**January 20, 2016**

**Docker fundamentals**

For the project, it is based around a microservice architecture. This means that each job/task is separated into its own ‘container’. Docker has gained popularity over the last two years for the productivity boost and means of releasing to production in a hassle free manner.

**Containers**

A container allows the running of applications in isolated, virtualized environments. They are light in size and are more comparable to processes than Virtual Machines which are a lot slower.

**Deployment**

With Dockers containers, these and the other features provided from the Docker platform to provide easy means of deployment. Sonrasc can be deployed on a local machine, or a cloud provider such as Amazon Web Services or Digital Ocean.

**The Workflow**

Utilising Docker is to make deployment seamless and hassle free with the right configuration. With the use of a simple bash script that will clone the GitHub repo which contains a Dockerfile containing instructions for what to do.

**Environment setup**

1. Clone the GitHub Repo.
2. Build the Docker image, pulling the necessary images from the Docker Hub.
3. Install dependencies for each container. I.e. the node modules for the front end to work.
4. Use Docker Compose to start the containers images, exposing and mapping the appropriate ports so that they are available publicly outside the internal network.

For local development, I use docker-osx-dev which will watch all files in the working directory and push them to the container providing the directory is mounted. This makes features such as Webpack’s bundling (eg. auto browser refresh on code change) and gulp plugins work like in local development.

It makes it so that technically the whole directory and Dockerfile could be deleted and rebuilt with the execution of one bash script and have the environment perfectly back intact.

As said, when it comes to pushing into production, the system just has to be pushed onto a cloud service (e.g. AWS) for deployment.

Docker compose about and file structure

Dockerfile structure