Building and scaling an Android emulator device farm

Prabhagaran D K
Martin Schneider

About us

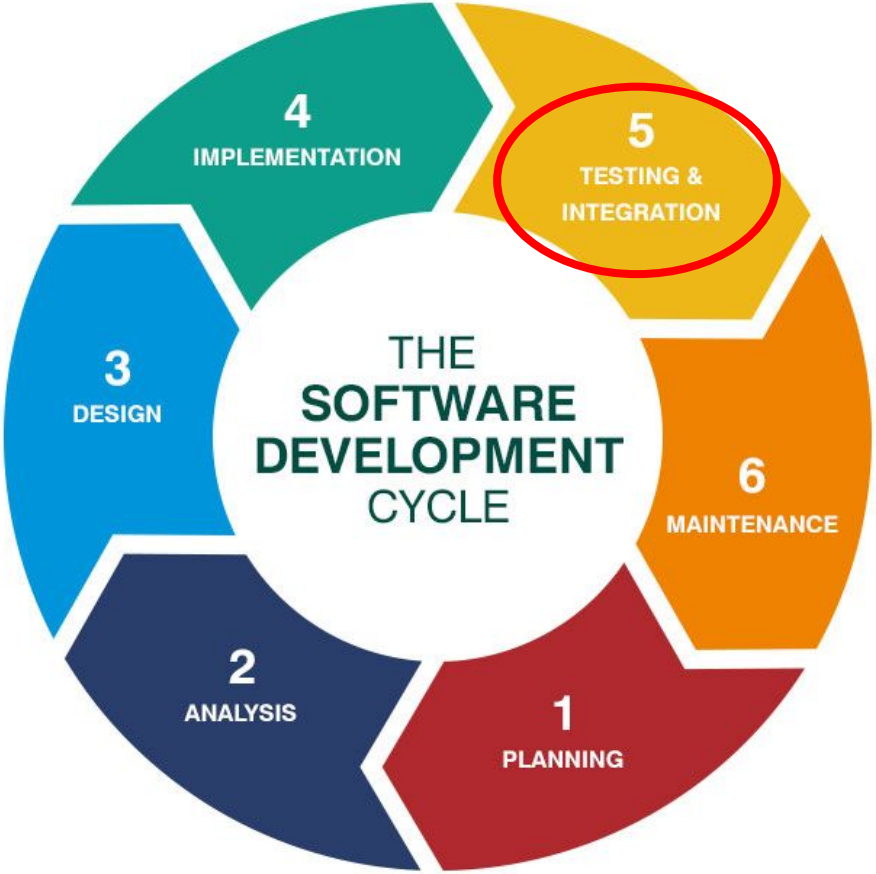
Prabhagaran DK
Software Engineer (Test) @ Carousell
prabagharan@gmail.com

