Continuous Integration for Spark Apps

...

uncharted

Continuous Integration for Spark Apps

...

because testing things is pretty important.

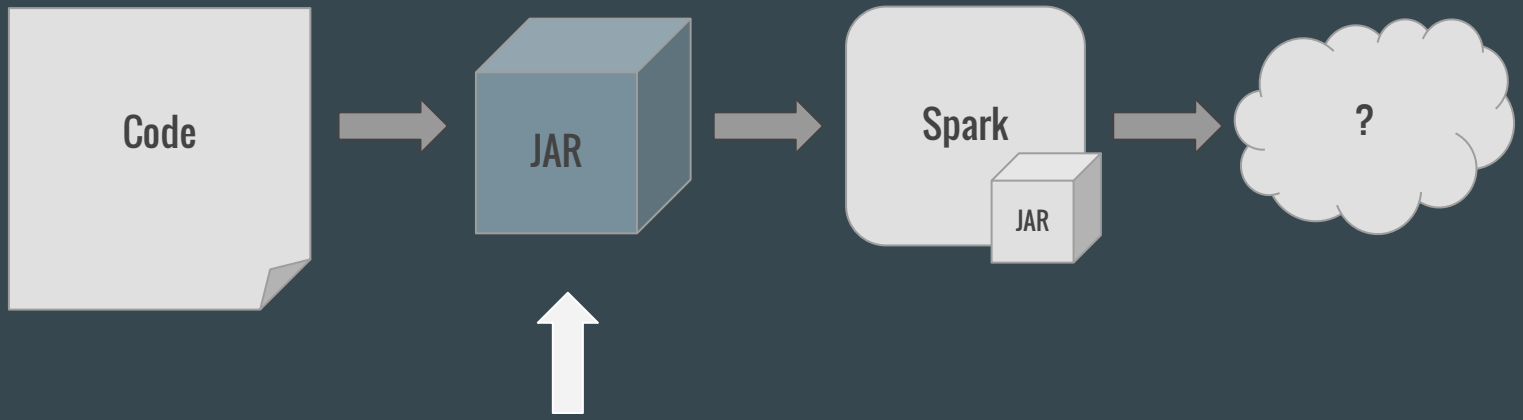
uncharted

Hi, I'm Sean!

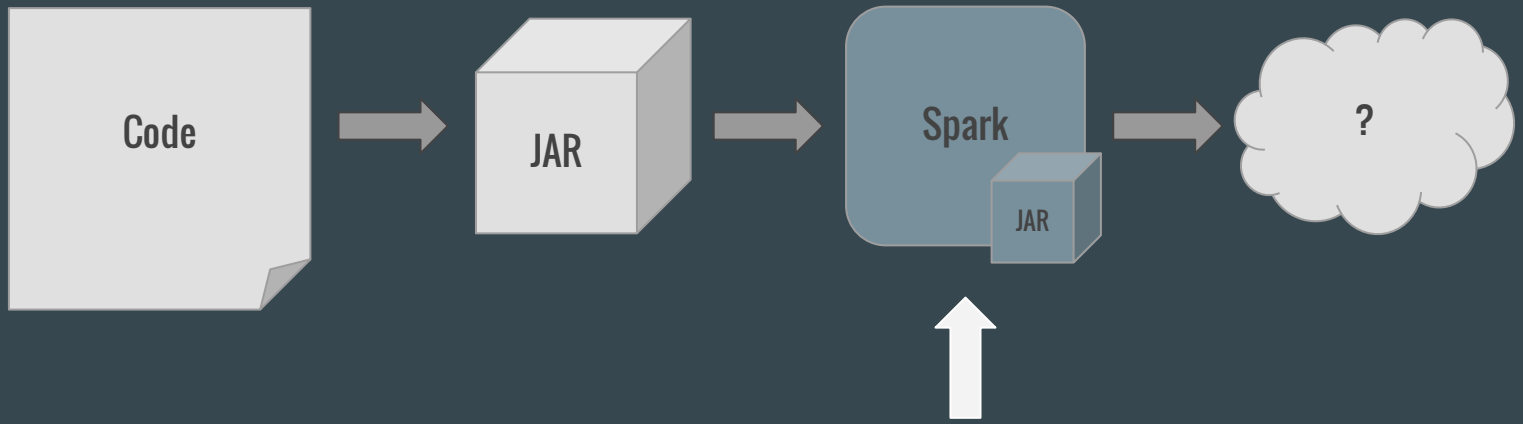
I'm writing
Spark libraries at
Uncharted Software

Why is it hard to test Spark apps?

