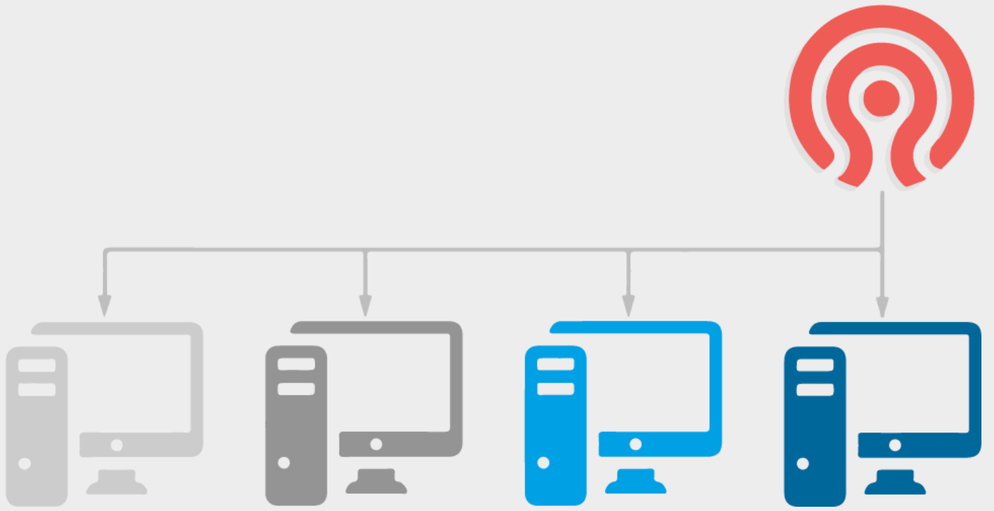
СЛЁРМ
+
Southbridge

Хранение данных на примере ceph

Установка Ceph

Сценарий https://github.com/ceph/ceph-ansible

```
git clone https://gitlab.slurm.io/slurm/ceph.git
 cd ceph
 pip install notario
 sh  _deploy_cluster.sh
```



Подключение к подам

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test
      name: test-volume
  volumes:
  - name: test-volume
    # This AWS EBS volume must already exist.
    awsElasticBlockStore:
      volumeID: <volume-id>
      fsType: ext4
```

Подключение к подам

Типы томов:

https://kubernetes.io/docs/concepts/storage/volumes/#types-of-volumes

- configMap
- emptyDir
- hostPath
- secret

Проблемы:

- тома надо создавать вручную
- параметры доступа прописывать для каждого тома, каждого пода
- чтобы поменять тип подключенного тома - надо менять манифесты

Подключение к подам SC/PVC/PV

```
volumes:
    - name: mypd
      persistentVolumeClaim:
        claimName: myclaim
```

Storage class: хранит параметры подключения

PersistentVolumeClaim: описывает требования к тому

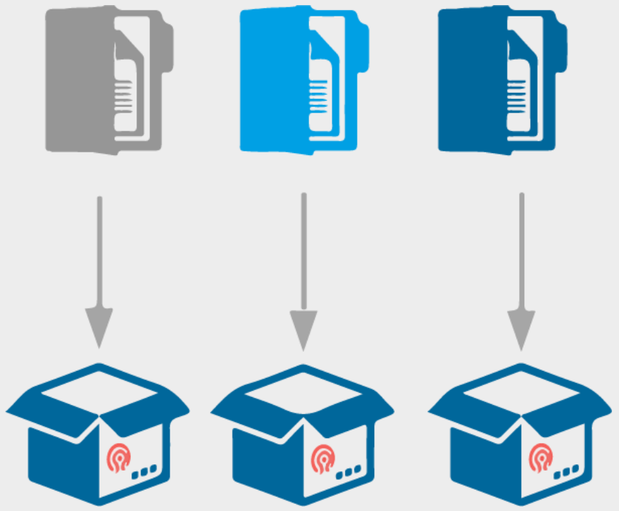
PersistentVolume: хранит параметры и статус тома

Provisioner: параметр SC, плагин создания томов

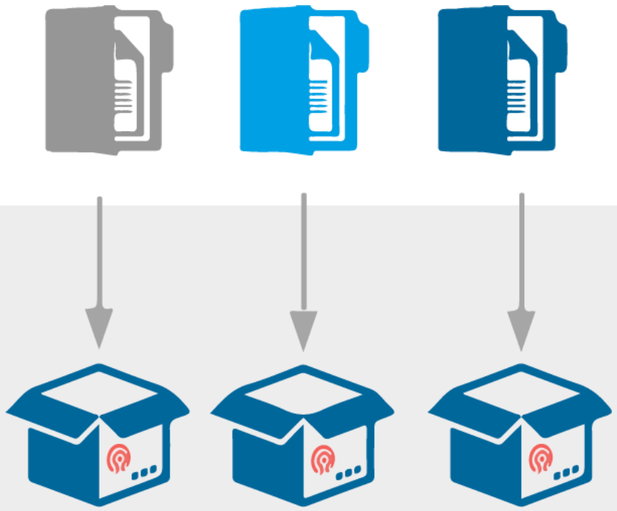
Создание пула для RBD

