minio集群扩容

常见的集群扩容方法可分为两类：水平扩容和垂直扩容。

水平扩容：一般指通过增加节点数扩展系统性能；

垂直扩容：则指提升各节点自身的性能，例如增加节点的磁盘存储空间。

直接采用垂直扩容方式扩容MinIO集群的节点磁盘空间，会为集群运行带来若干问题，**官方也并不推荐。**

对等扩容

MinIO分布式集群并**不支持向集群中添加单个节点**并进行**自动调节**的扩容方式，这是因为加入单个节点后所**引发的数据均衡以及纠删组划分等问题会为整个集群带来复杂的调度和处理过程**，并不利于维护。因此，MinIO提供了一种**对等扩容**的方式，即要求**增加的节点数和磁盘数均需与原集群保持对等。**

例如：原集群包含4个节点4块磁盘，则在**扩容时必须同样增加4个节点4块磁盘（或为其倍数）**，以便系统维持相同的数据冗余SLA，从而极大地降低扩容的复杂性。

如上例，在扩容后，MinIO集群并**不会对全部的8个节点进行完全的数据均衡**，而是将原本的4个节点**视作一个区域，新加入的4节点视作另一区域**，当有新对象上传时，集群将依据各区域的**可用空间比例确定存放区域**，在各区域内**仍旧通过哈希算法确定对应的纠删组进行最终的存放。**此外，集群进行一次对等扩容后，还可依据扩容规则继续进行对等扩容，但出于安全性考虑，集群的最大节点数一般不得超过32个。

对等扩容优点：配置操作简单易行，通过一条命令即可完成扩容（注意：推荐使用连续的节点IP，并参照MinIO官网在扩容命令中使用{}）

对等扩容缺点：①扩容需重启；②扩容存在限制，集群节点数一般不超过32个，这是由于MinIO集群通过分布式锁保证强一致性，若集群节点数过大，维护强一致性将带来性能问题。

环境说明

节点IP	集群配置	备注
192.168.251.133	2 磁盘 4 drive	测试环境每个磁盘下创建两个目录作为 drive
192.168.251.88	2 磁盘 4 drive	测试环境每个磁盘下创建两个目录作为 drive
192.168.251.98	2 磁盘 4 drive	测试环境每个磁盘下创建两个目录作为 drive

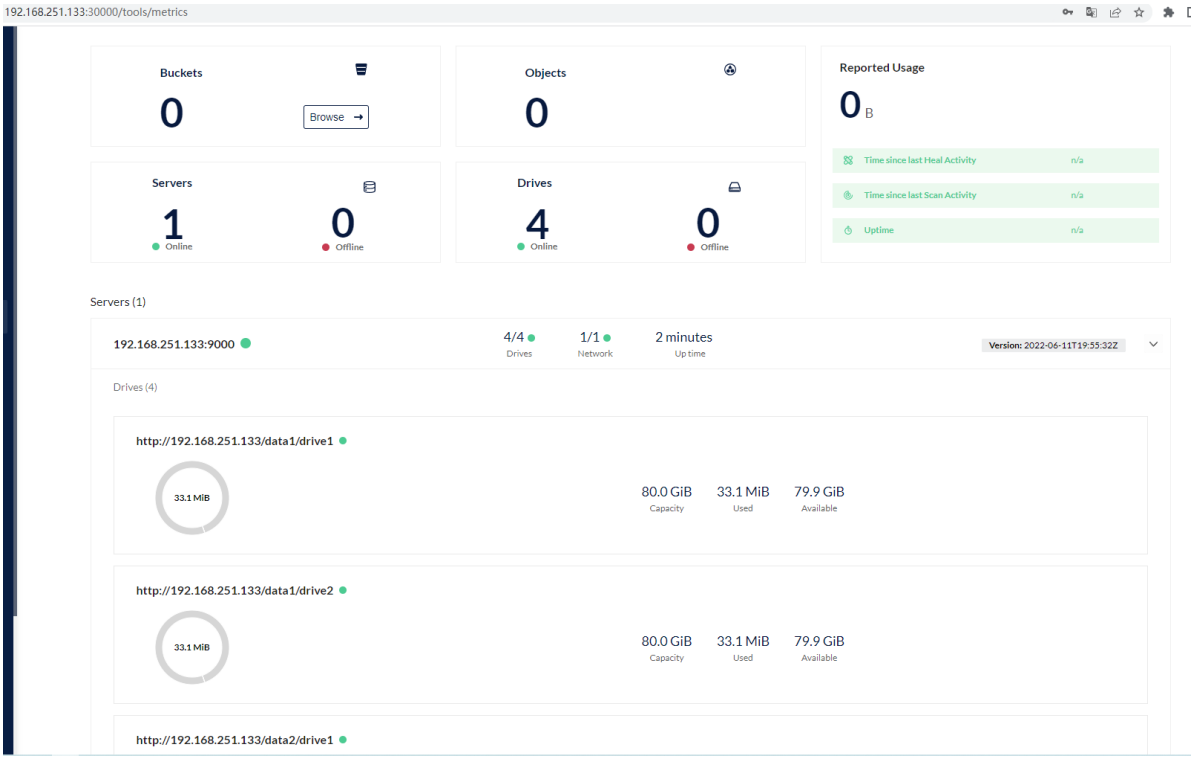
首先有一个minio集群，2磁盘4drive 的最小集群，后续通过对等扩容添加88和98两个集群到第一个集群。

