-isolated entities. They become isolated by having a Network Policy that selects them. Once there is any network policy in a namespace selecting a particular pod, that pod will reject any connections that are not allowed by any NetworkPolicy.

NetworkPolicy controller is not in the core controller manager group. On GKE, Calico is an optional add-on.

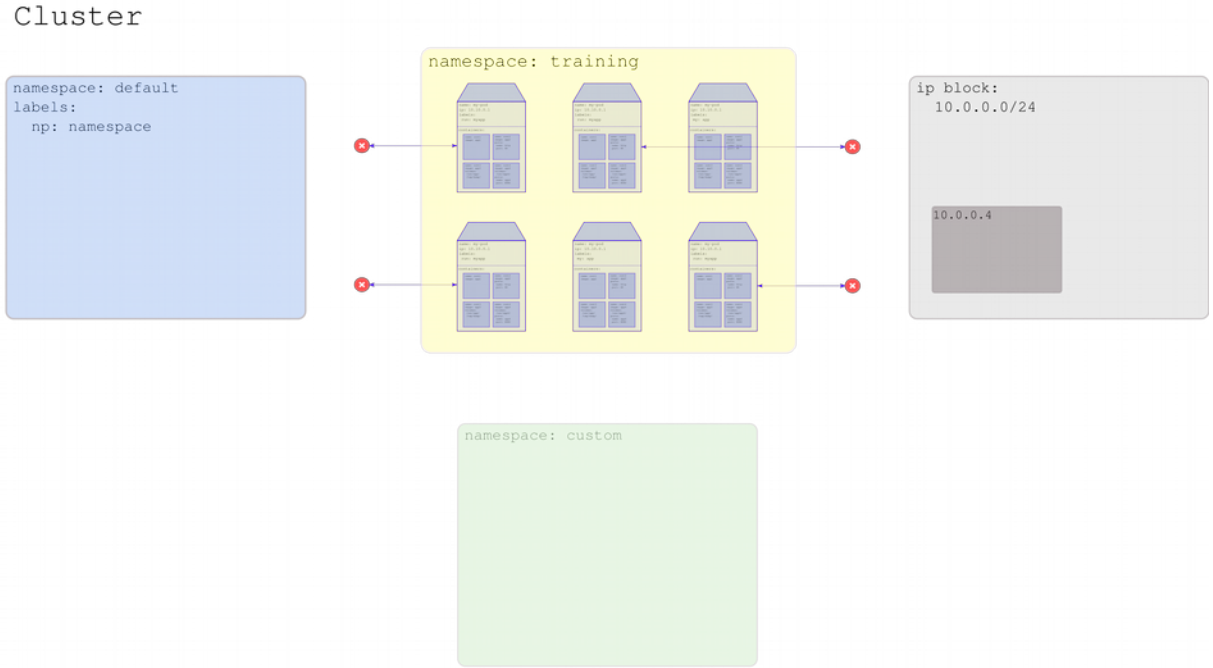
Network Policies

```
name: my-network-policy
namespace: training
  ...
spec:
podSelector:
    matchLabels:
      run: myapp
  policyTypes:
  - Ingress
  - Egress
```

