



{KODE} {LOUD}

# Course Objectives

## Scheduling

Manual Scheduling

daemon Sets

Configure Kubernetes Scheduler

Labels & Selectors

Multiple Schedulers

Resource Limits

Scheduler Events

## Logging Monitoring

## Application Lifecycle Management

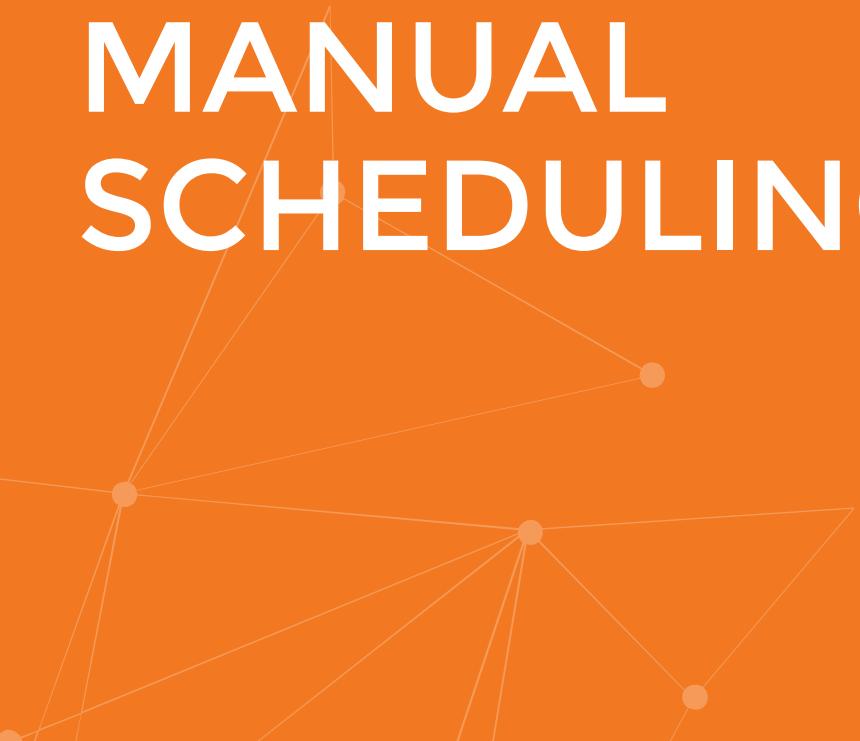
## Cluster Maintenance

## Security

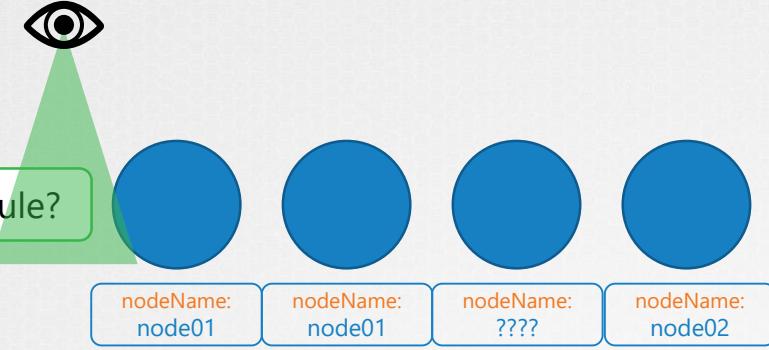
## Storage

## Troubleshooting

# MANUAL SCHEDULING



# | How scheduling works



Which node to schedule?

scheduler looks for Pods that have the property nodeName not set. These are candidates for scheduling

(Schedule)Bind Pod to Node

nodeName:  
node02

pod-definition.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
  ports:
  - containerPort: 8080
    nodeName: node02
```

# No Scheduler!