添加节点扩容操作

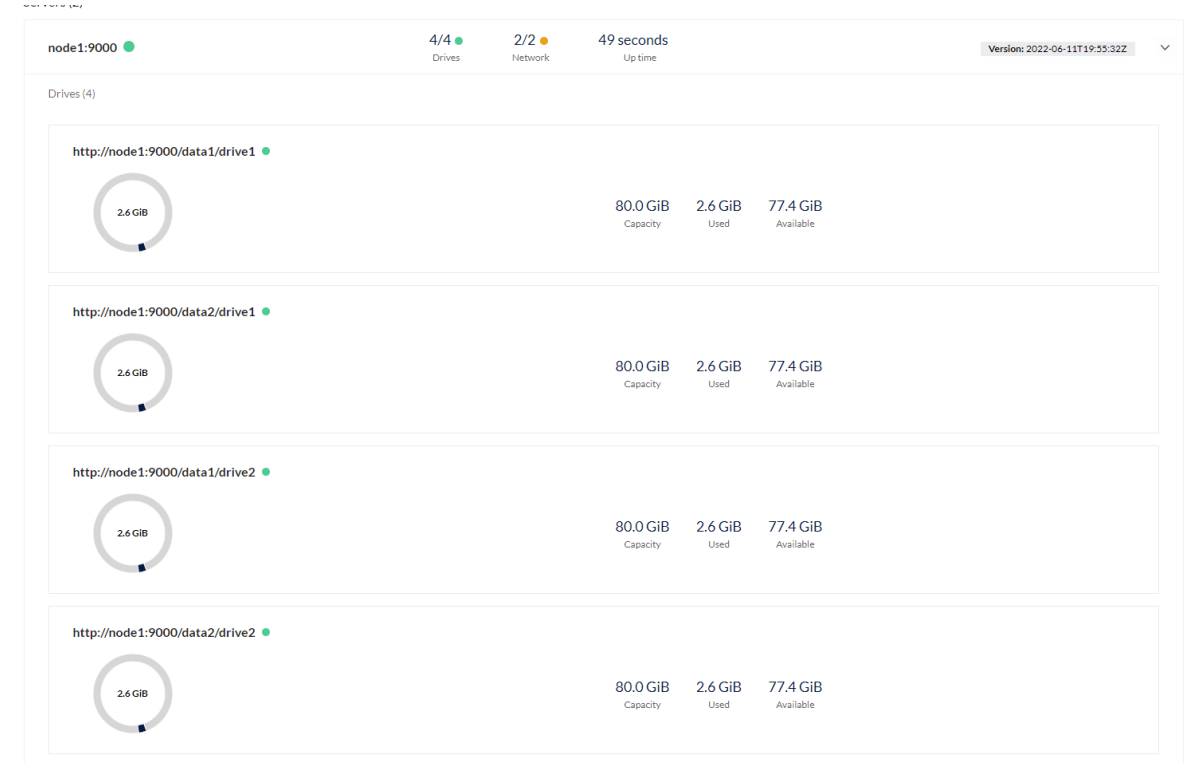
启动第一个节点作为原始集群

```
## 配置hosts
[root@node1 minio]# cat /etc/hosts
192.168.251.133 node1
192.168.251.88 node2
192.168.251.98 node3
```

```
[root@node1 minio]# cat run.sh
#!/bin/bash

export MINIO_ROOT_USER=admin
export MINIO_ROOT_PASSWORD=edoc2@edoc2

/opt/minio/minio server --config-dir /opt/minio/config/ --console-address
":30000"\
    http://node1:9000/data{1...2}/drive{1...2}
```



