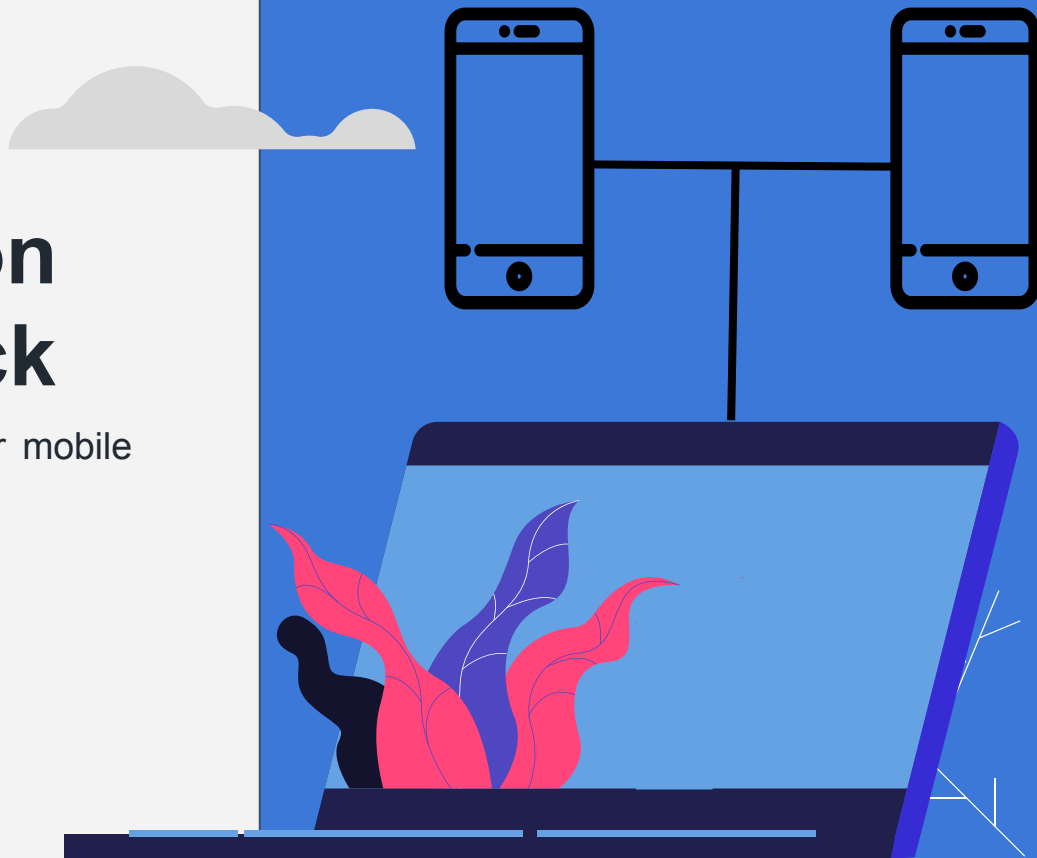


Mobile Automation with BrowserStack

How to build an integration between your mobile automation, BrowserStack and App Center



About me



Thialison Souza

- QA Engineer in SumUp
- Coffee helps me to improve my development process
- Brief experience in China
- Like to talk about Finance

github.com/Thialison
linkedin.com/in/thialison
[@thialison](#)



Goals

What is the purpose of this presentation?

- Initial Challenges;



Goals

What is the purpose of this presentation?

- Initial Challenges;
- SumUp's case in Mobile Testing;



Goals

What is the purpose of this presentation?

- Initial Challenges;
- SumUp's case in Mobile Testing;
- Starting an Automation project;



Goals

What is the purpose of this presentation?

- Initial Challenges;
- SumUp's case in Mobile Testing;
- Starting an Automation project;
- Integrating with BrowserStack;



Goals

What is the purpose of this presentation?

- Initial Challenges;
- SumUp's case in Mobile Testing;
- Starting an Automation project;
- Integrating with BrowserStack;
- Running on Jenkins.



Where to start?

Initial Challenges

- Understanding company tech stack and development process;



Where to start?

Initial Challenges

- Understanding company tech stack and development process;
- Evaluate efforts to create a new project;



Where to start?

