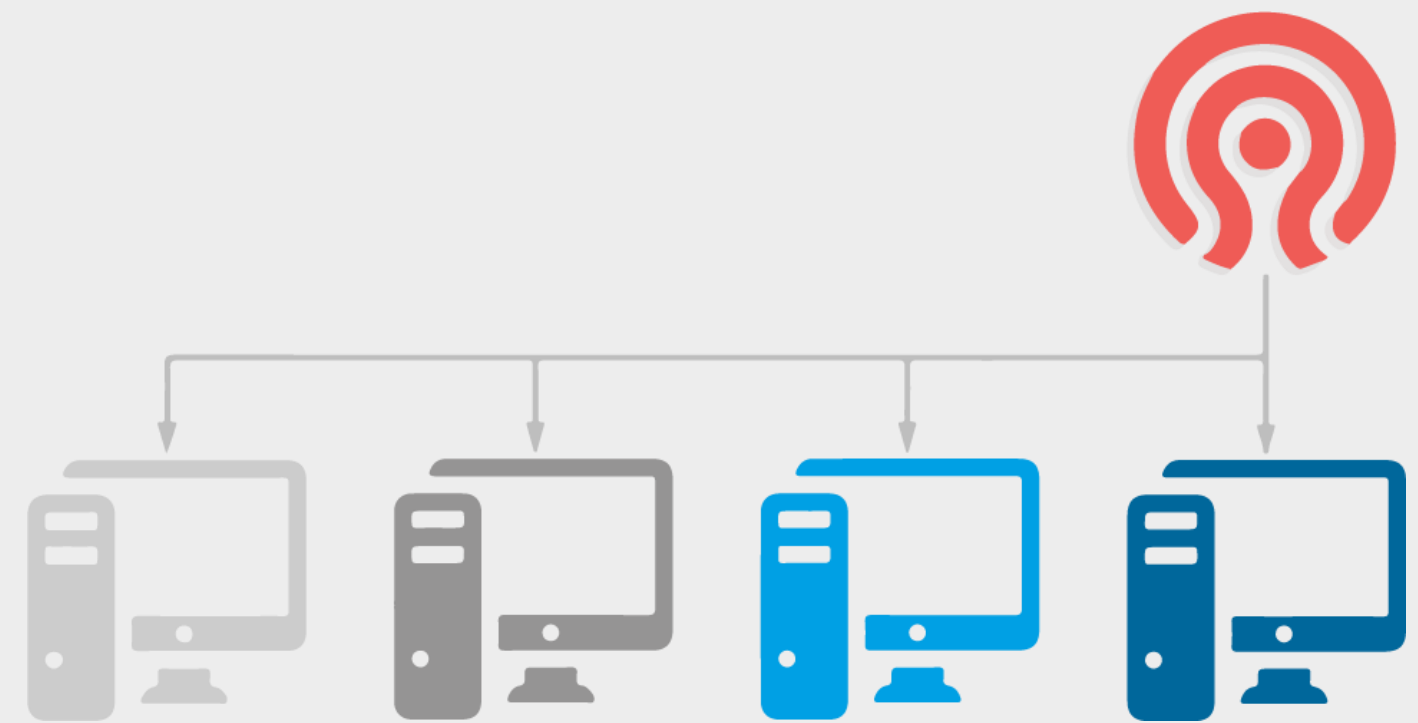


# Хранение данных на примере serph

# Установка Ceph

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

```
git clone git@gitlab.slurm.io:slurm/ceph-nautilus.git  
cd ceph-nautilus  
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
- просто устарело

# 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

# Container Storage Interface

Унифицированный интерфейс хранилищ

