# **NC DPR**

## **Requirements**

* Docker=20.10.12
* Docker-Compose=2.2.3

## **Quick Start**

### **Secrets Management**

To run the DPR stack, several secrets need to be installed into the project structure. These secrets are files that are placed under the secrets/ directory and read into appropriate containers at run time. For securit purposes, these files are ignored by version control and must be configured at the local level before runtime. See the following table for a list of secrets:

| **Filename** | **Description** | **Example** |
| --- | --- | --- |
| domain.crt | SSL certificate of the hostname you're using for the stack for https purposes. | -----BEGIN CERTIFICATE----- ... |
| domain.key | Encryption key used in for https. | -----BEGIN PRIVATE KEY----- ... |
| mysql\_root | Password for the MariaDB root user | password |
| legacy\_mysql\_user | User for the legacy stack to access | root |
| legacy\_mysql\_password | Password for the MariaDB root user | password |

### **Run Commands**

The compose stack is started with a call to Docker Compose and configured with an enviorment variable file. The envs/ folder contains .env files that specify different run configurations such as dev and prod. This env file is specified in the command line call to docker compose. See example.env for an explanation of the different environment configuration options.

Development

docker compose --env-file envs/dev.env up --build --force-recreate

Production

docker compose --env-file envs/prod.env up

## Troubleshooting

* A common issue is not running the Docker Daemon before trying to run the Docker Compose command. If the following message is encountered, ensure that Docker is installed and running.

Couldn't connect to Docker daemon at http+unix://var/run/docker.sock - is it running?

If it's at a non-standard location, specify the URL with the DOCKER\_HOST environment variable.

* Another common issue is failing to specify an environment file when running Docker Compose. The compose file requires a fair number of environment variables be set at runtime. These are supplied with the --env-file cmd arg when running docker compose. If not supplied, the console will indicate that they have been set to empty strings and the deployment will exhibit undocumented behavior.
* Something that can happen, especially during development is the persistence of old versions of databases. The DB schema is described in SQL files stored in the db\_schema directory. These files are imported when the MariaDB container is created for the first time and the resultant DB objects are stored on the mounted volume db\_persistance. This volume is ignored by git and needs to be deleted while Mariadb is not running, forcing a recreate, for any schema updates to take affect.
* There has been some inconsistent behavior observed when switching between the dev and prod .env files. It seems like Docker is not always the best at recognizing and rebuilding continers when you change the dockerfile they're supposed to use in Docker Compose. If you ever suspect that changes aren't taking effect, try deleting the containers manually, forcing a recreate.

## Deployment Verification

To check whether the deployment was successful, navigate to localhost. The legacy application stack landing page should be displayed with links to all of DPR's applications properly displayed. Furthermore, one should see the rewritten DPRCal app displayed at localhost/new/dprcal. If you suspect an issue with the database, its schema and contents can be accessed at localhost/dbadmin via PHPMyAdmin.