Initial Challenges

- Understanding company tech stack and development process;
- Evaluate efforts to create a new project;
- Sell to the team and company the initiative;

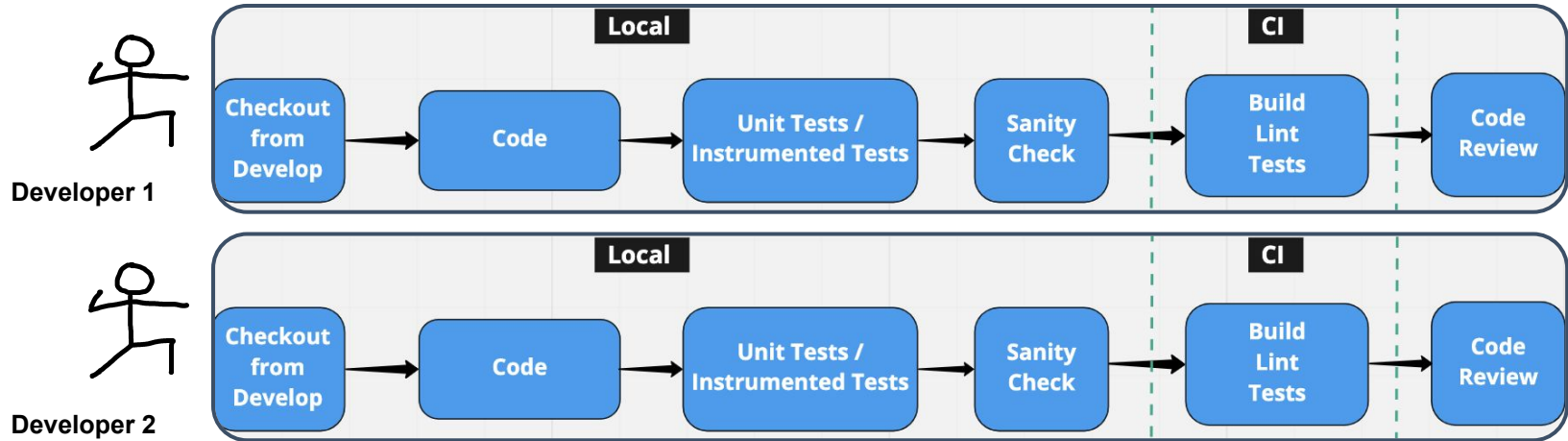


Where to start?

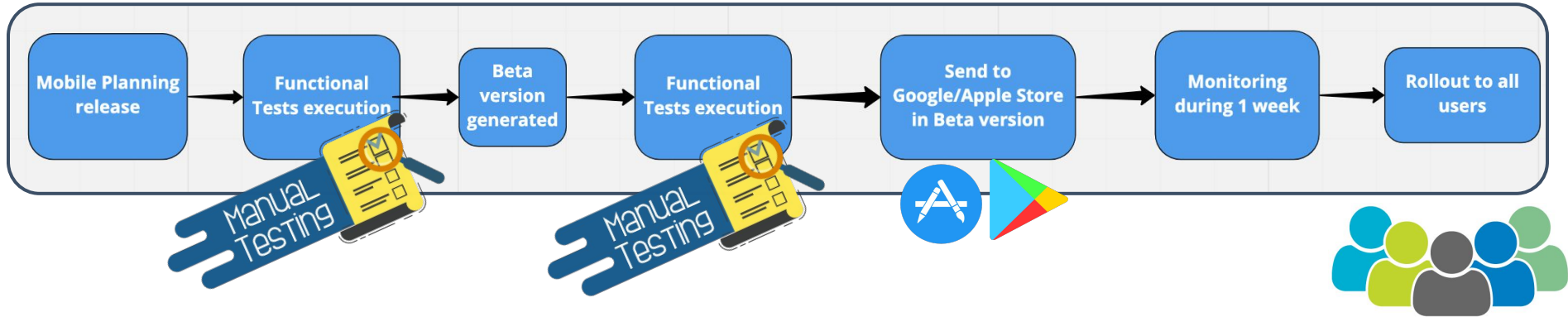
Initial Challenges

- Understanding company tech stack and development process;
- Evaluate efforts to create a new project;
- Sell to the team and company the initiative;
- Include tasks during the Sprint.