Martin Schneider
Senior Software Engineer @ Carousell
mart.schneider@gmail.com

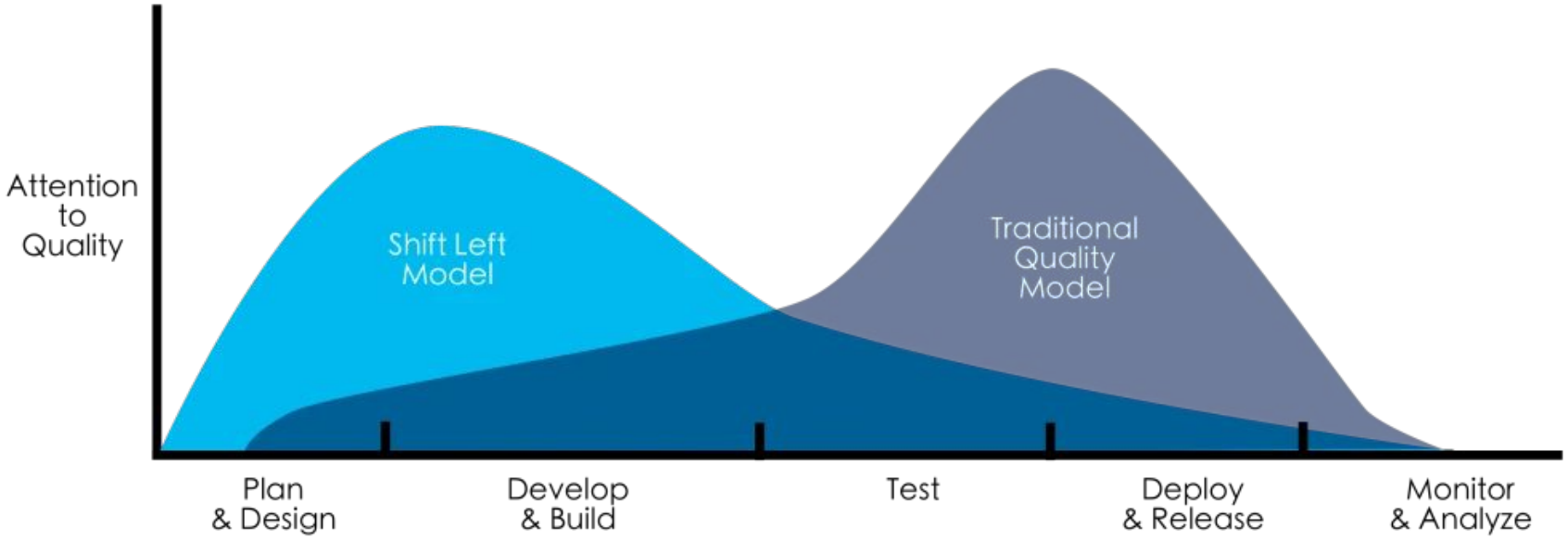
Quality statement

“After development	“Throughout the development cycle
QA	everyone
tests the build	does their part
to ensure the quality of	to ensure the quality of
the release.”	our processes and products.”

When do we need to care about quality?

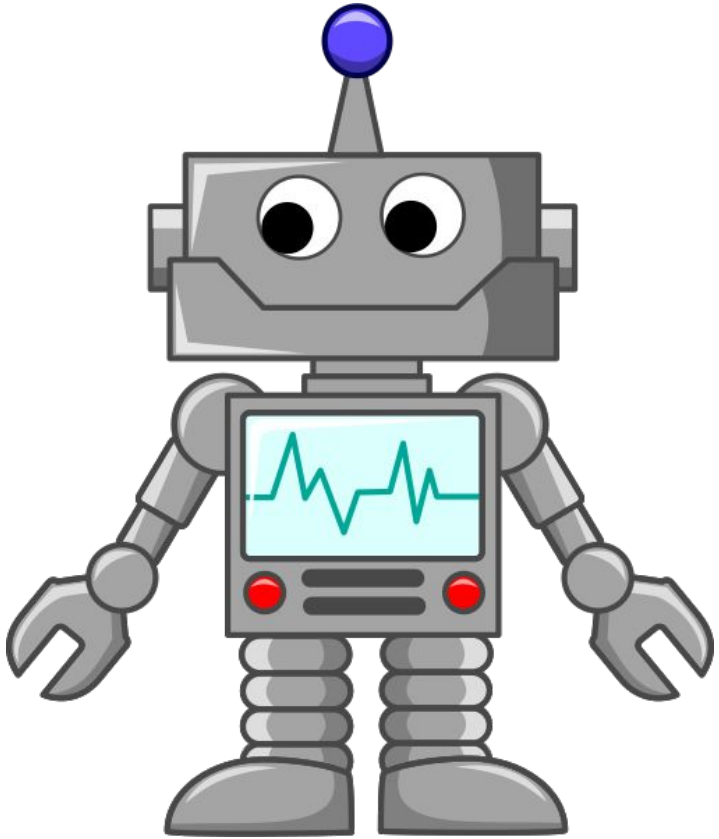


Shift Left



The earlier a problem is found the easier and cheaper it is to fix it!

Automation - a key ingredient



Automation - a key ingredient

- Manual regression is a waste of precious engineering resources (exploratory testing is a different story)

- We currently have ~150 automated test scenarios for our mobile apps.