Cluster

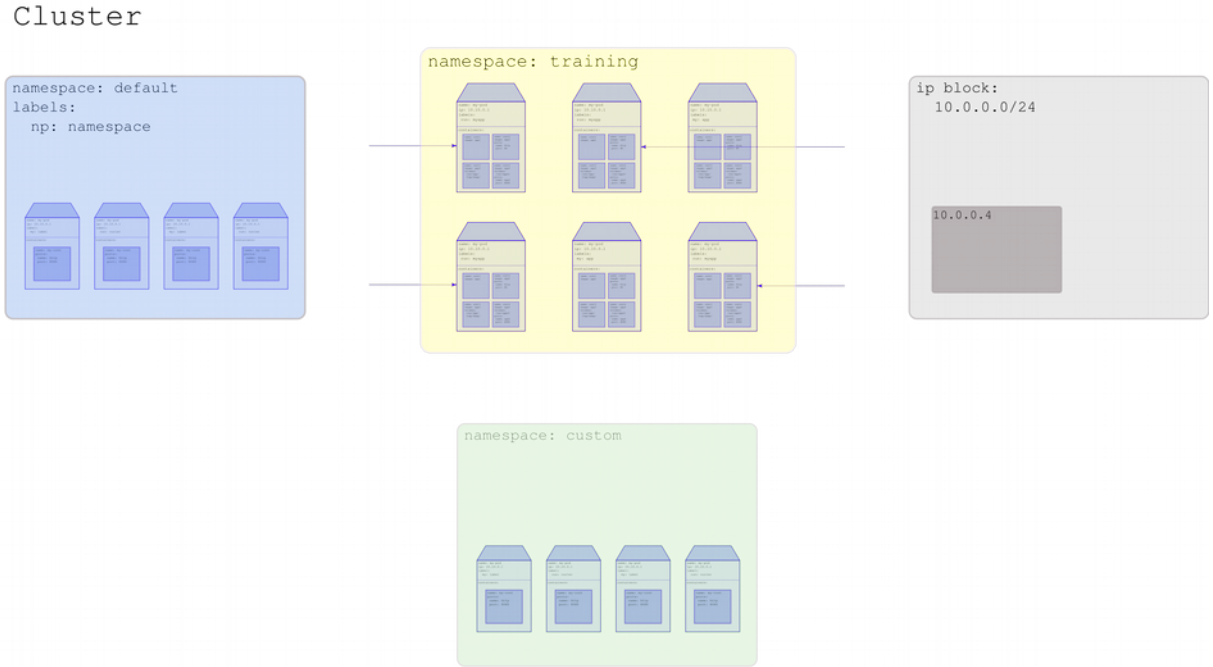
Network Policies

```
name: my-network-policy
namespace: training
  ...
spec:
podSelector:
    matchLabels:
      run: myapp
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          np: namespace
    - podSelector:
        matchLabels:
          run: curler
    ports:
    - protocol: TCP
      port: 80
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
        except:
        - 10.0.0.4/32
    ports:
    - protocol: TCP
      port: 80
```



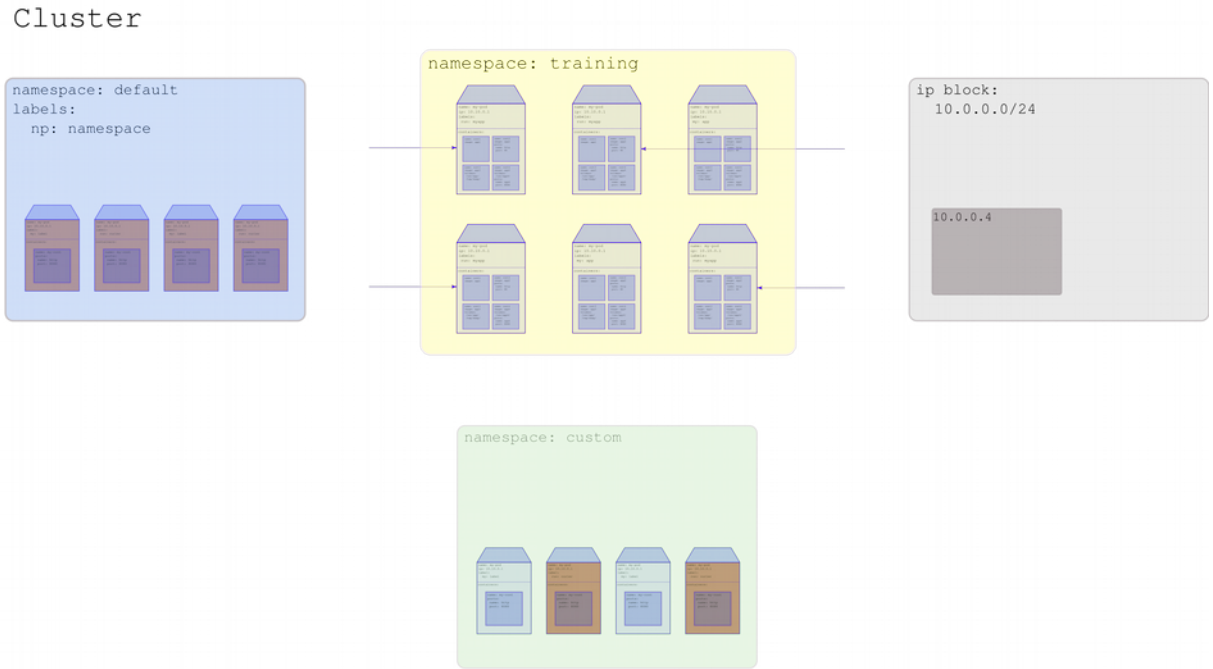
Network Policies

```
name: my-network-policy
namespace: training
  ...
spec:
podSelector:
    matchLabels:
      run: myapp
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          np: namespace
    - podSelector:
        matchLabels:
          run: curler
    ports:
    - protocol: TCP
      port: 80
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
        except:
        - 10.0.0.4/32
    ports:
    - protocol: TCP
      port: 80
```



