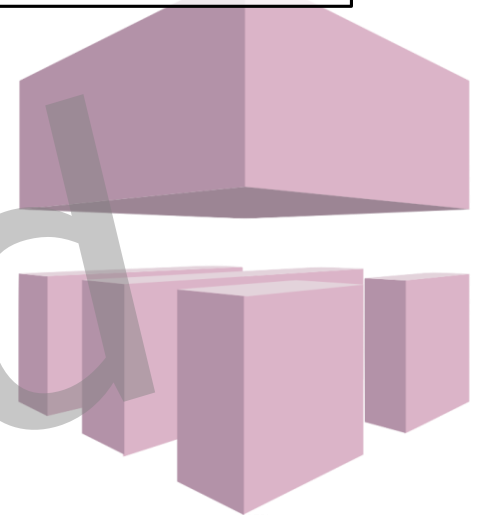


このコンテンツは公開から3年以上経過しており内容が古い可能性があります
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

AWS Device Farm

クラウドにホスティングされた Android, Fire OS の
実際のデバイスでモバイルアプリのテストオートメーション



AWS Black Belt Tech Webinar 2015 (旧マイスターシリーズ)

Amazon Data Services Japan

Solutions Architect - Takayuki Shimizu

2015.7.22

AWS Black Belt Tech Webinar へようこそ！

参加者は、自動的にミュートになっています

質問を投げることができます！

- オンラインセミナーの仕組みを使って、書き込んでください
- 最後の Q & A の時間で、できるだけ回答させていただきます

Twitter のハッシュタグは #awsblackbelt でどうぞ

AWS Black Belt Tech Webinar 2015 (旧マイスターシリーズ)

今後の配信予定

- **7/22 AWS Device Farm**
- 7/22 AWS CLI & AWS Tools for Windows PowerShell
- 7/29 AWS CloudHSM & AWS Key Management Service

申し込みサイト

- http://aws.amazon.com/jp/event_schedule/

自己紹介

```
{  
  "Name" : "西谷圭介",  
  "Twitter" : "@Keisuke69",  
  "Profile" : {  
    "Role" : "Solution Architect",  
    "Customers": [  
      "Web Services",  
      "Start-up"  
    ],  
    "Services" : [  
      "AWS Lambda",  
      "Mobile Services"  
    ]  
  }  
}
```



自己紹介

```
{  
  "Name" : "清水 崇之",  
  "Twitter" : "@shimy_net",  
  "Profile" : {  
    "Role" : "Solutions Architect",  
    "Customers": [  
      "Web Services",  
      "なんでもOK"  
    ],  
    "Services" : [  
      "AWS Device Farm",  
      "Mobile Services"  
    ]  
  }  
}
```



アジェンダ

- 解決したい課題と背景
- AWS Device Farm の紹介
- ユースケース
- 利用料金
- まとめ
- Q&A

アジェンダ

- **解決したい課題と背景**
- AWS Device Farm の紹介
- ユースケース
- 利用料金
- まとめ
- Q&A

アプリの品質は重要である

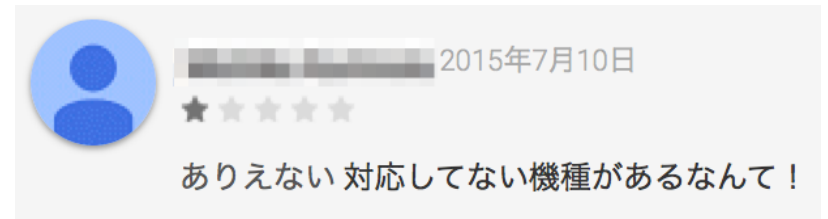
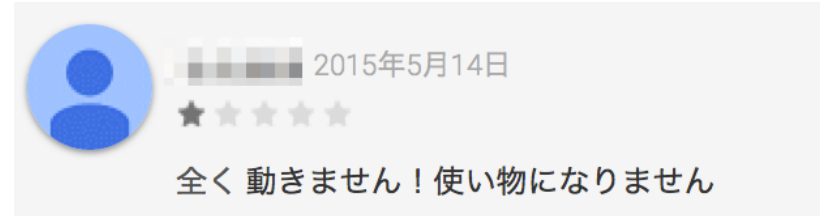
- **48%** のユーザーは、バグやパフォーマンスの問題があれば、アプリを削除する。
- **32%** のユーザーは、このような問題が発生すると、競合アプリを探す。
- ユーザー獲得コストは、アプリごとに **2.74ドル** となっており、2014年4月に比べて年間で **80%** 増加している。

Failing to meet mobile app user expectations - A mobile app user survey from Dimensional Research (2015/3/30)

<http://go.saas.hp.com/apppulse-mobile-survey>

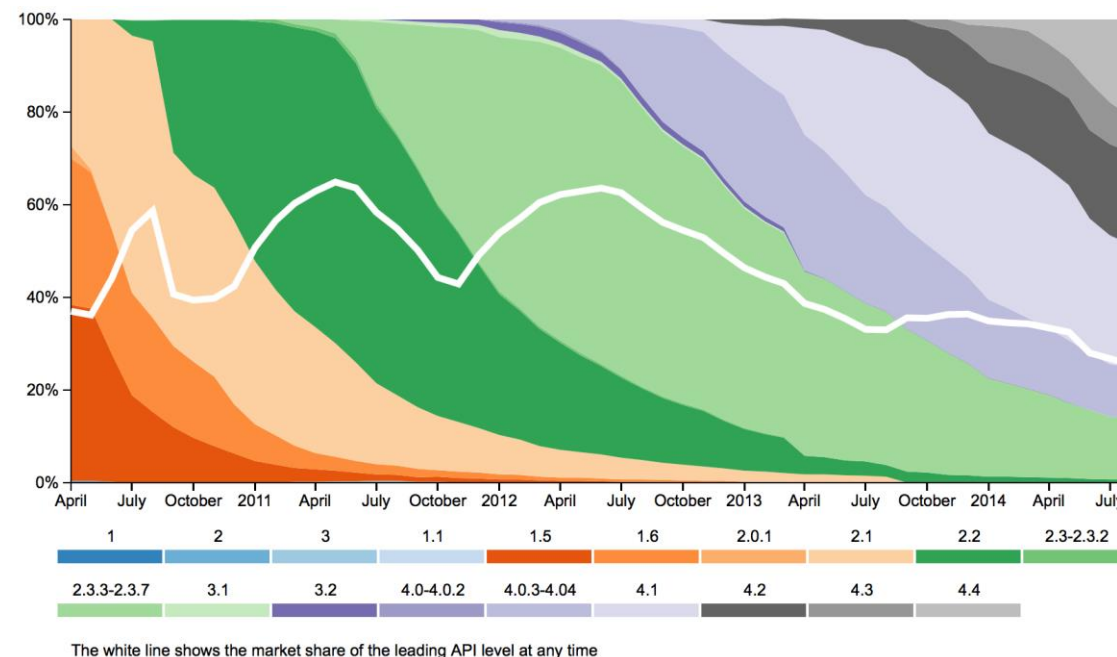
Mobile Marketing Costs See Unprecedented Year-over-Year Growth, According to the Fiksu Indexes (2015/5/29)

