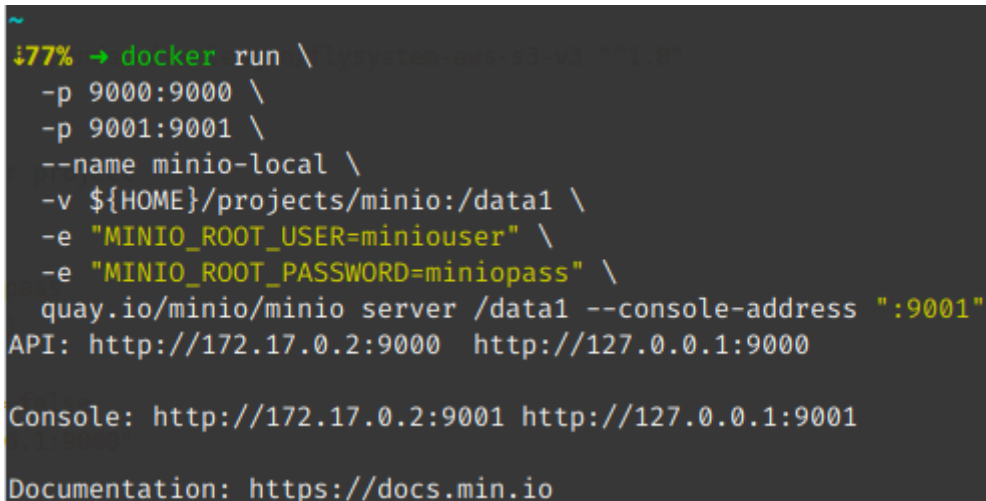


# Setup MinIO as AWS S3 alternative in development with docker

## Installation

1. Make sure your local machine already have Docker installed.
2. Run MinIO:

```
docker run \
  -p 9000:9000 \
  -p 9001:9001 \
  --name minio-local \
  -v ${HOME}/projects/minio:/data1 \
  -e "MINIO_ROOT_USER=miniouser" \
  -e "MINIO_ROOT_PASSWORD=miniopass" \
  quay.io/minio/minio server /data1 --console-address ":9001"
```

A terminal window with a dark background. The prompt is '~'. The command 'docker run \ -p 9000:9000 \ -p 9001:9001 \ --name minio-local \ -v \${HOME}/projects/minio:/data1 \ -e "MINIO\_ROOT\_USER=miniouser" \ -e "MINIO\_ROOT\_PASSWORD=miniopass" \ quay.io/minio/minio server /data1 --console-address ":9001"' is entered. The output shows 'API: http://172.17.0.2:9000 http://127.0.0.1:9000' and 'Console: http://172.17.0.2:9001 http://127.0.0.1:9001'. At the bottom, it says 'Documentation: https://docs.min.io'.

```
~
❯ docker run \
  -p 9000:9000 \
  -p 9001:9001 \
  --name minio-local \
  -v ${HOME}/projects/minio:/data1 \
  -e "MINIO_ROOT_USER=miniouser" \
  -e "MINIO_ROOT_PASSWORD=miniopass" \
  quay.io/minio/minio server /data1 --console-address ":9001"
API: http://172.17.0.2:9000 http://127.0.0.1:9000
Console: http://172.17.0.2:9001 http://127.0.0.1:9001
Documentation: https://docs.min.io
```

3. On your Laravel project, install the dependency

```
composer require --with-all-dependencies league/flysystem-aws-s3-v3 "^1.0"
```