```
▶ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	0/1	Pending	0	3s

```
▶ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
nginx	1/1	Running	0	9s	10.40.0.4	node02

```
pod-definition.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 8080
      nodeName: node02
```

# | No Scheduler!

Pod-bind-definition.yaml

```
apiVersion: v1
kind: Binding
metadata:
  name: nginx
target:
  apiVersion: v1
  kind: Node
  name: {"apiVersion": "v1", "kind": "Node", "name": "node02"}
```

pod-definition.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 8080
  nodeName: node02
```

```
▶ curl --header "Content-Type:application/json" --request POST --data
http://$SERVER/api/v1/namespaces/default/pods/$PODNAME/binding/
```



{KODE} {LOUD}

# Course Objectives

## Scheduling

- Manual Scheduling
- Labels & Selectors
- Resource Limits
- Daemon Sets
- MultipleSchedulers
- Scheduler Events
- Configure Kubernetes Scheduler

## Logging Monitoring

## Application Lifecycle Management

## Cluster Maintenance

## Security

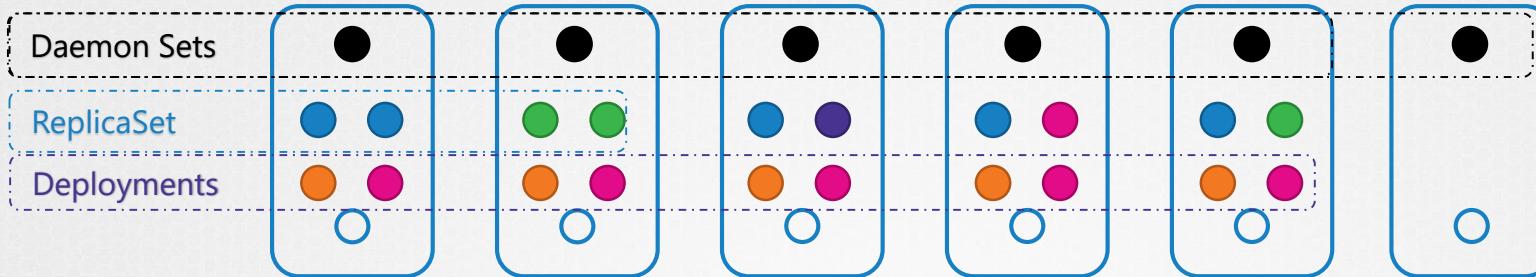
## Storage

## Troubleshooting

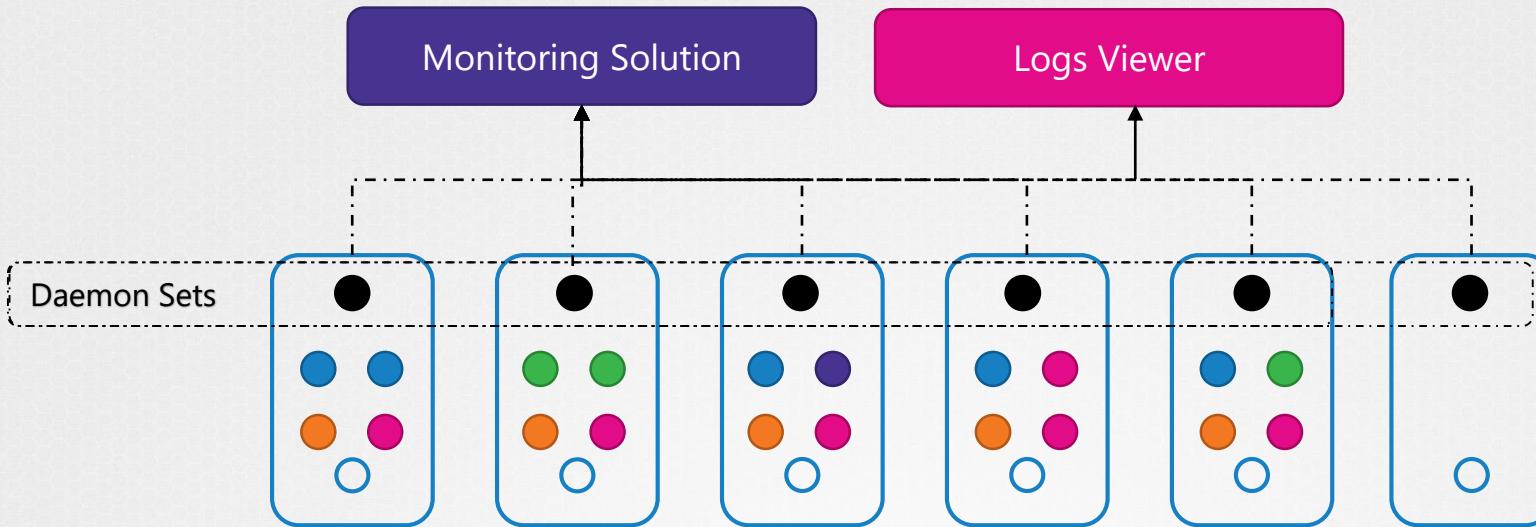
