

AI-03971 Zammad Docker Compose Inline Postgres

👤 Owner	© Rigel Arcayan
☰ Tags	Project
🕒 Created time	June 24, 2026 8:36 AM

Generate a docker compose for Zammad and use the following specification. I have already an NPM running in my Dockerlab in the dockerlab-network bridge. Include instructions for reverse proxy setup.

Domain-blindcaveman.duckdns.org,
Reachable via zammad.blindcaveman.duckdns.org,
Join to bridge network-dockerlab-network,
Use postgres user-postgres,
Postgres password-SecurePassword1

Troubleshooting Zammad

```
# View ONLY the containers in the Zammad stack (clean, no clutter)
docker compose ps

# Completely shut down the Zammad stack and clear network states
docker compose down

# Start the Zammad stack back up in the background
docker compose up -d

# Force start a specific container (e.g., Nginx)
docker start zammad-nginx
```

```

# View the last 50 lines of logs for Nginx
docker logs zammad-nginx --tail 50

# View the last 50 lines of logs for the Rails server
docker logs zammad-zammad-railsserver-1 --tail 50

# List all containers on the VM that match the name "zammad"
docker ps -a --filter "name=zammad"

# Ultra-clean custom table view (strips out long image names
and IDs)
docker ps -a --filter "name=zammad" --format "table {{.Name
s}}\t{{.Status}}\t{{.Ports}}"

# Check real-time CPU and Memory usage for all running contai
ners
docker stats

```

Here is your updated `docker-compose.yml` file tailored to your specifications, followed by the exact configuration required to hook it up to Nginx Proxy Manager (NPM).

1. Updated `docker-compose.yml`

This setup connects all necessary containers to your existing external bridge network (`dockerlab-network`), updates the PostgreSQL credentials, and configures the environment variables so Zammad knows its public address.

YAML

```

x-shared:
  zammad-service: &zammad-service
  environment: &zammad-environment
  MEMCACHE_SERVERS: ${MEMCACHE_SERVERS:-zammad-memcached:11211}
  POSTGRES_DB: ${POSTGRES_DB:-zammad_production}
  POSTGRES_HOST: ${POSTGRES_HOST:-zammad-postgresql}

```

```

POSTGRES_USER: ${POSTGRES_USER:-user-postgres}
POSTGRES_PASS: ${POSTGRES_PASS:-SecurePassword1}
POSTGRES_PORT: ${POSTGRES_PORT:-5432}
POSTGRES_OPTIONS: ${POSTGRES_OPTIONS:-?pool=50}
POSTGRES_DB_CREATE:
# Tell the Rails backend to trust the secure protocol passed by NPM
NGINX_SERVER_SCHEME: https

# Tell Rails to trust incoming traffic from proxy networks
RAILS_TRUSTED_PROXIES: '127.0.0.1,::1,172.16.0.0/12,192.168.0.0/16,10.0.0.0/8'

# Redis standalone
REDIS_URL: ${REDIS_URL:-redis://zammad-redis:6379}
REDIS_SENTINELS:
REDIS_SENTINEL_NAME:
REDIS_USERNAME:
REDIS_PASSWORD:
REDIS_SENTINEL_USERNAME:
REDIS_SENTINEL_PASSWORD:

S3_URL:
# Backup settings
BACKUP_DIR: "${BACKUP_DIR:-/var/tmp/zammad}"
BACKUP_TIME: "${BACKUP_TIME:-03:00}"
BACKUP_ON_START: "${BACKUP_ON_START:-true}"
HOLD_DAYS: "${HOLD_DAYS:-10}"
TZ: "${TZ:-Europe/Berlin}"

# Required FQDN and proxy configurations for Zammad
ZAMMAD_FQDN: zammad.blindcaveman.duckdns.org
ZAMMAD_HTTP_TYPE: https
NGINX_SERVER_NAME: zammad.blindcaveman.duckdns.org

AUTOWIZARD_JSON:
AUTOWIZARD_RELATIVE_PATH:
ELASTICSEARCH_ENABLED: "true"
ELASTICSEARCH_SCHEMA: http
ELASTICSEARCH_HOST: zammad-elasticsearch
ELASTICSEARCH_PORT: 9200
ELASTICSEARCH_USER:
ELASTICSEARCH_PASS:
ELASTICSEARCH_NAMESPACE:
ELASTICSEARCH_REINDEX:

image: ${IMAGE_REPO:-ghcr.io/zammad/zammad}:${VERSION:-7.1.0-0000}
restart: ${RESTART:-always}
networks:
  - dockerlab-network
volumes:
  - zammad-backup:/var/tmp/zammad:ro
  - zammad-storage:/opt/zammad/storage
depends_on:

```

```

zammad-memcached:
  condition: service_healthy
zammad-postgresql:
  condition: service_healthy
zammad-redis:
  condition: service_healthy

services:
zammad-backup:
  <<: *zammad-service
  command: ["zammad-backup"]
  volumes:
    - zammad-backup:/var/tmp/zammad
    - zammad-storage:/opt/zammad/storage
  user: 0:0

zammad-elasticsearch:
  image: elasticsearch:${ELASTICSEARCH_VERSION:-9.4.2}
  restart: ${RESTART:-always}
  networks:
    - dockerlab-network
  volumes:
    - elasticsearch-data:/usr/share/elasticsearch/data
  environment:
    discovery.type: single-node
    xpack.security.enabled: "false"
    ES_JAVA_OPTS: ${ELASTICSEARCH_JAVA_OPTS:--Xms1g -Xmx1g}

zammad-init:
  <<: *zammad-service
  command: ["zammad-init"]
  depends_on:
    zammad-postgresql:
      condition: service_healthy
  restart: on-failure
  user: 0:0

zammad-memcached:
  command: memcached -m 256M
  image: memcached:${MEMCACHE_VERSION:-1.6.42-alpine}
  restart: ${RESTART:-always}
  networks:
    - dockerlab-network
  healthcheck:
    test: ["CMD", "nc", "-z", "127.0.0.1", "11211"]
    interval: 10s
    timeout: 5s
    start_period: 10s
    retries: 5

zammad-nginx:
  <<: *zammad-service

```

```

container_name: zammad-nginx
command: ["zammad-nginx"]
expose:
  - "${NGINX_PORT:-8080}"
depends_on:
  zammad-railsserver:
    condition: service_healthy
networks:
  - dockerlab-network # <-- Explicitly anchoring it to the bridge here

# zammad-nginx:
# <<: *zammad-service
# command: ["zammad-nginx"]
# expose:
#   - "${NGINX_PORT:-8080}"
# # Removed host port mapping since NPM will route traffic internally via the bridge
network
# depends_on:
#   zammad-railsserver:
#     condition: service_healthy

zammad-postgresql:
environment:
  POSTGRES_DB: ${POSTGRES_DB:-zammad_production}
  POSTGRES_USER: user-postgres
  POSTGRES_PASSWORD: SecurePassword1
image: postgres:${POSTGRES_VERSION:-17.10-alpine}
restart: ${RESTART:-always}
networks:
  - dockerlab-network
volumes:
  - postgresql-data:/var/lib/postgresql/data
healthcheck:
  test: ["CMD-SHELL", "pg_isready -U $$POSTGRES_USER -d $$POSTGRES_DB"]
  interval: 10s
  timeout: 5s
  start_period: 60s
  retries: 5

zammad-railsserver:
<<: *zammad-service
command: ["zammad-railsserver"]
healthcheck:
  test: ["CMD", "curl", "-sf", "http://127.0.0.1:${ZAMMAD_RAILS_SERVER_PORT:-3000}"]
  interval: 30s
  timeout: 5s
  start_period: 120s
  retries: 3

zammad-redis:
image: redis:${REDIS_VERSION:-8.8-alpine}
restart: ${RESTART:-always}

```

```

networks:
  - dockerlab-network
volumes:
  - redis-data:/data
healthcheck:
  test: ["CMD", "redis-cli", "ping"]
  interval: 10s
  timeout: 5s
  start_period: 10s
  retries: 5

zammad-scheduler:
  <<: *zammad-service
  command: ["zammad-scheduler"]

zammad-websocket:
  <<: *zammad-service
  command: ["zammad-websocket"]

volumes:
  elasticsearch-data:
    driver: local
  postgresql-data:
    driver: local
  redis-data:
    driver: local
  zammad-backup:
    driver: local
  zammad-storage:
    driver: local

networks:
  dockerlab-network:
    external: true

```

2. Nginx Proxy Manager (NPM) Configuration

Because both NPM and Zammad share the `dockerlab-network` bridge, NPM can target the Zammad frontend container directly using its container name.

Proxy Host Settings

In your NPM admin UI, click **Add Proxy Host** and fill out the **Details** tab:

- **Domain Names:** `zammad.blindcaveman.duckdns.org`
- **Scheme:** `http`

- **Forward Hostname / IP:** `zammad-nginx` (*Docker handles the internal DNS resolution*)
- **Forward Port:** `8080`
- **Cache Assets:** Enabled
- **Block Common Exploits:** Enabled
- **Websockets Support: Enabled** (*Crucial for Zammad's live-updating interface*)

SSL Settings

Switch to the **SSL** tab:

- **SSL Certificate:** Select or request your Let's Encrypt certificate for the domain.
- **Force SSL:** Enabled
- **HTTP/2 Support:** Enabled
- **HSTS Enabled:** Optional (Recommended)

Advanced Settings (Required for File Uploads)

Zammad handles attachments and tickets that can easily exceed default proxy upload limits. To prevent `413 Request Entity Too Large` errors when uploading attachments, go to the **Advanced** tab and insert:

Nginx

```
client_max_body_size 50M;
```