Deterministic

Workloads

in

k8s

whoami

Allan Højgaard Jensen

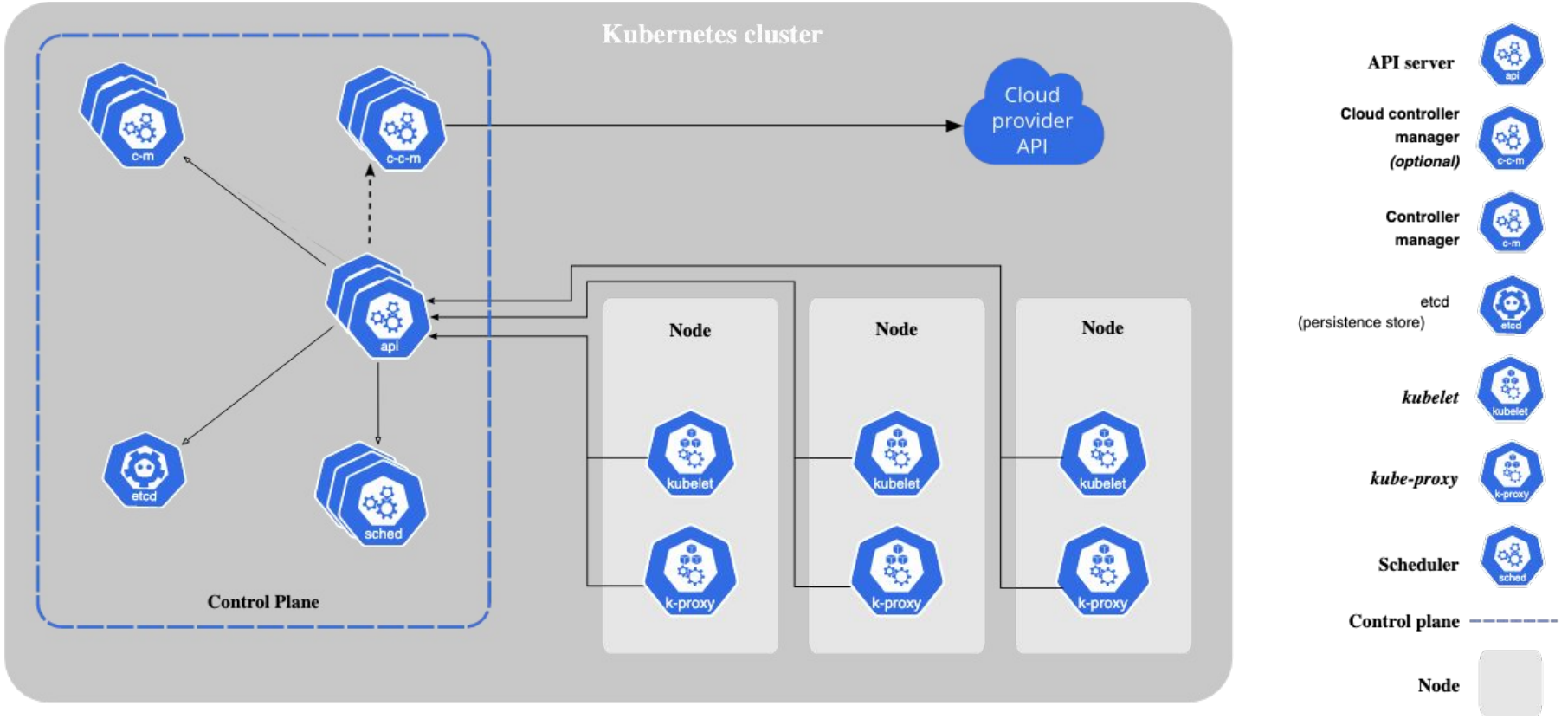
Platform Development

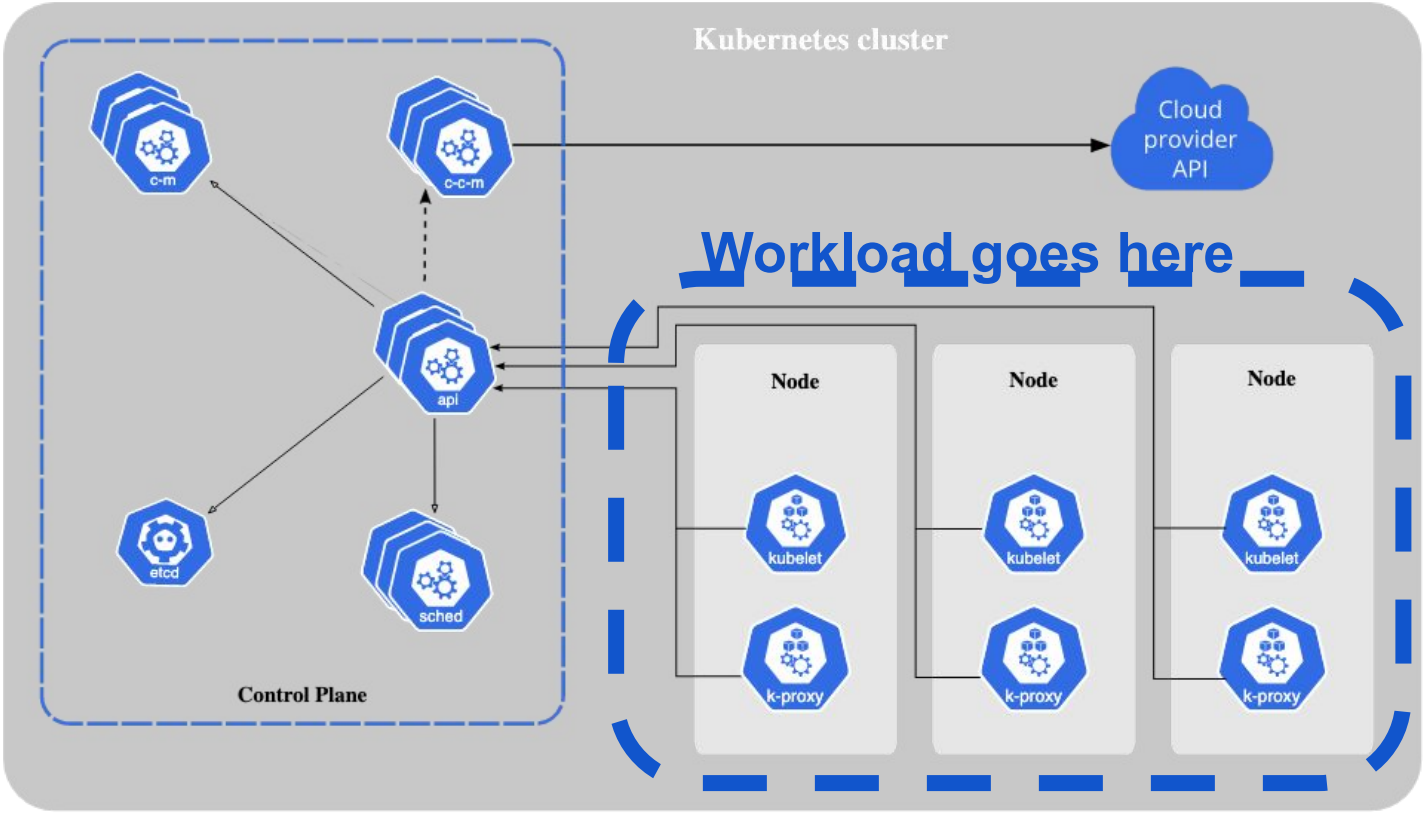
Working with all aspects of information technology, among that building a **Secure Cloud Stack** based on Cloud Native technologies. This exciting work is done together with a great group of people, who shares a particular passion for creating value for the customers and users of Cloud Native Applications and Services.

I founded Cloud Native Aalborg together with people from other companies in Aalborg to create a better knowledge about Cloud Native technologies in the Northern part of Denmark.

Presentation Style
SlideMotion’ish

a Printable version is located in github together with a workshop, where you can work on this if you find this interesting.


Kubernetes cluster
c-m
c-c-m
Cloud provider API
api
etcd
sched
Control Plane
Node
Node
Node
kubelet
kubelet
kubelet
k-proxy
k-proxy
k-proxy
API server
Cloud controller manager (optional)
Controller manager
etcd (persistence store)
kubelet
kube-proxy
Scheduler
Control plane
Node



