Deploying Gitea on Docker

Deploying Gitea on Docker involves setting up the Gitea application along with a database backend. Here's a step-by-step guide to deploy Gitea using Docker Compose:

Step 1: Prepare Docker Compose File

Create a `docker-compose.yml` file to define the services (Gitea and MySQL in this case):

```
version: '3'

services:
  server:
    image: gitea/gitea:latest
    environment:
      - USER_UID=1000  # Replace with your host user UID if necessary
      - USER_GID=1000  # Replace with your host group GID if necessary
    restart: always
    volumes:
      - ./gitea:/data
    ports:
      - "3000:3000"
      - "2222:22"  # SSH port (optional, adjust as needed)
    depends_on:
      - db
    networks:
      - gitea_network

  db:
    image: mysql:5.7
    environment:
      MYSQL_DATABASE: gitea
      MYSQL_USER: gitea
      MYSQL_PASSWORD: password
      MYSQL_ROOT_PASSWORD: root_password
    volumes:
```

```
      - ./mysql:/var/lib/mysql
    restart: always
    networks:
      - gitea_network


networks:
  gitea_network:
    driver: bridge
```

Step 2: Configure Docker Compose File

- **Gitea (`server` service)**:
  - `image` : Pulls the latest Gitea Docker image.
  - `environment` : Sets user UID and GID for file permissions (adjust if necessary).
  - `volumes` : Mounts a local directory `./gitea` to persist Gitea data.
  - `ports` : Maps container ports `3000` (HTTP) and `2222` (SSH) to host ports (adjust as needed).
  - `depends_on` : Ensures the `db` service starts before Gitea.
  - `networks` : Connects to a custom bridge network `gitea_network` .
- **MySQL (`db` service)**:
  - `image` : Pulls the MySQL 5.7 Docker image.
  - `environment` : Sets MySQL database name, user, and passwords.
  - `volumes` : Mounts a local directory `./mysql` to persist MySQL data.
  - `restart` : Always restarts the MySQL container if it stops unexpectedly.
  - `networks` : Connects to the `gitea_network` for communication with Gitea.

Step 3: Deploy Gitea

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

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

This command starts both Gitea and MySQL containers in detached mode (`-d`), allowing them to run in the background.

Step 4: Access Gitea

After deployment, access Gitea through your browser at `http://localhost:3000` . Replace `localhost` with your server's IP address or domain name if accessing remotely.

Step 5: Initial Setup

Follow the on-screen instructions to complete the initial setup of Gitea, including setting up an admin account and configuring basic settings.

Notes:

- **Data Persistence**: Data for Gitea and MySQL will persist in local directories `./gitea` and `./mysql`. Adjust paths and volumes as needed.
- **Customization**: Customize environment variables and ports according to your specific requirements.
- **Security**: Consider securing Gitea with HTTPS using Nginx or another reverse proxy and obtaining SSL/TLS certificates.

This setup provides a basic deployment of Gitea using Docker Compose. Modify as per your environment and security requirements for production use.

Revision #2
Created 11 December 2024 03:36:36 by Ahmad
Updated 11 December 2024 03:37:02 by Ahmad