- Techstack: Cucumber + Appium + Java
- Re-use most of our scenarios and step definitions across multiple platforms.
- Re-use a majority of our page objects across multiple platforms.
- Maintain one framework for 3 platforms (Android, iOS, Web).
- We are considering native automation (XCUITest, Espresso) for specific use-cases in the future.

Humble beginnings

We automated our
UI regression tests!

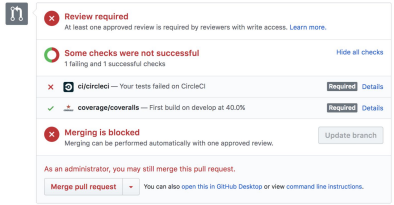
How often do they run?

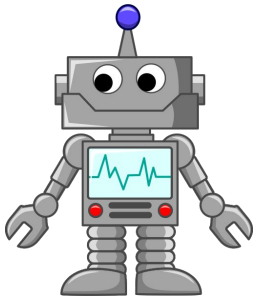
On release day...

BHAG: Run functional UI tests on every code change

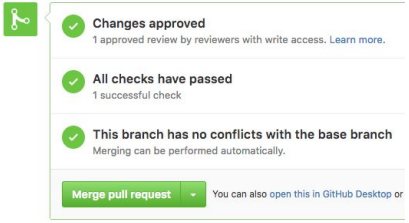
Feedback in < 20 minutes

build the app package





appium



Parameters

- Our target platforms are Android, iOS and Web.
- We want to keep our existing test scenarios.
- **We want to keep our test framework platform independent.**

→ rules out native automation (Espresso, XCUITest)

Challenges

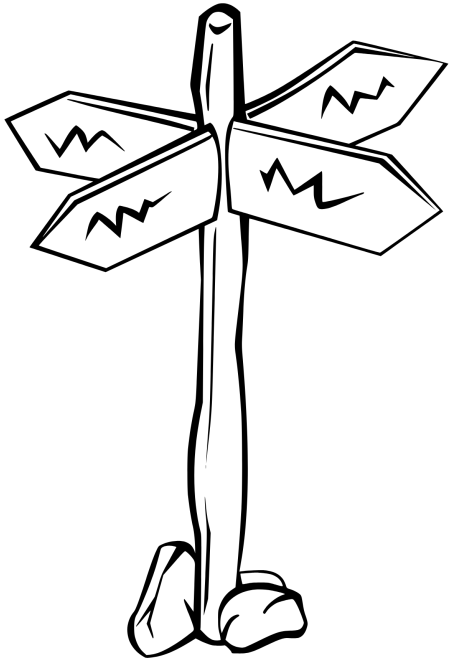
- **Fluctuating load**
  - ~10 pull requests per day per platform
  - spikes before code-freeze
  - usually no tests on the weekend

- **20 minutes is ambitious**
  - on iOS, building the app package already takes ~10-15 minutes
  - heavy parallelization needed
  - waiting for a device is not acceptable

→ **scaling becomes important**

Quo vadis?

Where to go from here?



How can we scale?

