UDACITY

PROJECT SPECIFICATION

Hosting a Full-Stack Application

Preparing source code infrastructure for deployment

CRITERIA	MEETS SPECIFICATIONS
Write code that demonstrates parameterized environment variables	No environment variables that change from the development environment and production should be present in the source code. A central configuration file is used in order to set the environment variables and make them available to the code. No authentication strings are hard-coded in the source code.
Write a project-level package.json file and organize it properly	A project-level package.json file should contain scripts for running: • Tests • Builds Any new dependencies should be located in the `devDependencies` section of the `package.json`.

CRITERIA	MEETS SPECIFICATIONS
Configure the needed infrastructure for a web application	Screenshots of the AWS console indicate that the following services are properly set up, i.e. healthy and accessible: • AWS RDS for the database • AWS ElasticBeanstalk (or alternatives like lambda) for the API • AWS s3 for web hosting The app is accessible via the link provided.

Configuring Continuous Integration Pipeline with Github

CRITERIA	MEETS SPECIFICATIONS
Trigger a successful pipeline on each push to the main branch	A screenshot of the last build shows that the student's CircleCi account is authorized to access his/her repo on Github and is detecting changes each time he/she is pushing to the main branch. Optionally, a build status badge is present in the README.md, indicating the current state of the main branch build.

CRITERIA	MEETS SPECIFICATIONS
Write a proper pipeline file using the config.yml format used by CircleCi	The submission includes a config.yml that ensures the build occurs in a logical sequence. Comments help explain the flow of the pipeline and are straight to the point. The pipeline file uses correct syntax and can be executed by CircleCi.
Configure secrets via the Continuous Integration software	All the secrets found in the application are configured inside CircleCi and passed to the production application. A screenshot of the configuration screen is present to show where secrets were added.

Documenting Deployment Process

CRITERIA	MEETS SPECIFICATIONS
Write code that demonstrates a well-organized docs folder	A documentation folder should include separate pages on different topics that cannot be discovered by just quickly glancing at code: • Infrastructure description • App dependencies • Pipeline process

CRITERIA	MEETS SPECIFICATIONS
Prepare an architecture diagram to document the deployment flow	The submission contains a simple diagram giving a high-level overview of the infrastructure and another diagram showing the overview of the pipeline. The diagram Includes the different AWS services used for hosting the following: • DB • API • Front-End A representation of the communication between the services is present in the diagram (ex: arrows between services).

Suggestions to Make Your Project Stand Out!

1. Make your pipeline run the front-end unit tests! You will need to figure out how to run mocha in a CI env
2. Make Pull Requests builds so that each time you open a pull request against the main branch, your tests and build are executed.
3. Make separate environments for staging and production in elastic beanstalk and s3. Deploy PR code to the staging and Main to Production.