**Node plugin** – запущен на каждом узле

**Controller plugin** – взаимодействие с хранилищем

Cloud Foundry + Kubernetes + Mesos + Docker

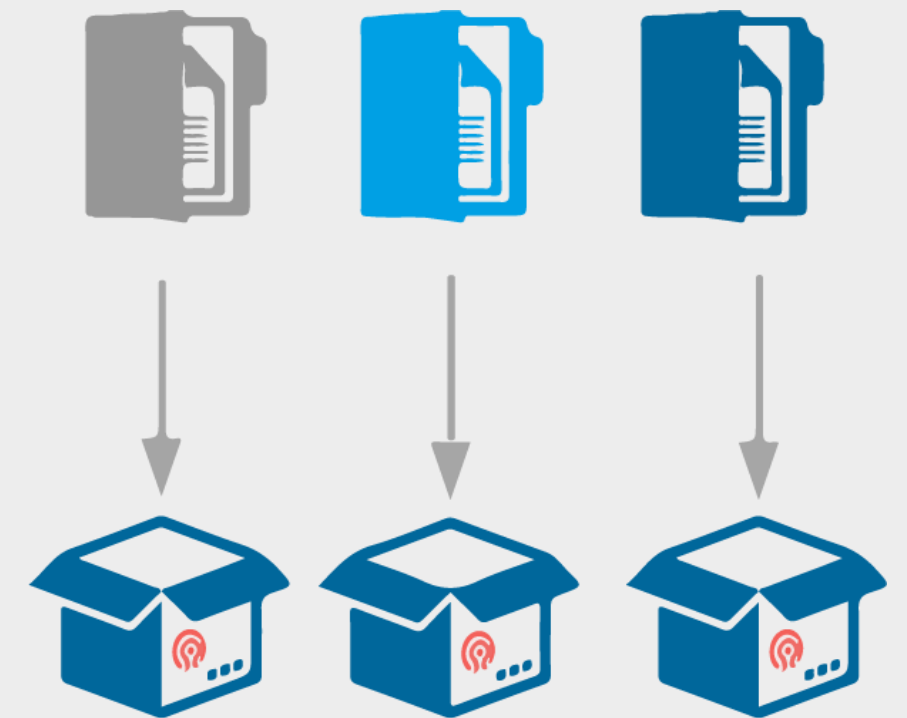




# Создание пула для RBD

```
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.rbdkube mon 'allow r, allow command "osd  
blacklist"' osd 'allow rwx pool=kube'
```



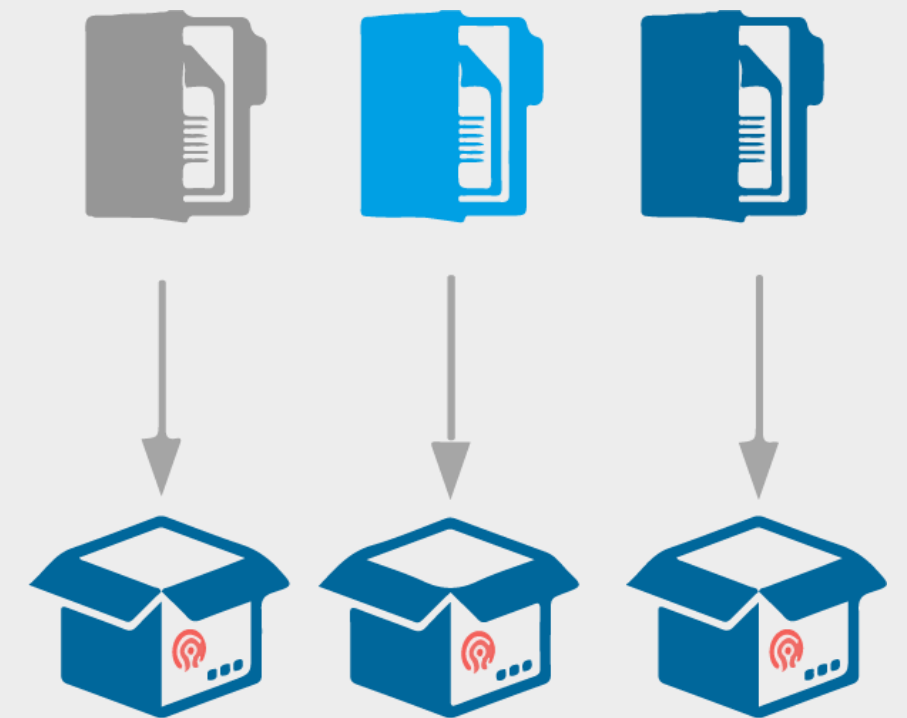
# Подключение RBD

```
helm repo add ceph-csi https://ceph.github.io/csi-charts  
helm inspect values ceph-csi/ceph-csi-rbd --version 2.1.0 >cephrbd.yml
```