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

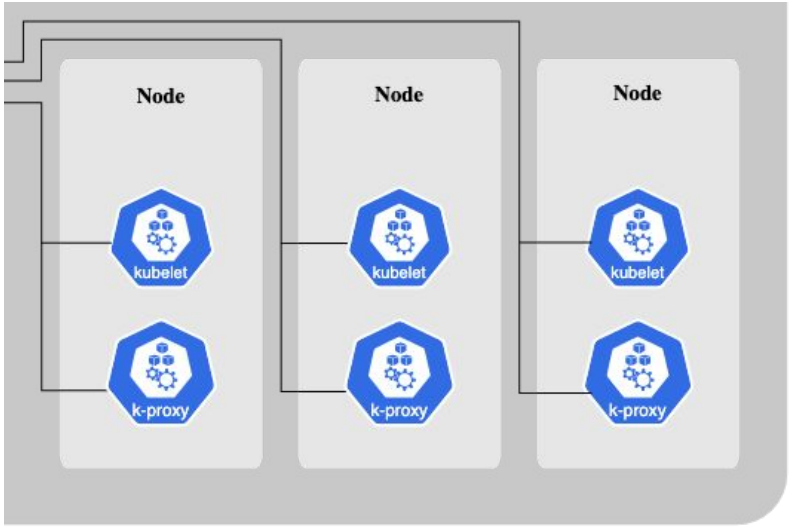
kubelet

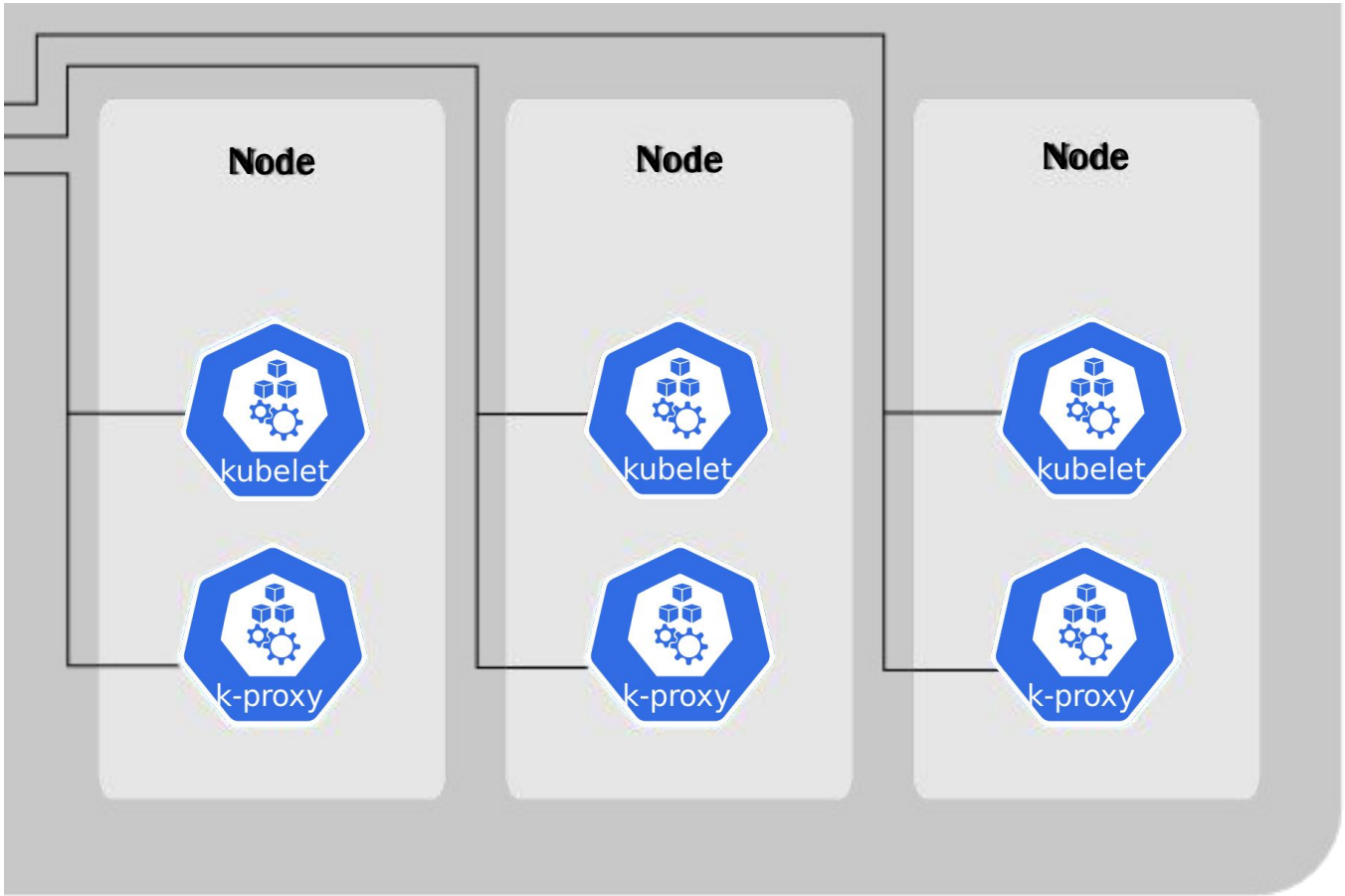
kube-proxy

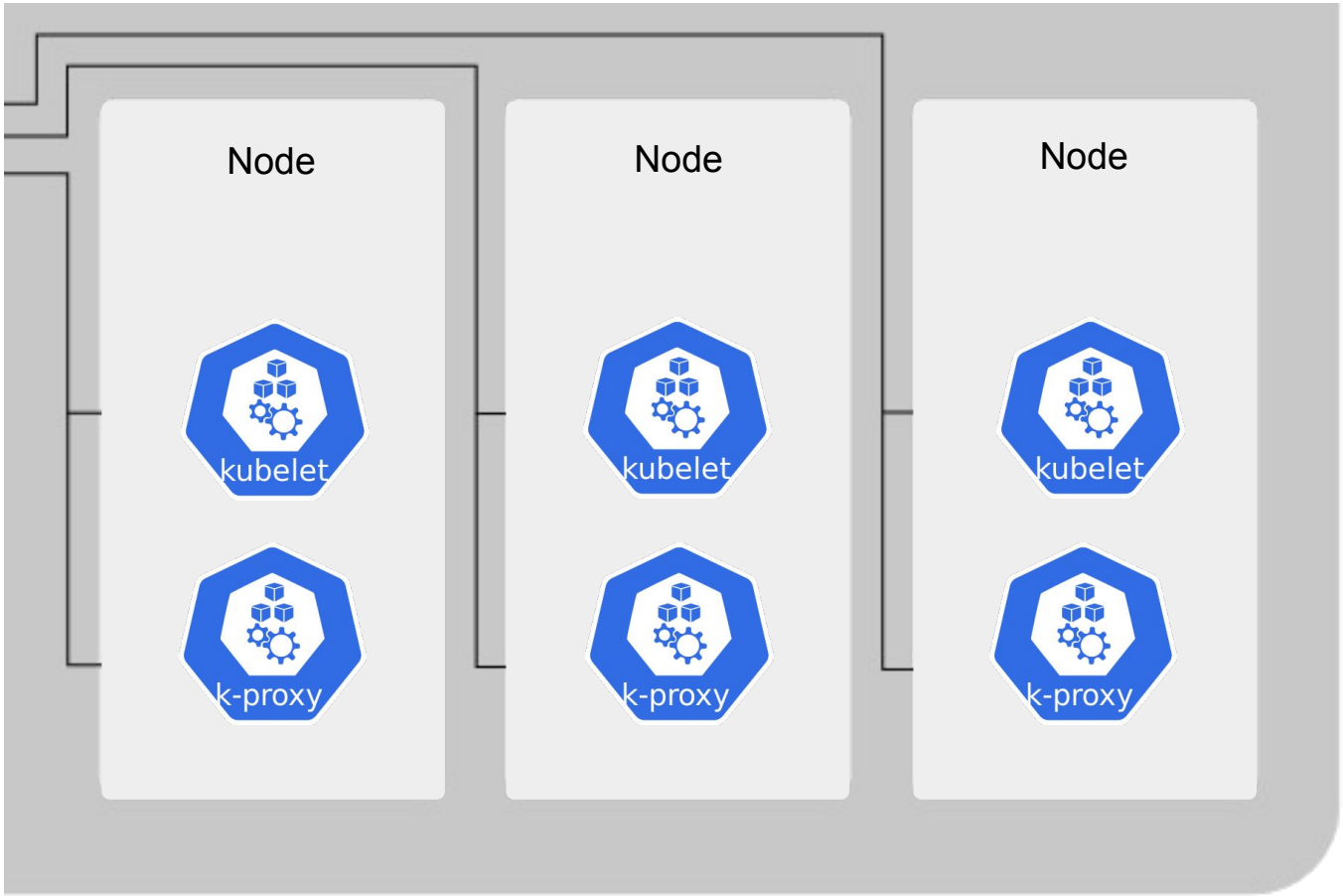
Scheduler

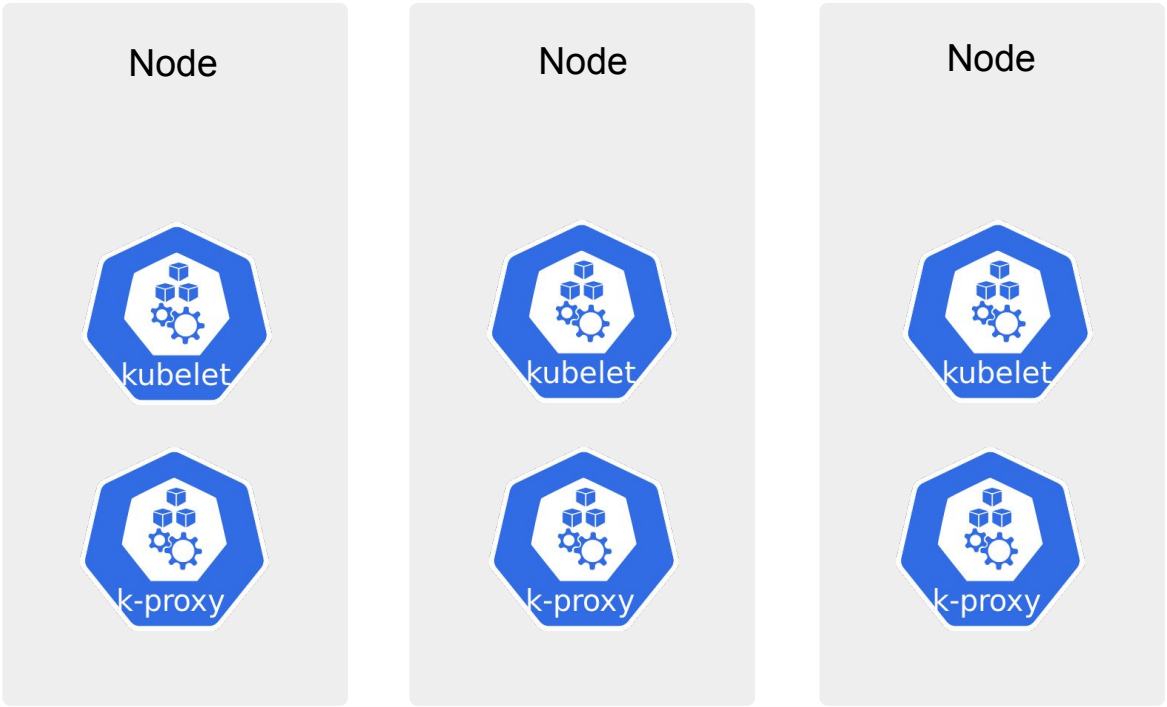
Control plane

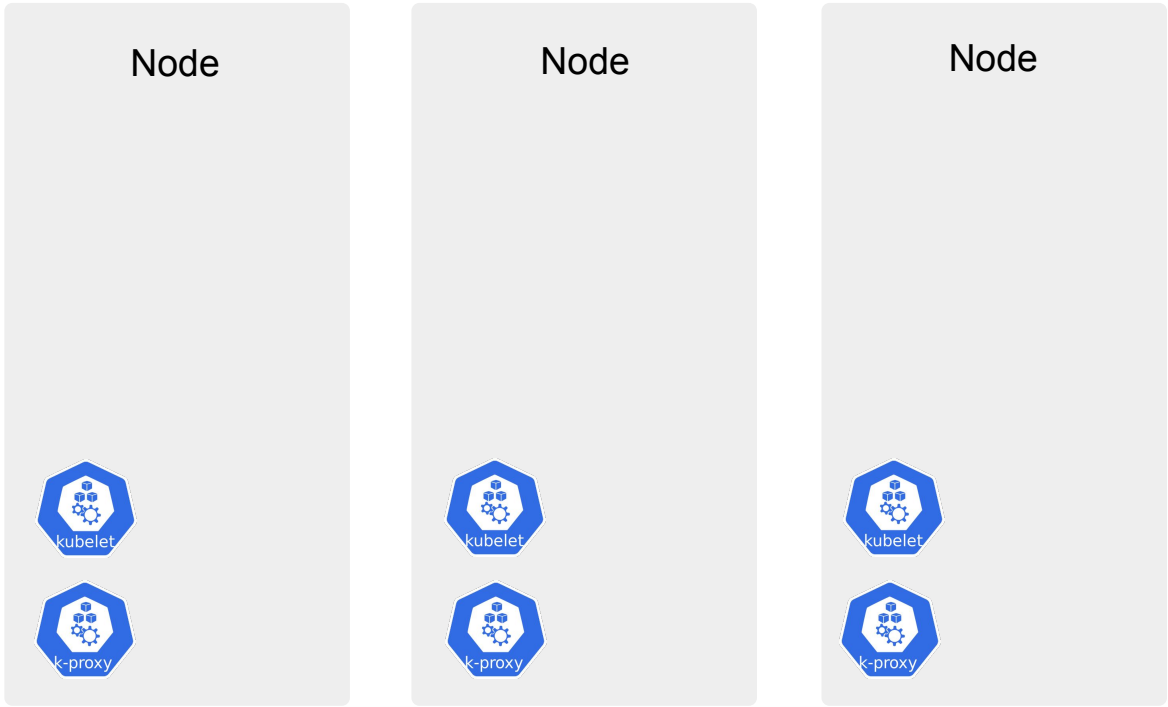
Node

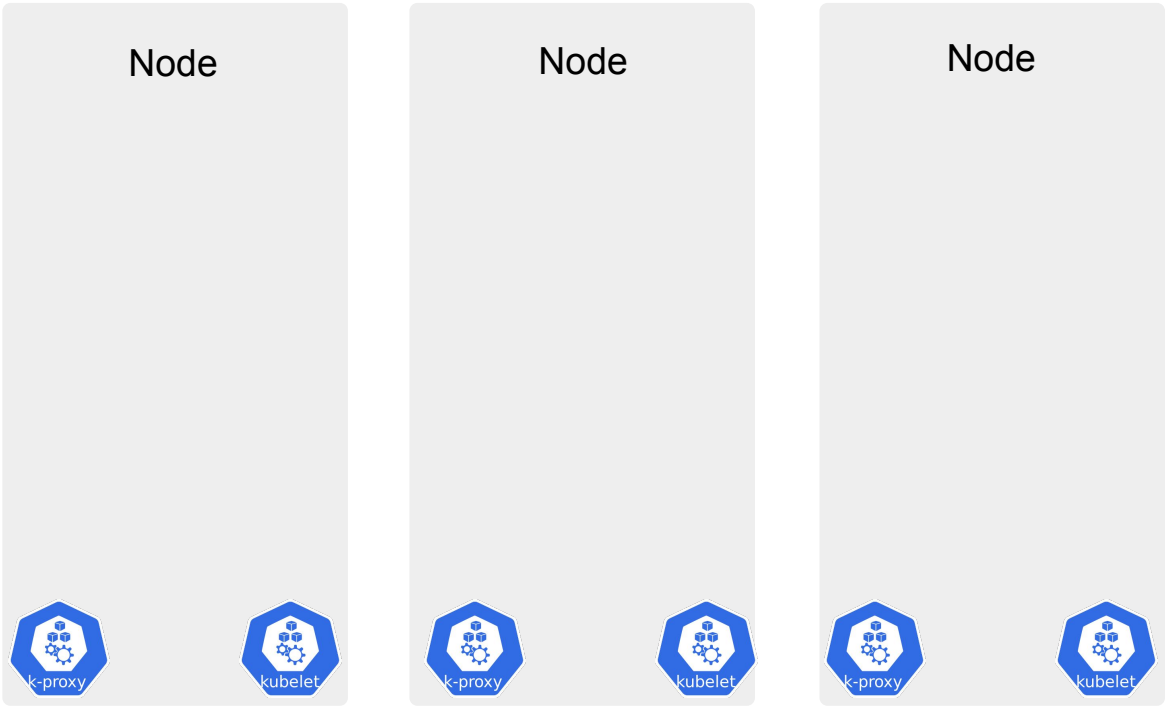








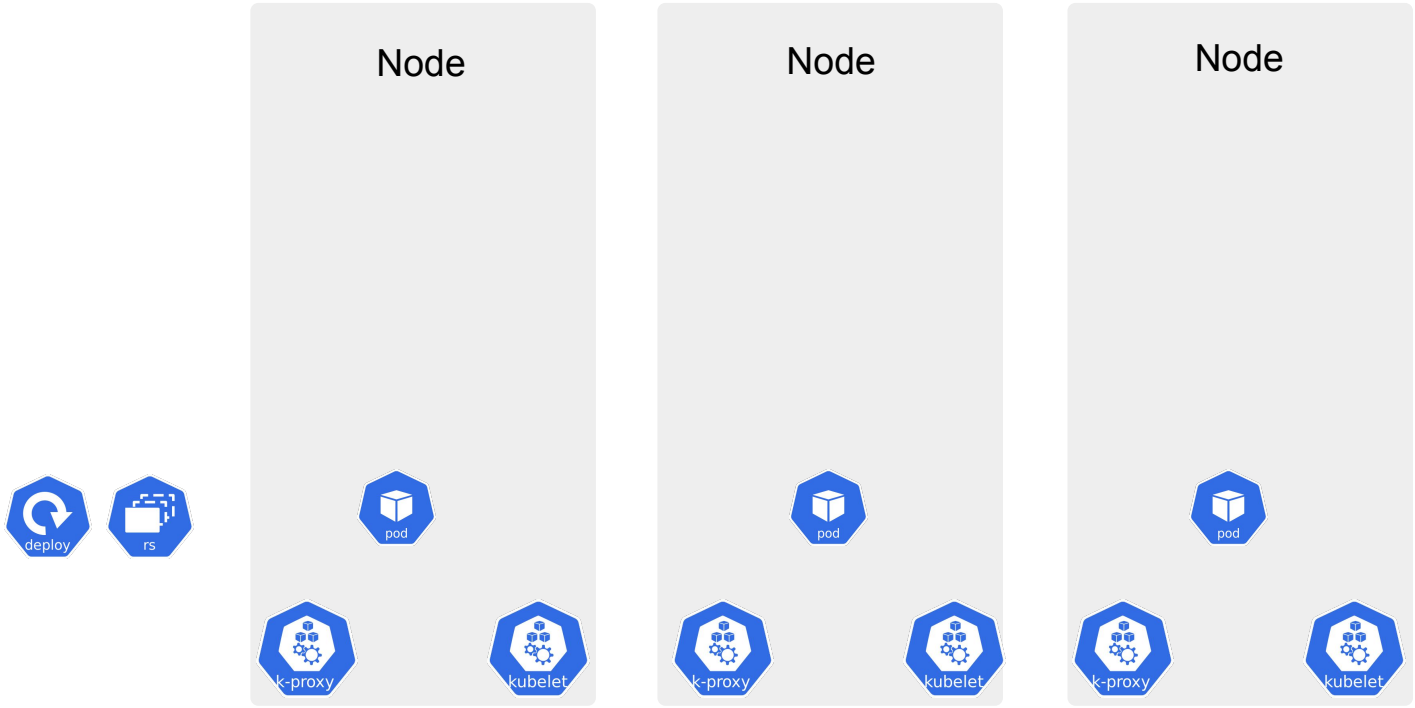


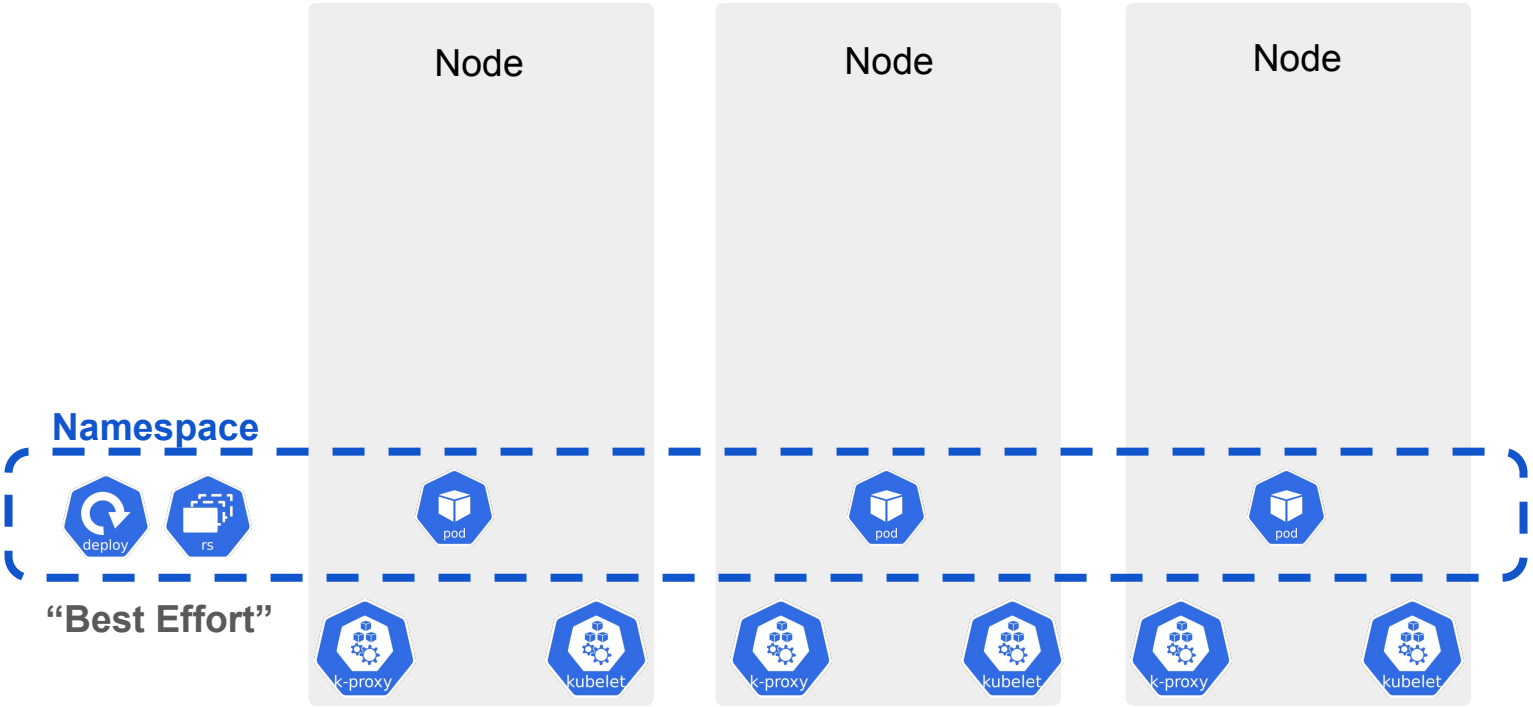


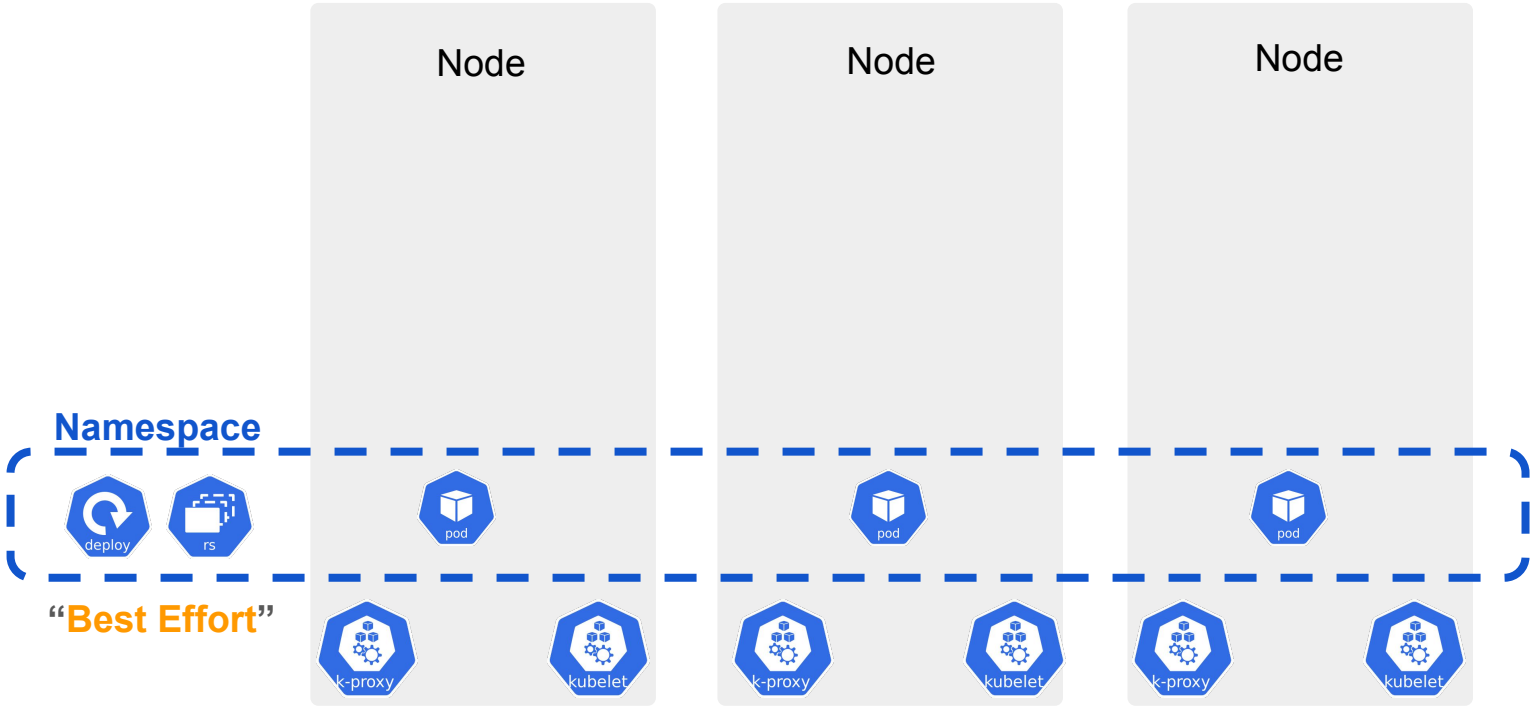
Node

Node

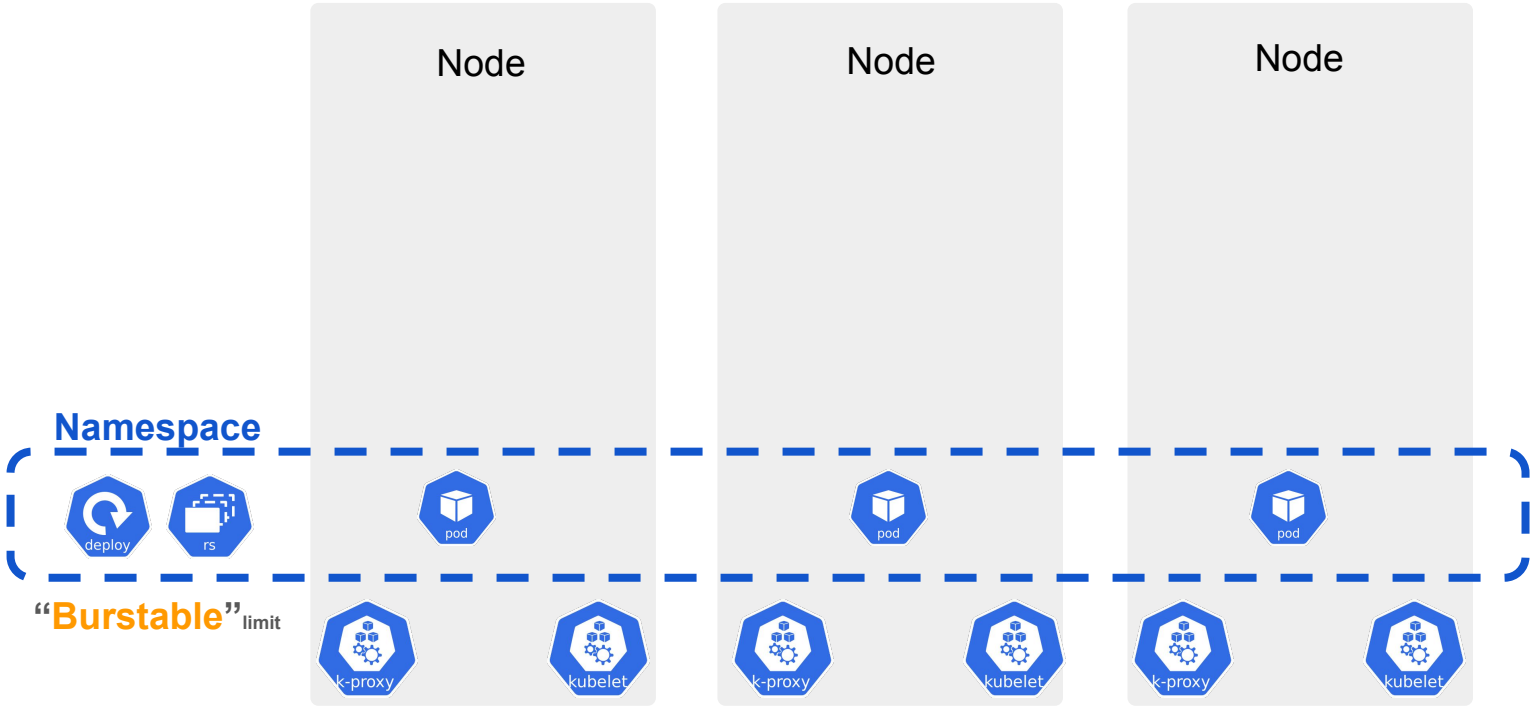
Node



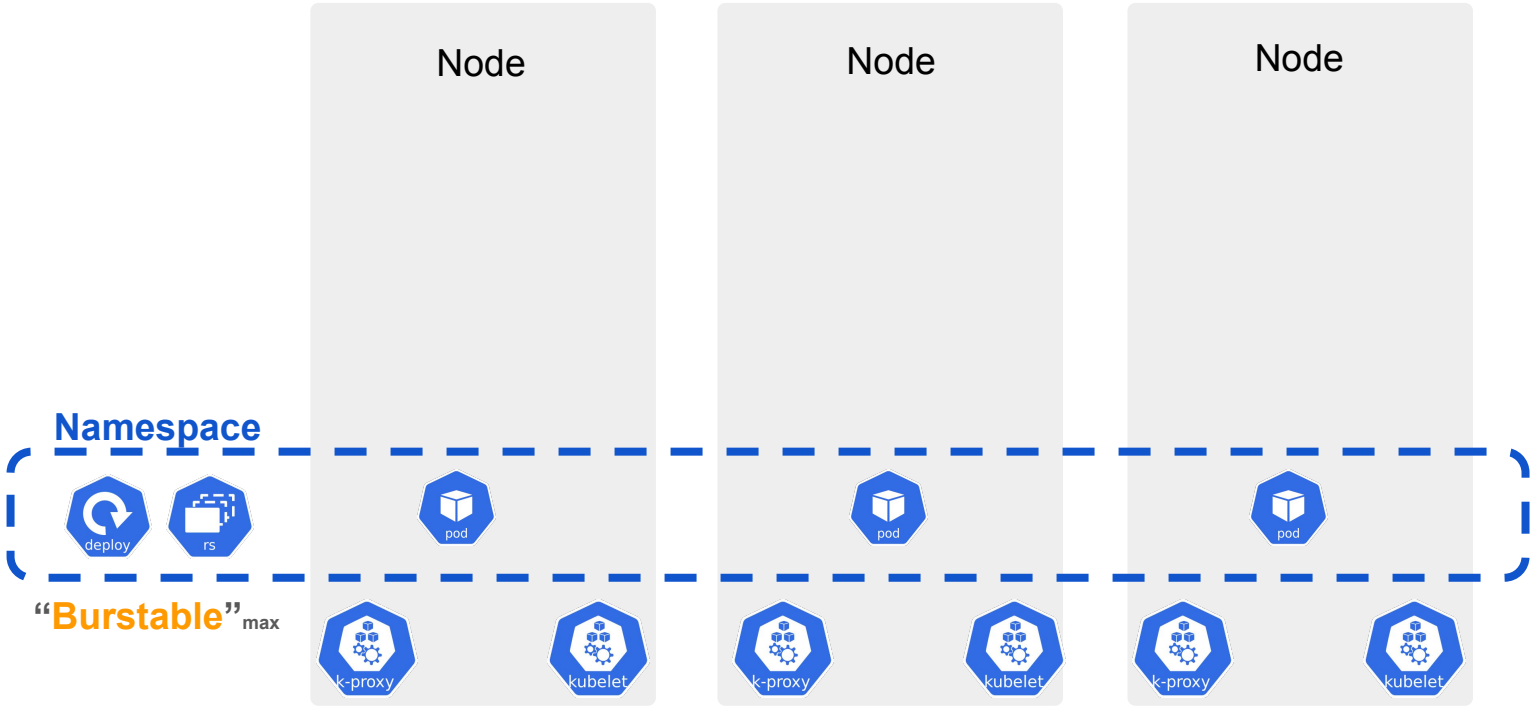




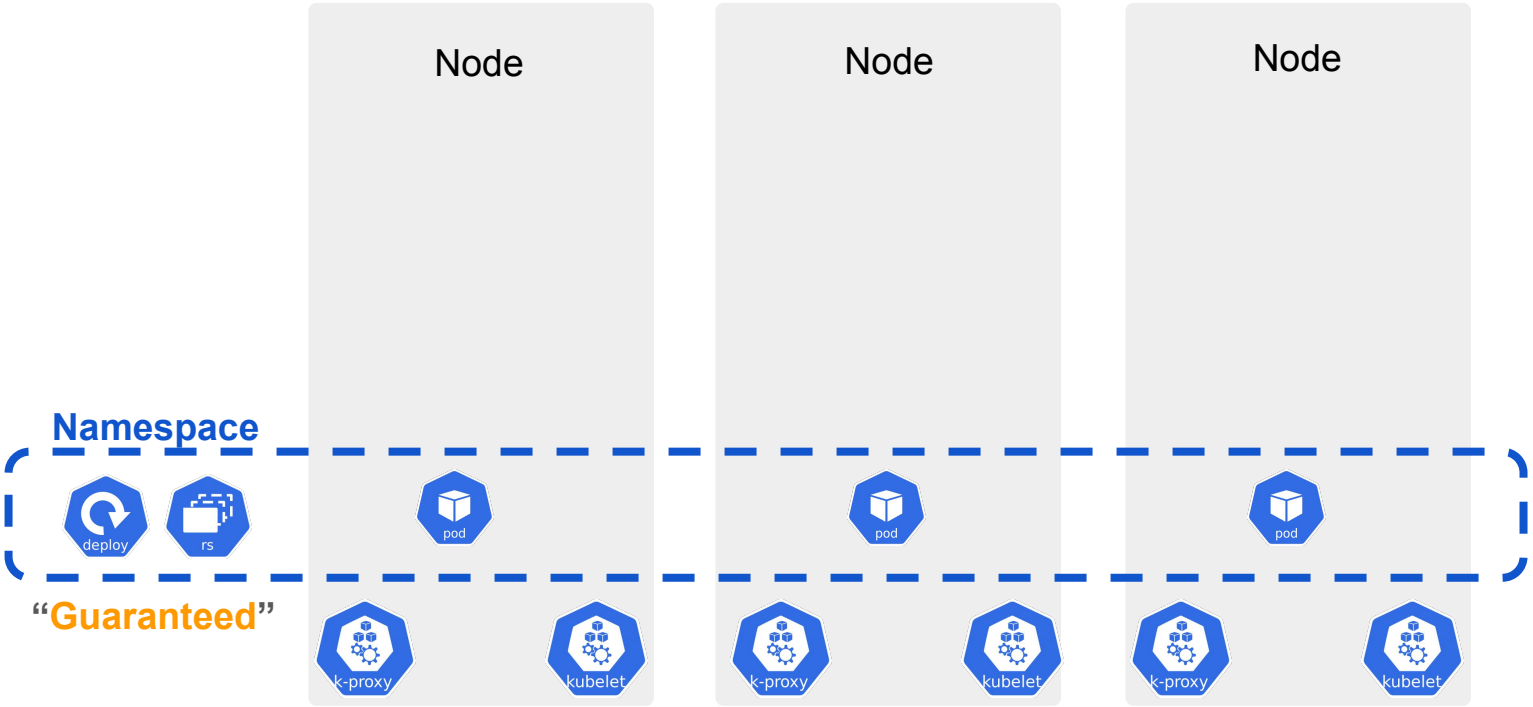

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2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