<https://www.fiksu.com/about/news-releases/fiksu-indexes-mobile-marketing-costs-see-unprecedented-year-over-year-growth>



デバイス/OS の置き換えは加速している

- ユーザーが持っている Android デバイスの種類は前年 **11,868** に対して、2014 年 8 月では **18,796** に増加している。
- Android ユーザーの 38% は Jelly Bean, 39% は KitKat, 12% は Lollipop, 10% はまだ Gingerbread と Ice Cream Sandwich を利用している。



Open Signal - Android Fragmentation Visualized (2014/8)

<http://opensignal.com/reports/2014/android-fragmentation/>

Android Developers (2015/6)

<http://developer.android.com/intl/zh-CN/about/dashboards/index.html>

テストにかかるコストと複雑さは増加している

- エンタープライズ IT テストと QA 予算の全体比率は 2012 年の **18%** から 2014 年の **26%** に増加しており、**35%** の組織はこの予算配分に満足していない。
- **10** 人に **4** 人の IT 経営者が、アプリを適切にテストする時間がないと言っている。
- **51%** の開発者が、コストと時間の制約のため、テストできないと言っている。



Capgemini - World Quality Report 2014-15

<https://www.capgemini.com/thought-leadership/world-quality-report-2014-15>

77% of Mobile App Purchasers Have Concerns When Buying Mobile Apps - Harris (2013/8/20)

<http://www.businesswire.com/news/home/20130820006244/en/77-Mobile-App-Purchasers-Concerns-Buying-Mobile#.Vaz9yaad1N0>

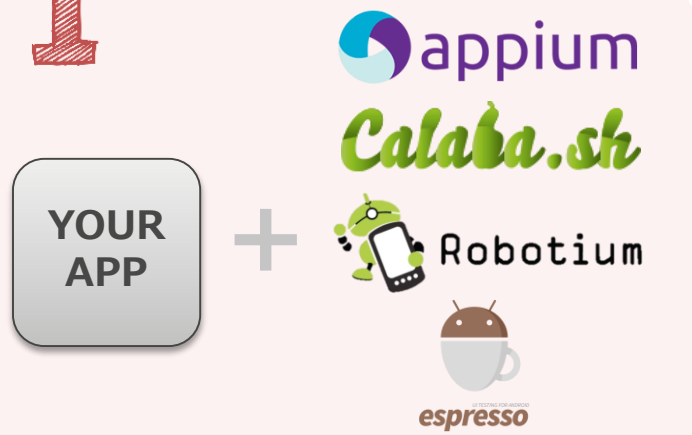
アジェンダ

- 解決したい課題と背景
- **AWS Device Farm の紹介**
- ユースケース
- 利用料金
- まとめ
- Q&A

AWS Device Farm

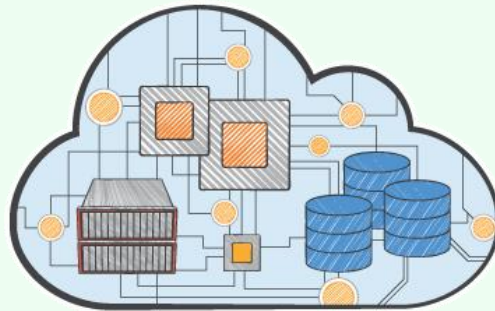
AWS クラウドにホスティングされた数多くのスマートフォン、タブレット、その他の実デバイスで、迅速かつ安全に Android、Fire OS アプリをテストして品質を向上できる。

1



Android、Fire OS アプリをアップロードして、テスト内容を設定して、デバイスを選択する。

2



選択したデバイスをまたいで並列にアプリをテストできる。
迅速かつ安全である。

3



バグ、パフォーマンス、その他の問題を、数分のうちに詳細にレポートする。

DEMO

プロジェクトの作成

Projects allow you to organize your tests, device pools, and other settings. You can test multiple Android builds within a single project.

Project name

BlackBelt

Cancel

Create project

アプリのアップロード

Step 1: Choose application

Step 2: Configure test

Step 3: Select devices

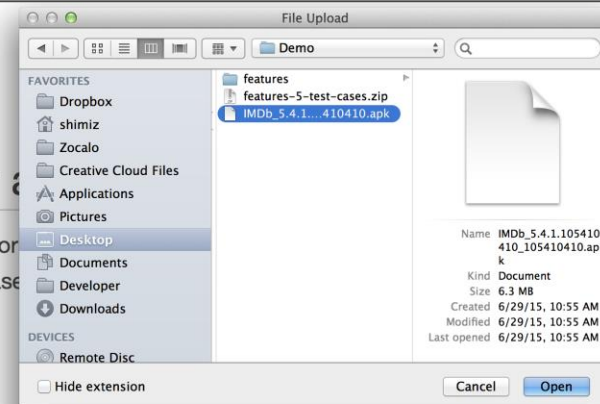
Step 4: Specify device state

Step 5: Review

Choose your application

Upload your application or application's platform base

Upload



Cancel

Next step

アプリ情報の確認

Package	com.imdb.mobile
Activity	com.imdb.mobile.HomeActivity
Minimum SDK	14
Native Code	
Screens	small,normal,large,xlarge
Target SDK	18
Version Code	105410410
Version Name	5.4.1.105410410