```
node-1# ceph osd pool create kube 32
node-1# ceph osd pool application enable kube kubernetes

node-1# ceph auth get-or-create client.user mon 'allow r, allow command "osd blacklist"'
osd 'allow rwx pool=kube'
```



Подключение RBD

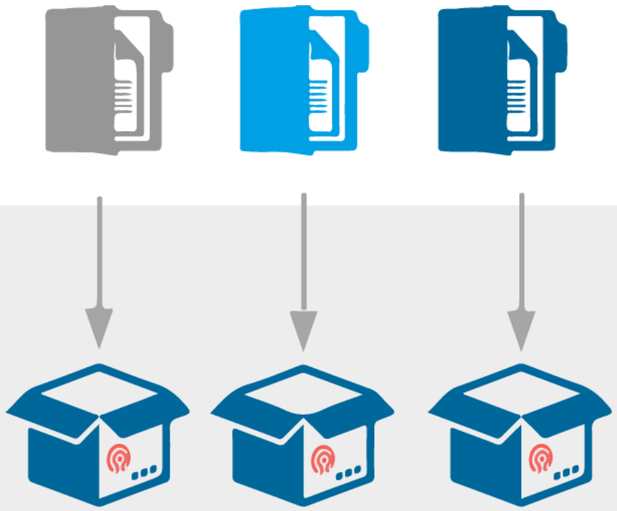


```
Admin key:
node-1# ceph auth get-key client.admin

master-1# kubectl create secret generic ceph-secret \
--type="kubernetes.io/rbd" \
--from-literal=key='AQAcU7JaU4NALBAyyyyyyyy==' \
--namespace=kube-system

Client key:
node-1# ceph auth get-key client.user
master1# kubectl create secret generic ceph-secret-user \
--type="kubernetes.io/rbd" \
--from-literal=key='AQAcU7Jaxxxxxxx==' \
--namespace=default
```

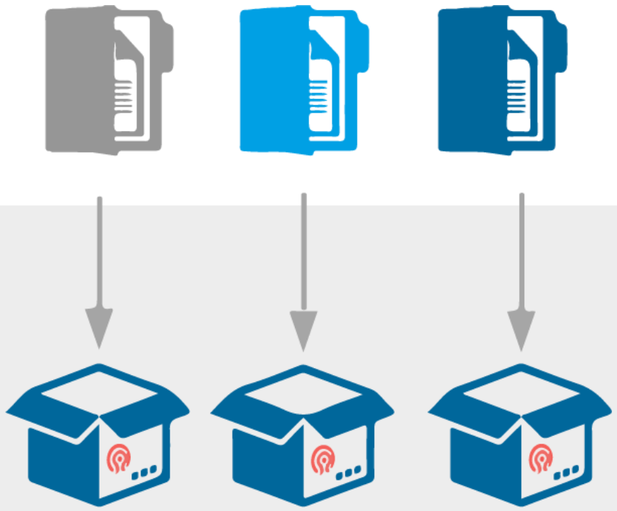
Подключение RBD



Storage class

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: kube
provisioner: kubernetes.io/rbd
parameters:
  monitors: <monitor-1-ip>:6789, <monitor-2-ip>:6789, <monitor-3-ip>:6789
  adminId: admin
  adminSecretName: ceph-secret
  adminSecretNamespace: "kube-system"
  pool: kube
  userId: user
  userSecretName: ceph-secret-user
```