AWS

SAUCELABS

Perfecto

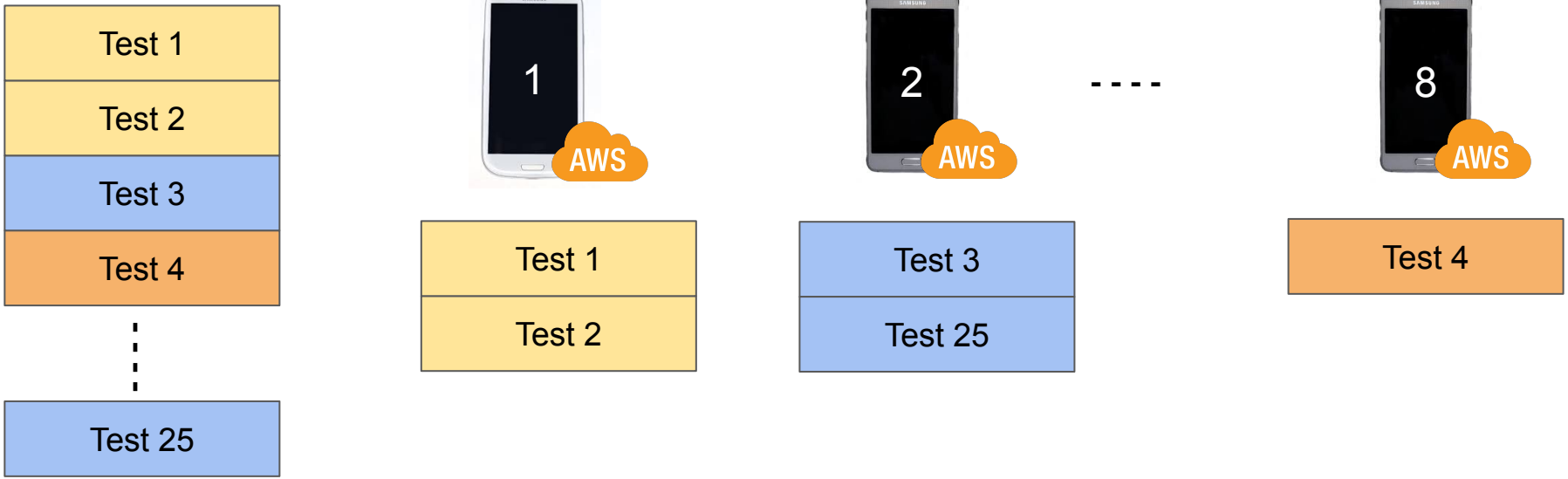
pCloudy.com

On premise vs. cloud

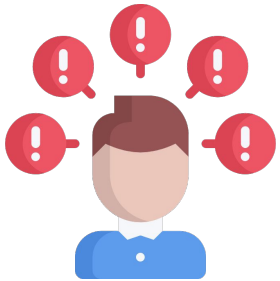
• No maintenance required
• Fixed costs
• Support
	• Full control
	  ○ Number of devices
	  ○ Types of devices
	  ○ Software libraries
	  ○ …

• Less flexibility
  ○ Available devices
  ○ Software update cycle? (Appium, Java)
• Expensive at scale
	• Maintenance
	  ○ replace devices
	  ○ fix issues
	  ○ infrastructure
	  ○ ...

Devicelab v0.1 - AWS Devicefarm



- AWS Devicefarm requires packaging and uploading all test code and libraries (server-side execution) → overhead
- Setting-up and tearing-down devices takes a lot of time
- No transparency during the run
- Device availability issues



Virtual vs. physical device

	Virtual device	Physical device
App basic functionality and UI layout	✔	✔
Application load speed	✘	✔
Test interrupts, battery consumption, CPU & memory utilization	✘	✔
Colors and resolutions which render differently on different screen densities	✘	✔
UI performance (transitions and orientation)	✘	✔
Hardware functionality (bluetooth, camera, gps and more)	✘	✔
App performance across device configurations (chipsets, memory etc.)	✘	✔
100% real world accuracy	✘	✔

