CHINA OpenInfra Days

CHINA OpenInfra Days | IT大咖说 知识共享平台

# 基于 OpenStack、Kubernetes 构建一体化云平台

王博、王后明
EasyStack

# 内容提纲

- 三种融合场景
- 相关开源项目
- 生产最佳实践

# OpenStack 与 Kubernetes 融合的3种形态

- Kubernetes on OpenStack

用OpenStack 为容器生态提供更好的数据中心基础设施管理和云的能力。

- OpenStack on Kubernetes

用容器来解决 OpenStack 的部署和升级问题。

- OpenStack with Kubernetes

两个生态的组件同级别共存，优势互补。

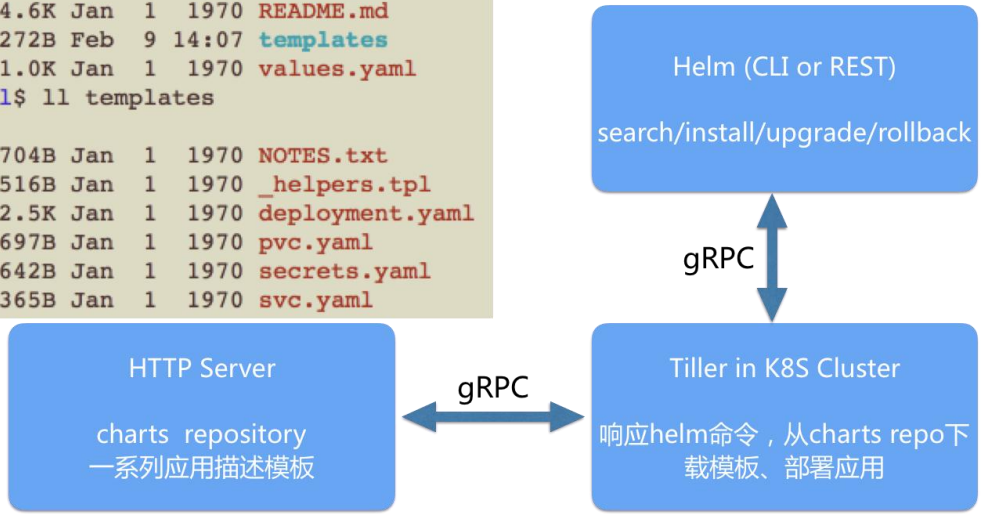
# Kolla/LOCI 介绍

- LOCI is a project designed to quickly build Lightweight OCI compatible images of OpenStack services.
- Kolla: Provide production-ready containers and deployment tools for operating OpenStack services.

# Helm/Charts 介绍

Kubernetes的应用包管理器，类似于Linux里的yum/apt；Helm为客户端、Tiller为服务端、Charts为应用模板描述文件。

```
[houming@rmbp ~/helm_charts/mysql$ ll
total 32
-rwxr-xr-x  1 houming  staff   288B Jan  1  1970 Chart.yaml
-rwxr-xr-x  1 houming  staff   4.6K Jan  1  1970 README.md
drwxr-xr-x  8 houming  staff   272B Feb  9 14:07 templates
-rwxr-xr-x  1 houming  staff   1.0K Jan  1  1970 values.yaml
[houming@rmbp ~/helm_charts/mysql$ ll templates
total 48
-rwxr-xr-x  1 houming  staff   704B Jan  1  1970 NOTES.txt
-rwxr-xr-x  1 houming  staff   516B Jan  1  1970 _helpers.tpl
-rwxr-xr-x  1 houming  staff   2.5K Jan  1  1970 deployment.yaml
-rwxr-xr-x  1 houming  staff   697B Jan  1  1970 pvc.yaml
-rwxr-xr-x  1 houming  staff   642B Jan  1  1970 secrets.yaml
-rwxr-xr-x  1 houming  staff   365B Jan  1  1970 svc.yaml
```



# OpenStack-Helm 介绍

- Helm charts for deploying OpenStack (including MariaDB、Ceph、RabbitMQ, etc) on Kubernetes; supporting both LOCI and Kolla containers