Подключение RBD

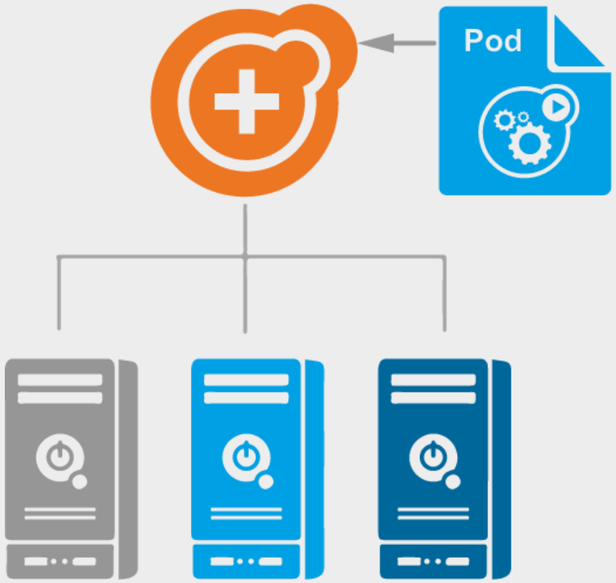


PVC

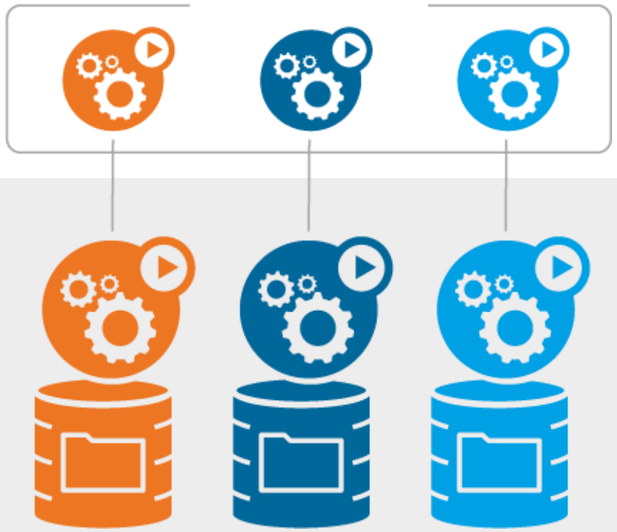
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: prometheus-3gb
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
  storageClassName: kube
```

Deployment

```
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    spec:
      containers:
      - volumeMounts:
        - name: data
          mountPath: /data
      volumes:
        - name: data
          persistentVolumeClaim:
            claimName: prometheus-3gb
```

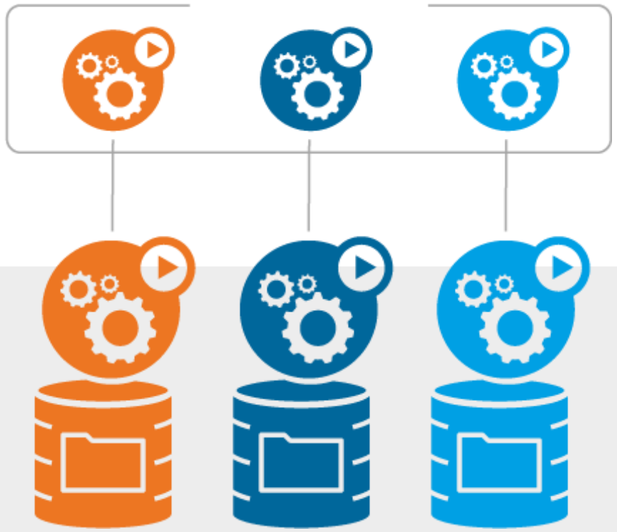


StatefulSet