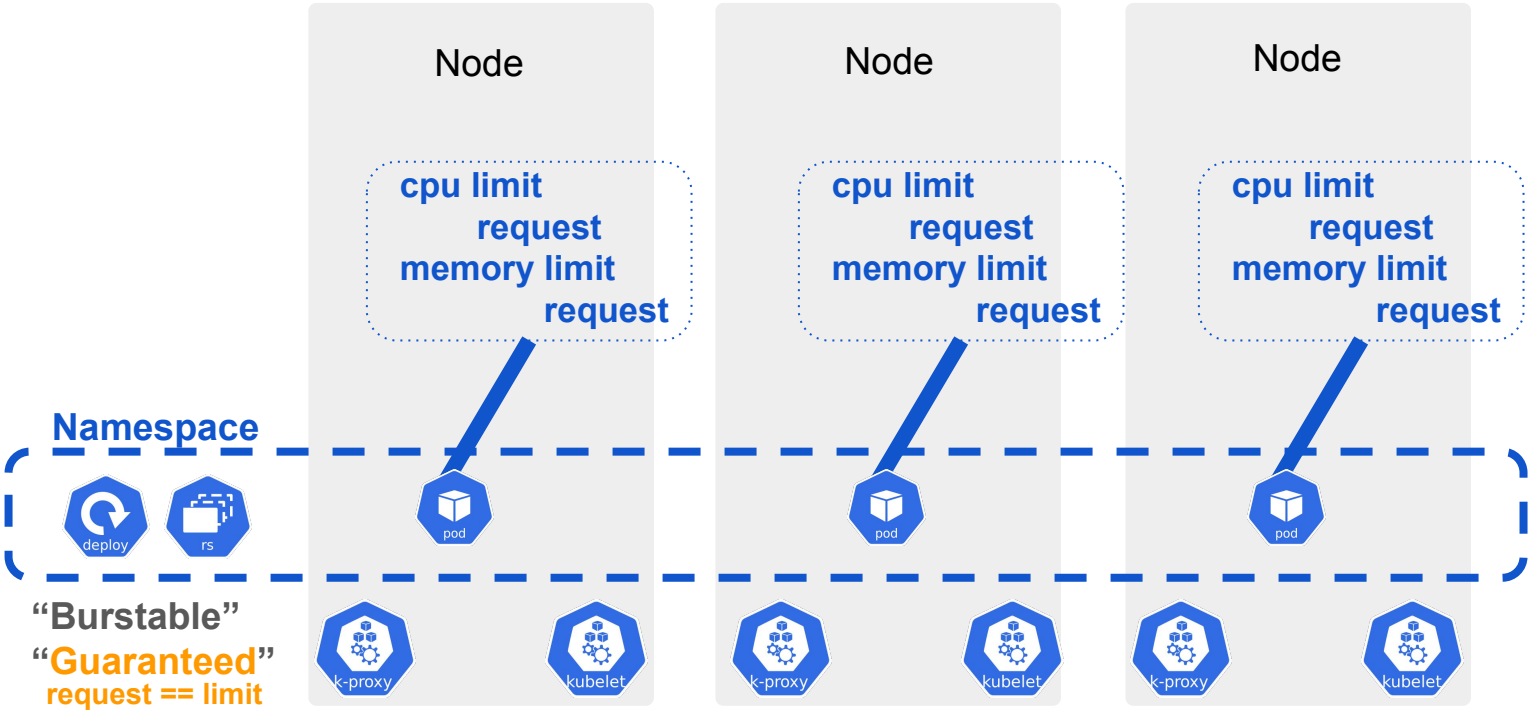

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "200Mi"
            cpu: "1.5"
          requests:
            memory: "200Mi"
            cpu: "1"
```

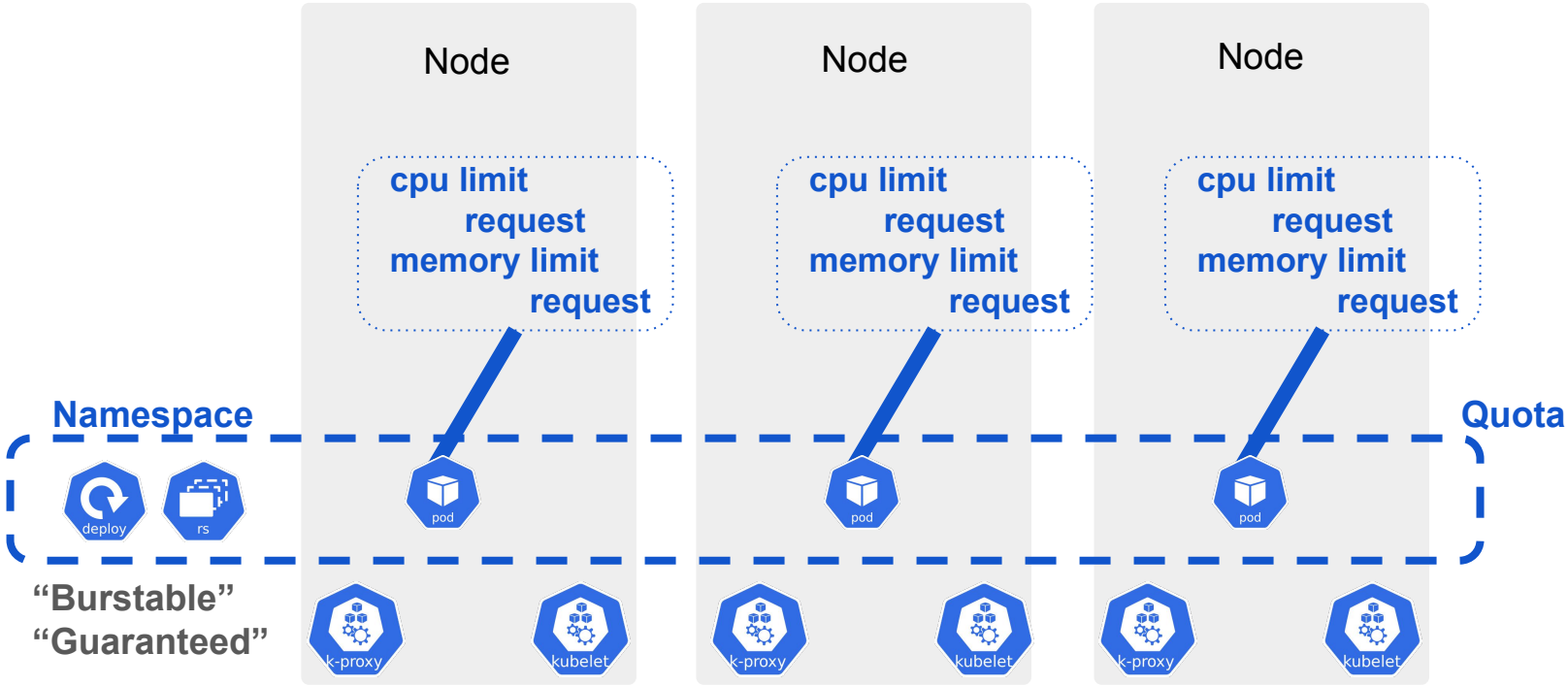


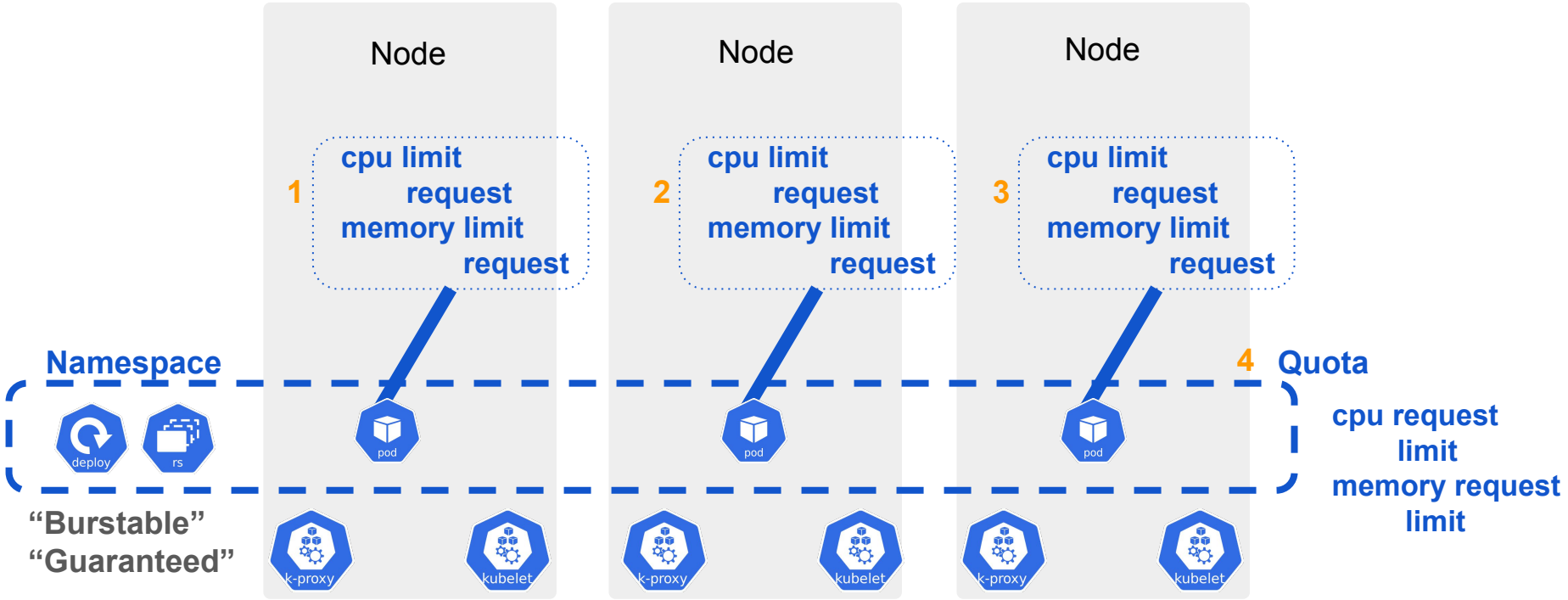

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "200Mi"
          requests:
            memory: "200Mi"
            cpu: "1"
```




```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "200Mi"
            cpu: "1.2"
          requests:
            memory: "200Mi"
            cpu: "1.2"
```

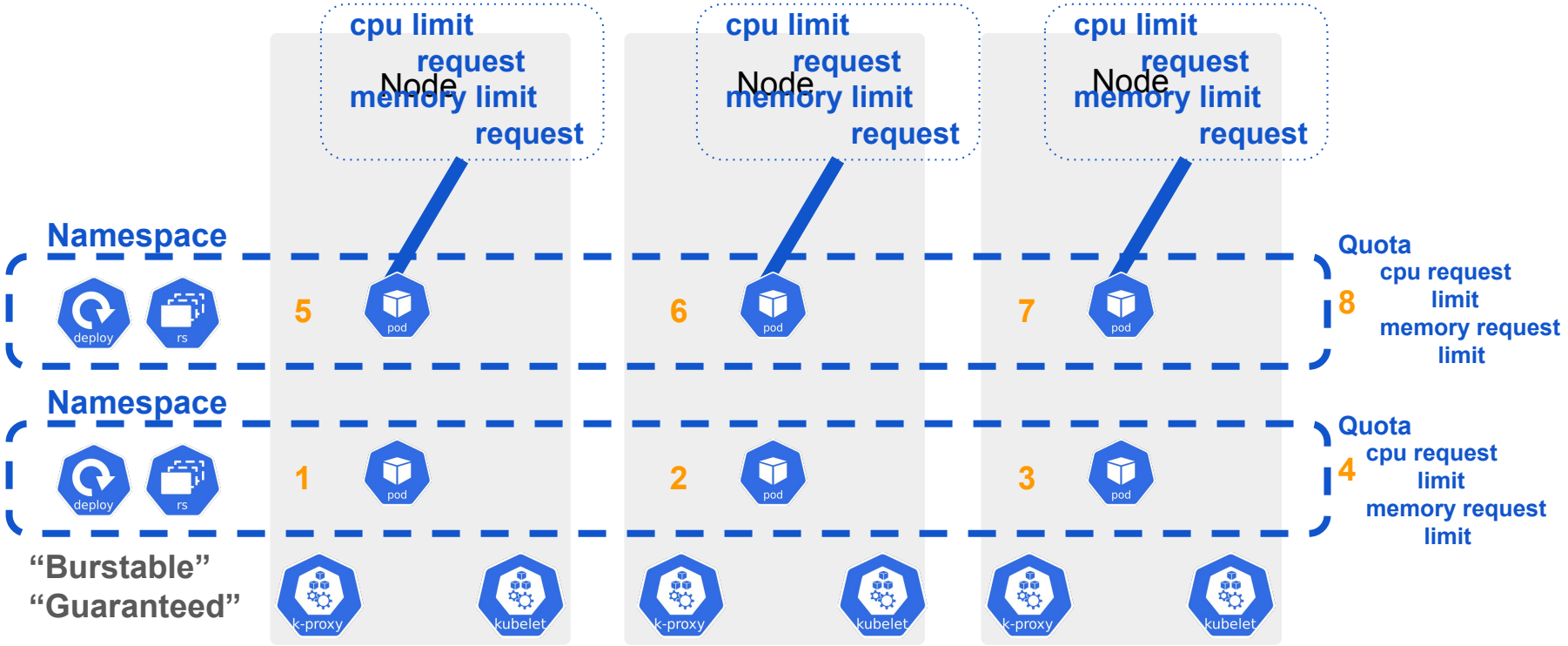

Node
Node
Node
cpu limit
request
memory limit
request
cpu limit
request
memory limit
request
cpu limit
request
memory limit
request
Namespace
deploy
rs
pod
pod
pod
k-proxy
kubelet
k-proxy
kubelet
k-proxy
kubelet
“Burstable”
“Guaranteed”
request == limit


Node
Node
Node
cpu limit
request
memory limit
request
cpu limit
request
memory limit
request
cpu limit
request
memory limit
request
Namespace
Quota
deploy
rs
pod
pod
pod
k-proxy
kubelet
k-proxy
kubelet
k-proxy
kubelet
“Burstable”
“Guaranteed”



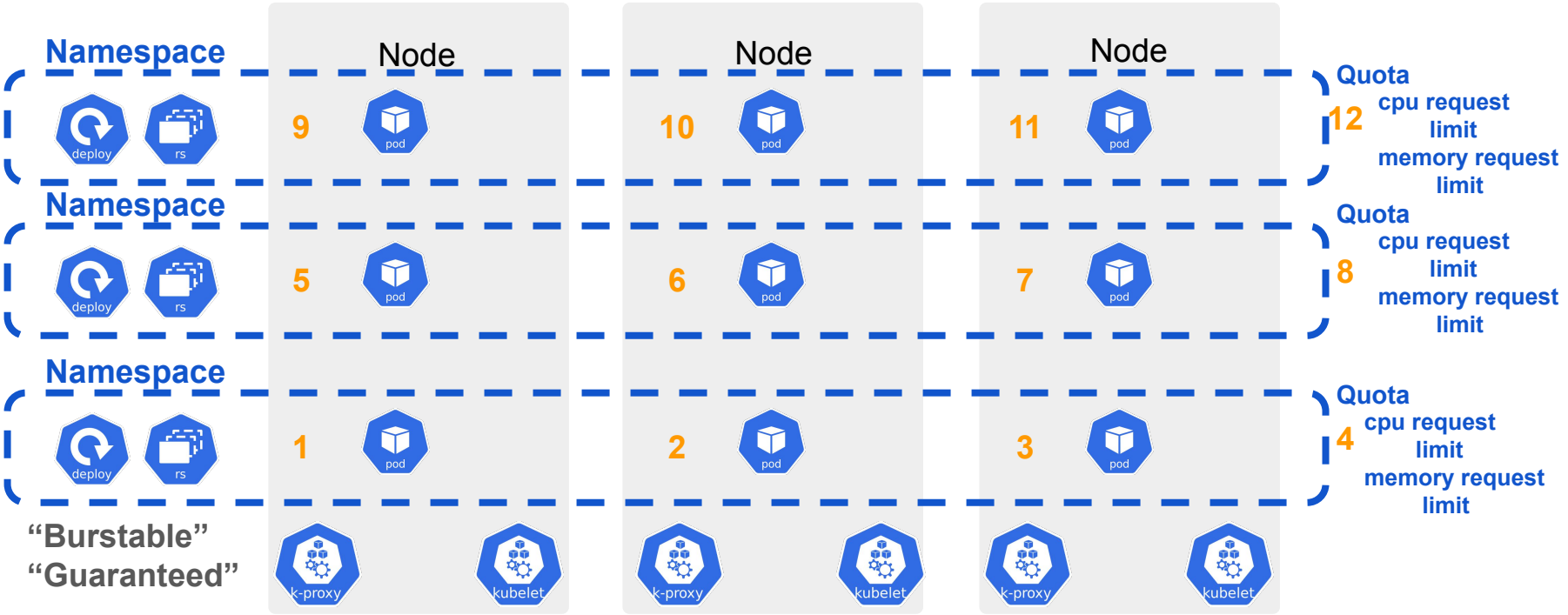
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

$1+2+3 < 4$



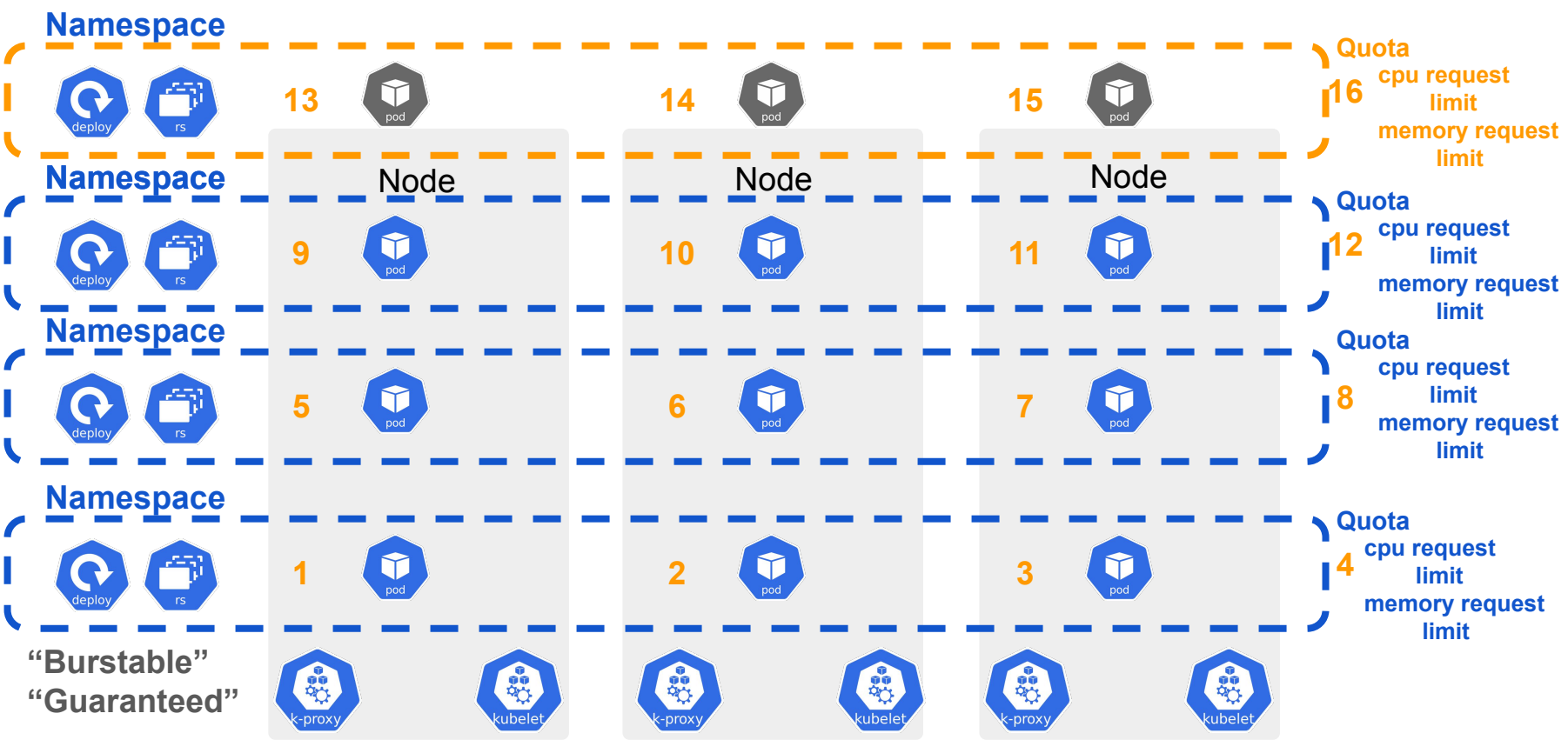
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

1+2+3 < 4 and 5+6+7 < 8



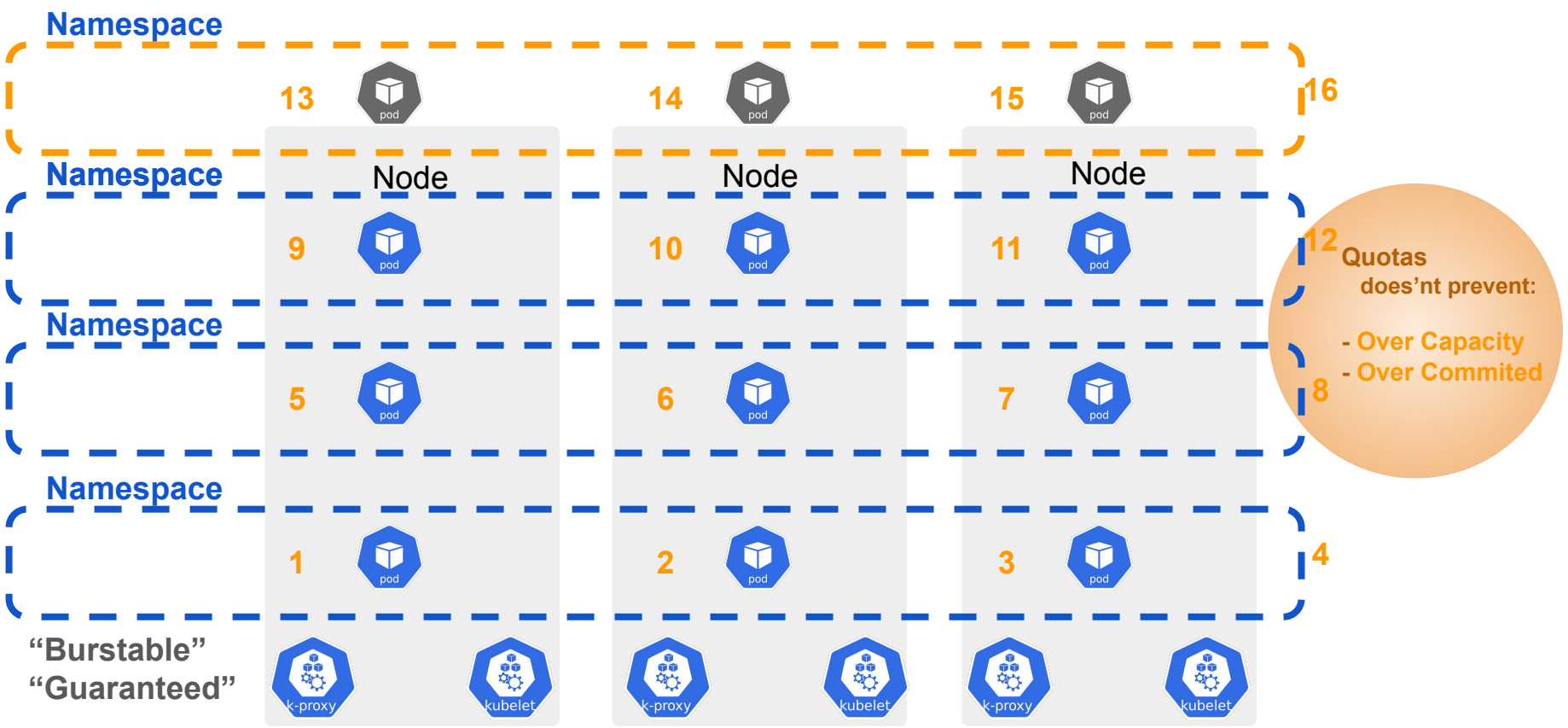
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

$1+2+3 < 4$ and $5+6+7 < 8$ and $9+10+11 < 12$



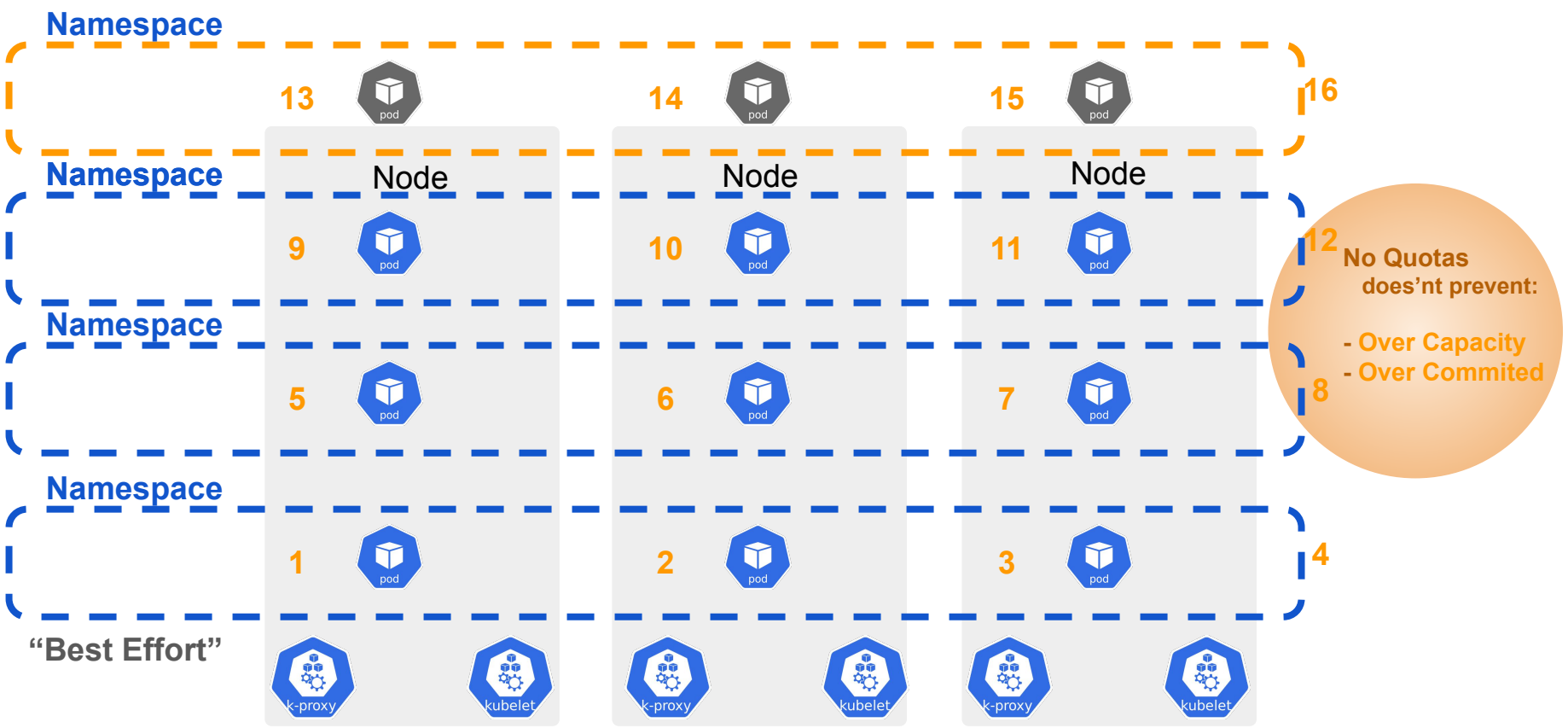
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