# 有状态应用高可用

- StatefulSet + PVC用来支撑有状态应用如MariaDB、MongoDB 等， 但有的应用需要持久化文件，多副本直接没有同步机制，需要以单副本运行。
- 需要使用支持多写的存储后端如 NFS；或者对单副本有状态应用进行特殊处理。

# CustomResourceDefinitions(CRD)以及自定义 Controller

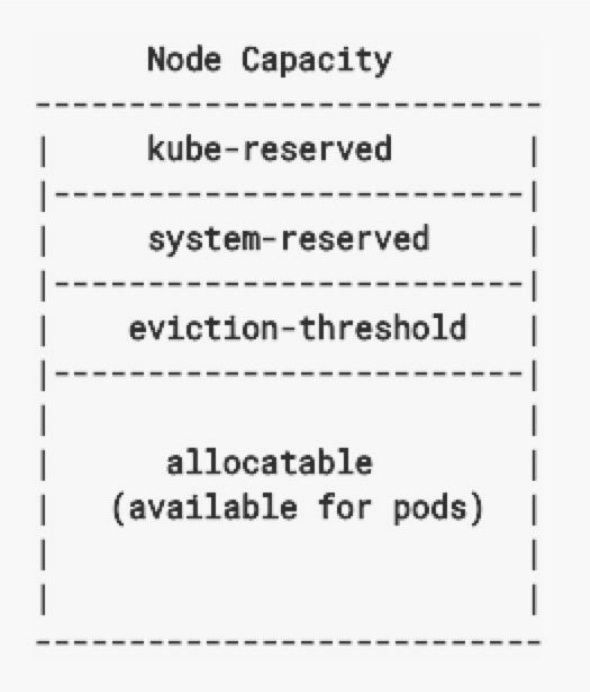
- Kubernetes 内置对象不一定能满足所有应用场景。
- 可利用 CRD + 自定义 controller + 自定义 apiserver来扩展和丰富 K8S功能。比如 Rook

# 运行时稳定性

- Docker-CE 是以时间点而非功能进行发布；Release 1年后 EOL。如何跟进？如何 backport 关键 bug？
- 运行在什么操作系统之上？
- 是否有其它可替换的稳定、生产级运行时？OCI runc、Containerd？

- Resource Management

- kube-reserved: 为kubelet，container runtime 预留资源
- system-reserved: 为系统daemons预留资源，如: sshd, systemd等
- eviction-threshold：当整个Node资源使用过高时，配置kubelet evicts Pods的阈值，以保证system资源可用。目前可以配的资源包括 memory 和 ephemeral-storage。
- allocatable：Node所有资源减去上面的配置，剩下的部分供Pods调度使用。



Scheduling of Pods with Requests and "Node Allocatable"

