Lambda Deployment

Current Problem:

* Lambda Sizes are greater than 50MB
* Serverless is packaging the whole repository inside of each file

Restrictions:

* No deployment bundle over the size 50MB

Solution would include:

* Problem 1
  + Eliminating the error where serverless is bundling the whole repository
* Problem 2
  + Decreasing Lambda deployment sizes to less than 50MB

Problem 1 – Possible Solutions

Option #1:

Serverless package and individuality

I believe that we need to create the correct statement of how to package individually. A solution may be to exclude all files and then in each of the functions specifically include which folder to package.

<https://www.serverless.com/framework/docs/providers/aws/guide/packaging/>

<https://www.serverless.com/framework/docs/providers/aws/guide/deploying/>

The deployment will handle packaging and pointing everything at the right package

Option #2:

Scrap Serverless and go for a full SAM solution (mostly a joke but is a possible solution)

Problem 2 – Possible Solutions:

Option 1# Zip Lambda file and upload them to S3

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-cli-package.html>

This process could be done with an automated script which upon activation from crane would upload zip the file and then upload it to S3 and then replace the cloudformation template with the proper URL to the zipped file in S3

PROS:

* This will still allow for use of cloud9
* No matter the file size the application should be deployable
  + (may only be up to 500MB I believe)

CONS:

* Files will only be compressed and not minimized which could add extra costs to the overall project
* Files will not be at the smallest possible size

Option 2# Lambda Layers

<https://docs.aws.amazon.com/lambda/latest/dg/nodejs-package.html#nodejs-package-dependencies>

By creating a custom runtime at the lambda layers we can upload similar dependencies that lambdas share. This could include axios or a newer version of the SDK etc.

PROS:

* Decreases the number of dependencies to upload reducing file size
* Maintain the use of cloud9

Cons:

* Does not compress the literal size of the other non-shared dependencies

Option 3# Serverless Webpack

<https://www.serverless.com/plugins/serverless-webpack/>

Bundles files in a way that works with webpack. This is a 3rd party plugin for serverless framework that will create bundles of code for lambda deployment

PROS:

* Smallest possible bundle size
* If bundle sizes are below 50MB (which they should) avoid S3 costs

CONS:

* Dependency on a user library for deployments (this means if say babel or webpack updates and we want to update as well we have to wait for the library to be updated by the user).
* Lose the use of Cloud 9

Option 4#: Webpack

<https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/webpack.html>

Bundling the files directly with the webpack and publishing each one separately, and targeting them within the pathways for serverless. I believe the only difficulty will be figuring out how to tell the lambda to use the specific handler.

PROS:

* Smallest possible bundle size
* If bundle sizes are below 50MB (which they should) avoid S3 costs

CONS:

* Lose the use of Cloud 9
* Troubleshooting how to have lambda recognize the handler function

**New Serverless Framework Deployment Architecture Proposal**

**CB Teams Use Either Option Below**

**Use Serverless Webpack**

<https://bitbucket.collegeboard.org/projects/DEX/repos/data-center-migration/browse/batch-rdw-score-data-app/serverless.yml>

**Don’t Use Serverless Webpack**

<https://bitbucket.collegeboard.org/projects/EBRE/repos/ebr-bsbulkreg-pg/browse>

<https://bitbucket.collegeboard.org/projects/EBRE/repos/ebr-jobs-pg-serverless/browse/config/lambda/file>