

# Ceph 向导

- Ceph开源项目地址：
  - <https://github.com/ceph/ceph>
- Ceph官方开发的相关项目总览：
  - <https://github.com/ceph>
- 推荐部署方式：
  - 早期版本（Nautilus 14.X.X版本或更早）：
    - ceph-deploy（使用SSH的方式进行部署）
  - 现代版本（Octopus 15.X.X及之后的版本）：
    - cephadm（新的部署工具）
    - ceph-container（利用容器方式部署）：<https://github.com/ceph/ceph-container>
    - ceph-ansible（利用Ansible剧本部署）：<https://github.com/ceph/ceph-ansible>
    - ceph-salt（利用SaltStack部署）：<https://github.com/ceph/ceph-salt>
    - ceph-chef（利用Chef菜谱部署）：<https://github.com/ceph/ceph-chef>

# 硬件要求

- CPU与内存需求

服务	CPU	内存	存储容量	理由
MDS	4C+	每进程1G+	每进程1M+	快速寻址，元数据映射到缓存
OSD	每服务1~2C	每TB数据~1G	独立块设备	支撑RADOS服务、执行CRUSH MAP、维护副本，数据重构时有更多内存需求
MON	每服务1~2C	每进程1G+	每进程10G+	存算融合架构，根据计算需求调整

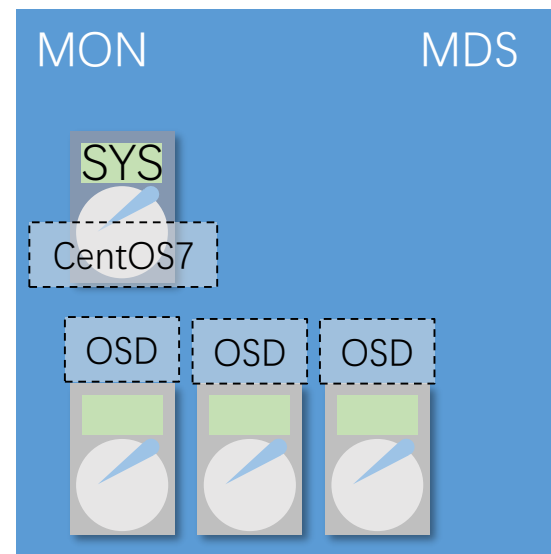
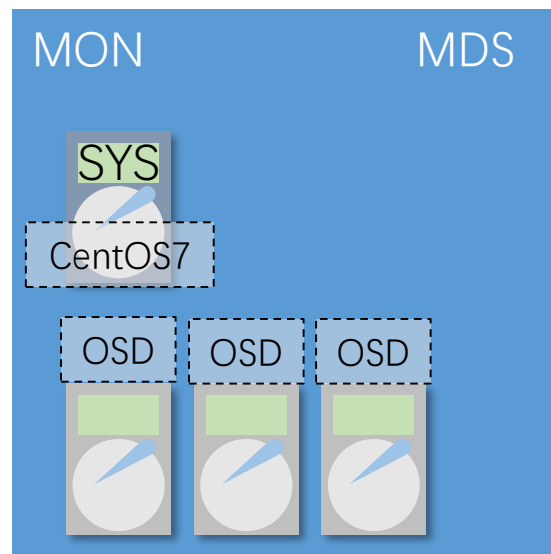
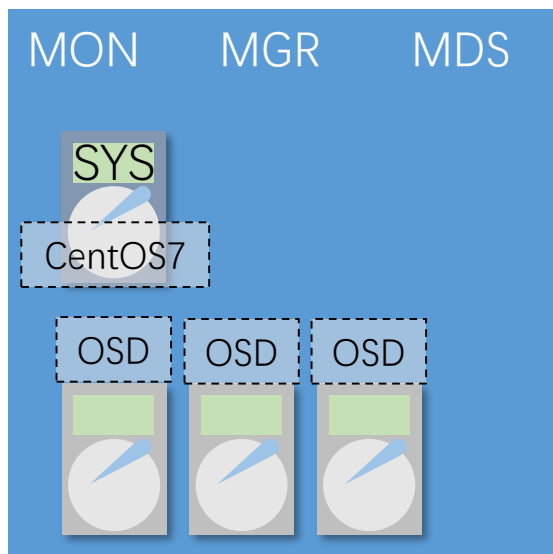
- 存储介质需求

