

Notesnook sync server (local storage)

Overview

This guide will help you set up a self-hosted instance of [Notesnook](#) using Docker Compose.

Components Breakdown

Core Components (Required)

1. Notesnook Server (notes.domain.com:5264)

- **Purpose:** Main synchronization server
- **Accessibility:** Must be publicly accessible
- **Why:** Handles all note synchronization between clients
- **Dependencies:** Requires MongoDB and Identity Server

2. Identity Server (auth.domain.com:8264)

- **Purpose:** Authentication and user management
- **Accessibility:** Must be publicly accessible
- **Why:**
 - Handles user registration and login
 - Issues JWT tokens for authentication
 - Manages user sessions
 - Separate domain for security isolation
- **Dependencies:** Requires MongoDB

3. SSE Server (events.domain.com:7264)

- **Purpose:** Real-time updates
- **Accessibility:** Must be publicly accessible
- **Why:**
 - Provides Server-Sent Events for real-time sync
 - Notifies clients when changes occur

- Essential for multi-device synchronization
- Dependencies: Requires Notesnook Server

4. MongoDB (Internal)

- **Purpose:** Database storage
- **Accessibility:** Internal only
- **Why:** Stores user data, notes, and authentication info
- **Configuration:** Runs as a replica set for reliability

Optional Components

5. Monograph Server (mono.domain.com:6264)

- **Purpose:** Public note sharing
- **Accessibility:** Optional, can be disabled
- **Why:**
 - Allows users to share notes publicly
 - Can be omitted if you don't need public sharing
- **Dependencies:** Requires Notesnook Server

6. MinIO (Internal, ports 9009/9090)

- **Purpose:** S3-compatible storage (only locally in this setup)
- **Why:**
 - Stores file attachments
 - Used by Notesnook Server internally
 - No need for public access
- **Note:** Console (9090) can be restricted to localhost

Minimum Required Setup

For a basic working installation:

- Notesnook Server (public)
- Identity Server (public)
- SSE Server (public)
- MongoDB (internal)
- MinIO (internal)

You can omit:

- Monograph Server (if public sharing not needed)
- MinIO Console access (manage via CLI)

Security Considerations

1. Only these need public domains:

- notes.domain.com
- auth.domain.com
- events.domain.com
- mono.domain.com (optional)

2. Keep internal:

- MongoDB (27017)
- MinIO API (9009)
- MinIO Console (9090)

3. JWT tokens ensure secure communication between components

Prerequisites

- **Linux server** with Docker and Docker Compose installed.
- **Domain name** with the ability to create subdomains.
- Basic understanding of terminal commands.
- Ports **5264**, **6264**, **7264**, **8264**, **9090** and **9009** available. Or you can change them but take good note of your changes.

1. Directory Structure Setup

Create the required directories:

```
# Create data directories
mkdir -p /srv/Files/Notesnook/db
mkdir -p /srv/Files/Notesnook/s3
mkdir -p /srv/Files/Notesnook/setup
```

2. Configuration Files

2.1. Environment File

Create the `.env` file:

```
cd /srv/Files/Notesnook/setup
nano .env
```

Add the following content (modify the values accordingly):

```
# Instance Configuration
INSTANCE_NAME=My Notesnook
DISABLE_SIGNUPS=false
NOTESNOOK_API_SECRET=your_secure_api_secret_here

# SMTP Configuration
SMTP_USERNAME=your_email@domain.com
SMTP_PASSWORD=your_smtp_password
SMTP_HOST=smtp.your-server.com
SMTP_PORT=587

# Public URLs (replace domain.com with your domain)
AUTH_SERVER_PUBLIC_URL=https://auth.domain.com/
NOTESNOOK_APP_PUBLIC_URL=https://notes.domain.com/
MONOGRAPH_PUBLIC_URL=https://mono.domain.com/
ATTACHMENTS_SERVER_PUBLIC_URL=https://files.domain.com/

# MinIO Configuration
MINIO_ROOT_USER=admin
MINIO_ROOT_PASSWORD=your_secure_password_here
```