(1+2+3 < 4 and 5+6+7 < 8 and 9+10+11 < 12 and 13+14+15 < 16)

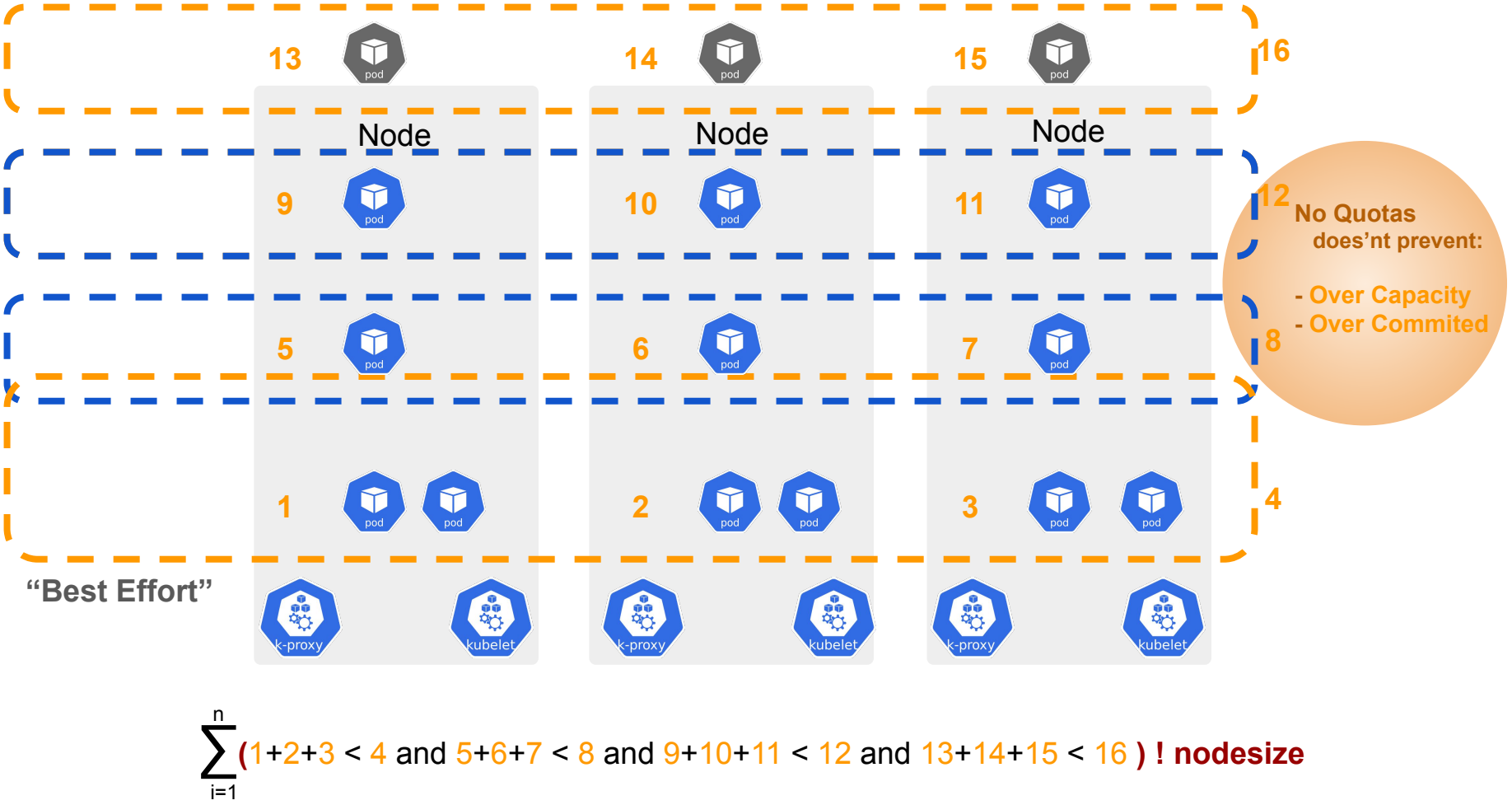


$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

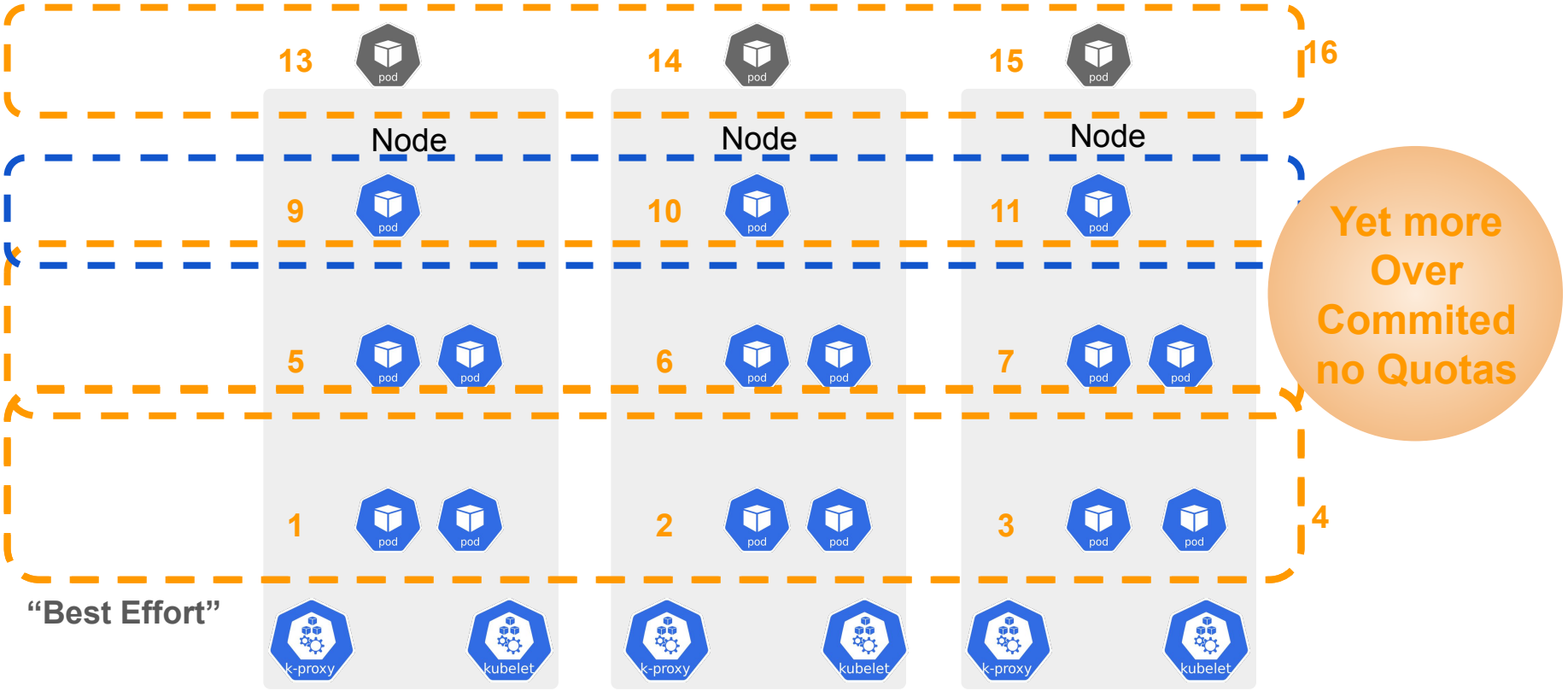
(1+2+3 < 4 and 5+6+7 < 8 and 9+10+11 < 12 and 13+14+15 < 16) ! **nodesize**



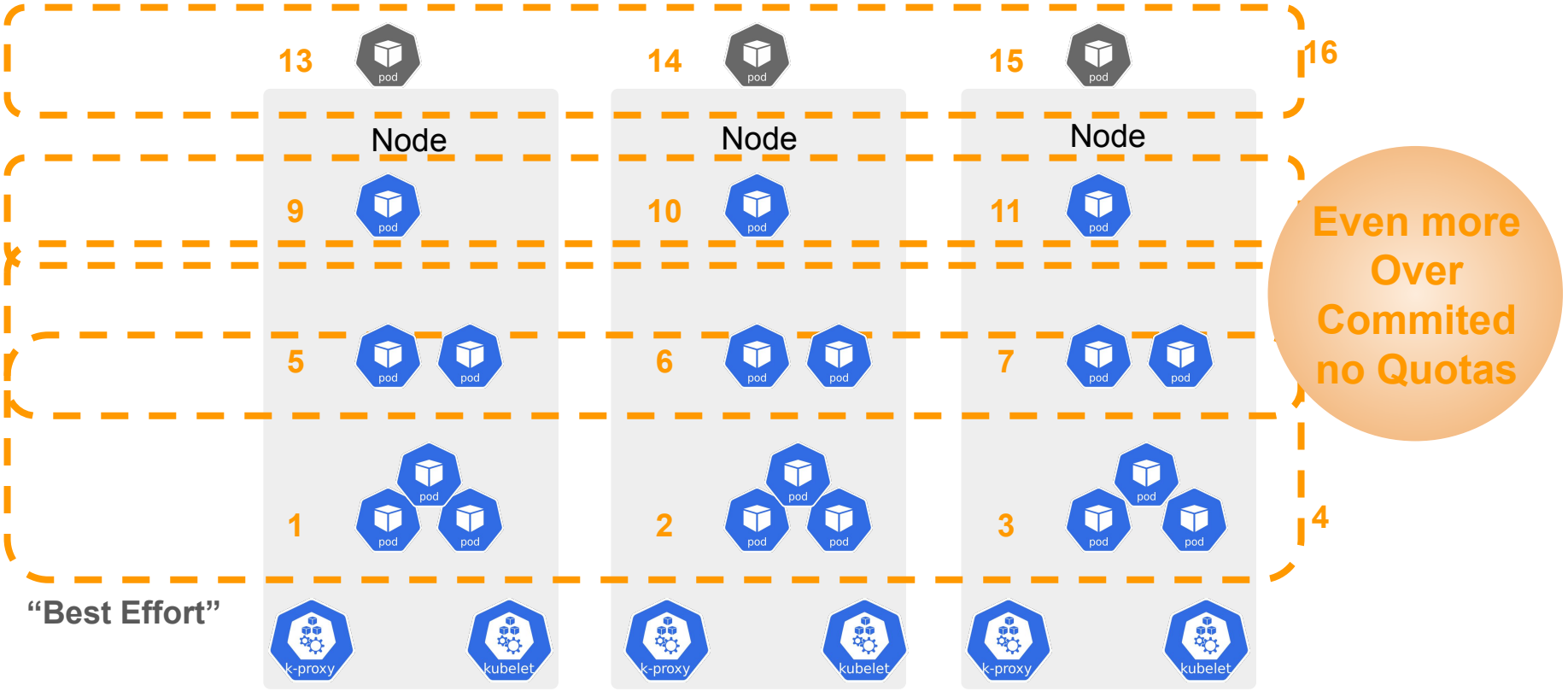
no quotas $\sum_{i=1}^{n} Pod_{iSize}$ (1+2+3 < 4 and 5+6+7 < 8 and 9+10+11 < 12 and 13+14+15 < 16) ! nodesize



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 9+10+11 < 12 \text{ and } 13+14+15 < 16) \text{ ! nodesize}$$



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 9+10+11 < 12 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 9+10+11 < 12 \text{ and } 13+14+15 < 16) \text{ ! nodesize}$$

What are the options?

- Scale Out + nodes across AZ's
  ⇒ Increased Spending - typically "Best Effort", "Bustable" (!FinOps'y)

- Introduce cross namespace scheduling management
  ⇒ Limiting Scale using "Guaranteed, Burstable" Class pods (FinOps'y)

What are the options?

- Scale Out + nodes across AZ's
  ⇒ Increased Spending - typically "Best Effort", "Bustable" (!FinOps'y)

- Introduce cross namespace scheduling management
  ⇒ Limiting Scale using "Guaranteed, Burstable" Class pods (FinOps'y)

- Let kubernetes do a "Best Effort" based on its algorithms

What are the options?

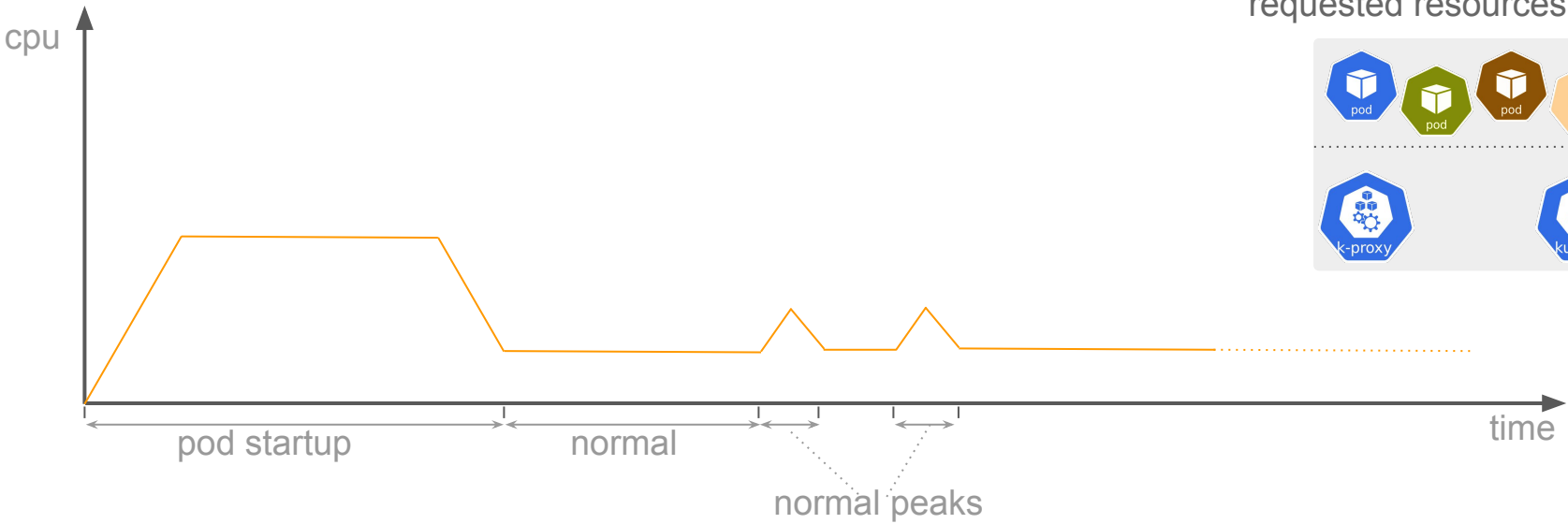
- Scale Out + nodes across AZ's
  ⇒ Increased Spending - typically "Best Effort", "Bustable" (!FinOps'y)

- Introduce cross namespace scheduling management
  ⇒ Limiting Scale using "Guaranteed, Burstable" Class pods (FinOps'y)

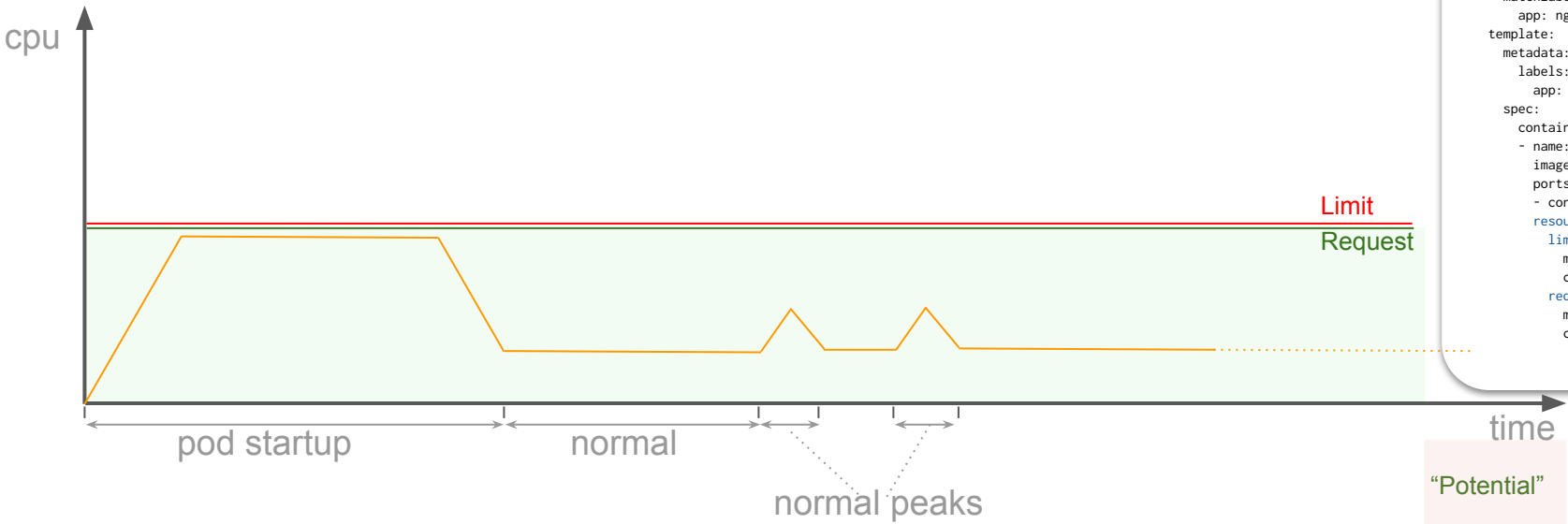
- Let kubernetes do a "Best Effort" based on its algorithms

- **Let kubernetes do a "Prioritised Effort" & increase workloads utilisation**
  ⇒ Prioritised Scale within Budget

Pod Quality of Service Classes



Pod Quality of Service Classes - Guaranteed



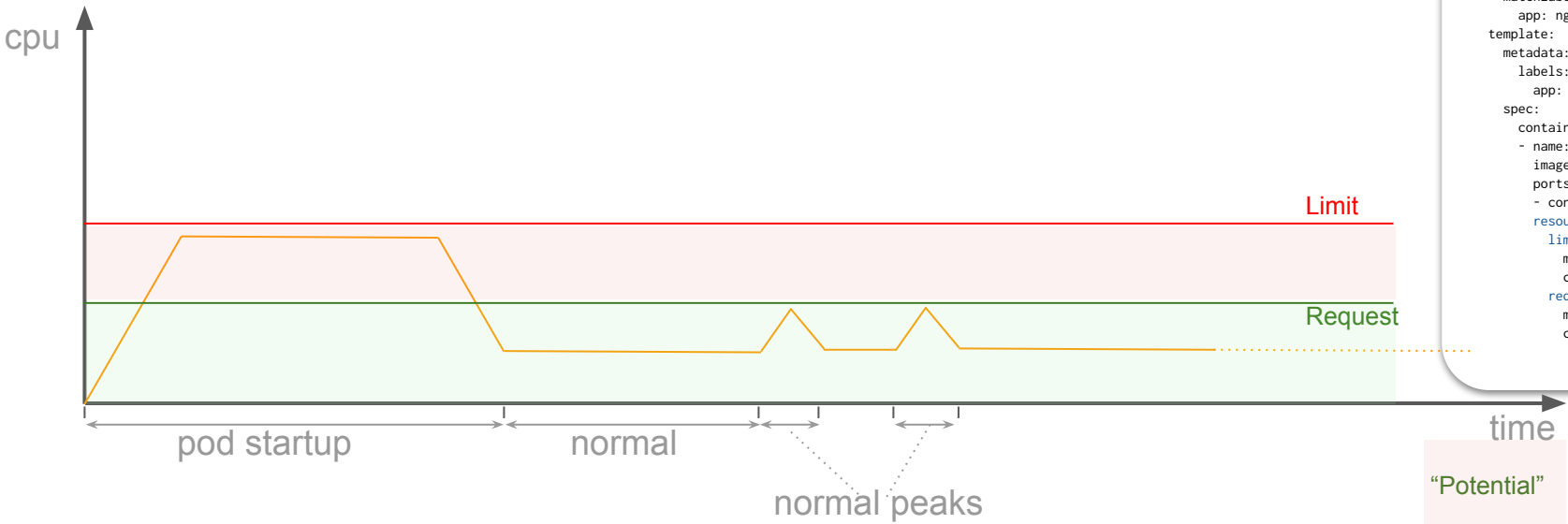

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "200Mi"
            cpu: "1.2"
          requests:
            memory: "200Mi"
            cpu: "1.2"
```

"Potential"

"Promise"

Limit == Request

Pod Quality of Service Classes - Burstable I



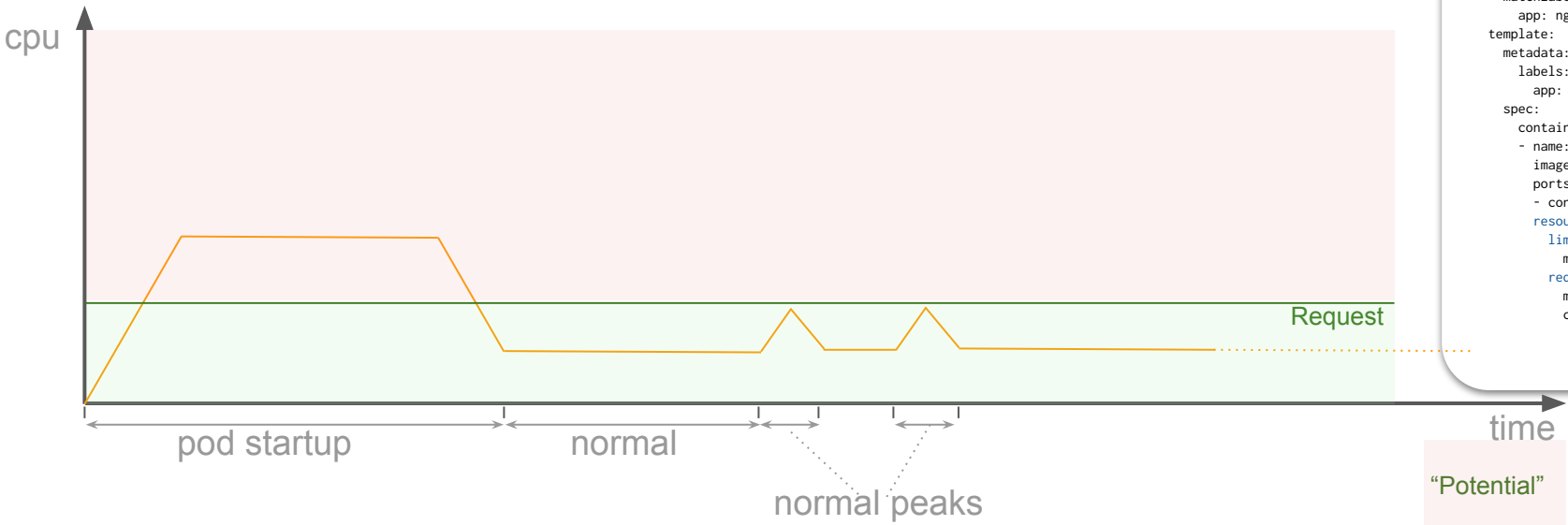

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "200Mi"
            cpu: "1.2"
          requests:
            memory: "200Mi"
            cpu: "0.6"
```

