Cloud Applications end report

System

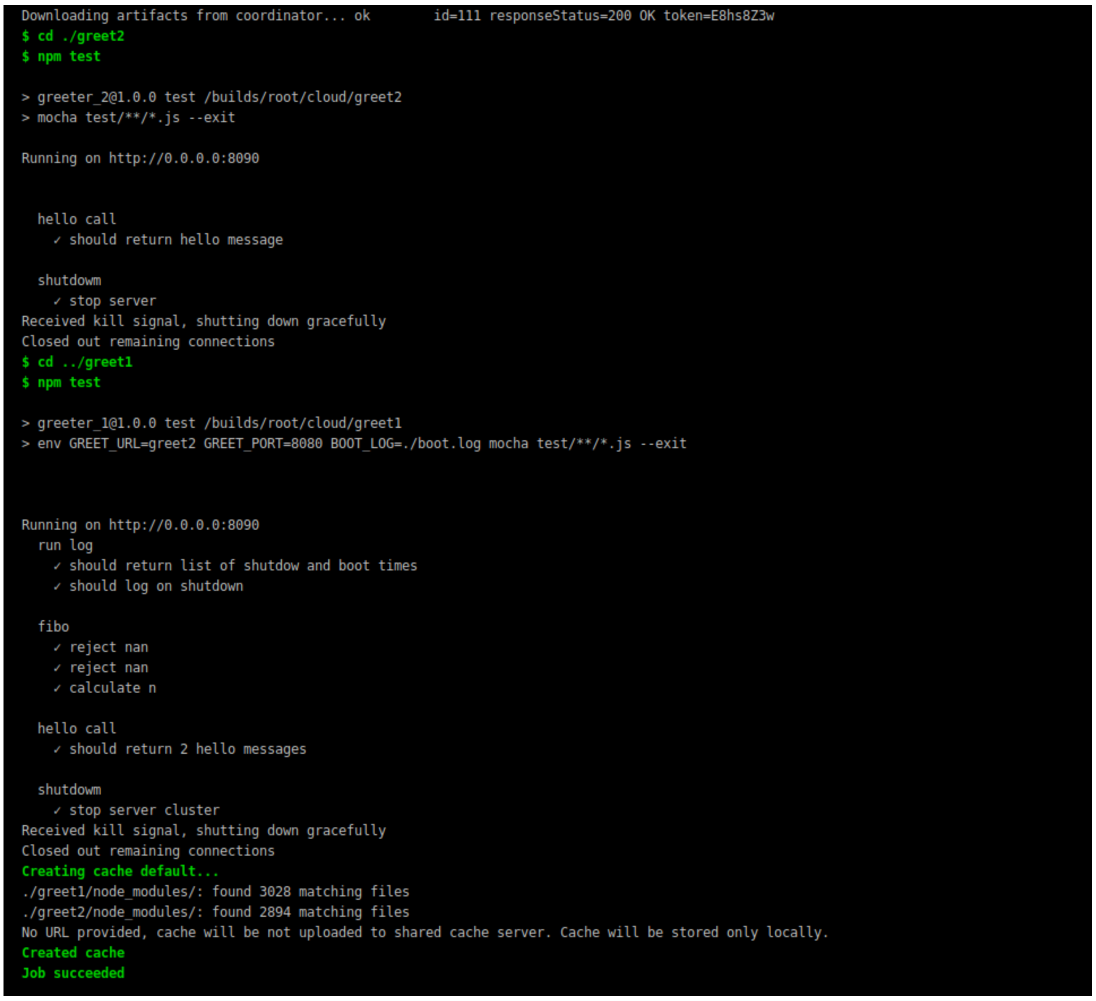
System consist of two different servers. Each server is individual nodejs express server with all of the development related features (dependencies, server application, tests, build script). In greet1's root folder there is packacge.json which describes available commands and dependencies for greet1. Actual functional server code is inside app.js which holds required functionality. App.js contains 3 separate tasks server infrastructure, application routes which implements required features, and server lifecycle events which are mainly used for application state logging. In test folder there is tests for every feature separated into different files.

Greet2 follows same structure as greet1 only with less features.

Root folder contains instruction to run the system (readme.md). Docker compose file for local testing and gitlab-ci.yml for automated building and testing.

