








This document provides an overview of the functionality and configuration of the services within the **Liberty Framework**, including **Node.js**, **PostgreSQL**, **pgAdmin**, **Airflow**, **OIDC**, and **Gitea**. These services are integrated with **Traefik** as a reverse proxy, enabling both HTTP and HTTPS access with automated routing.



---

## 1. Node.js Service ( `liberty-node` )



- **Image:** `ghcr.io/fblettner/liberty-node:latest`
  - **Command:** Runs the Node.js app ( `app.js` ) on port `3002`.
  - **Security Options:**
    -  `label:disable` : Disables SELinux labels.
    -  `cap_drop` : Removes unnecessary Linux capabilities like `MKNOD` and `AUDIT_WRITE`.
  - **Networks:** Connected to the `liberty-network`.
  - **Working Directory:** `/opt/liberty`
  - **Depends on:** PostgreSQL ( `pg` ) service.
  - **Traefik Configuration:**
    -  **API Routing:** HTTP and HTTPS routing for `/api` using `PathPrefix`.
    -  **Socket Routing:** HTTP and HTTPS routing for `/socket` and `/socket.io`.
    -  **React Application:** Handles HTTP and HTTPS routing for the React app with a middleware for error pages.
    -  **Compression:** `compress-middleware` applied to several routes for better performance.
    -  **Port Configuration:** Node.js runs on port `3002`.
- 

## 2. PostgreSQL Service ( `liberty-pg` )


- **Image:** `ghcr.io/fblettner/liberty-pg:latest`
- **Command:** Runs the PostgreSQL server with optimized settings for performance:
  - `shared_buffers=2GB`
  - `track_activity_query_size=1MB`
  - `work_mem=256MB`
  - `maintenance_work_mem=128MB`
- Other configurations to optimize WAL size, checkpoint timing, and costs.
- **Volumes:** Data stored in the `pg-data` volume.

- **Networks:** Connected to `liberty-network`.
  - **Traefik Configuration:**
  -  **TCP Router:** Routes PostgreSQL traffic via `db` entry point.
  -  **Port:** Exposed on port `5432`.
- 






## 3. pgAdmin Service ( `liberty-pgadmin` )

- **Image:** `ghcr.io/fblettner/liberty-pgadmin:latest`
  - **User:** Root privileges enabled.
  - **Volumes:** pgAdmin data stored in the `pgadmin-data` volume.
  - **Environment:** Sets the `SCRIPT_NAME=/pgadmin` for pgAdmin web access.
  - **Depends on:** PostgreSQL ( `pg` ).
  - **Networks:** Connected to `liberty-network`.
  - **Traefik Configuration:**
  -  **HTTP Router:** Routes requests for `/pgadmin`.
  -  **Port:** Exposed on port `3003`.
- 




## 4. Airflow Service ( `liberty-airflow` )

- **Image:** `ghcr.io/fblettner/liberty-airflow:latest`
  - **Security Options:**
  -  Disables SELinux labels.
  -  Drops capabilities `MKNOD` and `AUDIT_WRITE`.
  - **Volumes:**
  - Logs stored in the `airflow-logs` volume.
  - **Depends on:** PostgreSQL ( `pg` ), Gitea ( `gitea` ).
  - **Networks:** Connected to `liberty-network`.
  - **Traefik Configuration:**
  -  **Routing:** Handles HTTP and HTTPS requests for `/airflow/home`.
  -  **Error Pages Middleware:** Applied to both HTTP and HTTPS routes.
  -  **Port:** Exposed on port `8080`.
-

## 5. OIDC Service ( liberty-keycloak )

- **Image:** `ghcr.io/fblettner/liberty-keycloak:latest`
  - **Command:** Starts the Keycloak OIDC server with proxy headers and hostname settings.
  - **Environment Variables:**
    -  `PROXY_ADDRESS_FORWARDING` : Enables proxy address forwarding.
    -  `KC_HOSTNAME_PATH` and `KC_HTTP_RELATIVE_PATH` : Configured to `/oidc` .
  - **Depends on:** PostgreSQL ( `pg` ).
  - **Networks:** Connected to `liberty-network` .
  - **Traefik Configuration:**
    -  **HTTP and HTTPS Routing:** Routes `/oidc` requests.
    -  **Port:** OIDC runs on port `9080` (Keycloak internally uses port `8080` ).
    -  **CORS Middleware:** Configures Cross-Origin Resource Sharing (CORS) for all origins and credentials.
- 

## 6. Gitea Service ( liberty-gitea )

- **Image:** `ghcr.io/fblettner/liberty-gitea:latest`
  - **Healthcheck:** Ensures service health by checking `/` endpoint every 30 seconds.
  - **Volumes:**
    - Configuration and data in `liberty-gitea` .
  - **Restart Policy:** Set to `unless-stopped` .
  - **Networks:** Connected to `liberty-network` .
  - **Traefik Configuration:**
    -  **Routing:** Routes HTTP requests to `/gitea` .
    -  **Middleware:** Uses `stripprefix` to remove `/gitea` from the path for internal routing.
    -  **Port:** Exposed on port `3000` .
- 

## Volumes

- **node-logs:** Stores Logs for backend and frontend.
- **pg-data:** Stores PostgreSQL data.
- **pg-logs:** Stores Logs for database.
- **pgadmin-data:** Stores pgAdmin data.

- **liberty-gitea**: Stores gitea config and data.
  - **airflow-logs**: Stores logs for Airflow.
  - **airflow-dags**: Stores Dags for Airflow.
  - **airflow-plugins**: Stores Plugins for Airflow.
  - **traefik-certs**: Stores Traefik certificates (external).
  - **traefik-config**: Stores Traefik configuration (external).
  - **shared-data**: Stores shared data (external).
- 

## Networks

- **liberty-network**: External network for inter-service communication.
- 

This configuration enables a scalable, containerized microservice architecture with **Node.js** for application logic, **PostgreSQL** for database management, **pgAdmin** for database administration, **Airflow** for automation, **Keycloak OIDC** for authentication, and **Gitea** for file management and versioning. **Traefik** serves as the reverse proxy, handling routing and applying security middleware for all services.