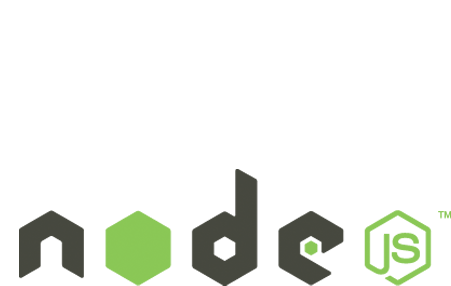
**Tech Stack Wiki**

**Purpose**

The purpose of this wiki is to document evidence on our tech stack and supporting details on technical approach.

**Evidence & Artifacts**

**Attachment E: Criteria #3 - Over 5 Modern, Open Source Tools**

[](https://github.com/cloudamatic) [](https://www.chef.io/) [](https://nodejs.org/)

[](http://gulpjs.com/) [](https://angularjs.org/) [](https://github.com/eGT-Labs/egt-gsa-proto/wiki/www.docker.com)

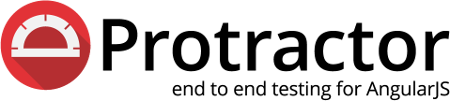
[](https://angular.github.io/protractor/%23/) [](http://getbootstrap.com/2.3.2/)

Figure 1 Open Source Tech Stack

**Attachment E: Criteria #4 - Infrastructure as a Service Provider**

Below is a screenshot for our Infrastructure as a Service Provider, Amazon Web Services.