Mobile Development Process

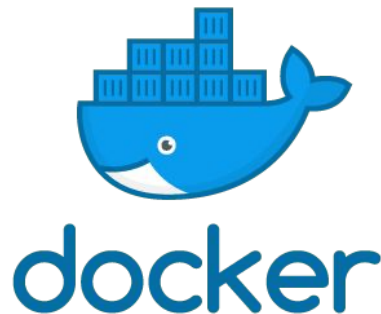


Mobile Release



ONCE A
WEEK

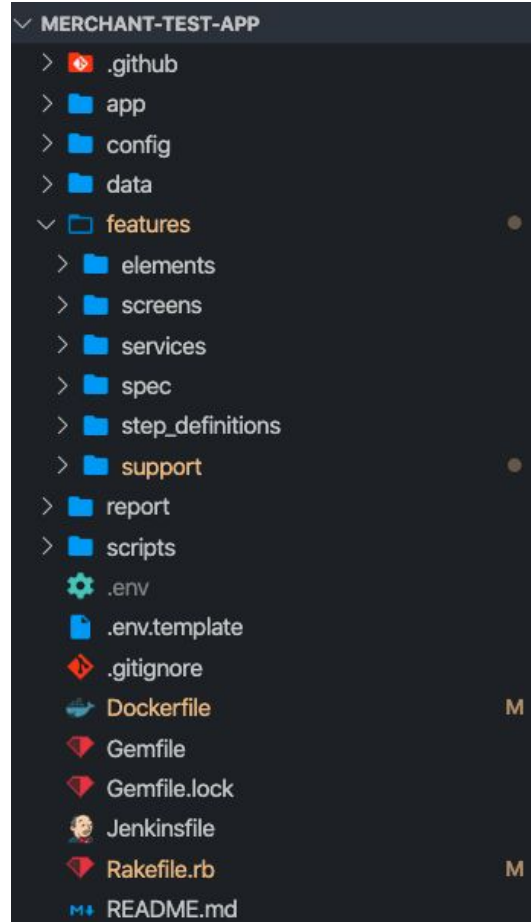
Starting a new Repo



Jenkins

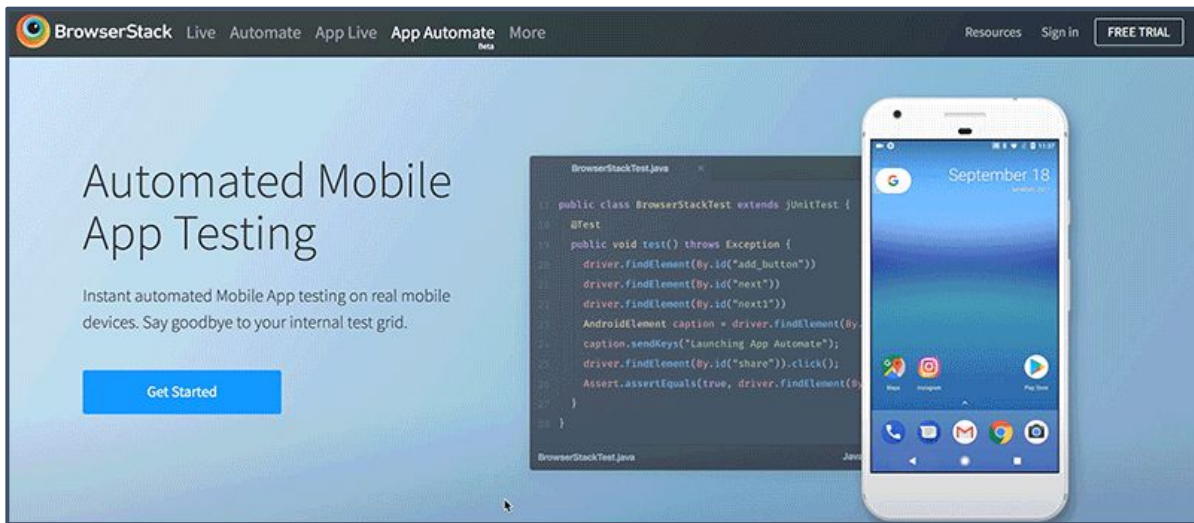
Automation Project

Final Architecture



What is BrowserStack?