向node1写入数据



更改原始集群与新集群启动文件

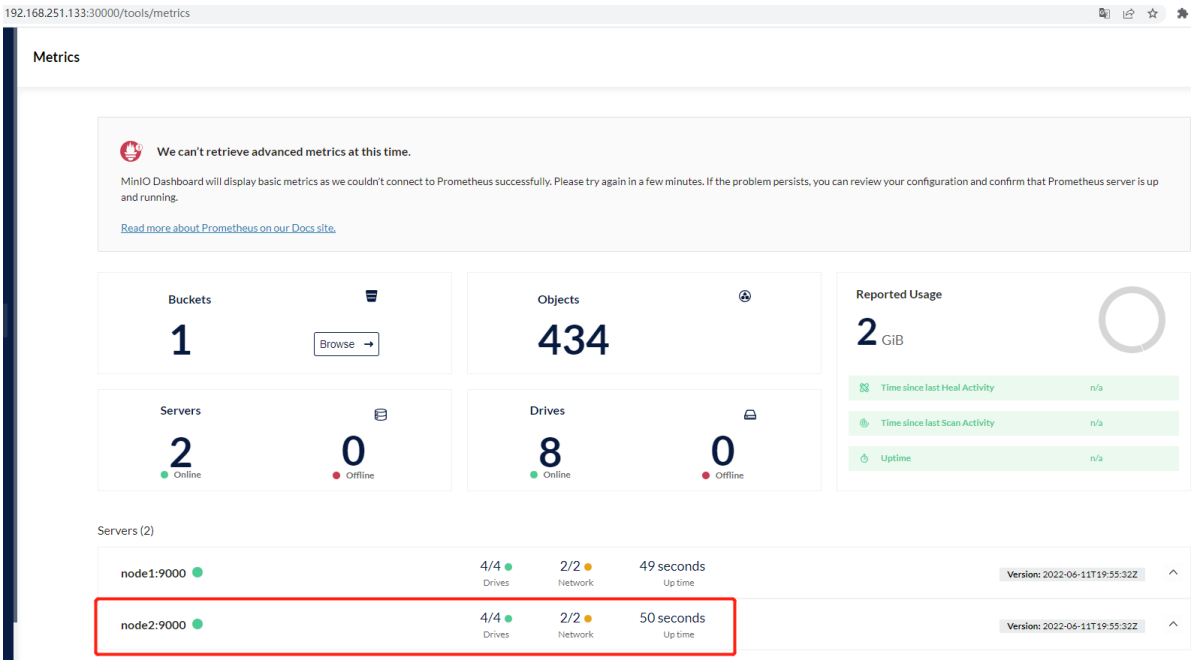
```
[root@node1 minio]# cat run.sh
#!/bin/bash

export MINIO_ROOT_USER=admin
export MINIO_ROOT_PASSWORD=edoc2@edoc2

/opt/minio/minio server --config-dir /opt/minio/config/ --console-address
":30000"\
    http://node1:9000/data{1...2}/drive{1...2} \
    http://node2:9000/data{1...2}/drive{1...2}
```

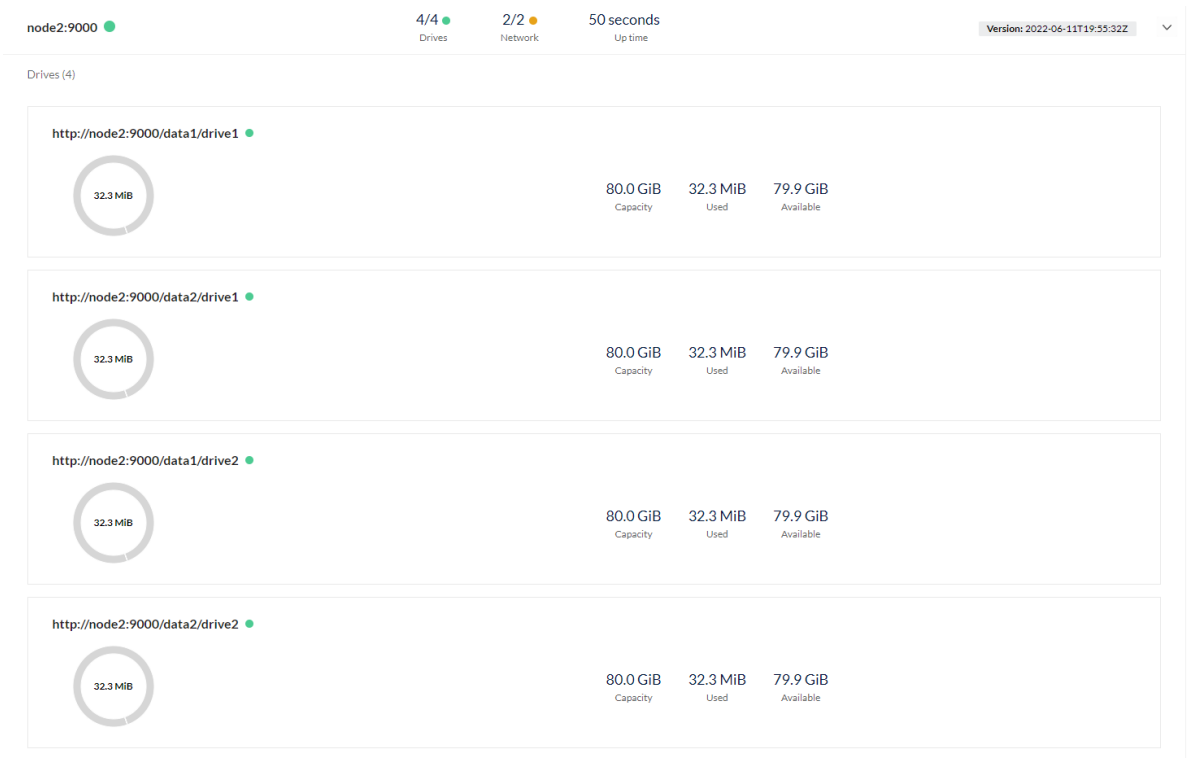
新节点和原有节点都配置为以上配置，添加新的存储节点及drive。

```
[root@node1 minio]# systemctl restart minio       // 所有节点同时重启
[root@node2 minio]# systemctl restart minio
```

