

Building Custom Docker Images

Building an Image from a Dockerfile

Once you have a Dockerfile, you can tell docker to build it by running the following command:

```
docker build -t tag-name -f path/to/Dockerfile.extension path/to/build/context
i.e.
docker build -t clowder -f ./Dockerfile.clowder .
```

Tagging an Image

Tags are nice for saving a "snapshot" of an image, to make it easily runnable later. This is similar to using Git tags for new releases.

```
docker tag image-name [repository/]new-image-name[:tag-name]
i.e.
docker tag clowder ndslabs/clowder:0.9.1
```

NOTE: Don't forget to push your new tag up to Docker Hub!

Writing a Dockerfile

Docker images are built from files called Dockerfiles.

A simple Dockerfile example can be found below:

```
# Use centos66 as the base image for this build
FROM      centos:centos6

# Enable Extra Packages for Enterprise Linux (EPEL) for CentOS
RUN      yum install -y epel-release
# Install Node.js and npm
RUN      yum install -y nodejs npm

# Copy files from the host into the container
COPY package.json /src/package.json

# Run 'npm install' to install dependencies
RUN cd /src; npm install --production

# Tell Docker we plan to use this port
EXPOSE   8080

# Specify the default command for when the container runs
CMD ["node", "/src/index.js"]
```