# Daemon Sets



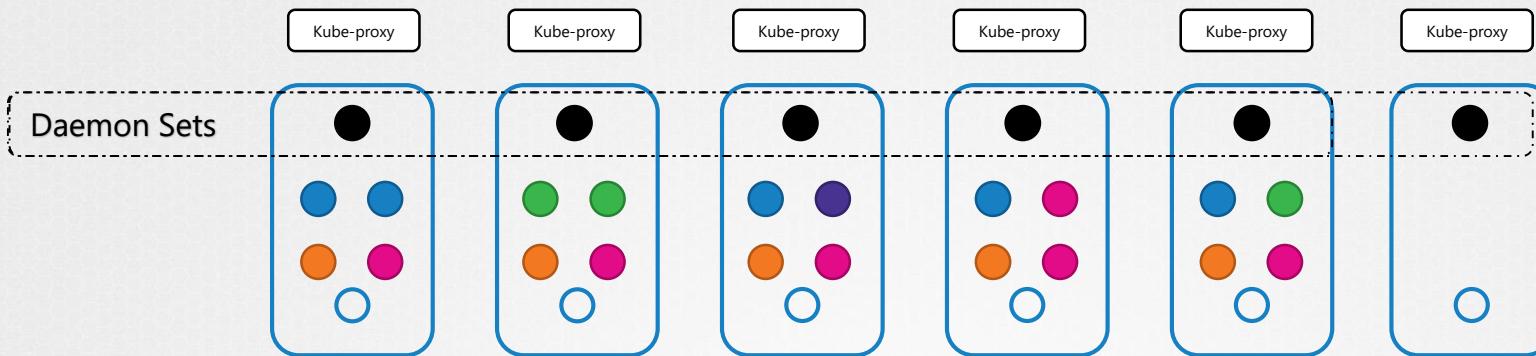
# Daemon Sets



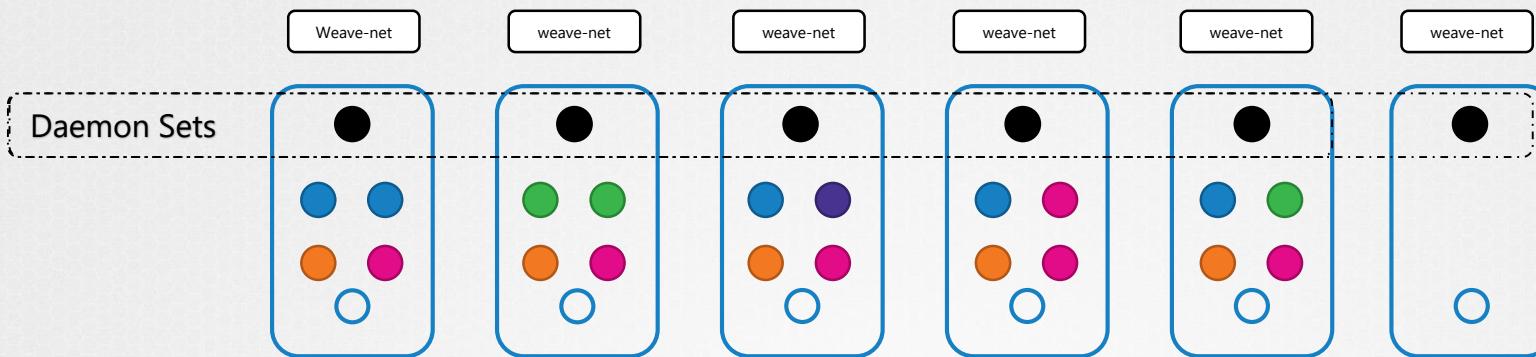
# Daemon Sets – UseCase



# Daemon Sets - UseCase - kube-proxy



# Daemon Sets - UseCase - Networking



# ■ DaemonSet Definition

daemon-set-definition.yaml

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: monitoring-daemon
spec:
  selector:
    matchLabels:
      app: monitoring-agent
  template:
    metadata:
      labels:
        app: monitoring-agent
  spec:
    containers:
    - name: monitoring-agent
      image: monitoring-agent
```

replicaset-definition.yaml

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: monitoring-daemon
spec:
  selector:
    matchLabels:
      app: monitoring-agent
  template:
    metadata:
      labels:
        app: monitoring-agent
  spec:
    containers:
    - name: monitoring-agent
      image: monitoring-agent
```

▶ kubectl create -f daemon-set-definition.yaml

daemon-set Created

# | View DaemonSets

```
▶ kubectl get daemonsets
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	AGE
monitoring-daemon	1	1	1	1	1	41

```
▶ kubectl describe daemonsets monitoring-daemon
```

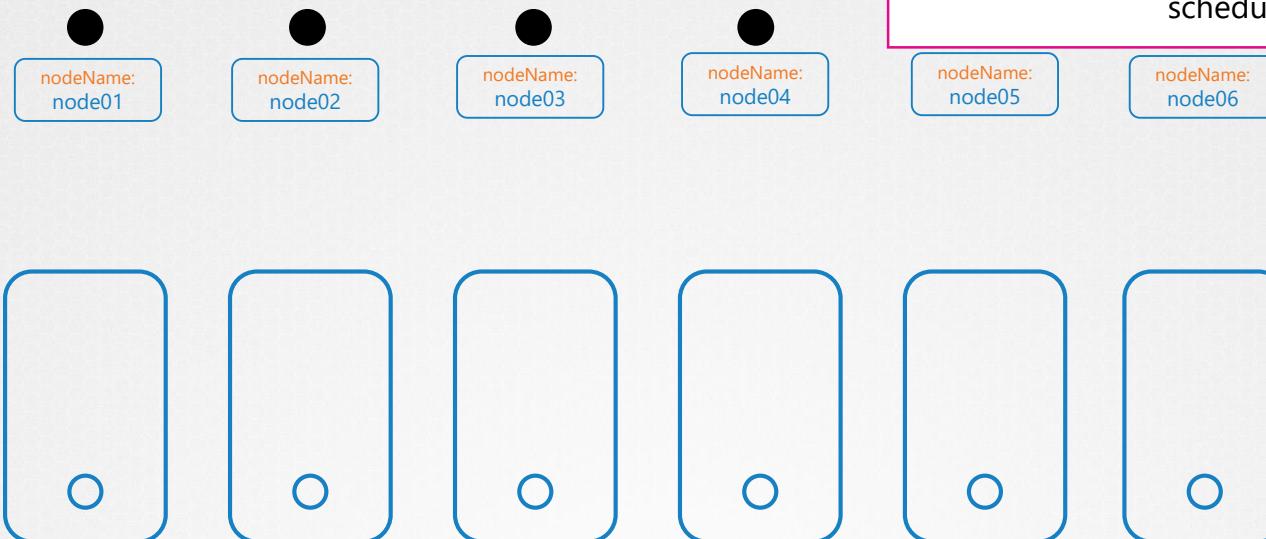