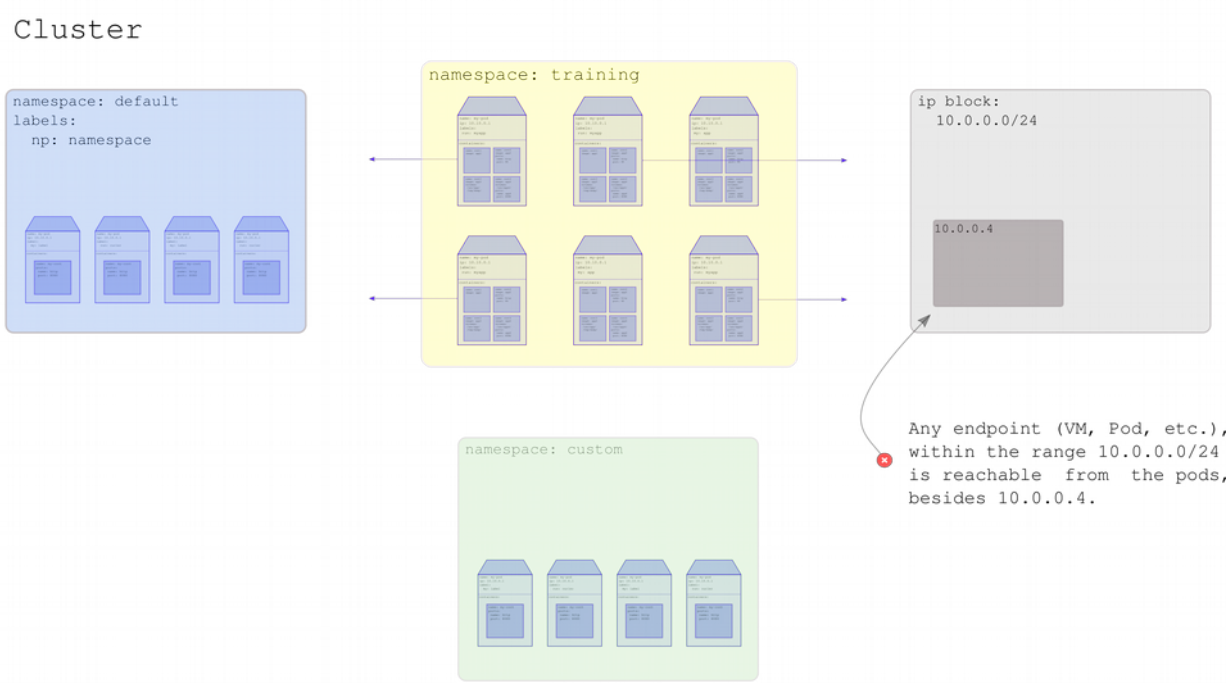
Network Policies

```
name: my-network-policy
namespace: training
  ...
spec:
podSelector:
    matchLabels:
      run: myapp
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          np: namespace
    - podSelector:
        matchLabels:
          run: curler
    ports:
    - protocol: TCP
      port: 80
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
        except:
        - 10.0.0.4/32
    ports:
    - protocol: TCP
      port: 80
```