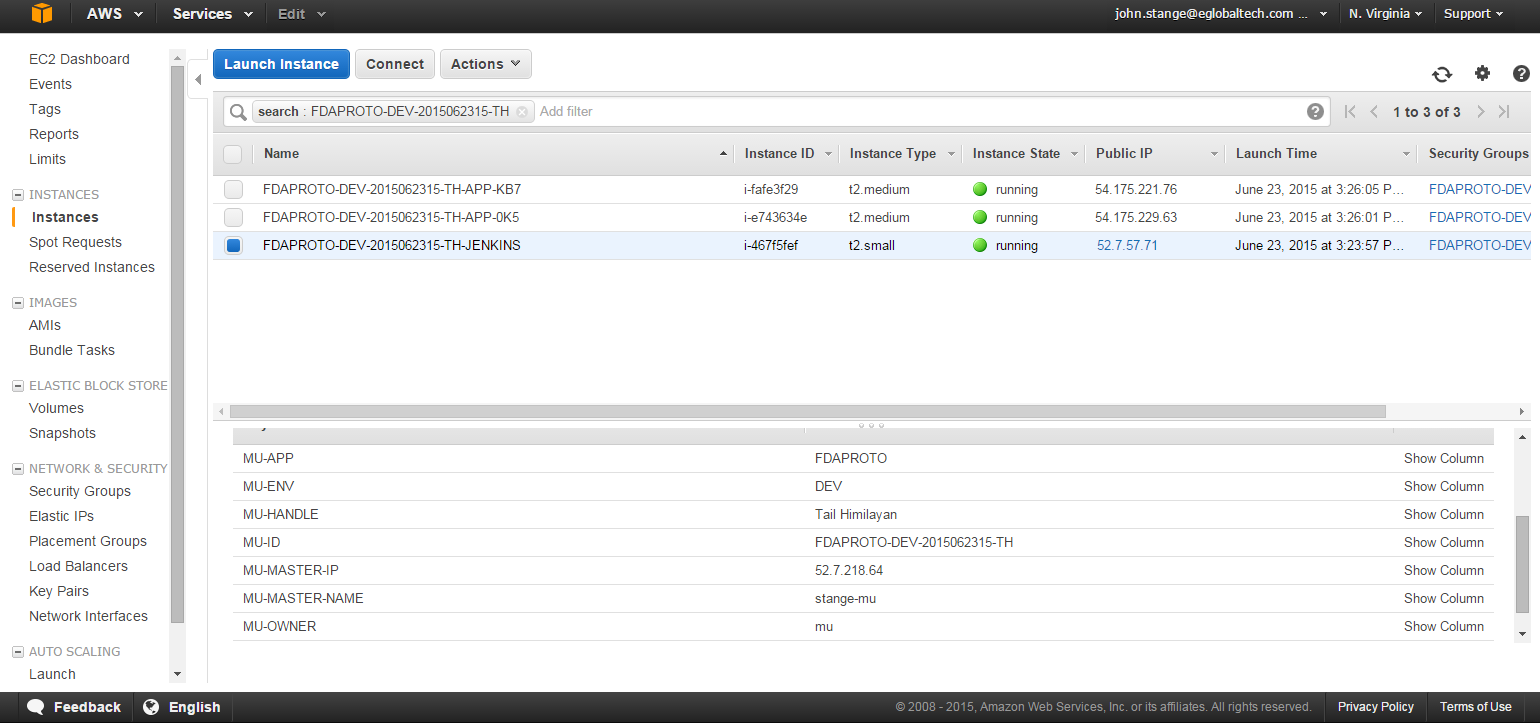


Figure 2 - Infrastructure as a Service - Amazon Web Services

**Attachment E: Criteria #5 – Wrote Unit Tests**

We used Protractor to write automated unit tests. Because the display of data is the biggest feature of our application, the tests center around this feature. Prototyping this application involved minimal testing, but as features are added, testing mocks and request caching would be added to enhance testing.

[Unit Tests](https://github.com/eGT-Labs/egt-gsa-proto/blob/master/client/views)

Sample test:

it('should show search results', function () {

browser.get('/label-search');

var search\_box = element(by.name('fulltext'));

var search = 'TYLENOL';

search\_box.sendKeys(search);

search\_box.submit();

var result = element.all(by.repeater('row in vm.resp.data.results')).get(0).getText();

expect(result).toContain('ACETAMINOPHEN');

});

**Attachment E: Criteria #6 - Continuous Integration System (to automate the running of tests & deployments)**

To rapidly ship product increments and enable asynchronous feedback, we deployed the app on Amazon Web Services using our Continuous Integration framework in Jenkins. Using our test, build, and deploy automation directly from our source control repository, Github, this enabled a rapid and frequent validation from business and end users. It also enabled that the app provides value and promoted keeping the software always working, and enables frequent releases.

Below is an illustration of CI system within our system architecture:

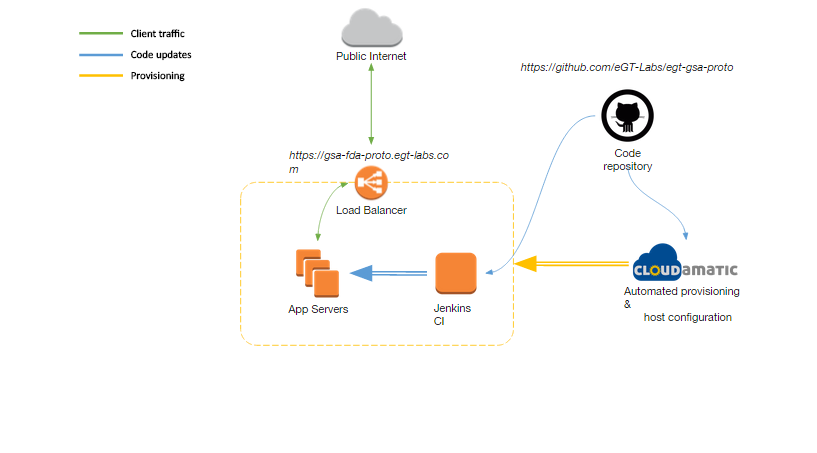


Figure 3 - System Architecture

Below is a sample screenshot from Jenkins:

