

# TEST FRAMEWORK CHALLENGE

UI testing with Docker/Selenoid

# GOAL

- To set up a testing system consisting of Selenium, GGR and a trivial test case (let's say: open browser, go to google.com). These are the relevant projects, containing also links to documentation:  
<https://github.com/aerokube/selenium> <https://github.com/aerokube/ggr> <https://github.com/aerokube/cm>.
- What we expect from you would be e.g. a collection of a docker compose file with config files for the various components and the test case, ideally in e.g. a git repo.

# WHERE DO I STAND?

- What do I know?
  - Docker
  - Selenium
  - Load Balancing
  - Configuration Manager
  - Git

- What I do not?
  - Docker compose
  - Selenoid
  - GGR Load balancer
  - CM

# APPROACH- LEARNING BY DOING

- Installing Docker on Mac
- Registering on Docker site
- Creating New repository
- Logging to GitHub, creating new repository
- Connecting Docker repo with GitHub repo
- Downloading the solenoid docker image

# SELENOID WITH CM

- Why Selenoid?
  - <https://hackernoon.com/selenium-testing-a-new-hope-7fa87a501ee9>
  - <https://hackernoon.com/selenium-testing-a-new-hope-a00649cdb100>
- Hello World Selenoid!
  - `curl -Lo cm https://github.com/aerokube/cm/releases/download/1.2.0/cm\_darwin\_amd64`
  - `./cm selenoid start —vnc`
- Created a sample test, ran it on Chrome successfully. No luck with Firefox :(
  - <https://github.com/nightwatchjs/nightwatch/issues/1628>
- What's wrong with <http://localhost:4444/wd/hub/> while <http://localhost:8080> Working fine!
  - Selenoid was created to be run in big Selenium clusters and thus has no built-in user interface. This is why trying to open an URL above in browser returns **404** and this is the expected behavior :)

# DOCKER-COMPOSE

- What is it and Why do we need it?
  - <https://github.com/prvenkat22/automation.git> - [docker-compose.yaml](#)

Use case:

- Can be used to bring the test setup with required pre-requisites and states

# DOCKER GGR

- Failed to start docker on default path, solution-
  - `docker run -d --name ggr -v /tmp/grid-router/:/tmp/grid-router --net host aerokube/ggr:latest-release`

volumes:

- `"/tmp/selenoid:/etc/selenoid/"` # assumed current dir contains browsers.json

Issues faced and resolved:

- <https://github.com/aerokube/cm/issues/50>
- <https://github.com/aerokube/selenoid/issues/155>

# CM WAY OF SELENOID

- Configure and Download selenium
  - `./cm selenium configure [--browsers firefox,chrome] [--last-versions 2] [--tmpfs 128]`
- Start selenium
  - `./cm selenium start --vnc`
- Download and start selenium-ui
  - `./cm selenium-ui start`
- Run tests pointing to <http://localhost:4444>
  - `./node_modules/nightwatch/bin/nightwatch ./src/test/login.js ./src/test/login.l.js -e chrome`



# SELENOID

- Configure and run selenoid

```
docker run -d \
--name selenoid \
-p 4444:4444 \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /Users/prvenkat/nw/./etc/selenoid:/ro \
-v /Users/prvenkat/nw/video/./opt/selenoid/video/ \
-e OVERRIDE_VIDEO_OUTPUT_DIR=/Users/prvenkat/nw/video/ \
-e DOCKER_API_VERSION=1.35 \
aerokube/selenoid:latest-release
```

- Configure and start selenoid-ui

- `docker run -d --name selenoid-ui -p 8080:8080 aerokube/selenoid-ui --selenoid-uri http://172.17.0.1:4444`

# CONCURRENT TEST EXECUTION - CONCURRENTLY

Run multiple commands concurrently!

- `concurrently "./node_modules/nightwatch/bin/nightwatch ./src/test/login.js --conf "nightwatch.json" -e "chrome" " ./node_modules/nightwatch/bin/nightwatch ./src/test/login.js --conf "nightwatch.1.json" -e "chrome" " ./node_modules/nightwatch/bin/nightwatch ./src/test/login.js --conf "nightwatch.2.json" -e "chrome"`

Use case

- Multiple browsers at once - Chrome 64, Chrome 63, etc.,
- Can be run in multiple configurations - geckodriver-0.191, geckodriver-0.190
- Can be used to simulate concurrent test scenarios

# LEARNINGS/FINDINGS

- create container: Error response from daemon: client version 1.36 is too new. Maximum supported API version is 1.35
  - export DOCKER\_API\_VERSION=1.35
- Debugging through VNC session
- Capturing Videos and hosting it on the host machine
- Capturing Screenshot and asserting using nightwatch-vrt
- Failure to run tests on Firefox - Issue with NightwatchJS and Selenium Grid version 3.8+ - Stackoverflow post
- Docker - Great hands-on, pushed my first image to the hub! NightwatchJS v0.9.19 with nodejs

# LESSONS LEARNT

- Assume that you will be blocked in every turn!
- Expect too much of NOISE along with required information
- Document everything and give IT back!
- Learn to debug docker environment

# FUTURE PLAN

- Integrate with Jenkins
- Integrate with log stash/kibana
- Exploring load balancing using ggr