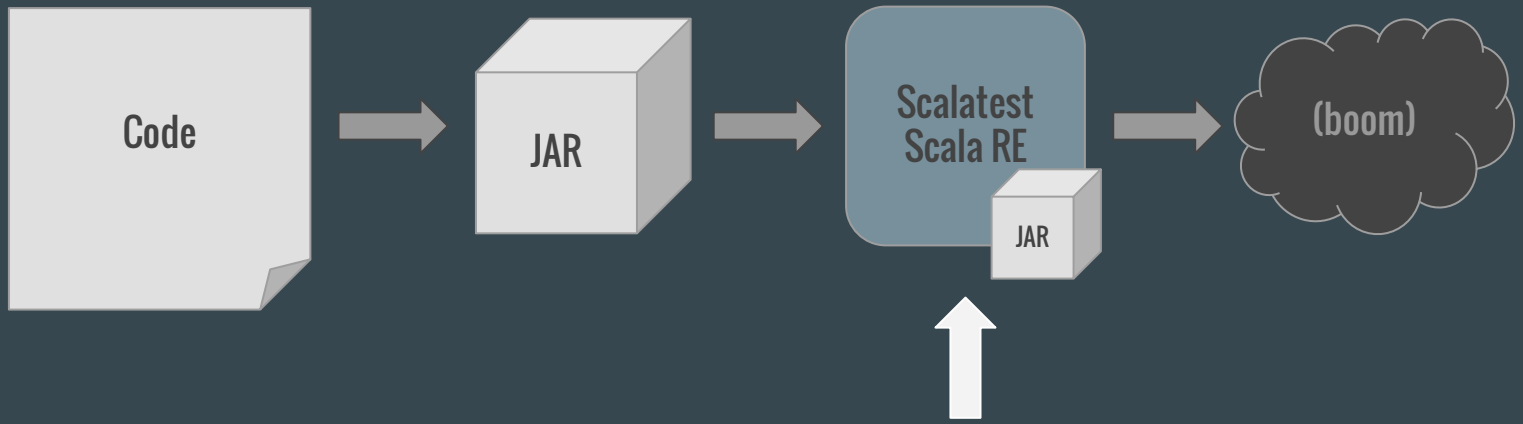
What is a Spark app?



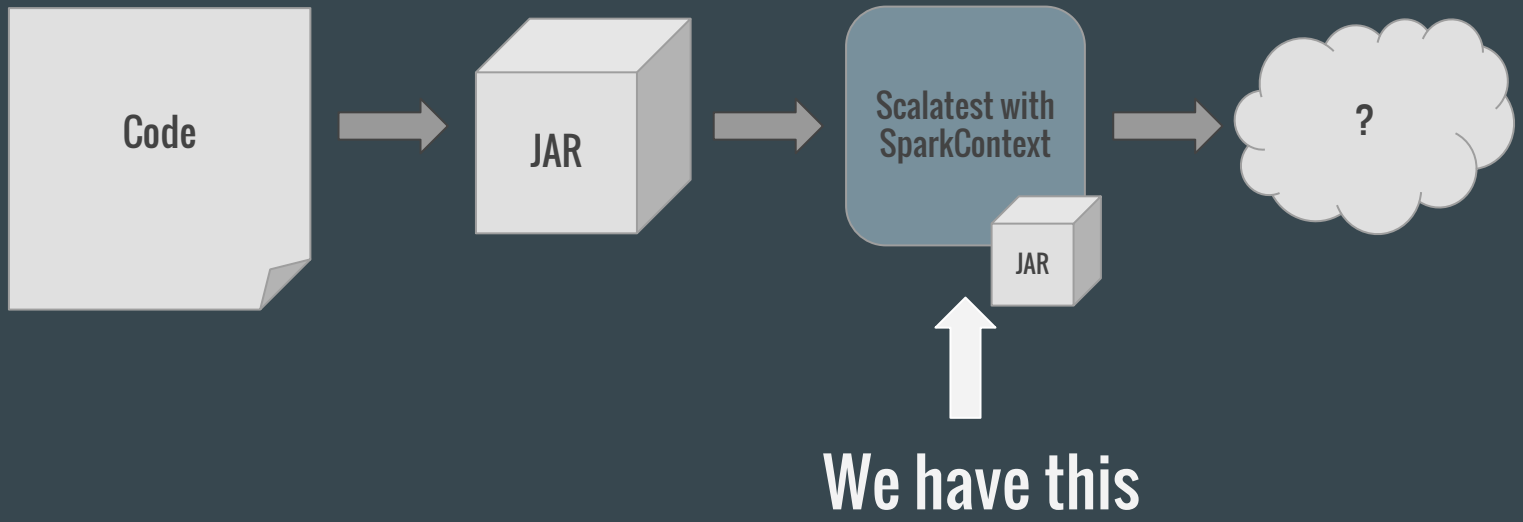
And...