"Potential"

"Promise"

Limit > Request

Pod Quality of Service Classes - Burstable II



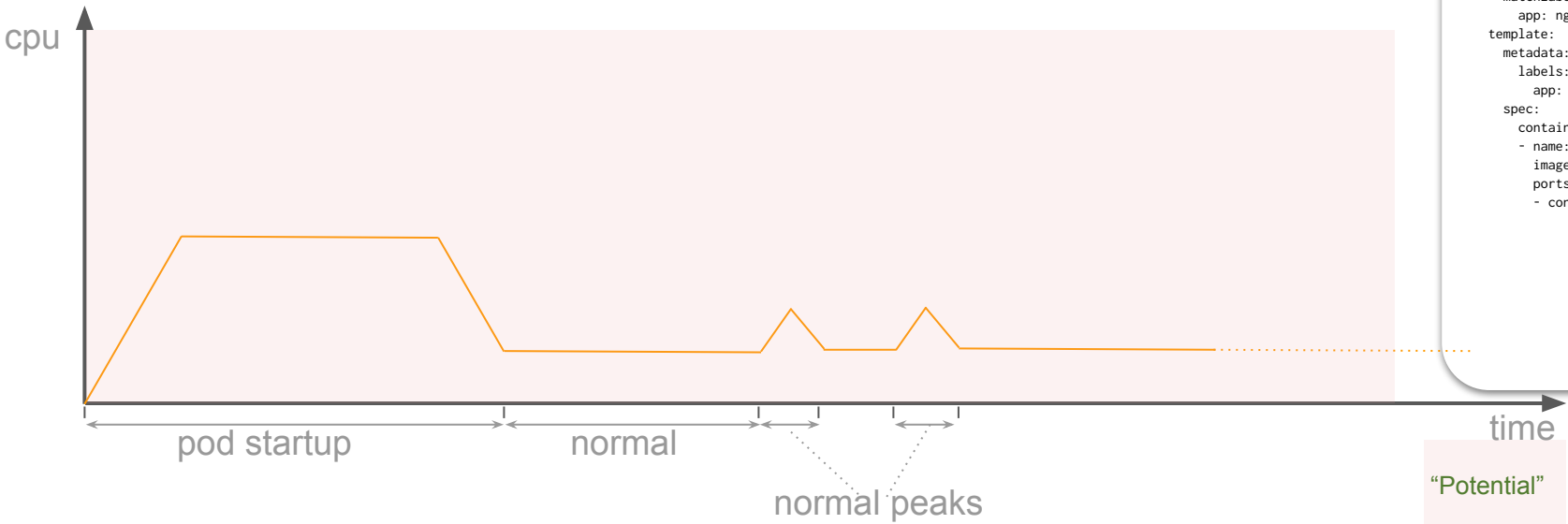

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "200Mi"
          requests:
            memory: "200Mi"
            cpu: "0.6"
```

"Potential"

"Promise"

noLimit Request

Pod Quality of Service Classes - Best Effort




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

“Potential”

“Promise”

noLimit noRequest

You cannot inflate the physic capacity

image generated at : picsart.com

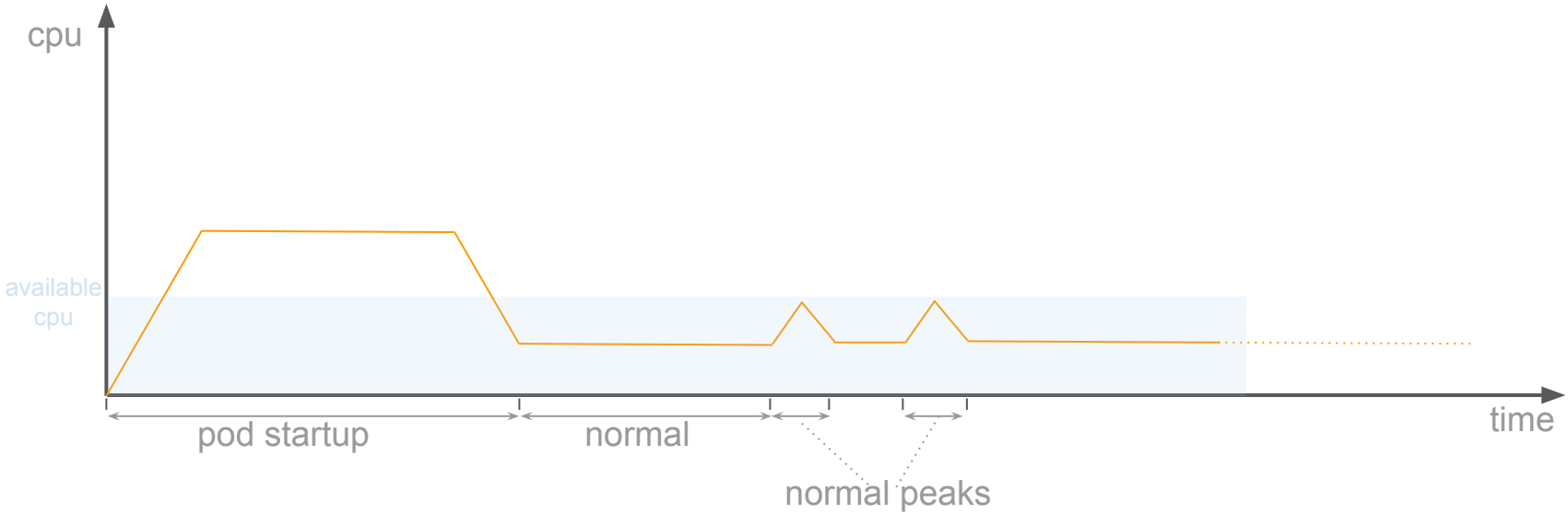
You have the **memory** available - that exists
- not more

You have the **cores** available - that exists
- not more

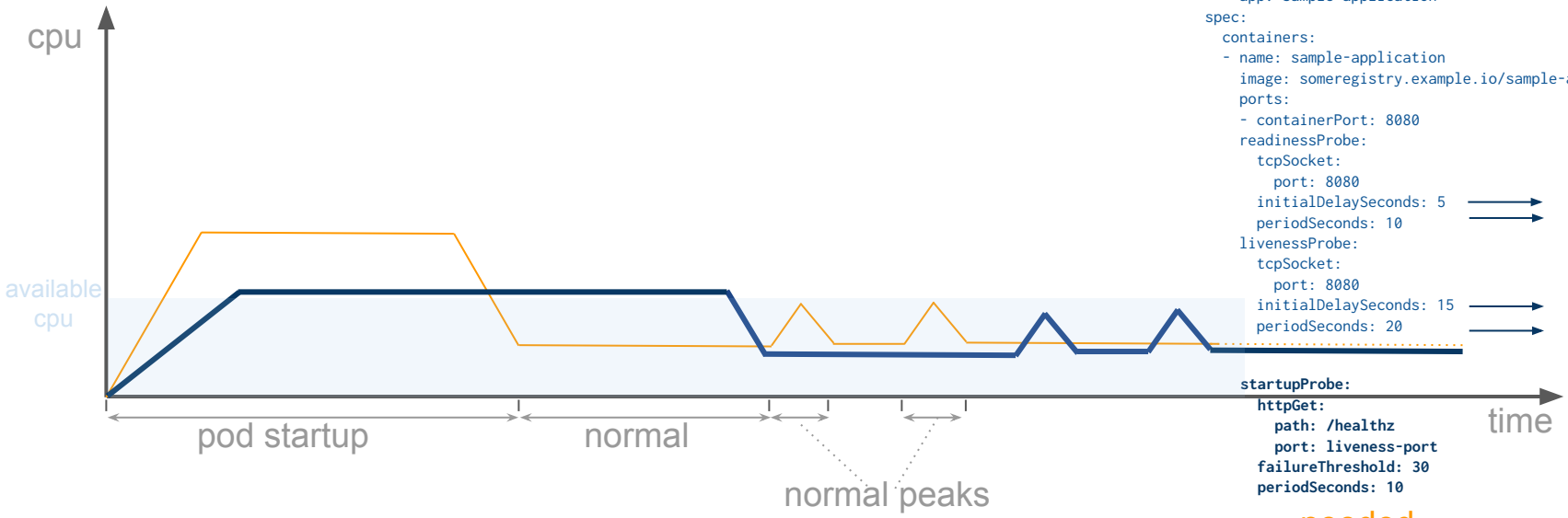
however you may use “what you have”
for different things at different times.

- and you may experience throttling cpu if not enough at given time
- and going forward you may experience memory swapping

Pod Quality of Service Classes



Pod Quality of Service Classes




