

Outline (Wiki, Confluence Alternative)

outline docker-compose.yml

<https://docs.getoutline.com/s/hosting/doc/docker-7pfeLP5a8t>

```
version: "3.2"
services:

  outline:
    restart: unless-stopped
    image: outlinewiki/outline:1.4.0
    env_file: stack.env
    ports:
      - "3000:3000"
    volumes:
      - storage:/var/lib/outline/data
    depends_on:
      - postgres
      - redis

  redis:
    restart: unless-stopped
    image: redis
    env_file: stack.env
    ports:
      - "6379:6379"
    volumes:
      - ./redis.conf:/redis.conf
    command: ["redis-server", "/redis.conf"]
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 10s
```

timeout: 30s

retries: 3

postgres:

restart: unless-stopped

image: postgres:18

env_file: stack.env

ports:

- "5432:5432"

volumes:

- db:/var/lib/postgresql

healthcheck:

test: ["CMD", "pg_isready", "-d", "outline", "-U", "user"]

interval: 30s

timeout: 20s

retries: 3

environment:

POSTGRES_USER: 'user'

POSTGRES_PASSWORD: 'pass'

POSTGRES_DB: 'outline'

https-portal:

image: steveltn/https-portal

env_file: stack.env

ports:

- '80:80'

- '443:443'

links:

- outline

restart: always

volumes:

- https-portal-data:/var/lib/https-portal

healthcheck:

test: ["CMD", "service", "nginx", "status"]

interval: 30s

timeout: 20s

retries: 3

environment:

DOMAINS: 'docs.mycompany.com -> http://outline:3000'

STAGE: 'production'

```
# WEBSOCKET: 'true'
# CLIENT_MAX_BODY_SIZE: '0'

volumes:
  #https-portal-data:
  storage:
  db:
```

stack.env (env-vars in portainer immer mit stack.env einbinden (siehe oben))

.env mit Authentik als OIDC

```
APP_NAME='MEIN WIKI'
NODE_ENV=production
SECRET_KEY=040.....65dba72 (openssl rand -hex 32 oder openssl rand 24 -base64)
UTILS_SECRET=7367ab.....e5e9ca (openssl rand -hex 32 oder openssl rand 24 -base64)
DATABASE_URL=postgres://pladmin:gi5ll.....tch@postgres:5432/outline #< PW AUS docker-
compose.yml ÜBERNEHMEN!
PGSSLMODE=disable
POSTGRES_USER='pladmin'
POSTGRES_PASSWORD='gi.....tch'
POSTGRES_DB='outline'
REDIS_URL=redis://redis:6379
URL=https://wiki.MEINEDOMAIN.DE
PORT=3000
COLLABORATION_URL=
FILE_STORAGE=local
FILE_STORAGE_LOCAL_ROOT_DIR=/var/lib/outline/data
FILE_STORAGE_UPLOAD_MAX_SIZE=262144000
FORCE_HTTPS=true
ENABLE_UPDATES=true
WEB_CONCURRENCY=1
DEBUG=http
LOG_LEVEL=info
OIDC_CLIENT_ID=v7xa.....8L6T8
OIDC_CLIENT_SECRET=uSVUg.....pVFvtuxYSJ7QPBN44S2LnLjMMxQxcZ8jjEj
OIDC_AUTH_URI=https://auth.MEINEDOMAIN.DE/application/o/authorize/
OIDC_TOKEN_URI=https://auth.MEINEDOMAIN.DE/application/o/token/
OIDC_USERINFO_URI=https://auth.MEINEDOMAIN.DE/application/o/userinfo/
```

```
OIDC_LOGOUT_URI=https://auth.MEINEDOMAIN.DE/application/o/wiki/end-session/  
OIDC_USERNAME_CLAIM=preferred_username  
OIDC_DISPLAY_NAME=authentik  
OIDC_SCOPES=openid profile email  
DEFAULT_LANGUAGE=de_DE  
SMTP_HOST=smtp.strato.de  
SMTP_PORT=465  
SMTP_USERNAME=info@MEINEDOMAIN.DE  
SMTP_PASSWORD=MEIN.....MAILPW  
SMTP_FROM_EMAIL=info@MEINEDOMAIN.DE  
SMTP_SECURE=true
```