Cancel

Next step

テストのアップロード

Step 1: Choose application

Step 2: Configure test

Step 3: Select devices

Step 4: Specify device state

Step 5: Review and start run

Configure a test

Select the test type you would like to use. If you do not

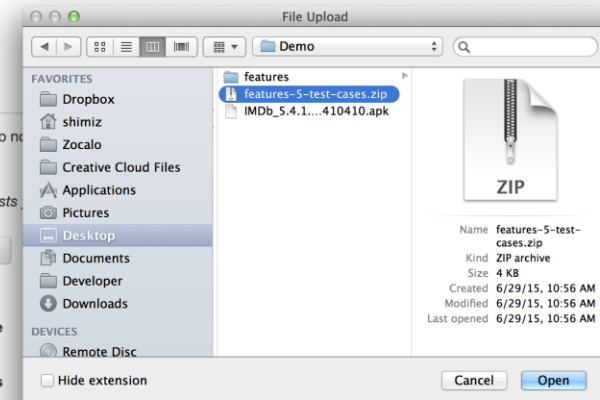
- Built-in: Fuzz
- Appium Java JUnit
- Appium Java TestNG
- Calabash
- Instrumentation
- uiautomator

Run Calabash tests

Upload

Profile

Tags



Cancel

Previous

Next step

デバイスの選択

S

Step 2: Configure test

Select from one of the available device pools or create a new device pool.

Step 3: Select devices

Device pool Top Devices

Create a new device pool

100% Compatibility

Your app is compatible with **5 out of 5** devices in the selected pool.

	Device	Reason
✓	Samsung Galaxy Tab 4 10.1 Nook (WiFi)	4.4.2
✓	Amazon Kindle Fire HDX 7 (WiFi)	4.4.3
✓	Motorola DROID Ultra (Verizon)	4.4.4
✓	Samsung Galaxy S6 (Verizon)	5.0.2
✓	Samsung Galaxy S5 (AT&T)	4.4.2

Cancel

Previous

Next step

環境の設定

Step 1: Choose application

Step 2: Configure test

Step 3: Select devices

Step 4: Specify device state

Step 5: Review and start run

Specify device state

Specify settings to simulate real-world scenarios and device configurations.

Add extra data Upload

Install other apps Upload

Set radio states WiFi Bluetooth GPS NFC

Device location 47.6204 -122.3491

Device locale English, US (en_US)

Cancel

Previous

Review and start run

テスト内容の確認

Step 1: Choose application

Step 2: Configure test

Step 3: Select devices

Step 4: Specify device state

Step 5: Review and start run

Review and start run

Review your run below. Look good? Confirm to start your run.

Application

Name IMDb_5.4.1.105410410_105410410.apk

Test

Type Calabash

Devices

Pool Top Devices

Cancel

Confirm and start run

テストの実施

apk

Your app is currently being tested. Results will appear here as tests complete.

Details Screenshots

Devices

	Device	Test Results	Created
🔄	Amazon Kindle Fire HDX 7 (WiFi)	4.4.3	2015-07-29T10:12+0900
🔄	Motorola DROID Ultra (Verizon)	4.4.4	2015-07-29T10:12+0900
🔄	Samsung Galaxy S5 (AT&T)	4.4.2	2015-07-29T10:12+0900
🔄	Samsung Galaxy S6 (Verizon)	5.0.2	2015-07-29T10:12+0900
🔄	Samsung Galaxy Tab 4 10.1 Nook (WiFi)	4.4.2	2015-07-29T10:12+0900

テスト結果のサマリ

6

Details Screenshots

Unique problems

1 UNIQUE FAILURE FOUND

- As a valid user perform movie search failed: Timeout waiting for elements: * {text CONTAINS[c] 'Home'} (Calabash::Android::WaitHelpers::WaitError) **2**
- !** Samsung Galaxy S5 (Sprint) 4.4.4 **Movie search > As a valid user perform movie search**
- !** Samsung Galaxy S5 (AT&T) 4.4.2 **Movie search > As a valid user perform movie search**

1 UNIQUE WARNING FOUND

テスト結果の詳細

apk

Details Screenshots Performance Files

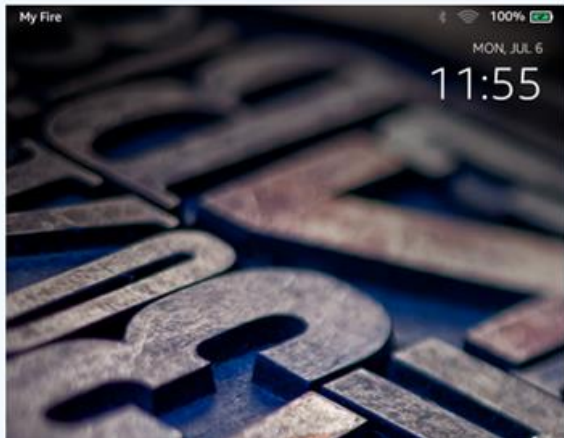
Suites

Suite	Test Results	Created
Setup Suite	1	2015-07-22T12:17+0900
STARMeter	1	2015-07-22T12:17+0900
Featured Movie	1	2015-07-22T12:17+0900
Movie Search	1	2015-07-22T12:17+0900
Featured TV Show	1	2015-07-22T12:17+0900
Top News	1	2015-07-22T12:17+0900
! Teardown Suite	1 !	2015-07-22T12:17+0900