2.2. Docker Compose File

Create the `docker-compose.yml` file:

```
nano docker-compose.yml
```

Paste the following content:

```
x-server-discovery: &server-discovery
  NOTESNOOK_SERVER_PORT: 5264
  NOTESNOOK_SERVER_HOST: notesnook-server
  IDENTITY_SERVER_PORT: 8264
  IDENTITY_SERVER_HOST: identity-server
```

```
SSE_SERVER_PORT: 7264
SSE_SERVER_HOST: sse-server
SELF_HOSTED: 1
IDENTITY_SERVER_URL: ${AUTH_SERVER_PUBLIC_URL}
NOTESNOOK_APP_HOST: ${NOTESNOOK_APP_PUBLIC_URL}
```

```
x-env-files: &env-files
```

```
- .env
```

```
services:
```

```
  validate:
```

```
    image: vandot/alpine-bash
```

```
    entrypoint: /bin/bash
```

```
    env_file: *env-files
```

```
    command:
```

```
      - -c
```

```
      - |
```

```
        required_vars=(
```

```
          "INSTANCE_NAME"
```

```
          "NOTESNOOK_API_SECRET"
```

```
          "DISABLE_SIGNUPS"
```

```
          "SMTP_USERNAME"
```

```
          "SMTP_PASSWORD"
```

```
          "SMTP_HOST"
```

```
          "SMTP_PORT"
```

```
          "AUTH_SERVER_PUBLIC_URL"
```

```
          "NOTESNOOK_APP_PUBLIC_URL"
```

```
          "MONOGRAPH_PUBLIC_URL"
```

```
          "ATTACHMENTS_SERVER_PUBLIC_URL"
```

```
        )
```

```
        for var in "${required_vars[@]}"; do
```

```
          if [ -z "${!var}" ]; then
```

```
            echo "Error: Required environment variable $$var is not set."
```

```
            exit 1
```

```
          fi
```

```
        done
```

```
        echo "All required environment variables are set."
```

```
      restart: "no"
```

```
notesnook-db:
```

```
image: mongo:7.0.12
hostname: notesnook-db
volumes:
  - /srv/Files/Notesnook/db:/data/db
  - /srv/Files/Notesnook/db:/data/configdb
networks:
  - notesnook
command: --replSet rs0 --bind_ip_all
depends_on:
  validate:
    condition: service_completed_successfully
healthcheck:
  test: echo 'db.runCommand("ping").ok' | mongosh mongodb://localhost:27017 --quiet
  interval: 40s
  timeout: 30s
  retries: 3
  start_period: 60s
```

initiate-rs0:

```
image: mongo:7.0.12
networks:
  - notesnook
depends_on:
  - notesnook-db
entrypoint: /bin/sh
command:
  - -c
  - |
    mongosh mongodb://notesnook-db:27017 <<EOF
    rs.initiate();
    rs.status();
    EOF
```

notesnook-s3:

```
image: minio/minio:RELEASE.2024-07-29T22-14-52Z
ports:
  - 9009:9000
  - 9090:9090
networks:
  - notesnook
```

volumes:

- /srv/Files/Notesnook/s3:/data/s3

environment:

MINIO_BROWSER: "on"

depends_on:

validate:

condition: service_completed_successfully

env_file: *env-files

command: server /data/s3 --console-address :9090

healthcheck:

test: timeout 5s bash -c ':> /dev/tcp/127.0.0.1/9000' || exit 1

interval: 40s

timeout: 30s

retries: 3

start_period: 60s

setup-s3:

image: minio/mc:RELEASE.2024-07-26T13-08-44Z

depends_on:

- notesnook-s3

networks:

- notesnook

entrypoint: /bin/bash

env_file: *env-files

command:

- -c

- |

until mc alias set minio http://notesnook-s3:9000/ \${MINIO_ROOT_USER:-minioadmin}