modified from www.browserstack.com

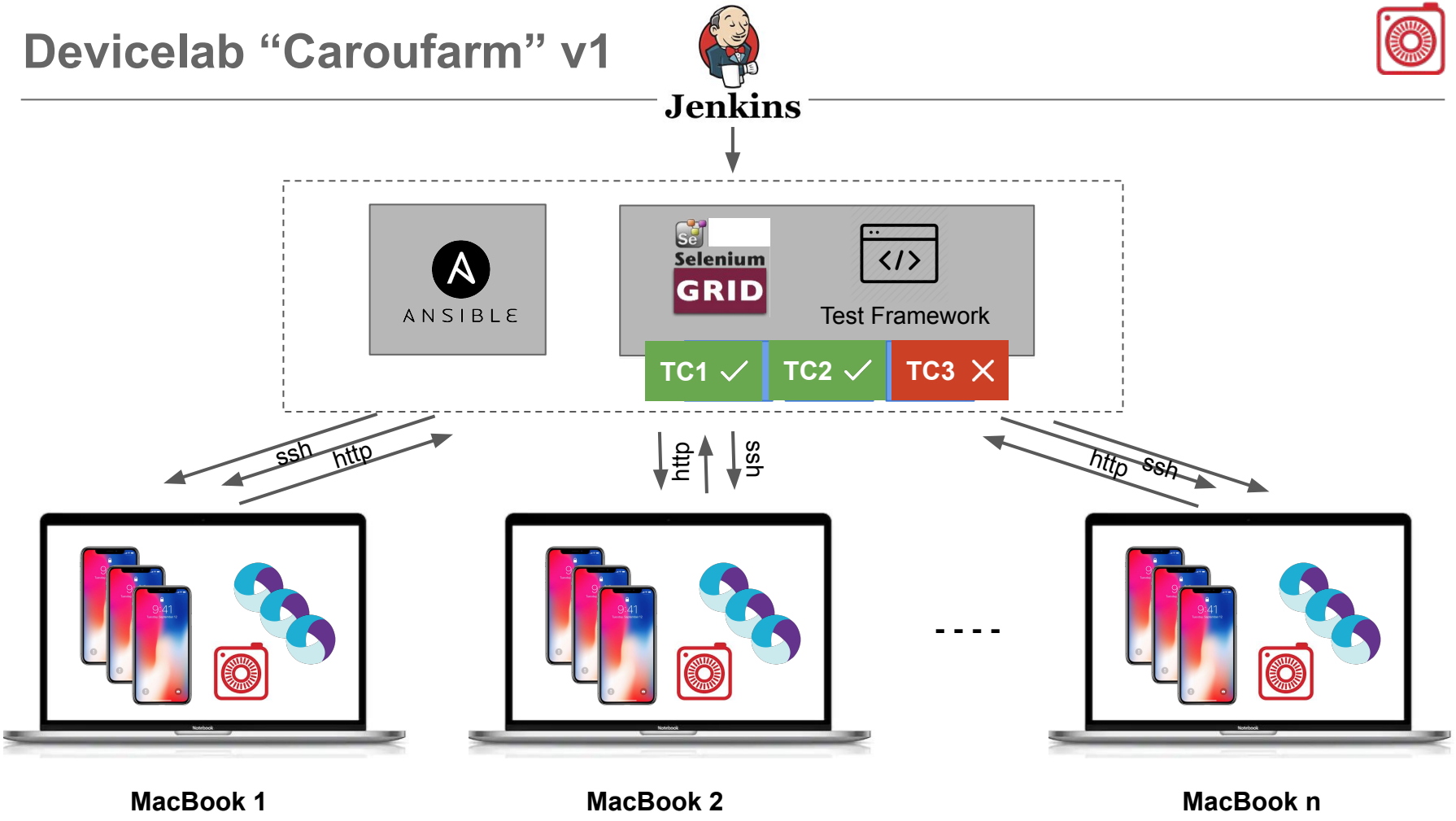
So many options

Physical 	Expensive	Messy + expensive
Virtual 	-	🤔

Caroufarm v1

- Macbooks - reused old macbooks to run the simulators.
- Ansible - to ssh multiple machines and configure the hosts.
- Selenium Grid - to distribute the tests to different simulators.
- Python & shell scripts.
- Existing test framework (Cucumber + Appium + Java + Spring + Maven)