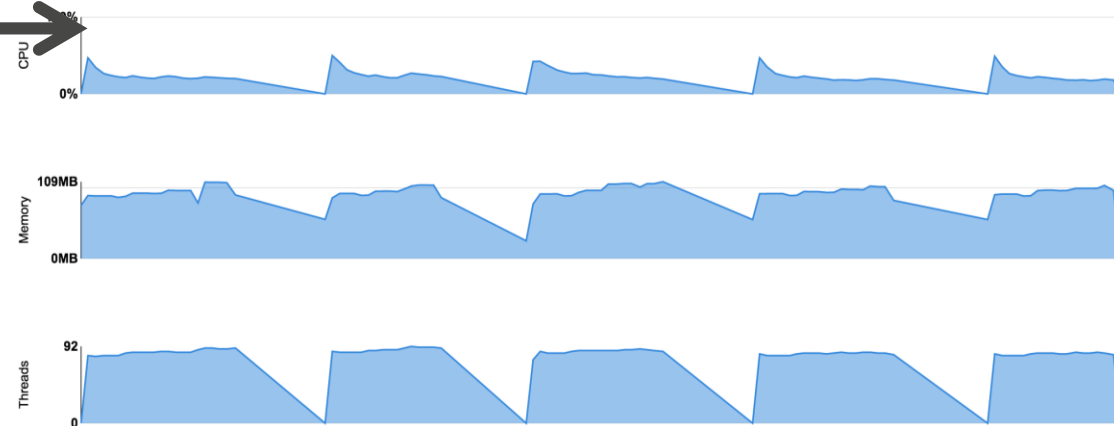
Download logs

スクリーンショットの確認

Screenshots

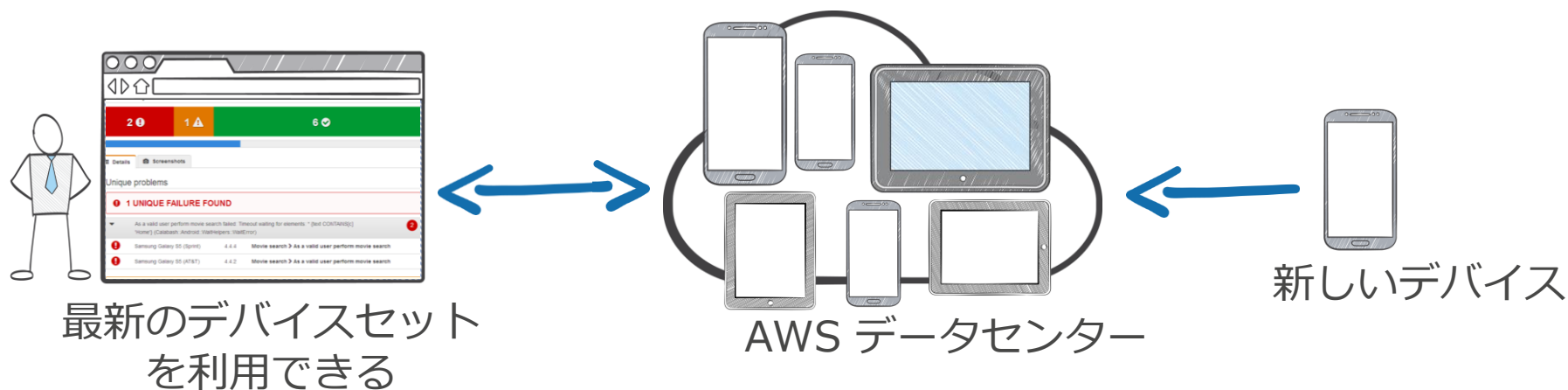


パフォーマンスの確認



デバイス

- デバイスのプロファイルは、製造メーカー／型 & モデル／OS バージョン／キャリアの組み合わせである。
- AWS データセンターにホスティングされており、ユーザーの要求に応じてスケールできる。
- 十分なキャパシティ（多くの同一のデバイスプロファイル）を保有している。
- ルート権限では動作しない。
- 市場情報に基づいてデバイスは用意される。
- 新しいデバイスが発売されたら追加される。



デバイス

テストでは、互換性のあるすべてのデバイスから個別に選択できる。もしくは、利用したいデバイスを事前に選択しておきデバイスプールを作成することもできる。

✔ Select devices

Select from one of the available device pools or create a new device pool.

Device pool: All Devices Create a new device pool

100% Compatibility
Your app is compatible with **7 out of 7** devices in the selected pool.

	Device	Reason
✔	Samsung Galaxy S5 (T-Mobile)	4.4.2
✔	Samsung Galaxy S5 (Verizon)	4.4.4
✔	Samsung Galaxy S4 (Verizon)	4.4.2
✔	Amazon Kindle Fire HD 7 (WiFi)	4.4.3
✔	LG G3 (Sprint)	4.4.2
✔	Samsung Galaxy S5 (Sprint)	4.4.4
✔	Samsung Galaxy S5 (AT&T)	4.4.2

Create a new device pool

Create your own device pools to test on specific devices. You can select devices individually or use rules to create dynamic pools that adapt as new devices become available.

Name: Test Pool

Description: Enter a short description for your new device pool (optional)

Add rules

Field	Operator	Operand
Add rule		

Select devices

<input type="checkbox"/>	Name	Platform	OS	Form factor
<input type="checkbox"/>	<input type="text"/>	ANDROID ✕	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	Motorola DROID Ultra (Verizon)	ANDROID	4.4.4	PHONE
<input type="checkbox"/>	Samsung Galaxy S5 (Verizon)	ANDROID	4.4.4	PHONE
<input checked="" type="checkbox"/>	LG Optimus L70 (MetroPCS)	ANDROID	4.4.2	PHONE
<input checked="" type="checkbox"/>	Samsung Galaxy S4 (AT&T)	ANDROID	4.4.4	PHONE

注意) デフォルトでは数個の試用デバイスしか利用できない。すべてのデバイスを利用するには申請が必要。

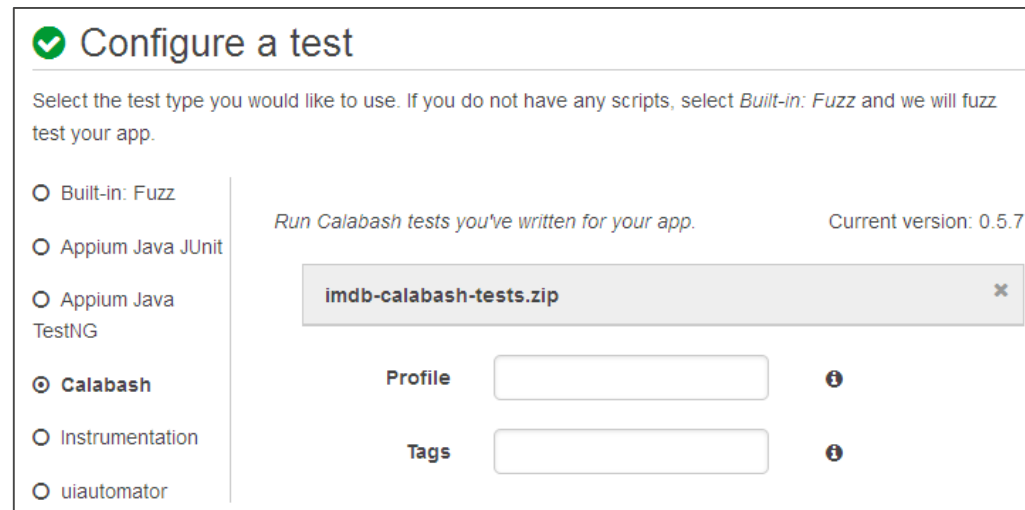
複数のフレームワークをサポート

テストオートメーションフレームワーク

- Appium (JUnit, TestNG)
- Calabash
- Instrumentation (JUnit, Espresso, Robotium)
- uiautomator

ポイント

- プロジェクトごとに最適なソリューションを使ってテストを作れる。
- 開発チームと QA チームで要件が異なっても、ひとつの共通プラットフォームに統合できる。
- 新しく出現してくるフレームワークもサポートされることで統合できる。



Configure a test

Select the test type you would like to use. If you do not have any scripts, select *Built-in: Fuzz* and we will fuzz test your app.

