With Docker - Multiple Deployment Environments

This examples shows how to use Docker with Next.js and deploy to multiple environment with different env values. Based on the deployment documentation.

How to use

Execute `create-next-app` with npm or Yarn to bootstrap the example:

```
npx create-next-app --example with-docker-multi-env nextjs-docker-multi-env
# or
yarn create next-app --example with-docker-multi-env nextjs-docker-multi-env
# or
pnpm create next-app -- --example with-docker-multi-env nextjs-docker-multi-env
```

Enter the values in the `.env.development.sample`, `.env.staging.sample`, `.env.production.sample` files to be used for each environments.

Using Docker and Makefile

Development environment - for doing testing

```
make build-development
make start-development
```

Open http://localhost:3001

Staging environment - for doing UAT testing

```
make build-staging
make start-staging
```

Open http://localhost:3002

Production environment - for users

```
make build-production
make start-production
```

Open http://localhost:3003

Running Locally

First, run the development server:

```
npm run dev
# or
yarn dev
```

Open http://localhost:3000 with your browser to see the result.

You can start editing the page by modifying `pages/index.js`. The page auto-updates as you edit the file.

API routes can be accessed on http://localhost:3000/api/hello. This endpoint can be edited in `pages/api/hello.js`.

The `pages/api` directory is mapped to `/api/*`. Files in this directory are treated as API routes instead of React pages.