

This guide will walk you through deploying Liberty Framework using Portainer, based on the Compose file located at the following URL: [liberty-framework.yml](https://github.com/nomana-it/liberty-framework/blob/main/compose.yml).

## Prerequisites

Before you begin, ensure the following prerequisites are met:

- You have Docker installed and running on your server. Installation instructions can be found [here](#).
- You have Portainer installed and running on your server. Installation instructions can be found [here](#).
- You have access to the Portainer web interface. The URL typically looks like `http://your-server-ip:3000` or `https://your-server-ip:3443`.

## Accessing Portainer

1. Open a web browser and navigate to the Portainer web interface.
2. Log in with your Portainer credentials.
3. Set a password first time you log into Portainer

## Logging into a Custom Registry

1. In the Portainer web interface, navigate to `Registries` from the sidebar.
2. Click on the `+ Add registry` button.
3. Provide the following details for your custom registry:
  - **Name:** A friendly name for your registry.
  - **URL:** The URL of your custom registry (e.g., `ghcr.io/fblettner`).
  - **Username:** Your registry username (this user will be provided by Nomana-IT).
  - **Password:** Your registry password (this token will be provided by Nomana-IT).
4. After filling in the details, click on the `Add Registry` button to save the registry.

## Deploy the Stack

1. In the Portainer web interface, navigate to `Stacks` from the sidebar.
2. Click on the `+ Add Stack` button.
3. Provide a name for your stack in the `Name` field.
4. Under the `Git repository` tab:

- Enter the **Repository URL**:

```
1 | https://github.com/fblettner/liberty-public
```

- In the **Compose path** field, specify:

```
1 | release/latest/liberty-framework.yml
```

5. Scroll down and click on the `Deploy the stack` button.

## Verify Deployment

1. Once the stack is deployed, navigate to `Containers` from the sidebar.
2. Verify that the containers listed in the Compose file are running.
3. Access the services through the designated ports to ensure everything is functioning as expected.

## Alternative: Pull Docker Images from Terminal

If you prefer to pull Docker images directly from the terminal, you can do so using the following commands:

1. Open a terminal and log in to the custom registry:

```
1 | docker login ghcr.io
```

When prompted, enter your username and password (token).

2. Pull the required Docker images manually:

```
1 | docker pull ghcr.io/fblettner/liberty-node:latest
2 | docker pull ghcr.io/fblettner/liberty-pg:latest
3 | docker pull ghcr.io/fblettner/liberty-pgadmin:latest
4 | docker pull ghcr.io/fblettner/liberty-rundeck:latest
5 | docker pull ghcr.io/fblettner/liberty-keycloak:latest
6 | docker pull ghcr.io/fblettner/liberty-filebrowser:latest
```

## Steps for AWS Users

If you are using AWS and need to connect via AWS CLI, follow these steps:

1. Configure your AWS CLI:

```
1 | aws configure
```

Follow the prompts to enter your AWS Access Key, Secret Access Key, default region name, and output format.

## 2. Log in to the AWS Elastic Container Registry (ECR):

```
1 | aws ecr get-login-password --region eu-west-1 | docker login --username AWS --password-stdin  
  | <your-aws-account-id>.dkr.ecr.eu-west-1.amazonaws.com
```

Replace `<your-aws-account-id>` with your actual AWS account ID.

## Additional Resources

- [Portainer Documentation](#)
- [Docker Compose Documentation](#)
- [GitHub Repository - liberty-framework.yml](#)

By following this guide, you should be able to deploy Liberty Framework using Portainer seamlessly. If you run into any issues or have any questions, refer to the additional resources provided or reach out to the respective support communities.

## Summary

**URLs:** - **Web Application:** `/` - **API:** `/api` - **PgAdmin:** `/pgadmin` - **Rundeck:** `/rundeck` - **OIDC:** `/oidc` - **Filebrowser:** `/filebrowser`

**Services:** - **node:** ghcr.io/fblettner/liberty-node:latest (Port 3002) - **pg:** ghcr.io/fblettner/liberty-pg:latest (Port 5432) - **pgadmin:** ghcr.io/fblettner/liberty-pgadmin:latest (Port 3003) - **rundeck:** ghcr.io/fblettner/liberty-rundeck:latest (Port 4440) - **oidc:** ghcr.io/fblettner/liberty-keycloak:latest (Port 8080) - **filebrowser:** ghcr.io/fblettner/liberty-filebrowser:latest (Port 80)

Details of all Liberty Framework Services can be found [here](#).