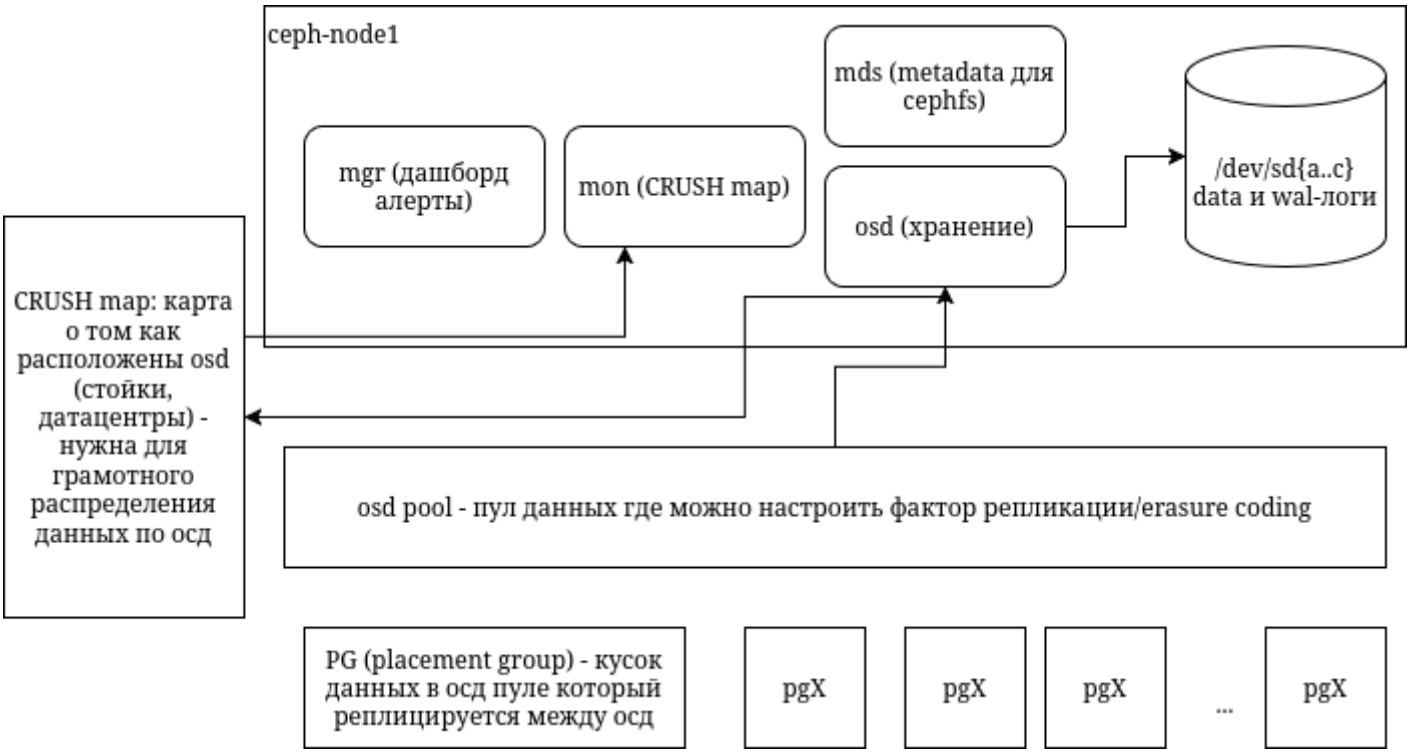
ceph



Если будешь ставить через cephadm, то качай версию pacific, почему-то более новые не работают.

Удобный установщик: https://git.lulzette.ru/lulzette/ceph-installer

???????

- osd: хранение данных
- mon: условно, controlplane, управляет данными, балансирует данные, управляет кворумом, метаданные тоже тут
- mgr: These include the Ceph Dashboard and the cephadm manager module, фигня на питоне которая дает дашборд, прометеус, restful api и всякую такую некритичную для работы кластера фигню

??? ????????????? ?????????