It is a solution that enables developers to test their websites and mobile applications manually or automatically.



Main Features



Easy App Upload

Never upload the same build twice. Use REST API to query your most recent builds or upload a new build.



Integrate effortlessly

Integrate with BrowserStack using Appium, Espresso, XCUITest and EarlGrey.



Real Device Cloud

Instant access to wide range of Real iOS and Android devices for more accurate App testing results.



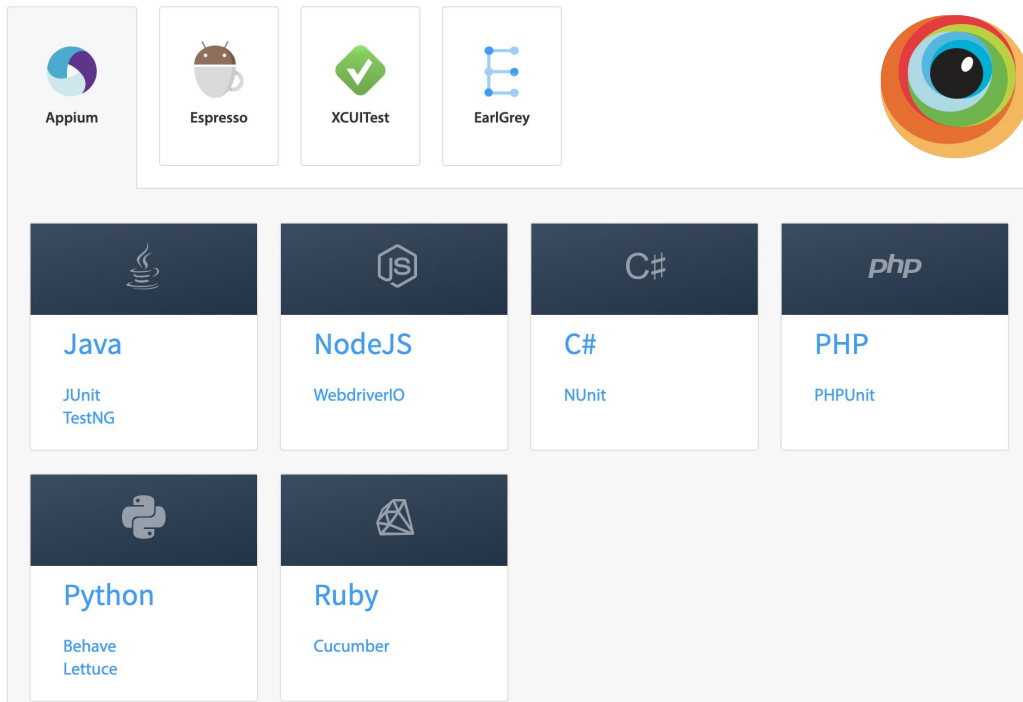
Debug and fix instantly

Debug your Apps instantly using text logs, video recordings and screenshots of the test run.



BrowserStack

Frameworks Integration



Getting Started

Step 1: Setup environment

Ensure you have Ruby libraries installed.

```
#Install using rubygems  
gem install 'appium_lib'  
gem install selenium-webdriver
```

Step 2: Upload your app

Upload your Android app (.apk file) or iOS app (.ipa file) to the BrowserStack servers using the REST API.

```
curl -u "thialisonsoz[redacted]" \\  
-X POST "https://api-cloud.browserstack.com/app-automate/upload" \\  
-F "file=@/path/to/app/file/Application-debug.apk"
```

Please note the App URL (bs://<hashed appid>) returned in the response of this call:

```
{"app_url": "bs://<hashed appid>"}
```

Basic Setup

```
require 'rubygems'
require 'appium_lib'

.

username = 'user_browser_stack'
access_key = 'key_browser_stack'

.

caps = {}
caps['build'] = 'Ruby Appium Sample'
caps['name'] = 'single_test'
caps['device'] = 'Samsung Galaxy S8 Plus'
caps['platformName'] = 'android'
caps['browserstack.debug'] = true
caps['app'] = 'bs://'
```

