

# AI-03964 Fix Zammad Docker Setup

Owner	Ⓜ Rigel Arcayan
Tags	Project
Created time	June 22, 2026 8:00 PM

Create a docker-compose.yml for Zammad app but configure it with a separate postgres-shared configuration which running in dockerlab-network bridge. Also, provide steps on how to configure the Zammad db in pgAdmin to create the db and user. Also, I have an NPM proxy (dockerlab-network) which I need instructions on to setup the proxy host

Domain-blindcaveman.duckdns.org

Host-postgres-shared

bridge network-dockerlab-network

user-postgres

Postgres password-SecurePassword1

domain-blindcaveman.duckdns.org

path-/home/rarcayan/dockerlab/apps/zammad (I want zammad here)

Here is the complete setup to deploy Zammad using your external shared PostgreSQL database, along with configurations for pgAdmin and Nginx Proxy Manager (NPM).

## 1. pgAdmin Configuration (Database & User Creation)

Before spinning up Zammad, you need to create the dedicated database and user inside your shared PostgreSQL container ( `postgres-shared` ).

1. Log into your **pgAdmin** web interface.
2. Expand your server connection to `postgres-shared` .
3. **Create the Role (User):**

- Right-click **Login/Group Roles**  $\rightarrow$  **Create**  $\rightarrow$  **Login/Group Role...**
- **General** tab  $\rightarrow$  Name: `zammad`
- **Definition** tab  $\rightarrow$  Password: Choose a secure password (e.g., `ZammadSecurePass123` )
- **Privileges** tab  $\rightarrow$  Toggle **Can login?** to **Yes**.
- Click **Save**.

#### 4. Create the Database:

- Right-click **Databases**  $\rightarrow$  **Create**  $\rightarrow$  **Database...**
- **General** tab  $\rightarrow$  Database: `zammad_production`
- **General** tab  $\rightarrow$  Owner: Select `zammad` from the dropdown list.
- Click **Save**.

## 2. Zammad `docker-compose.yml`

Create your directory structure at your preferred path:

Bash

```
mkdir -p /home/rarcayan/dockerlab/apps/zammad
cd /home/rarcayan/dockerlab/apps/zammad
nano docker-compose.yml
```

Paste the following configuration. This file uses the official Zammad images, connects to your existing `dockerlab-network`, bypasses an embedded database service, and explicitly points to `postgres-shared`.

YAML

```
version: '3.8'

services:
  zammad-nginx:
    image: zammad/zammad-docker-compose:6.2.0-79
```

```

container_name: zammad-nginx
command: ["zammad-nginx"]
expose:
  - "8080"
volumes:
  - zammad-data:/opt/zammad
environment:
  - NGINX_PORT=8080
restart: always
depends_on:
  - zammad-railsserver
  - zammad-websocket

zammad-railsserver:
image: zammad/zammad-docker-compose:6.2.0-79
container_name: zammad-railsserver
command: ["zammad-railsserver"]
volumes:
  - zammad-data:/opt/zammad
environment:
  - POSTGRESQL_HOST=postgres-shared
  - POSTGRESQL_USER=zammad
  - POSTGRESQL_PASS=ZammadSecurePass123
  - POSTGRESQL_DB=zammad_production
  - ELASTICSEARCH_HOST=zammad-elasticsearch
  - ELASTICSEARCH_PORT=9200
restart: always
depends_on:
  - zammad-elasticsearch

zammad-websocket:
image: zammad/zammad-docker-compose:6.2.0-79
container_name: zammad-websocket
command: ["zammad-websocket"]
volumes:
  - zammad-data:/opt/zammad
environment:
  - POSTGRESQL_HOST=postgres-shared
  - POSTGRESQL_USER=zammad
  - POSTGRESQL_PASS=ZammadSecurePass123
  - POSTGRESQL_DB=zammad_production
restart: always

zammad-worker:
image: zammad/zammad-docker-compose:6.2.0-79
container_name: zammad-worker
command: ["zammad-worker"]
volumes:
  - zammad-data:/opt/zammad
environment:
  - POSTGRESQL_HOST=postgres-shared
  - POSTGRESQL_USER=zammad

```

```

- POSTGRESQL_PASS=ZammadSecurePass123
- POSTGRESQL_DB=zammad_production
- ELASTICSEARCH_HOST=zammad-elasticsearch
- ELASTICSEARCH_PORT=9200
restart: always
depends_on:
  - zammad-elasticsearch

zammad-elasticsearch:
  image: zammad/zammad-docker-compose:6.2.0-79
  container_name: zammad-elasticsearch
  command: ["zammad-elasticsearch"]
  volumes:
    - elasticsearch-data:/usr/share/elasticsearch/data
  restart: always

volumes:
  zammad-data:
  elasticsearch-data:

networks:
  default:
  external:
    name: dockerlab-network

```

**Note:** Ensure you substitute `ZammadSecurePass123` with the actual database user password you provisioned in pgAdmin. Run `docker compose up -d` to build the stack.

### 3. Nginx Proxy Manager (NPM) Configuration

Since all components (NPM, Zammad, and Postgres) share the `dockerlab-network` bridge, NPM can reach the Zammad web container directly via its internal container name.

1. Open your **Nginx Proxy Manager** dashboard.
2. Navigate to **Hosts**  $\rightarrow$  **Proxy Hosts**  $\rightarrow$  **Add Proxy Host**.
3. Configure the **Details** tab:
  - **Domain Names:** `blindcaveman.duckdns.org`
  - **Scheme:** `http`

- **Forward Hostname / IP:** `zammad-nginx` (*Docker's internal DNS resolves this directly*)
  - **Forward Port:** `8080`
  - **Block Common Exploits:** Toggle **On**
  - **Websockets Support:** Toggle **On** (*Crucial for Zammad's live updates*)
4. Configure the **SSL** tab:
- **SSL Certificate:** Select your existing certificate or choose *Request a new SSL Certificate*.
  - **Force SSL:** Toggle **On**
  - **HTTP/2 Support:** Toggle **On**
5. Click **Save**.