Built-in: Fuzz

Appium Java JUnit

Appium Java TestNG

Calabash

Instrumentation

uiautomator

Run Calabash tests you've written for your app. Current version: 0.5.7

imdb-calabash-tests.zip

Profile

Tags

```

@Before
public void setUp() throws Exception {
    File classpathRoot = new File(System.getProperty("user.dir"));
    File appDir = new File(classpathRoot, "../../apps/ApiDemos/bin");
    File app = new File(appDir, "ApiDemos-debug.apk");
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability("deviceName", "Android Emulator");
    capabilities.setCapability("platformVersion", "4.4");
    capabilities.setCapability("app", app.getAbsolutePath());
    capabilities.setCapability("appPackage", "io.appium.android.apis");
    capabilities.setCapability("appActivity", ".ApiDemos");
    driver = new AndroidDriver<>(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);
}

@After
public void tearDown() throws Exception {
    driver.quit();
}

@Test
public void apiDemo(){
    WebElement el = driver.findElement(By.name("Animation"));
    assertEquals("Animation", el.getText());
    el = driver.findElementByClassName("android.widget.TextView");
    assertEquals("API Demos", el.getText());
    el = driver.findElement(By.name("App"));
    el.click();
    List<WebElement> els = driver.findElementsByClassName("android.widget.TextView");
    assertEquals("Activity", els.get(2).getText());
}

```

```

@Before
public void setUp() throws Exception {
    // set up appium
    File classpathRoot = new File(System.getProperty("user.dir"));
    File appDir = new File(classpathRoot, "../../apps/ContactManager");
    File app = new File(appDir, "ContactManager.apk");
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability("deviceName", "Android Emulator");
    capabilities.setCapability("platformVersion", "4.4");
    capabilities.setCapability("app", app.getAbsolutePath());
    capabilities.setCapability("appPackage", "com.example.android.contactmanager");
    capabilities.setCapability("appActivity", ".ContactManager");
    driver = new AndroidDriver<>(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);
}

@After
public void tearDown() throws Exception {
    driver.quit();
}

@Test
public void addContact(){
    WebElement el = driver.findElement(By.name("Add Contact"));
    el.click();
    List<AndroidElement> textFieldsList = driver.findElementsByClassName("android.widget.EditText");
    textFieldsList.get(0).sendKeys("Some Name");
    textFieldsList.get(2).sendKeys("Some@example.com");
    driver.swipe(100, 500, 100, 100, 2);
    driver.findElementByName("Save").click();
}

```

```
initial_experience.feature - Calabash - Atom
initial_experience.feat... x
1 Feature: Initial experience
2   As a user I want a helpful and simple initial
3   experience with the app. I should be able to get help
4   and login to an existing WordPress site.
5
6 @reinstall
7 Scenario: Obtaining more information # features/login.feature:
8   Given I am on the first experience screen # features/step_definitio
9   And I choose to get more information # features/step_definitio
10  Then I am taken to the information screen # features/step_definitio
11
12 Scenario: Create new account # features/login.feature:13
13   Given I am about to login # features/step_definitions/log
14   Then I am able to create an account # features/step_definitions/log
15
16 Scenario: Add site - Invalid login
17   Given I am about to login #
18   When I enter invalid credentials #
19   Then I am presented with an error message to correct credentials #

initial_experience.feature 19,3 UTF-8 Cucumber Steps
```

ビルトインのテストスイート

ビルトイン Fuzz テスト

Fuzz テストはアプリにランダムなユーザー入力（タッチ、スワイプ、キーボード入力）を与えて UI テストを実施する。イベント数、イベント間隔、ランダムイベントに使われる乱数根 (seed) を設定できる。テストをまたいで同じ乱数根 (seed) を利用すれば、同じイベントのシーケンスを再現できる。

ポイント

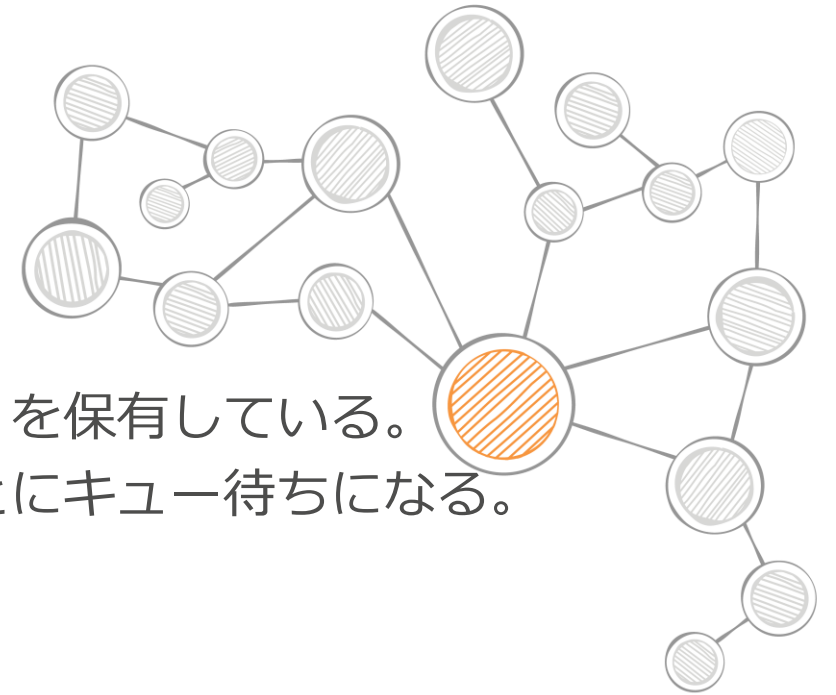
- テストスクリプトを書かずにテストを実施できる。
- 数分で互換性問題を特定できる。
- テストスクリプトは以下の動作をする。
 - アプリのインストール
 - アプリの起動
 - Fuzz テスト
 - アプリのアンインストール



