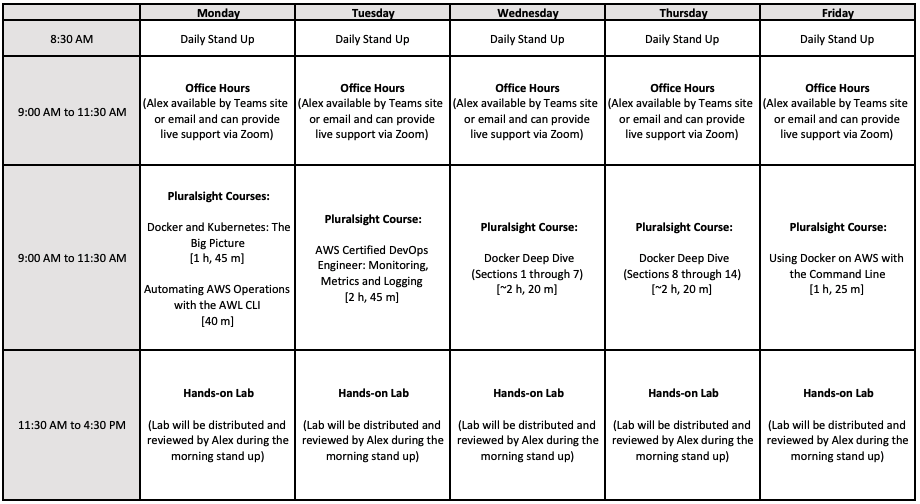
Use the class cheat sheet and materials for reference: <https://bitbucket.org/lmcohort2/materials/src/master/cheatsheet.md>



# Nginx Docker Image - from scratch

Our goal is to create a runnable Nginx container image by writing a Dockerfile that installs and configures Nginx.

## Helpful Links

* Nginx installation docs (<https://www.nginx.com/resources/wiki/start/topics/tutorials/install/>)
* Nginx configuration docs (<https://www.nginx.com/resources/wiki/start/topics/examples/full/>)
* Dockerfile reference (<https://docs.docker.com/engine/reference/builder/>)

## Tips:

* Nginx by default logs to a file, we need to change it's "error\_log" and "access\_log" to log to stdout. stdout can be written to in linux via the /dev/stdout file. You can test this by echoing into /dev/stdout...`echo "hello" > /dev/stdout` will simply output the word "hello". You will need to create an nginx.conf file. nginx looks for this file in /etc/nginx/nginx.conf by default.
* You can run the nginx command in the foreground with 'nginx -g "daemon off;"'

## Requirements:

* Nginx listens on port 80 and serves up the following landing page when hit from browser
  + index.html

|  |
| --- |
| <html>  <head>  <title>Hello, world</title>  </head>  <body>  <h1>Hello, world</h1>  </body>  </html> |

* All Nginx logs appear when running docker logs

## Bonus - Advanced, What's the smallest you can make your image?

* Build your dockerfile "FROM scratch"
* Buildroot (<https://buildroot.org/downloads/manual/manual.html>) can help you create a base linux filesystem
* You can generate your base filesystem in an intermediate stage of your Dockerfile with multi-stage builds. (you can also generate a tarball of the FS outside of the docker build process and copy it in)
* Nginx can be built from source