```
apiVersion: apps/v1
kind: StatefulSet
spec:
  template:
   spec:
     containers:
       - volumeMounts:
        - name: mysql
          mountPath: /var/lib/mysql
  volumeClaimTemplates:
  - metadata:
      name: mysql
    spec:
      accessModes: [ "ReadWriteOnce" ]
      storageClassName: "rbd"
      resources:
        requests:
          storage: 1Gi
```

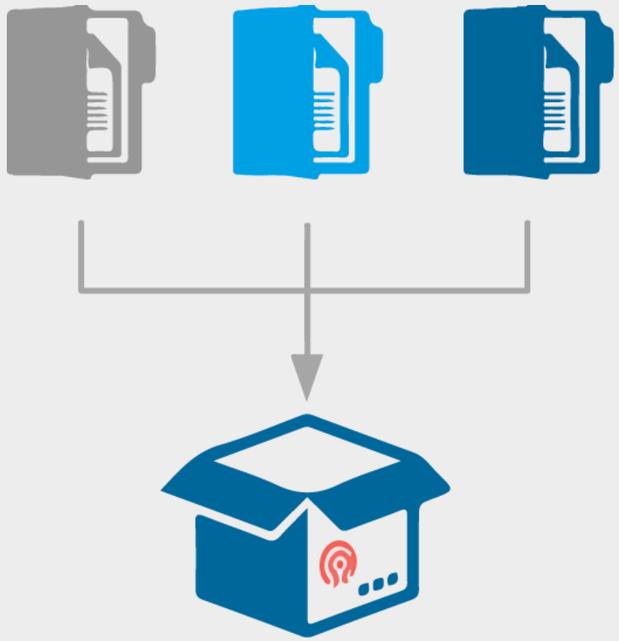
initContainers



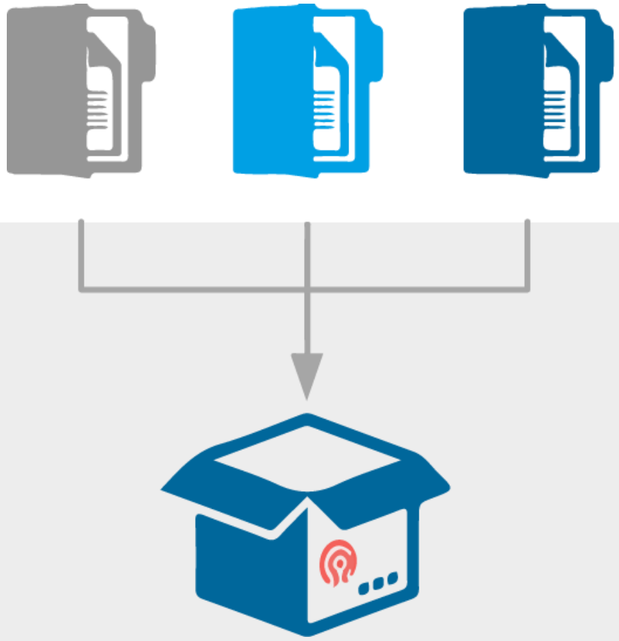
```
spec:
  template:
   spec:
     initContainers:
     - image: busybox
       command: ["sh", "-c", "chown -R 27:27 /var/lib/mysql"]
       volumeMounts:
         - name: mysql
           mountPath: /var/lib/mysql
     containers:
     - image: mysql
       volumeMounts:
         - name: mysql
           mountPath: /var/lib/mysql
```

Создание пула для CephFS

