

# Containers

- [samba](#)
- [rsync-server](#)
- [postgres](#)
- [hass](#)
- [mysql](#)
- [nginx](#)
- [photoprism](#)
- [resilio-sync](#)
- [vaultwarden](#)
- [gitea](#)
- [traefik](#)
- [tt-rss](#)
- [mongo](#)
- [bookstack](#)
- [webdav](#)

# samba

```
version: '3.8'

services:
  samba:
    image: dperson/samba
    restart: always
    networks:
      samba_net:
        ipv4_address: 10.1.0.20
    volumes:
      - /opt/samba:/mnt/share:rw
    command: >
      -s "public;/mnt/share;yes;no;yes;all"
      -p
  tmpfs:
    - /tmp
  environment:
    TZ: "Asia/Irkutsk"
    USERID: "1000"
    GROUPLD: "1000"
    ANON: "yes"
    ANON_UID: "65534"
    ANON_GID: "65534"
  ports:
    - "137:137/udp"
    - "138:138/udp"
    - "139:139/tcp"
    - "445:445/tcp"
  deploy:
    resources:
      limits:
        cpus: '0.5'
        memory: '512M'
    placement:
      constraints: [node.role == worker]
```

networks:

samba\_net:

driver: macvlan

driver\_opts:

parent: enp3s0

ipam:

driver: default

config:

- subnet: 10.1.0.0/21

gateway: 10.1.0.1

# rsync-server

```
version: "2.1"
services:
  rsync-server:
    image: apnar/rsync-server
    container_name: rsync-server
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Asia/Irkutsk
      - USERNAME=user
      - PASSWORD=user
      - VOLUME=/backup
      - ALLOW=0.0.0.0/0
    volumes:
      - backup:/backup
    ports:
      - 873:873
    restart: unless-stopped

volumes:
  backup:
    external: true

networks:
  default:
    name: sync
```

# postgres

```
version: '3.7'
services:
  pgsql:
    container_name: postgres
    image: timescale/timescaledb:latest-pg15
    restart: always
    environment:
      - POSTGRES_PASSWORD=password
      - POSTGRES_USER=root
      - POSTGRES_DB=postgres
    volumes:
      - /opt/pgsql/data : /var/lib/postgresql/data
    ports:
      - "5432:5432"
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U root -d postgres"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 10s
    restart: unless-stopped
    deploy:
      resources:
        limits:
          cpus: '4'
          memory: 16G

networks:
  default:
    name: databases
```

# hass

```
version: '3'
services:
  homeassistant:
    container_name: homeassistant
    image: "ghcr.io/home-assistant/home-assistant: stable"
    volumes:
      - /opt/hass:/config
      - /etc/localtime:/etc/localtime:ro
    restart: unless-stopped
    privileged: true
    network_mode: host
```

# mysql

```
version: '3.9'
```

```
services:
```

```
  db:
```

```
    container_name: mysql
```

```
    image: mariadb
```

```
    restart: always
```

```
    command: mysqld --port=3306 --innodb-strict-mode=1 --innodb-buffer-pool-size=256M --  
transaction-isolation=READ-COMMITTED --character-set-server=utf8mb4 --collation-  
server=utf8mb4_unicode_ci --max-connections=512 --innodb-rollback-on-timeout=OFF --innodb-  
lock-wait-timeout=120
```

```
    volumes:
```

- /opt/mysql/data: /var/lib/mysql
- /opt/mysql/etc: /etc/mysql/conf.d

```
    ports:
```

- 3306: 3306

```
    environment:
```

```
      MARIADB_AUTO_UPGRADE: "1"
```

```
      MARIADB_INITDB_SKIP_TZINFO: "1"
```

```
      MARIADB_ROOT_PASSWORD: "password"
```

```
  adminer:
```

```
    container_name: adminer
```

```
    image: adminer
```

```
    restart: always
```

```
    ports:
```

- 8086: 8080

```
networks:
```

```
  default:
```

```
    name: databases
```

```
    driver: bridge
```

```
    external: true
```

# nginx

```
version: '3'

services:
  webserver:
    image: nginx:latest
    container_name: nginx
    ports:
      - 80:80
      - 443:443
    command: '/bin/sh -c ''while :; do sleep 6h & wait $$!}; nginx -s reload; done & nginx -g
"daemon off;''''
    restart: unless-stopped
    volumes:
      - /opt/nginx/conf:/etc/nginx/conf.d:ro
      - /opt/nginx/certbot/www:/var/www/certbot/
      - /opt/nginx/certbot/conf:/etc/nginx/ssl/
      - /opt/nginx/certbot/conf:/etc/letsencrypt/
  certbot:
    image: certbot/certbot:latest
    container_name: certbot
    volumes:
      - /opt/nginx/certbot/www:/var/www/certbot/:rw
      - /opt/nginx/certbot/conf:/etc/letsencrypt/:rw
    entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 12h & wait
$$!}; done;'"

networks:
  default:
    name: webserver
```

