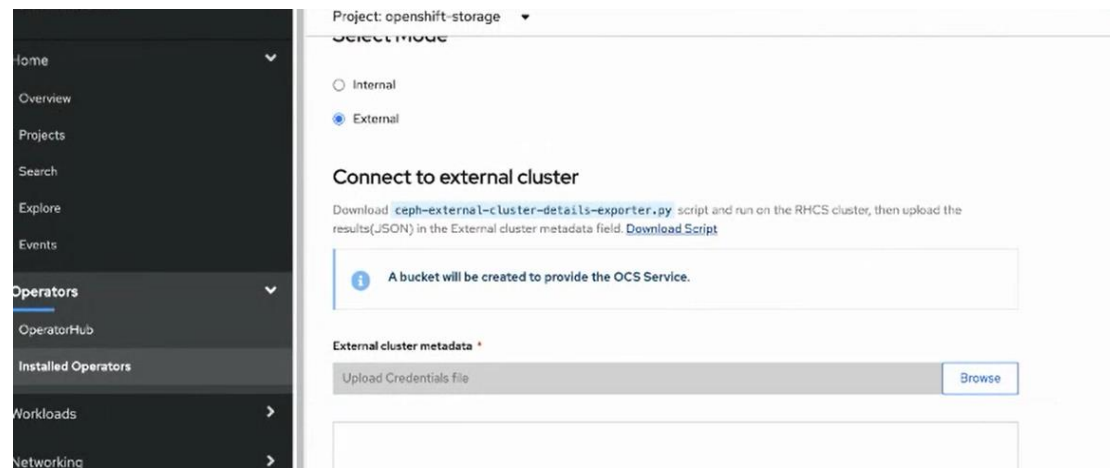


OCS 4.5 支持外部的存储模式。也就是说，通过 OCP 上安装的 OCS operator，可以对接在外部物理机上安装的 Ceph。然后以 OCP 中的 rook 方式，管理外部物理机上的 Ceph，实现 OCP 和存储的解耦。

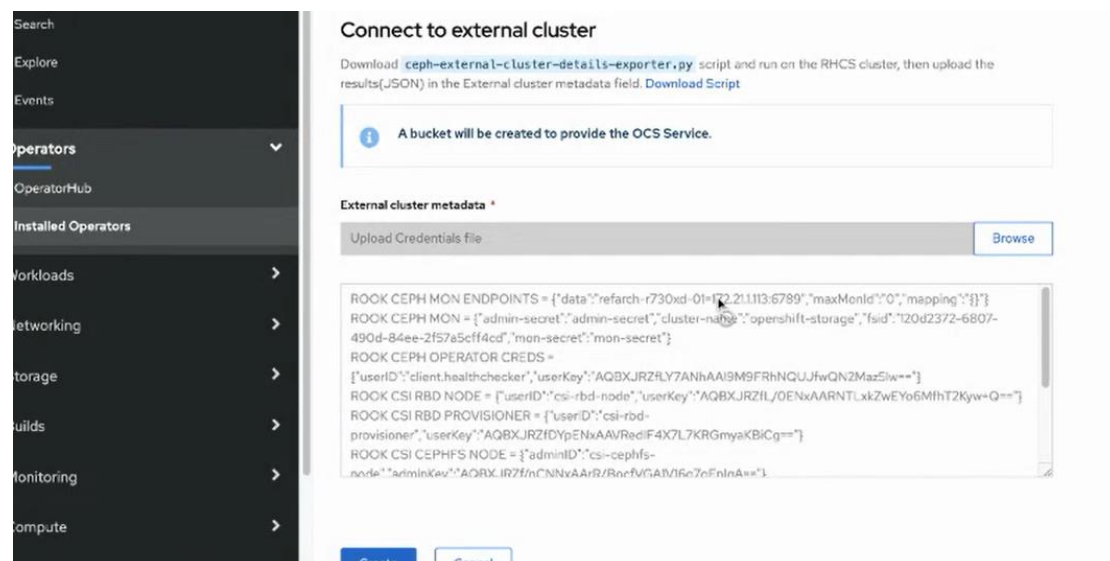
我们在 OCP 上部署 ocs operator 后，可以选择连接外部的 cluster：然后提示使用一个 py 脚本。



我们将这个脚本在外置的 ceph 的 mon 的节点上执行：

```
[root@refarch-r730xd-01 ~]# python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name rbd --rgw-endpoint 172.21.1.116:8080
[{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data": "refarch-r730xd-01=172.21.1.113:6789", "maxMonId": "0", "mapping": "{": "Secret", "data": {"admin-secret": "admin-secret", "cluster-name": "openshift-storage", "fsid": "120d2372-6807-490d-84ee-2f57a5c5ff4cd", "mon-secret": "mon-secret", "data": {"admin-secret": "admin-secret", "cluster-name": "openshift-storage", "fsid": "120d2372-6807-490d-84ee-2f57a5c5ff4cd", "mon-secret": "mon-secret"}, {"name": "rook-ceph-operator-creds", "kind": "Secret", "data": {"userID": "client.healthchecker", "userKey": "AQ8XJRZFLY7ANhAA19M9FRhNQJfwQN2Maz5lw=="}, {"name": "rook-ceph-provisioner", "kind": "Secret", "data": {"userID": "csi-rbd-node", "userKey": "AQ8XJRZFLY7ANhAA19M9FRhNQJfwQN2Maz5lw=="}, {"name": "rook-cephfs-provisioner", "kind": "Secret", "data": {"adminID": "csi-cephfs-provisioner", "adminKey": "AQ8XJRZFLY7ANhAA19M9FRhNQJfwQN2Maz5lw=="}, {"name": "ceph-rbd", "kind": "StorageClass", "data": {"pool": "rbd"}, {"name": "cephfs", "kind": "StorageClass", "data": {"fsName": "cephfs"}, {"name": "ceph-rgw", "kind": "StorageClass", "data": {"endpoint": "172.21.1.116:8080"}]}
```