官方建议所有的集群节点同时进行重启，因为minio是原子性的操作，不会对应用产生中断影响。

登录管理界面，可以看到新添加的 server



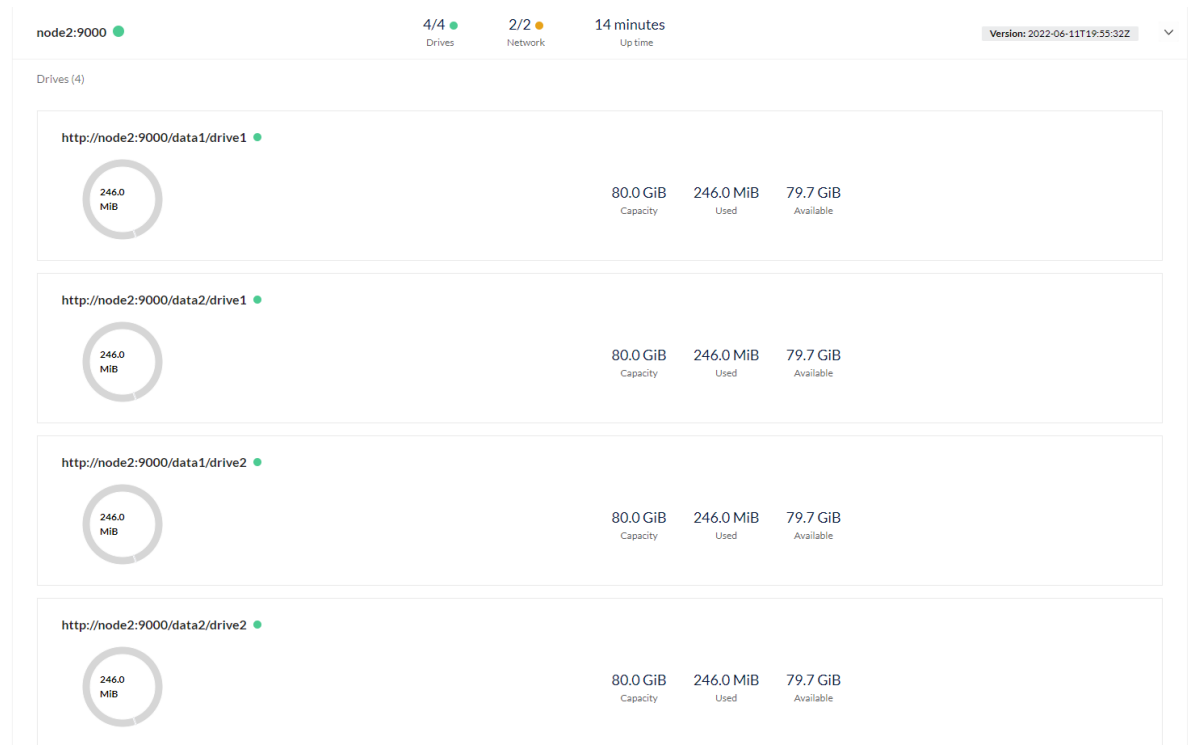


新扩容的 并不会同步数据

```
[root@node1 minio]# mc ls node1
[2022-07-05 15:26:39 CST]     0B warp-benchmark-bucket/
[root@node1 minio]# mc ls node2
[2022-07-05 15:26:39 CST]     0B warp-benchmark-bucket/
[root@node1 minio]# mc mb node2/edoc2
Bucket created successfully `node2/edoc2`.
[root@node1 minio]# mc ls node2
[2022-07-05 15:41:41 CST]     0B edoc2/
[2022-07-05 15:26:39 CST]     0B warp-benchmark-bucket/
[root@node1 minio]# mc ls node1
[2022-07-05 15:41:41 CST]     0B edoc2/
[2022-07-05 15:26:39 CST]     0B warp-benchmark-bucket/
```

创建新桶edoc2 写入数据测试

```
[root@test opt]# warp mixed --host=192.168.251.133:9000 --objects=4000 --obj.size=100kib --access-key admin --secret-key edoc2@edoc2 --bucket edoc2 --autoterm  --analyze.v
```