```
Name:           monitoring-daemon
Selector:       name=monitoring-daemon
Node-Selector:  <none>
Labels:         name=monitoring-daemon
Desired Number of Nodes Scheduled: 2
Current Number of Nodes Scheduled: 2
Number of Nodes Scheduled with Up-to-date Pods: 2
Number of Nodes Scheduled with Available Pods: 1
Number of Nodes Misscheduled: 0
Pods Status:   2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:        app=monitoring-agent
  Containers:
```

# | How does it work?



Default Behavior till v1.12

From v1.12 - uses NodeAffinity and default scheduler





{KODE} {LOUD}

# Course Objectives

## Scheduling

- Manual Scheduling

- daemon Sets

- Configure Kubernetes Scheduler

- Labels & Selectors

- Multiple Schedulers

- Resource Limits

- Scheduler Events

## Logging Monitoring

## Application Lifecycle Management

## Cluster Maintenance

## Security

## Storage

## Troubleshooting

# MULTIPLE SCHEDULERS





## Master

Manage, Plan, Schedule, Monitor  
Nodes



## Worker Nodes

Host Application as Containers



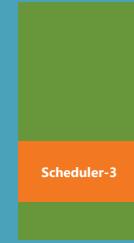
Scheduler-3



Scheduler-2



Kube-Scheduler



# | Deploy Additional Scheduler