```yaml
apiVersion: v1
kind: Pod
metadata:
  name: sample-application
  labels:
    app: sample-application
spec:
  containers:
  - name: sample-application
    image: someregistry.example.io/sample-application:1.0
    ports:
    - containerPort: 8080
    readinessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 5      → 10
      periodSeconds: 10           → 20
    livenessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 15     → 30
      periodSeconds: 20           → 40
    startupProbe:
      httpGet:
        path: /healthz
        port: liveness-port
      failureThreshold: 30
      periodSeconds: 10
```

needed

got

Capacity Balance vs Work Life Balance

You probably do NOT want to do this:

- for every workload
- whilst trying to imagine their interdynamics

Capacity Balance vs Work Life Balance

You probably do NOT want to do this:

- for every workload
- whilst trying to imagine their interdynamics

Unless you have a workload that is static, e.g:

- non changing
- non dynamic

iconscout.com

Capacity Balance vs Work Life Balance

You probably do NOT want to do this:

- for every workload
- whilst trying to imagine their interdynamics

Unless you have a workload that is static, e.g:

- non changing
- non dynamic

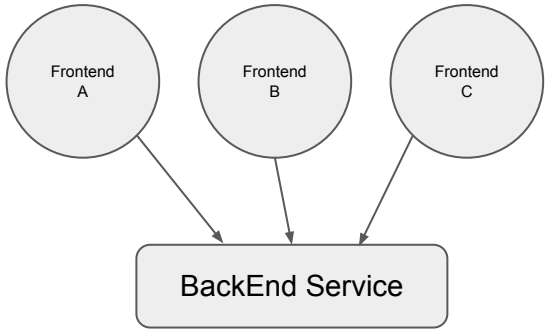
Instead you want to get the most out of Kubernetes ability to make the right decisions based on information about what are the important workloads.

Get the most of Kubernetes

Instead you want to get the most out of Kubernetes ability to make the right decisions based on information about what are the important workloads.

Thus if you have a backend service that a bunch of frontend stuff is dependent on, this would probably be more important than any of the frontend workloads.

However some of the frontend workloads may be more important than others

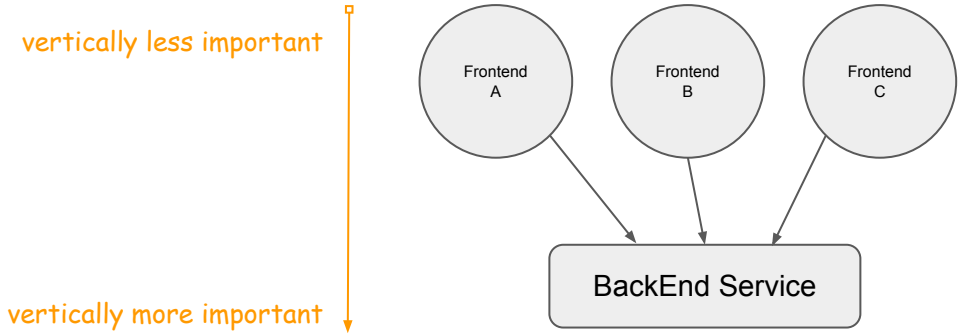


Get the most of Kubernetes

Instead you want to get the most out of Kubernetes ability to make the right decisions based on information about what are the important workloads.

Thus if you have a backend service that a bunch of frontend stuff is dependent on, this would probably be more important than any of the frontend workloads.

However some of the frontend workloads may be more important than others

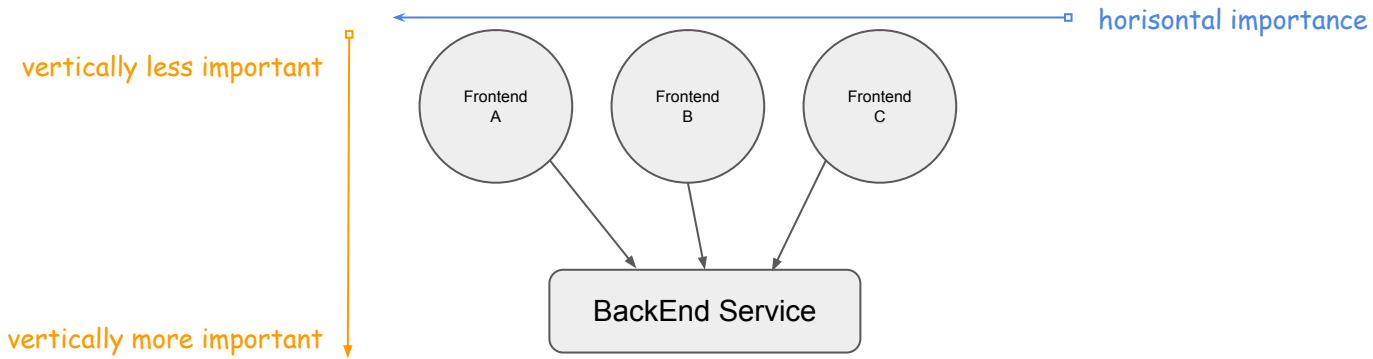


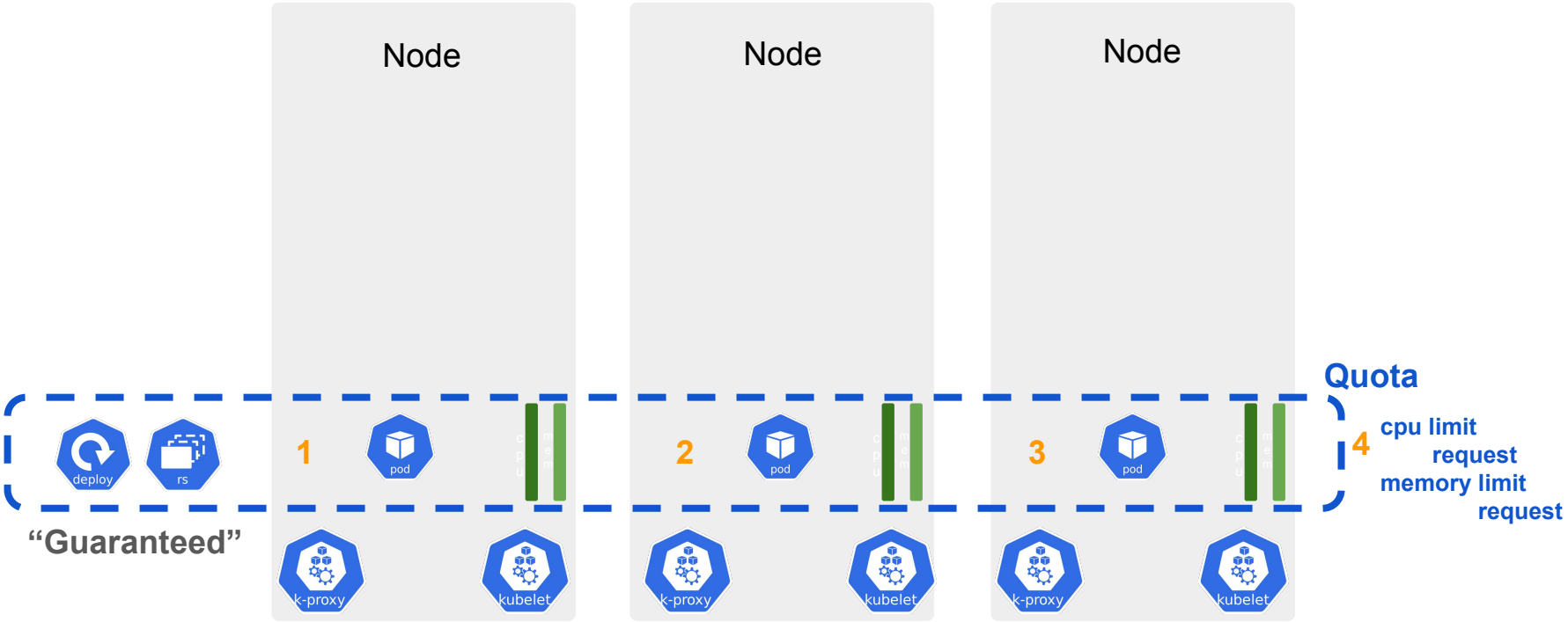
Get the most of Kubernetes

Instead you want to get the most out of Kubernetes ability to make the right decisions based on information about what are the important workloads.

Thus if you have a backend service that a bunch of frontend stuff is dependent on, this would probably be more important than any of the frontend workloads.

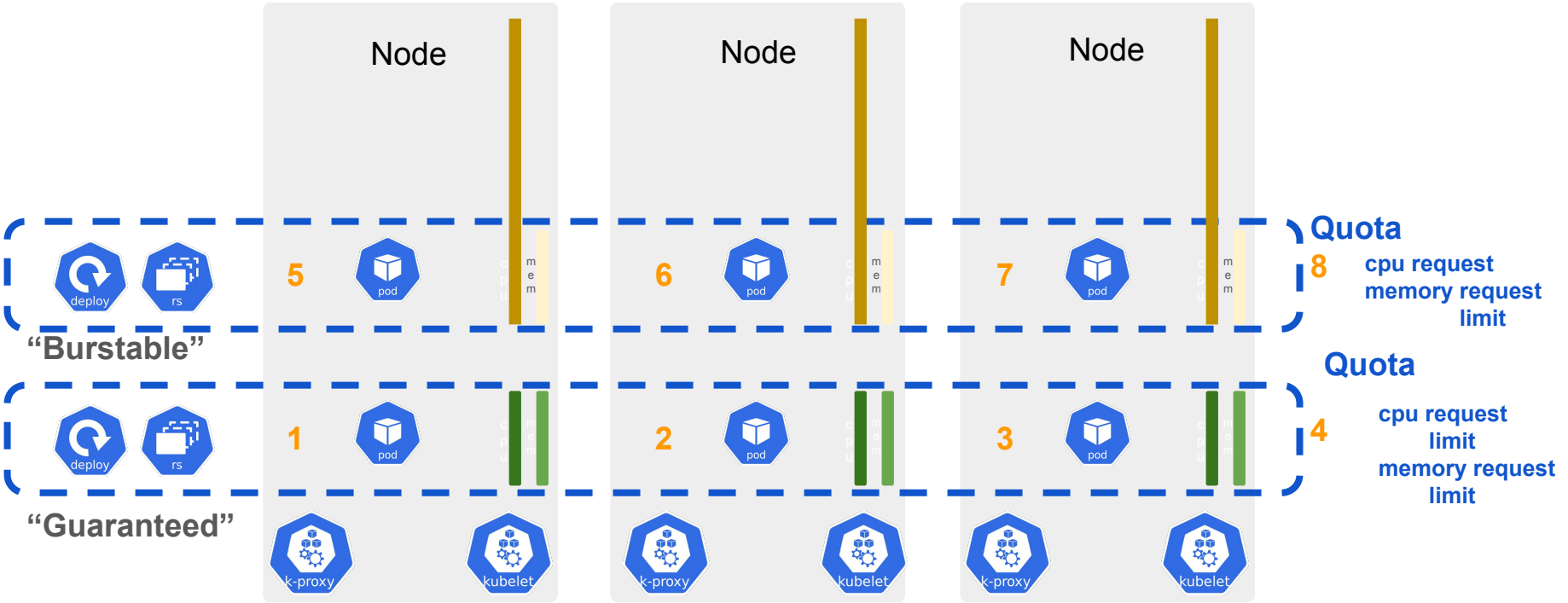
However some of the frontend workloads may be more important than others





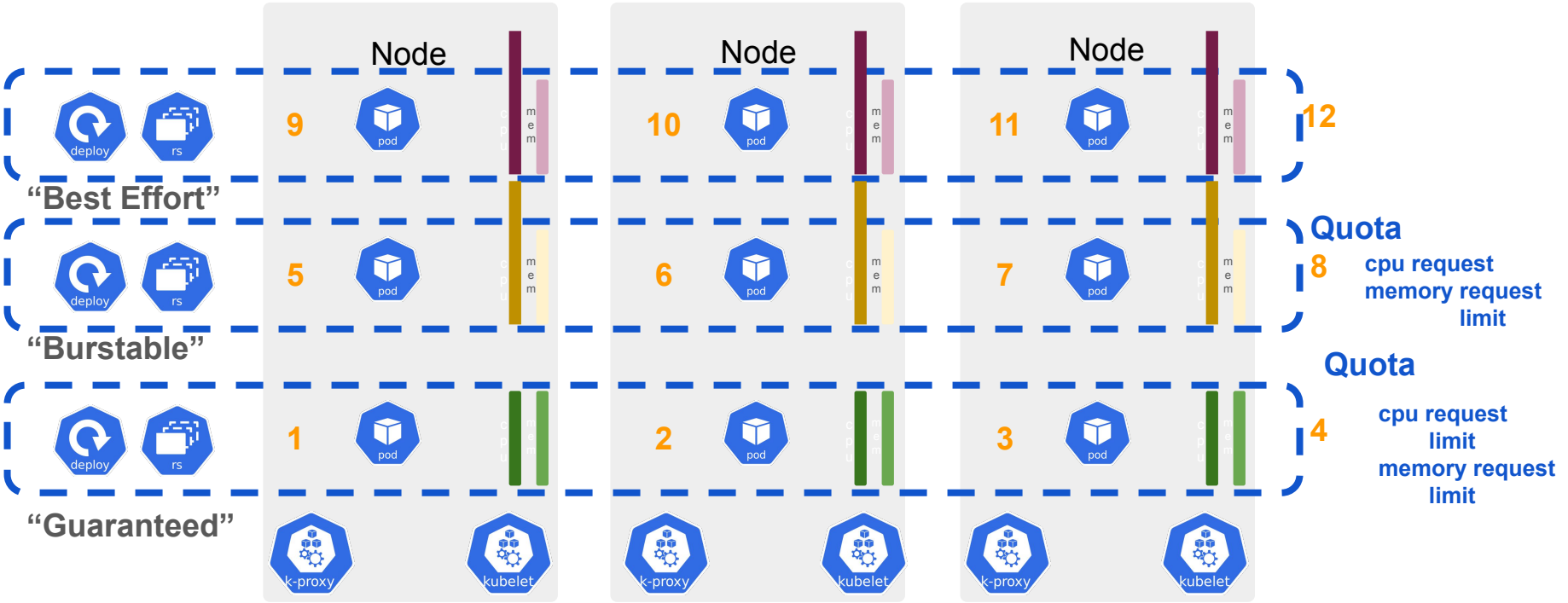
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

$1+2+3 < 4$



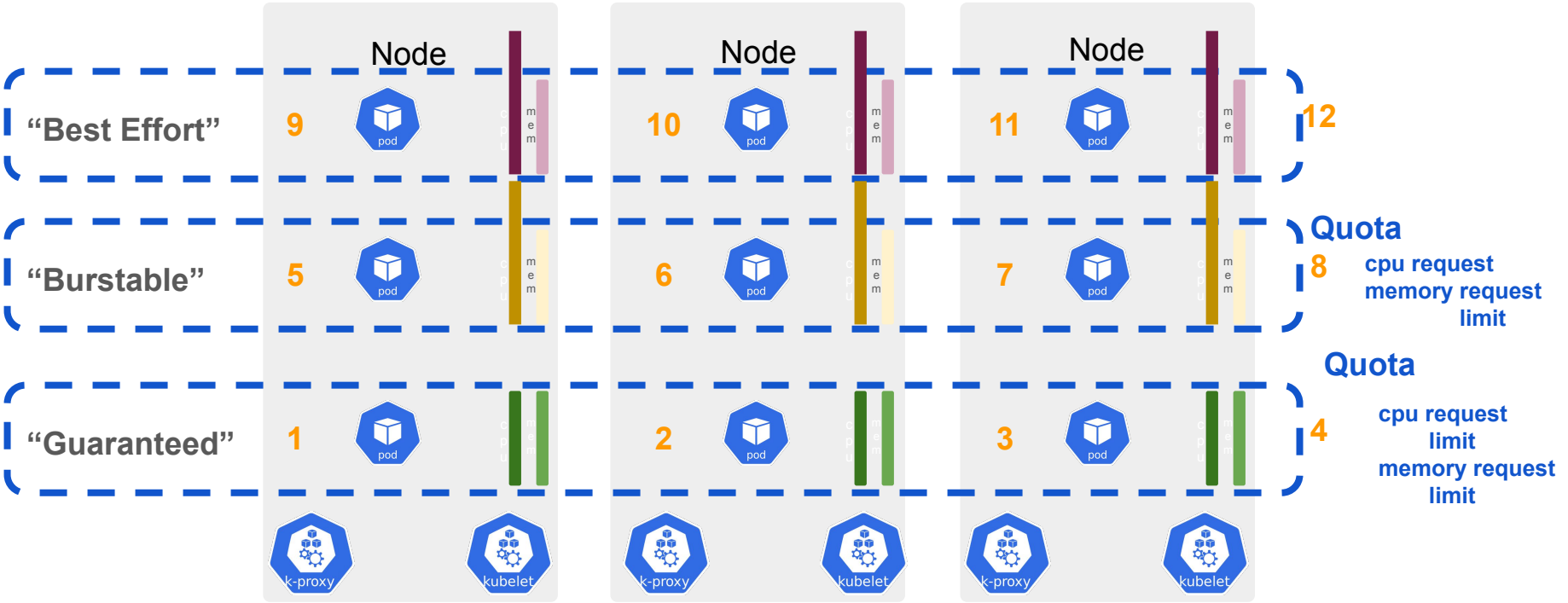
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

1+2+3 < 4 and 5+6+7 < 8



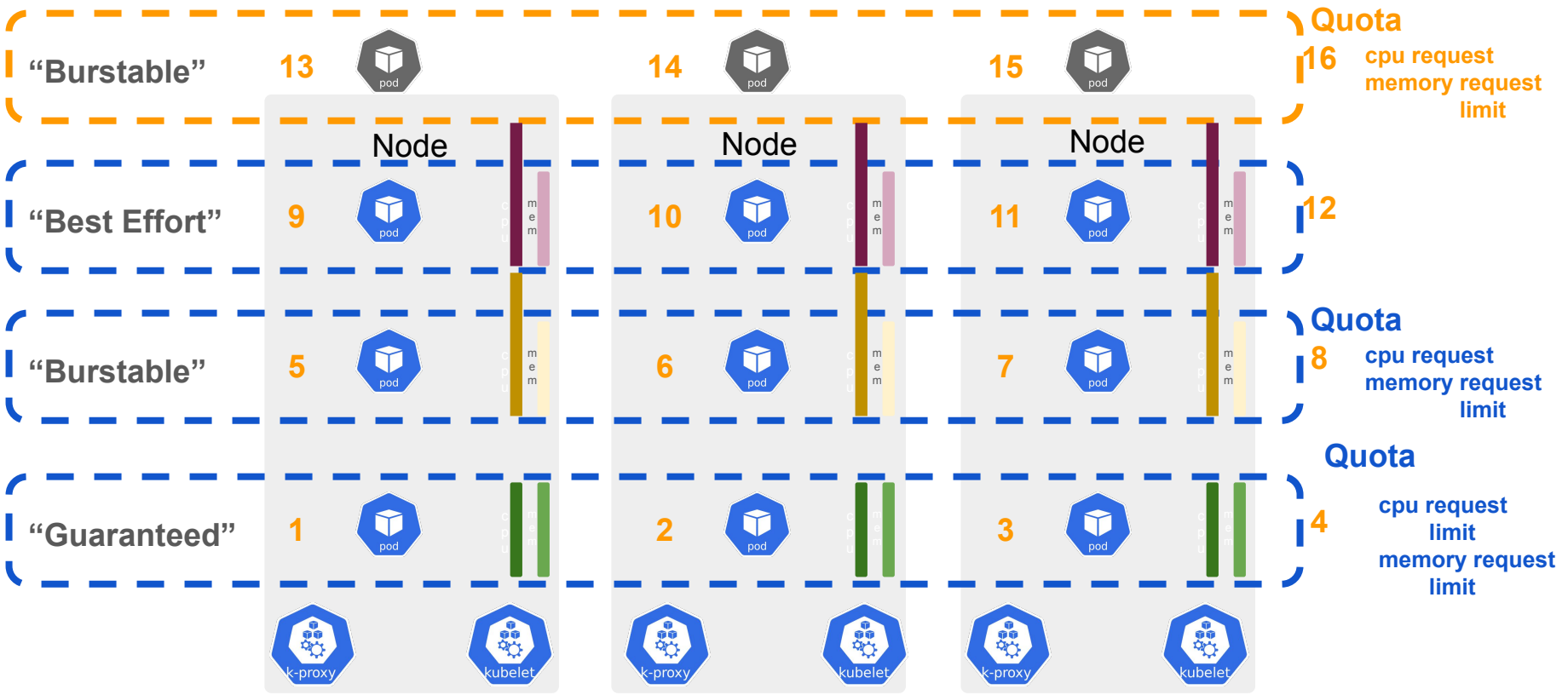
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

1+2+3 < 4 and 5+6+7 < 8



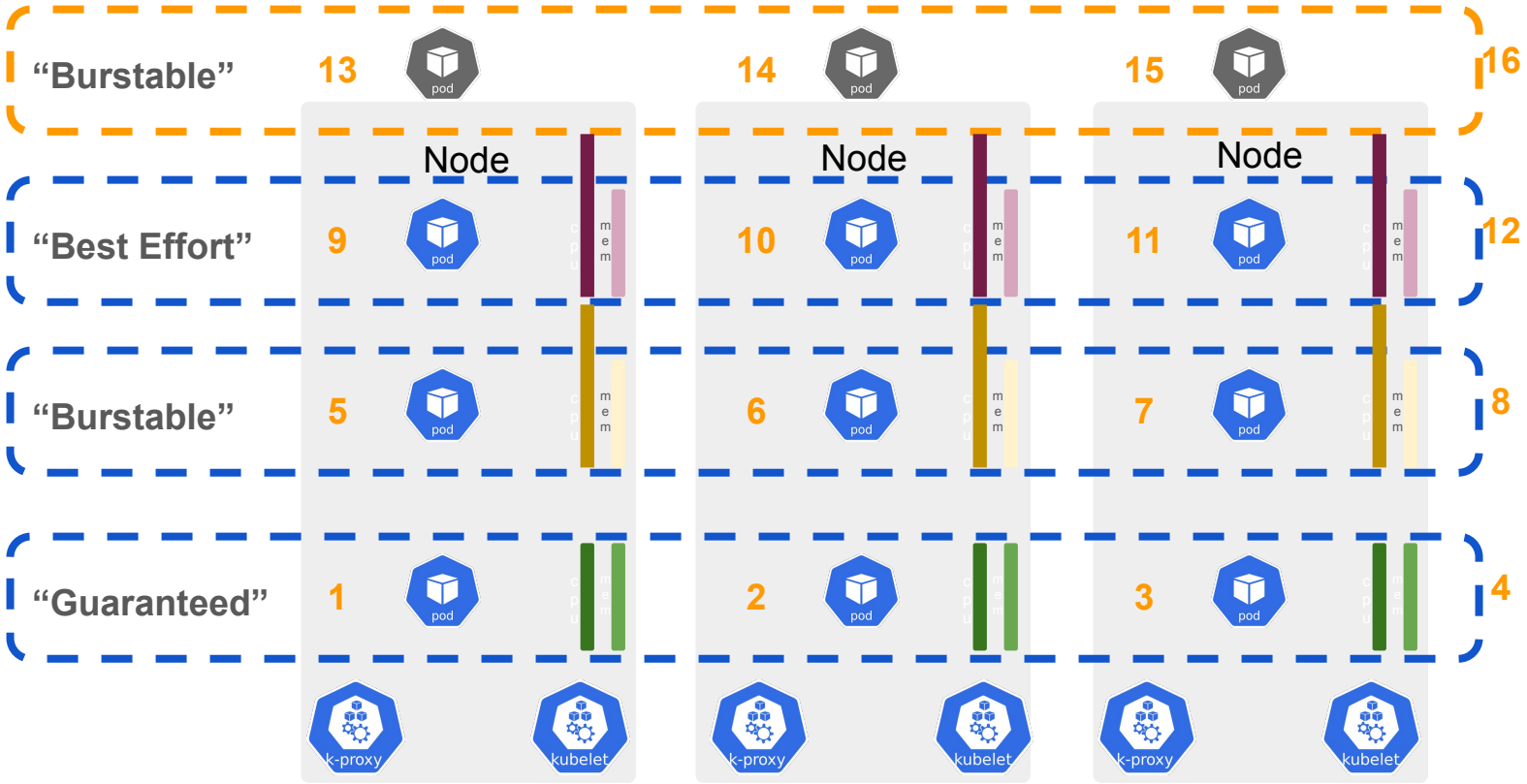
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

1+2+3 < 4 and 5+6+7 < 8



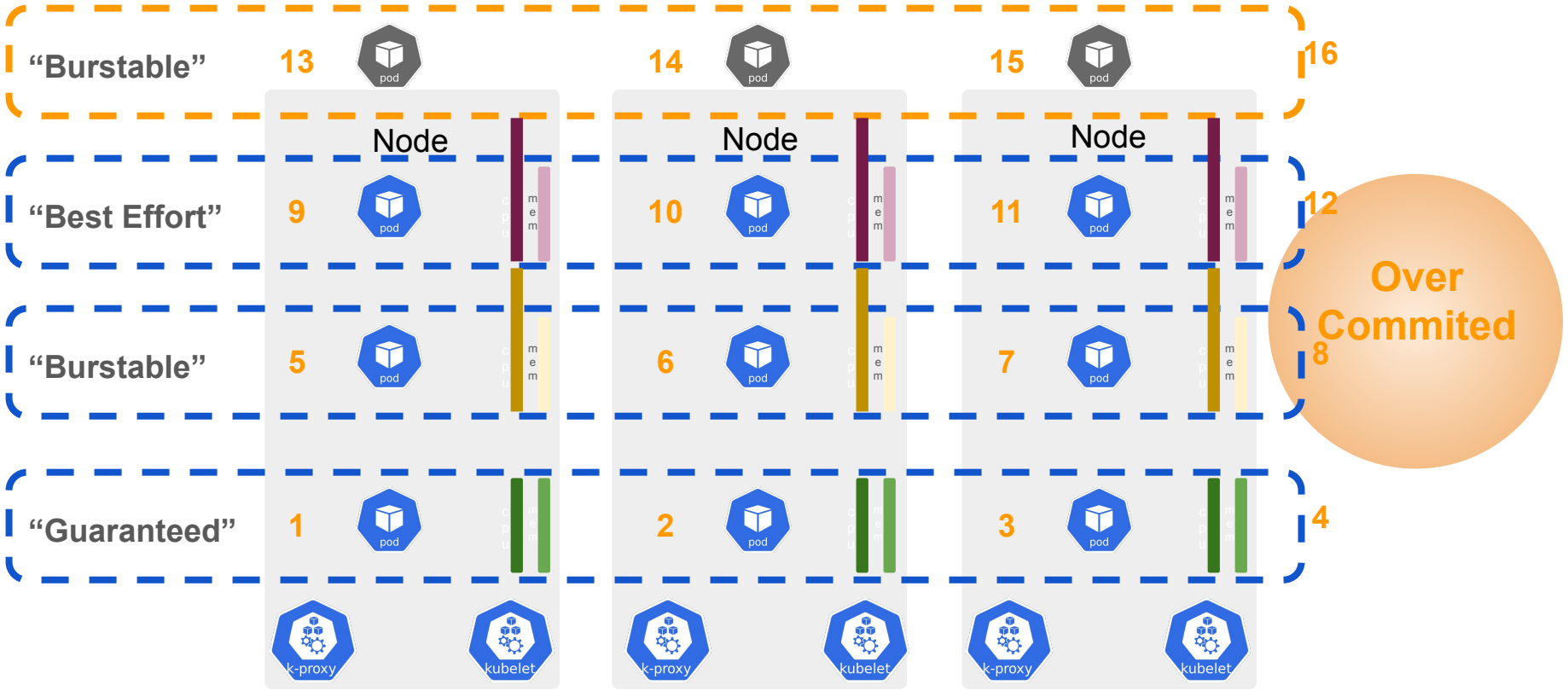
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

1+2+3 < 4 and 5+6+7 < 8 and 13+14+15 < 16



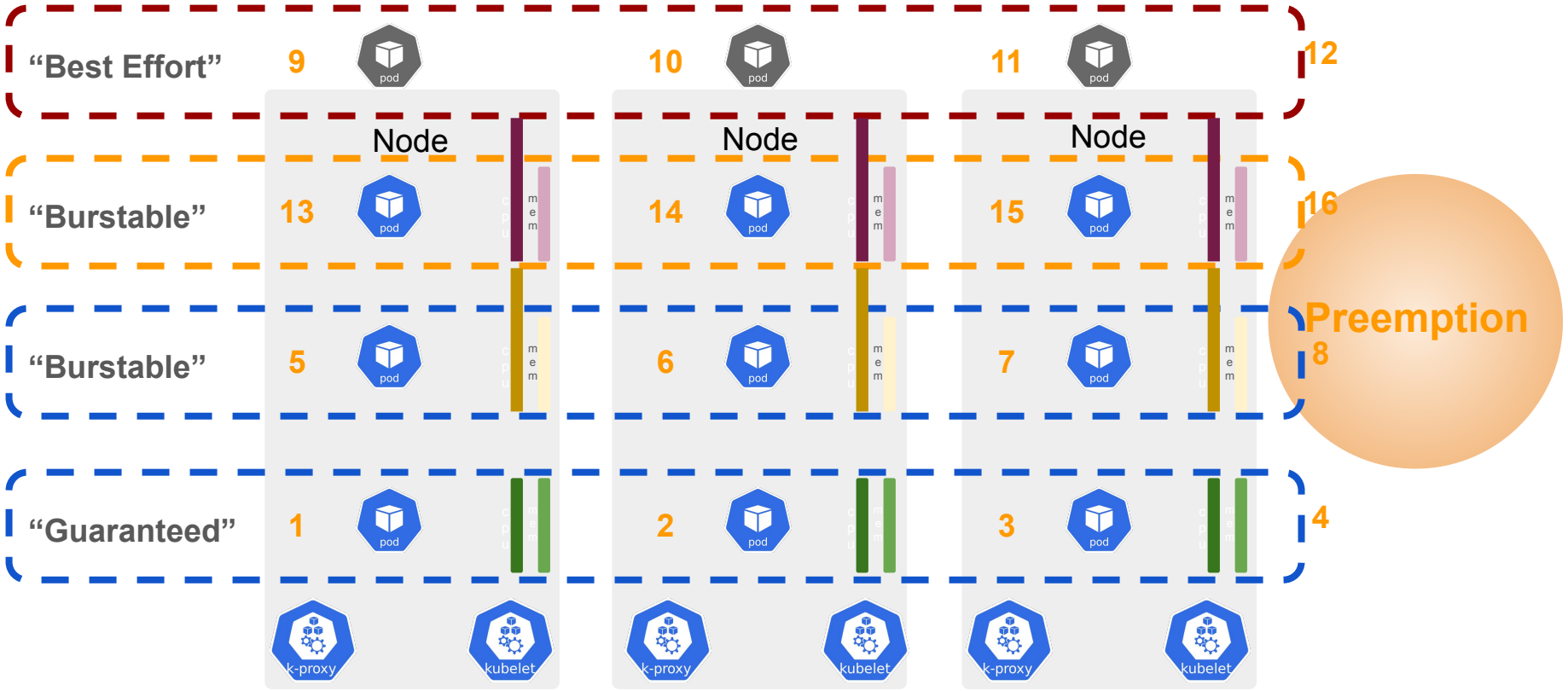
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

(1+2+3 < 4 and 5+6+7 < 8 and 9+10+11 < 12 and 13+14+15 < 16) ! **nodesize**



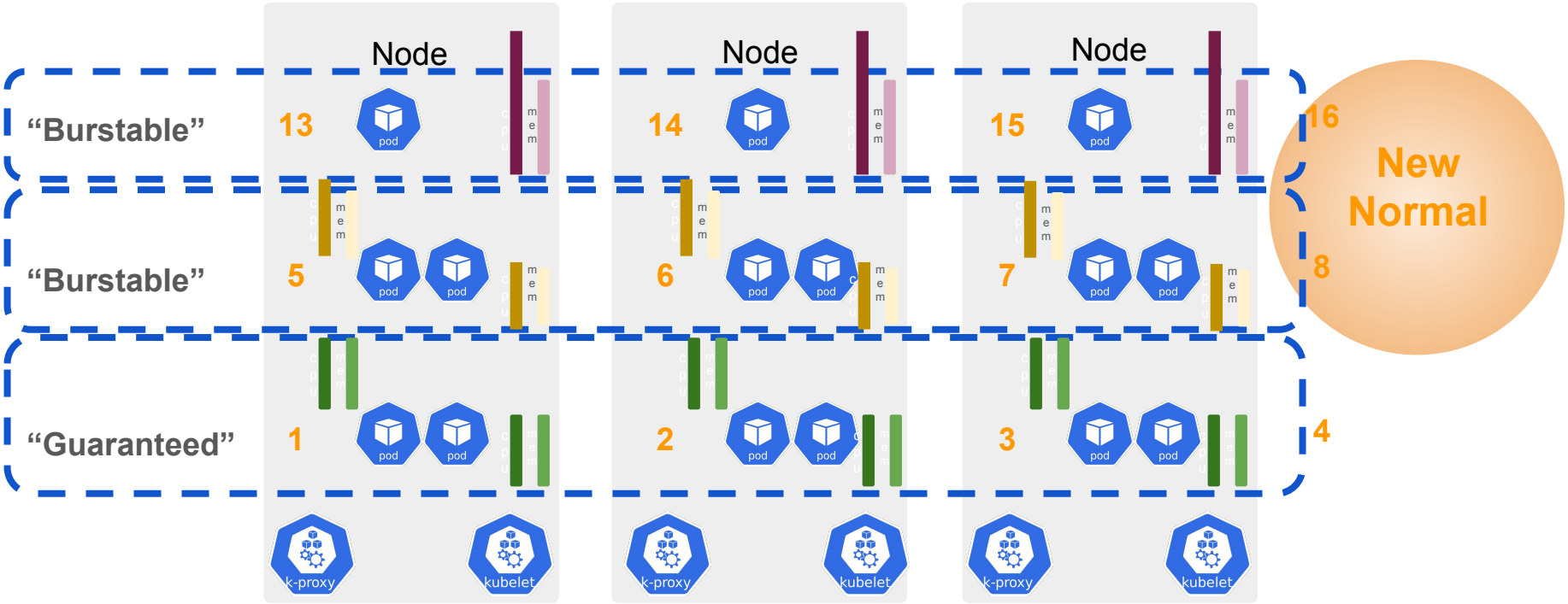
$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

(1+2+3 < 4 and 5+6+7 < 8 and 9+10+11 < 12 and 13+14+15 < 16) ! **nodesize**



$$\sum_{i=1}^{n} \text{pod quota}_i < \text{namespace quota}$$

(1+2+3 < 4 and 5+6+7 < 8 and 9+10+11 < 12 and 13+14+15 < 16) ! **nodesize**



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16) < \textbf{nodesize}$$

How does that look at the OS level?

```
/> kubectl get pods hello-foo-pdb-app-7bbcb979bb-4b4jq -o json | jq '.spec.containers | .[].resources'
{
  "limits": {
    "cpu": "90m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "80m",
    "memory": "25Mi"
  }
}
```

```
/> kubectl get pods hello-foo-pdb-app-7bbcb979bb-4b4jq -o json | jq '.spec.containers | .[].resources'
{
  "limits": {
    "cpu": "90m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "80m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-foo-pdb-app-7bbcb979bb-4b4jq -c hello-foo-pdb-app -- cat /sys/fs/cgroup/cpu.max
9000 100000
```

```
/> kubectl get pods hello-foo-pdb-app-7bbcb979bb-4b4jq -o json | jq '.spec.containers | .[].resources'
{
  "limits": {
    "cpu": "90m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "80m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-foo-pdb-app-7bbcb979bb-4b4jq -c hello-foo-pdb-app -- cat /sys/fs/cgroup/cpu.max
9000 100000

/> kubectl exec hello-foo-pdb-app-7bbcb979bb-4b4jq -c hello-foo-pdb-app -- cat /sys/fs/cgroup/cpu.weight
4
```

```
/> kubectl get pods hello-bar-pdb-app-5f9d4787f6-2n5vb -o json | jq '.spec.containers | .[].resources '
{
  "limits": {
    "cpu": "125m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "125m",
    "memory": "25Mi"
  }
}
```

```
/> kubectl get pods hello-bar-pdb-app-5f9d4787f6-2n5vb -o json | jq '.spec.containers | .[].resources '
{
  "limits": {
    "cpu": "125m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "125m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-bar-pdb-app-5f9d4787f6-2n5vb -c hello-bar-pdb-app -- cat /sys/fs/cgroup/cpu.max
12500 100000
```

```
/> kubectl get pods hello-bar-pdb-app-5f9d4787f6-2n5vb -o json | jq '.spec.containers | .[].resources '
{
  "limits": {
    "cpu": "125m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "125m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-bar-pdb-app-5f9d4787f6-2n5vb -c hello-bar-pdb-app -- cat /sys/fs/cgroup/cpu.max
12500 100000

/> kubectl exec hello-bar-pdb-app-5f9d4787f6-2n5vb -c hello-bar-pdb-app -- cat /sys/fs/cgroup/cpu.weight
5
```

```
/> kubectl get pods hello-foobar-app-76fd9fc6cb-6w46d -o json | jq '.spec.containers | .[].resources'
{
  "limits": {
    "cpu": "505m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "185m",
    "memory": "25Mi"
  }
}
```

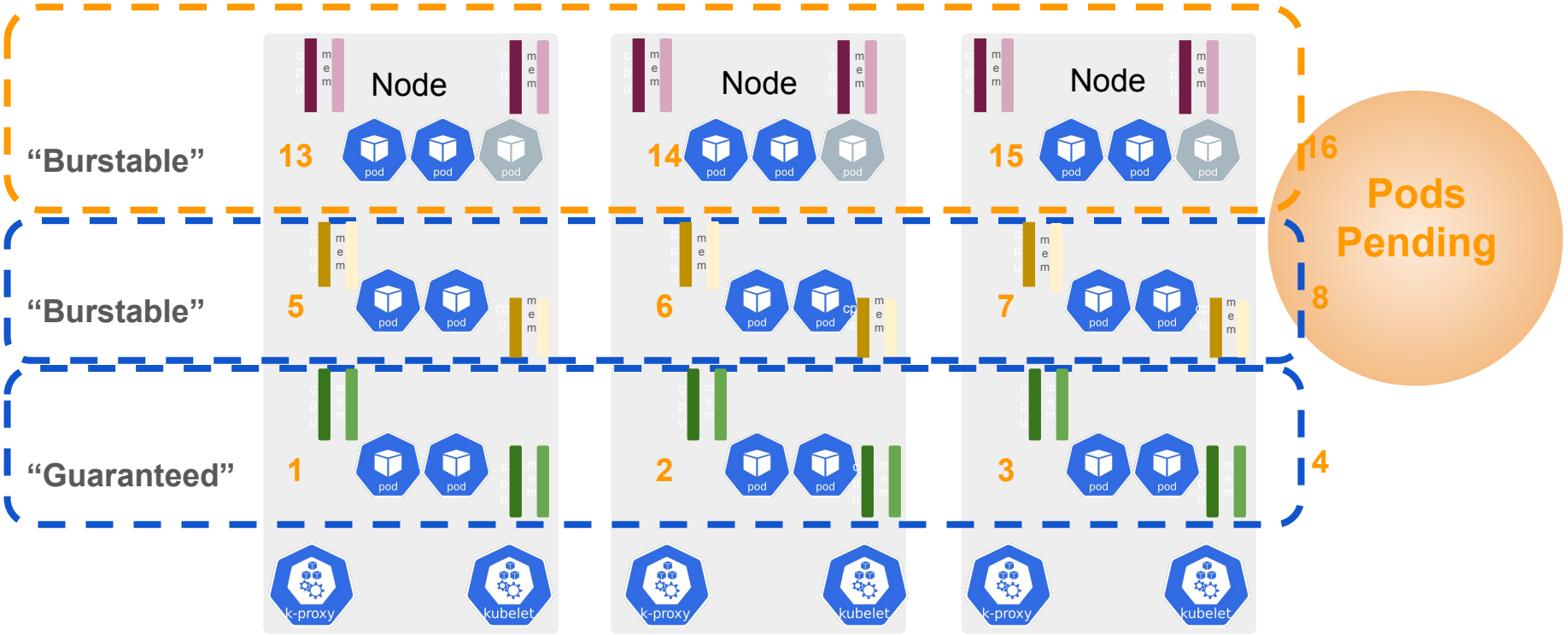
```
/> kubectl get pods hello-foobar-app-76fd9fc6cb-6w46d -o json | jq '.spec.containers | .[].resources'
{
  "limits": {
    "cpu": "505m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "185m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-foobar-app-76fd9fc6cb-6w46d -c hello-foobar-app -- cat /sys/fs/cgroup/cpu.max
50500 100000
```

