

法律声明

□ 本课件包括：演示文稿，示例，代码，题库，视频和声音等，讲师及小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

□ 课程详情请咨询

■ 微信公众号：小象

■ 新浪微博：ChinaHadoop



Kubernetes应用管理-下



目录

1. Job详解
2. Volume详解
3. NameSpace机制
4. 调度机制入门

1. Job详解

- ❑ Job负责批处理任务，仅执行一次
- ❑ 目前的Serverless平台实现大多构筑在Job上面
- ❑ K8s的Job系统目前还比较原始，没有实现大部分批处理系统有的DAG，[相关issue](#)

1. Job Demo

apiVersion: batch/v1

kind: Job

metadata:

name: pi

spec:

template:

metadata:

name: pi

spec:

containers:

- name: pi

image: perl

command: ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"]

restartPolicy: Never

1.CronJob

- ❑ 在给定时间点只运行一次
- ❑ 周期性地在给定时间点运行
- ❑ 使用语法类似于crontab

1.CronJob Demo

apiVersion: batch/v2alpha1

kind: CronJob

metadata:

name: hello

spec:

schedule: "*/1 * * * *"

jobTemplate:

spec:

template:

spec:

containers:

- name: hello

image: busybox

args:

- /bin/sh

- -c

- date; echo Hello from the Kubernetes cluster

restartPolicy: OnFailure

1.扩展阅读

《瓜子云的任务调度系统》

- 把AirBnb的Airflow和k8s集成
- 没有漂亮的解决HA的问题——其实可以把整套Airflow搬上k8s

2. Volume

- 用于持久化存储
- \geq Pod 的生命周期
- 支持多种类型
- CSI 行业标准

2. Volume Demo

apiVersion: v1

kind: Pod

metadata:

name: test-pd

spec:

containers:

- image: k8s.gcr.io/test-webserver

name: test-container

volumeMounts:

- mountPath: /cache

name: cache-volume

volumes:

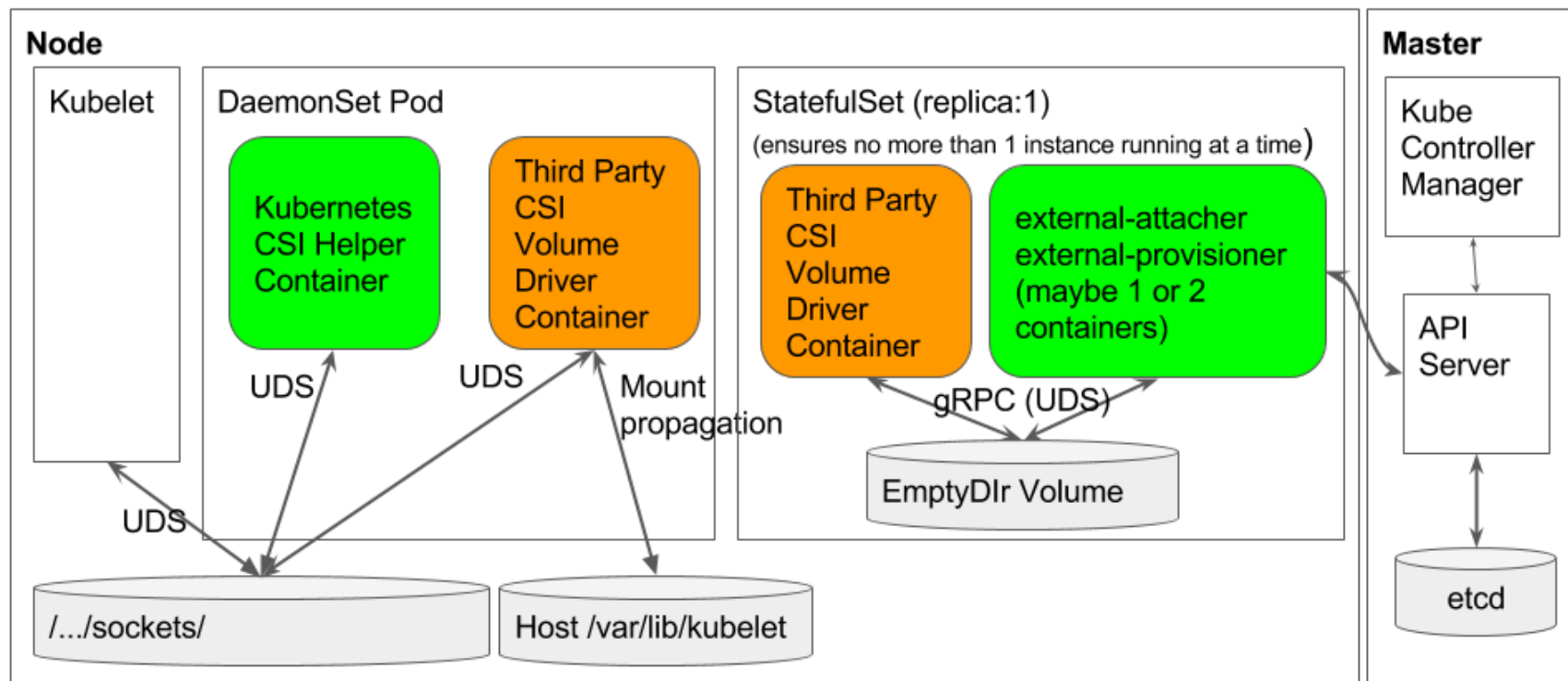
- name: cache-volume

emptyDir: {}

1. Volume -> volumeMounts对应
2. 通过名称对应
3. 1: n对应, 在多个容器里面同时挂载

emptyDir	hostPath
生命周期 = Pod	生命周期 = Node
不能指定路径	指定路径
同Pod的多容器间可共享	同Pod的多容器 同Node的多Pod