# photoprism

```
version: '3.5'

services:
  photoprism:
    container_name: photo
    image: photoprism/photoprism:latest
    security_opt:
      - seccomp:unconfined
      - apparmor:unconfined
    ports:
      - "2344:2342"
    environment:
      PHOTOPRISM_INIT: "http"
      PHOTOPRISM_UID: ${UID:-1000}
      PHOTOPRISM_GID: ${GID:-1000}                # group id
      PHOTOPRISM_ADMIN_USER: "admin"             # admin username
      PHOTOPRISM_ADMIN_PASSWORD: "photoprism"    # initial admin password (minimum 8
characters)
      PHOTOPRISM_AUTH_MODE: "password"           # authentication mode (public,
password)
      PHOTOPRISM_SITE_URL: "https://photo/"      # server URL in the format
"http(s)://domain.name(:port)/(path)"
      PHOTOPRISM_SITE_CAPTION: "Latest"
      PHOTOPRISM_SITE_DESCRIPTION: "Tags and finds pictures without getting in your way!"
      PHOTOPRISM_SITE_AUTHOR: "@3err0"
      PHOTOPRISM_DEBUG: "false"
      PHOTOPRISM_READONLY: "false"
      PHOTOPRISM_EXPERIMENTAL: "false"
      PHOTOPRISM_HTTP_MODE: "release"
      PHOTOPRISM_HTTP_HOST: "0.0.0.0"
      PHOTOPRISM_HTTP_PORT: 2342
      PHOTOPRISM_HTTP_COMPRESSION: "gzip"        # improves transfer speed and
bandwidth utilization (none or gzip)
      PHOTOPRISM_DATABASE_DRIVER: "mysql"
      PHOTOPRISM_DATABASE_SERVER: "10.0.0.1:3306"
```

```
PHOTOPRISM_DATABASE_NAME: "photoprism"
PHOTOPRISM_DATABASE_USER: "root"
PHOTOPRISM_DATABASE_PASSWORD: "password"
PHOTOPRISM_DISABLE_CHOWN: "false"      # disables updating storage permissions via
chmod and chown on startup
PHOTOPRISM_DISABLE_BACKUPS: "true"    # disables backing up albums and photo metadata
to YAML files
PHOTOPRISM_DISABLE_WEBDAV: "true"     # disables built-in WebDAV server
PHOTOPRISM_DISABLE_SETTINGS: "false"  # disables settings UI and API
PHOTOPRISM_DISABLE_PLACES: "false"    # disables reverse geocoding and maps
PHOTOPRISM_DISABLE_EXIFTOOL: "false"  # disables creating JSON metadata sidecar files
with ExifTool
PHOTOPRISM_DISABLE_TENSORFLOW: "false" # disables all features depending on TensorFlow
PHOTOPRISM_DETECT_NSFW: "false"       # automatically flags photos as private that MAY
be offensive (requires TensorFlow)
PHOTOPRISM_UPLOAD_NSFW: "false"       # allows uploads that MAY be offensive (no
effect without TensorFlow)
PHOTOPRISM_RAW_PRESETS: "false"       # enables applying user presets when converting
RAW files (reduces performance)
PHOTOPRISM_THUMB_FILTER: "lanczos"    # resample filter, best to worst: blackman,
lanczos, cubic, linear
PHOTOPRISM_THUMB_UNCACHED: "true"     # enables on-demand thumbnail rendering (high
memory and cpu usage)
PHOTOPRISM_THUMB_SIZE: 2048           # pre-rendered thumbnail size limit (default
2048, min 720, max 7680)
# PHOTOPRISM_THUMB_SIZE: 4096         # Retina 4K, DCI 4K (requires more storage);
7680 for 8K Ultra HD
PHOTOPRISM_THUMB_SIZE_UNCACHED: 7680  # on-demand rendering size limit (default 7680,
min 720, max 7680)
PHOTOPRISM_JPEG_SIZE: 7680            # size limit for converted image files in pixels
(720-30000)
PHOTOPRISM_JPEG_QUALITY: 85           # a higher value increases the quality and file
size of JPEG images and thumbnails (25-100)
TF_CPP_MIN_LOG_LEVEL: 0               # show TensorFlow log messages for development
working_dir: "/photoprism"
volumes:
- photo: /photoprism/originals
# - ". /storage: /photoprism/storage"

networks:
```

