

Fullstack Development

Preflight project - backend

[Github Repo](#)

Backend applications

- Data storage
- Business logic
- Authentication / authorization
- APIs for frontend frameworks (*depending on architecture*)
- Connection to other services

Node JS backend frameworks

- [Github stars](#)
- [State of JS 2023](#)

Packages

- `npm init -y`
- Express JS
 - `npm i express cors helmet`
- Typescript
 - `npm i typescript ts-node tsconfig-paths`
 - `npm i -D @types/cors @types/express tsc-alias`
- ORM
 - `npm i drizzle-orm postgres dotenv`
 - `npm i -D drizzle-kit nodemon`

Package explanation

Package	Details
<code>express</code>	Express (<i>duh!</i>)
<code>helmet</code>	Set default HTTP response header
<code>cors</code>	Enable Cross-origin resource sharing (CORS)
<code>nodemon</code>	Script monitoring during dev

ORM code

- All [files](#) in `./db` folder.
 - No need to copy `migration` folder.
 - You can change schema.
- `.\drizzle.config.ts` ([Link](#))

Files

- `./.env` (Copy from [here](#))
- `./nodemon.json` ([Link](#))
- `./.gitignore` ([Link](#))
- `./tsconfig.json` ([Link](#))
- Scripts in `package.json` ([Link](#))

Note on `tsconfig.json`

- Once we use `outDir` option any non-`./src` folder needs to be included, including the root.

```
{
  "compilerOptions": {
    "outDir": "./dist",
  },
  "include": ["./*", "./src/**/*", "./db/**/*"]
}
```