並列実行

並列実行

- 選択したデバイスで並列にテストを実施できる。
- 十分なキャパシティ（多くの同一のデバイスプロファイル）を保有している。
- デバイスプロファイルの利用が満杯になると、デバイスごとにキュー待ちになる。
- テストが完了したらデバイスごとに結果を得られる。



ポイント

- オートメーションテストを一日何回も実施するような場合に有効。
- 自前のオートメーションソリューションで、大量のデバイスで並列実行できるようなインフラを構築することは、複雑で技術的なチャレンジであるが、AWS Device Farm であれば簡単に実現。

実世界のシナリオを再現するために環境を設定できる

実世界のシナリオを再現するためにデバイスの状態を設定できる。

- 言語／地域の選択
- 位置情報（経緯度）
- ネットワークの状況
 - WiFi、Bluetooth、GPS、NFC
- 追加のデータ
- 他アプリのインストール

Specify device state

Specify settings to simulate real-world scenarios and device configurations.

Add extra data ⓘ

Install other apps ⓘ

Set radio states

- WiFi ⓘ
- Bluetooth
- GPS
- NFC

Device location ⓘ ⓘ

Device locale ⓘ

レポート

レポートは、各デバイスごとの詳細情報だけでなく高度な結果を含んでいる。

- 成功／失敗の情報
- クラッシュレポート
- テストログ
- デバイスログ
- スクリーンショット
- パフォーマンス情報

The screenshot displays a test report interface with a summary bar at the top showing 2 failures (red), 1 warning (orange), and 6 successes (green). Below this, there are tabs for 'Details' and 'Screenshots'. The main content area is titled 'Unique problems' and contains a red box indicating '1 UNIQUE FAILURE FOUND'. This failure is expanded to show a detailed error message: 'As a valid user perform movie search failed: Timeout waiting for elements: * {text CONTAINS[c] 'Home'} (Calabash::Android::WaitHelpers::WaitError)'. Below this, two rows of device-specific failure details are shown for Samsung Galaxy S5 (Sprint and AT&T). A yellow box indicates '1 UNIQUE WARNING FOUND', which is expanded to show a warning for an Amazon Kindle Fire HD 7 (WiFi).


Device	Version	Action	Count
Samsung Galaxy S5 (Sprint)	4.4.4	Movie search > As a valid user perform movie search	2
Samsung Galaxy S5 (AT&T)	4.4.2	Movie search > As a valid user perform movie search	1
Amazon Kindle Fire HD 7 (WiFi)	4.4.3	Movie search > As a valid user perform movie search	1

レポート - 原因調査

個々の問題を掘り下げて調査できる。

Summary > Amazon Kindle Fire HD 7 (WiFi) > Movie search > As a valid user perform movie search

Details Screenshots Performance Files

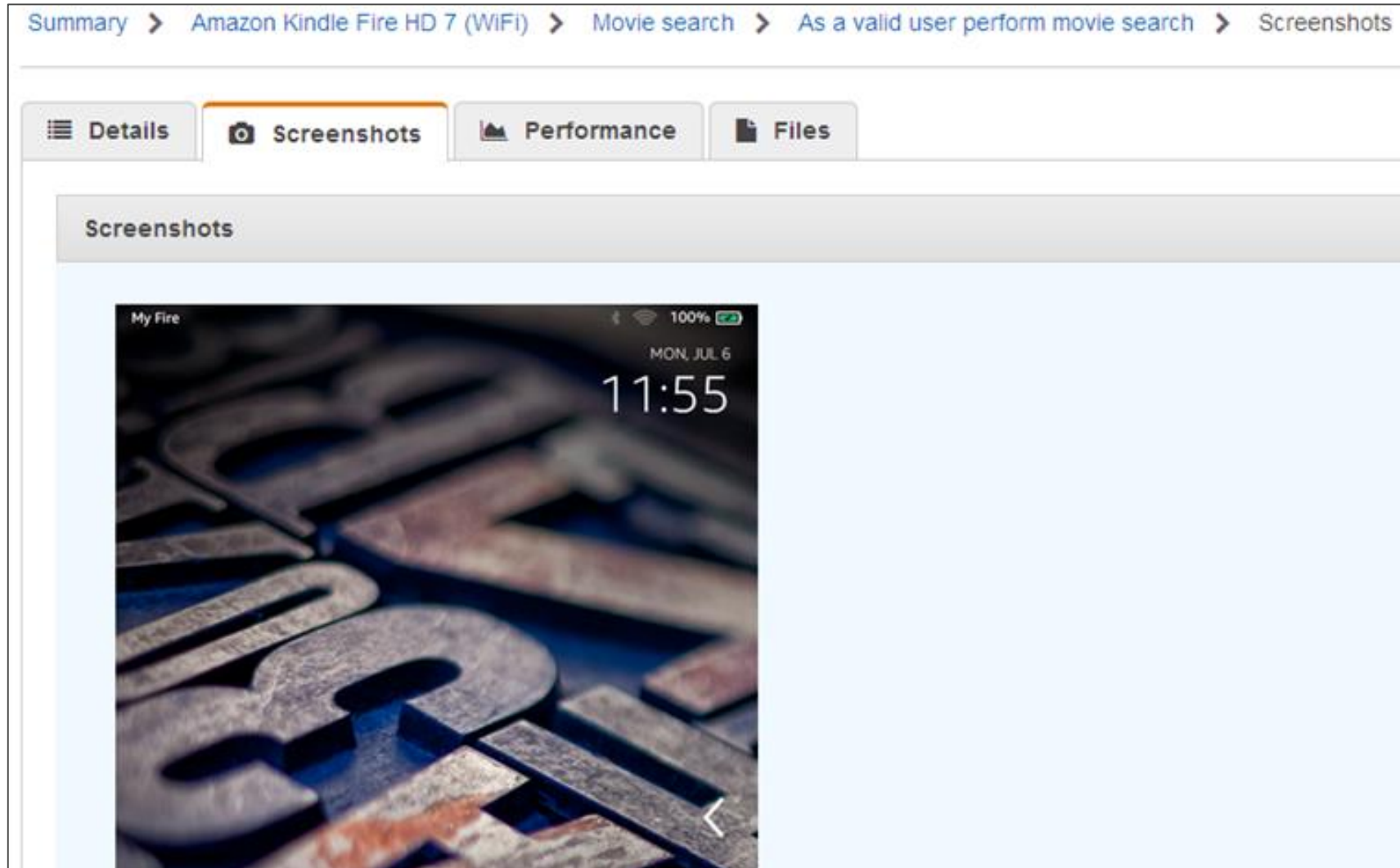
 As a valid user perform movie search
2015-07-06T11:52:0700

Download logs

Source	Time	PID	Level	Tag	Message
Harness ✕			warn ✕		
Harness	00:37.508	2210	Warned		As a valid user perform movie s...