default:

name: media

volumes:

photo:

external: true

# resilio-sync

```
version: "2.1"
services:
  resilio-sync:
    image: lscr.io/linuxserver/resilio-sync:latest
    container_name: resilio-sync
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Asia/Irkutsk
    volumes:
      - backup: /sync/backup
      - photo: /sync/photo
      - /opt/sync/config: /config
      - /opt/sync/data: /sync
    ports:
      - 8888: 8888
      - 55555: 55555
    restart: unless-stopped

volumes:
  backup:
    external: true
  photo:
    external: true

networks:
  default:
    name: sync
```

# vaultwarden

```
version: "3"
services:
  vaultwarden:
    image: vaultwarden/server:latest
    container_name: vaultwarden
    restart: unless-stopped
    ports:
      - 9445:80 #map any custom port to use (replace 8445 not 80)
    volumes:
      - /opt/bitwarden:/data:rw
    environment:
      - ADMIN_TOKEN=token
      - WEBSOCKET_ENABLED=false
      - SIGNUPS_ALLOWED=true
      # - SMTP_HOST=smtp.yandex.ru
      # - SMTP_FROM=mail@yandex.ru
      # - SMTP_PORT=465
      # - SMTP_SECURITY=starttls
      # - SMTP_USERNAME=mail@yandex.ru
      # - SMTP_PASSWORD=password
      - DOMAIN=https://password

networks:
  default:
    name: apps
```

# gitea

```
version: "3"

networks:
  apps:
    external: false

services:
  server:
    image: gitea/gitea:1.18.3
    container_name: gitea
    environment:
      - USER_UID=1000
      - USER_GID=1000
      - GITEA__database__DB_TYPE=mysql
      - GITEA__database__HOST=10.0.0.1:3306
      - GITEA__database__NAME=git
      - GITEA__database__USER=root
      - GITEA__database__PASSWORD=password
    restart: always
    networks:
      - apps
    volumes:
      - /opt/gitea:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
```

- Gitea with traefik

```
version: "3"

services:
  gitea:
    image: gitea/gitea:1.18.3
```

container\_name: gitea

environment:

- USER\_UID=1000
- USER\_GID=1000
- GITEA\_\_database\_\_DB\_TYPE=mysql
- GITEA\_\_database\_\_HOST=10.10.10.2:3306
- GITEA\_\_database\_\_NAME=base
- GITEA\_\_database\_\_USER=user
- GITEA\_\_database\_\_PASSWD=pass

restart: always

networks:

- web

volumes:

- /opt/gitea: /data
- /etc/timezone: /etc/timezone: ro
- /etc/localtime: /etc/localtime: ro

labels:

- "traefik.enable=true"
- "traefik.http.routers.gitea.rule=Host(`gitea\_url`)"
- "traefik.http.services.gitea-websecure.loadbalancer.server.port=3000"
- "traefik.http.routers.gitea.tls=true"
- "traefik.http.routers.gitea.tls.certresolver=http"

networks:

web:

external: true

# traefik

```
version: '3'

services:
  traefik:
    image: traefik:latest
    container_name: traefik
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    ports:
      - 80:80
      - 443:443
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /opt/traefik/data/traefik.yml:/traefik.yml:ro
      - /opt/traefik/ssl/acme.json:/acme.json
      - /opt/traefik/custom/:/custom/:ro
    networks:
      - web
      - internal
    labels:
      - 'traefik.enable=true'
      - 'traefik.http.routers.traefik.entrypoints=http'
      - 'traefik.http.routers.traefik.rule=Host(`traefik_url`)'
      - 'traefik.http.middlewares.traefik-auth.basicauth.users=root:password'
      - 'traefik.http.middlewares.traefik-https-redirect.redirectscheme.scheme=https'
      - 'traefik.http.routers.traefik.middlewares=traefik-https-redirect'
      - 'traefik.http.routers.traefik-secure.entrypoints=https'
      - 'traefik.http.routers.traefik-secure.rule=Host(`traefik_url`)'
      - 'traefik.http.routers.traefik-secure.middlewares=traefik-auth'
      - 'traefik.http.routers.traefik-secure.tls=true'
      - 'traefik.http.routers.traefik-secure.tls.certresolver=http'
      - 'traefik.http.routers.traefik-secure.service=api@internal'
    networks:

```

```
web:
  external: true
internal:
  external: false
```

- traefik.yml

```
api:
  dashboard: true

entryPoints:
  http:
    address: ":80"
  https:
    address: ":443"

providers:
  docker:
    endpoint: "unix:///var/run/docker.sock"
    exposedByDefault: false
  file:
    directory: /custom
    watch: true

certificatesResolvers:
  http:
    acme:
      email: mail@3err0.ru
      storage: acme.json
      httpChallenge:
        entryPoint: http
```

- custom\_service.yaml in custom directory

```
ttp:
  routers:
    custom-secure:
      rule: "Host(`url_service`)"
      service: "custom-service"
      entrypoints: ["https"]
```

```
middlewares:  
  - "custom-https-redirect"
```

```
tls:
```

```
  certResolver: "http"
```

```
psw:
```

```
  rule: "Host(`url_service`)"
```

```
  entrypoints: ["http"]
```

```
  middlewares:
```

```
    - "custom-https-redirect"
```

```
  service: "custom-service"
```

```
middlewares:
```

```
  custom-https-redirect:
```

```
    redirectScheme:
```

```
      scheme: "https"
```

```
      permanent: true
```

```
services:
```

```
  custom-service:
```

```
    loadBalancer:
```

```
      servers:
```

```
        - url: "http://service:80"
```

# tt-rss

TTRss with traefik and postgresql database

```
version: '3'
services:
  ttrss:
    image: wangqiru/ttrss
    container_name: ttrss
    environment:
      DB_NAME: rss
      DB_USER: user
      DB_PASS: pass
      DB_HOST: 10.10.10.2
      DB_PORT: 5432
      SELF_URL_PATH: https://rss_url
    volumes:
      - /opt/ttrss/plugins:/var/www/plugins.local
    networks:
      - web
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.ttrss.rule=Host(`rss_url`)"
      - "traefik.http.services.ttrss.loadbalancer.server.port=80"
      - "traefik.http.routers.ttrss.tls=true"
      - "traefik.http.routers.ttrss.tls.certresolver=http"
    restart: always
networks:
  web:
    external: true
```

# mongo

```
version: '3.9'

services:
  mongo:
    container_name: mongo
    image: mongo:latest
    restart: always
    volumes:
      - /opt/mongodb/data: /data/db
    ports:
      - 27017:27017
    environment:
      MONGO_INITDB_ROOT_USERNAME: user
      MONGO_INITDB_ROOT_PASSWORD: pass

networks:
  default:
    name: databases
```

# bookstack

- Bookstack with traefik

```
version: '3'
services:
  bookstack:
    image: ghcr.io/linuxserver/bookstack
    container_name: bookstack
    restart: always
    environment:
      - DB_HOST=10.10.10.2:3306
      - DB_DATABASE=base
      - DB_USERNAME=user
      - DB_PASSWORD=password
      - APP_URL=https://docs.3err0.ru
      - STORAGE_TYPE=local
    networks:
      - web
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.bookstack.rule=Host(`docs.3err0.ru`)"
      - "traefik.http.services.bookstack.loadbalancer.server.port=80"
      - "traefik.http.routers.bookstack.tls=true"
      - "traefik.http.routers.bookstack.tls.certresolver=http"
networks:
  web:
    external: true
```

# webdav

```
version: '3'
services:
  webdav:
    image: bytemark/webdav
    container_name: webdav
    restart: unless-stopped
    environment:
      AUTH_TYPE: Basic
      USERNAME: admin
      PASSWORD: password
      SERVER_NAMES: webdav.site
    networks:
      - web
    security_opt:
      - no-new-privileges:true
    volumes:
      - /opt/webdav:/var/lib/dav
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.webdav.entrypoints=http"
      - "traefik.http.routers.webdav.rule=Host(`webdav.site`)"
      - "traefik.http.middlewares.webdav-https-redirect.redirectscheme.scheme=https"
      - "traefik.http.routers.webdav.middlewares=webdav-https-redirect"
      - "traefik.http.routers.webdav-secure.entrypoints=https"
      - "traefik.http.routers.webdav-secure.rule=Host(`webdav.site`)"
      - "traefik.http.routers.webdav-secure.tls=true"
      - "traefik.http.routers.webdav-secure.tls.certresolver=http"
      - "traefik.http.routers.webdav-secure.service=webdav"
      - "traefik.http.services.webdav.loadbalancer.server.port=80"
      - "traefik.docker.network=web"
    networks:
      web:
        external: true
```