```
/> kubectl get pods hello-foobar-app-76fd9fc6cb-6w46d -o json | jq '.spec.containers | .[].resources'
{
  "limits": {
    "cpu": "505m",
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "185m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-foobar-app-76fd9fc6cb-6w46d -c hello-foobar-app -- cat /sys/fs/cgroup/cpu.max
50500 100000

/> kubectl exec hello-foobar-app-76fd9fc6cb-6w46d -c hello-foobar-app -- cat /sys/fs/cgroup/cpu.weight
8
```

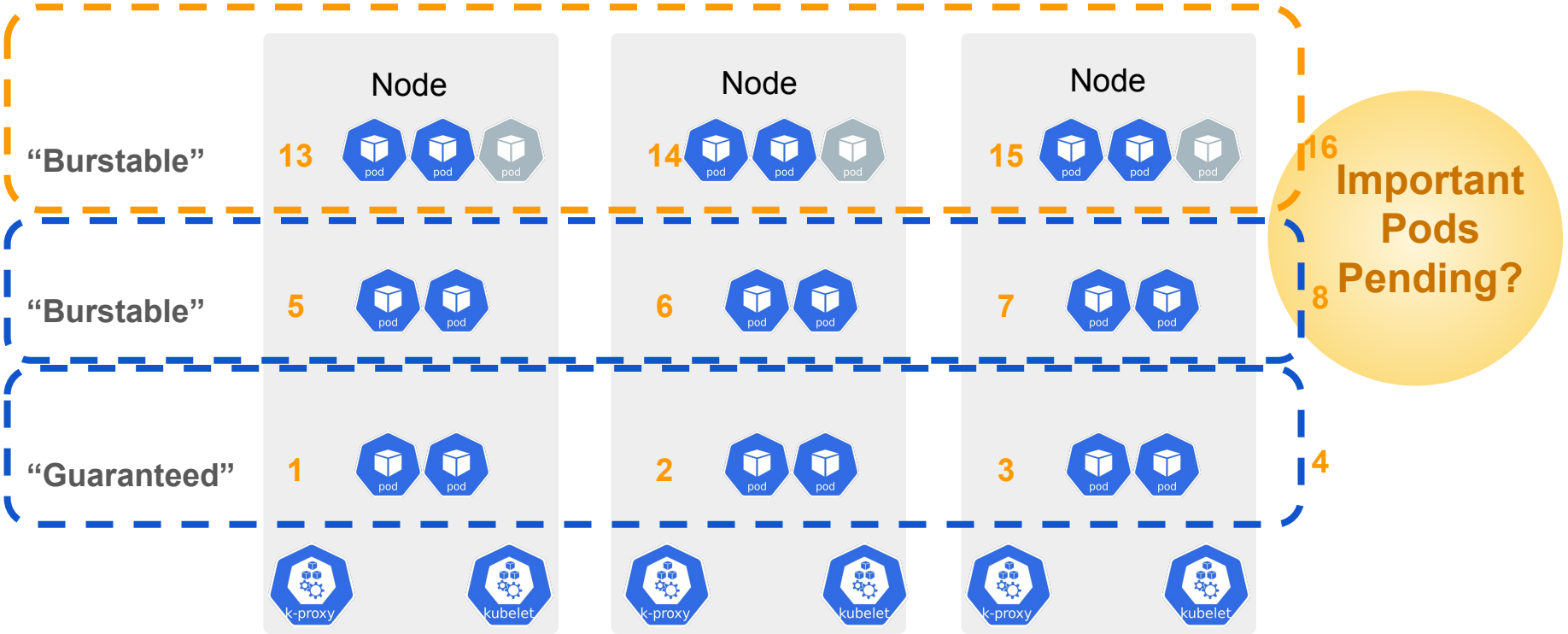
```
/> kubectl get pods hello-foo-app-nl-769d648d56-jnv9t -o json | jq '.spec.containers | .[].resources
{
  "limits": {
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "2800m",
    "memory": "25Mi"
  }
}
```

```
/> kubectl get pods hello-foo-app-nl-769d648d56-jnv9t -o json | jq '.spec.containers | .[].resources
{
  "limits": {
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "2800m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-foo-app-nl-769d648d56-jnv9t -c hello-foo-app-nl -- cat /sys/fs/cgroup/cpu.max
max 100000
```

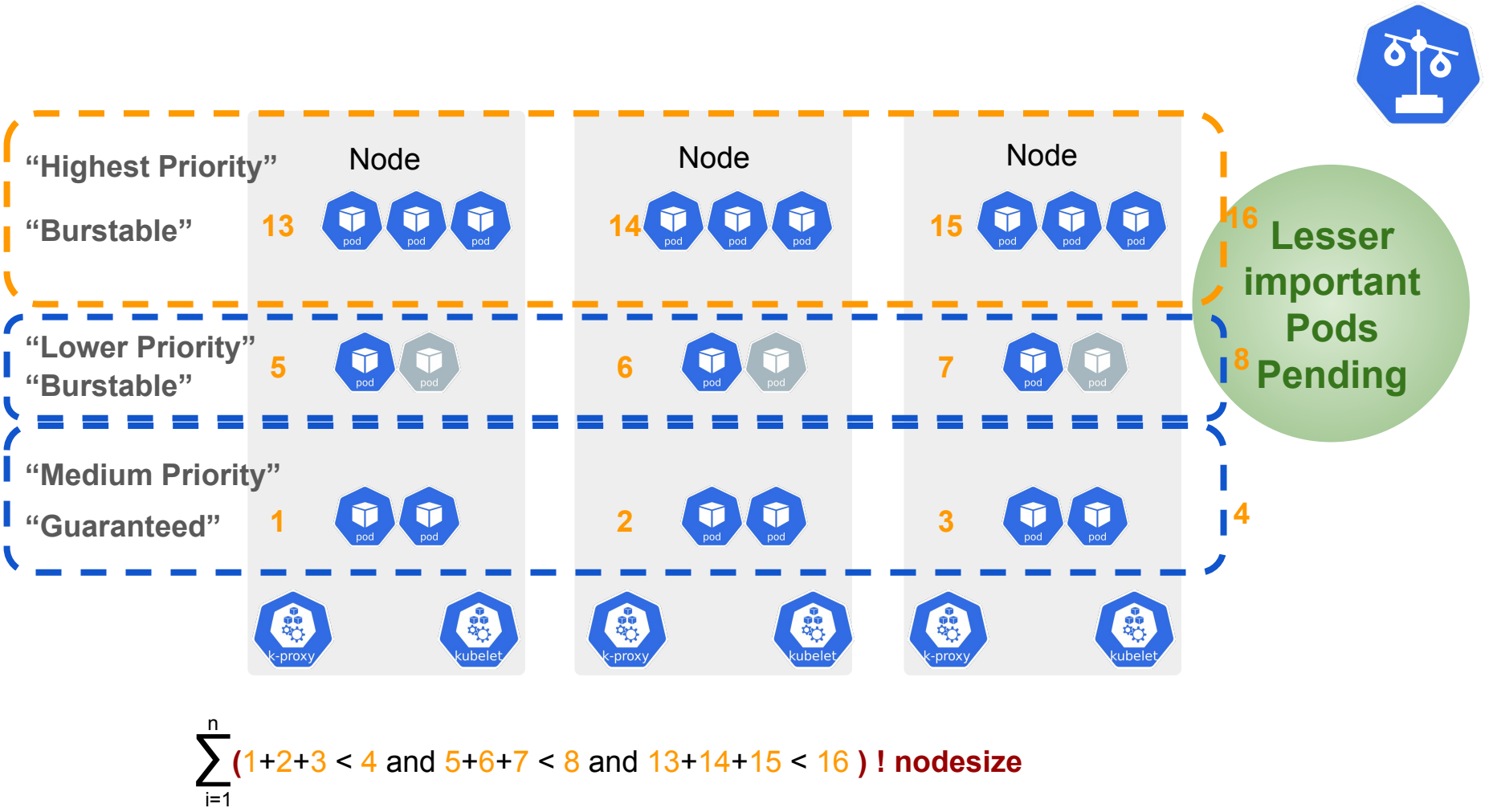
```
/> kubectl get pods hello-foo-app-nl-769d648d56-jnv9t -o json | jq '.spec.containers | .[].resources
{
  "limits": {
    "memory": "25Mi"
  },
  "requests": {
    "cpu": "2800m",
    "memory": "25Mi"
  }
}

/> kubectl exec hello-foo-app-nl-769d648d56-jnv9t -c hello-foo-app-nl -- cat /sys/fs/cgroup/cpu.max
max 100000

/> kubectl exec hello-foo-app-nl-769d648d56-jnv9t -c hello-foo-app-nl -- cat /sys/fs/cgroup/cpu.weight
110
```



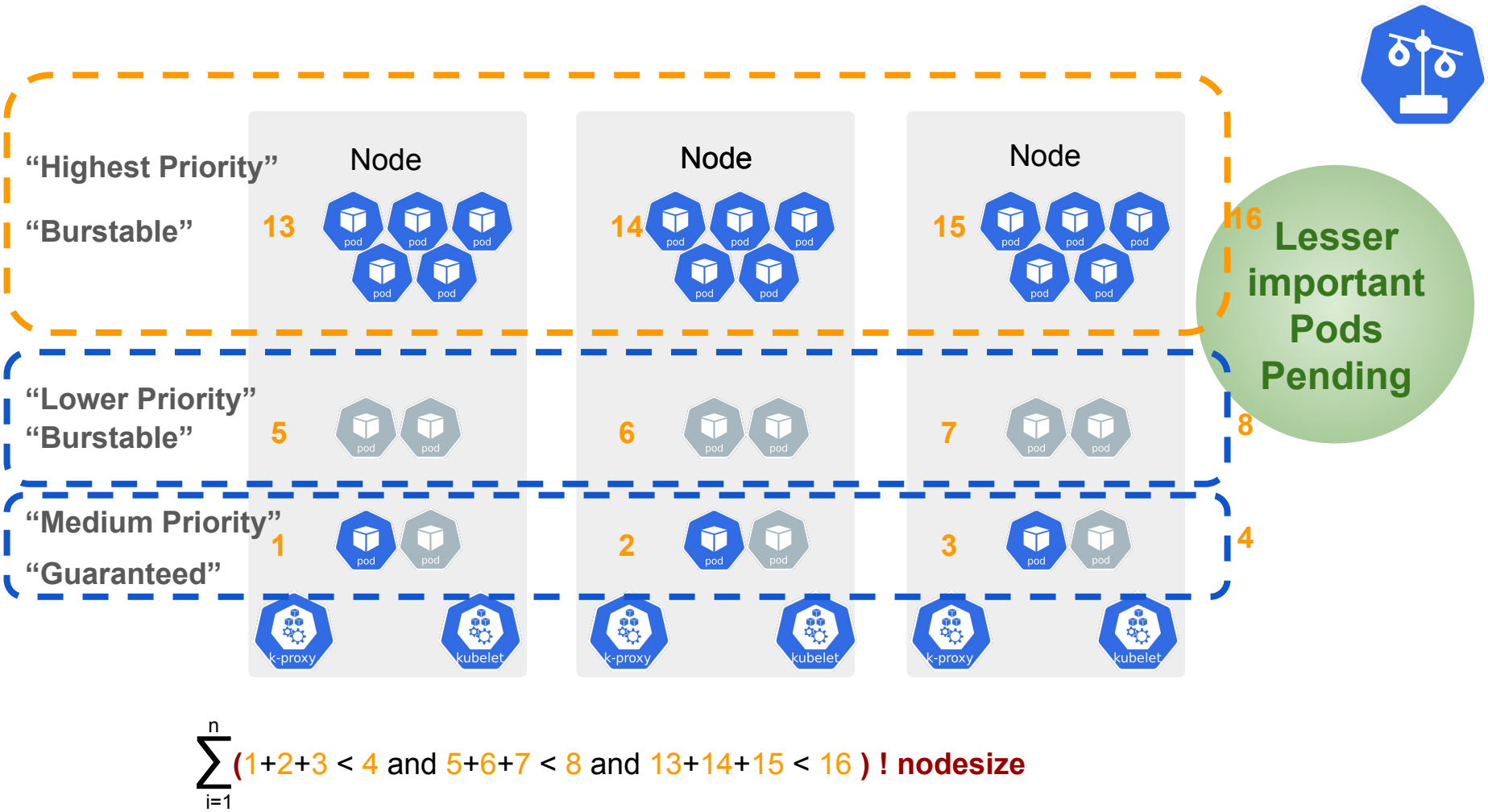
$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$

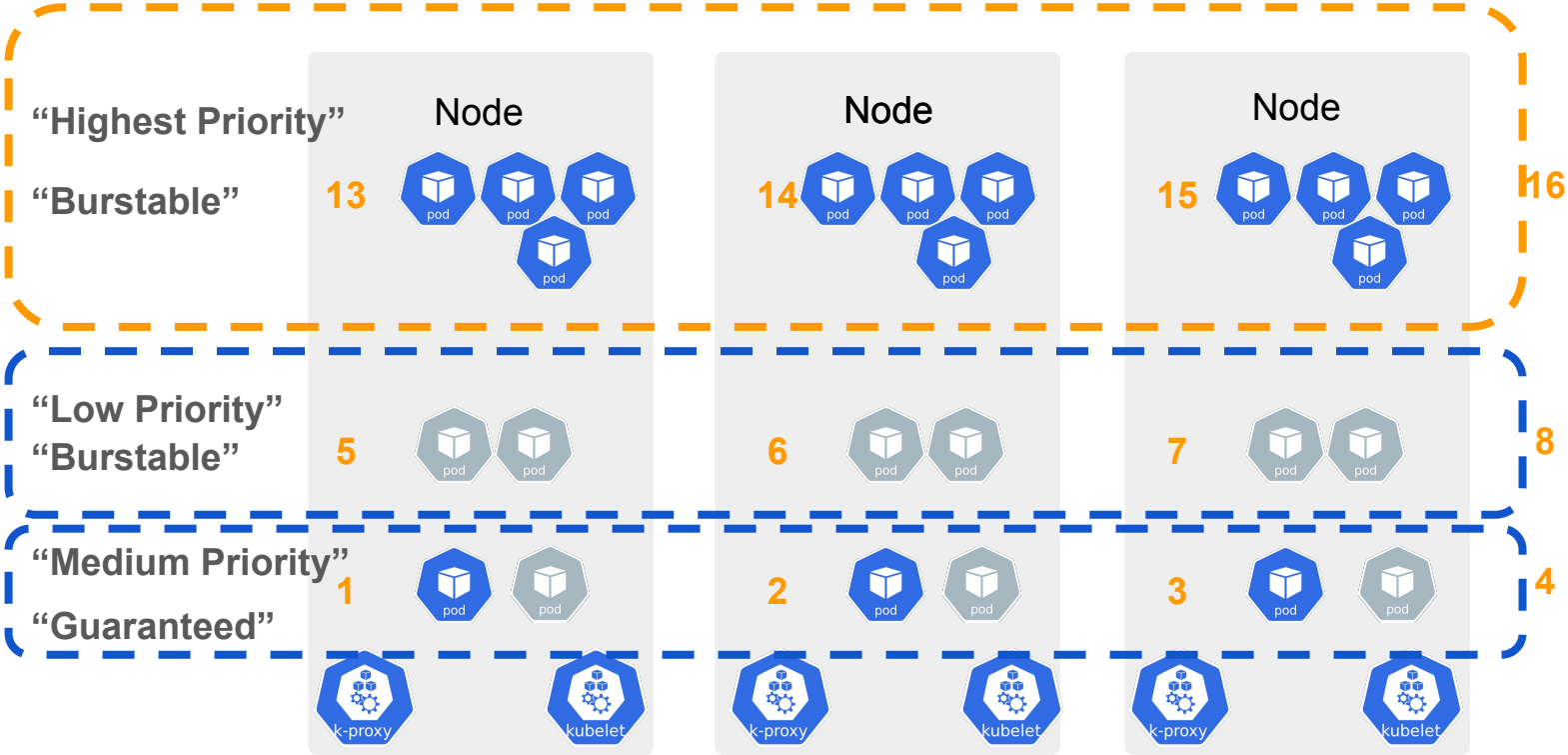


$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$

Priorities?



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$

```yaml
kind: PriorityClass
metadata:
  name: higest-priority
value: 0000090000
preemptionPolicy: "PreemptLowerPriority"
globalDefault: false
description: "This is the higest Priority Class value for this example"
---
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: medium-priority
value: 0000080000
preemptionPolicy: "PreemptLowerPriority"
globalDefault: false
description: "This is the medium Priority Class value for this example"
---
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: low-priority
value: 0000060000
preemptionPolicy: "PreemptLowerPriority"
globalDefault: true
description: "This is the normal and lowest Priority Class value for this example"
---
```

```
/> kubectl create -f ./priority-classes.yaml
priorityclass.scheduling.k8s.io/higest-priority created
priorityclass.scheduling.k8s.io/medium-priority created
priorityclass.scheduling.k8s.io/low-priority created
```

```
/> kubectl create -f ./priority-classes.yaml
priorityclass.scheduling.k8s.io/higest-priority created
priorityclass.scheduling.k8s.io/medium-priority created
priorityclass.scheduling.k8s.io/low-priority created

/> kubectl get priorityclasses

NAME                      VALUE        GLOBAL-DEFAULT   AGE
higest-priority           90000        false            7s
medium-priority           80000        false            7s
low-priority              24576        true             7s
system-cluster-critical   2000000000   false            100s
system-node-critical      2000001000   false            100s
```

```
/> kubectl create -f ./priority-classes.yaml
priorityclass.scheduling.k8s.io/higest-priority created
priorityclass.scheduling.k8s.io/medium-priority created
priorityclass.scheduling.k8s.io/low-priority created

/> kubectl get priorityclasses

NAME                      VALUE        GLOBAL-DEFAULT   AGE
higest-priority           90000        false            7s
medium-priority           80000        false            7s
low-priority              24576        true             7s
system-cluster-critical   2000000000   false            100s
system-node-critical      2000001000   false            100s

/> kubectl get priorityclasses -o yaml | grep preemptionPolicy
  preemptionPolicy: PreemptLowerPriority
  preemptionPolicy: PreemptLowerPriority
  preemptionPolicy: PreemptLowerPriority
  preemptionPolicy: PreemptLowerPriority
  preemptionPolicy: PreemptLowerPriority
```

```
/> kubectl get priorityclasses -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]
|  sort_by (.priority) | reverse' | jq -r ' .[] | .name +  " -  " + (.priority|tostring) '
system-node-critical -  2000001000
system-cluster-critical -  2000000000
higest-priority -  90000
medium-priority -  80000
low-priority -  24576
```

```
/> cat deployment-high-prio.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/name: hello-foo
    app.kubernetes.io/instance: hello-foo
  name: hello-foo-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app.kubernetes.io/name: hello-foo
      app.kubernetes.io/instance: hello-foo
  template:
    metadata:
      labels:
        app.kubernetes.io/name: hello-foo
        app.kubernetes.io/instance: hello-foo
```

```yaml
spec:
  priorityClassName: higest-priority
  containers:
    - name: hello-foo-app
      command:
        - /agnhost
        - netexec
        - --http-port=8080
      image: registry.k8s.io/e2e-test-images/agnhost:2.39
      ports:
        - name: http
          containerPort: 8080
          protocol: TCP
      resources:
        limits:
          cpu: 90m
          memory: 25Mi
        requests:
          cpu: 80m
          memory: 25Mi
```

```
/> cat deployment-low-prio.yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/name: hello-baz
    app.kubernetes.io/instance: hello-baz
  name: hello-baz-app
spec:
  replicas: 8
  selector:
    matchLabels:
      app.kubernetes.io/name: hello-baz
      app.kubernetes.io/instance: hello-baz
  template:
    metadata:
      labels:
        app.kubernetes.io/name: hello-baz
        app.kubernetes.io/instance: hello-baz
```

```yaml
spec:
  priorityClassName: low-priority
  containers:
    - command:
        - /agnhost
        - netexec
        - --http-port=8080
      image: registry.k8s.io/e2e-test-images/agnhost:2.39
      name: hello-baz-app
      ports:
        - name: http
          containerPort: 8080
          protocol: TCP
      resources:
        limits:
          cpu: 505m
          memory: 25Mi
        requests:
          cpu: 185m
          memory: 25Mi
```

```
/> cat deployment-no-prio.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/name: hello-foobar
    app.kubernetes.io/instance: hello-foobar
  name: hello-foobar-app
spec:
  replicas: 16
  selector:
    matchLabels:
      app.kubernetes.io/name: hello-foobar
      app.kubernetes.io/instance: hello-foobar
  template:
    metadata:
      labels:
        app.kubernetes.io/name: hello-foobar
        app.kubernetes.io/instance: hello-foobar
```

```yaml
spec:
  containers:
    - command:
        - /agnhost
        - netexec
        - --http-port=8080
      image: registry.k8s.io/e2e-test-images/agnhost:2.39
      name: hello-foobar-app
      ports:
        - name: http
          containerPort: 8080
          protocol: TCP
      resources:
        limits:
          cpu: 505m
          memory: 25Mi
        requests:
          cpu: 185m
          memory: 25Mi
```

```
/> kubectl create -f ./deployment-no-prio.yaml
deployment.apps/hello-foobar-app created
```

```
/> kubectl create -f ./deployment-no-prio.yaml
deployment.apps/hello-foobar-app created

/> kubectl get deployments/hello-foobar-app
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   16/16   16           16          36s
```

```
/> kubectl create -f ./deployment-no-prio.yaml
deployment.apps/hello-foobar-app created

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   16/16   16           16          36s

/> kubectl get deployments/hello-foobar-app -o yaml | grep priorityClassName
priorityClassName: low-priority
```

```
/> kubectl get nodes -o json | jq '.items | .[].status.allocatable.pods'
“110”
“110”
“110”
```

```
/> kubectl get nodes -o json | jq '.items | .[].status.allocatable.pods'
“110”
“110”
“110”

/> kubectl scale --replicas=90 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled
```

```
/> kubectl get nodes -o json | jq '.items | .[].status.allocatable.pods'
“110”
“110”
“110”

/> kubectl scale --replicas=90 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   86/90   90           86          13m
```

```
/> kubectl get nodes -o json | jq '.items | .[].status.allocatable.pods'
“110”
“110”
“110”

/> kubectl scale --replicas=90 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   86/90   90           86          13m

/> kubectl create -f ./deployment-medium-prio.yaml
deployment.apps/hello-bar-app created
```

```
/> kubectl get nodes -o json | jq '.items | .[].status.allocatable.pods'
“110”
“110”
“110”

/> kubectl scale --replicas=90 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   86/90   90           86          13m

/> kubectl create -f ./deployment-medium-prio.yaml
deployment.apps/hello-bar-app created

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   83/90   90           83          13m
```

```
/> kubectl get nodes -o json | jq '.items | .[].status.allocatable.pods'
“110”
“110”
“110”

/> kubectl scale --replicas=90 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   86/90   90           86          13m

/> kubectl create -f ./deployment-medium-prio.yaml
deployment.apps/hello-bar-app created