- 建议分别使用独立的硬盘承载：操作系统和软件，单个OSD服务，WAL预写日志
- 按需选择硬盘的大小和个数，基于成本考量，建议选择大容量硬盘的同时避免边际效应
- 可选择SSD作为承载OSD的预写日志或CephFS元数据的存储介质，以达到更优的性价比，但需要注意：
  - 并发写性能（IPS）：WAL涉及写密集语义，廉价SSD的高负载并发写性能可能较低
  - 顺序写：WAL也有一定的顺序写需求，在承载多OSD的WAL时要考虑顺序写的极限

- 网络需求

- 存储前后端服务非常依赖网络，因此至少需要2xGE网络，推荐使用10Gb、25Gb、40Gb或100Gb传输速率的设备组网\*

# 基础架构



# 系统基本配置（单节点）

1/2

- 设置防火墙
  - systemctl stop firewalld
  - systemctl disable firewalld
- 关闭selinux
  - setenforce 0
  - sed -i 's/enforcing/disabled/' /etc/selinux/config
- 设置文件描述符
  - echo 'ulimit -SHn 102400' >> /etc/rc.local
  - cat >> /etc/security/limits.conf << EOF
    - \* soft nofile 65535
    - \* hard nofile 65535
  - EOF
- 内核参数优化
  - cat >> /etc/sysctl.conf << EOF
    - kernel.pid\_max = 4194303
    - vm.swappiness = 0
  - EOF

##/etc/profile亦可

##用户打开最大进程数,  $2^{22}-1$   
##关闭交换分区


# 系统基本配置

2/2

- 配置yum仓库\*
  - `cd /etc/yum.repos.d/`
  - `rm -f ./*.repo` ##可以先确定网络仓库源，之后再按需操作
  - `curl -O https://mirrors.aliyun.com/repo/Centos-7.repo`
  - `yum clean all`
  - `yum makecache` ##也可改用yum list
  - `yum -y install epel-release.noarch` ##Extra Packages for Enterprise Linux
  - `yum -y install vim net-tools bash-completion unzip sshpass wget`
- 调优，按需选择
  - `echo "8192" > /sys/block/sdX/queue/read_ahead_kb` ##预读策略，预读大小
  - `echo "deadline" > /sys/block/sdX/queue/scheduler` ##适用于HDD的IO调度
  - `echo "noop" > /sys/block/sdX/queue/scheduler` ##适用于SSD的IO调度
- 安装时间同步软件chrony（注意部分系统可能已经内置该软件）
  - `yum -y install chrony`

# 节点预配置

1/3

- 配置hosts，使节点间可通过主机名互访（适用于无DNS服务场景）
  - vim /etc/hosts
    - <IP> <HostName>  hosts参考命令 ##插入每个节点的地址+主机名，所有节点都要配置\*  
##IP参考后续的规划表，或预先自行规划
- 创建用于部署Ceph的用户
  - useradd cephuser ##用户名可自定义，但不要使用ceph\*
  - passwd cephuser ##在生产环境中，密码必须尽量复杂
- 确保各 Ceph 节点上新创建的用户都有 sudo 权限
  - echo "cephuser ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/cephuser
  - sudo chmod 0440 /etc/sudoers.d/cephuser
  - su - cephuser ##切换到该用户，后续命令将使用此用户执行
  - sudo -l ##查看当前用户在sudoers配置文件中的权限
  - sudo id ##测试，如果后续碰到权限问题请在命令前加sudo
- 额外克隆两个节点，若资源足够，建议使用完整克隆