Kubernetes1.8开始，引入了Pod Priority 和 Preemption

Kubernetes < v1.9时， Eviction的顺序按照Pods的QoS Class

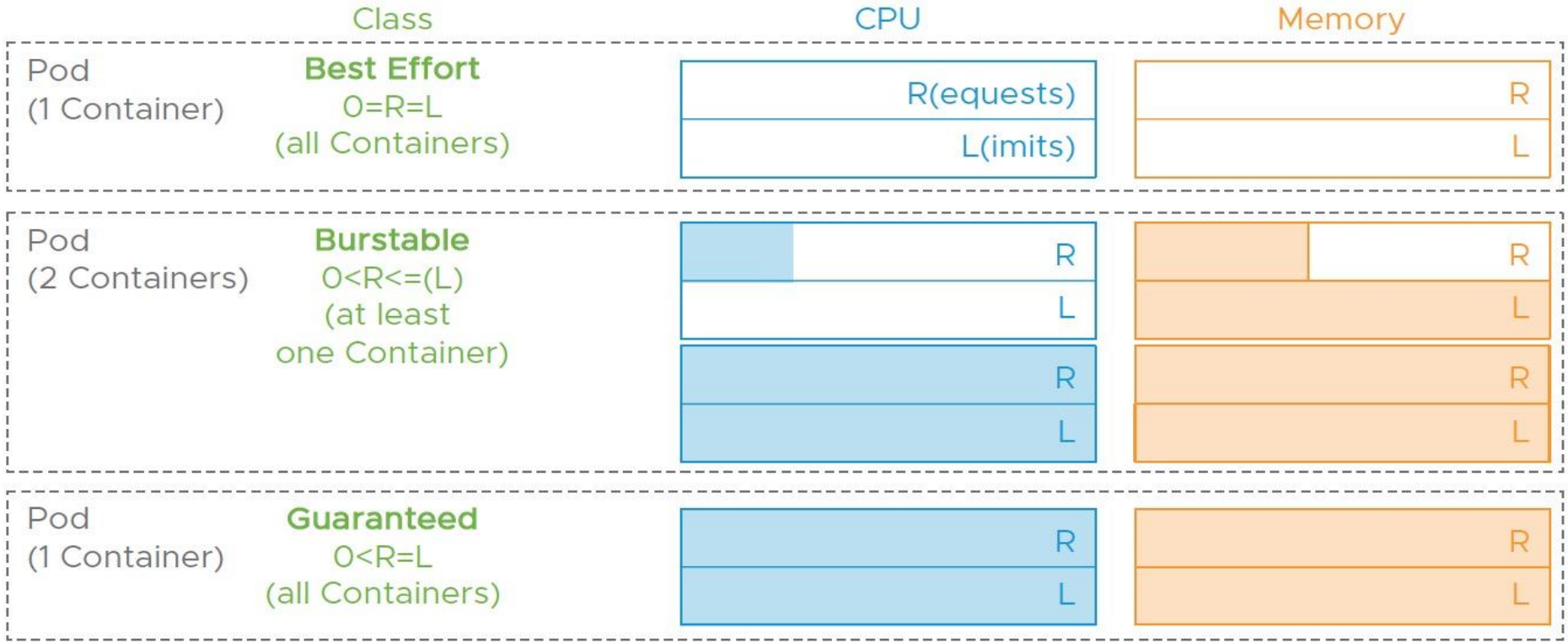
Best Effort -> Burstable -> Guaranteed， Best Effort先会被 evicted

Kubernetes>=v1.9， Eviction顺序更复杂一些

首先找到资源实际使用值超过requests值的Pods(usage > requests)

接着在这些Pods中，按Priority排序

最后计算资源实际使用值 减去 requests值的差值




QoS Examples

Enforcement:

CPU Requests 通过Cgroup中cpu 子系统设定cpu.shares值实现

CPU Limits 通过Cgroup中cpu子系统设定cpu.cfs_quota_us/ cpu.cfs_period_us的比值实现

Memory Limits通过Cgroup中memory子系统设定memory.limit_in_bytes的值实现

```
$ kubectl run nginx --image=nginx --limits cpu=1,memory=200Mi --restart=Never
```