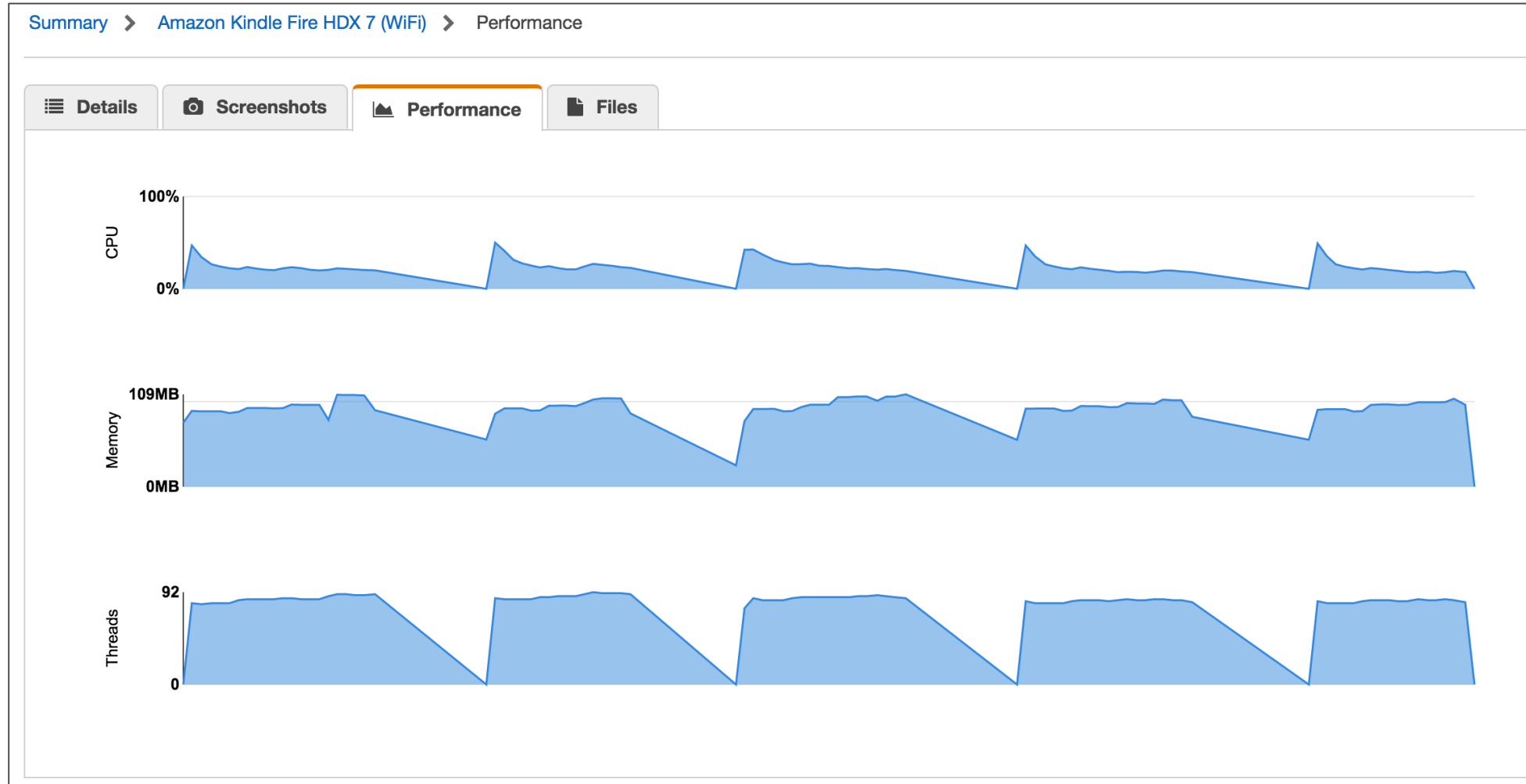
レポート – スクリーンショット

テストでキャプチャされたスクリーンショットを取得できる。



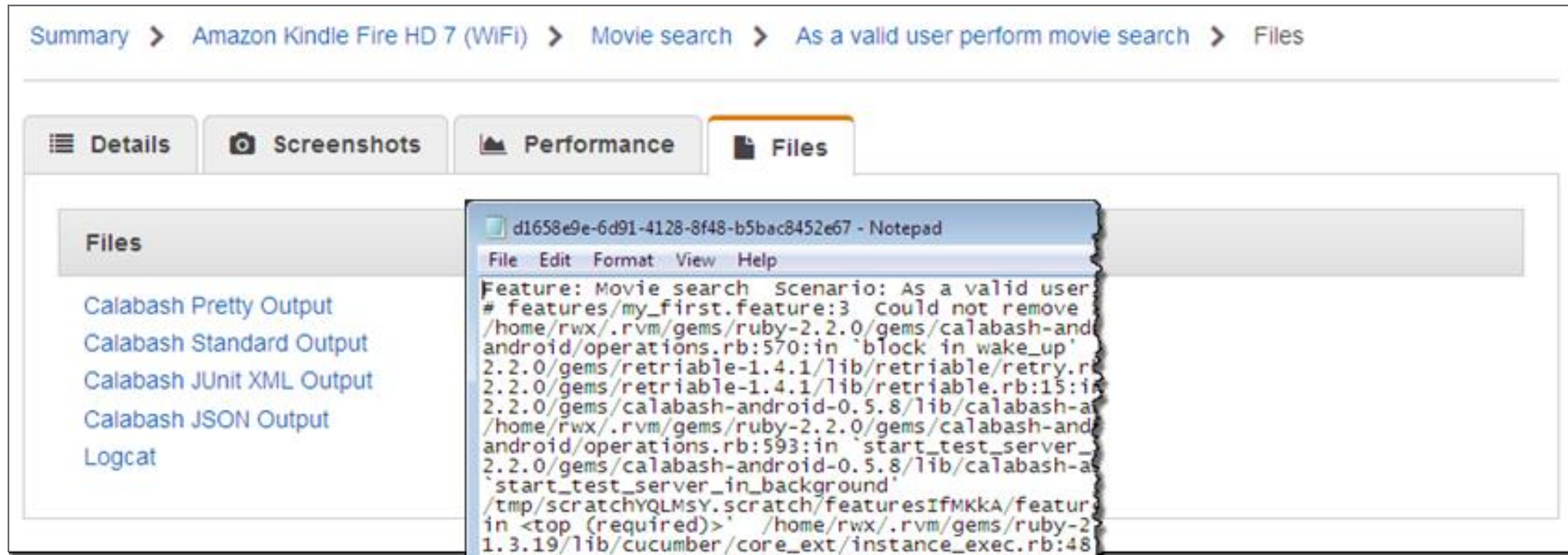
レポート - パフォーマンス

テストにおけるパフォーマンス（CPU、メモリ利用、スレッド数）を調査できる。



レポート - ログ

テストで作成されたログなどのファイルを取得できる。



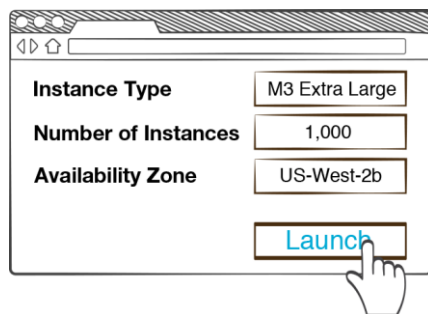
The screenshot displays the AWS CloudWatch console interface. The breadcrumb navigation at the top reads: Summary > Amazon Kindle Fire HD 7 (WiFi) > Movie search > As a valid user perform movie search > Files. Below this, there are four tabs: Details, Screenshots, Performance, and Files. The Files tab is selected. On the left side, under the 'Files' heading, there is a list of log files: Calabash Pretty Output, Calabash Standard Output, Calabash JUnit XML Output, Calabash JSON Output, and Logcat. A Notepad window is overlaid on the right side of the console, displaying the content of a log file. The Notepad window title is 'd1658e9e-6d91-4128-8f48-b5bac8452e67 - Notepad'. The log content includes the following text:

```
Feature: Movie search Scenario: As a valid user
# features/my_first.feature:3 Could not remove
/home/rwx/.rvm/gems/ruby-2.2.0/gems/calabash-and
android/operations.rb:570:in `block in wake_up'
2.2.0/gems/retriable-1.4.1/lib/retriable/retry.r
2.2.0/gems/retriable-1.4.1/lib/retriable.rb:15:i
2.2.0/gems/calabash-android-0.5.8/lib/calabash-a
/home/rwx/.rvm/gems/ruby-2.2.0/gems/calabash-and
android/operations.rb:593:in `start_test_server_
2.2.0/gems/calabash-android-0.5.8/lib/calabash-a
`start_test_server_in_background'
/tmp/scratchYQLMSY.scratch/featuresIfMKKA/featur
in <top (required)>` /home/rwx/.rvm/gems/ruby-2
1.3.19/lib/cucumber/core_ext/instance_exec.rb:48
/home/rwx/.rvm/gems/ruby-2.2.0/gems/cucumber-
1.3.19/lib/cucumber/core_ext/instance_exec.rb:48
/home/rwx/.rvm/gems/ruby-2.2.0/gems/cucumber-
1.3.19/lib/cucumber/core_ext/instance_exec.rb:69