1. Хосты с ролью OSD

2. OSDшки (абстракция на уровне Ceph'а): 1 OSD на 1 Диск хоста, т.е. 10 дисков - 10 OSD на одном хосте
3. PG (Placement Group): Группа размещения, т.е. на каких OSD размещать объекты Ceph'а (не путать с объектом на уровне S3/swift/rgw). Также есть **Карта OSD**, которая ассоциирована с PG, в ней указана одна Primary OSD и одна или несколько Replica OSD. Вся работа происходит с Primary OSD, а синк данных на реплики идет асинхронно.

???? ????? ???????????

Есть 3 разных хранилища (?):

- cephfs
- RBD (Rados Block Device)
- S3/Swift который предоставляется Rados Gateway
- Rados/librados - библиотека для твоего приложения которая может общаться с радосом минуя промежуточные RBD/S3/Swift

?????????:

- Рассинхронизация времени на хоста
- See also:

https://bogachev.biz/2017/08/23/zametki-administratora-ceph-chast-1/

CephFS

Поднять MDS - сервер метаданных

Вообще достаточно просто

```
# mon
ceph fs volume create cephfs


root@microceph:~# ceph config generate-minimal-conf
# minimal ceph.conf for 56db00e1-912c-4ac1-9d1a-1f4194c55834
[global]
```

```
        fsid = 56db00e1-912c-4ac1-9d1a-1f4194c55834
        mon_host = [v2:10.99.99.74:3300/0,v1:10.99.99.74:6789/0]

root@microceph:~# ceph fs authorize cephfs client.foo / rw
[client.foo]
        key = AQCGVs5lyBLmIxAApqSed51BlHOvQlyawvG2Uw==

# client
# может показаться что все что ниже, кроме команды монтирования, не имеет смысла, но не
root@test0:~# mkdir /etc/ceph
root@test0:~# vim /etc/ceph/ceph.conf
root@test0:~# vim /etc/ceph/ceph.client.foo.keyring
root@test0:~# chmod 644 /etc/ceph/ceph.conf
root@test0:~# chmod 600 /etc/ceph/ceph.client.foo.keyring

root@test0:~# mount -t ceph 10.99.99.74:/ /mnt/mycephfs -o
secret=AQCGVs5lyBLmIxAApqSed51BlHOvQlyawvG2Uw== -o name=foo

# fstab:
10.99.99.74:/ /mnt/cephfs ceph
name=foo,secret=AQCGVs5lyBLmIxAApqSed51BlHOvQlyawvG2Uw==,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.
    # If not set and create is true, a name is generated using the fullname
    name:
  provisioner:
    # Specifies whether a ServiceAccount should be created
    create: true
    # The name of the ServiceAccount to use.
    # If not set and create is true, a name is generated using the fullname
    name:

# Configuration for the CSI to connect to the cluster
# Ref: https://github.com/ceph/ceph-csi/blob/devel/examples/README.md
# Example:
csiConfig:
  - clusterID: "56db00e1-912c-4ac1-9d1a-1f4194c55834"
    monitors:
      - "10.99.99.74:6789"
#     cephFS:
#       subvolumeGroup: "csi"
#       netNamespaceFilePath: "{{ .kubeletDir }}/plugins/{{ .driverName }}/net"
#csiConfig: []

# Labels to apply to all resources
commonLabels: {}

# Set logging level for csi containers.
# Supported values from 0 to 5. 0 for general useful logs,
```

```
# 5 for trace level verbosity.
# logLevel is the variable for CSI driver containers's log level
logLevel: 5
# sidecarLogLevel is the variable for Kubernetes sidecar container's log level
sidecarLogLevel: 1

nodeplugin:
  name: nodeplugin
  # if you are using ceph-fuse client set this value to OnDelete
  updateStrategy: RollingUpdate
  podSecurityPolicy:
    enabled: true
  # set user created priorityclassName for csi plugin pods. default is
  # system-node-critical which is highest priority
  priorityClassName: system-node-critical

  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: 8091

    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
```

```
    ## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips
    ##
    externalIPs: []

    loadBalancerIP: ""
    loadBalancerSourceRanges: []

## Reference to one or more secrets to be used when pulling images
##
imagePullSecrets: []
# - name: "image-pull-secret"

profiling:
  enabled: false

registrar:
  image:
    repository: registry.k8s.io/sig-storage/csi-node-driver-registrar
    tag: v2.9.1
    pullPolicy: IfNotPresent
  resources: {}

plugin:
  image:
    repository: quay.io/cephcsi/cephcsi
    tag: v3.10.2
    pullPolicy: IfNotPresent
  resources: {}

nodeSelector: {}

tolerations: []

affinity: {}

# Set to true to enable Ceph Kernel clients
# on kernel < 4.17 which support quotas
# forcecephkernelclient: true

# common mount options to apply all mounting
```

```
  # example: kernelmountoptions: "recover_session=clean"
  kernelmountoptions: ""
  fusemountoptions: ""

provisioner:
  name: provisioner
  replicaCount: 1
  podSecurityPolicy:
    enabled: true
  strategy:
    # RollingUpdate strategy replaces old pods with new ones gradually,
    # without incurring downtime.
    type: RollingUpdate
    rollingUpdate:
      # 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
    ## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips
    ##
    externalIPs: []

    loadBalancerIP: ""
    loadBalancerSourceRanges: []

## Reference to one or more secrets to be used when pulling images
##
imagePullSecrets: []
# - name: "image-pull-secret"

profiling:
  enabled: false

provisioner:
  image:
    repository: registry.k8s.io/sig-storage/csi-provisioner
    tag: v3.6.2
    pullPolicy: IfNotPresent
  resources: {}
  ## For further options, check
  ## https://github.com/kubernetes-csi/external-provisioner#command-line-options
  extraArgs: []

# set metadata on volume
```