Gruppen zwischen Outline und Authentik synchronisieren

<https://github.com/burritosoftware/Outline-Authentik-Connector>

aktuell ist das Docker-Image nur für amd64 und nicht für ARM, man kann aber auch einfach n8n für die Automation anlegen, wenn man die webhooks von outline übernimmt

siehe github repo für Anleitung

outline-authentik-connector docker-compose.yml

```
name: outline-authentik-connector  
services:  
  outline-authentik-connector:  
    image: burritosoftware/outline-authentik-connector:1.1  
    build:  
      context: .  
      dockerfile: Dockerfile  
    ports:  
      - 8430:80  
    env_file: "stack.env"
```

.env

```
AUTHENTIK_URL=https://auth.MEINEDOMAIN.DE
AUTHENTIK_TOKEN=8lnL6Ke7u.....t7vmVYvEPJoe
OUTLINE_URL=https://wiki.MEINEDOMAIN.DE
OUTLINE_TOKEN=o1_api_UWBWl.....cKtMLIopMebfTks7
OUTLINE_WEBHOOK_SECRET=o1_whs_kMP.....1y30RBL6E2K
DEBUG=False
# Seit v1.1 werden Gruppen mit dieser var automatisch hinzugefügt, wenn sie in Outline fehlen:
AUTO_CREATE_GROUPS=True
```