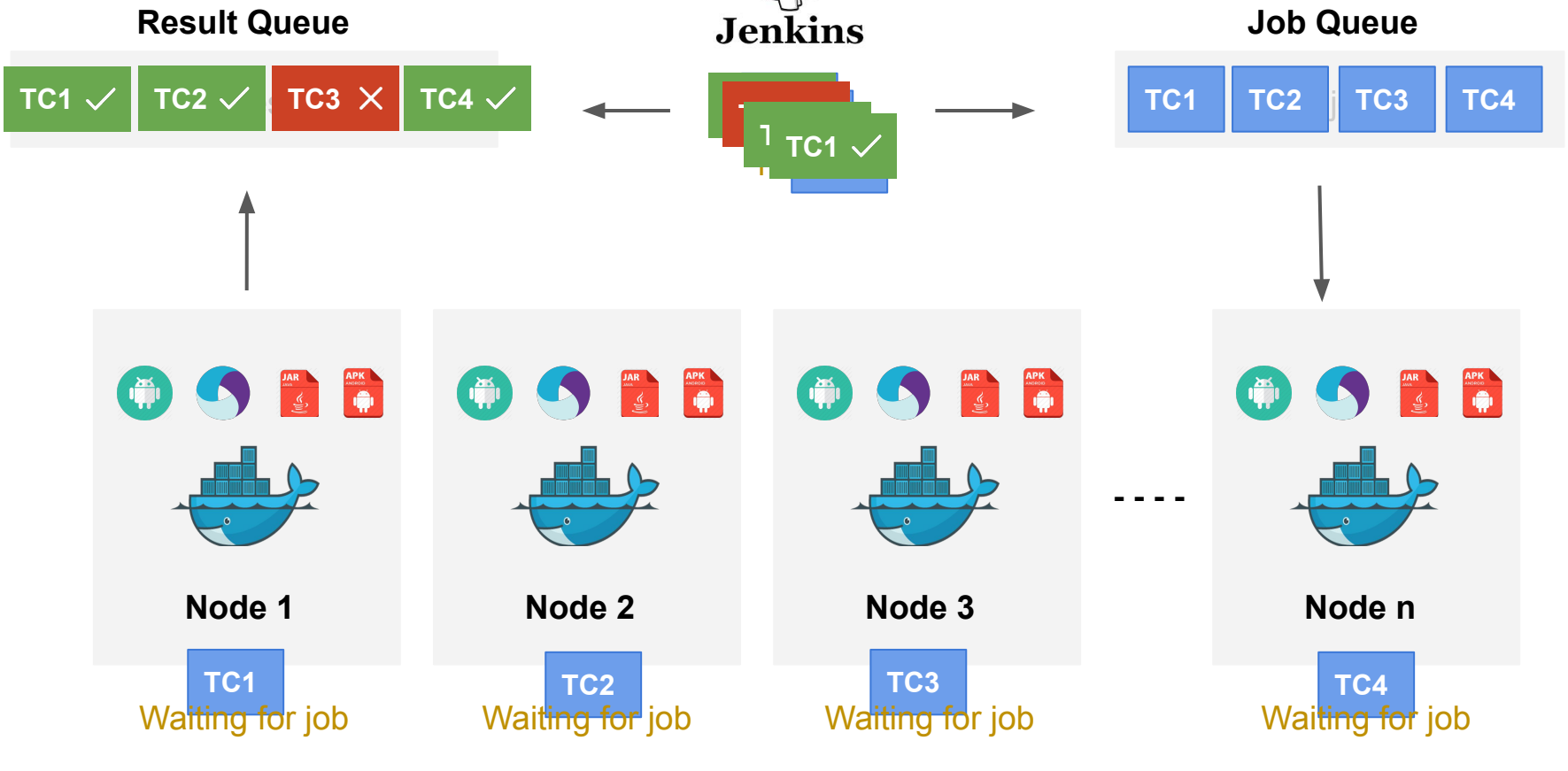
Devicelab “Caroufarm” v1



Learnings from Caroufarm V1

- Distribution of tests to simulators was not efficient.
- Scaling up / down was not easy enough.
- Resetting Caroufarm.
- Maintenance of Macbooks
  - Wifi connectivity issues
  - Software updates
  - HW replacement
  - ...
- Debugging from remote location (outside SG office network).