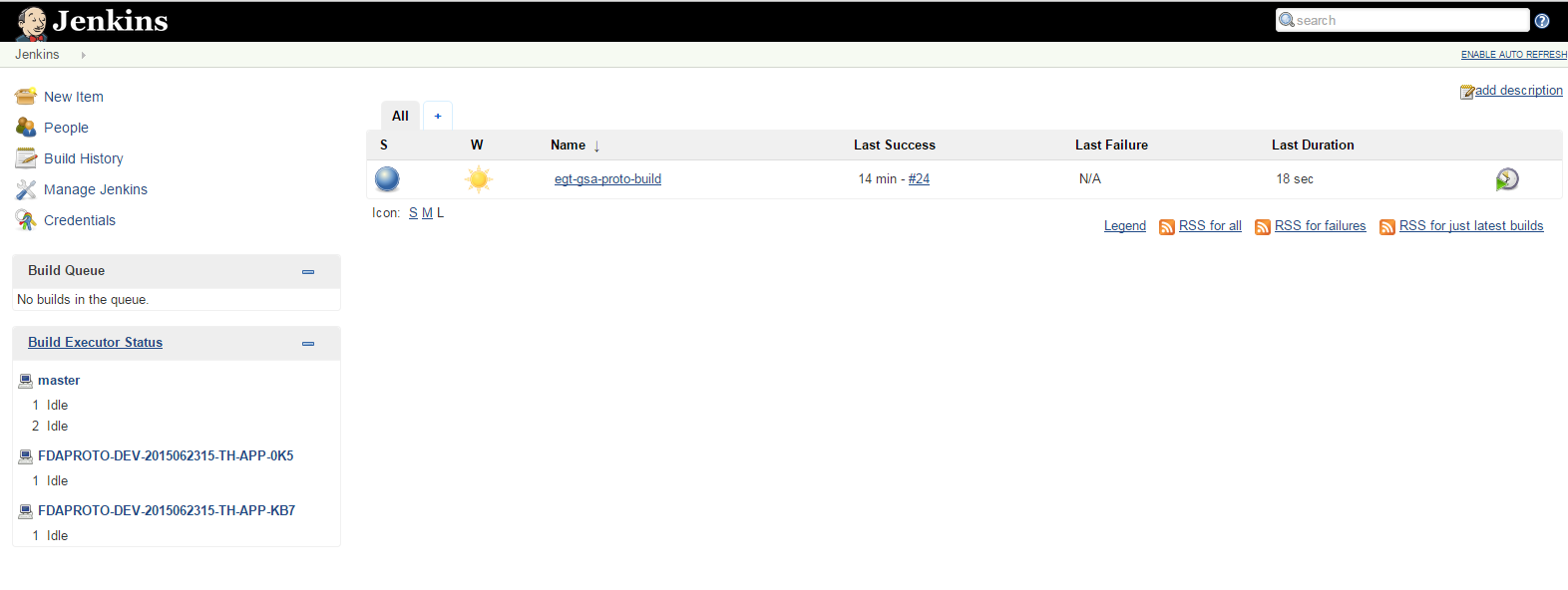


Figure 4 - Continuous Integration - Jenkins

Below is a successful log demonstrating the automated running of tests:

Started by an SCM change

Building remotely on FDAPROTO-DEV-2015062315-TH-APP-0K5 (app-server) in workspace /var/node/workspace/catalyst-testing

> git rev-parse --is-inside-work-tree # timeout=10

Fetching changes from the remote Git repository

Fetching upstream changes from git@github.com:eGT-Labs/egt-gsa-proto.git

> git --version # timeout=10

using GIT\_SSH to set credentials GSA FDA Dataset Prototype Github credentials

> git fetch --tags --progress git@github.com:eGT-Labs/egt-gsa-proto.git +refs/heads/\*:refs/remotes/origin/\*

> git rev-parse origin/master^{commit} # timeout=10

Checking out Revision 608652d45ba10cb9394c0b3e558beaa4cb3fc632 (origin/master)

> git config core.sparsecheckout # timeout=10

> git checkout -f 608652d45ba10cb9394c0b3e558beaa4cb3fc632

> git rev-list cce329aa1da0db395b0e9f784132368f12e7cdf7 # timeout=10

[catalyst-testing] $ /bin/sh -xe /tmp/hudson5596531870567137468.sh

+ mkdir -p /testing

+ export BUILD\_ID=dontKillMe

+ BUILD\_ID=dontKillMe

+ export DISPLAY=:10

+ DISPLAY=:10

+ cd /testing

+ '[' '!' -d /testing/egt-gsa-proto ']'

+ cd egt-gsa-proto

+ git clean -f

+ git reset --hard

HEAD is now at cce329a really switch back to firefox

+ git pull origin master

From github.com:eGT-Labs/egt-gsa-proto

\* branch master -> FETCH\_HEAD

Updating cce329a..608652d

Fast-forward

+ npm install

npm WARN package.json egtGsaProto@0.1.0 No repository field.

npm WARN package.json egtGsaProto@0.1.0 No license field.

