

Deployment Manual for SmartHirePro by Team Kodak

1. Docker

1.1. Audience Definition

This manual is intended for system administrators or IT personnel responsible for deploying the software.

1.2. Platform-Specific Deployment Instructions

This application is containerized using Docker, so the deployment instructions are the same for all platforms that support Docker.

1.3. Docker Installation

- **Windows/Mac:** Install Docker Desktop from [Docker official website](#).
- **Linux:** Use the appropriate package manager for your distro. For Ubuntu, you can use “sudo apt install docker.io”

1.4. Prerequisite Installation

Before deploying the software, Docker and Docker Compose (if applicable) need to be installed on the system. Docker Compose is included with Docker Desktop on Windows and Mac, but on Linux, you'll need to install it separately. You can download it from the [Docker Compose GitHub repository](#).

1.5. Configuration Instructions

The Dockerfile and docker-compose.yml (if applicable) in the repository contain all the necessary configuration. They specify the base image for the application, install necessary packages, set environment variables, and expose the necessary ports.

1.6. Deployment Scripts or Code Snippets

To build the Docker image, navigate to the directory containing the Dockerfile

- For backend: “2024S-Kodak/Sprint 4/flask api/”
- For frontend: “2024S-Kodak/Sprint 4/SHP/”

Run the following command:

```
“docker build -t my-app .”
```

This command builds a Docker image and tags it as my-app.

To run the application, use the following command:

```
“docker run -d -p <host-port>:<container-port> my-app”
```

This command starts a container from the my-app image in detached mode and maps the specified port in the container to the same port on the host.

If a docker-compose.yml file is present, navigate to its directory and run:

```
“docker-compose up -d”
```

This command will start the application and its dependencies in detached mode.

1.7. Testing and Troubleshooting

After deployment, to test the application by navigate to `http://localhost:<host-port>` in the web browser. If the application doesn't load, check the logs of the Docker container for any error messages:

`"docker logs <container-id>"`

Replace `<container-id>` with the ID of the Docker container. You can get the ID by running `"docker ps"`.

If you encounter any issues, make sure that Docker is running and that the Dockerfile and docker-compose.yml (if applicable) are in the current directory. If the problem persists, please refer to the Docker and Docker Compose documentation or seek help from the Docker community.

2. Manual Installation

2.1. Pre-Deployment Checklist

- 2.1.1. Confirm that the target server meets the minimum system requirements for running React with Node.js and Flask applications.
- 2.1.2. Ensure Node.js, npm (Node Package Manager) and python are installed on the target server.
- 2.1.3. Verify that you have access to the source code repository (e.g., GitHub, Bitbucket) where the Node.js application code is hosted.
- 2.1.4. Update any environment-specific configurations (e.g., database connection strings, API keys) in the application code.
 - Update database details and credentials in config.py
 - Update external api details in config.yaml

2.2. Deployment

2.2.1. Environment setup

2.2.1.1. Linux:

Using Package Manager (apt for Debian/Ubuntu):

`"sudo apt update"`

`"sudo apt install nodejs"`

`"sudo apt install npm"`

`"sudo apt install python3-pip"`

2.2.1.2. Windows:

- Download the Windows installer from the official Node.js website: <https://nodejs.org/en/download>
- Run the installer and follow the installation instructions.
- Repeat the installation instructions for Python on <https://www.python.org/downloads/>

2.2.2. Code Preparation

- 2.2.2.1. If it's a first installation, then clone the repository, if it is an update then Pull the latest version of the application code from the repository:
"git clone <https://github.com/htmhw/2024S-Kodak.git>"
OR
"git pull origin master"
- 2.2.2.2. The application has dependencies listed in the package.json file:
"npm install"
- 2.2.2.3. Similarly the flask_api application has dependencies listed in the requirements.txt file. Use command: "pip install -r requirements.txt"

2.2.3. Deployment

- 2.2.3.1. To deploy the react app first, it needs to be built and started like shown :
"npm run build"
"npm start"
- 2.2.3.2. To deploy flask application, just run the file with main script, in this case:
"python flask_api/app.py"

2.2.4. Rollback procedure (if required)

- Revert codebase to the previous release tag or desired version.
- Repeat steps from procedure 2.2.2.2 onwards.

2.2.5. Testing and Troubleshooting

- After deploying the application, perform thorough testing to ensure that all functionalities are working as expected. System Integration Testing (SIT) and User acceptance testing (UAT) are recommended.
- If the application crashes or stops responding please view server logs to find the root cause and exception.