# 节点预配置

2/3

- 根据规划表配置IP和主机名（主机名和IP可自行规划）

主机名	NIC1（存储前端，要求能通外网）	NIC2（存储后端）
ceph01	192.168.127.101/24	172.18.1.101/24
ceph02	192.168.127.102/24	172.18.1.102/24
ceph03	192.168.127.103/24	172.18.1.103/24

- 配置主机名和网络（sudo）
  - hostnamectl set-hostname XXX\*
  - nmcli con mod ensXXX ipv4.addresses x.x.x.x/xx ipv4.method manual ipv4.dns x.x.x.x ipv4.gateway x.x.x.x autoconnect yes && nmcli con up ensXXX
- 配置免密互信，使节点间可直接登录，为之后Ceph-Deploy批量配置作准备\*
  - ssh-keygen ##按3次回车在默认路径生成无口令的密钥对
  - ssh-copy-id cephuser@<hostname> ##将公钥拷贝到该主机；需输入目标主机密码，要向所有节点操作以达成互信，完成后输入ssh <hostname>测试\*
- 所有节点生成Ceph yum源文件
  - cd /etc/yum.repos.d/
  - 输入文件中的命令  Ceph\_yum.txt (网易源和清华源二选一，注意不要全部执行)
  - scp /etc/yum.repos.d/ceph.repo <hostname>:/etc/yum.repos.d/ ##拷贝至目标主机，需要依次拷贝到其它所有节点；记得重建仓库缓存

# 节点预配置

3/3

- 指定其中一台作为时钟源服务器，配置chrony (sudo)
  - vim /etc/chrony.conf
    - 注释掉公网的时间同步服务器域名
    - 插入 **【server <IP> iburst】**，此处<IP>为主节点IP，此操作每个节点都要执行
    - 插入 **【allow <IP段/MASK>】**，<IP>为集群所在的网络地址（网段），**仅主节点**执行
    - 插入 **【local stratum 10】**，设置本地同步级别为10（1-15）\*，而且即便本机时间没有向上级同步，也允许客户端向本地获取时间，**仅主节点**执行
  - systemctl restart chronyd ##重启chrony服务
  - chronyc sources -v ##MS项的值为^\*时同步即为同步成功\*
- 在所有节点安装Ceph-Deploy (v2.0.1) (sudo)
  - yum list|grep ceph-deploy ##检查Ceph-Deploy版本是否为2.0.1
  - yum install -y ceph-deploy ##安装Ceph-Deploy
- 在主节点中创建一个目录，用于存放Ceph生成的配置文件、密钥对
  - mkdir ceph\_cluster && cd ceph\_cluster ##这里将Ceph01作为主节点，后续部署操作将在此节点，以ceph\_cluster为工作目录执行ceph-deploy命令



# ceph-deploy命令概述

常用命令	描述
new	开始部署一个新建ceph存储集群，并生成CLUSTER.conf集群配置文件和keyring认证文件
install	在远程主机上安装ceph相关的软件包，可以通过--release指定安装的版本
rgw	管理RGW守护进程
mgr	管理MGR守护进程
mds	管理MDS守护进程
mon	管理MON守护进程
gatherkeys	收集用于配置新节点的身份验证密钥
disk	管理远程主机的硬盘
osd	在远程主机上准备数据磁盘；即为指定主机的指定数据盘绑定OSD服务
repo	仓库定义配置管理
admin	分发集群配置文件和admin认证文件到远程主机
config	分发集群配置文件ceph.conf到远程主机，或从远程主机拷贝此文件
uninstall	删除远程主机的Ceph软件包
purgedata	删除远程主机的Ceph数据，包括/var/lib/ceph和/etc/ceph下的内容
purge	删除远程主机的Ceph软件包和数据，相当于uninstall+purgedata
forgetkeys	遗忘（删除）本地主机所有的验证keyring，包括client.admin、monitor、bootstrap等认证文件
pkg	管理远程主机的安装包
calamari	安装和配置 Calamari 节点；它是一个web监控平台；需要预配置包含该软件包的软件仓库

# 集群配置

1/3

- 在主节点上创建Ceph集群（创建MON服务，并在当前目录生成配置文件、mon密钥等）
  - `ceph-deploy new <hostname1> <hostname2> ...*`
    - 若此处报错“ImportError: No module named pkg\_resources”则需重装pip；此问题通常是Python升级至2.7产生的，pip是Python中的安装工具，推荐解决方法如下：\*
      - `sudo yum -y install wget && wget https://bootstrap.pypa.io/pip/2.7/get-pip.py`
      - `python get-pip.py` ##若报错则多尝试几遍 
      - 解决之后重新创建集群 ##另提供备用解决办法 备选解决方
- 在集群配置目录下的集群配置文件【ceph.conf】的global段中定义网段
  - `public_network = 192.168.127.0/24` ##用于客户端访问集群服务
  - `cluster_network = 172.18.1.0/24` ##用于内部数据同步（如OSD数据复制、心跳检测）
  - 修改集群块服务副本数为“2”（可选，针对低配实验环境）
    - `osd pool default size = 2` ##在ceph.conf的global段中添加
- 推送配置到所有节点并重启服务
  - `ceph-deploy --overwrite-conf config push ceph01 ceph02 ceph03`
  - `sudo systemctl restart ceph-*.target`

# 集群配置

2/3

- 为每个节点安装Ceph

- `ceph-deploy install --no-adjust-repos --nogpgcheck ceph01 ceph02 ceph03` ##  
批量安装，同时安装ceph-radosgw，--no-adjust-repos: 安装过程中不调整为官方源

或

- `sudo yum install -y ceph` ##手动为所有节点执行，并行操作效率更高  
完成后输入`ceph -v`查询版本

- 在主节点上开始部署（注意参考架构图）

- `ceph-deploy mon create-initial` ##初始化mon，生成检测集群所需的密钥\*
  - `ceph-deploy admin <hostname1> <hostname2> ...` ##分发集群配置和admin认证文件
    - `sudo setfacl -m u:cephduser:rw /etc/ceph/ceph.client.admin.keyring` ##生成的客户端管理员密钥需要对主节点的集群部署用户开放权限

完成后可在mon节点输入`ceph -s`查询集群状态；也可在mon节点查找带mon关键字的进程和其侦听的端口，默认端口为6789；可为mon配置DNS轮询或3层负载均衡，但通常mon压力较小，不会成为性能瓶颈

- `ceph-deploy mgr create <hostname1> ...` ##部署mgr，用于监测集群

★请确保每次执行ceph-deploy命令时都处于正确的工作目录；另外执行此命令不需要sudo提权

# 集群配置

3/3

- 在主节点上开始部署（请确保集群后端网络正常）
  - `ceph-deploy disk list <hostnameX>` ##列出该节点的块设备
  - `ceph-deploy disk zap <hostnameX> /dev/sdX` (可选) ##擦除指定主机的指定磁盘
  - `ceph-deploy osd create --data /dev/sdX <hostnameX>` ##为每个主机的每个业务硬盘创建OSD，要在主节点上为每个节点的每个业务硬盘执行\*
- 安装完毕，检查集群状态
  - 输入`ceph -s`，检查集群状态信息；或输入`ceph df`，检查集群容量信息
    - 集群信息：可能会有一个告警，因为集群运行在不安全模式下，执行`ceph config set mon auth_allow_insecure_global_id_reclaim false`命令关闭不安全模式即可
    - 服务信息：包含mon、mgr、mds、osd、rgw，且数量符合节点配置
    - 数据信息：存储池、PG、对象的数量，已用和可用的容量、PG状态等符合实配
  - 输入`ceph osd tree`，可查询到集群各节点的osd服务和它们对应的硬盘
  - 若其它节点中没有密钥，可向主节点手动请求&同步密钥（可选）
    - `cd /etc/ceph` ##以此目录为工作目录，输入pwd可查询
    - `ceph-deploy gatherkeys <hostname>` ##向主节点同步；其它所有节点都要执行
  - 完成Ceph部署，建议重新登录用户以刷新环境变量

# 其它说明

- 如果安装过程中碰到难以解决的阻碍，可以尝试执行以下命令清理环境，然后重新部署Ceph
  - `ceph-deploy purgedata <hostname1> [.....]`
  - `ceph-deploy forgetkeys`
- 还可以执行下列命令，将本地安装包一并清理
  - `ceph-deploy purge <hostname1> [.....]`
- 如果需要删除某OSD或移除对应的块设备，必须遵循以下流程：
  - 剔除设备：`ceph osd out <osd-num>`
    - 不允许批量操作，除非存储池中没有任何重要数据；
  - 停止服务：`sudo systemctl stop ceph-osd@<osd-num>`
  - 擦除数据（可选）：`ceph osd purge <osd-num> --yes-i-really-mean-it`
  - 移出 crush map：`ceph osd crush remove <name>`
  - 删除 ceph auth 记录：`ceph auth del osd.<osd-num>`
  - 移出 osdmap：`ceph osd rm <osd-num>`
  - 手动删除 ceph.conf 中的记录（如果有）
    - 必须等待集群自愈后（PG 状态为 active+clean）才能继续删除\*

# 扩展Ceph集群

- 扩展Ceph-mon节点（参考命令）

- `yum install ceph-common ceph-mon` ##仅安装ceph通用组件和mon组件
- `ceph-deploy mon add <hostname>` ##为集群添加mon节点\*
- `ceph quorum_status [--format json-pretty]` ##查验集群mon节点状态，将返回JSON数据；带上--format json-pretty参数会格式化显示，可读性更佳

- 扩展Ceph-mgr节点（参考命令）

- `yum install ceph-mgr` ##仅安装ceph-mgr组件
- `ceph-deploy mgr create <hostname>` ##为集群添加mgr节点
- `ceph-deploy admin <hostname>` ##分发集群配置和admin认证文件
- `ceph -s` ##查验集群mgr节点状态

- 手动复制集群配置文件（包括认证配置文件）（可选）

- `scp -p /etc/ceph/ceph.conf <IP>:/etc/ceph/` ##IP替换为节点IP或主机名
- `scp -p /etc/ceph/ceph.client.<user>.keyring <IP>:/etc/ceph/` ##复制密钥  
到客户端，生产环境下建议限制客户端访问密钥的权限

# 简单测试

- 管理Ceph-RBD存储池（在主节点工作目录下操作）\*
  - `ceph osd pool create <poolName> <pgNum> [pgpNum]` ##创建存储池，pg数量与osd数量正相关，此处建议设为16-64；pgp数量通常与pg一致，可缺省\*
  - `ceph osd pool ls [detail]` ##列出存储池，带上detail表示列出细节\*
- 利用rados命令直接使用存储池
  - `rados put <oid> <fileName> -p <poolName>` ##指定oid并上传文件对象到存储池
  - `rados ls -p <poolName>` ##列出该存储池中的对象
  - `ceph osd map <poolName> <oid>` ##查询存储池中指定对象的映射信息
  - `rados get <oid> -p <poolName> <fileName>` ##从存储池中获取对象并保存到本地文件系统
  - `rados rm <oid> -p <poolName>` ##从存储池中删除对象
  - `ceph osd pool delete <poolName>` ##删除存储池（该操作可能受阻）

# 删除存储池

- 设置为允许删除存储池
  - `ceph config set mon mon_allow_pool_delete true`
- 删除存储池
  - `ceph osd pool delete <poolName> <poolName> --yes-i-really-really-mean-it`

需要按照要求重复两次存储池的名字，并且附带确认信息



# RBD业务测试

1/2

- 创建Ceph-RBD存储池（在主节点工作目录下操作）
  - `ceph osd pool create <poolName> <pgNum> <pgpNum>` ##pg数量与osd数量正相关，此处建议设为16-64；pgp数量通常与pg一致，可缺省
- 将存储池关联为RBD应用类型
  - `ceph osd pool application enable <poolName> <appName>` ##应用名可选cephfs、rbd、rgw，此处为rbd
  - `rbd pool init -p <poolName>` ##初始化存储池（只针对rbd类型）
- 在存储池中创建1G容量映像
  - `rbd create <poolName/imgName> -s 1G` ##后续可用rbd resize扩容\*
  - `rbd info <poolName/imgName>` ##查询映像信息
- 若要在客户端访问RBD，则需要安装ceph-common，客户端和集群块设备之间并非使用iSCSI协议沟通，而是通过Linux内核（所以客户端不能为Windows）
  - 首先配置yum，参考系统配置2/2和集群配置2/4
  - 安装ceph-common
    - `yum install -y ceph-common.x86_64`

# RBD业务测试

2/2

- 在主节点复制集群配置文件到客户端，客户端利用其中的信息访问集群（sudo）
  - `scp -p /etc/ceph/ceph.conf <IP>:/etc/ceph/` ##IP替换为客户端IP或主机名
  - `scp -p /etc/ceph/ceph.client.admin.keyring <IP>:/etc/ceph/` ##复制密钥到客户端，生产环境下建议限制客户端访问密钥的权限
  - 完成后在客户端可输入`ceph -s`查询集群状态
  - `ceph osd pool ls [detail]` ##列出存储池，带上detail表示列出细节\*
- `rbdm map <poolName/imgName>` ##映射映像，若报错参考以下部分
  - 若客户端的Linux内核版本过旧，则需禁用存储池的部分功能，具体可参考报错回显内容
  - `rbdm info <poolName/imgName>` ##查询&确认映像的信息
  - `rbdm feature disable <poolName/imgName> object-map fast-diff deep-flatten` ##调整功能
  - `rbdm info <poolName/imgName>` ##再次查询&确认映像的信息，确认无误后重新映射
- `lsblk`应该可查询新块设备，之后正常格式化挂载使用
- 自动挂载（可选）\*
  - `vim /etc/ceph/rbdmap`  
#末尾插入<poolName/imgName> id=admin,keyring=/etc/ceph/ceph.client.admin.keyring
  - `systemctl enable rbdmap` ##启用rbdmap服务：负责把RBD镜像映射为本地块设备
  - `vim /etc/fstab` ##插入UUID=<uuid> <dir> <fsType> defaults,\_netdev 0 0

# RGW业务测试

1/3

- 对象存储没有目录层级结构，而是采用唯一的全局标识符（如 bucket-name/object-key）访问数据，数据以“对象”形式存储，每个对象包含数据、元数据和唯一标识符（如UUID或哈希值）
- RGW提供RESTful API，客户端可通过标准的 HTTP/HTTPS 接口完成数据的增删改查操作
- 部署RGW业务前必须先部署radosgw服务；选择合适的节点，执行以下命令：

- `yum install -y ceph-radosgw` ##安装rgw, 用于对象存储服务
- `ceph-deploy [--overwrite-conf] rgw create <hostname>` ##在主节点操作, 填写安装了rgw的主机名; 可选项--overwrite-conf: 覆盖式生成新的配置文件

若节点是新部署的，则需要复制集群配置文件

- `scp -p /etc/ceph/ceph.conf /etc/ceph/ceph.client.admin.keyring <IP>:/etc/ceph/` ##在主节点操作, 复制集群配置文件到rgw节点, IP可替换为主机名

完成后可在mon节点输入`ceph -s`查询集群状态；也可在rgw节点查找带radosgw关键字的进程和其侦听的端口，访问rgw节点的该端口能得到对象存储API相关的xml配置信息\*

- 生产环境建议至少部署 3 个 RGW，并通过负载均衡器（如 Nginx、HAProxy）对外提供服务

# RGW业务测试

2/3

- 主节点创建 RADOSGW 用户

- `radosgw-admin user create --uid="testuser" --display-name="Test User"`  
# 记录输出结果中的`access\_key` 和 `secret\_key`:  
# {  
# "user\_id": "testuser",  
# "access\_key": "ABCDEFGHJKLMNOPQRST",  
# "secret\_key": "abcdefghijklmnopqrstuvwxyz0123456789ABCD"  
# }

- 客户端安装和配置awscli:

- `yum install awscli` ##安装awscli, 用于访问S3 API
  - `aws --version` ##查询awscli版本, 验证是否可用
  - `aws configure set aws_access_key_id <access_key>` ##配置密钥ID (必填)
  - `aws configure set aws_secret_access_key <secret_key>` ##配置密钥 (必填)
  - `aws configure set default.region us-east-1` ##配置默认区域名称
  - `aws configure set default.output_format json` ##配置默认输出格式

也可以直接输入aws configure命令, 交互式地输入以上信息

# RGW业务测试

3/3

- 使用 awscli 测试 S3 接口

- export ENDPOINT=http://<RGW节点IP>:7480 ##设置环境变量
- aws --endpoint-url \$ENDPOINT s3 mb s3://<BUCKET> ##创建 Bucket
- echo 'Hello Ceph RGW!' > test.txt
- aws --endpoint-url \$ENDPOINT s3 cp test.txt s3://<BUCKET>/ ##上传文件
- aws --endpoint-url \$ENDPOINT s3 ls s3://<BUCKET> ##列出文件
- aws --endpoint-url \$ENDPOINT s3 cp s3://<BUCKET>/test.txt ./123.txt ##下载文件

如果 Bucket 是公共可读的，可直接用浏览器或 curl 访问，如：

- curl http://<RGW节点IP>:7480/<BUCKET>/test.txt

设置Bucket为公共可读

- aws --endpoint-url \$ENDPOINT s3api put-bucket-acl \
- --bucket <BUCKET> \
- --acl public-read

设置单个Object为公共可读

- aws --endpoint-url \$ENDPOINT s3api put-object-acl \
- --bucket <BUCKET> \
- --key <OBJECT\_NAME> \
- --acl public-read

\*以上操作可能需要等待一定的时间才能生效；访问对象时路径末尾不要加“/”

# CephFS业务测试

1/2

- 部署MDS服务

- `yum install -y ceph-mds` ##安装MDS， 用于CephFS服务
- `ceph-deploy mds create <hostname1> ...` ##在主节点操作， 填写安装了MDS的主机名
- `ceph mds stat` ##查询MDS服务状态， 此时应为已上线（up）+待命（standby）的状态；在配置完必须的存储池和文件系统之后，MDS会转为活动状态（active）

- 生产环境建议部署3个mds服务（1主2备）；如需高性能访问，可双活或多活部署mds，但需内核  $\geq 5.4$  并启用 `ceph fs set <fs_name> max_mds 2`

- CephFS文件系统需要两个存储池，一个用于存储CephFS数据，一个用于存储CephFS元数据

- `ceph osd pool create <cephfs_metadata> <pgNum> <pgpNum>` ##pg建议设为64
- `ceph osd pool create <cephfs_data> <pgNum> <pgpNum>` ##pg建议设为128
- `ceph fs new <cephfsName> <cephfs_metadata> <cephfs_data>` ##创建CephFS
- `ceph fs ls`