```
▶ wget https://storage.googleapis.com/kubernetes-release/release/v1.12.0/bin/linux/amd64/kube-scheduler
```

```
kube-scheduler.service
```

```
ExecStart=/usr/local/bin/kube-scheduler \
--config=/etc/kubernetes/config/kube-scheduler.yaml \
--scheduler-name= default-scheduler
```

```
my-custom-scheduler.service
```

```
ExecStart=/usr/local/bin/kube-scheduler \
--config=/etc/kubernetes/config/kube-scheduler.yaml \
--scheduler-name= my-custom-scheduler
```

# | Deploy Additional Scheduler - kubeadm

/etc/kubernetes/manifests/kube-scheduler.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: kube-scheduler
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-scheduler
    - --address=127.0.0.1
    - --kubeconfig=/etc/kubernetes/scheduler.conf
    - --leader-elect=true
    image: k8s.gcr.io/kube-scheduler-amd64:v1.11.3
    name: kube-scheduler
```

my-custom-scheduler.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: my-custom-scheduler
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-scheduler
    - --address=127.0.0.1
    - --kubeconfig=/etc/kubernetes/scheduler.conf
    - --leader-elect=true
    image: k8s.gcr.io/kube-scheduler-amd64:v1.11.3
    name: my-custom-scheduler
```

# | View Schedulers

```
▶ kubectl get pods --namespace=kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
coredns-78fcdf6894-bk4ml	1/1	Running	0	1h
coredns-78fcdf6894-ppr6m	1/1	Running	0	1h
etcd-master	1/1	Running	0	1h
kube-apiserver-master	1/1	Running	0	1h
kube-controller-manager-master	1/1	Running	0	1h
kube-proxy-dbgvv	1/1	Running	0	1h
kube-proxy-fptbr	1/1	Running	0	1h
kube-scheduler-master	1/1	Running	0	1h
my-custom-scheduler	1/1	Running	0	9s
weave-net-4tfpt	2/2	Running	1	1h
weave-net-6j6zs	2/2	Running	1	1h

# | Use Custom Scheduler

```
▶ kubectl get pods --namespace=kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
coredns-78fcdf6894-bk4ml	1/1	Running	0	1h
coredns-78fcdf6894-ppr6m	1/1	Running	0	1h
etcd-master	1/1	Running	0	1h
kube-apiserver-master	1/1	Running	0	1h
kube-controller-manager-master	1/1	Running	0	1h
kube-proxy-dbgvv	1/1	Running	0	1h
kube-proxy-fptbr	1/1	Running	0	1h
kube-scheduler-master	1/1	Running	0	1h
my-custom-scheduler	1/1	Running	0	9s
weave-net-4tfpt	2/2	Running	1	1h
weave-net-6j6zs	2/2	Running	1	1h

