

# Rook Ceph 集群部署

---

AUTHOR: 彭玲 TIME: 2022/2/25

---

## Rook Ceph 集群部署

Ceph 存储

RADOS

Ceph 存储类型

Rook 部署准备

1. K8s 版本要求

2. 存储要求

3. 内核要求

RBD

CephFS

LVM 包

Rook 部署

部署步骤

部署检查

Ceph Dashboard

启用 Dashboard

集群内部访问

集群外部访问 (NodePort)

Ceph Dashboard 系统

1. 登录凭证

2. Ceph 控制面板

Rook 环境清理

删除 CRD

删除 Operator 和相关资源

---

## Ceph 存储

---

Ceph 是一个分布式存储系统，具备大规模、高性能、无单点失败的特点。

## RADOS

RADOS (分布式对象存储) 是 Ceph 分布式存储系统的组成部分。RADOS 包含两类节点：OSD 存储节点、管理节点。

## Ceph 存储类型

- 块存储：硬盘等裸设备的存储方式，Ceph 创建时，就是创建一个虚拟的硬盘。
- 文件存储：为了克服块存储文件无法共享的问题，所以有了文件存储。
- 对象存储：用 S3 兼容接口开放存储服务。

## Rook 部署准备

---

## 1. K8s 版本要求

Kubernetes **v1.16** or higher is supported for the Ceph operator.

## 2. 存储要求

为了配置 Ceph 存储集群，至少需要以下本地存储选项之一：

- 原始设备（无分区或格式化文件系统）
- 原始分区（无格式化文件系统）
- `block` 模式下存储类可用的 PV。

## 3. 内核要求

### RBD

一般发行版的内核都编译有，但你最好确定下：

```
1 dragon@test-pc:~$ lsmod | grep rbd
2 Module          Size      Used by
3 rbd              106496    0
4 libceph          327680    1 rbd
```

### CephFS

如果你想使用 `cephfs`，内核最低要求是 4.17。

```
1 # 查看 Linux 内核版本
2 dragon@test-pc:~$ uname -r
3 5.4.0-90-generic
```

### LVM 包

在以下情况下，Ceph OSD 依赖于 LVM：

- OSD 在原始设备或分区上创建。
- 如果启用了加密（`encryptedDevice: true` in the cluster CR）。
- 指定了 `metadata` 设备。

在以下情况下，OSD 不需要 LVM：

- 使用 `storageClassDeviceSets` 在 PVC 上创建 OSD。

## Rook 部署

---

### 部署步骤

可以使用以下 `kubect1` 命令创建一个简单的 Rook 集群：

```
1 $ git clone --single-branch --branch v1.8.5 https://github.com/rook/rook.git
2 $ cd rook/deploy/examples
3
4 # 1. 创建 CRD 和 Common Resources
5 $ kubectl create -f crds.yaml -f common.yaml
6 # 2. 创建 Operator
7 $ kubectl create -f operator.yaml
8 # 3. 创建 Ceph storage cluster
9 $ kubectl create -f cluster.yaml
```

## 部署检查

下面这些 Pods 应该正常运行：

```
1 $ kubectl -n rook-ceph get pods
2 NAME                READY   STATUS    AGE
3 rook-ceph-mgr-a     1/1    Running   61s
4 rook-ceph-mon-a     1/1    Running   92s
5 rook-ceph-mon-b     1/1    Running   84s
6 rook-ceph-mon-c     1/1    Running   71s
7 rook-ceph-osd-0     1/1    Running   21s
8 rook-ceph-operator  1/1    Running   4m17s
9 ...
```

## Ceph Dashboard

Ceph 有一个 Dashboard 工具，我们可以在上面查看集群的状态，包括总体运行状态，mgr、osd 和其他 Ceph 进程的状态，查看池和 PG 状态，以及显示守护进程的日志等等。