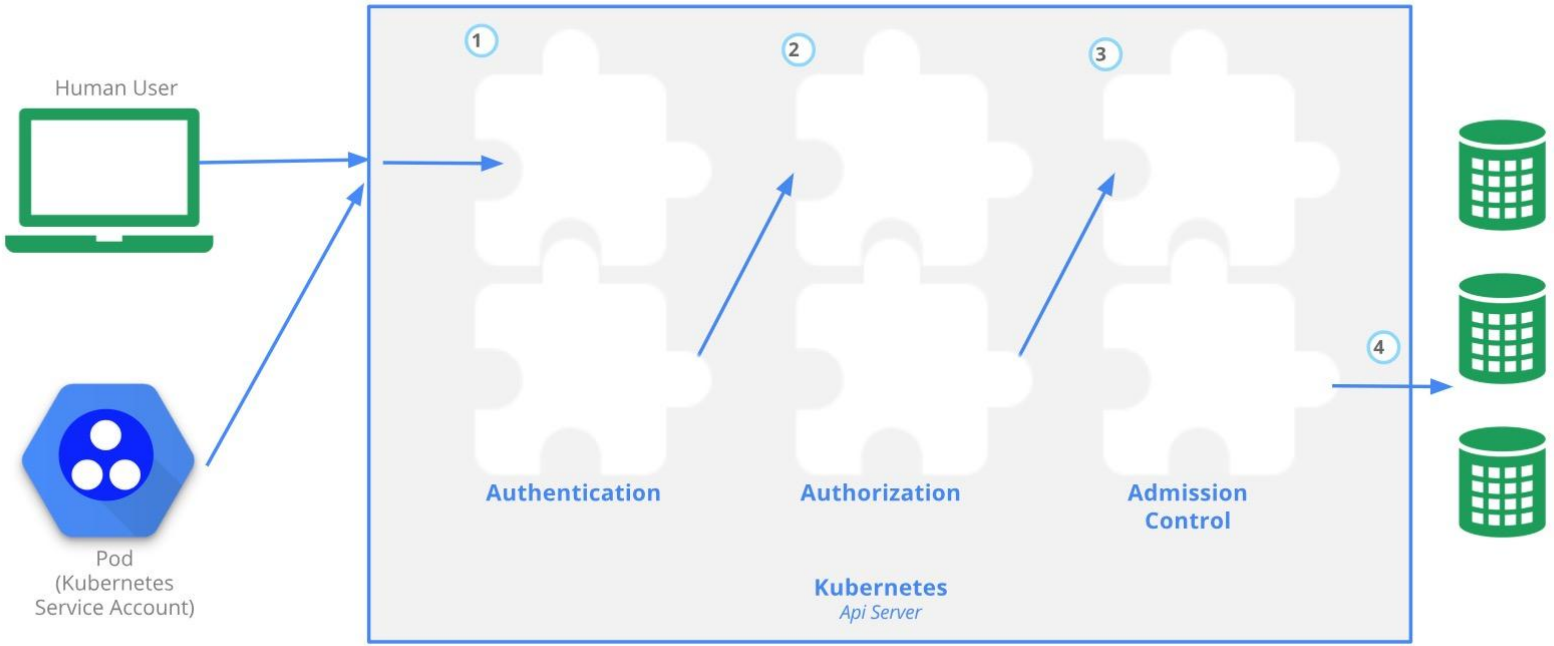
Kubelet View

```
$ cat /sys/fs/cgroup/cpu/kubepods/pod8d69938d-3e49-11e8-8690-000c29f521a3/cpu.{shares,cfs_*}
1024        # cpu.shares        (*relative* weight)
100000      # cpu.cfs_period_us (*absolute* enforcement interval in µs)   } cpu=1
100000      # cpu.cfs_quota_us  (*absolute* limit in µs)

$ cat /sys/fs/cgroup/memory/kubepods/pod8d69938d-3e49-11e8-8690-000c29f521a3/memory.limit_in_bytes
209715200   # *absolute* memory limit in bytes   } memory=200Mi
```

- keystone
- cinder
- manila

https://github.com/kubernetes/cloud-provider-openstack


Human User
Pod
(Kubernetes
Service Account)
1
2
3
4
Authentication
Authorization
Admission
Control
Kubernetes
Api Server