/> kubectl get deployments/hello-foobar-app
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
hello-foobar-app   83/90   90           83          13m

/> kubectl get deployments/hello-bar-app
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app   4/4     4            4           29s
```

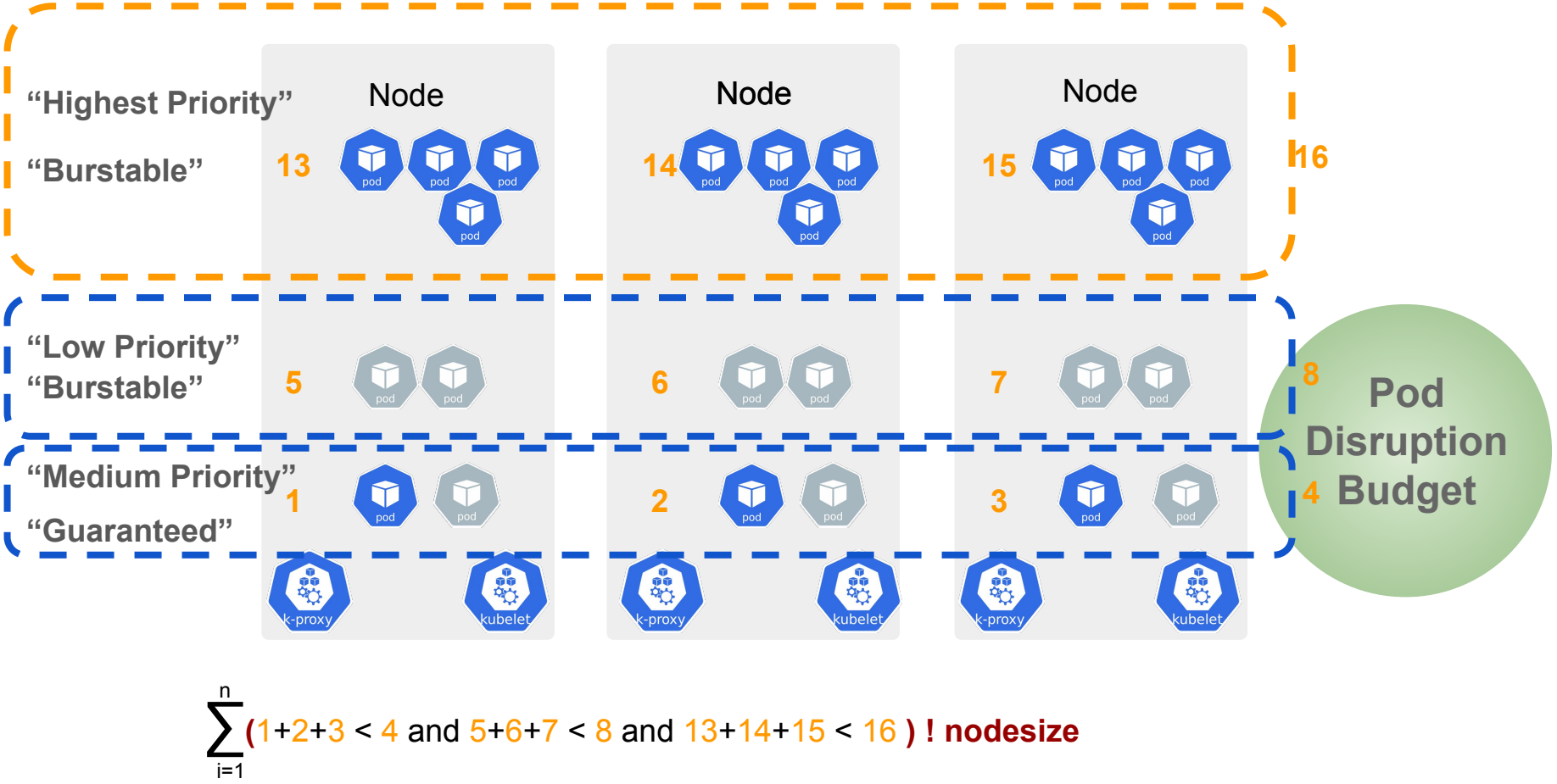
```
/> kubectl create -f ./deployment-low-prio.yaml
deployment.apps/hello-baz-app created
```

```
/> kubectl create -f ./deployment-low-prio.yaml
deployment.apps/hello-baz-app created

/> kubectl get deployments/hello-baz-app
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
hello-baz-app   0/8     8            0           17s
```

```
/> kubectl create -f ./deployment-low-prio.yaml
deployment.apps/hello-baz-app created

/> kubectl get deployments/hello-baz-app
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
hello-baz-app   0/8     8            0           17s

/> kubectl create -f ./deployment-high-prio.yaml
deployment.apps/hello-foo-app created

/> kubectl get deployments/hello-foo-app
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
hello-foo-app   2/2     2            2           18s
```

```
/> kubectl create -f ./deployment-low-prio.yaml
deployment.apps/hello-baz-app created

/> kubectl get deployments/hello-baz-app
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
hello-baz-app   0/8     8            0           17s

/> kubectl create -f ./deployment-high-prio.yaml
deployment.apps/hello-foo-app created

/> kubectl get deployments/hello-foo-app
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
hello-foo-app   2/2     2            2           18s

/> kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app     4/4     4            4           112s
hello-baz-app     0/8     8            0           63s
hello-foo-app     2/2     2            2           30s
hello-foobar-app  82/90   90           82          15m
```

What about PodDisruptionBudgets?



$$\sum_{i=1}^{n} (1+2+3 < 4 \text{ and } 5+6+7 < 8 \text{ and } 13+14+15 < 16 \text{) ! nodesize}$$

```
/> cat ./poddisruptionbudgets.yaml
---
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: high-prio-pdb
spec:
  minAvailable: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: hello-foo-pdb
---
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: medium-prio-pdb
spec:
  minAvailable: 3
  selector:
    matchLabels:
      app.kubernetes.io/name: hello-bar-pdb
---
```

```
/> kubectl create -f ./poddisruptionbudgets.yaml
poddisruptionbudget.policy/high-prio-pdb created
poddisruptionbudget.policy/medium-prio-pdb created
```

```
/> kubectl create -f ./poddisruptionbudgets.yaml
poddisruptionbudget.policy/high-prio-pdb created
poddisruptionbudget.policy/medium-prio-pdb created

/> kubectl create -f ./deployment-high-prio-pdb.yaml
deployment.apps/hello-foo-pdb-app created
```

```
/> kubectl create -f ./poddisruptionbudgets.yaml
poddisruptionbudget.policy/high-prio-pdb created
poddisruptionbudget.policy/medium-prio-pdb created

/> kubectl create -f ./deployment-high-prio-pdb.yaml
deployment.apps/hello-foo-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           2m48s
hello-baz-app       0/8     8            0           119s
hello-foo-app       2/2     2            2           86s
hello-foo-pdb-app   4/4     4            4           7s
hello-foobar-app    80/90   90           80          16m
```

```
/> kubectl create -f ./poddisruptionbudgets.yaml
poddisruptionbudget.policy/high-prio-pdb created
poddisruptionbudget.policy/medium-prio-pdb created

/> kubectl create -f ./deployment-high-prio-pdb.yaml
deployment.apps/hello-foo-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           2m48s
hello-baz-app       0/8     8            0           119s
hello-foo-app       2/2     2            2           86s
hello-foo-pdb-app   4/4     4            4           7s
hello-foobar-app    80/90   90           80          16m

/> kubectl scale --replicas=80 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl create -f ./poddisruptionbudgets.yaml
poddisruptionbudget.policy/high-prio-pdb created
poddisruptionbudget.policy/medium-prio-pdb created

/> kubectl create -f ./deployment-high-prio-pdb.yaml
deployment.apps/hello-foo-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           2m48s
hello-baz-app       0/8     8            0           119s
hello-foo-app       2/2     2            2           86s
hello-foo-pdb-app   4/4     4            4           7s
hello-foobar-app    80/90   90           80          16m

/> kubectl scale --replicas=80 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m
hello-baz-app       0/8     8            0           3m11s
hello-foo-app       2/2     2            2           2m38s
hello-foo-pdb-app   80/80   80           80          79s
hello-foobar-app    48/90   90           48          17m
```

```
/> kubectl create -f ./deployment-medium-prio-pbd.yaml
deployment.apps/hello-bar-pdb-app created
```

```
/> kubectl create -f ./deployment-medium-prio-pbd.yaml
deployment.apps/hello-bar-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m24s
hello-bar-pdb-app   8/8     8            8           6s
hello-baz-app       0/8     8            0           3m35s
hello-foo-app       2/2     2            2           3m2s
hello-foo-pdb-app   80/80   80           80          103s
hello-foobar-app    42/90   90           42          18m
```

```
/> kubectl create -f ./deployment-medium-prio-pbd.yaml
deployment.apps/hello-bar-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m24s
hello-bar-pdb-app   8/8     8            8           6s
hello-baz-app       0/8     8            0           3m35s
hello-foo-app       2/2     2            2           3m2s
hello-foo-pdb-app   80/80   80           80          103s
hello-foobar-app    42/90   90           42          18m

/> echo “balancing”

/> kubectl scale --replicas=80 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled
```

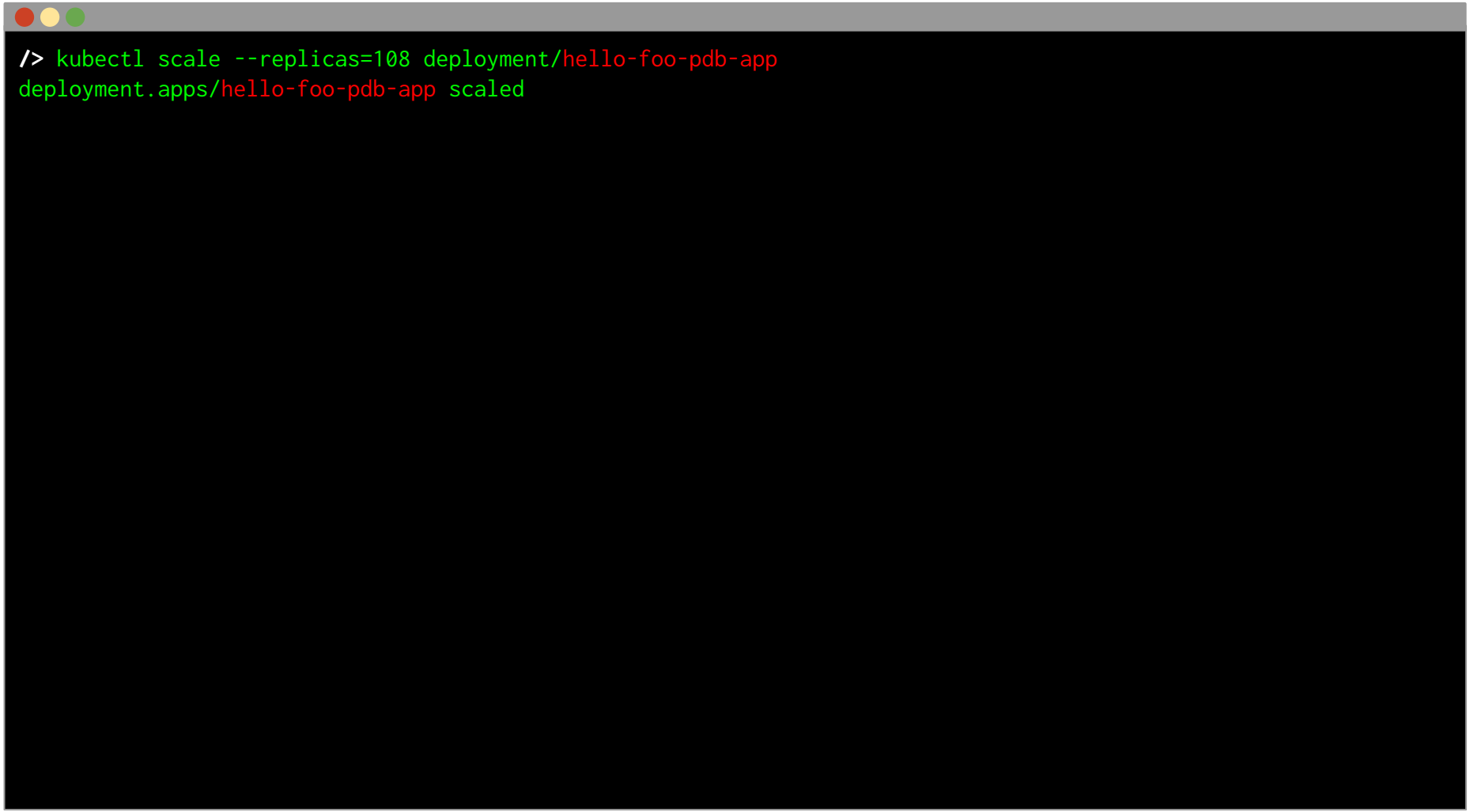
```
/> kubectl create -f ./deployment-medium-prio-pbd.yaml
deployment.apps/hello-bar-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m24s
hello-bar-pdb-app   8/8     8            8           6s
hello-baz-app       0/8     8            0           3m35s
hello-foo-app       2/2     2            2           3m2s
hello-foo-pdb-app   80/80   80           80          103s
hello-foobar-app    42/90   90           42          18m

/> echo “balancing”

/> kubectl scale --replicas=80 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl scale --replicas=80 deployment/hello-baz-app
deployment.apps/hello-baz-app scaled
```

```
/> kubectl create -f ./deployment-medium-prio-pbd.yaml
deployment.apps/hello-bar-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m24s
hello-bar-pdb-app   8/8     8            8           6s
hello-baz-app       0/8     8            0           3m35s
hello-foo-app       2/2     2            2           3m2s
hello-foo-pdb-app   80/80   80           80          103s
hello-foobar-app    42/90   90           42          18m

/> echo “balancing”

/> kubectl scale --replicas=80 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl scale --replicas=80 deployment/hello-baz-app
deployment.apps/hello-baz-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-app
deployment.apps/hello-bar-app scaled
```

```
/> kubectl create -f ./deployment-medium-prio-pbd.yaml
deployment.apps/hello-bar-pdb-app created

/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m24s
hello-bar-pdb-app   8/8     8            8           6s
hello-baz-app       0/8     8            0           3m35s
hello-foo-app       2/2     2            2           3m2s
hello-foo-pdb-app   80/80   80           80          103s
hello-foobar-app    42/90   90           42          18m

/> echo “balancing”

/> kubectl scale --replicas=80 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl scale --replicas=80 deployment/hello-baz-app
deployment.apps/hello-baz-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-app
deployment.apps/hello-bar-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-pdb-app
deployment.apps/hello-bar-pdb-app scaled
```

```
/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       4/4     4            4           4m24s
hello-bar-pdb-app   8/8     8            8           6s
hello-baz-app       0/8     8            0           3m35s
hello-foo-app       2/2     2            2           3m2s
hello-foo-pdb-app   80/80   80           80          103s
hello-foobar-app    42/90   90           42          18m

/> echo “balancing”

/> kubectl scale --replicas=80 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl scale --replicas=80 deployment/hello-baz-app
deployment.apps/hello-baz-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-app
deployment.apps/hello-bar-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-pdb-app
deployment.apps/hello-bar-pdb-app scaled

/> kubectl scale --replicas=80 deployment/hello-foo-app
deployment.apps/hello-foo-app scaled
```

```
/> kubectl scale --replicas=80 deployment/hello-foobar-app
deployment.apps/hello-foobar-app scaled

/> kubectl scale --replicas=80 deployment/hello-baz-app
deployment.apps/hello-baz-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-app
deployment.apps/hello-bar-app scaled

/> kubectl scale --replicas=80 deployment/hello-bar-pdb-app
deployment.apps/hello-bar-pdb-app scaled

/> kubectl scale --replicas=80 deployment/hello-foo-app
deployment.apps/hello-foo-app scaled/> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       17/80   80           17          8m48s
hello-bar-pdb-app   8/80    80           8           4m30s
hello-baz-app       0/80    80           0           7m59s
hello-foo-app       80/80   80           80          7m26s
hello-foo-pdb-app   80/80   80           80          6m7s
hello-foobar-app    0/80    80           0           21m

/> echo “everything scaled to 80 and we see that no of the low is available, and not all medium are either”

/> echo “medium hello-bar-pdb-app are well above its PodDisruptionBudget”
```


/> kubectl scale --replicas=108 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

```
/> kubectl scale --replicas=108 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       3/80      80           3           25m
hello-bar-pdb-app   4/80      80           4           20m
hello-baz-app       0/80      80           0           24m
hello-foo-app       80/80     80           80          23m
hello-foo-pdb-app   108/108   108          108         22m
hello-foobar-app    0/80      80           0           38m
```

```
/> kubectl scale --replicas=108 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       3/80      80           3           25m
hello-bar-pdb-app   4/80      80           4           20m
hello-baz-app       0/80      80           0           24m
hello-foo-app       80/80     80           80          23m
hello-foo-pdb-app   108/108   108          108         22m
hello-foobar-app    0/80      80           0           38m

/> kubectl scale --replicas=110 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=108 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       3/80      80           3           25m
hello-bar-pdb-app   4/80      80           4           20m
hello-baz-app       0/80      80           0           24m
hello-foo-app       80/80     80           80          23m
hello-foo-pdb-app   108/108   108          108         22m
hello-foobar-app    0/80      80           0           38m

/> kubectl scale --replicas=110 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       2/80      80           2           25m
hello-bar-pdb-app   4/80      80           4           21m
hello-baz-app       0/80      80           0           24m
hello-foo-app       80/80     80           80          24m
hello-foo-pdb-app   110/110   110          110         23m
hello-foobar-app    0/80      80           0           38m
```

```
/> kubectl scale --replicas=111 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=111 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       1/80      80           1           25m
hello-bar-pdb-app   4/80      80           4           21m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          24m
hello-foo-pdb-app   111/111   111          111         23m
hello-foobar-app    0/80      80           0           39m
```

```
/> kubectl scale --replicas=111 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       1/80      80           1           25m
hello-bar-pdb-app   4/80      80           4           21m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          24m
hello-foo-pdb-app   111/111   111          111         23m
hello-foobar-app    0/80      80           0           39m

/> kubectl scale --replicas=112 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=111 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       1/80      80           1           25m
hello-bar-pdb-app   4/80      80           4           21m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          24m
hello-foo-pdb-app   111/111   111          111         23m
hello-foobar-app    0/80      80           0           39m

/> kubectl scale --replicas=112 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       1/80      80           1           26m
hello-bar-pdb-app   4/80      80           4           21m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          24m
hello-foo-pdb-app   112/112   112          112         23m
hello-foobar-app    0/80      80           0           39m
```

```
/> kubectl scale --replicas=114 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=114 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           26m
hello-bar-pdb-app   3/80      80           3           22m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          25m
hello-foo-pdb-app   114/114   114          114         24m
hello-foobar-app    0/80      80           0           39m
```

```
/> kubectl scale --replicas=114 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           26m
hello-bar-pdb-app   3/80      80           3           22m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          25m
hello-foo-pdb-app   114/114   114          114         24m
hello-foobar-app    0/80      80           0           39m

/> kubectl scale --replicas=115 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=114 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           26m
hello-bar-pdb-app   3/80      80           3           22m
hello-baz-app       0/80      80           0           25m
hello-foo-app       80/80     80           80          25m
hello-foo-pdb-app   114/114   114          114         24m
hello-foobar-app    0/80      80           0           39m

/> kubectl scale --replicas=115 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           26m
hello-bar-pdb-app   3/80      80           3           22m
hello-baz-app       0/80      80           0           26m
hello-foo-app       80/80     80           80          25m
hello-foo-pdb-app   115/115   115          115         24m
hello-foobar-app    0/80      80           0           40m
```

```
/> kubectl scale --replicas=117 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=117 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           52m
hello-bar-pdb-app   1/80      80           1           47m
hello-baz-app       0/80      80           0           51m
hello-foo-app       80/80     80           80          50m
hello-foo-pdb-app   117/117   117          117         49m
hello-foobar-app    0/80      80           0           65m
```

```
/> kubectl scale --replicas=117 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           52m
hello-bar-pdb-app   1/80      80           1           47m
hello-baz-app       0/80      80           0           51m
hello-foo-app       80/80     80           80          50m
hello-foo-pdb-app   117/117   117          117         49m
hello-foobar-app    0/80      80           0           65m

/> kubectl scale --replicas=119 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled
```

```
/> kubectl scale --replicas=117 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           52m
hello-bar-pdb-app   1/80      80           1           47m
hello-baz-app       0/80      80           0           51m
hello-foo-app       80/80     80           80          50m
hello-foo-pdb-app   117/117   117          117         49m
hello-foobar-app    0/80      80           0           65m

/> kubectl scale --replicas=119 deployment/hello-foo-pdb-app
deployment.apps/hello-foo-pdb-app scaled

/> kubectl get deployments
NAME                READY     UP-TO-DATE   AVAILABLE   AGE
hello-bar-app       0/80      80           0           53m
hello-bar-pdb-app   0/80      80           0           49m
hello-baz-app       0/80      80           0           52m
hello-foo-app       80/80     80           80          52m
hello-foo-pdb-app   119/119   119          119         50m
hello-foobar-app    0/80      80           0           66m
```

What are our priorities?

```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```

```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```

Kubernetes Standard

```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```

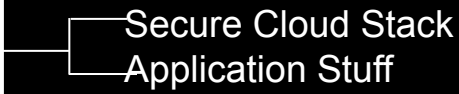
```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```

Platform Critical Stuff

```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```

Secure Cloud Stack Stuff

```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```



```
/> kubectl get pc -A -o json | jq '[ .items[] | { name: .metadata.name, priority: (.value|tonumber) }]  |
sort_by (.priority) | reverse' | jq -r ' .[] | .name + " " +  (.priority|tostring)'
system-node-critical 2000001000
system-cluster-critical 2000000000
calico-priority 1000000000
platform-system-critical 800000000
platform-critical 700000000
secure-cloud-stack-critical 600000000
secure-cloud-stack-technical-operations-critical 500000000
secure-cloud-stack-technical-management-critical 400000000
secure-cloud-stack-tenant-namespace-application-operations-critical 90000
secure-cloud-stack-tenant-namespace-application-operation-non-critical 80000
secure-cloud-stack-tenant-namespace-application-critical 60100
secure-cloud-stack-tenant-namespace-application-less-critical 60050
secure-cloud-stack-tenant-namespace-application-lesser-critical 60020
secure-cloud-stack-non-critical 60001
secure-cloud-stack-tenant-namespace-application-non-critical 60000
```

Least Important Stuff

Limitations?

Controlling Opportunities and Configurations

You can control the use of the prioritisation(s) in a number of ways:

- OPA Gatekeeper, which we use, because it fits very nicely with our current needs

Check the Node shutdown situation

- `shutdownGracePeriod`
- `shutdownGracePeriodCriticalPods`

`kubelet configuration` example

```yaml
shutdownGracePeriodByPodPriority:
  - priority: 100000
    shutdownGracePeriodSeconds: 10
  - priority: 10000
    shutdownGracePeriodSeconds: 180
  - priority: 1000
    shutdownGracePeriodSeconds: 120
  - priority: 0
    shutdownGracePeriodSeconds: 60
```

and more…

Additional Information

We only touched the surface in this area and only demonstrated this for cpu

If you want to see more about ResourceQuota(s), there is a lot of information available here:

https://kubernetes.io/docs/concepts/policy/resource-quotas/

https://kubernetes.io/docs/concepts/scheduling-eviction/pod-priority-preemption/

https://kubernetes.io/docs/concepts/scheduling-eviction/node-pressure-eviction/

https://kubernetes.io/docs/concepts/architecture/nodes/

https://kubernetes.io/docs/reference/config-api/kubelet-config.v1beta1/

and much more…

What else?

Non-Preemptive Priorities

Probably a separate talk

I hope it was useful for you?

THANK YOU

QUESTIONS?

a Printable version is located in github together with a workshop, where you can work on this if you find this interesting.
https://github.com/neticdk/k8s-workshop/simple-kubernetes-with-priority

a Printable version is located in github together with a workshop, where you can work on this if you find this interesting.
https://github.com/neticdk/k8s-workshop/simple-kubernetes-with-priority