Network Policies

```
name: my-network-policy
namespace: training
  ...
spec:
podSelector:
    matchLabels:
      run: myapp
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          np: namespace
    - podSelector:
        matchLabels:
          run: curler
    ports:
    - protocol: TCP
      port: 80
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
        except:
        - 10.0.0.4/32
    ports:
    - protocol: TCP
      port: 80
```



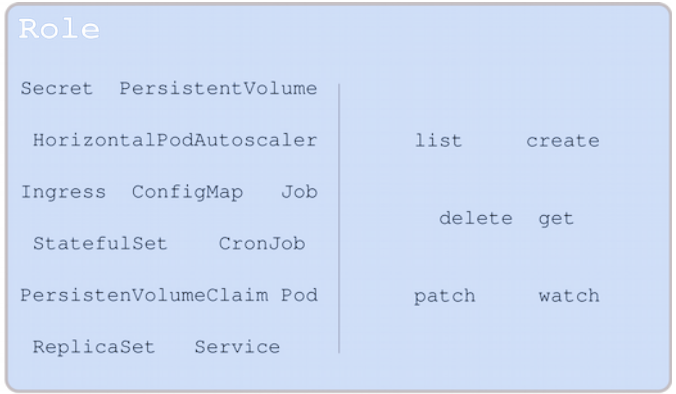
RBAC

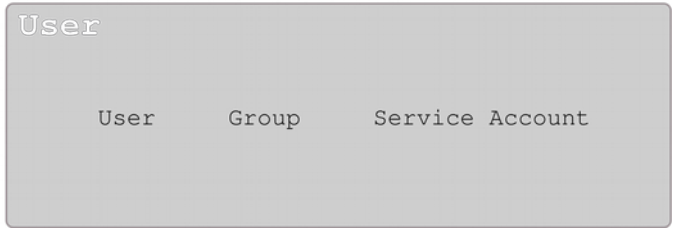
RBAC, Role-based access control, is an authorization mechanism for managing permissions around Kubernetes resources.

It binds a role, which is a set of permissions over an object, to a user.

Working with full administrator privileges, we might not have seen the importance of RBAC, but in a real environment we need to:

- Have multiple users with different properties, establishing a proper authentication mechanism.
- Have full control over which operations each user or group of users can execute.
- Have full control over which operations each process inside a pod can execute.
- Limit the visibility of certain resources of namespaces.