## 启用 Dashboard

创建 Dashboard，只需要在 Ceph 集群创建中，对 `cluster.yaml` 设置 `dashboard.enable=true` 即可：

```
1 apiVersion: ceph.rook.io/v1
2 kind: CephCluster
3 metadata:
4   name: rook-ceph
5   namespace: rook-ceph # namespace:cluster
6 spec:
7   # enable the ceph dashboard for viewing cluster status
8   dashboard:
9     enabled: true
10    # serve the dashboard under a subpath (useful when you are accessing the
11    dashboard via a reverse proxy)
12    # urlPrefix: /ceph-dashboard
13    # serve the dashboard at the given port.
14    # port: 8443
15    # serve the dashboard using SSL
16    ssl: true
17 ...
```

这样 Rook Operator 就会创建一个 K8s Service 来暴露该服务，将启用 8443 端口进行 https 访问。

## 集群内部访问

如果在集群内部，我们可以通过 DNS 名称 `http://rook-ceph-mgr-dashboard.rook-ceph:<port>` 或者 CluterIP `<cluster_ip>:<port>` 来进行访问。

## 集群外部访问 (NodePort)

如果想要外部访问的话，则使用下面的 `kubect1` 命令将 `type` 修改为 `NodePort` 即可。

```
1 $ kubectl edit svc -n rook-ceph rook-ceph-mgr-dashboard
2
3 apiVersion: v1
4 kind: Service
5 spec:
6   type: NodePort
7   ...
```

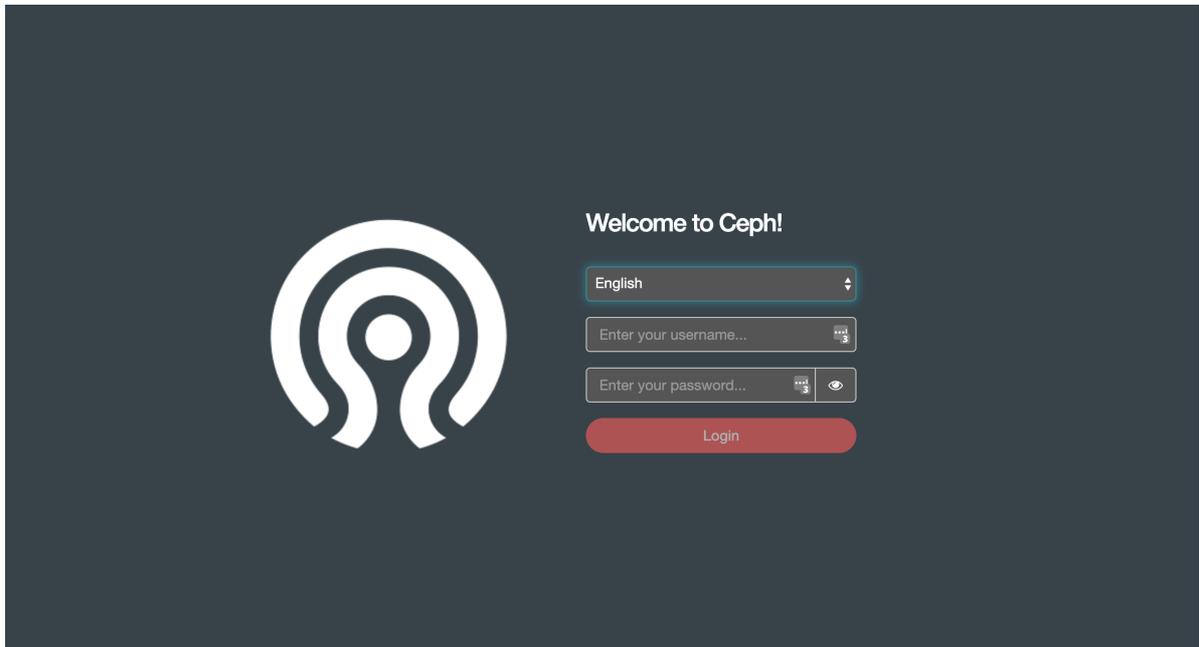
这时就可以使用 `http://<node_ip>:<port>` 就可以访问到 Dashboard 了。