Devicelab “Caroufarm” v2

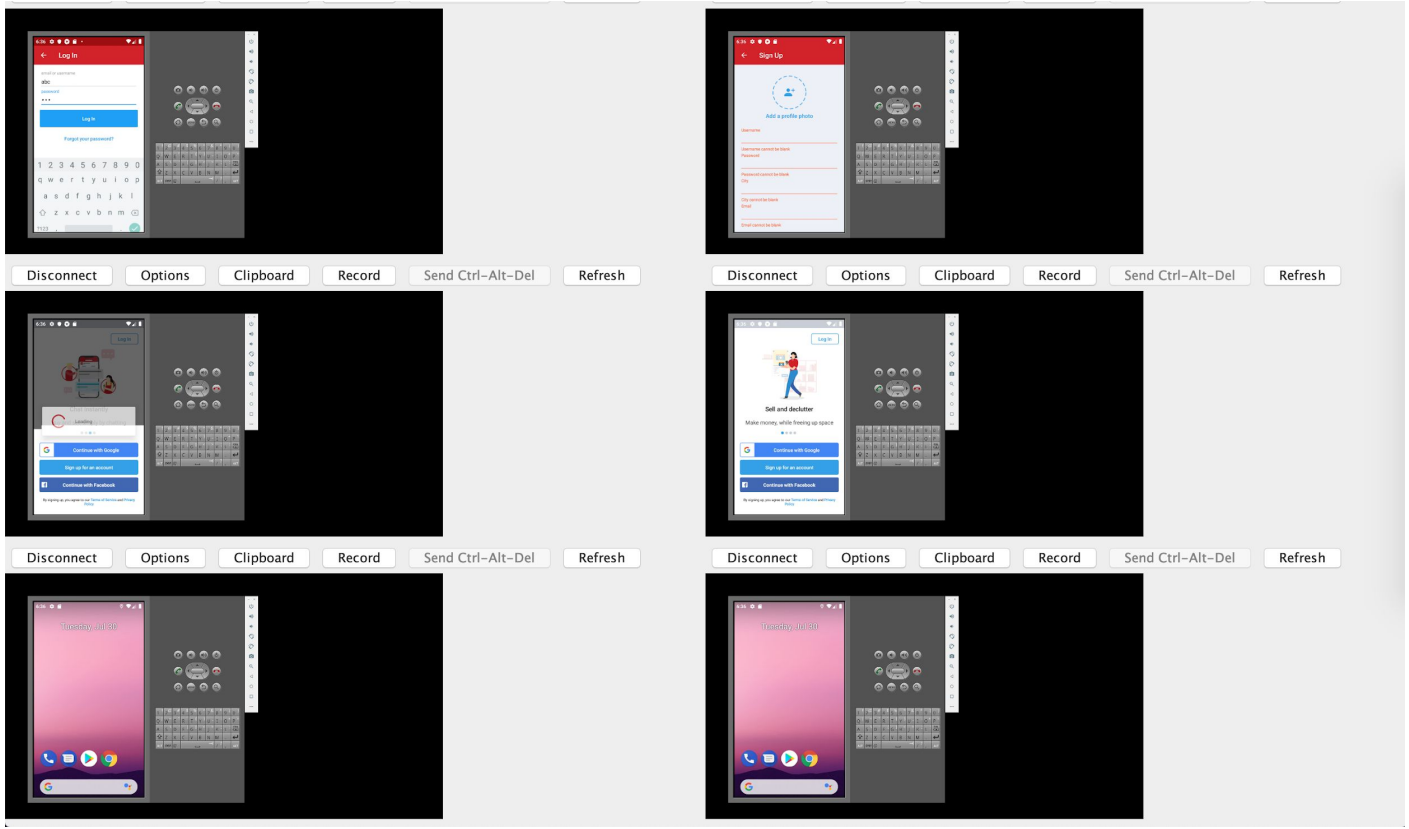


Caroufarm v2 infrastructure

- 50 nodes (docker containers), each requires ~ 16 GB RAM
- AWS EC2 r5d-metal (96 core and 768 GB RAM)

https://aws.amazon.com/blogs/aws/now-available-five-new-amazon-ec2-bare-metal-instances-m5-m5d-r5-r5d-and-z1d/

Caroufarm v2 Demo

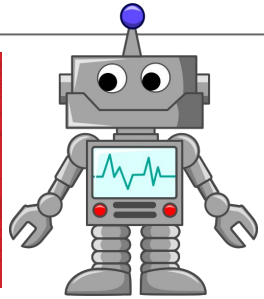


Caroufarm v3 (Vision)

Test execution

- test definition
- required device type
- priority
- ...

enqueue test



iOS internal

Virtual

Physical

Android internal

Virtual

Physical

External vendor(s)

Perfecto

SAUCELABS

AWS

pCloudy.com

carousell

Thank you!