But...

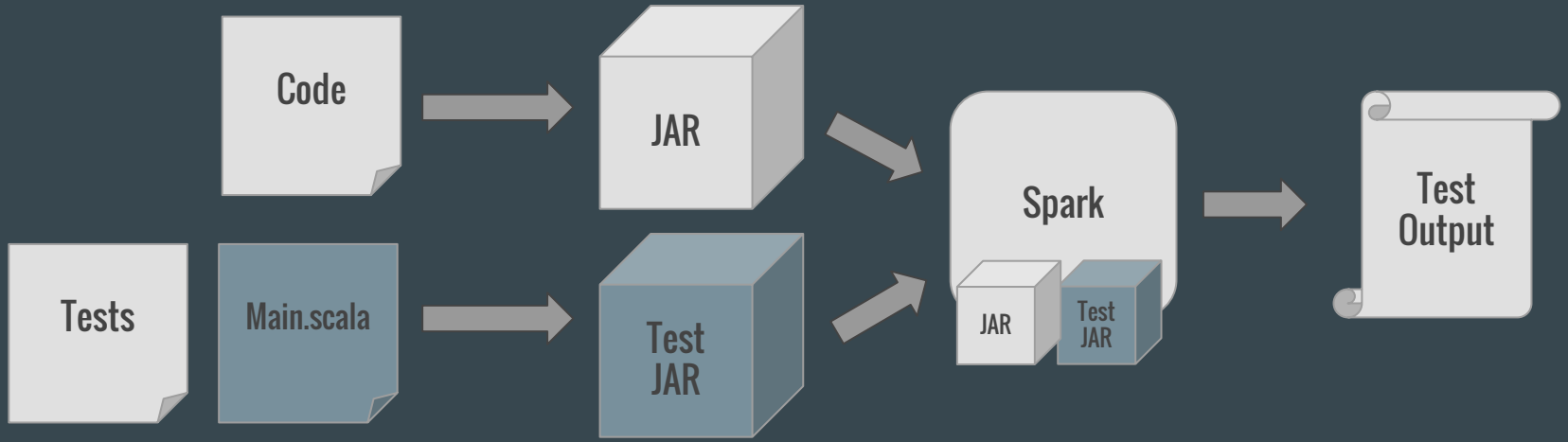


Or at best...



Can we test a Spark app?