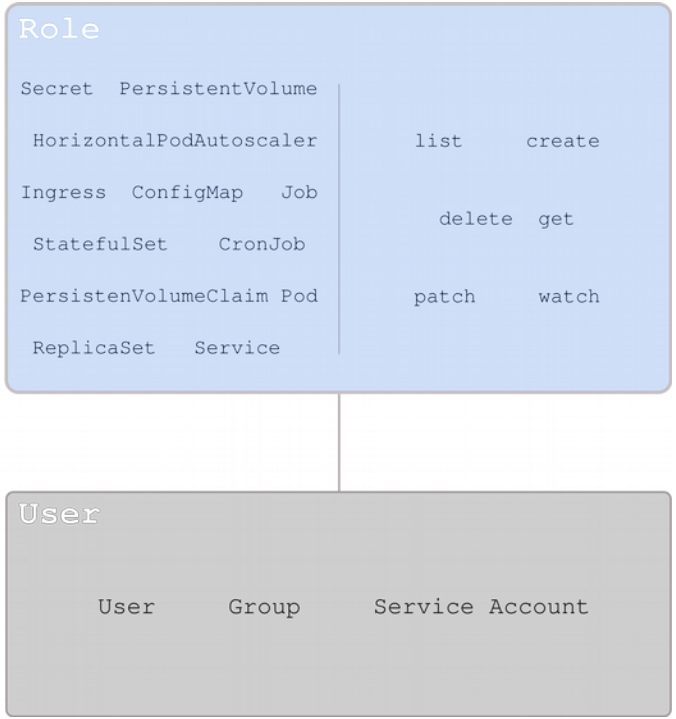
RBAC

RBAC, Role-based access control, is an authorization mechanism for managing permissions around Kubernetes resources.

It binds a role, which is a set of permissions over an object, to a user.

Working with full administrator privileges, we might not have seen the importance of RBAC, but in a real environment we need to:

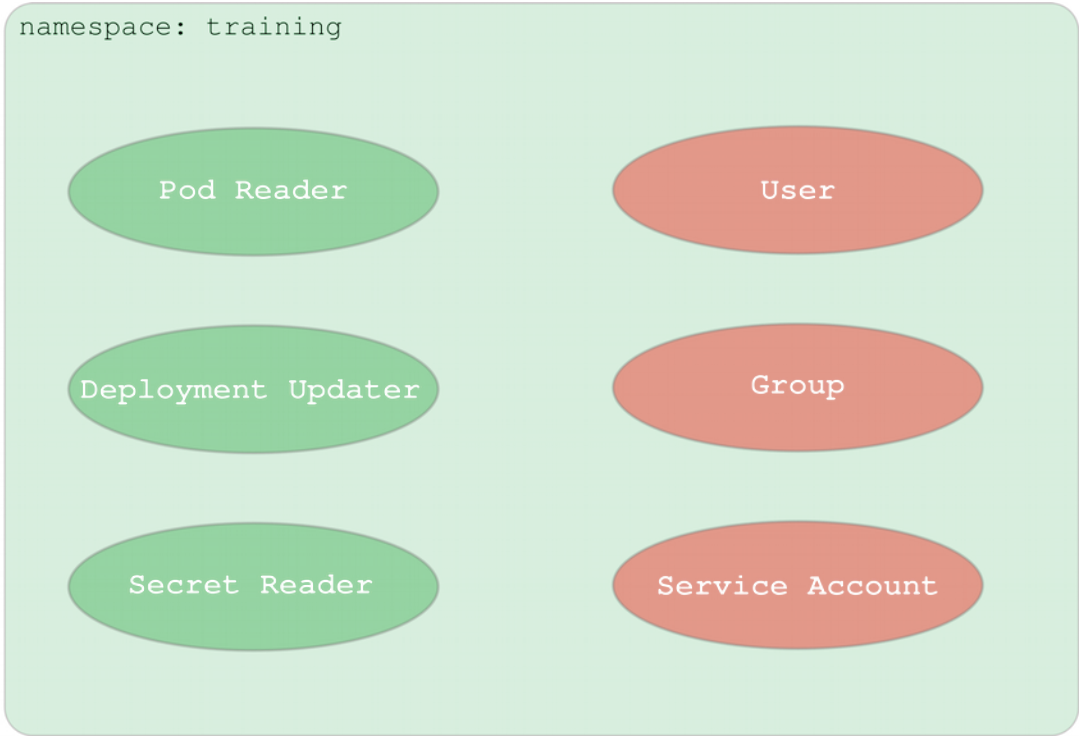
- Have multiple users with different properties, establishing a proper authentication mechanism.
- Have full control over which operations each user or group of users can execute.
- Have full control over which operations each process inside a pod can execute.
- Limit the visibility of certain resources of namespaces.