Basic Setup

```
appium_driver = Appium::Driver.new({
  'caps' => caps,
  'appium_lib' => {
    :server_url => "http://#{username}:#{access_key}@hub-cloud.browserstack.com/wd/hub"
  }, true)
driver = appium_driver.start_driver

wait = Selenium::WebDriver::Wait.new(:timeout => 30)
wait.until { driver.find_element(:accessibility_id, "Search Wikipedia").displayed? }
element = driver.find_element(:accessibility_id, "Search Wikipedia")
element.click
```



BrowserStack

REST API

- App Upload
- Projects
- Builds
- Sessions
- Parameters
- Devices
- Media Upload
- API Status

Project Configuration

```
def mark_test_passed_failed(status)
  session_id = filter_sessions_running(filter_build_running)
  @options[:body] = { status: status.to_s }
  self.class.put("/sessions/#{session_id}.json", @options)
end

def filter_build_running
  res = self.class.get('/builds.json?status=running', @options)
  res.parsed_response[0]['automation_build']['hashed_id']
end

def filter_sessions_running(build_id)
  res = self.class.get("/builds/#{build_id}/sessions.json?status=running", @options)
  res.parsed_response[0]['automation_session']['hashed_id']
end
```

```
After do |scenario|
  status = scenario.passed? ? 'passed' : 'failed'
  if ENV['BUILD'] == 'browser_stack'
    bs = BrowserStackHelpers.new
    bs.mark_test_passed_failed(status)
  end
end
```

Docker

Creating your own Dockerfile

1. Create layer from some ruby version;
2. Execute the installations that is required inside your container;
3. Copy your folder to a new folder created in the Docker Image;
4. Install your project dependencies with bundle install.

```
Dockerfile ×
Dockerfile > ...
You, a few seconds ago | 2 authors (You and others)
1 FROM ruby:2.6.3
2
6
7 RUN apt-get update -qq && apt-get install -y nodejs
8
9 RUN mkdir /automation
10 WORKDIR /automation
11
12 COPY Gemfile /automation/Gemfile
13 COPY Gemfile.lock /automation/Gemfile.lock
14
15 RUN bundle install
16 COPY . /automation
```


JenkinsFile

```
pipeline {  
  agent { dockerfile true }  
  options {  
    disableConcurrentBuilds()  
  }  
  triggers {  
    cron('H 11 * * 1-5')  
  }  
}
```

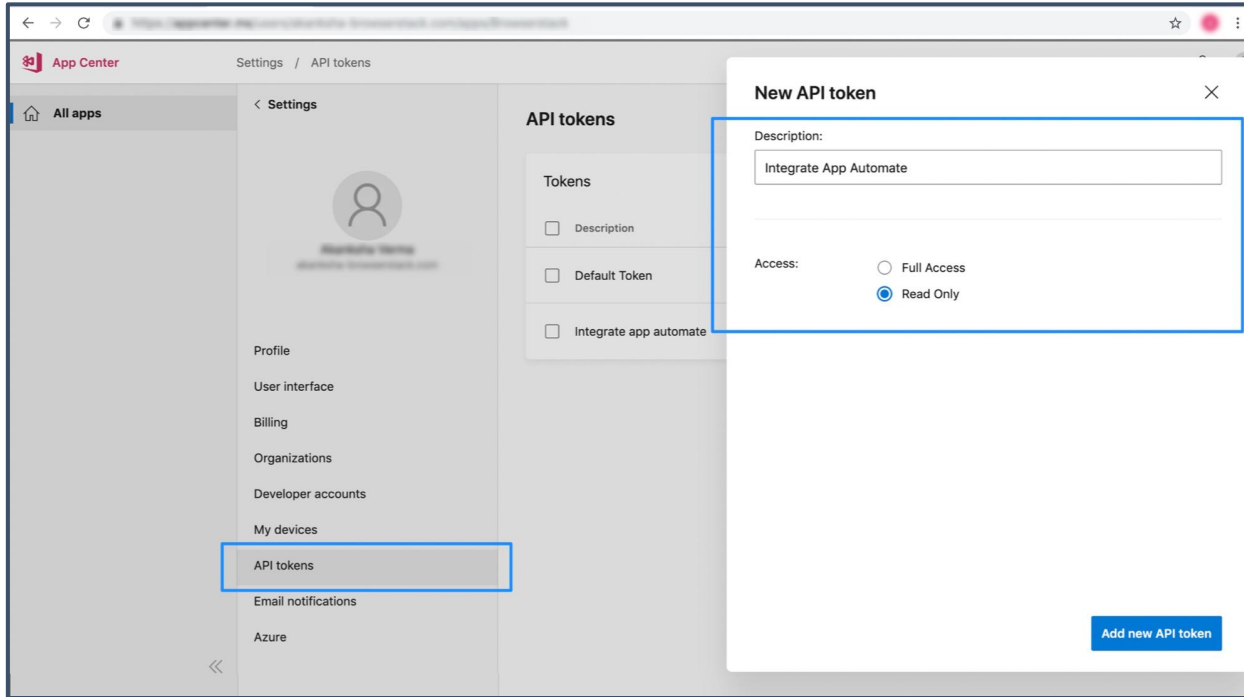
1. Create an Image / Container
2. Options
3. Triggers -> Run at 11am between monday to friday.