将脚本的输出内容粘贴到 ocs 连接外部存储的位置：



然后我们看到，在 OCP 中，会自动创建 csi 的 pod

```
[root@refarch-r220-04 ~]# oc get pods
```

NAME	READY	STATUS	RESTARTS	AGE
csi-cephfsplugin-gftx5	3/3	Running	0	18s
csi-cephfsplugin-provisioner-777b6c59df-7rp8g	5/5	Running	0	17s
csi-cephfsplugin-provisioner-777b6c59df-rktcc	5/5	Running	0	11s
csi-cephfsplugin-qwfgg	3/3	Running	0	20s
csi-cephfsplugin-vw4cm	3/3	Running	0	19s
csi-rbdplugin-54tsj	3/3	Running	0	23s
csi-rbdplugin-55pbt	3/3	Running	0	22s
csi-rbdplugin-jz5vb	3/3	Running	0	22s
csi-rbdplugin-provisioner-6bcd8df965-6zjr8	5/5	Running	0	21s
csi-rbdplugin-provisioner-6bcd8df965-czwbj	5/5	Running	0	19s
noobaa-core-0	1/1	Running	0	22s
noobaa-db-0	0/1	Init:0/1	0	22s
noobaa-operator-d8d7d7c84-27vb7	1/1	Running	0	12m
ocs-operator-b4847d86f-j4r6w	1/1	Running	0	12m
rook-ceph-operator-6955497b9c-5ltdv	1/1	Running	0	12m

查看外部存储:

```
[root@refarch-r220-04 ~]# oc get storagecluster
```

NAME	AGE	PHASE	EXTERNAL	CREATED AT	VERSION
ocs-external-storagecluster	105s	Ready	true	2020-09-04T03:26:03Z	4.5.0

查看自动创建的 storageclass:

```
[root@refarch-r220-04 ~]# oc get sc
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
ocs-external-storagecluster-ceph-rbd	openshift-storage.rbd.csi.ceph.com	Delete	Immediate	true	2m26s
ocs-external-storagecluster-ceph-rgw	openshift-storage.ceph.rook.io/bucket	Delete	Immediate	false	2m26s
ocs-external-storagecluster-cephfs	openshift-storage.cephfs.csi.ceph.com	Delete	Immediate	true	2m26s
openshift-storage.noobaa.io	openshift-storage.noobaa.io/obc	Delete	Immediate	false	62s

使用自动创建的 storageclass 创建 pvc:

```

[root@refarch-r220-04 ocs-config]# oc create -f rbd-loop.yaml
namespace/my-rbd-storage created
persistentvolumeclaim/pvc-cephrbd1 created
persistentvolumeclaim/pvc-cephrbd2 created
job.batch/batch2 created
[root@refarch-r220-04 ocs-config]# oc get pods -n my-rbd-storage
NAME          READY   STATUS             RESTARTS   AGE
batch2-kks7t   0/1     ContainerCreating   0           9s
[root@refarch-r220-04 ocs-config]# oc get pods -n my-rbd-storage
NAME          READY   STATUS             RESTARTS   AGE
batch2-kks7t   0/1     ContainerCreating   0          11s
[root@refarch-r220-04 ocs-config]# oc get pods -n my-rbd-storage -w
NAME          READY   STATUS             RESTARTS   AGE
batch2-kks7t   0/1     ContainerCreating   0          14s
batch2-kks7t   1/1     Running            0          18s
^C[root@refarch-r220-04 ocs-config]# oc logs -n my-rbd-storage batch2-kks7t
Creating temporary file
1+0 records in
1+0 records out
1048576 bytes (1.0 MB) copied, 0.0142279 s, 73.7 MB/s
Copying temporary file
Going to sleep

```

登录到 ceph 上，vol 会自动创建：

```

[root@refarch-r730xd-01 ~]# rbd ls
csi-vol-69866a28-ee5e-11ea-a501-0a580a810076
csi-vol-7163f856-ed72-11ea-ba9b-0a580a80007f
csi-vol-f5225d9a-ee5e-11ea-a501-0a580a810076
csi-vol-f53df475-ee5e-11ea-a501-0a580a810076

```

使用 rook 方式去管理外置 ceph，能够享受到不少 ocs 的好处。例如:pvc 的动态扩容。

在下图中，利用 cephfs 的 sc 创建 pvc:

---  
apiVersion: v1

```
kind: Namespace
metadata:
  name: my-rwx-storage
spec: {}
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-cephfs
  namespace: my-rwx-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 500Mi
  storageClassName: ocs-storagecluster-cephfs
---
kind: PersistentVolumeClaim
```

创建完后，可以对 pvc 进行扩容：

