# Установка Ceph CSI driver для подключения RBD

Все работы проводим под root'ом

После создания стенда авторизуйтесь на adminbox по адресу **sbox.slurm.io.** По SSH с помощью логина и пароля, которые находятся в настройках профиля или над текстом по кнопке "**Доступы**".

С **sbox** на сервер `master-1.s000000` (s000000 меняем на номер своего студента)

```
ssh master-1.s000000
```

Клонируем репозиторий наш в домашней папке

```
cd ~
```

```
git clone git@gitlab.slurm.io:edu/slurm.git
```

```
cd ~/slurm/practice/7.datastorage/rbd/
```

Далее будем работать на двух серверах: `node-1` и `master-1`. Все команды `kubectl` выполняются на `master-1`. Команды `ceph` и `rbd` выполняются на `node-1`

На серверах ingress-1, node-1 и node-2 развернут кластер ceph, более подробно установка рассмотрена в курсе по ceph.

На сервере `node-1` настроен доступ в ceph с правами администратора.

**Проверяем состояние кластера**

1) Открываем 2-ое окно консоли и с sbox заходим на node-1

```
ssh node-1.s<номер вашего логина>
```

```
sudo -s
```

2) Проверяем что ceph живой

```
ceph health
```

```
ceph -s
```

**Создаем пул в Ceph'e для RBD дисков**

3) Запускаем на node-1

```
ceph osd pool create kube 32
```

```
ceph osd pool application enable kube rbd
```

**Получаем файл с настройками chart**

4) Переключаемся на `master-1` и добавляем репозиторий с chart'ом, получаем набор переменных chart'a ceph-csi-rbd

```
helm repo add ceph-csi https://ceph.github.io/csi-charts
```

```
helm inspect values ceph-csi/ceph-csi-rbd > cephrbd.yml
```

**Заполняем переменные в cephrbd.yml**

5) Выполняем на `node-1`, чтобы узнать необходимые параметры:

Получаем clusterID

```
ceph fsid
```

Получаем список мониторов кластера Ceph

```
ceph mon dump
```

6) Переключаемся снова на `master-1` и правим файл `cephrbd.yml`

Заносим свои значение clusterID, и адреса мониторов. Включаем создание политик PSP, и увеличиваем таймаут на создание дисков Список изменений в файле cephrbd.yml. Опции в разделах nodeplugin и provisioner уже есть в файле, их надо исправить так, как показано ниже.

```
csiConfig:
  - clusterID: "bcd0d202-fba8-4352-b25d-75c89258d5ab"
    monitors:
      - "v2:172.18.8.5:3300/0,v1:172.18.8.5:6789/0"
      - "v2:172.18.8.6:3300/0,v1:172.18.8.6:6789/0"
      - "v2:172.18.8.7:3300/0,v1:172.18.8.7:6789/0"
```

```
nodeplugin:
  podSecurityPolicy:
    enabled: true

provisioner:
  replicaCount: 1
  podSecurityPolicy:
    enabled: true
```

При необходимости можно свериться с файлом `rbd/cephrbd-values-example.yml`

**Устанавливаем чарт**

7) Выполняем команду

```
helm upgrade -i ceph-csi-rbd ceph-csi/ceph-csi-rbd -f cephrbd.yml -n ceph-csi-rbd
--create-namespace
```

**Создаем секрет и storageclass**

8) Создаем пользователя в ceph, с правами записи в пул kube

Запускаем на `node-1`

```
ceph auth get-or-create client.rbdkube mon 'profile rbd' osd 'profile rbd
pool=kube'
```

9) Смотрим ключ доступа для пользователя rbdkube

```
ceph auth get-key client.rbdkube
```

вывод:

```
AQCO9NJbhYipKRAAMqZsnqqS/T8OYQX20xIa9A==
```

10) Заполняем манифесты

Выполняем на `master-1`, подставляем значение ключа в манифест секрета.

```
cd ~/slurm/practice/7.datastorage/rbd/
```

`vi secret.yaml`

Заносим значение ключа в

`userKey: AQBRYK1eo++dHBAATnZzl8MogwwqP/7KEnuYyw==`

11) Создаем секрет

`kubectl apply -f secret.yaml`

12) Получаем id кластера ceph

Выполняем на `node-1`

`ceph fsid`

13) Заносим clusterid в `storageclass.yaml`

Выполняем на `master-1`

`vi storageclass.yaml`

`clusterID: bcd0d202-fba8-4352-b25d-75c89258d5ab`

14) Создаем storageclass

`kubectl apply -f storageclass.yaml`

**Проверяем как работает.**

15) Создаем pvc, и проверяем статус и наличие pv

`kubectl apply -f pvc.yaml`

`kubectl get pvc`

`kubectl get pv`

**Проверяем создание тома в ceph**

16) Получаем список томов в пуле и просматриваем информацию о томе

Выполняем на `node-1`

```
rbd ls -p kube
```

```
rbd -p kube info csi-vol-eb3d257d-8c6c-11ea-bff5-6235e7640653
```

**Пробуем как работает resize**

17) Изменяем размер тома в манифесте `pvc.yaml` и применяем

Выполняем на `master-1`

```
vi pvc.yaml
resources:
 requests:
   storage: 2Gi
kubectl apply -f pvc.yaml
```

18) Немножко ждем и проверяем размер тома в ceph, pv, pvc

Выполняем на `node-1`

```
rbd -p kube info csi-vol-eb3d257d-8c6c-11ea-bff5-6235e7640653
```

Выполняем на `master-1`

```
kubectl get pv
```

```
kubectl get pvc
```

**Разбираемся почему у pvc размер не изменился**

19) Смотрим его манифест в yaml

Выполняем на `master-1`

```
kubectl get pvc rbd-pvc -o yaml
```

То увидим сообщение `message: Waiting for user to (re-)start a pod to finish file system resize of volume on node. type: FileSystemResizePending` Необходимо смонтировать том, для того чтобы увеличить на нем файловую систему. А у нас этот PVC/PV не используется никаким подом.

20) Создаем под, который использует этот PVC/PV, и смотрим на размер, указанный в pvc

```
kubectl apply -f pod.yaml

kubectl get pvc
```

+2