## Ceph Dashboard 系统

### 1. 登录凭证

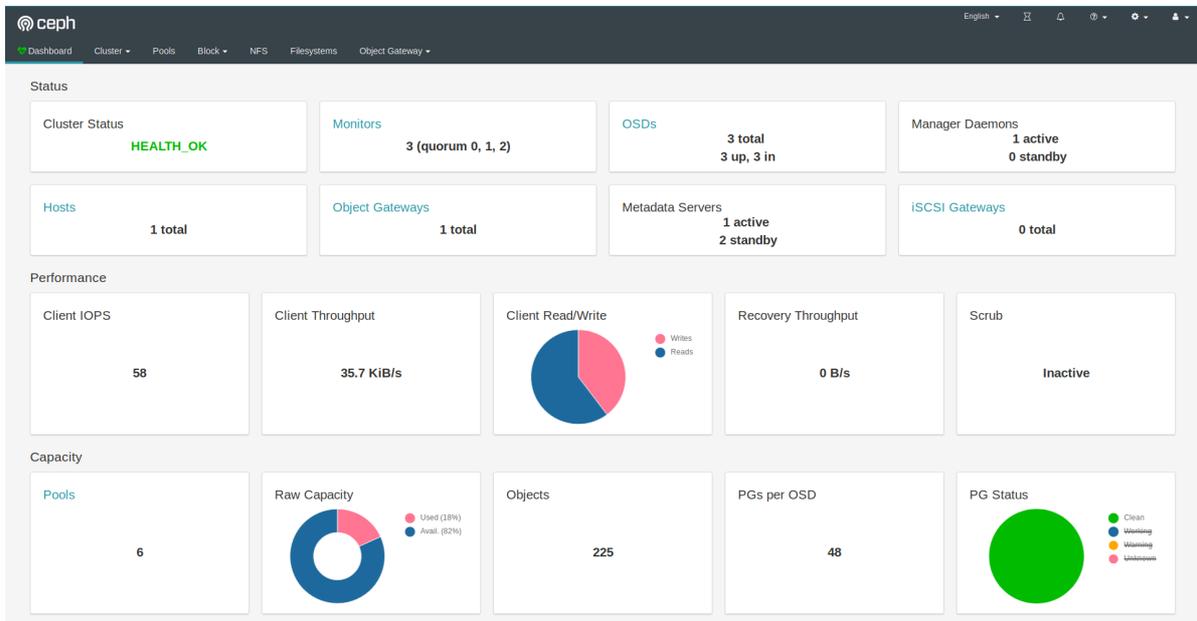
Dashboard 的访问需要登录：Rook 创建了一个默认的用户 `admin`，并在运行 Rook 的命名空间 `rook-ceph` 中生成了一个名为 `rook-ceph-dashboard-admin-password` 的 Secret。要获取密码，可以运行以下命令：

```
1 $ kubectl -n rook-ceph get secret rook-ceph-dashboard-password -o jsonpath="{['data']['password']}" | base64 --decode && echo
```



### 2. Ceph 控制面板

用上面获得的 密码 和用户名 `admin` 就可以登录 Dashboard 了，在上面可以查看到整个集群的状态：



## Rook 环境清理

如果你想要关闭集群并启动一个新的集群，请注意以下需要清理的资源：

- `rook-ceph` namespace: The Rook operator and cluster created by `operator.yaml` and `cluster.yaml` (the cluster CRD).
- `/var/lib/rook`: Path on each host in the cluster where configuration is cached by the ceph mons and osds.

## 删除 CRD

Delete the CephCluster CR.

```
1 | anxin@node38:~$ kubectl -n rook-ceph delete cephcluster rook-ceph
```

Verify that the cluster CR has been deleted before continuing to the next step.

```
1 | anxin@node38:~$ kubectl -n rook-ceph get cephcluster
```

## 删除 Operator 和相关资源

```
1 | anxin@node38:~/pengling/k8s/rook-ceph$ kubectl delete -f operator.yaml
2 | anxin@node38:~/pengling/k8s/rook-ceph$ kubectl delete -f common.yaml
3 | anxin@node38:~/pengling/k8s/rook-ceph$ kubectl delete -f crds.yaml
```