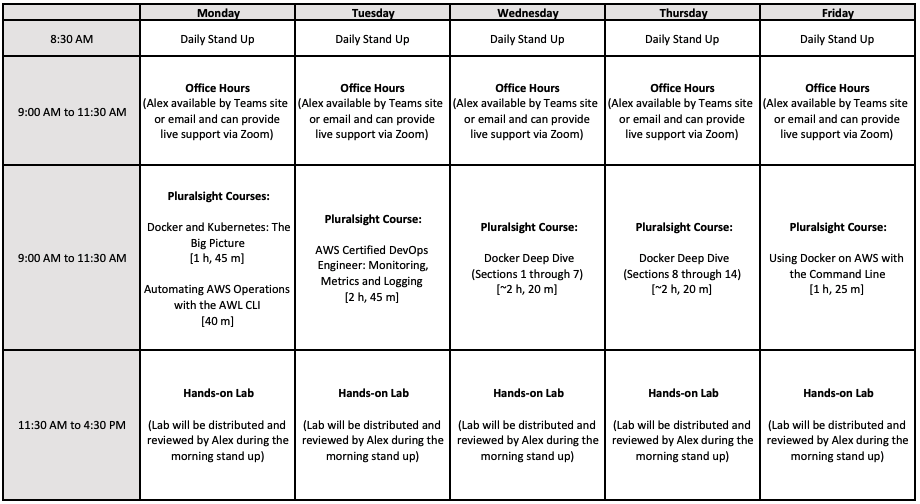
Use the class cheat sheet and materials for reference: <https://bitbucket.org/lmcohort2/materials/src/master/cheatsheet.md>



## Weather - Event-driven Microservice

The weather app you wrote did the following:

* fetch coordinates using ip address
* using coordinates fetch the current weather
* using coordinates and time fetch weather forecast for that time.

For this lab, we are going to make each one of these operations a \*\*Lambda function\*\* that will be callable via an HTTP request.

### 1. Lambdas

Create a lambda for each of the operations above except the fetch IP call; that will need to be done in the client code (the code calling the service). Do you know why this is true?

* Fetch coordinates from ip address
  + You can hard code the ip address you're are using or rely on the lambdas source ip. In the next step, API Gateway will give the client ip information to your lambda in the "context" parameter.
* Use [https://docs.aws.amazon.com/lambda/latest/dg/lambda-python-how-to-create-deploymnt-package.html​](https://docs.aws.amazon.com/lambda/latest/dg/lambda-python-how-to-create-deploymnt-package.html%E2%80%8B) to see how to bundle the requests library with your code for the lambda.

### 2. API Gateway

Create an API in API gateway with endpoints for each lambda

### 3. Client App

Create a command line application in Python that prompts these endpoints to get results.

The client app won’t be too different than the script you have already written. The difference will be that you have to call the lambda functions rather than then APIs you used to get the data from the specific services. Resulting in a simpler script.