\${MINIO_ROOT_PASSWORD:-minioadmin}; do

sleep 1;

done;

mc mb minio/attachments -p

identity-server:

image: streetwriters/identity:latest

ports:

- 8264:8264

networks:

- notesnook

env_file: *env-files

```
depends_on:
  - notesnook-db
healthcheck:
  test: wget --tries=1 -nv -q http://localhost:8264/health -O- || exit 1
  interval: 40s
  timeout: 30s
  retries: 3
  start_period: 60s
environment:
  <<: *server-discovery
  MONGODB_CONNECTION_STRING: mongodb://notesnook-db:27017/identity?replSet=rs0
  MONGODB_DATABASE_NAME: identity
```

```
notesnook-server:
  image: streetwriters/notesnook-sync:latest
  ports:
    - 5264:5264
  networks:
    - notesnook
  env_file: *env-files
  depends_on:
    - notesnook-s3
    - setup-s3
    - identity-server
  healthcheck:
    test: wget --tries=1 -nv -q http://localhost:5264/health -O- || exit 1
    interval: 40s
    timeout: 30s
    retries: 3
    start_period: 60s
  environment:
    <<: *server-discovery
    MONGODB_CONNECTION_STRING: mongodb://notesnook-db:27017/?replSet=rs0
    MONGODB_DATABASE_NAME: notesnook
    S3_INTERNAL_SERVICE_URL: "http://notesnook-s3:9000/"
    S3_INTERNAL_BUCKET_NAME: "attachments"
    S3_ACCESS_KEY_ID: "${MINIO_ROOT_USER:-minioadmin}"
    S3_ACCESS_KEY: "${MINIO_ROOT_PASSWORD:-minioadmin}"
    S3_SERVICE_URL: "${ATTACHMENTS_SERVER_PUBLIC_URL}"
    S3_REGION: "us-east-1"
```


S3_BUCKET_NAME: "attachments"

sse-server:

image: streetwriters/sse:latest

ports:

- 7264:7264

env_file: *env-files

depends_on:

- identity-server
- notesnook-server

networks:

- notesnook

healthcheck:

test: wget --tries=1 -nv -q http://localhost:7264/health -O- || exit 1

interval: 40s

timeout: 30s

retries: 3

start_period: 60s

environment:

<<: *server-discovery

monograph-server:

image: streetwriters/monograph:latest

ports:

- 6264:3000

env_file: *env-files

depends_on:

- notesnook-server

networks:

- notesnook

healthcheck:

test: wget --tries=1 -nv -q http://localhost:3000/api/health -O- || exit 1

interval: 40s

timeout: 30s

retries: 3

start_period: 60s

environment:

<<: *server-discovery

API_HOST: http://notesnook-server:5264/

PUBLIC_URL: \${MONOGRAPH_PUBLIC_URL}

```
networks:
  notesnook:
```

3. Docker Images Preparation

Pull all required images to avoid timeout issues:

```
cd /srv/Files/Notesnook/setup
docker pull mongo:7.0.12
docker pull minio/minio:RELEASE.2024-07-29T22-14-52Z
docker pull streetwriters/identity:latest
docker pull streetwriters/notesnook-sync:latest
docker pull streetwriters/sse:latest
docker pull streetwriters/monograph:latest
docker pull vandot/alpine-bash
```

4. Deployment

Start the services:

```
cd /srv/Files/Notesnook/setup
docker compose up -d
```

5. Service Verification

5.1. Check Container Status

```
docker compose ps
```

Expected status:

- **Running containers:**
 - notesnook-db

- `notesnook-s3`
- `identity-server`
- `notesnook-server`
- `sse-server`
- `monograph-server`
- **Completed containers** (should show `Exit 0`):
 - `validate`
 - `initiate-rs0`
 - `setup-s3`

5.2. Check Logs

```
docker compose logs
```

5.3. Test MinIO Access

Visit: `http://your-server:9009/`

6. Reverse Proxy Configuration with Nginx and SSL

Enable WebSockets Support for: `notes.domain.com` (port 5264) - For real-time synchronization
`events.domain.com` (port 7264) - For real-time notifications

