

# Mobile Testing with Appium & Docker

@Testival Meetup  
April 3rd, 2018

@a\_prpic

|

[aprpic@gmail.com](mailto:aprpic@gmail.com)

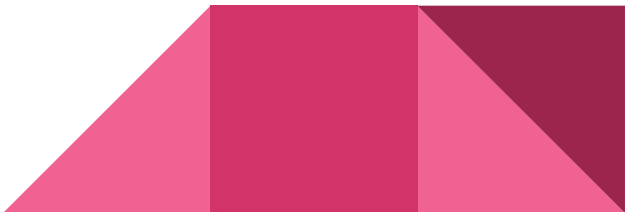
|

Oradian

The background is a solid pink color. In the top right corner, there is a decorative pattern of overlapping triangles in various shades of pink and magenta, creating a geometric, abstract design.

What do you need to test  
a mobile app?

# Roadmap

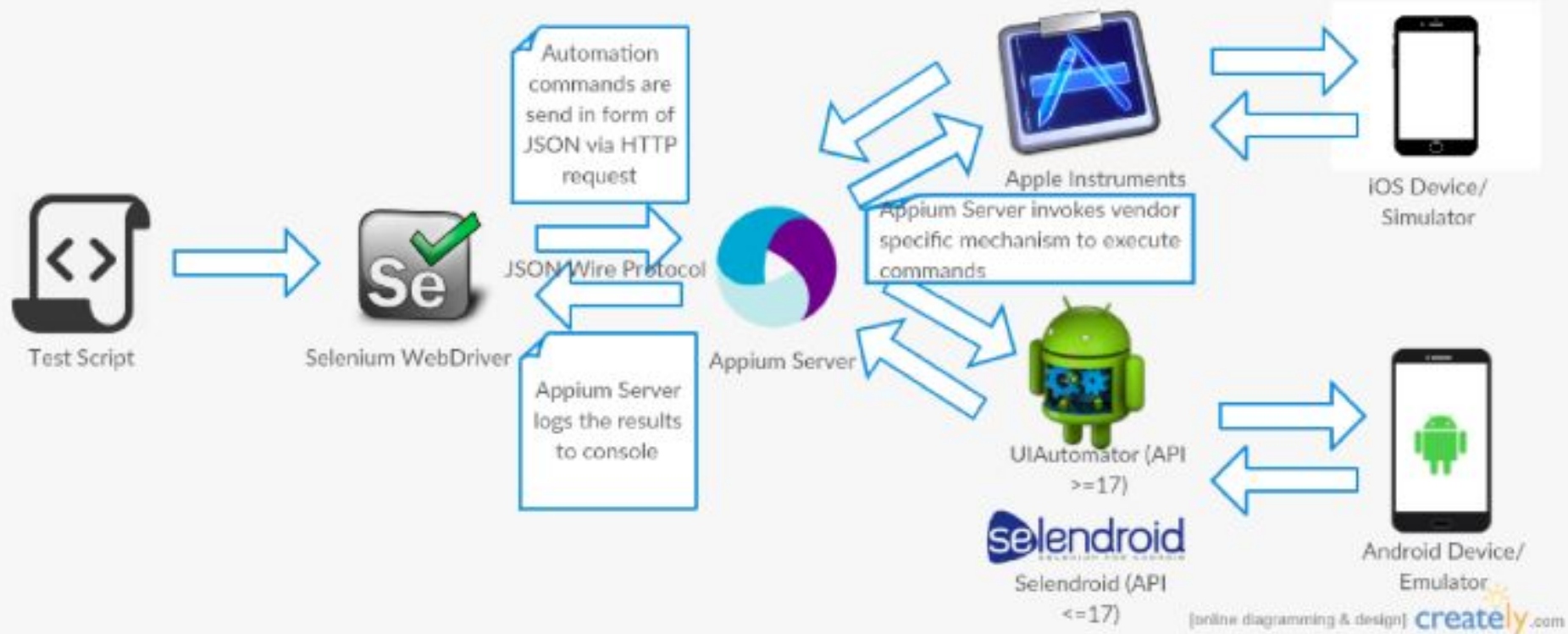
- What is Docker?
  - Docker images & containers
  - What is Appium?
  - Appium concepts
  - Docker + Appium
  - Our workflow @Oradian
  - Resources
- 

# Appium intro

- What is Appium?
  - A tool for automating native, mobile web, and hybrid apps
  - Open sourced & cross-platform - enabling us to run our tests against multiple platforms without code duplication
- Native & Mobile & Hybrid apps
  - Native apps - written and built using the OS-provided SDK and native APIs
  - Web apps - apps which are written in HTML/JS/CSS and deployed via a web server
  - Hybrid apps - apps which mix the two modes into the same app



# Appium design



# Appium

- Appium was designed to meet mobile automation needs according to a philosophy outlined by the following points:
  - You **shouldn't** have to **recompile your app** or modify it in any way **in order to automate it**
  - You **shouldn't be locked into a specific language or framework** to write and run your tests
  - A mobile automation framework **shouldn't reinvent the wheel** when it comes to automation APIs
  - A mobile automation framework **should be open source**, in spirit and practice as well as in name