# CephFS业务测试

2/2

- 使用ceph-fuse挂载（需要预先安装）
  - `ceph-fuse --keyring /etc/ceph/ceph.client.<user>.keyring --name client.cephfs -m <mon1IP>:6789,<mon2IP>:6789,<mon3IP>:6789 /mnt/cephfs`
  - `echo "id=<user>,keyring=/etc/ceph/ceph.client.<user>.keyring,conf=/etc/ceph/ceph.conf /mnt/cephfs fuse.ceph defaults,_netdev 0 0" >> /etc/fstab` [##自动挂载](#)
- 使用内核客户端（ceph-common）挂载
  - `ceph auth get-key client.<user> -o /etc/ceph/cephfskey` 查询密钥并导出到文件\*
  - `mount -t ceph <mon1IP>:6789,<mon2IP>:6789,<mon3IP>:6789:/ /mnt/cephfs -o name=<user>,secret=<Base64Code>`
  - `echo "<mon1IP>:6789,<mon2IP>:6789,<mon3IP>:6789:/ /mnt/cephfs ceph name=<user>,secretfile=/etc/ceph/cephfskey,noatime,_netdev 0 0" >> /etc/fstab` [##自动挂载](#)

挂载时若使用密钥文件，则文件中必需只有密钥的base64字符串，否则将不能正常映射；  
<user>在本实验中是admin，实际应用中请创建普通用户用来对接业务；  
若要使用ceph-fuse进行挂载，需要预先在client上安装ceph-fuse。

# 清理环境

- `umount /<dir>`
- `rbid unmap <poolName/imgName>`
- `rbid remove <poolName/imgName>`    *##remove换成trash, 则移至回收站*
- `ceph fs rm <fs-name> --yes-i-really-mean-it`    *##移除文件系统\**
- 编辑集群配置文件, 使mon服务可以删除存储池
  - `vim /etc/ceph/ceph.conf`
    - 在<mon>段落内插入 `mon allow pool delete = true`    *##没有MON段落就直接插入*
  - `ceph-deploy --overwrite-conf config push [host<m,n>]`    *##配置文件同步到其它节点*
  - `systemctl restart ceph-mon.target`    *##重启mon, 所有节点执行*
- `ceph osd pool delete <poolName> <poolName> --yes -i-really-really-mean-it` *##删除pool*
- 执行成功即为移除存储池



# 常见问题汇总和处理方法

- mon初始化失败
  - mon地址与集群配置文件ceph.conf中的前端网络public\_network网段不符
  - 主机名解析配置错误
  - 集群配置文件没有推送到其它节点
  - 以上都正确但仍然失败的处理方法：
    - 卸载ceph: `ceph-deploy purge <hostname>`
    - 删除缓存配置: `rm -rf /var/lib/ceph`
    - 之后重新安装ceph后再初始化
- OSD初始化失败
  - 存储后端网络配置错误或与集群配置文件不符
  - 配置文件和认证文件错误
  - 块设备错误 (选择错设备 或 设备有数据未擦除干净)

# 常用查询命令总结

ceph osd pool ls [detail]	列出存储池
ceph osd lspools	
ceph osd pool stats <rbdPoolName>	查询块业务存储池状态
ceph pg stat	查询PG状态
ceph df [detail]	查询集群存储空间信息
ceph osd dump	查询OSD的底层详细信息
ceph osd tree	列出OSD的树状关系图
ceph mon stat	查询mon节点状态
ceph mon dump	查询mon节点的底层详细信息
ceph fs ls	列出集群中的文件系统（CephFS）
ceph -s	查询集群总览信息
ceph -w	持续监察集群总览信息
rbd info <poolName/imgName>	查询映像信息
rados ls -p <poolName>	列出该存储池中的对象
ceph osd map <poolName> <oid>	查询存储池中指定对象的映射信息
ceph quorum_status [--format json-pretty]	查验集群mon节点状态，将返回JSON数据