Step 1: Squish Scalatest into Spark

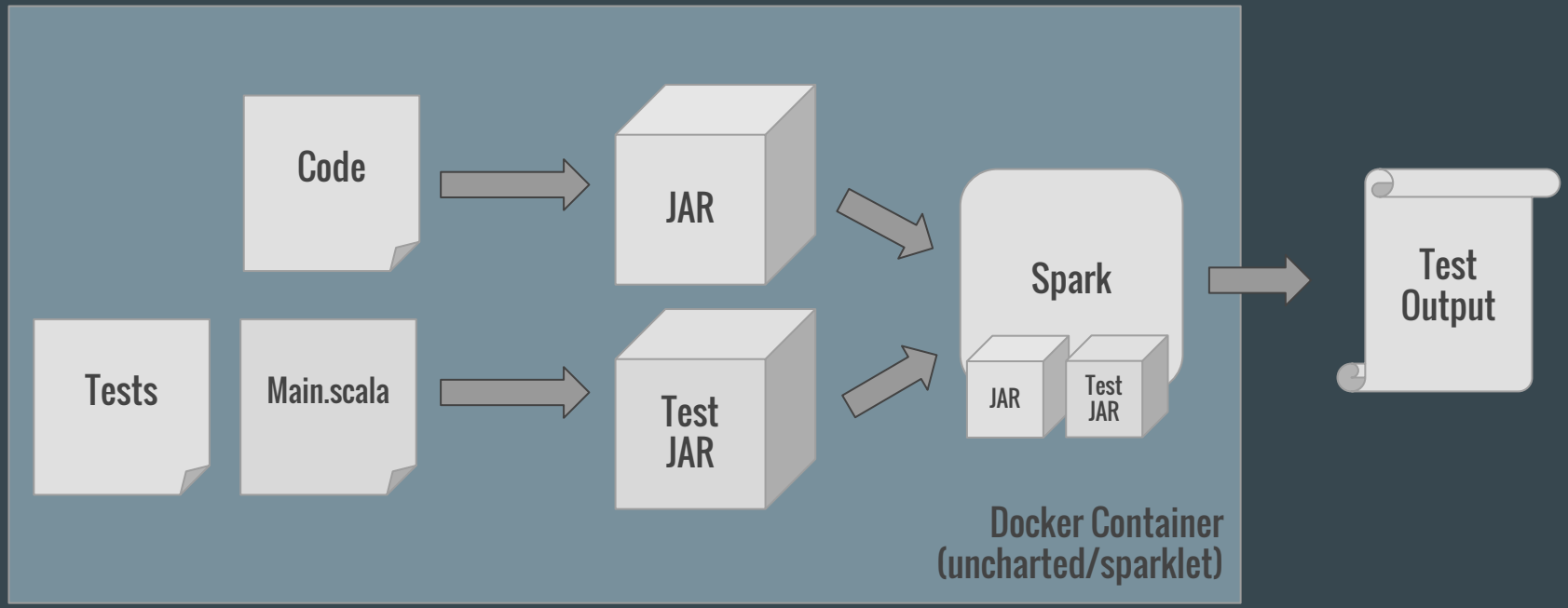


Main.scala

```scala
import org.scalatest.tools.Runner

object Main {
  def main(args: Array[String]): Unit = {
    val testResult = Runner.run(Array("-o", "-R", "build/classes/test"))
    if (!testResult) {
      System.exit(1) // exit with an error code if a test failed
    }
  }
}
```

Continuous Integration?

Step 2: Squish Spark and Test JAR into Docker



test-environment.sh

```
#!/bin/sh

docker run \
-e GRADLE_OPTS="-Dorg.gradle.daemon=true" \
-v $(pwd)/src/test/resources/log4j.properties:/usr/local/spark/conf/log4j.properties \
-v $(pwd):/opt/mycode \
-it \
--workdir="/opt/mycode" \
uncharted/sparklet:1.5.2 bash
```