Enable Cache Assets for: `mono.domain.com` (port 6264) - For optimizing public notes loading

Step 1: Install Certbot

```
sudo apt-get update
sudo apt-get install certbot python3-certbot-nginx
```

Step 2: Obtain SSL Certificates

```
sudo certbot --nginx -d auth.domain.com -d notes.domain.com -d events.domain.com -d
mono.domain.com
```

Step 3: Modify Nginx Configuration

Use the following example configurations for each subdomain:

```
# Auth Server - Basic (no cache/websocket needed)
server {
    listen 80;
    server_name auth.domain.com;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name auth.domain.com;

    ssl_certificate /etc/letsencrypt/live/auth.domain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/auth.domain.com/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_pass http://localhost:8264/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

# Notes Server - With WebSocket
server {
    listen 80;
    server_name notes.domain.com;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name notes.domain.com;

    ssl_certificate /etc/letsencrypt/live/notes.domain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/notes.domain.com/privkey.pem;
```

```

include /etc/letsencrypt/options-ssl-nginx.conf;
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

location / {
    proxy_pass http://localhost:5264/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_read_timeout 3600;
    proxy_send_timeout 3600;
}
}

# Events Server - With WebSocket
server {
    listen 80;
    server_name events.domain.com;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name events.domain.com;

    ssl_certificate /etc/letsencrypt/live/events.domain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/events.domain.com/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_pass http://localhost:7264/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_read_timeout 3600;
        proxy_send_timeout 3600;
    }
}

```

```
# Monograph Server - With Cache
server {
    listen 80;
    server_name mono.domain.com;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name mono.domain.com;

    ssl_certificate /etc/letsencrypt/live/mono.domain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/mono.domain.com/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_pass http://localhost:6264/;
        proxy_cache_use_stale error timeout http_500 http_502 http_503 http_504;
        proxy_cache_valid 200 60m;
        add_header X-Cache-Status $upstream_cache_status;
        expires 1h;
        add_header Cache-Control "public, no-transform";
    }
}
```

7. Useful Commands

Service Management

```
# View real-time logs
docker compose logs -f

# View logs for specific service
docker compose logs [service-name]
```

```
# Restart specific service
docker compose restart [service-name]

# Stop all services
docker compose down

# Update services
docker compose pull
docker compose up -d
```

8. Maintenance

8.1. Backup

Regularly backup these directories:

- `/srv/Files/Notesnook/db/` (MongoDB data)
- `/srv/Files/Notesnook/s3/` (MinIO data)
- `/srv/Files/Notesnook/setup/.env` (Configuration)

8.2. Updates

To update all services:

```
cd /srv/Files/Notesnook/setup
docker compose down
docker compose pull
docker compose up -d
```

9. Troubleshooting

Common Issues:

Service won't start

- Check logs: `docker compose logs [service-name]`

- Verify port availability.
- Check directory permissions.
- Verify environment variables.

Database Connection Issues

- Ensure MongoDB replica set is initialized.
- Check MongoDB logs: `docker compose logs notesnook-db`.

Storage Issues

- Verify MinIO credentials.
- Check MinIO logs: `docker compose logs notesnook-s3`.

Email Not Working

- Verify SMTP settings in `.env`.
 - Check identity-server logs.
-

Security Notes

- Change default passwords in `.env`.
 - Use strong passwords for MinIO and API secret.
 - Keep your `.env` file secure.
 - Regularly update all services.
 - Enable HTTPS on your reverse proxy.
 - Consider implementing `fail2ban`.
 - Regularly monitor logs for suspicious activity.
-

Support

If you encounter issues:

- Check the logs.
 - Visit the [Notesnook GitHub repository](#).
 - Join the [Notesnook Discord](#) for support.
-

Revision #14

Created 20 January 2025 00:03:15 by Admin

Updated 22 January 2025 09:25:51 by Admin