+ bower install --allow-root

+ npm install

npm WARN package.json egtGsaProto@0.1.0 No repository field.

npm WARN package.json egtGsaProto@0.1.0 No license field.

+ bower install --allow-root

+ gulp e2e:update

[23:05:28] Using gulpfile /testing/egt-gsa-proto/gulpfile.js

[23:05:28] Starting 'e2e:update'...

selenium standalone is up to date.

chromedriver is up to date.

[23:05:29] Finished 'e2e:update' after 215 ms

+ gulp e2e

[23:05:31] Using gulpfile /testing/egt-gsa-proto/gulpfile.js

[23:05:31] Starting 'sass'...

[23:05:31] Finished 'sass' after 16 ms

[23:05:31] Starting 'inject'...

[23:05:31] gulp-inject 5 files into index.html.

[23:05:31] gulp-inject 19 files into index.html.

[23:05:31] Finished 'inject' after 152 ms

[23:05:31] Starting 'watch'...

[23:05:31] Finished 'watch' after 24 ms

[23:05:31] Starting 'serve'...

[23:05:31] Finished 'serve' after 2.19 ms

[23:05:31] Starting 'e2e'...

[23:05:31] Finished 'e2e' after 1.64 ms

[23:05:31] [nodemon] v1.3.7

[23:05:31] [nodemon] to restart at any time, enter `rs`

[23:05:31] [nodemon] watching: \*.\*

[23:05:31] [nodemon] starting `node server/server.js`

Express server listening on port 9000, in development mode.

OPEN\_FDA\_API\_KEY environment variable NOT provided

Starting selenium standalone server...

[launcher] Running 1 instances of WebDriver

Selenium standalone server started at http://172.31.36.99:35123/wd/hub

[32m.[0m[32m.[0m[32m.[0m[32m.[0m[32m.[0m[32m.[0m[32m.[0m

Finished in 40.217 seconds

[32m7 tests, 9 assertions, 0 failures

[0m

Shutting down selenium standalone server.

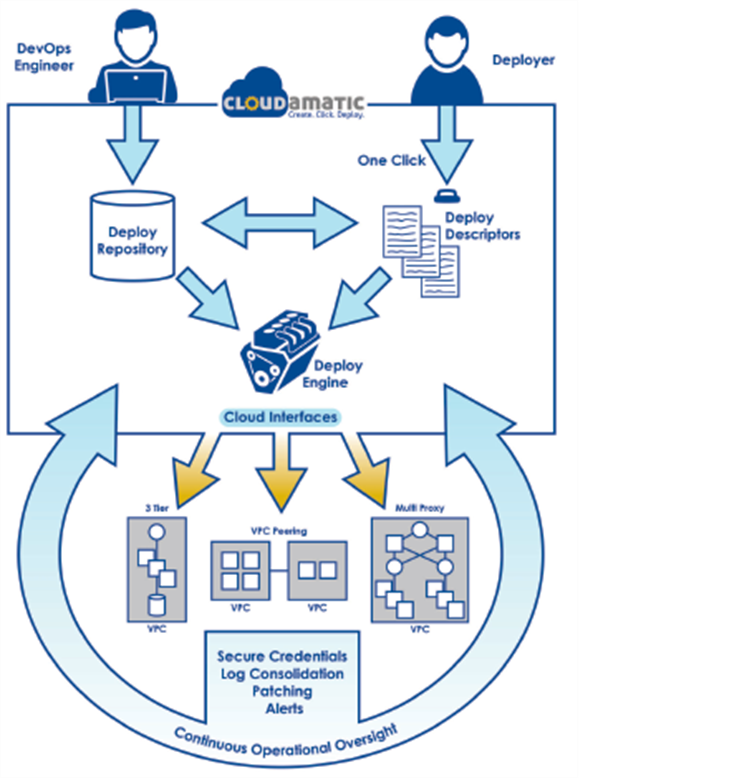
[launcher] 0 instance(s) of WebDriver still running