(from : [Appium documentation](#))



# Appium Concepts

- Client/Server Architecture
- Session
- Desired Capabilities
- Appium Server
- Appium Client
- Appium Desktop



# Getting started with Appium

1. Install Appium by typing `npm install -g appium` into your terminal
2. Start appium by typing `appium`
3. Install webdriver.io client `npm install -g webdriverio`
4. Go to Appium's GitHub page and download this [example ApiDemos-debug.apk](#) file
5. Open your editor of choice, and paste the following code:





# example.js

```
const wdio = require('webdriverio');

const opts = {
  port: 4723,
  desiredCapabilities: {
    platformName: "Android",
    platformVersion: "5.1",
    deviceName: "Nexus 4 API 22",
    app: "/path/to/the/apk/ApiDemos-debug.apk"
  }
};

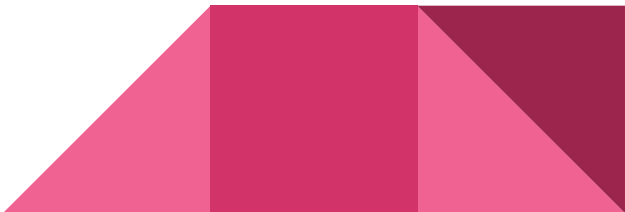
const client = wdio.remote(opts);

client
  .init()
  .click("~App")
  .click("~Alert Dialogs")
  .back()
  .back()
  .end();
```

Execute this by typing:

```
node example.js
```

... assuming you already have Node, Appium, Java, Android Studio & an emulated device sporting an Android 5.1!



# Docker intro

- Docker
  - Tool for running applications in an isolated environment
  - Enables separating applications from the infrastructure
  - Provides the ability to package and run an app in a loosely isolated environment - a container
- Benefits:
  - Same environment
  - Sandboxing projects
  - It just works!





Appium test script



JSON Wire protocol  
(WebDriver usage)



node.js



XCTest / UIAutomation (iOS)  
Selendroid (Android 2.3+)  
UiAutomator (Android 4.2+)  
WinAppDriver (Windows)  
Marionette (Firefox OS)



([image link](#))

# What is a container anyway?

- Container - a sandbox for a process
- Docker Image
  - Image inheritance
- Dockerfile

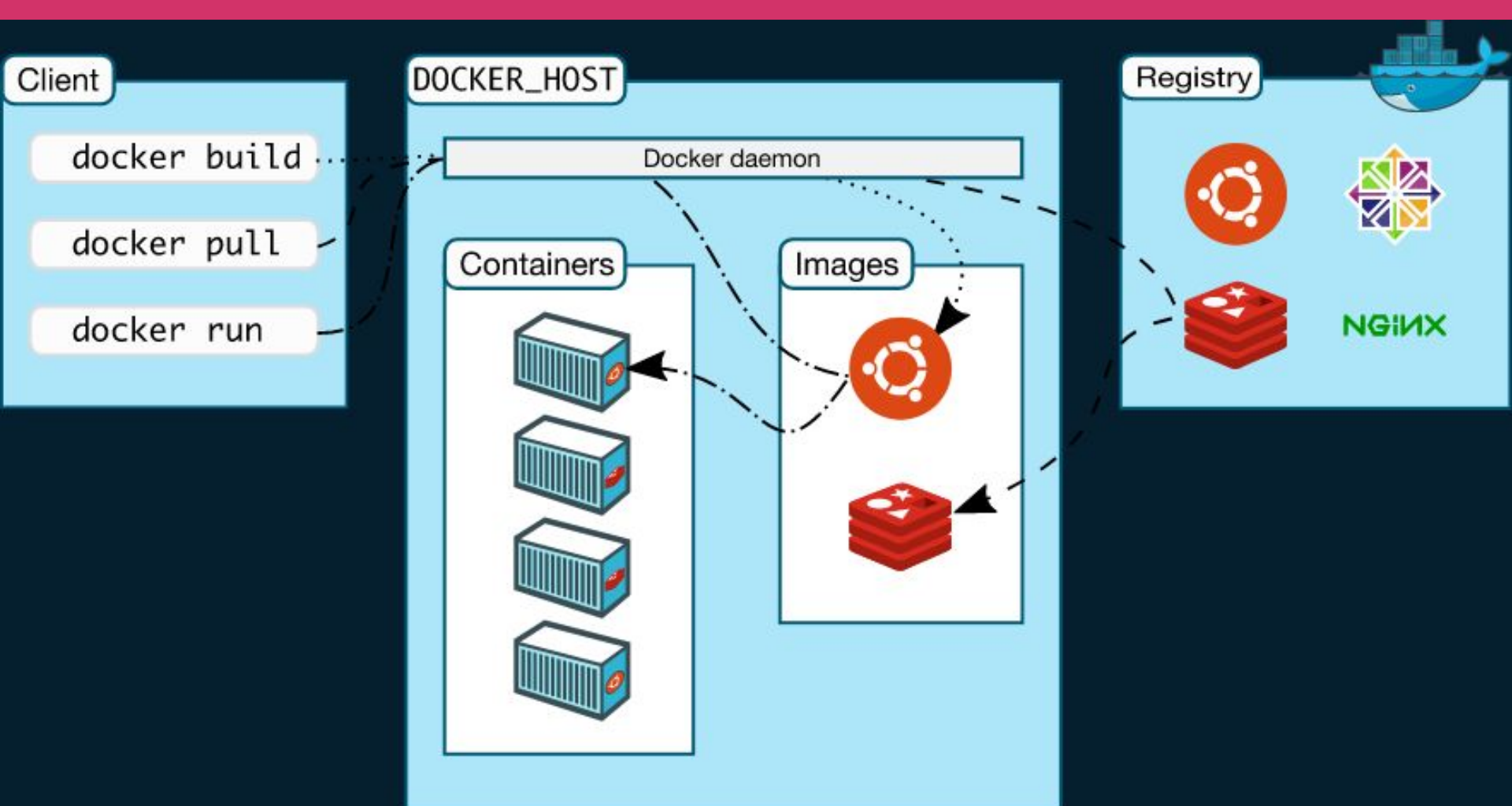
```
FROM ubuntu:16.04
LABEL maintainer="Ana Prpic"

# Set WORKDIR
WORKDIR /root

# Install Java
ENV JAVA_VERSION=8

RUN apt-get update -y \
    && apt-get install -y \
    openjdk-${JAVA_VERSION}-jdk g++-multilib \
    && apt-get -y clean

# Set JAVA_HOME
ENV
JAVA_HOME="/usr/lib/jvm/java-${JAVA_VERSION}-openjdk-amd64" \
    PATH=$PATH:$JAVA_HOME/bin
```