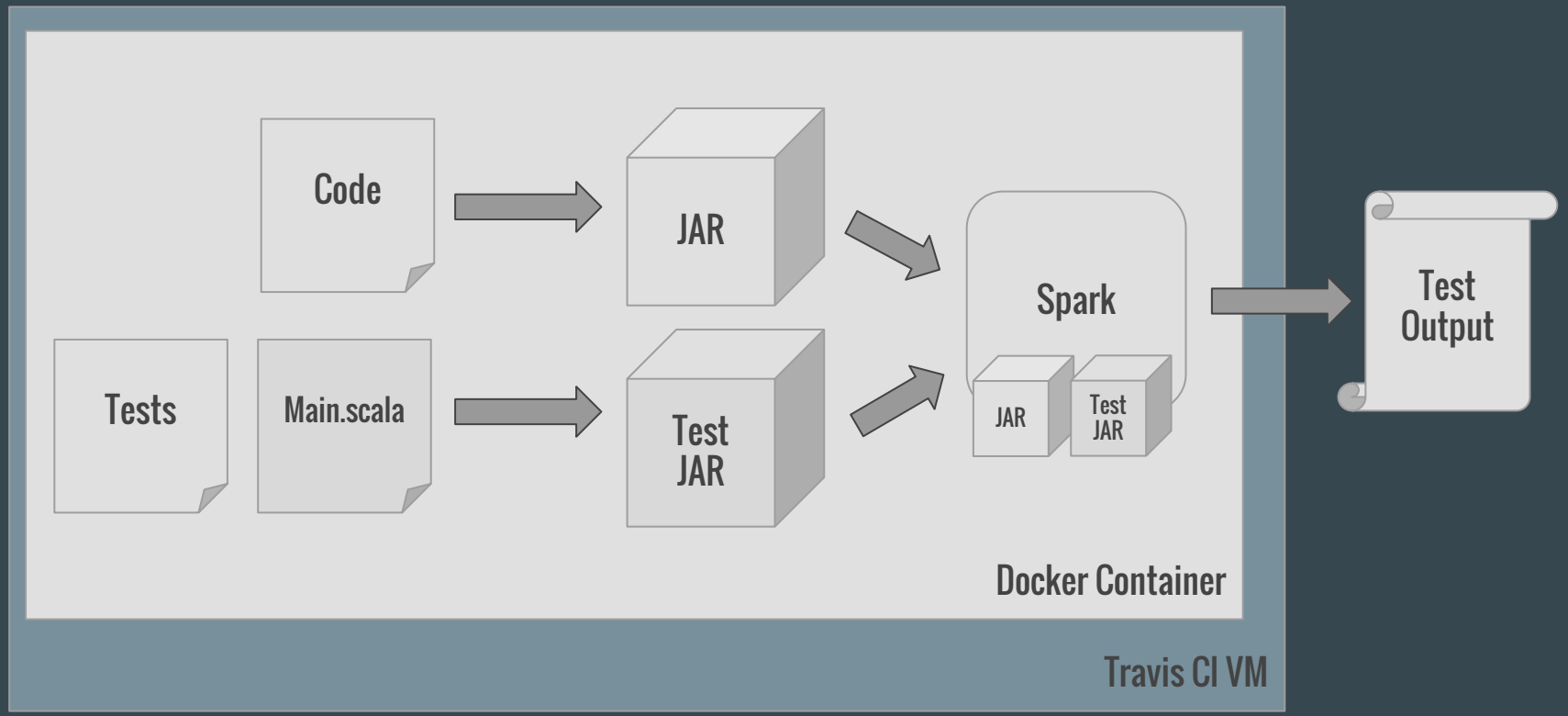
test-environment.sh

```
#!/bin/sh

docker run \
# the gradle daemon speeds things up
-e GRADLE_OPTS="-Dorg.gradle.daemon=true" \
# this log4j.properties file silences some of the verbose output from spark
-v $(pwd)/src/test/resources/log4j.properties:/usr/local/spark/conf/log4j.properties \
# mount our code as a volume within the container
-v $(pwd):/opt/mycode \
# interactive mode
-it \
# set working directory where we'll have the container prompt start
--workdir="/opt/mycode" \
uncharted/sparklet:1.5.2 bash
```

build.gradle

```
// create an Exec task to test code via spark-submit
task test(overwrite: true, type: Exec, dependsOn: [jar, testJar]) {
  executable = 'spark-submit'
  args = [
     //the --packages flag allows us to place our dependencies on the path
    "--packages","org.scalatest:scalatest_${scalaBinaryVersion}:2.2.5",
    //the --jars flag allows us to place our code on the path
    "--jars","/opt/${repoName}/build/libs/${artifactName}-${version}.jar",
     //Main, from before
    "--class","my.company.project.Main",
    "build/libs/${artifactName}-${version}-tests.jar"
  ]
}
```