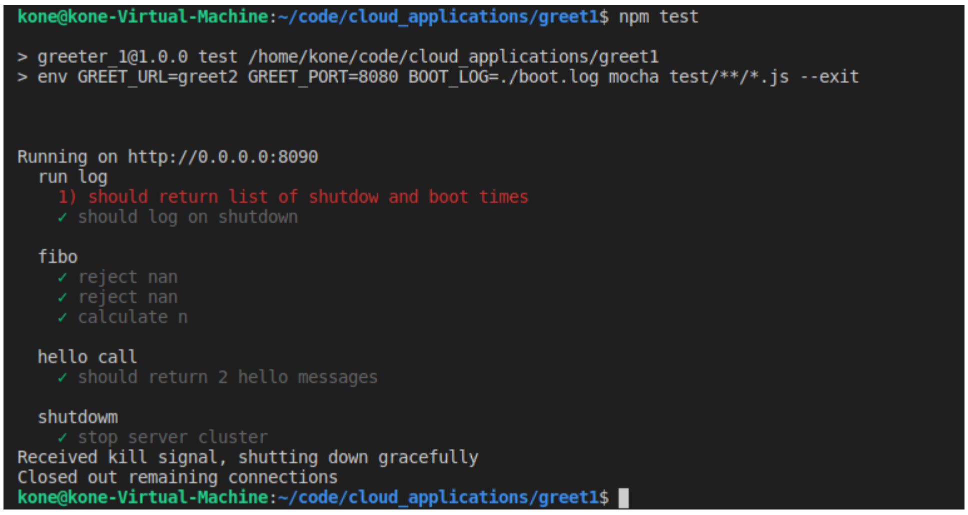
Test Results

Successful test run:



```
Downloading artifacts from coordinator... ok        id=111 responseStatus=200 OK token=E8hs8Z3w
$ cd ./greet2
$ npm test

> greeter_2@1.0.0 test /builds/root/cloud/greet2
> mocha test/**/*.js --exit

Running on http://0.0.0.0:8090


  hello call
    ✓ should return hello message

  shutdown
    ✓ stop server
Received kill signal, shutting down gracefully
Closed out remaining connections
$ cd ../greet1
$ npm test

> greeter_1@1.0.0 test /builds/root/cloud/greet1
> env GREET_URL=greet2 GREET_PORT=8080 BOOT_LOG=./boot.log mocha test/**/*.js --exit



Running on http://0.0.0.0:8090
  run log
    ✓ should return list of shutdow and boot times
    ✓ should log on shutdown

  fibo
    ✓ reject nan
    ✓ reject nan
    ✓ calculate n

  hello call
    ✓ should return 2 hello messages

  shutdown
    ✓ stop server cluster
Received kill signal, shutting down gracefully
Closed out remaining connections
Creating cache default...
./greet1/node_modules/: found 3028 matching files
./greet2/node_modules/: found 2894 matching files
No URL provided, cache will be not uploaded to shared cache server. Cache will be stored only locally.
Created cache
Job succeeded
```

Test failure when no run-log end point exists:



```
kone@kone-Virtual-Machine:~/code/cloud_applications/greet1$ npm test

> greeter_1@1.0.0 test /home/kone/code/cloud_applications/greet1
> env GREET_URL=greet2 GREET_PORT=8080 BOOT_LOG=./boot.log mocha test/**/*.js --exit



Running on http://0.0.0.0:8090
  run log
    1) should return list of shutdow and boot times
    ✓ should log on shutdown

  fibo
    ✓ reject nan
    ✓ reject nan
    ✓ calculate n

  hello call
    ✓ should return 2 hello messages

  shutdowm
    ✓ stop server cluster
Received kill signal, shutting down gracefully
Closed out remaining connections
kone@kone-Virtual-Machine:~/code/cloud_applications/greet1$ 
```

Problems:

Most of problems and hardship was related to configuration of the ci. Setup of ci went smoothly, but first problem was that ci skipped reporting failed test. Second problem was that when using node as base image, there is no docker tools available. Finding solution for that took most of effort. Final solution was to use different image for dist stage. This introduced bug with ci. Ci might fail in test stage with error missing command npm because image switching failed.

Second problem that took time was setting node to handle all shutdown stages correctly so that every shutdown gets reported.

Conclusion:

I haven't set up ci pipeline previously. Most of problems and learning were related to this. Something that I would do differently if started from scratch would be ci pipeline. Docker images should have been built at first stage and testing should be run inside those containers. That would have spared some time and would fix that image conflict.