```
ceph osd pool create cephfs_data 32
ceph osd pool create cephfs_metadata 32
ceph fs new cephfs cephfs_metadata cephfs_data
ceph fs ls
```



Подключение CephFS



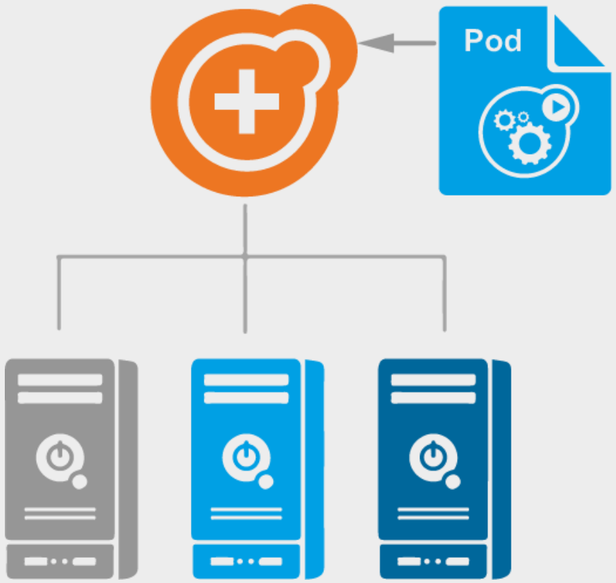
```
node-1# mkdir -p /mnt/cephfs
node-1# mount.ceph 172.21.0.6:/ /mnt/cephfs \
  -o name=admin, \
  secret=`ceph auth get-key client.admin`
node-1# mkdir -p /mnt/cephfs/data_path

node-1# ceph auth get-or-create client.username mon 'allow r' mds 'allow r, allow rw
path=/data_path' osd 'allow rw pool=cephfs_data'

master-1#
create secret  generic cephfs-secret-username \
  --from-literal=key='AQQxxxxxxxx' --namespace app
```

Deployment

```
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    spec:
      containers:
      - volumeMounts:
        - name: cephfs
          mountPath: /path
      volumes:
      - name: cephfs
        cephfs:
          monitors:
          - 172.21.0.5:6789
          path: /namespace/app_name
          user: username
          secretRef:
            name: cephfs-secret-username
```



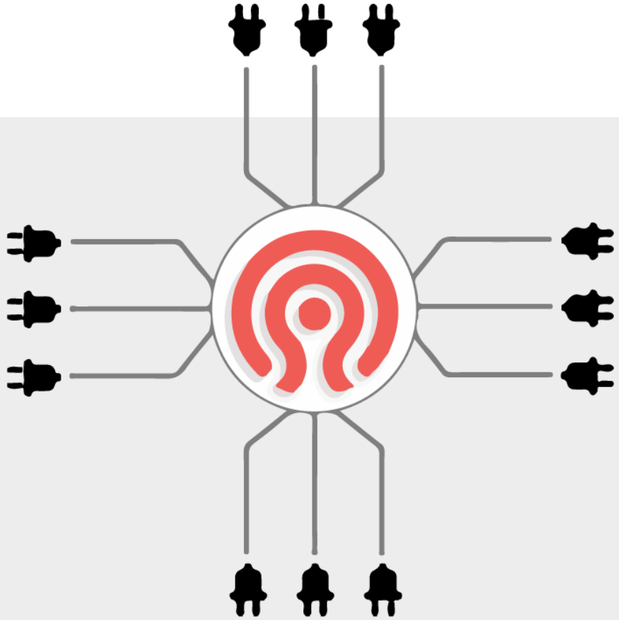
CephFS provisioner

```
helm repo add sb https://charts.southbridge.ru

helm inspect values sb/cephfs-provisioner \
   > values.yaml
vim values.yaml
<<<<<<<<<<<
   adminSecretName: ceph-secret-admin
   monitors:
       - 172.21.0.5:6789
       - 172.21.0.6:6789
       - 172.21.0.7:6789
<<<<<<<<<<<

kubectl create secret generic ceph-secret-admin \
   --namespace=kube-system --from-literal=secret=XX

helm upgrade --install cephfs-provisioner \
sb/cephfs-provisioner -f values.yaml --namespace=kube-system
```




СЛЁРМ
+
Southbridge
slurm.io