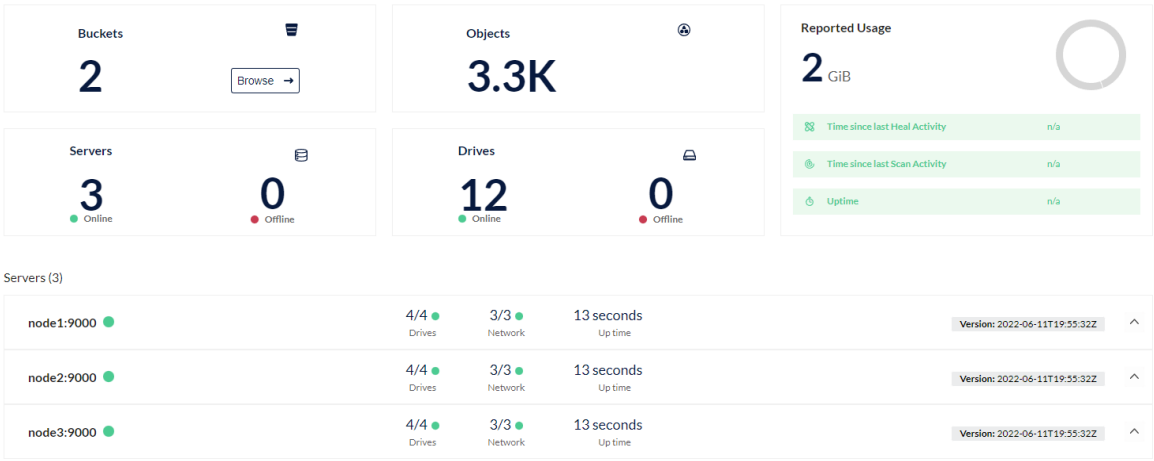
node2 有数据写入，minio会根据磁盘使用量，来确定数据放到哪个drive

扩容第三个节点

```
[root@node1 minio]# cat run.sh
#!/bin/bash

export MINIO_ROOT_USER=admin
export MINIO_ROOT_PASSWORD=edoc2@edoc2

/opt/minio/minio server --config-dir /opt/minio/config/ --console-address
":30000"\
    http://node1:9000/data{1...2}/drive{1...2} \
    http://node2:9000/data{1...2}/drive{1...2} \
    http://node3:9000/data{1...2}/drive{1...2}
```

启动脚本里配置第三个节点的磁盘及drive。

同时重启 三个节点的minio服务。



继续添加磁盘扩容

以已经组成的三个节点的集群，来对每个节点模拟添加磁盘进行扩容。

对node1节点两个磁盘下创建 drive3和drive4目录，模拟添加四块磁盘进行扩容。

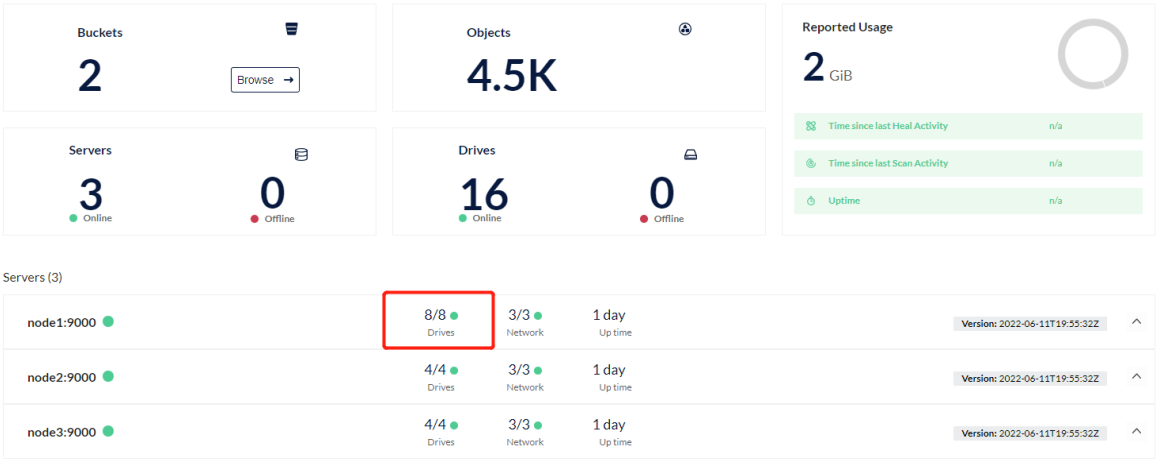
```
[root@node1 minio]# ls /data*/
/data1/:
drive1  drive2  drive3  drive4

/data2/:
drive1  drive2  drive3  drive4
[root@node1 minio]#
```