Step 3: Squish Docker container into Travis CI



.travis.yml

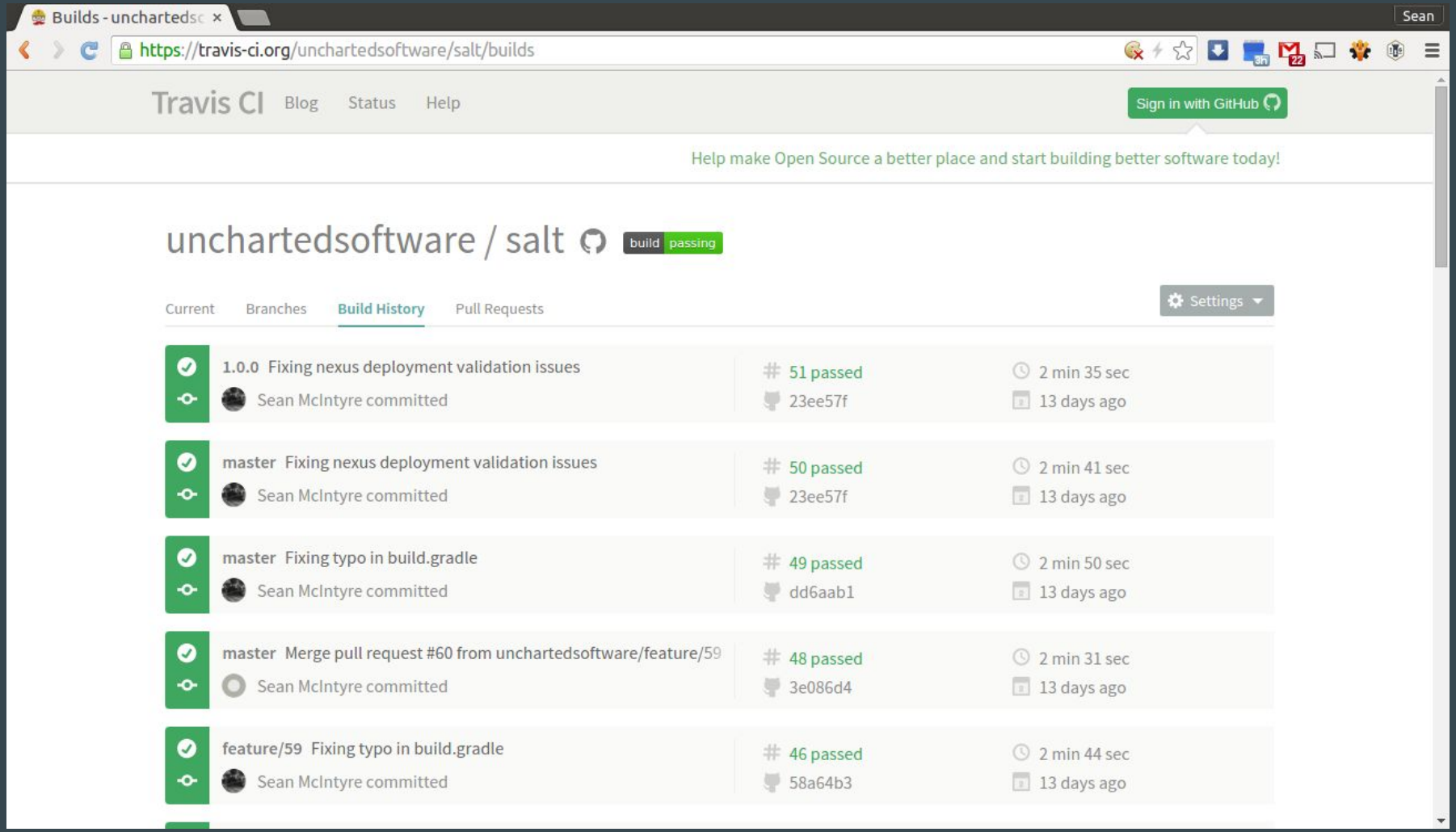
```
sudo: required
language: bash
services:
  - docker

before_script:
  # generate environment file so we can give the travis environment to our test container
  - env | grep TRAVIS_ > travis.env
  - echo "GRADLE_OPTS=-Dorg.gradle.daemon=false" >> travis.env
  - echo "CI_NAME=travis_ci" >> travis.env
  - echo "CI=true" >> travis.env
  - echo "TRAVIS=true" >> travis.env
  - echo "CONTINUOUS_INTEGRATION=true" >> travis.env
  - echo "DEBIAN_FRONTEND=noninteractive" >> travis.env
  - echo "HAS_JOSH_K_SEAL_OF_APPROVAL" >> travis.env

script:
  # run test container
  - docker run --env-file travis.env -v $(pwd)/src/test/resources/log4j.properties:/usr/local/spark/conf/log4j.properties -v $(pwd):
/opt/salt --rm --entrypoint=/startup.sh --workdir="/opt/salt" uncharted/sparklet:1.5.1 ./gradlew coverage coveralls
```