```
node-1# ceph fsid -> - clusterID:
```

```
node-1# ceph mon dump -> monitors:  
- "<MONValue1>"  
- "<MONValue2>"
```

```
kubectl create ns ceph-csi-rbd  
helm upgrade -i ceph-csi-rbd ceph-csi/ceph-csi-rbd \  
-f cephcbd.yml -n ceph-csi-rbd
```





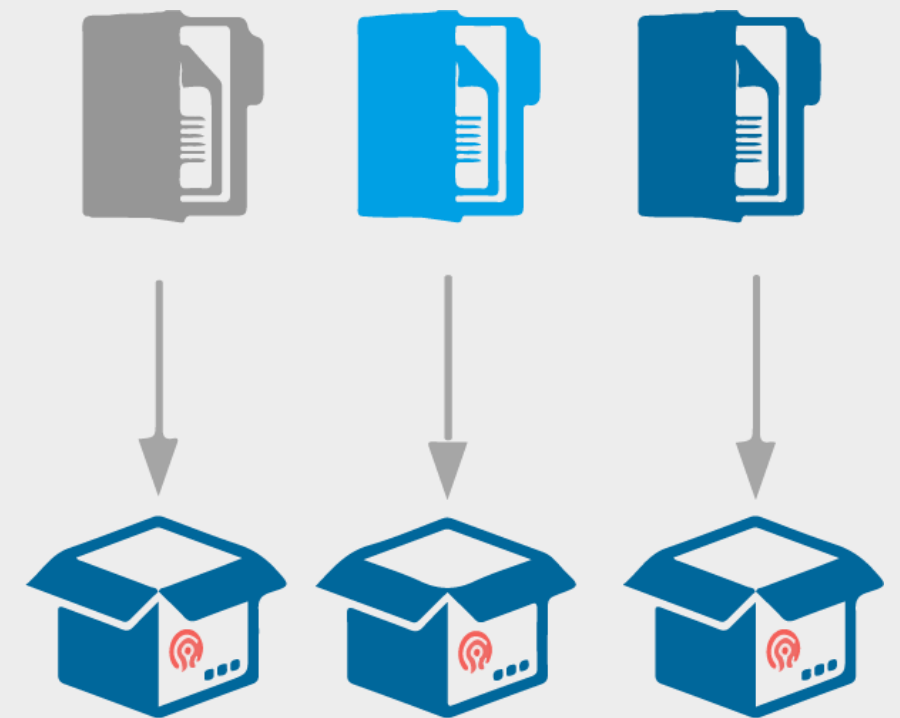
# Подключение RBD

Client key:

```
node-1# ceph auth get-key client.rbdkube
```

```
master-1# kubectl apply -f secret.yaml
```

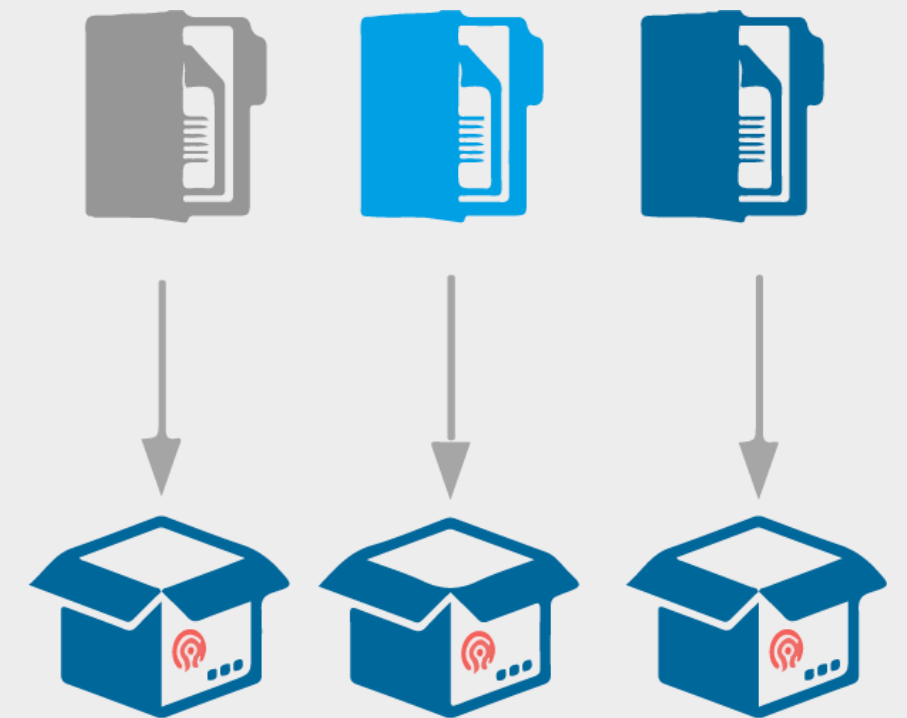
```
master-1# kubectl apply -f storageclass.yaml
```



# Подключение RBD

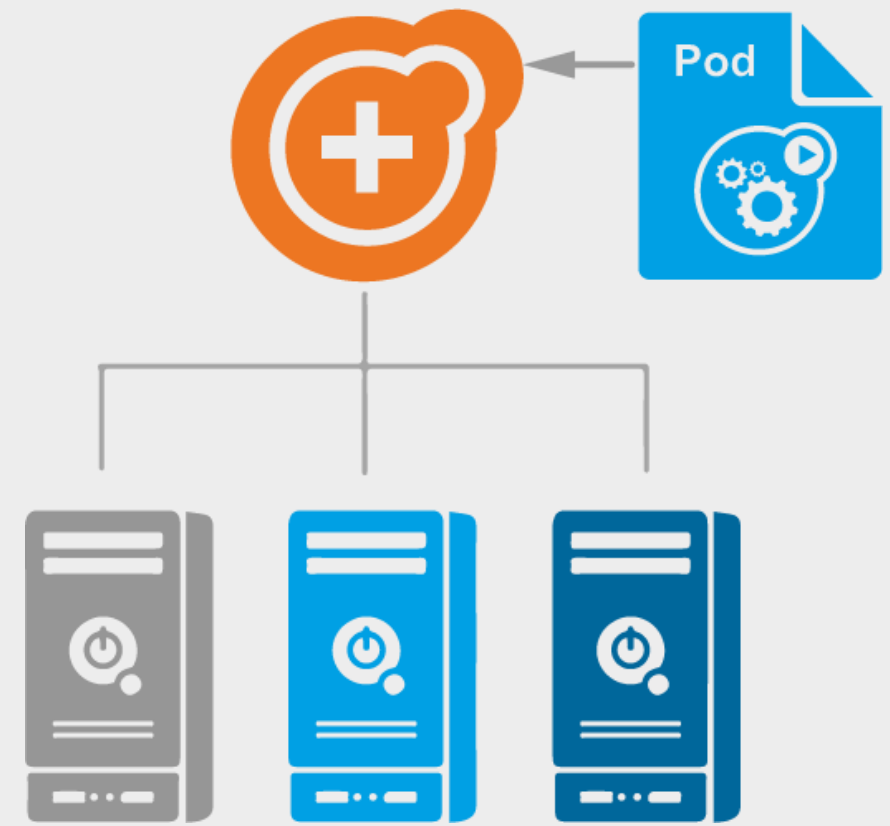
## PVC

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-rbd-sc
```



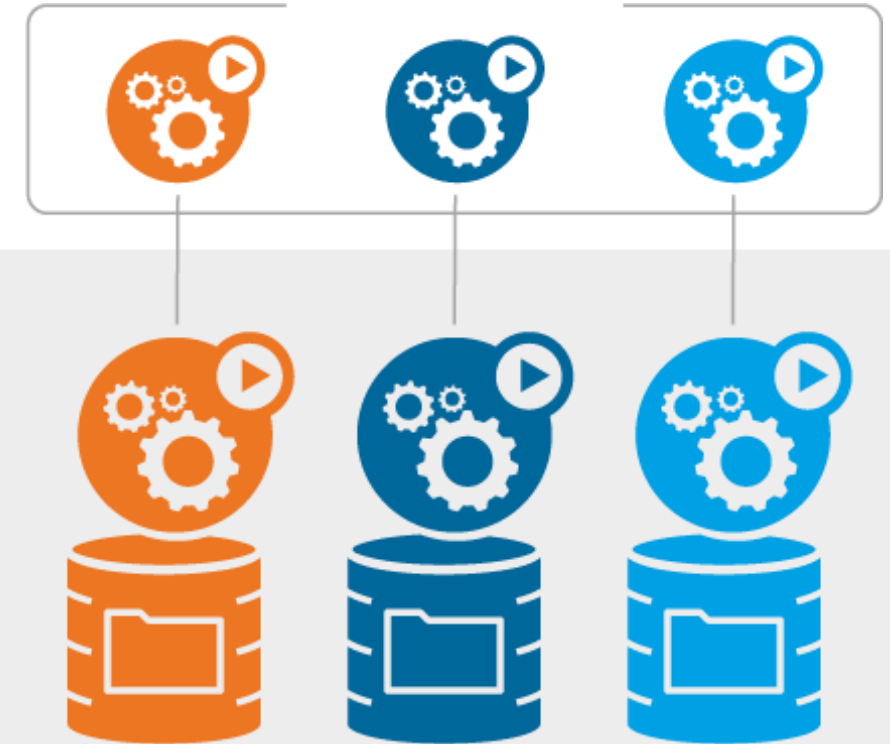
# Deployment

```
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    spec:
      containers:
      - volumeMounts:
        - name: data
          mountPath: /data
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: rbd-pvc
```



