

version: "3.4"

services:

**postgresql:**

```
image: docker.io/library/postgres:12-alpine
restart: unless-stopped
healthcheck:
  test: ["CMD-SHELL", "pg_isready -d ${POSTGRES_DB} -U ${POSTGRES_USER}"]
  start_period: 20s
  interval: 30s
  retries: 5
  timeout: 5s
volumes:
  - ./database:/var/lib/postgresql/data
environment:
  POSTGRES_PASSWORD: ${PG_PASS:?database password required}
  POSTGRES_USER: ${PG_USER:-authentik}
  POSTGRES_DB: ${PG_DB:-authentik}
env_file:
  - .env
```

PostgreSQL wird installiert  
Version 12 wird verwendet  
Container startet sich neu, solange er nicht gestoppt wird

Tests des Containers, um sicherzustellen, dass er läuft, wie lange er läuft und ab wann wie oft wiederholt wird

Erstellung eines Speichermediums im Unterordner von Authentik **database**

Anforderung der Umgebungsvariablen aus der zu erstellten **.env**-Datei

Redis hilft bei der Zwischenspeicherung  
Version alpine wird verwendet  
Befehl: Betrieb von Redis soll aufgenommen werden

Erstellung des Speichermediums im Unterordner von Authentik **redis**

Authentik Server mit dem zu ziehenden Image (Abbild) der spezifischen Version

Command: Start des Servers  
Referenzen zu o.g. Redis und PostgreSQL

Zuordnung der freien Ports für HTTP und HTTPS

abhängig von postgresql und redis

Worker entlastet den Hauptserver

Definition der Volumes (Speichermedien), der inkludierten Datenbank und Redis

**redis:**

```
image: docker.io/library/redis:alpine
command: --save 60 1 --loglevel warning
restart: unless-stopped
healthcheck:
  test: ["CMD-SHELL", "redis-cli ping | grep PONG"]
  start_period: 20s
  interval: 30s
  retries: 5
  timeout: 3s
volumes:
  - ./redis:/data
```

**server:**

```
image: ${AUTHENTIK_IMAGE:-ghcr.io/goauthentik/server}:${AUTHENTIK_TAG:-2023.8.3}
restart: unless-stopped
command: server
environment:
  AUTHENTIK_REDIS_HOST: redis
  AUTHENTIK_POSTGRES_HOST: postgresql
  AUTHENTIK_POSTGRES_USER: ${PG_USER:-authentik}
  AUTHENTIK_POSTGRES_NAME: ${PG_DB:-authentik}
  AUTHENTIK_POSTGRES_PASSWORD: ${PG_PASS}
volumes:
  - ./media:/media
  - ./custom-templates:/templates
env_file:
  - .env
ports:
  - "${COMPOSE_PORT_HTTP:-9000}:9000"
  - "${COMPOSE_PORT_HTTPS:-9443}:9443"
depends_on:
  - postgresql
  - redis
```

**worker:**

```
image: ${AUTHENTIK_IMAGE:-ghcr.io/goauthentik/server}:${AUTHENTIK_TAG:-2023.8.3}
restart: unless-stopped
command: worker
environment:
  AUTHENTIK_REDIS_HOST: redis
  AUTHENTIK_POSTGRES_HOST: postgresql
  AUTHENTIK_POSTGRES_USER: ${PG_USER:-authentik}
  AUTHENTIK_POSTGRES_NAME: ${PG_DB:-authentik}
  AUTHENTIK_POSTGRES_PASSWORD: ${PG_PASS}
# `user: root` and the docker socket volume are optional.
# See more for the docker socket integration here:
# https://goauthentik.io/docs/outposts/integrations/docker
# Removing `user: root` also prevents the worker from fixing the permissions
# on the mounted folders, so when removing this make sure the folders have the correct UID/GID
# (1000:1000 by default)
user: root
volumes:
  - /var/run/docker.sock:/var/run/docker.sock
  - ./media:/media
  - ./certs:/certs
  - ./custom-templates:/templates
env_file:
  - .env
depends_on:
  - postgresql
  - redis
```

volumes:

```
database:
  driver: local
redis:
  driver: local
```