Anhang

sample.env für alle outline settings <https://github.com/outline/outline/blob/main/.env.sample>

```
NODE_ENV=production

# This URL should point to the fully qualified, publicly accessible, URL. If using a
# proxy this will be the proxy's URL.
URL=

# The port to expose the Outline server on, this should match what is configured
# in your docker-compose.yml
PORT=3000

# See [documentation](docs/SERVICES.md) on running a separate collaboration
# server, for normal operation this does not need to be set.
COLLABORATION_URL=

# If using a Cloudfront/Cloudflare distribution or similar it can be set below.
# This will cause paths to javascript, stylesheets, and images to be updated to
# the hostname defined in CDN_URL. In your CDN configuration the origin server
# should be set to the same as URL.
CDN_URL=

# How many processes should be spawned. As a reasonable rule divide your servers
# available memory by 512 for a rough estimate
WEB_CONCURRENCY=1

# Generate a hex-encoded 32-byte random key. Use `openssl rand -hex 32` in your
# terminal to generate a random value.
```

```
SECRET_KEY=generate_a_new_key
```

```
# Generate a unique random key. The format is not important but you could still use
```

```
# `openssl rand -hex 32` in your terminal to generate a random value.
```

```
UTILS_SECRET=generate_a_new_key
```

```
# The default interface language. See translate.getoutline.com for a list of
```

```
# available language codes and their rough percentage translated.
```

```
DEFAULT_LANGUAGE=en_US
```

```
# _____
```

```
# _____ DATABASE _____
```

```
# _____
```

```
# The database URL for your production database, including username, password, and database name.
```

```
DATABASE_URL=postgres://user:pass@postgres:5432/outline
```

```
# The in-memory database pool per-process settings. Ensure that the pool size that will not exceed
```

```
# the maximum number of connections allowed by your database. Defaults to 0 and 5.
```

```
DATABASE_CONNECTION_POOL_MIN=
```

```
DATABASE_CONNECTION_POOL_MAX=
```

```
# Uncomment this line if you will not use SSL for connecting to Postgres. This is acceptable
```

```
# if the database and the application are on the same machine.
```

```
# PGSSLMODE=disable
```

```
# _____
```

```
# _____ REDIS _____
```

```
# _____
```

```
# The Redis URL for your environment you can either specify an ioredis compatible url or a Base64
```

```
# encoded configuration object.
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/redis-LGM4BFXyp4
```

```
REDIS_URL=redis://redis:6379
```

```
# To enable horizontal scaling of the collaboration service you must provide a Redis URL, it
may
# be the same as above, or a different server.
# DOCS: https://docs.getoutline.com/s/hosting/doc/horizontal-scaling-hkfU5Stao7
REDIS_COLLABORATION_URL=

# _____
# _____ FILE STORAGE _____
# _____

# Specify what storage system to use. Possible value is one of "s3" or "local".
# For "local" images and document attachments will be saved on local disk, for "s3" they
# will be stored in an S3-compatible network store.
# DOCS: https://docs.getoutline.com/s/hosting/doc/file-storage-N4M0T6Ypu7
FILE_STORAGE=local

# If "local" is configured for FILE_STORAGE above, then this sets the parent directory under
# which all attachments/images are stored. Make sure that the process has permissions to
# create this path and also to write files to it.
FILE_STORAGE_LOCAL_ROOT_DIR=/var/lib/outline/data

# Maximum allowed size for the uploaded attachment.
FILE_STORAGE_UPLOAD_MAX_SIZE=262144000

# Override the maximum size of document imports, generally this should be lower
# than the document attachment maximum size.
FILE_STORAGE_IMPORT_MAX_SIZE=

# Override the maximum size of workspace imports, these can be especially large
# and the files are temporary being automatically deleted after a period of time.
FILE_STORAGE_WORKSPACE_IMPORT_MAX_SIZE=

# To support uploading of images for avatars and document attachments in a distributed
# architecture, an s3-compatible storage can be configured if FILE_STORAGE=s3 above.
AWS_ACCESS_KEY_ID=get_a_key_from_aws
AWS_SECRET_ACCESS_KEY=get_the_secret_of_above_key
AWS_REGION=xx-xxxx-x
AWS_S3_ACCELERATE_URL=
AWS_S3_UPLOAD_BUCKET_URL=http://s3:4569
```

```
AWS_S3_UPLOAD_BUCKET_NAME=bucket_name_here
```

```
AWS_S3_FORCE_PATH_STYLE=true
```

```
AWS_S3_ACL=private
```

```
# _____
```

```
# _____ SSL _____
```

```
# _____
```

```
# Base64 encoded private key and certificate for HTTPS termination. This is one  
# of three ways to configure SSL and can be left empty.
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/ssl-pzk7W08d1n
```

```
SSL_KEY=
```

```
SSL_CERT=
```

```
# Auto-redirect to https in production. The default is true but you may set to  
# false if you can be sure that SSL is terminated at an external loadbalancer.
```

