

СЛЁРМ

+



Southbridge

Продвинутые абстракции Kubernetes


План

- DaemonSet
- StatefulSet
- Job/CronJob
- Role Based Access Control (RBAC)

Задача мониторинга


1. На каждой ноде (автоматически) запускается агент
2. Управляются агенты из одной точки
3. Конфигурируются так же из одной точки

Задача мониторинга

1. На каждой ноде (автоматически) запускается агент
 2. Управляются агенты из одной точки
 3. Конфигурируются так же из одной точки
- 


- Static pod
- PodAntiAffinity
<https://kubernetes.io/docs/tutorials/stateful-application/zookeeper/#tolerating-node-failure>
- DaemonSet

Задача мониторинга

1. На каждой ноде (автоматически) запускается агент
 2. Управляются агенты из одной точки
 3. Конфигурируются так же из одной точки
- 

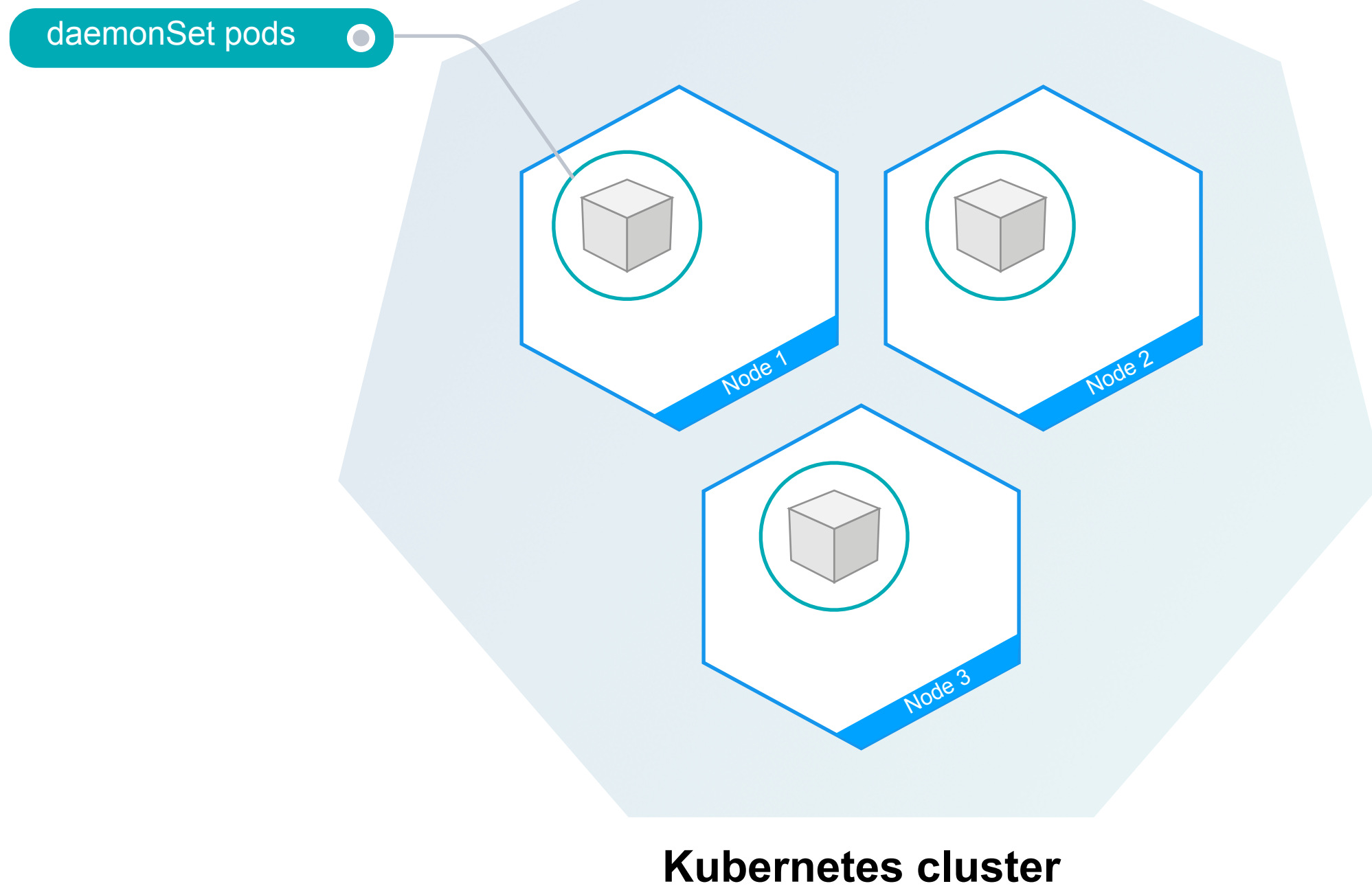
- Static pod (2)
- PodAntiAffinity
<https://kubernetes.io/docs/tutorials/stateful-application/zookeeper/#tolerating-node-failure>
- DaemonSet