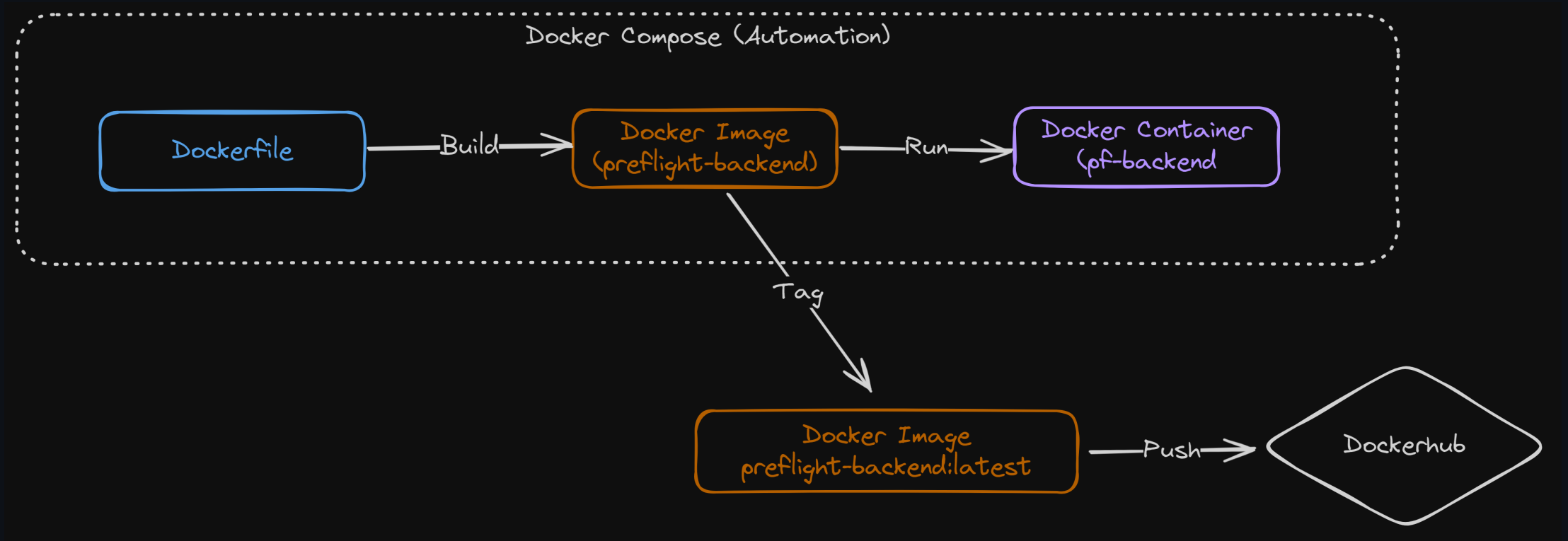
Minimal example

- `./src/index.ts` ([Link](#))
- Start dev
 - `npm run dev`






Full example

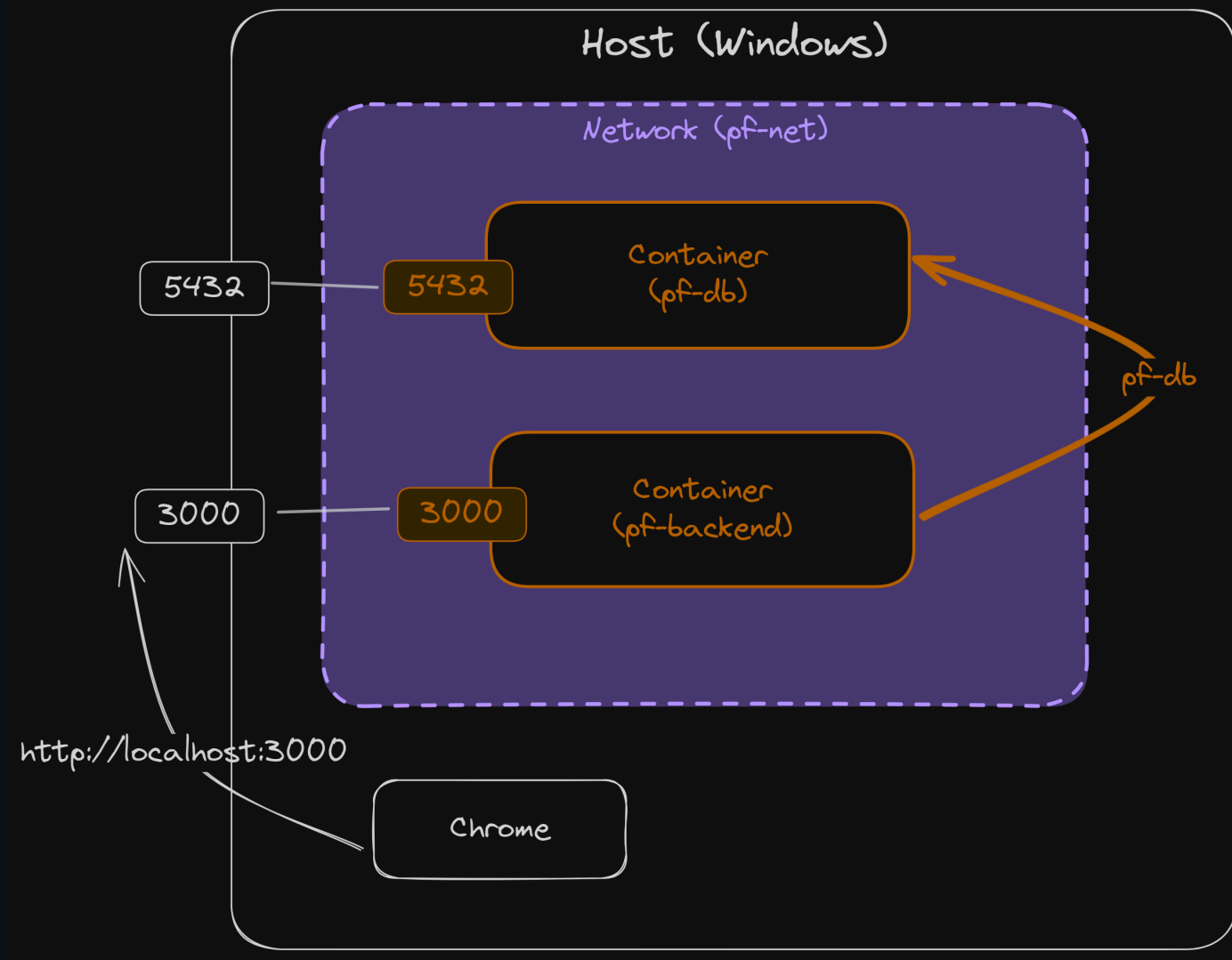
- `./src/index.ts` ([Link](#))
- Build
 - `npm run build`
- Start production
 - `npm run start`

Containerization



Steps

-  `./Dockerfile` ([Link](#))
-  `./dockerignore` ([Link](#))
-  `./docker-compose.yml` ([Link](#))
-  `./.env.test` from `./.env.test.example` ([Link](#))
-  `docker compose --env-file ./.env.test up -d --force-recreate --build`




Dockerhub

- Create an account at <https://hub.docker.com>.
- Create repository called `preflight-backend`.

Push to Dockerhub

- Tag image

-  `docker tag preflight-backend [DOCKERHUB_ACCOUNT]/preflight-backend:latest`

- `docker login`

- Push image

-  `docker push [DOCKERHUB_ACCOUNT]/preflight-backend:latest`

Useful docker commands

- Inspect

- `docker ps`
- `docker network ls`
- `docker volume ls`

- Cleaning

- `docker image prune -a`
- `docker builder prune`
- `docker volume prune`
- `docker network prune`