MinIO

Sistema de almacenamiento distribuido

Instalar MinIO

MinIO es un servicio de almacenamiento en la nube compatible con Amazon S3, liberado bajo Licencia Apache v2.

MinIO permite establecer un sistema distribuido con alta disponibilidad gestionando diferentes servidores con diferentes almacenamientos como un sólo sistema de almacenamiento.

MinIO permite almacenar datos en formato archivo (como datos desestructurados fotos, vídeos, archivos de registro, copias de seguridad e imágenes de contenedor)

El tamaño máximo de un objeto es 50TB.

Instalación de MinIO en Debian

En primer lugar, instalaremos wget si no está instalado

```
apt install wget
```

Después descargaremos el archivo deb del repositorio de MinIO.

```
wget https://dl.min.io/server/minio/release/linux-amd64/archive/minio_20230930070229.0.0_amd64.deb -O minio.deb
```

Y una procedido a la descarga, lo instalaremos

```
dpkg -i minio.deb
```

Configuración de MinIO

Agregamos un grupo para el servicio de MinIO

```
groupadd -r minio-user
```

Agregamos un usuario también

```
useradd -M -r -g minio-user minio-user
```

Establecemos permisos en la carpeta o en el disco de almacenamiento (en nuestro ejemplo hemos montado en el punto de montaje /mnt/minio

```
chown minio-user:minio-user /mnt/minio
```

El paquete de instalación nos configurará el servicio de minio para poder ejecutarlo, lo comprobamos viendo el fichero del servicio.

```
cat /usr/lib/systemd/system/minio.service
```

Comprobaremos que los valores son correctos (usuario y grupo del servicio)

```
[Unit]
Description=MinIO
Documentation=https://docs.min.io
Wants=network-online.target
After=network-online.target
AssertFileIsExecutable=/usr/local/bin/minio

[Service]
Type=notify

WorkingDirectory=/usr/local

User=minio-user
Group=minio-user
ProtectProc=invisible

EnvironmentFile=-/etc/default/minio
ExecStartPre=/bin/bash -c "if [ -z \"${MINIO_VOLUMES}\" ]; then echo \"Variable MINIO_VOLUMES
not set in /etc/default/minio\"; exit 1; fi"
ExecStart=/usr/local/bin/minio server $MINIO_OPTS $MINIO_VOLUMES

# Let systemd restart this service always
Restart=always

# Specifies the maximum file descriptor number that can be opened by this process
LimitNOFILE=1048576
```

```
# Specifies the maximum number of threads this process can create
TasksMax=infinity

# Disable timeout logic and wait until process is stopped
TimeoutStopSec=infinity
SendSIGKILL=no

[Install]
WantedBy=multi-user.target

# Built for ${project.name}-${project.version} (${project.name})
```

Editamos el fichero de configuración de MinIO para ajustar los parámetros a nuestro sistema.

```
nano /etc/default/minio
```

Y configuramos nuestro MinIO. En este ejemplo hemos configurado los siguientes parámetros

MINIO_ROOT_USER=minioadmin
MINIO_ROOT_PASSWORD=miniopassword

Ambos valores están puestos a efectos de esta Wiki, recomendamos usar unos valores para el usuario y la password más seguros

MINIO_VOLUMES="/mnt/minio"

Es la carpeta o el disco en el que MinIO va a almacenar los datos.

MINIO_SERVER_URL="http://10.200.3.191:9000"

Es la url del portal de administración de MinIO.

```
# MINIO_ROOT_USER and MINIO_ROOT_PASSWORD sets the root account for the MinIO server.
# This user has unrestricted permissions to perform S3 and administrative API operations on
any resource in the deployment.
# Omit to use the default values 'minioadmin:minioadmin'.
# MinIO recommends setting non-default values as a best practice, regardless of environment

MINIO_ROOT_USER=minioadmin
MINIO_ROOT_PASSWORD=miniopassword

# MINIO_VOLUMES sets the storage volume or path to use for the MinIO server.
```

```
MINIO_VOLUMES="/mnt/minio"

# MINIO_SERVER_URL sets the hostname of the local machine for use with the MinIO Server
# MinIO assumes your network control plane can correctly resolve this hostname to the local machine

# Uncomment the following line and replace the value with the correct hostname for the local machine and port for the MinIO server (90>


MINIO_SERVER_URL="http://10.200.3.191:9000"
```

Por último habilitaremos el servicio para que arranque al inicio y lo iniciaremos

```
systemctl enable minio
systemctl start minio
```

El vídeo explicativo del proceso, lo puedes ver en nuestro canal de Youtube.

Montar S3 MinIO en un servidor

Cuando el minio está activo y funcionando, procederemos a crear un bucket como en el ejemplo que se llama s3bucket.

Ahora vamos a ver como montarlo en una ruta /mnt de nuestro servidor para usarlo como sistema de almacenamiento.