RBAC

```
kind: Role
name: pod-reader
namespace: training
  ...
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

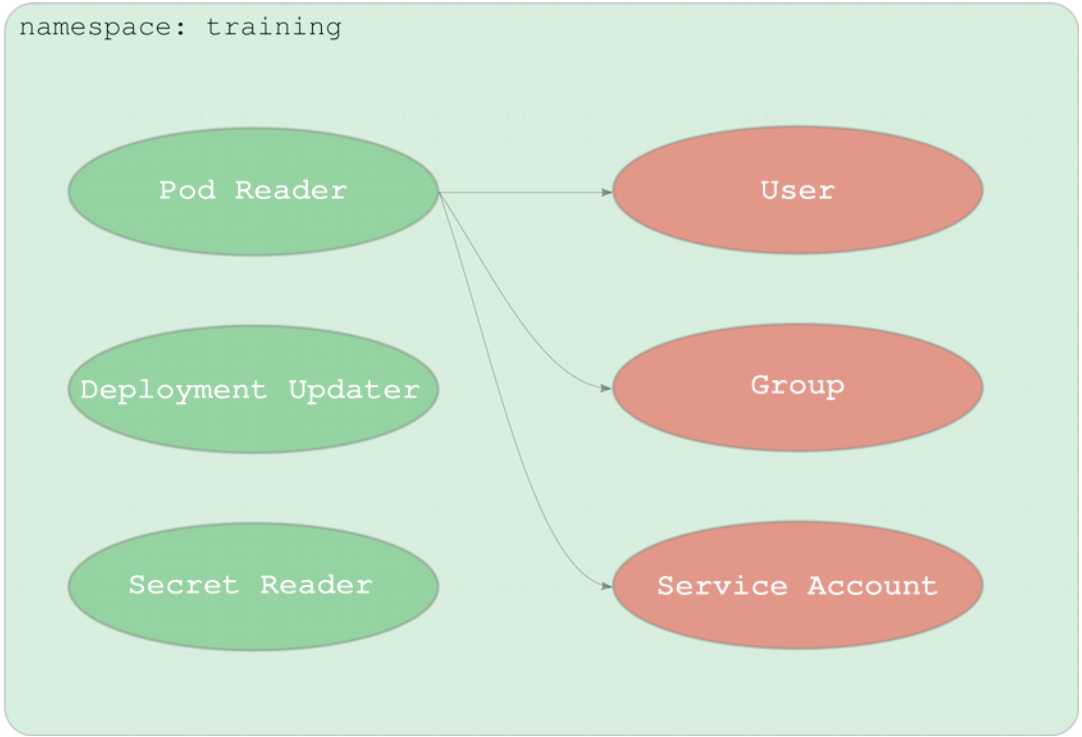
```
kind: RoleBinding
name: read-pods
namespace: training
subjects:
- kind: User
  name: my-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```



