

Swarm Minio HowTo!

Πίνακας περιεχομένων

1. configure Minio	1
1.1. Connect to Server	1
1.2. Create policy	1
1.3. Apply policy	2
1.4. Sync local2Server	3
1.5. Sync Server2Local	3
1.6. Docker-compose env	4
1.7. Automating sync with cron	4

1. configure Minio

1.1. Connect to Server

create policy

```
mc config host add --insecure [SESSION] https://ip:9443 key secret --api s3v4
e.g.
mc config host add --insecure mysession https://83.212.119.119:9443 mykey mysecret
--api s3v4
```

1.2. Create policy

create Default policy file: test.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          ""
        ]
      },
      "Action": [
        "s3:GetBucketLocation", ①
        "s3:HeadBucket", ①
        "s3:ListBucket", ①
        "s3:ListBucketMultipartUploads" ①
      ],
      "Resource": [
        "arn:aws:s3:::test" ②
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:DeleteObject", ③
        "s3:GetObject", ③
        "s3:ListMultipartUploadParts", ③
        "s3:PutObject" ③
      ],
      "Resource": [
        "arn:aws:s3:::test/*" ④
      ]
    }
  ]
}
```

① Bucket properties [more info](#)

② Bucket source

③ Object properties [more info](#)

④ Object source

1.3. Apply policy

create policy

```
#!/bin/bash

mc mb --insecure mymsession/test ①

mc admin policy add --insecure mymisation test ./test.json ②

mc admin user add --insecure mymsession usertest
bfksdkdjhdhbfbsdhbf79fcbc7idjfdsjfsdbhfbhdbfhsd ③

mc admin policy set --insecure mymsession test user=usertest ④
```

- ① create bucket
- ② Apply policy
- ③ create user
- ④ Apply policy User2bucket

1.4. Sync local2Server

sync local2server

```
export MINIO_MIRROR_PARAMETERS="--delete"
export MINIO_SERVER_URL="ip:9443" ①
export MINIO_PROT=https
export MINIO_SERVER_DOCUMENTROOT="/home/user/syncdir" ②
export MINIO_WEB_BUCKET=test1
export MINIO_WEB_PASS="bfksdkdjhdhbfbsdhbf79fcbc7idjfdsjfsdbhfbhdbfhsd"
export AWS_CONFIG_FILE="$HOME/.aws/config"
export AWS_ACCESS_KEY_ID=$MINIO_WEB_BUCKET
export AWS_SECRET_ACCESS_KEY=$MINIO_WEB_PASS

aws --endpoint-url $MINIO_PROT://$MINIO_SERVER_URL s3 sync MINIO_SERVER_DOCUMENTROOT
s3://$MINIO_WEB_BUCKET $MINIO_MIRROR_PARAMETERS
```

- ① Minio Server ip
- ② Directory sync

1.5. Sync Server2Local

sync server2local

```
export MINIO_MIRROR_PARAMETERS="--delete"
export MINIO_SERVER_URL="ip:9443" ①
export MINIO_PROT=https
export MINIO_SERVER_DOCUMENTROOT="/var/www/html" ②
export MINIO_WEB_BUCKET=test
export MINIO_WEB_PASS="bfksdkdjhbhfsdhhbf79fcbc7idjfdsjfsdbhfbhdbfhsd"
export AWS_CONFIG_FILE="$HOME/.aws/config"
export AWS_ACCESS_KEY_ID=$MINIO_WEB_BUCKET
export AWS_SECRET_ACCESS_KEY=$MINIO_WEB_PASS

aws --endpoint-url $MINIO_PROT://$MINIO_SERVER_URL s3 sync s3://$MINIO_WEB_BUCKET
$MINIO_SERVER_DOCUMENTROOT $MINIO_MIRROR_PARAMETERS
```

① Minio Server ip

② Directory sync

1.6. Docker-compose env

sync docker-compose env

```
environment: ①
- MINIO_MIRROR_PARAMETERS= --delete
- MINIO_SERVER_URL=ip:9443
- MINIO_PROT=https
- MINIO_SERVER_DOCUMENTROOT=/var/www/html
- MINIO_WEB_BUCKET=test
- MINIO_WEB_PASS=bfksdkdjhbhfsdhhbf79fcbc7idjfdsjfsdbhfbhdbfhsd
- AWS_CONFIG_FILE=/home/user/.aws/config
```

① add env to docker-comose

1.7. Automating sync with cron

crond file

```
*/15 * * * * root /PATH/local2server >> /var/log/usercron.log 2>&1 / ①
```

① Every 15 minutes