Instalar el cliente S3

Vamos a usar el cliente S3 de Amazon en nuestro servidor. Para ello nos lo descargamos

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

Ahora si no lo tenemos, instalamos las utilidades de archivos zip

```
apt install zip
```

Y descomprimimos

```
unzip awscliv2.zip
```

Ejecutamos la instalación

```
./aws/install
```

Y configuramos el cliente de S3

```
aws configure
```

Nos pedirá el access key y la clave de nuestro bucket

```
AWS Access Key ID:
AWS Secret Access Key:
Default region name [None]:
Default output format [None]:
```

Una vez configurado, ajustaremos la configuración de compatibilidad

```
aws configure set default.s3.signature_version s3v4
```

Para comprobar, ejecutaremos un ls en nuestro bucket, al que previamente le habremos subido desde la interfaz web de MinIO un archivo de prueba

```
root:#aws --endpoint-url http://s3.tecnocratica.net:9000 s3 ls
2023-10-11 18:50:49 prueba.txt
root:#
```

Instalación de fuse para S3

Instalaremos S3 con fuse para poder usar el sistema de archivos de nuestro bucket

```
apt install s3fs
```

Ahora para conectar a nuestro bucket, crearemos un fichero de texto con el Access Key y el Secret Key en el formato siguiente: accesskey:secretkey

```
echo k4COvmy9tY4AKjsi0FBk:nuW4UQoPHQdt7LmSCp1hpC4jLTOcIUjgeSkierZq > pass.txt
```

Cambiaremos permisos por seguridad

```
chmod 600 pass.txt
```

Montar el bucket de S3

Creamos una carpeta por ejemplo /mnt/s3bucket

```
mkdir /mnt/s3bucket
```

Montamos el bucket

El comando para montar es s3fs

Usaremos los siguientes modificadores

/mnt/s3bucket la carpeta donde montaremos el bucket

-o

Bajo estas directivas añadiremos las opciones.

-o bucket=s3bucket

-o passwd_file=pass.txt

-o use_path_request_style

> Esta opción es muy importante para usar el estilo de ficheros que usamos normalmente en Linux

-o host=http://s3.tecnocratica.net:9000

El nombre del host al que nos conectaremos

El comando quedaría así

```
s3fs /mnt/s3bucket -o bucket=s3bucket -o passwd_file=pass.txt -o use_path_request_style -o host=http://s3.tecnocratica.net:9000
```

Ahora podemos comprobar con un df-h

```
root@eduardo:# df -h
Filesystem                Size  Used Avail Use% Mounted on
udev                      3.9G     0  3.9G   0% /dev
tmpfs                     794M  736K  793M   1% /run
/dev/mapper/pbs-root       17G  2.4G   14G  16% /
tmpfs                     3.9G     0  3.9G   0% /dev/shm
tmpfs                     5.0M     0  5.0M   0% /run/lock
copias                     39G   33M   39G   1% /mnt/datastore/copias
tmpfs                     794M     0  794M   0% /run/user/0
s3fs                       16E     0   16E   0% /mnt/s3bucket
```

Comprobaciones

Si no aparece el punto de montaje, podemos hacer debug añadiendo opciones al comando

```
s3fs /mnt/s3bucket -o bucket=s3bucket -o passwd_file=pass.txt -o use_path_request_style -o host=http://s3.tecnocratica.net:9000 -o dbglevel=info -f -o curldbg
```

Esto nos mostrará todos los mensajes de la conexión a fin de depurar

El vídeo explicativo del proceso, lo puedes ver en nuestro canal de Youtube.

Comandos s3

Vamos a describir una serie de comandos útiles para nuestra instancia de S3 que hemos creado con MinIO

Listar ficheros en el bucket

```
aws --endpoint-url http://A.B.C.D:9000 s3 ls
```

Borrar un objeto

```
aws --endpoint-url http://A.B.C.D:9000 s3 rm s3://bucket/carpeta/prueba.txt
```

Mover un objeto

Mover a local

```
aws --endpoint-url http://A.B.C.D:9000 s3 mv s3://bucket/carpeta/prueba.txt ./prueba.txt
```

Mover a otro bucket

```
aws --endpoint-url http://A.B.C.D:9000 s3 mv s3://bucket/carpeta/prueba.txt
s3://bucket2/carpeta2/
```

Copiar un objeto

Copiar a local

```
aws --endpoint-url http://A.B.C.D:9000 s3 cp s3://bucket/carpeta/prueba.txt ./prueba.txt
```

Copiar a otro bucket

```
aws --endpoint-url http://A.B.C.D:9000 s3 cp s3://bucket/carpeta/prueba.txt
s3://bucket2/carpeta2/
```

Copiar todos los objetos

```
aws --endpoint-url http://A.B.C.D:9000 s3 mv s3://bucket/carpeta/ s3://bucket2/carpeta2/
```

Sincronizar objetos

```
aws --endpoint-url http://A.B.C.D:9000 s3 sync s3://bucket/carpeta/prueba.txt
s3://bucket2/carpeta2/
```

Modificadores de los comandos de objetos

Podemos incluir o excluir patrones de objetos mediante el modificador include o exclude. EL siguiente comando excluirá todos los txt, pero incluirá los archivos fotos2023 y lo que sea, pero excluirá los fotos202308 y un carácter.

```
-exclude "*.txt" --include "fotos2023*.txt" --exclude "fotos202308?.txt"
```