

# MinIO Server - Docker Compose & NGINX reverse proxy

My basic example for a MinIO server.

“ ?? I'm leaving this up for historical reasons, but might be out of date as I am no longer relying on MinIO ??

## Docker Compose

```
version: '3'

services:
  minio:
    image: minio/minio:latest
    volumes:
      - /data/my-fancy-local-storage:/data
    ports:
      # I use an NGINX reverse proxy, so these ports are only exposed
      # to the loopback adapter on the host server.
      - "127.0.0.1:9000:9000"
      - "127.0.0.1:9001:9001"
    environment:
      MINIO_ROOT_USER: <REDACTED>
      MINIO_ROOT_PASSWORD: <REDACTED>
      MINIO_SERVER_URL: https://s3.example.com
      MINIO_BROWSER_REDIRECT_URL: https://s3-manage.example.com
      MINIO_DOMAIN: s3.example.com
    command: server --console-address :9001 /data
    healthcheck:
```

```
test: ["CMD", "curl", "-f", "http://minio:9000/minio/health/live"]
interval: 1m30s
timeout: 20s
retries: 3
start_period: 3m
restart: unless-stopped
```

# NGINX - Management console

Slightly tweaked as I use a lot of alias configs. This variant is consolidated into a singular configuration.

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name s3-manage.example.com;
    root /var/www/html;

    ssl_certificate /srv/ssl/minio/fullchain.pem;
    ssl_certificate_key /srv/ssl/minio/key.pem;

    server_tokens off;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-
POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;
    # You'll have to generate DH Parameters yourself
    ssl_dhparam /etc/nginx/dhparams.pem;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1d;
    ssl_session_tickets off;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    # To allow special characters in headers
    ignore_invalid_headers off;
```

```
# Allow any size file to be uploaded.
# Set to a value such as 1000m; to restrict file size to a specific value
client_max_body_size 0;
# To disable buffering
proxy_buffering off;

add_header Strict-Transport-Security "max-age=63072000; preload";

index index.html index.htm;

charset utf-8;

location / {
    satisfy any;
    # For whitelisting IPs
    allow 127.0.0.1;
    deny all;

    # auth_basic "Restricted Access";
    # auth_basic_user_file /etc/nginx/.htpasswd-minio;

    proxy_pass http://127.0.0.1:9001;
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    proxy_connect_timeout 300;
    # Default is HTTP/1, keepalive is only enabled in HTTP/1.1
    proxy_http_version 1.1;
    proxy_set_header Connection "";
    chunked_transfer_encoding off;
}

location /ws {
    satisfy any;
    # For whitelisting IPs
    allow 127.0.0.1;
    deny all;
```

```

proxy_pass http://127.0.0.1:9001;

proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "Upgrade";
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
chunked_transfer_encoding off;
}

location = /favicon.ico { access_log off; log_not_found off; }
location = /robots.txt { access_log off; log_not_found off; }

access_log /var/log/nginx/s3-manage.example.com-access.log combined;
error_log /var/log/nginx/s3-manage.example.com-error.log error;

location ~ /\.ht {
    deny all;
}
}

```

## NGINX - For uploads/downloads

```

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    # The leading period here is important to support virtualhost style vs. path style
    # E.g. `bucket-name.s3.example.com` instead of `s3.example.com/bucket-name`
    server_name .s3.example.com;
    root /var/www/html;

    ssl_certificate /srv/ssl/minio/fullchain.pem;
    ssl_certificate_key /srv/ssl/minio/key.pem;

    server_tokens off;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-
POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;

```

```
ssl_prefer_server_ciphers off;
# You'll have to generate DH Parameters yourself
ssl_dhparam /etc/nginx/dhparams.pem;
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 1d;
ssl_session_tickets off;

add_header X-Frame-Options "SAMEORIGIN";
add_header X-XSS-Protection "1; mode=block";
add_header X-Content-Type-Options "nosniff";

# To allow special characters in headers
ignore_invalid_headers off;
# Allow any size file to be uploaded.
# Set to a value such as 1000m; to restrict file size to a specific value
client_max_body_size 0;
# To disable buffering
proxy_buffering off;

add_header Strict-Transport-Security "max-age=63072000; preload";

index index.html index.htm;

charset utf-8;

location / {
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    proxy_connect_timeout 300;
    # Default is HTTP/1, keepalive is only enabled in HTTP/1.1
    proxy_http_version 1.1;
    proxy_set_header Connection "";
    chunked_transfer_encoding off;

    proxy_pass http://127.0.0.1:9000;
}

location = /favicon.ico { access_log off; log_not_found off; }
```

```
location = /robots.txt { access_log off; log_not_found off; }
```

```
access_log /var/log/nginx/s3.example.com-access.log combined;
```

```
error_log /var/log/nginx/s3.example.com-error.log error;
```

```
location ~ /\.ht {
```

```
    deny all;
```

```
}
```

```
}
```

---

Revision #2

Created 21 January 2022 21:19:35 by Alex Thomassen

Updated 24 September 2024 10:35:39 by Alex Thomassen