Задача мониторинга

1. На каждой ноде (автоматически) запускается агент
 2. Управляются агенты из одной точки
 3. Конфигурируются так же из одной точки
- 

- Static pod (2)
- PodAntiAffinity (1)
<https://kubernetes.io/docs/tutorials/stateful-application/zookeeper/#tolerating-node-failure>
- DaemonSet

DaemonSet



DaemonSet

- Запускает поды на **всех** нодах кластера
- При добавлении ноды **добавляет** под / при удалении ноды GC **удаляет** под из кластера
- Описание практически полностью **соответствует** Deployment

Tolerations

<https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/>

```
kubectl get node master...
```

taints:

- effect: NoSchedule
key: node-role.kubernetes.io/master

```
cat daemonset.yaml
```

tolerations:

- effect: NoSchedule
key: node-role.kubernetes.io/master

Tolerations

Запустить pod демонсета на ingress ноде

StatefulSet

- Позволяет запускать **группу подов** (как Deployment)
 - Гарантирует их **уникальность**
 - Гарантирует их **последовательность**

StatefulSet

- Позволяет запускать **группу подов** (как Deployment)
 - Гарантирует их **уникальность**
 - Гарантирует их **последовательность**
- PVC **template**

StatefulSet

- Позволяет запускать **группу подов** (как Deployment)
 - Гарантирует их **уникальность**
 - Гарантирует их **последовательность**
- PVC **template**
- Используется для запуска приложений с **сохранением состояния**
 - RabbitMQ
 - DBs
 - Redis
 - Kafka
 - ...

StatefulSet

- Позволяет запускать **группу подов** (как Deployment)
 - Гарантирует их **уникальность**
 - Гарантирует их **последовательность**
- PVC **template**
- Используется для запуска приложений с **сохранением состояния**
 - RabbitMQ
 - DBs
 - Redis
 - Kafka
 - ...
- При **удалении** не удаляет PVC

Affinity

<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-affinity>

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: kubernetes.io/e2e-az-name

- operator: In

- values:

- e2e-az1

- e2e-az2

Affinity

<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-affinity>

affinity:

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 1

preference:

matchExpressions:

- key: another-node-label-key

operator: Exists

Affinity

Требовать запуск каждого рода стэйтфулсета на отдельной ноде

Headless Service

- .spec.clusterIP: None
- Резолвится в IP **всех** Endpoints
- Создает **зону** с именами всех Endpoints

Job

- Создает под для выполнения задачи
- Перезапускает поды до успешного завершения задачи или истечения таймаутов
 - `activeDeadlineSeconds`
 - `backoffLimit`

CronJob

- Создает job по расписанию
- **Работает так себе**

RBAC

- Role
- ClusterRole
- RoleBinding
- ClusterRoleBinding
- ServiceAccount

RBAC (на практике)

- `cd practice/advanced-abstractions/rbac`
- `kubectl apply -f ./`
- `kubectl get service --as=system:serviceaccount:default:user`
- `kubectl delete service --as=system:serviceaccount:default:user
kubernetes`

СЛЁРМ

+



Southbridge

Перерыв