[launcher] firefox #1 passed

Finished: SUCCESS

**Attachment E: Criteria #7 – Set up or Used Configuration Management**

Catalyst deploys using the open source [Cloudamatic](https://github.com/eGT-Labs/egt-gsa-proto/wiki/cloudamatic.com) toolset, which completely models all aspects of the deploy in a “deployment descriptor”, then uses the descriptor to provision, configure, orchestrate and manage the complete deployment. The figure below illustrates a high level view of Cloudamatic architecture:

Figure 5 - Continuous Deployment - Cloudamatic

Below is a link to a sample deployment log in our repo illustrating our configuration management:

[Deploy Logs](https://github.com/eGT-Labs/egt-gsa-proto/blob/master/docs/Tech%20Stack/deploylog)

**Attachment E: Criteria #8 – Set up or Used Continuous Monitoring**

Below is a screenshot of our uptime & availability continuous monitoring using Nagios:

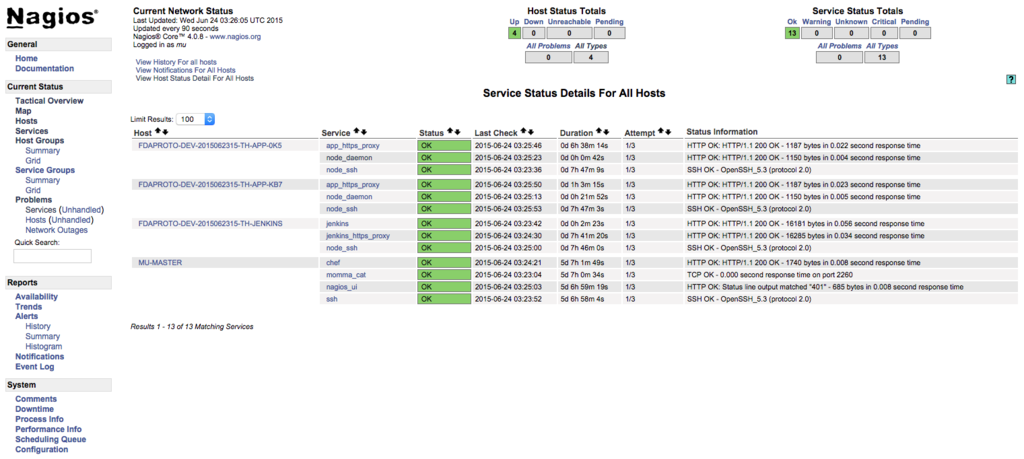


Figure 6 - Continuous Monitoring - Nagios

**Attachment E: Criteria # 9 - Deploy their software in a container**

Below is a screenshot and a link to our Docker README on how to deploy the app in a container:

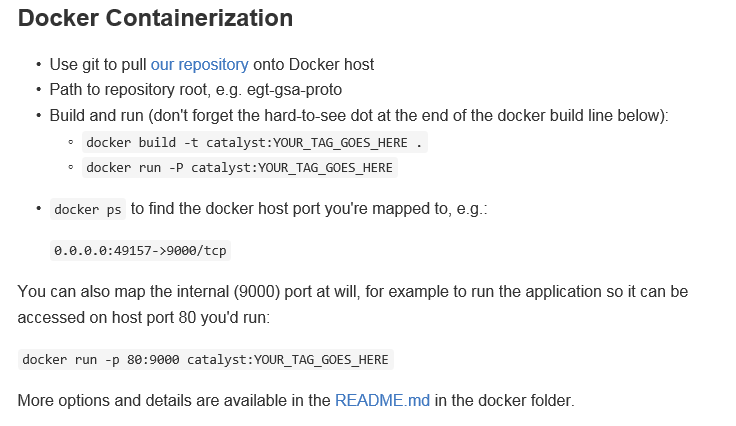


Figure 7 - Container - Docker

For additional information and our evidence see the [README.md](https://github.com/eGT-Labs/egt-gsa-proto/blob/master/docker/README.md) in the docker folder.