#### 4. Change the `.env` on your project

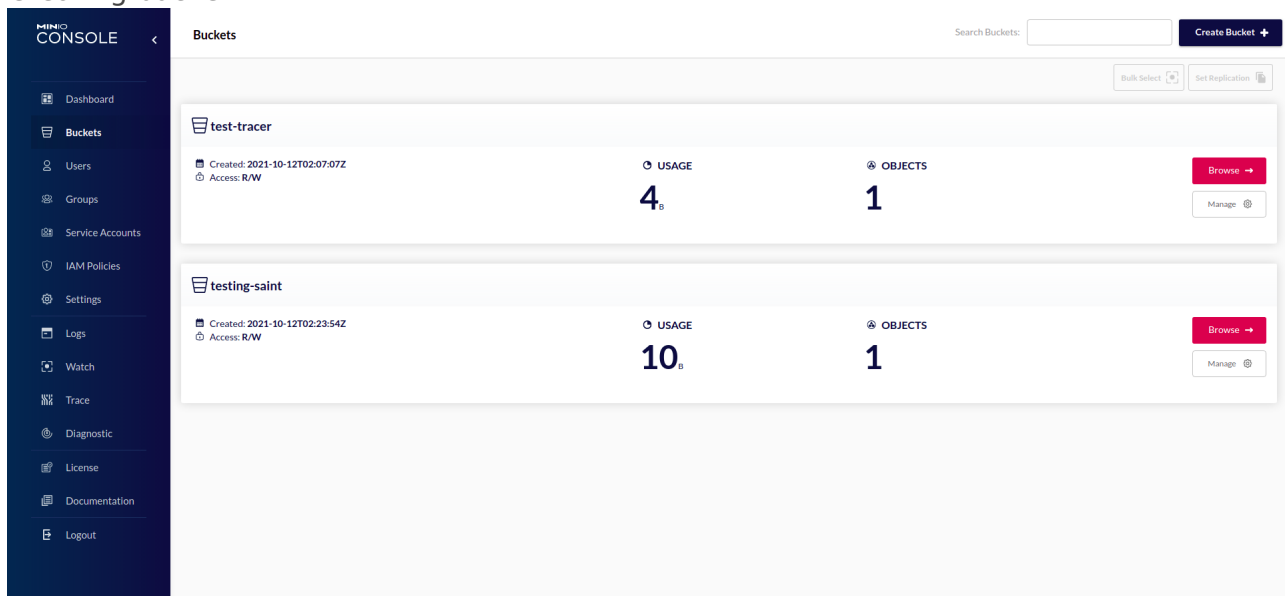
```
AWS_ACCESS_KEY_ID=miniouser
AWS_SECRET_ACCESS_KEY=miniopass
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=testing-saint
AWS_USE_PATH_STYLE_ENDPOINT=false
AWS_ENDPOINT="http://127.0.0.1:9000"
```

#### 5. Running after rebooting your PC / Laptop / Mac (shall I mention rpi too?)

```
docker start minio-local
```

#### 6. Accessing the web interface Open your web browser, access <http://localhost:9001>

#### 7. Creating bucket



The screenshot displays the Minio Console interface, specifically the 'Buckets' section. On the left is a dark sidebar with navigation links: Dashboard, Buckets, Users, Groups, Service Accounts, IAM Policies, Settings, Logs, Watch, Trace, Diagnostic, License, Documentation, and Logout. The main content area shows a list of buckets. At the top right of the main area, there is a 'Search Buckets' input field and a 'Create Bucket' button. Below this, there are two bucket entries:

Bucket Name	Created	Access	USAGE	OBJECTS	Actions
test-tracer	2021-10-12T02:07:07Z	R/W	4 <sub>B</sub>	1	<a href="#">Browse</a> <a href="#">Manage</a>
testing-saint	2021-10-12T02:23:54Z	R/W	10 <sub>B</sub>	1	<a href="#">Browse</a> <a href="#">Manage</a>

Click 'Create Bucket', and a modal window will be shown like this:

×

## Create Bucket

Bucket Name

## Features

Some these features are disabled as server is running in non-erasure coded mode.

Versioning

Off

Allows to keep multiple versions of the same object under the same key.

Object Locking

Off

Required to support retention and legal hold. Can only be enabled at bucket creation.

Quota

Off

Limit the amount of data in the bucket.

Clear

Save

Set the bucket name. The bucket name will be used as `AWS_BUCKET` in your app.

## Further Reading

[MinIO Documentation](#)

Revision #2

Created 12 October 2021 02:33:22 by Adli Ikhwanur Ifkar

Updated 12 October 2021 02:45:04 by Adli Ikhwanur Ifkar