# Authentication:

- x509 client certs
- static password/token file
- webhook

# Authorization:

- RBAC
- webhook

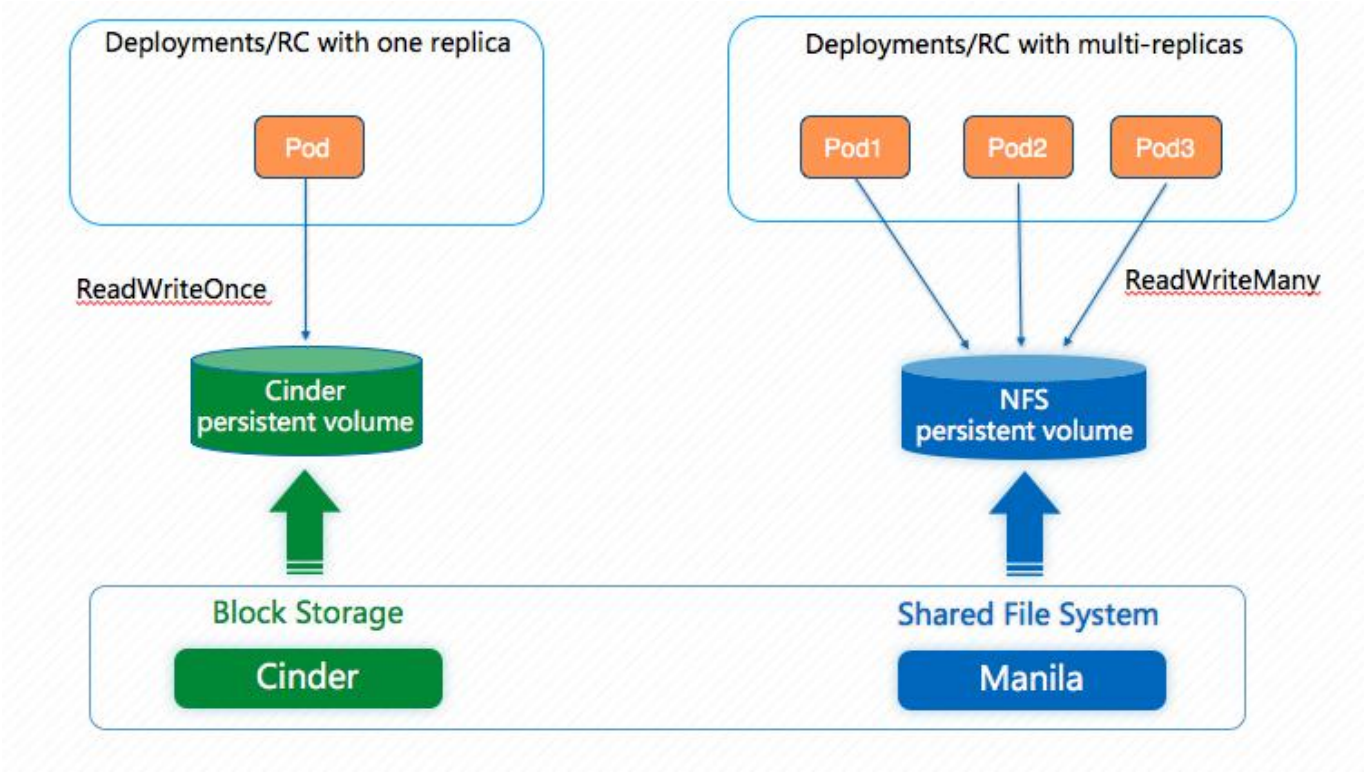
https://kubernetes.io/docs/reference/access-authn-authz/authentication/

部署keystone服务
created user, roles

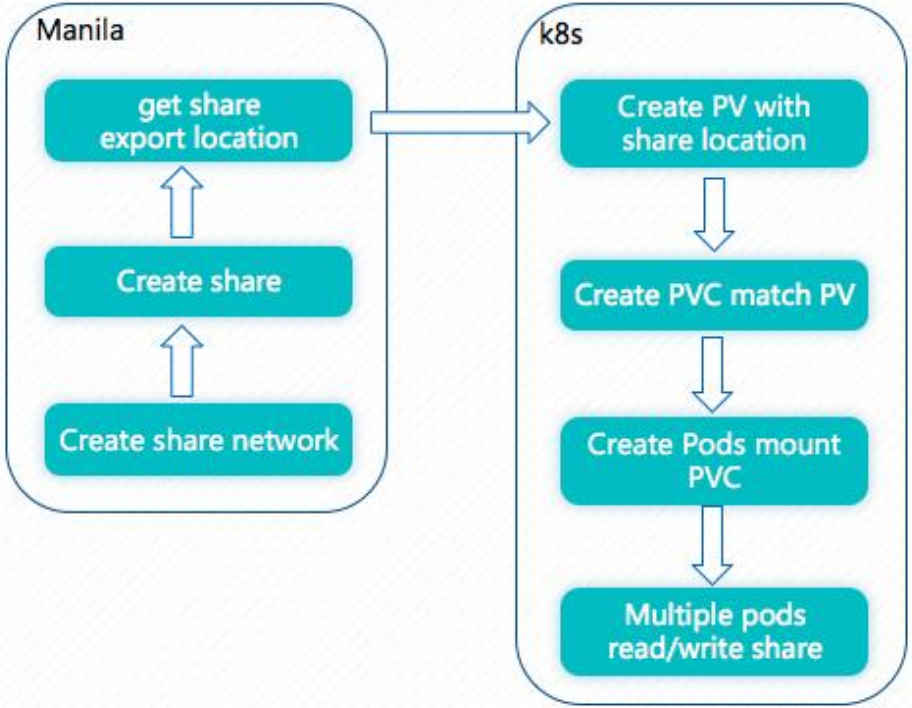
部署k8s-keystone-auth Pod：
keystone endpoints
webhook.json：projectA下面的 roleA角色的用户，可以在namespaceA下对哪些resources执行哪些verbs

