

Mattermost

Deploying Mattermost on Docker involves setting up containers for Mattermost itself and optionally for a database backend. Here's a basic guide to get you started:

Step 1: Create Docker Compose File

Create a `docker-compose.yml` file to define Mattermost and optionally a database service (MySQL in this example):

```
version: '3'

services:
  app:
    image: mattermost/mattermost-team:latest
    depends_on:
      - db
    ports:
      - "8065:8065"
    environment:
      - MM_SERVICESETTINGS_SITEURL=http://localhost:8065
      - MM_SQLSETTINGS_DRIVERNAME=mysql
      - MM_SQLSETTINGS_DATASOURCE=mmuser:password@tcp(db:3306)/mattermost?charset=utf8mb4,utf8&readTimeout=30s&writeTimeout=30s
    restart: unless-stopped

  db:
    image: mysql:5.7
    environment:
      MYSQL_DATABASE: mattermost
      MYSQL_USER: mmuser
      MYSQL_PASSWORD: password
      MYSQL_ROOT_PASSWORD: root_password
    volumes:
      - ./mysql:/var/lib/mysql
```

restart: always

Step 2: Deploy Mattermost

Run the following command in the directory containing your `docker-compose.yml` file:

```
docker-compose up -d
```

This command will download the Mattermost and MySQL Docker images (if not already downloaded) and start the containers in detached mode (`-d`).

Step 3: Access Mattermost

Access Mattermost through your browser at `http://localhost:8065`. Replace `localhost` with your server's IP address or domain name if accessing remotely.

Optional Steps:

- **Persistent Storage:** The MySQL container in the example uses a volume (`./mysql:/var/lib/mysql`) for persistent storage. Adjust paths and volumes as needed for your setup.
- **Custom Configuration:** Modify environment variables (`MM_*`) in the `docker-compose.yml` file to suit your specific requirements, such as SMTP settings, TLS configurations, etc.
- **Security:** Consider securing Mattermost with HTTPS using a reverse proxy like Nginx and obtaining SSL/TLS certificates.

Notes:

- **MySQL Setup:** If you prefer using a managed MySQL database, adjust `MM_SQLSETTINGS_DATASOURCE` in the `app` service environment variables to connect to your managed MySQL instance.
- **Database Choice:** While MySQL is used in this example, Mattermost also supports PostgreSQL as a database backend. Adjust the `MM_SQLSETTINGS_DRIVERNAME` and `MM_SQLSETTINGS_DATASOURCE` accordingly for PostgreSQL.

This setup provides a basic deployment of Mattermost on Docker, suitable for testing and development environments. Customize further based on your specific deployment needs and security requirements for production use.

Revision #1

Created 11 December 2024 06:01:22 by Ahmad

Updated 11 December 2024 06:01:49 by Ahmad