RBAC – RoleBinding

```
kind: Role
name: pod-reader
namespace: training
  ...
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

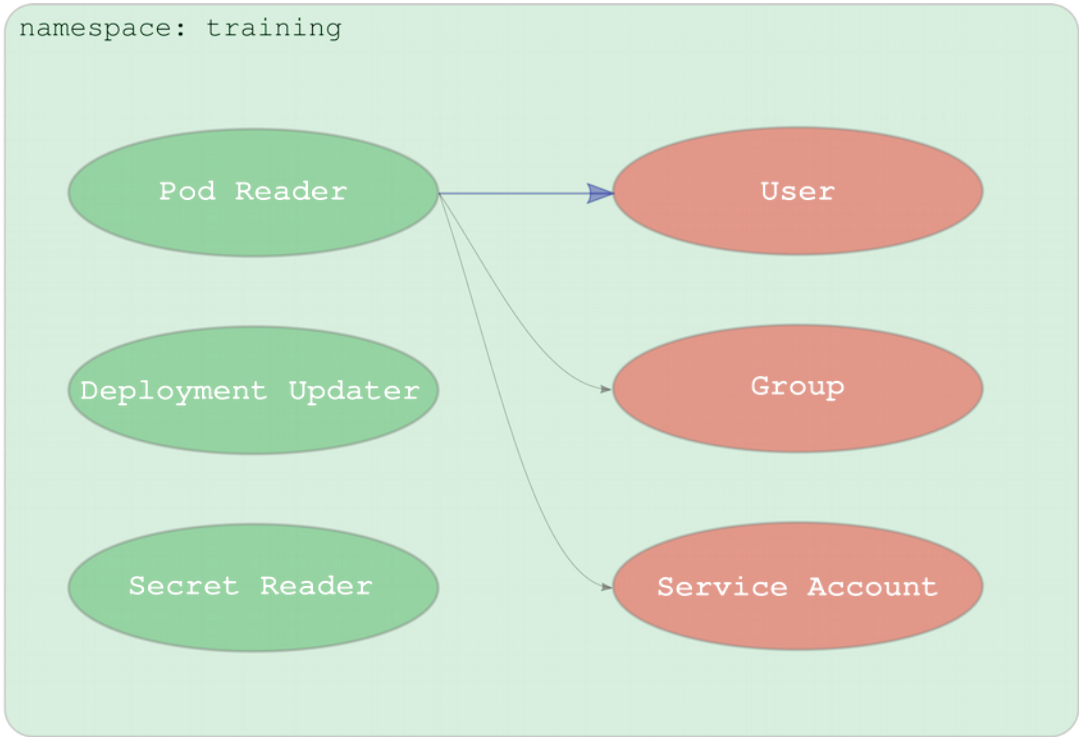
```
kind: RoleBinding
name: read-pods
namespace: training
subjects:
- kind: User
  name: my-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```



RBAC – RoleBinding

```
kind: Role
name: pod-reader
namespace: training
  ...
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

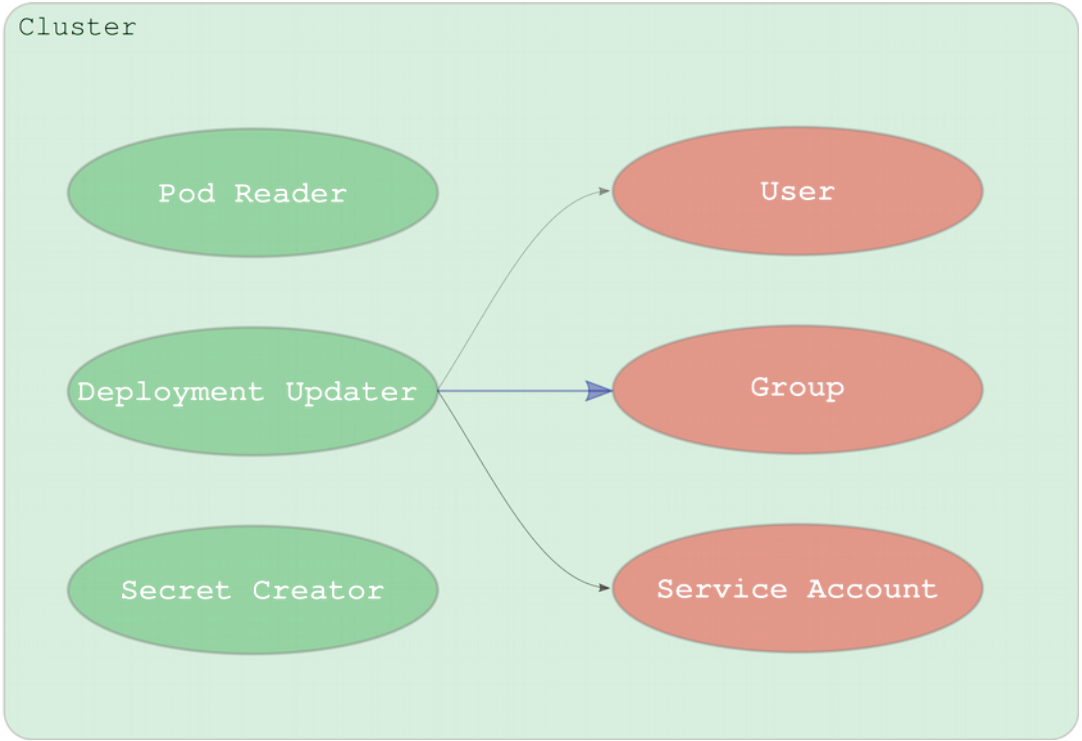
```
kind: RoleBinding
name: read-pods
namespace: training
subjects:
- kind: User
  name: my-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```