JenkinsFile

```
stages {
    stage ('Test') {
        when { branch 'master' }
        steps {
            withCredentials([...
            ]) {
                sh "Run Automation"
            }
        }
    }
    stage ('Reports') { ...
}
post {
    success { ...
}
```

1. Stages
2. When
3. With Credentials
4. Steps
5. Post -> Success or Failure

Integrating with App Center



Integrating with App Center

Enter the **app_secret** and **distribution_group_id** obtained from previous steps. The App URL is returned in the response of this API call.

```
curl -X GET "https://api.appcenter.ms/v0.1/public/sdk/apps/{app_secret}/distribution_groups/{distribution_group_id}/releases/latest"  
-H "accept: application/json"
```


Sample Response:

```
{  
  "app_name": "app name",  
  "app_display_name": "app display name",  
  "app_os": "app OS type",  
  "app_icon_url": "https://the public url for app download",  
  "id": 1,  
  "version": "version number",  
  .....  
  "download_url": "https://your url"  
  .....  
}]
```

CI

Jenkins Pipeline

| | | Declarative: Checkout SCM | Declarative: Agent Setup | Test | Reports | Declarative: Post Actions |
|--|----------------------------|------------------------------|-----------------------------|--------------------|-----------------|------------------------------|
| Average stage times: (Average full run time: ~3min 29s) | | 5s | 34s | 2min 19s | 521ms | 381ms |
| #7 | Feb 27 08:29 No Changes | 5s | 616ms | 2min 19s | 575ms | 374ms |
| #6 | Feb 26 08:29 No Changes | 5s | 635ms | 2min 31s failed | 533ms failed | 407ms |
| #5 | Feb 25 08:29 No Changes | 4s | 1s | 2min 11s | 570ms | 402ms |
| #4 | Feb 24 08:29 No Changes | 6s | 1min 18s | 2min 21s | 469ms | 407ms |
| #3 | Feb 21 08:29 No Changes | 5s | 609ms | 2min 29s | 499ms | 369ms |
| #2 | Feb 20 08:29 No Changes | 6s | 1min 22s | 2min 12s | 527ms | 396ms |



Learnings

- **Work together and divide your challenges with your team;**



Learnings

- **Work together and divide your challenges with your team;**
- **Start with small things;**



Learnings

- **Work together and divide your challenges with your team;**
- **Start with small things;**
- **Problem first;**



Learnings

- **Work together and divide your challenges with your team;**
- **Start with small things;**
- **Problem first;**
- **Celebrate every achievement.**



Where to find?

References

1. <https://www.browserstack.com/app-automate/appium-ruby>
2. <https://www.browserstack.com/app-automate/rest-api?framework=appium>
3. <https://www.browserstack.com/app-automate/appcenter>
4. <https://jenkins.io/doc/book/pipeline/jenkinsfile/>
5. <https://www.docker.com/play-with-docker>

Three parallel diagonal lines in shades of blue, starting from the left edge and extending towards the center of the slide.

Thanks and do you have Questions?