更改所有节点的启动脚本

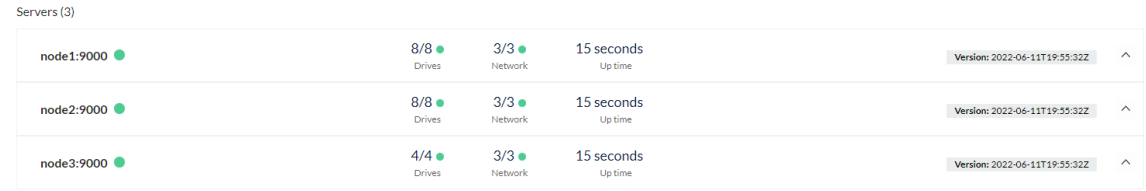
```
[root@node1 minio]# cat run.sh
#!/bin/bash

export MINIO_ROOT_USER=admin
export MINIO_ROOT_PASSWORD=edoc2@edoc2

/opt/minio/minio server --config-dir /opt/minio/config/ --console-address
":30000"\
    http://node1:9000/data{1...2}/drive{1...2} \
    http://node2:9000/data{1...2}/drive{1...2} \
    http://node3:9000/data{1...2}/drive{1...2} \
    http://node1:9000/data{1...2}/drive{3...4}      // 扩容的磁盘在进行添加，三个节点
都更改
```

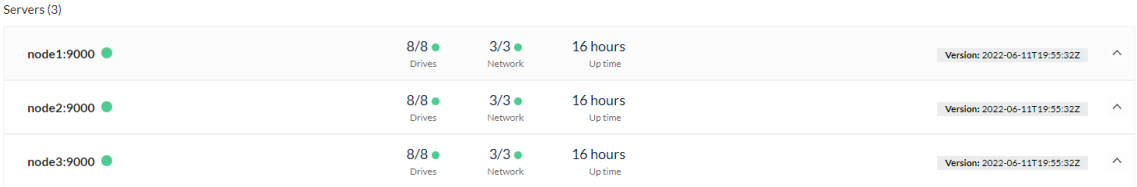
同时重启三个节点的minio服务

```
systemctl restart minio
```



node1的四个磁盘已经添加完成



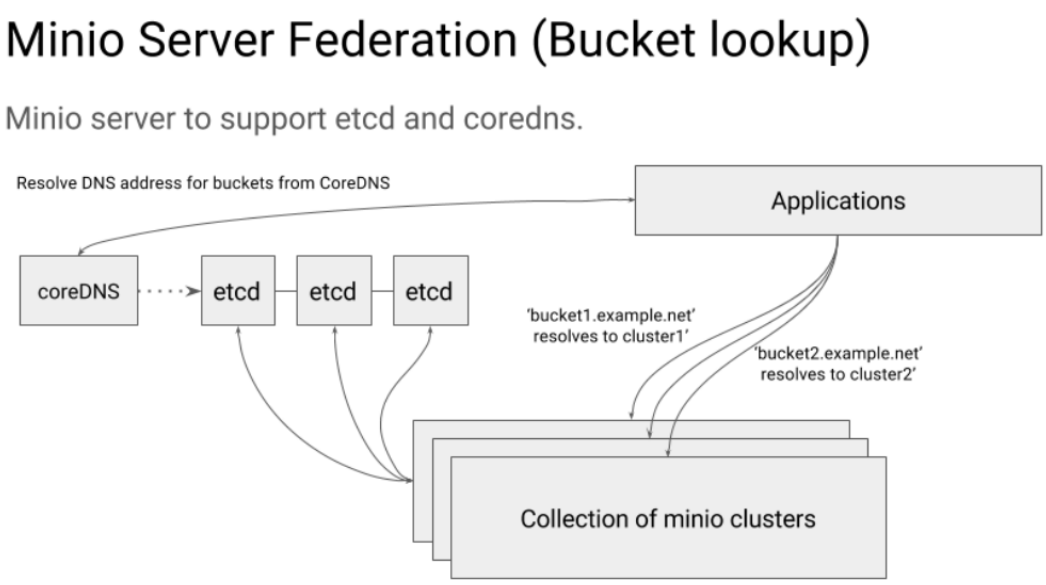


联邦扩容

minio之前提供了一种联邦扩容的方式来扩容集群，但在测试这个版本（RELEASE.2022-06-11T19-55-32Z）官方已经把联邦的这个特性置为deprecated，不在推荐使用，官方文档也只有上面的一种扩容方式。