2.PV-PVC-CSI



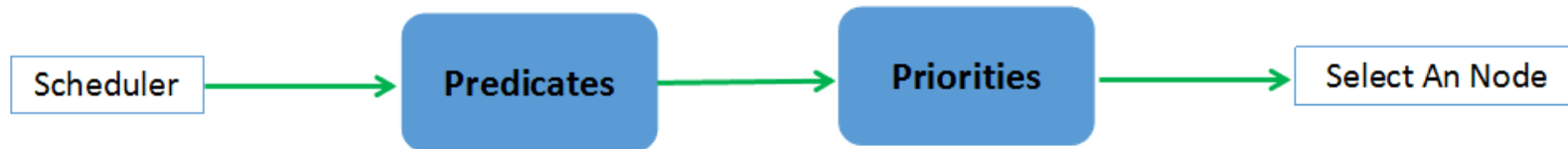
3.Namespace机制

- ❑ 为一种目前还不完全的隔离机制（例如node，PV不属于ns）
- ❑ 隔离k8s的Object
- ❑ 适用于多租场景、开发生产隔离等

3.Namespace Demo

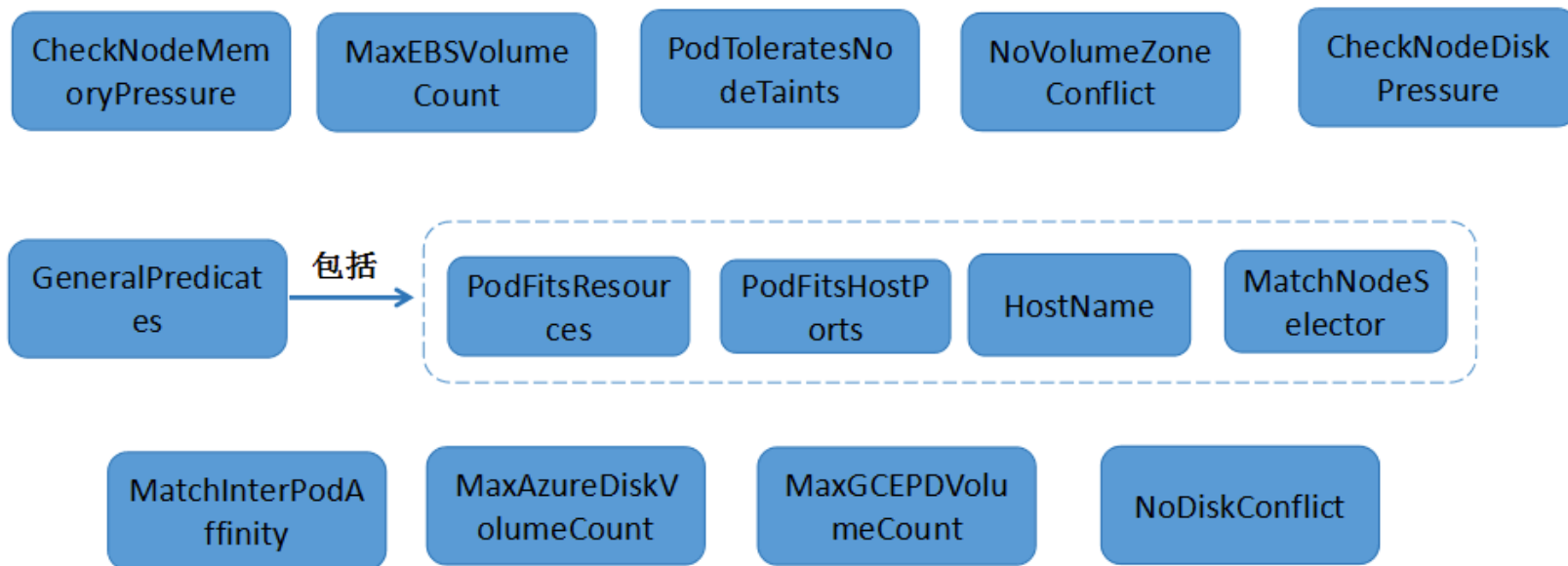
```
kubectl get ns
```

4.调度机制入门



1. 预选：根据配置的 Predicates Policies（默认为 DefaultProvider 中定义的 default predicates policies 集合）过滤掉那些不满足 Policies 的的Nodes，剩下的Nodes作为优选的输入。
2. 优选：根据配置的 Priorities Policies（默认为 DefaultProvider 中定义的 default priorities policies 集合）给预选后的Nodes进行打分排名，得分最高的Node即作为最适合的Node，该Pod就Bind到这个Node。

4. 预选



1. 系统当中有10个node，只有3台有GPU
2. 希望3个pod部署到这3个node上去
3. 当有GPU的机器扩容时，希望动态增加pod

答案：DaemonSet + MatchNodeSelector

4. 优选

SelectorSpreadP
riority

InterPodAffinity
Priority

LeastRequested
Priority

BalancedResour
ceAllocation

NodePreferAvoi
dPodsPriority

NodeAffinityPrio
rity

TaintTolerationP
riority

4.亲和/反亲和

亲和

传统应用进行容器化，拆分微服务之后的部署约束，需要按实例逐一配对就近部署，容器间通信就近路由，减少网络消耗。

反亲和

高可靠性考虑，同个应用的多个实例反亲和部署，减少宕机影响
互相干扰的应用反亲和部署，避免干扰。

<http://www.skycloudsoftware.com/index.php/2016/09/08/kubernetes.html>

作业

- 使用kubectI把上面的各种应用类型都玩一遍

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop