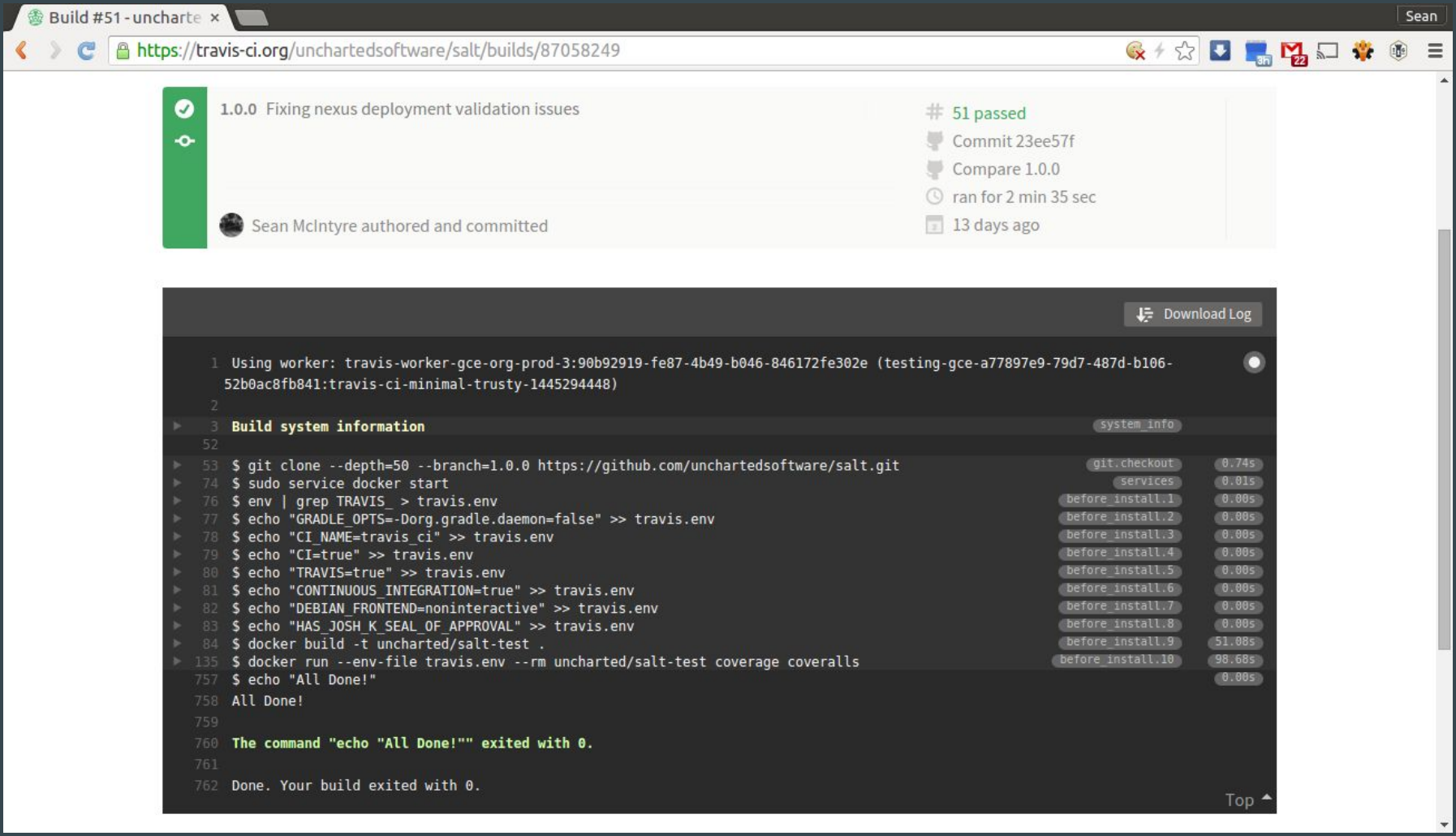
Voila!

Examples

Check out Salt (https://github.com/unchartedsoftware/salt) for a working example.

Salt's build is also connected to Coveralls for code coverage reports :)


Builds - unchartedsc
Sean
https://travis-ci.org/unchartedsoftware/salt/builds
Travis CI
Blog
Status
Help
Sign in with GitHub
Help make Open Source a better place and start building better software today!
unchartedsoftware / salt
build passing
Current
Branches
Build History
Pull Requests
Settings
1.0.0 Fixing nexus deployment validation issues
Sean McIntyre committed
51 passed
23ee57f
2 min 35 sec
13 days ago
master Fixing nexus deployment validation issues
Sean McIntyre committed
50 passed
23ee57f
2 min 41 sec
13 days ago
master Fixing typo in build.gradle
Sean McIntyre committed
49 passed
dd6aab1
2 min 50 sec
13 days ago
master Merge pull request #60 from unchartedsoftware/feature/59
Sean McIntyre committed
48 passed
3e086d4
2 min 31 sec
13 days ago
feature/59 Fixing typo in build.gradle
Sean McIntyre committed
46 passed
58a64b3
2 min 44 sec
13 days ago


Build #51 - uncharte
Sean
https://travis-ci.org/unchartedsoftware/salt/builds/87058249
1.0.0 Fixing nexus deployment validation issues
Sean McIntyre authored and committed
51 passed
Commit 23ee57f
Compare 1.0.0
ran for 2 min 35 sec
13 days ago
Download Log
1 Using worker: travis-worker-gce-org-prod-3:90b92919-fe87-4b49-b046-846172fe302e (testing-gce-a77897e9-79d7-487d-b106-52b0ac8fb841:travis-ci-minimal-trusty-1445294448)
2
3 Build system information system_info
52
53 $ git clone --depth=50 --branch=1.0.0 https://github.com/unchartedsoftware/salt.git git.checkout 0.74s
74 $ sudo service docker start services 0.01s
76 $ env | grep TRAVIS_ > travis.env before_install.1 0.00s
77 $ echo "GRADLE_OPTS=-Dorg.gradle.daemon=false" >> travis.env before_install.2 0.00s
78 $ echo "CI_NAME=travis_ci" >> travis.env before_install.3 0.00s
79 $ echo "CI=true" >> travis.env before_install.4 0.00s
80 $ echo "TRAVIS=true" >> travis.env before_install.5 0.00s
81 $ echo "CONTINUOUS_INTEGRATION=true" >> travis.env before_install.6 0.00s
82 $ echo "DEBIAN_FRONTEND=noninteractive" >> travis.env before_install.7 0.00s
83 $ echo "HAS_JOSH_K_SEAL_OF_APPROVAL" >> travis.env before_install.8 0.00s
84 $ docker build -t uncharted/salt-test . before_install.9 51.08s
135 $ docker run --env-file travis.env --rm uncharted/salt-test coverage coveralls before_install.10 98.68s
757 $ echo "All Done!" 0.00s
758 All Done!
759
760 The command "echo "All Done!"" exited with 0.
761
762 Done. Your build exited with 0.
Top

Coming soon...

- Non-linear Spark Pipeline
- Sparkplug

http://blog.uncharted.software

Questions?