```
FORCE_HTTPS=true
```

```
# _____
```

```
# _____ AUTHENTICATION _____
```

```
# _____
```

```
# Third party signin credentials, at least ONE OF EITHER Google, Slack,  
# Discord, or Microsoft is required for a working installation or you'll  
# have no sign-in options.
```

```
# Slack sign-in provider
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/slack-sgMujR8J9J
```

```
SLACK_CLIENT_ID=get_a_key_from_slack
```

```
SLACK_CLIENT_SECRET=get_the_secret_of_above_key
```

```
# Google sign-in provider
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/google-h0uvtCmTqQ
```

```
GOOGLE_CLIENT_ID=
```

```
GOOGLE_CLIENT_SECRET=
```

```
# Microsoft Entra / Azure AD sign-in provider
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/microsoft-entra-UVz6jsI0cv
```

```
AZURE_CLIENT_ID=  
AZURE_CLIENT_SECRET=  
AZURE_RESOURCE_APP_ID=  
  
# Discord sign-in provider  
# DOCS: https://docs.getoutline.com/s/hosting/doc/discord-g4JdWFFub6  
DISCORD_CLIENT_ID=  
DISCORD_CLIENT_SECRET=  
DISCORD_SERVER_ID=  
DISCORD_SERVER_ROLES=  
  
# Generic OIDC provider  
# DOCS: https://docs.getoutline.com/s/hosting/doc/oidc-8CPBm6uC0I  
OIDC_CLIENT_ID=  
OIDC_CLIENT_SECRET=  
OIDC_AUTH_URI=  
OIDC_TOKEN_URI=  
OIDC_USERINFO_URI=  
OIDC_LOGOUT_URI=  
  
# Specify which claims to derive user information from  
# Supports any valid JSON path with the JWT payload  
OIDC_USERNAME_CLAIM=preferred_username  
  
# Display name for OIDC authentication  
OIDC_DISPLAY_NAME=OpenID Connect  
  
# Space separated auth scopes.  
OIDC_SCOPES=openid profile email  
  
# _____  
# _____ EMAIL _____  
# _____  
  
# To support sending outgoing transactional emails such as "document updated" or  
# email sign-in you'll need to connect an SMTP server. Service can be configured  
# with any service from this list: https://community.nodemailer.com/2-0-0-beta/setup-smtp/well-known-services/  
# DOCS: https://docs.getoutline.com/s/hosting/doc/smtp-cqCJyZGMIB
```

```
SMTP_SERVICE=  
SMTP_USERNAME=  
SMTP_PASSWORD=  
SMTP_FROM_EMAIL=
```

```
# _____  
# _____ RATE LIMITER _____  
# _____
```

```
# Whether the rate limiter is enabled or not
```

```
RATE_LIMITER_ENABLED=true
```

```
# Individual endpoints have hardcoded rate limits that are enabled  
# with the above setting, however this is a global rate limiter  
# across all requests
```

```
RATE_LIMITER_REQUESTS=1000
```

```
RATE_LIMITER_DURATION_WINDOW=60
```

```
# _____  
# _____ INTEGRATIONS _____  
# _____
```

```
# GitHub integration allows previewing issue and pull request links
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/github-GchT3NNxI9
```

```
GITHUB_CLIENT_ID=
```

```
GITHUB_CLIENT_SECRET=
```

```
GITHUB_WEBHOOK_SECRET=
```

```
GITHUB_APP_NAME=
```

```
GITHUB_APP_ID=
```

```
GITHUB_APP_PRIVATE_KEY=
```

```
# Linear integration allows previewing issue links as rich mentions
```

```
LINEAR_CLIENT_ID=
```

```
LINEAR_CLIENT_SECRET=
```

```
# For a complete Slack integration with search and posting to channels the
```

```
# following configs are also needed in addition to Slack authentication:
```

```
# DOCS: https://docs.getoutline.com/s/hosting/doc/slack-G2mc8D0JHK
```

```
SLACK_VERIFICATION_TOKEN=your_token
SLACK_APP_ID=A0XXXXXXX
SLACK_MESSAGE_ACTIONS=true

# Figma integration allows previewing design files as rich mentions
FIGMA_CLIENT_ID=
FIGMA_CLIENT_SECRET=

# For Dropbox integration, follow these instructions to get the key
https://www.dropbox.com/developers/embedder#setup
# and do not forget to whitelist your domain name in the app settings
DROPBOX_APP_KEY=

# Optionally enable Sentry (sentry.io) to track errors and performance,
# DOCS: https://docs.getoutline.com/s/hosting/doc/sentry-jxcFttcDl5
SENTRY_DSN=
SENTRY_TUNNEL=

# Enable importing pages from a Notion workspace
# DOCS: https://docs.getoutline.com/s/hosting/doc/notion-2v6g7WY3l3
NOTION_CLIENT_ID=
NOTION_CLIENT_SECRET=

# The Iframely integration allows previews of third-party content within Outline.
# For example, hovering over an external link will show a preview.
# DOCS: https://docs.getoutline.com/s/hosting/doc/iframely-HwLF1EZ9mo
IFRAMELY_URL=
IFRAMELY_API_KEY=

# _____
# _____  DEBUGGING  _____
# _____

# Have the installation check for updates by sending anonymized statistics to
# the maintainers
ENABLE_UPDATES=true

# Debugging categories to enable – you can remove the default "http" value if
# your proxy already logs incoming http requests and this ends up being duplicative
```

```
DEBUG=http
```

```
# Configure lowest severity level for server logs. Should be one of
```

```
# error, warn, info, http, verbose, debug, or silly
```

```
LOG_LEVEL=info
```

Revision #9

Created 2024-07-23 08:54:17 UTC

Updated 2026-02-02 21:01:47 UTC