```
  setmetadata: true

  resizer:
    name: resizer
    enabled: true
    image:
      repository: registry.k8s.io/sig-storage/csi-resizer
      tag: v1.9.2
      pullPolicy: IfNotPresent
    resources: {}
    ## For further options, check
    ## https://github.com/kubernetes-csi/external-resizer#recommended-optional-arguments
    extraArgs: []

  snapshotter:
    image:
      repository: registry.k8s.io/sig-storage/csi-snapshotter
      tag: v6.3.2
      pullPolicy: IfNotPresent
    resources: {}
    ## For further options, check
    ## https://github.com/kubernetes-csi/external-snapshotter#csi-external-snapshotter-
sidecar-command-line-options
    extraArgs: []

  nodeSelector: {}

  tolerations: []

  affinity: {}

# readAffinity:
# Enable read affinity for CephFS subvolumes. Recommended to
# set to true if running kernel 5.8 or newer.
# enabled: false
# Define which node labels to use as CRUSH location.
# This should correspond to the values set in the CRUSH map.
# NOTE: the value here serves as an example
# crushLocationLabels:
#   - topology.kubernetes.io/region
```

```
#   - topology.kubernetes.io/zone

# Mount the host /etc/selinux inside pods to support
# selinux-enabled filesystems
selinuxMount: true

storageClass:
  # Specifies whether the Storage class should be created
  create: true
  name: csi-cephfs-sc
  # Annotations for the storage class
  # Example:
  # annotations:
  #   storageclass.kubernetes.io/is-default-class: "true"
  annotations: {}

  # String representing a Ceph cluster to provision storage from.
  # Should be unique across all Ceph clusters in use for provisioning,
  # cannot be greater than 36 bytes in length, and should remain immutable for
  # the lifetime of the StorageClass in use.
  clusterID: 56db00e1-912c-4ac1-9d1a-1f4194c55834
  # (required) CephFS filesystem name into which the volume shall be created
  # eg: fsName: myfs
  fsName: cephfs
  # (optional) Ceph pool into which volume data shall be stored
  # pool: <cephfs-data-pool>
  # For eg:
  # pool: "replicapool"
  pool: "cephfs.cephfs.data"
  # (optional) Comma separated string of Ceph-fuse mount options.
  # For eg:
  # fuseMountOptions: debug
  fuseMountOptions: ""
  # (optional) Comma separated string of Cephfs kernel mount options.
  # Check man mount.ceph for mount options. For eg:
  # kernelMountOptions: readdir_max_bytes=1048576,norbytes
  kernelMountOptions: ""
  # (optional) The driver can use either ceph-fuse (fuse) or
  # ceph kernelclient (kernel).
  # If omitted, default volume mounter will be used - this is
```

```
  # determined by probing for ceph-fuse and mount.ceph
  # mounter: kernel
  mounter: ""
  # (optional) Prefix to use for naming subvolumes.
  # If omitted, defaults to "csi-vol-".
  # volumeNamePrefix: "foo-bar-"
  volumeNamePrefix: ""
  # The secrets have to contain user and/or Ceph admin credentials.
  provisionerSecret: csi-cephfs-secret
  # If the Namespaces are not specified, the secrets are assumed to
  # be in the Release namespace.
  provisionerSecretNamespace: ""
  controllerExpandSecret: csi-cephfs-secret
  controllerExpandSecretNamespace: ""
  nodeStageSecret: csi-cephfs-secret
  nodeStageSecretNamespace: ""
  reclaimPolicy: Delete
  allowVolumeExpansion: true
  mountOptions: []
  # Mount Options
  # Example:
  # mountOptions:
  #   - discard

secret:
  # Specifies whether the secret should be created
  create: true
  name: csi-cephfs-secret
  annotations: {}
  # Key values correspond to a user name and its key, as defined in the
  # ceph cluster. User ID should have required access to the 'pool'
  # specified in the storage class
  adminID: admin
  adminKey: AQDpPctl9T9ZHhAAktyT6vNlGkSE3/rfqnkxKA==

# This is a sample configmap that helps define a Ceph configuration as required
# by the CSI plugins.
# Sample ceph.conf available at
# https://github.com/ceph/ceph/blob/master/src/sample.ceph.conf Detailed
# documentation is available at
```

```
# https://docs.ceph.com/en/latest/rados/configuration/ceph-conf/
cephconf: |
  [global]
    auth_cluster_required = cephx
    auth_service_required = cephx
    auth_client_required = cephx

    # ceph-fuse which uses libfuse2 by default has write buffer size of 2KiB
    # adding 'fuse_big_writes = true' option by default to override this limit
    # see https://github.com/ceph/ceph-csi/issues/1928
    fuse_big_writes = true

# Array of extra objects to deploy with the release
extraDeploy: []

################################################################
# Variables for 'internal' use please use with caution! #
################################################################

# The filename of the provisioner socket
provisionerSocketFile: csi-provisioner.sock
# The filename of the plugin socket
pluginSocketFile: csi.sock
# kubelet working directory,can be set using `--root-dir` when starting kubelet.
kubeletDir: /var/lib/kubelet
# Name of the csi-driver
driverName: cephfs.csi.ceph.com
# Name of the configmap used for state
configMapName: ceph-csi-config
# Key to use in the Configmap if not config.json
# configMapKey:
# Use an externally provided configmap
externallyManagedConfigmap: false
# Name of the configmap used for ceph.conf
cephConfConfigMapName: ceph-config
```

применяем-проверяем

```
root@node1:~/cephfs# helm upgrade -i ceph-csi-cephfs ceph-csi/ceph-csi-cephfs -f cephfs.yml -n
ceph-csi-cephfs --create-namespace
Release "ceph-csi-cephfs" does not exist. Installing it now.
NAME: ceph-csi-cephfs
LAST DEPLOYED: Thu Feb 15 18:58:51 2024
NAMESPACE: ceph-csi-cephfs
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Examples on how to configure a storage class and start using the driver are here:
https://github.com/ceph/ceph-csi/tree/v3.10.2/examples/cephfs

#### test

root@node1:~/cephfs# kubectl apply -f cephfs-claim.yml
persistentvolumeclaim/gimme-pvc created

# ypa!
root@node1:~/cephfs# kubectl get pvc
NAME        STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS    VOLUMEATTRIBUTESCLASS   AGE
gimme-pvc   Bound    pvc-5d0a4e00-1ace-4b1f-83b8-900340e63999   1Gi        RWX            csi-
cephfs-sc   <unset>                 2s

root@node1:~/cephfs# cat cephfs-claim.yml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gimme-pvc
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-cephfs-sc
```

Revision #18
Created 2023-12-07 22:26:18 MSK by Ivan
Updated 2026-07-09 11:12:54 MSK by Ivan