只是使用这个版本进行联邦扩容的测试**功能依然可用。**

通过引入etcd，将多个MinIO分布式集群在逻辑上组成一个联邦，对外以一个整体提供服务，并提供统一的命名空间。



etcd的作用就是路由寻址的作用，应用访问组成联邦中的任意一个集群，都可以通过查询etcd定位到存放bucket的集群的地址，进而路由到相应的机器访问。

部署etcd集群

192.168.251.133/88/98 三个节点部署etcd集群

```
yum -y install etcd
```

修改etcd配置文件：

```
[root@node1 minio]# cat /etc/etcd/etcd.conf
#[Member]
ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_NAME="etcd1"

#[Clustering]
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://192.168.251.133:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://192.168.251.133:2379"
ETCD_INITIAL_CLUSTER="etcd1=http://192.168.251.133:2380,etcd2=http://192.168.251
.88:2380,etcd3=http://192.168.251.98:2380"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER_STATE="new"
```

```
[root@node2 minio]# cat /etc/etcd/etcd.conf
#[Member]
ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_NAME="etcd2"

#[Clustering]
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://192.168.251.88:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://192.168.251.88:2379"
ETCD_INITIAL_CLUSTER="etcd1=http://192.168.251.133:2380,etcd2=http://192.168.251
.88:2380,etcd3=http://192.168.251.98:2380"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER_STATE="new"
```

```
[root@node3 minio]# cat /etc/etcd/etcd.conf
#[Member]
ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_NAME="etcd3"

#[Clustering]
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://192.168.251.98:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://192.168.251.98:2379"
ETCD_INITIAL_CLUSTER="etcd1=http://192.168.251.133:2380,etcd2=http://192.168.251
.88:2380,etcd3=http://192.168.251.98:2380"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER_STATE="new"
```

```
ETCD_DATA_DIR=数据存放位置
ETCD_LISTEN_PEER_URLS=监听的etcd节点URL，最好是指填写本集群内的所有节点，0.0.0.0表示监听全部地址
ETCD_LISTEN_CLIENT_URLS=监听的客户端URLs，最好填写MinIO集群的地址，只允许MinIO访问，但是考虑到未来扩容，这里监听全部
ETCD_NAME=etcd节点的名称，需要唯一

ETCD_INITIAL_ADVERTISE_PEER_URLS=广播节点URL
ETCD_ADVERTISE_CLIENT_URLS=广播客户端URL
ETCD_INITIAL_CLUSTER=集群内所有节点
ETCD_INITIAL_CLUSTER_TOKEN=令牌，每个节点需要相同
ETCD_INITIAL_CLUSTER_STATE=集群状态
```

所有的节点都进行上述配置。参数`ETCD_INITIAL_CLUSTER`填写的节点信息，必须与各节点上配置的`ETCD_NAME`参数以及`ETCD_INITIAL_ADVERTISE_PEER_URLS`参数进行对应。

启动etcd集群:

```
systemctl start etcd
systemctl enable etcd
```

```
[root@node3 minio]# etcdctl member list
6284eeb773af23cc: name=etcd3 peerURLs=http://192.168.251.98:2380 clientURLs=http://192.168.251.98:2379 isLeader=false
793ba0aaa9c9a810: name=etcd1 peerURLs=http://192.168.251.133:2380 clientURLs=http://192.168.251.133:2379 isLeader=false
80c8b20abbb616fc: name=etcd2 peerURLs=http://192.168.251.88:2380 clientURLs=http://192.168.251.88:2379 isLeader=true
```

使用联邦模式启动node1集群

```
[root@node1 minio]# cat run.sh
#!/bin/bash

export MINIO_ROOT_USER=admin
export MINIO_ROOT_PASSWORD=edoc2@edoc2

export
MINIO_ETCD_ENDPOINTS="http://192.168.251.133:2379,http://192.168.251.88:2379,htt
p://192.168.251.89:2379"
export MINIO_PUBLIC_IPS=192.168.251.133
export MINIO_DOMAIN=cluster.minio.com

/opt/minio/minio server --config-dir /opt/minio/config/ --console-address
":30000"\
    http://node1:9000/data{1...2}/drive{1...4}
```

192.168.251.133 这个节点，有八个drive组成单个minio集群。

MINIO_ETCD_ENDPOINTS： 配置etcd集群地址

MINIO_PUBLIC_IPS：配置当前集群节点地址，这里minio集群只有一个节点有多个节点的话逗号分割进行添加

MINIO_DOMAIN：组成联邦的多个minio集群需要保持一致

注意： `MINIO_ETCD_ENDPOINTS` 参数需与搭建的ETCD集群所有节点IP相对应； `MINIO_PUBLIC_IPS` 参数则为该集群的所有节点IP； `MINIO_DOMAIN` 参数必须进行配置，即使你并不通过域名访问存储桶，否则联邦无法生效，只有 `MINIO_DOMAIN` 参数值相同的集群，才会组成联邦。

添加bucket进行测试

```
[root@node1 minio]# mc mb node1/edoc2
Bucket created successfully `node1/edoc2`.
```

```
[root@node1 minio]# ETCDCTL_API=3 etcdctl get --from-key ''
/skydns/com/minio/cluster/edoc2/192.168.251.133
{"host":"192.168.251.133","port":9000,"ttl":30,"creationDate":"2022-07-08T07:13:31.448150349Z"}
```

创建edoc2之后，查看etcd可以看到edoc2 这个桶的记录是在 192.168.251.133 这个集群的。

```
[root@node1 minio]# mc cp Replication* node1/edoc2
...teUserPolicy.json:  2.12 KiB / 2.12 KiB ┃▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓┃ 99.23 KiB/s 0s
[root@node1 minio]#
[root@node1 minio]#
[root@node1 minio]# mc ls node1/edoc2
[2022-07-08 15:13:45 CST]   848B STANDARD ReplicationAdminPolicy.json
[2022-07-08 15:13:45 CST] 1.3KiB STANDARD ReplicationRemoteUserPolicy.json
```