```
pod-definition.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - image: nginx
      name: nginx
  schedulerName:
```

```
▶ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	0/1	Pending	0	6s



```
▶ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	6s



# View Events

```
kubectl get events
```

LAST SEEN	COUNT	NAME	KIND	TYPE	REASON	SOURCE	MESSAGE
9s	1	nginx.15	Pod	Normal	Scheduled	my-custom-scheduler	Successfully assigned default/nginx to node01
8s	1	nginx.15	Pod	Normal	Pulling	kubelet, node01	pulling image "nginx"
2s	1	nginx.15	Pod	Normal	Pulled	kubelet, node01	Successfully pulled image "nginx"
2s	1	nginx.15	Pod	Normal	Created	kubelet, node01	Created container
2s	1	nginx.15	Pod	Normal	Started	kubelet, node01	Started container

# | View Scheduler Logs

```
kubectl logs my-custom-scheduler --name-space=kube-system
```

```
I0204 09:42:25.819338      1 server.go:126] Version: v1.11.3
W0204 09:42:25.822720      1 authorization.go:47] Authorization is disabled
W0204 09:42:25.822745      1 authentication.go:55] Authentication is disabled
I0204 09:42:25.822801      1 insecure_serving.go:47] Serving healthz insecurely on 127.0.0.1:10251
I0204 09:45:14.725407      1 controller_utils.go:1025] Waiting for caches to sync for scheduler controller
I0204 09:45:14.825634      1 controller_utils.go:1032] Caches are synced for scheduler controller
I0204 09:45:14.825814      1 leaderelection.go:185] attempting to acquire leader lease  kube-system/my-custom-scheduler...
I0204 09:45:14.834953      1 leaderelection.go:194] successfully acquired lease kube-system/my-custom-scheduler
```



{KODE} {LOUD}

# Course Objectives

## Scheduling

Labels & Selectors

Resource Limits

Manual Scheduling

daemon Sets

Multiple Schedulers

Scheduler Events

Configure Kubernetes Scheduler

## Logging Monitoring

## Application Lifecycle Management

## Cluster Maintenance

## Security

## Storage

## Troubleshooting

# CONFIGURING SCHEDULER



# | Deploy Additional Scheduler

```
▶ wget https://storage.googleapis.com/kubernetes-release/release/v1.12.0/bin/linux/amd64/kube-scheduler
```

```
kube-scheduler.service
```

```
ExecStart=/usr/local/bin/kube-scheduler \
--config=/etc/kubernetes/config/kube-scheduler.yaml \
--scheduler-name= default-scheduler
```

```
my-custom-scheduler.service
```

```
ExecStart=/usr/local/bin/kube-scheduler \
--config=/etc/kubernetes/config/kube-scheduler.yaml \
--scheduler-name= my-custom-scheduler
```

# | Deploy Additional Scheduler - kubeadm

/etc/kubernetes/manifests/kube-scheduler.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: kube-scheduler
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-scheduler
    - --address=127.0.0.1
    - --kubeconfig=/etc/kubernetes/scheduler.conf
    - --leader-elect=true
    image: k8s.gcr.io/kube-scheduler-amd64:v1.11.3
    name: kube-scheduler
```

my-custom-scheduler.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: my-custom-scheduler
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-scheduler
    - --address=127.0.0.1
    - --kubeconfig=/etc/kubernetes/scheduler.conf
    - --leader-elect=true
    image: k8s.gcr.io/kube-scheduler-amd64:v1.11.3
    name: my-custom-scheduler
```



{KODE} {LOUD}

# Course Objectives

Scheduling

Logging Monitoring

Application Lifecycle Management

Cluster Maintenance

Security

Authentication & Authorization

Kubernetes Security

Network Policies

Storage

Troubleshooting

Secrets

TLS Certificates for Cluster Components

Images Securely

Security Contexts

Secure Persistent Key Value Store

# AUTHENTICATION