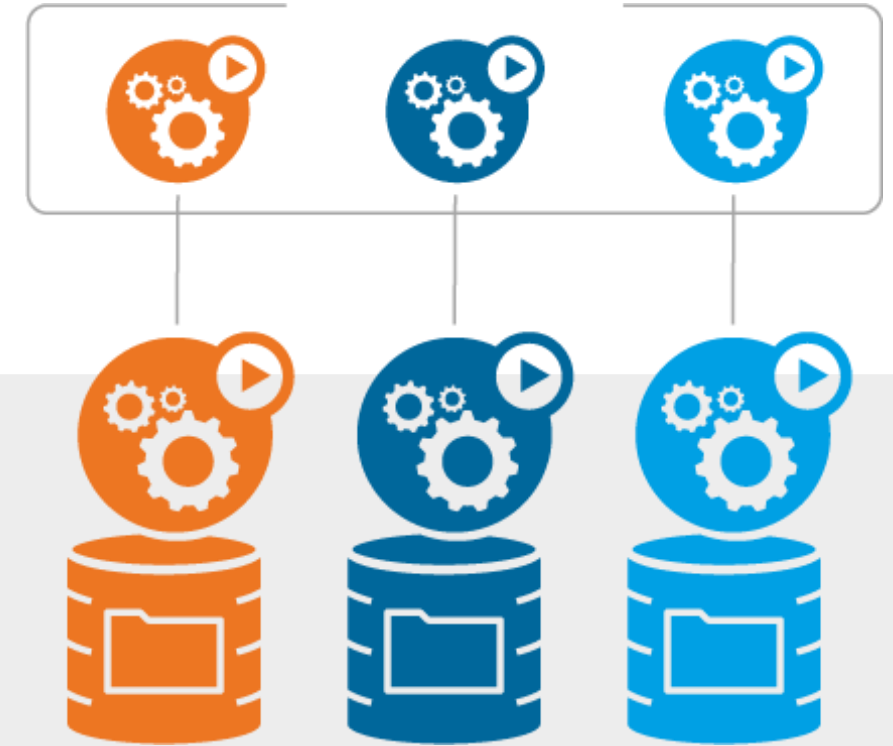
# StatefulSet

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



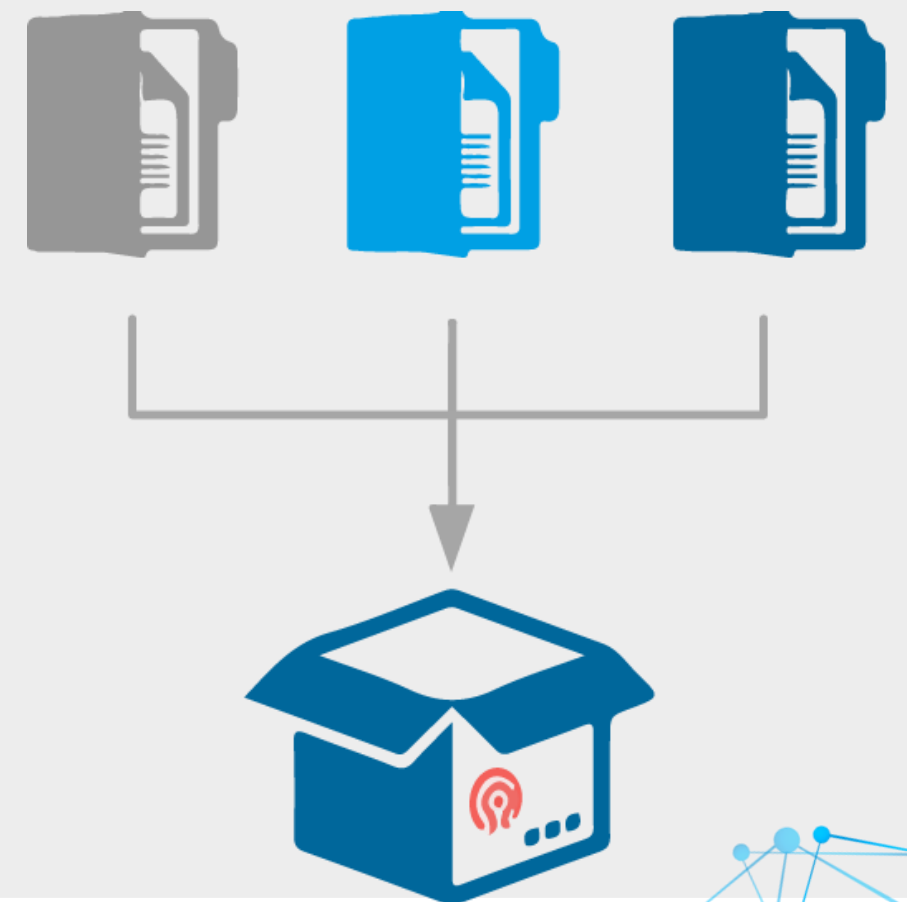
# initContainers

```
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

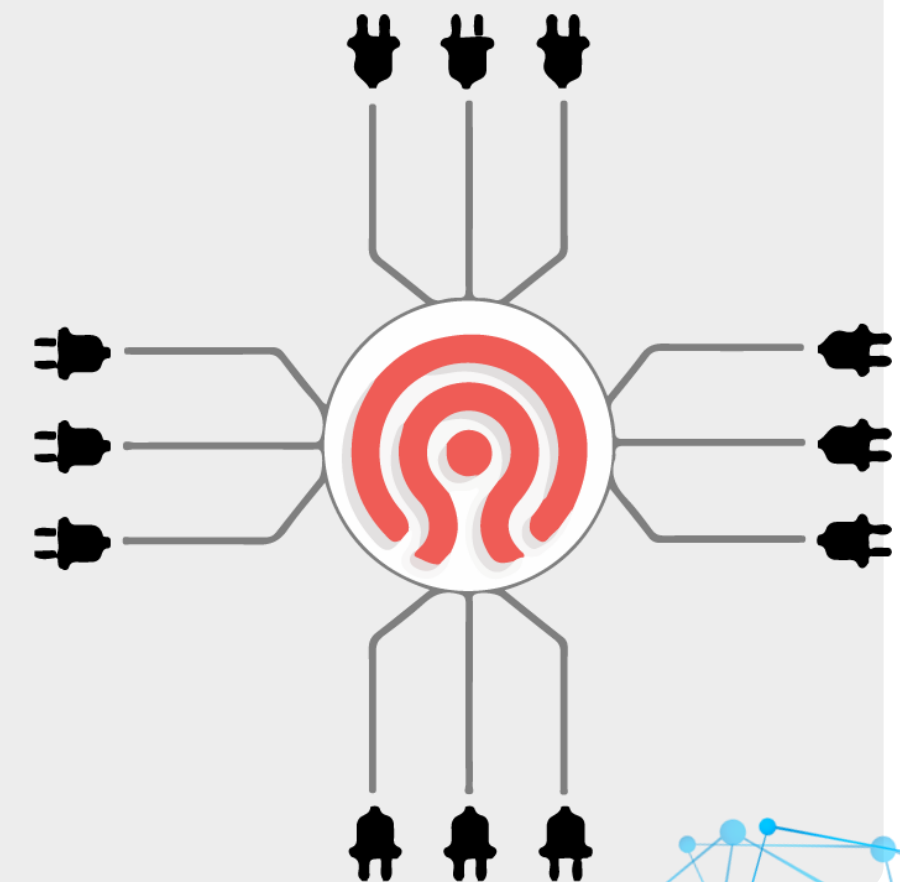
```
helm inspect values ceph-csi/ceph-csi-cephfs --version 2.1.0 >cephfs.yml
```

```
node-1# ceph fsid -> - clusterID:
```

```
node-1# ceph mon dump -> monitors:  
    - "<MONValue1>"  
    - "<MONValue2>"
```

```
kubectl create ns ceph-csi-cephfs
```

```
helm upgrade -i ceph-csi-cephfs ceph-csi/ceph-csi-cephfs \  
  -f cephfs.yml -n ceph-csi-cephfs
```



# Подключение CephFS

Admin key:

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

```
master-1# kubectl apply -f secret.yaml
```

```
master-1# kubectl apply -f storageclass.yaml
```