对node1的edoc2桶进行上传文件

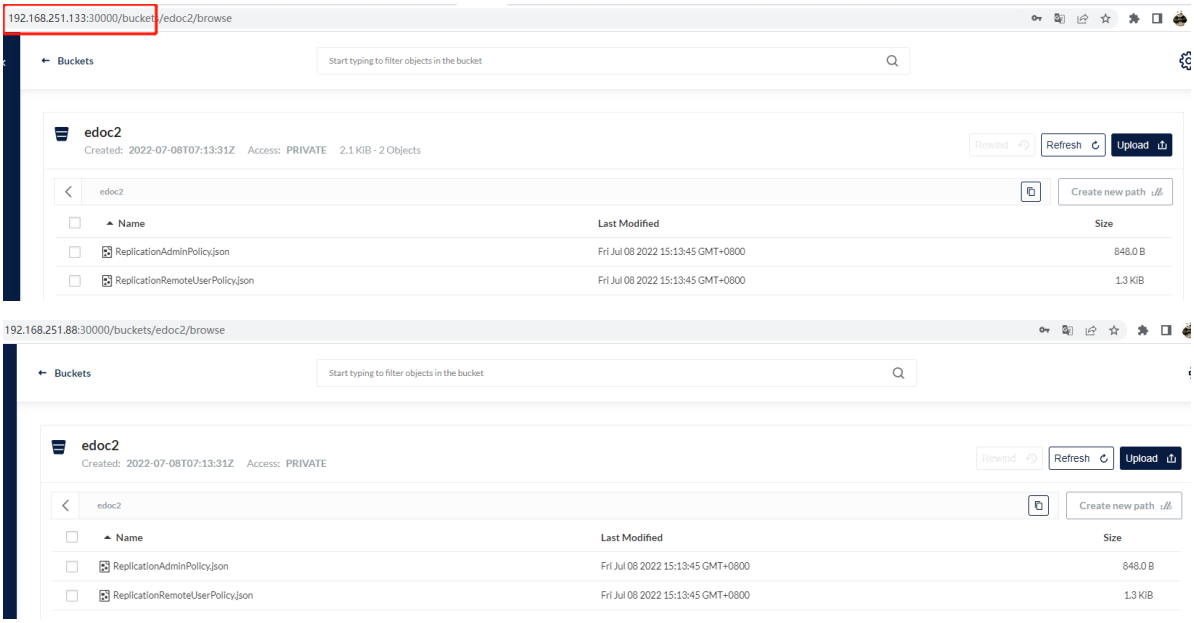
添加node2集群

新建node2集群

```
[root@node2 minio]# cat run.sh
#!/bin/bash

export MINIO_ROOT_USER=admin
export MINIO_ROOT_PASSWORD=edoc2@edoc2

export
MINIO_ETCD_ENDPOINTS="http://192.168.251.133:2379,http://192.168.251.88:2379,htt
p://192.168.251.89:2379"
export MINIO_PUBLIC_IPS=192.168.251.88
export MINIO_DOMAIN=cluster.minio.com

/opt/minio/minio server --config-dir /opt/minio/config/ --console-address
":30000"\
    http://node2:9000/data{1...2}/drive{1...4}
```

联邦集群测试

此时登录node1和node2都可以查看到edoc2的桶里的文件



在node2节点创建edoc3存储桶

```
[root@node2 minio]# mc mb node2/edoc3
Bucket created successfully `node2/edoc3`.
[root@node2 minio]# mc ls node2
[2022-07-08 14:48:24 CST]     0B edoc2/
[2022-07-08 15:08:32 CST]     0B edoc3/
```

```
[root@node2 minio]# ETCDCTL_API=3 etcdctl get --from-key ""
/skydns/com/minio/cluster/edoc2/192.168.251.133
{"host":"192.168.251.133","port":9000,"ttl":30,"creationDate":"2022-07-08T07:13:31.448150349Z"}
/skydns/com/minio/cluster/edoc3/192.168.251.88
{"host":"192.168.251.88","port":9000,"ttl":30,"creationDate":"2022-07-08T07:16:08.522853967Z"}
```

etcd中存储的 edoc3的集群信息。

此时不论访问 192.168.251.133 或这个访问192.168.251.88，都可以访问到分属于两个集群的存储桶数据。

注意：

组成联邦的minio集群，有统一的名称空间，在创建bucket的时候不能创建在每个集群创建相同的bucket名字。

```
[root@node3 minio]# mc mb node3/edoc2
mc: <ERROR> Unable to make bucket `node3/edoc2`. The requested bucket name is not available. The bucket namespace is shared b
y all users of the system. Please select a different name and try again.
[root@node3 minio]# mc mb node3/edoc4
Bucket created successfully `node3/edoc4`.
```