**Attachment E: Criteria #10 - Make use of an API**

Below is a screenshot of our README illustrating our evidence for making use of the OpenFDA APIs:

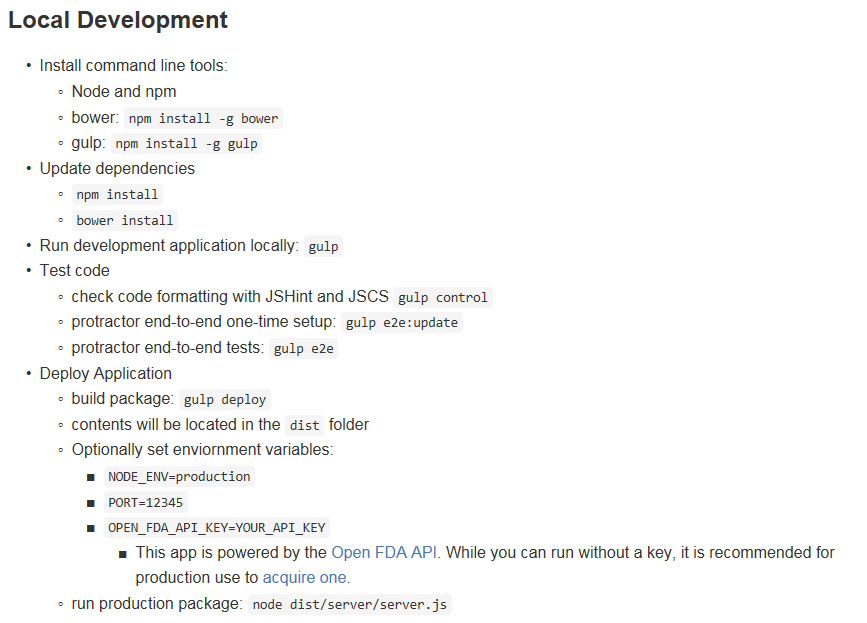


Figure 8 - Make use of API

For additional information and our evidence see the [README.md](https://github.com/eGT-Labs/egt-gsa-proto/blob/master/README.md).

**Attachment E: Criteria #12 - Documentation to install and run the application**

Below is a screenshot of our README illustrating our evidence to install & run the application:

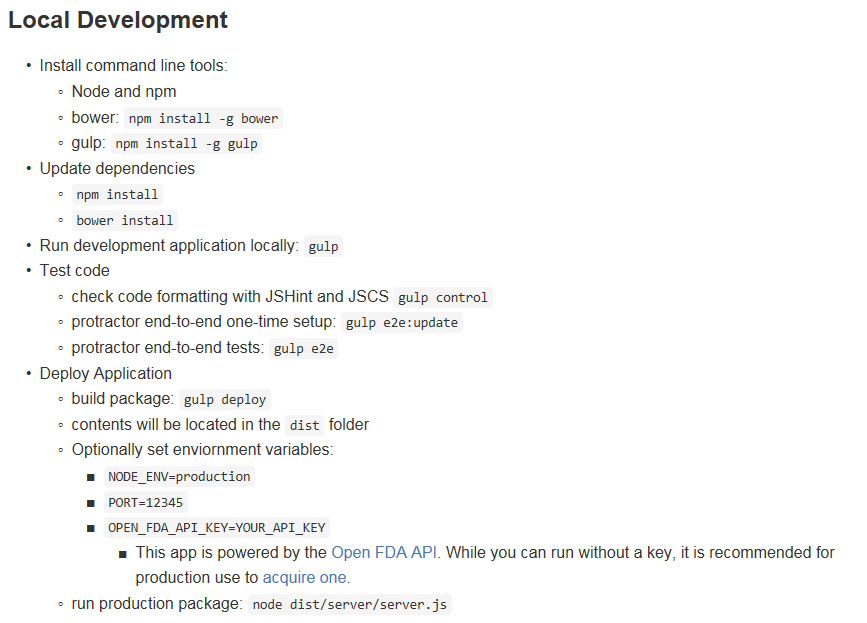


Figure 9 – Install & Run Application

For additional information and our evidence see the [README.md](https://github.com/eGT-Labs/egt-gsa-proto/blob/master/README.md).

**Attachment E: Criteria #13- Openly licensed and free of charge**

All platforms used to develop the application are openly licensed and free of charge. For our evidence see the [README.md](https://github.com/eGT-Labs/egt-gsa-proto/blob/master/README.md).