RBAC – ClusterRoleBinding

```
kind: ClusterRole
name: secret-reader
  ...
rules:
- apiGroups: [""]
  resources: ["deployments"]
  verbs: ["create", "delete", "patch"]
```

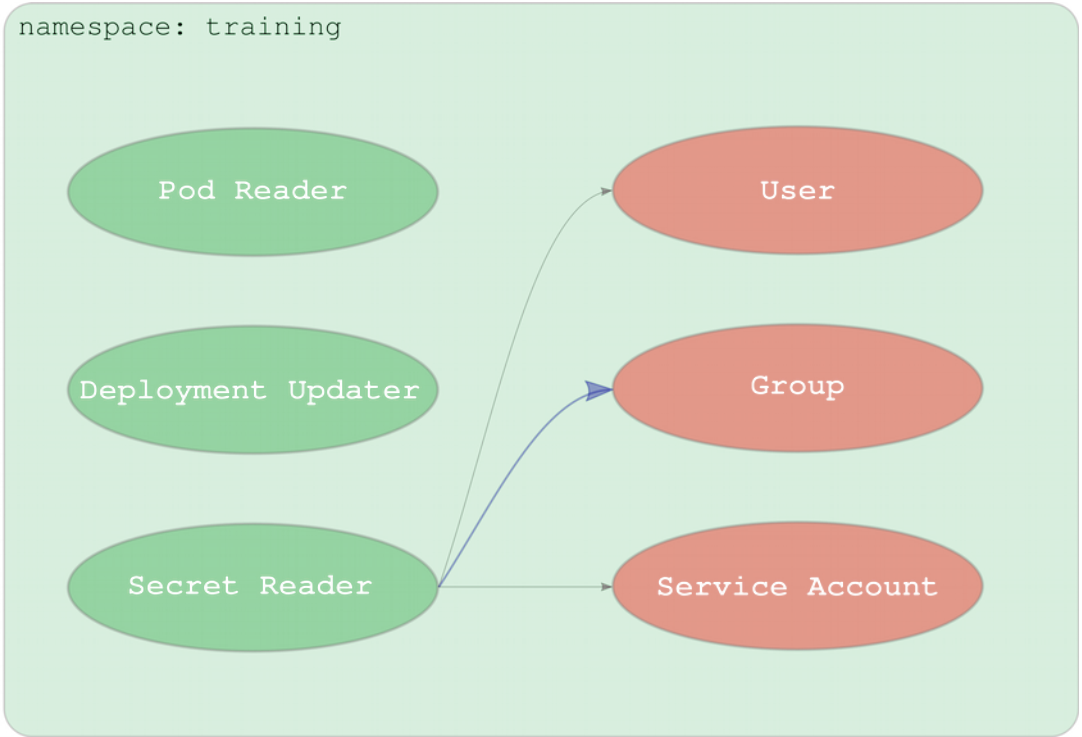
```
kind: ClusterRoleBinding
name: deploy-updater
subjects:
- kind: Group
  name: developers
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: deploy-updater-global
  apiGroup: rbac.authorization.k8s.io
```



RBAC – ClusterRoleBinding

```
kind: Role
name: pod-reader
namespace: training
  ...
rules:
- apiGroups: [""]
  resources: ["secrets"]
  verbs: ["get", "watch", "list"]
```

```
kind: RoleBinding
name: secret-reader
namespace: training
subjects:
- kind: Group
  name: managers
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: secret-reader-global
  apiGroup: rbac.authorization.k8s.io
```



end