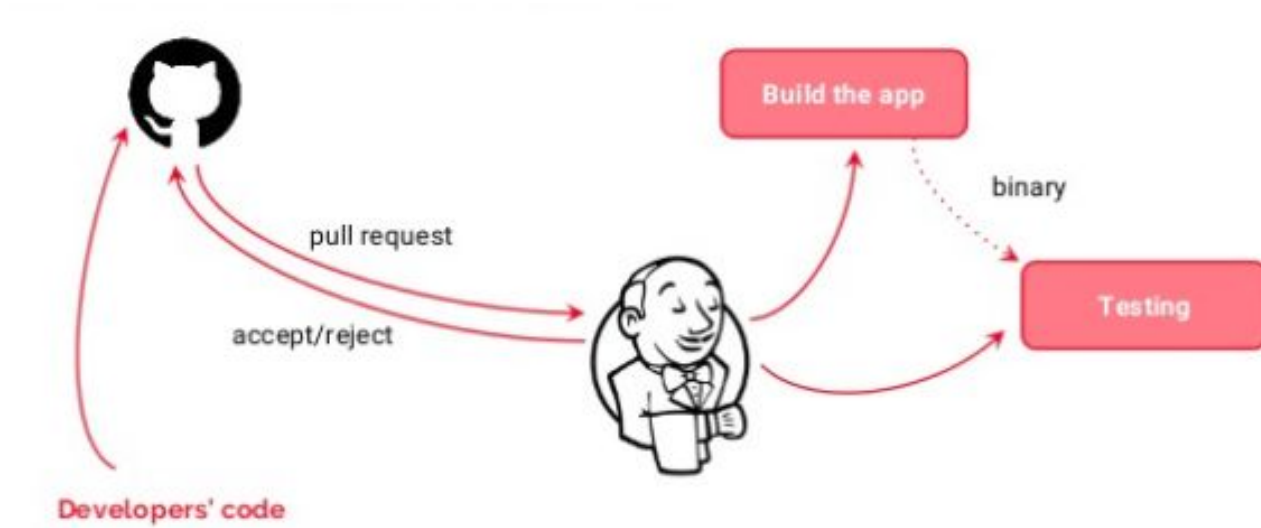
# Getting started with Docker

1. Go to [docker.com](https://docker.com) and download Docker for your OS of choice
2. Create an empty file called "Dockerfile" and define from which existing image you want to build from
3. Go to [hub.docker.com](https://hub.docker.com) and search for available images; the best ones to look out for are the "official" ones
4. Open your terminal, go to the folder where your Dockerfile is located and type `docker build -t image_name .` - this will build the image
5. When this is done, you can type `docker run --name container_name -t image_name`
6. Go into your container by typing `docker exec -ti container_hash /bin/bash`

The background is a solid pink color. In the top right corner, there is a decorative pattern of overlapping geometric shapes: a light pink triangle, a dark pink square, and another light pink triangle, all arranged in a way that suggests a larger, partially visible grid or architectural structure.

Tying it all together...

# Setup @Oradian





# Resources

- <https://github.com/appium> - Appium official docs
- <https://docs.docker.com/> - Docker official docs
- <https://github.com/appium/appium-docker-android> - Testing with real devices
- <https://github.com/butomo1989/docker-android> - Testing with emulated devices
- <https://github.com/SrinivasanTarget/awesome-appium> - All things Appium
- Advanced:  
<http://www.assertselenium.com/continuous-delivery/setting-up-selenium-grid-using-docker/>



# Thank you!

Questions?