配置k8s-apiserver：
--authentication-token-webhook-config-file
--authorization-webhook-config-file

http://superuser.openstack.org/articles/keystone-authentication-kubernetes-cluster/
http://superuser.openstack.org/articles/kubernetes-keystone-integration-test/


Deployments/RC with one replica
Pod
ReadWriteOnce
Cinder persistent volume
Deployments/RC with multi-replicas
Pod1
Pod2
Pod3
ReadWriteMany
NFS persistent volume
Block Storage
Cinder
Shared File System
Manila

# Manila 手动提供 nfs pvc 过程



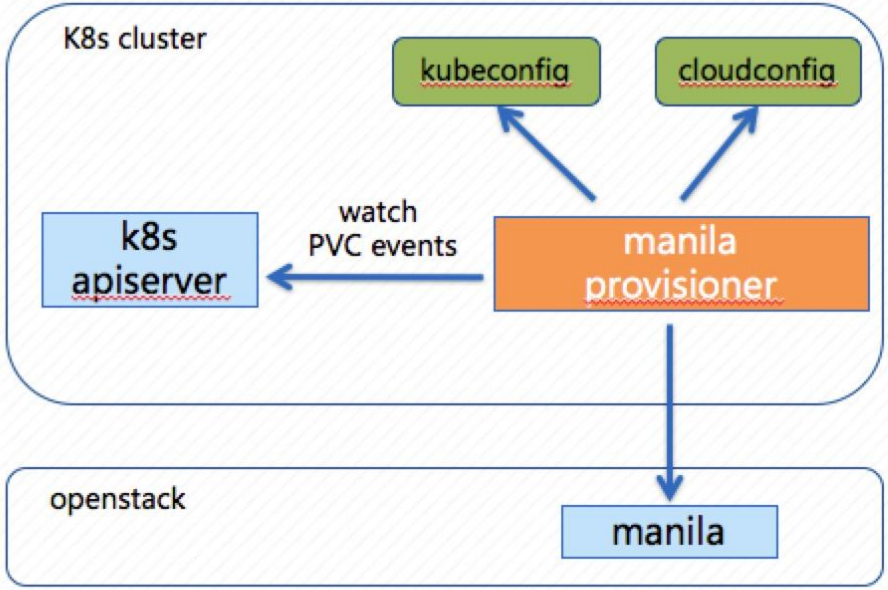

nfs-pv.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs
spec:
  capacity:
    storage: 100Mi
  accessModes:
    - ReadWriteMany
  nfs:
    server: 10.254.0.19
    path: "/shares/share-aef3f063-6209-4613-970e-e957bdde69c5"
```

nfs-pvc.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: ""
  resources:
    requests:
      storage: 100Mi
```

# Manila 动态提供 nfs pvc




manila storage class: (create during cluster initialization)

```
1  kind: StorageClass
2  apiVersion: storage.k8s.io/v1beta1
3  metadata:
4    name: manila-nfs
5  provisioner: openstack.org/manila
```

manila pvc: (create at any time dynamically)

```
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: es-manila-nfs
5  spec:
6    accessModes:
7      - ReadWriteMany
8    storageClassName: "manila-nfs"
9    resources:
10     requests:
11       storage: 100Mi
```

https://github.com/kubernetes-incubator/external-storage/pull/429
https://github.com/kubernetes/cloud-provider-openstack

CHINA
OpenInfra Days



# Thank You