


```
root@test0:~# mount -t ceph 10.99.99.74:/ /mnt/mycephfs -o
secret=AQCGVs5lyBLmlxAApqSed51BIHOvQlyawvG2Uw== -o name=foo

# fstab:
10.99.99.74:/ /mnt/cephfs ceph
name=foo,secret=AQCGVs5lyBLmlxAApqSed51BIHOvQlyawvG2Uw==,noatime,_netdev 0 2
```

+ k8s

Нам понадобится ключ для доступа к cephfs (а так же к пулу, здесь я указал админского юзера, но можно самому создать, грамотно выделив права), закинуть в наш куб CSI (Container Storage Interface) с указанными параметрами, storageclass с секретом и хранилкой можно пользоваться.

```
# в этом подтоме будут ФСки кластеров
root@microceph:~# ceph fs subvolumegroup create cephfs csi

root@node1:~/cephfs# snap install helm --classic
helm 3.14.1 from Snapcrafters installed

root@node1:~/cephfs# helm repo add ceph-csi https://ceph.github.io/csi-charts
"ceph-csi" has been added to your repositories

root@node1:~/cephfs# helm inspect values ceph-csi/ceph-csi-cephfs > cephfs.yml
```

valuesы у хельм чарта:

```
---
rbac:
  # Specifies whether RBAC resources should be created
  create: true

serviceAccounts:
  nodeplugin:
    # Specifies whether a ServiceAccount should be created
    create: true
    # The name of the ServiceAccount to use.
```



```
# maxUnavailable is the maximum number of pods that can be
# unavailable during the update process.
maxUnavailable: 50%

# Timeout for waiting for creation or deletion of a volume
timeout: 60s

# cluster name to set on the subvolume
# clustername: "k8s-cluster-1"

# set user created priorityClassName for csi provisioner pods. default is
# system-cluster-critical which is less priority than system-node-critical
priorityClassName: system-cluster-critical

# enable hostnetwork for provisioner pod. default is false
# useful for deployments where the podNetwork has no access to ceph
enableHostNetwork: false

httpMetrics:
  # Metrics only available for cephcsi/cephcsi => 1.2.0
  # Specifies whether http metrics should be exposed
  enabled: true
  # The port of the container to expose the metrics
  containerPort: 8081

service:
  # Specifies whether a service should be created for the metrics
  enabled: true
  # The port to use for the service
  servicePort: 8080
  type: ClusterIP

  # Annotations for the service
  # Example:
  # annotations:
  #   prometheus.io/scrape: "true"
  #   prometheus.io/port: "9080"
  annotations: {}

clusterIP: ""

## List of IP addresses at which the stats-exporter service is available
```