/home/rwx/.rvm/gems/ruby-2.2.0/gems/cucumber-
1.3.19/lib/cucumber/core_ext/instance_exec.rb:36
/home/rwx/.rvm/gems/ruby-2.2.0/gems/cucumber-1.3
`invoke` /home/rwx/.rvm/gems/ruby-2.2.0/gems/cu
1.3.19/lib/cucumber/language_support/language_me
/home/rwx/.rvm/gems/ruby-2.2.0/gems/cucumber-
```

開発／テストのワークフローと統合

- AWS マネージメントコンソールから利用できる。
- Jenkins プラグインを利用して Jenkins CI サーバーから利用できる。
- その他ワークフローのためにフル機能の API を利用できる。
 - テストのセットアップ、テストの開始、テスト状況の取得
 - レポーティング情報のダウンロード

Management Console



Plugin



CLI



SDKs



Jenkins プラグイン

Jenkins CI サーバーと AWS Device Farm を機能的に連携するプラグイン

Status	Build Number	Pass/Warn/Skip/Fail/Error/Stop	Web Report
Completed	#19	12✔ 0▲ 1⊕ 1⊖ 1! 0■	Full Report
Completed	#18	9✔ 0▲ 1⊕ 1⊖ 1! 0■	Full Report
Completed	#17	12✔ 0▲ 1⊕ 1⊖ 1! 0■	Full Report
Completed	#16	12✔ 0▲ 1⊕ 1⊖ 1! 0■	Full Report
Completed	#15	11✔ 0▲ 1⊕ 2⊖ 1! 0■	Full Report

#	Time
#19	Jul 15, 2015 4:25 AM
#18	Jul 15, 2015 1:35 AM
#17	Jul 15, 2015 1:21 AM
#16	Jul 15, 2015 1:06 AM
#15	Jul 14, 2015 10:55 PM

- [Last build \(#19\), 41 min ago](#)
- [Last failed build \(#19\), 41 min ago](#)
- [Last unsuccessful build \(#19\), 41 min ago](#)

Project: jenkins
Device Pool: Top Devices
Application: hello-world.apk

Choose test to run:
 Built-in Fuzz
 Appium Java JUnit
 Appium Java TestNG
 Calabash

Features: hello-world-tests.zip
Tags:

Save Apply

AWS Device Farm Jenkins Plugin

<https://github.com/aws-labs/aws-device-farm-jenkins-plugin>



Jenkins プラグイン

テスト結果（スクリーンショット、ログなど）をダウンロードすることも可能。



The screenshot shows the Jenkins web interface. At the top left is the Jenkins logo and name. Below it is a breadcrumb trail: Jenkins > Hello World App > #19. On the left side, there is a navigation menu with the following items: Back to Project (green arrow), Status (magnifying glass), Changes (notepad), Console Output (terminal), Edit Build Information (notepad), Delete Build (red circle with slash), AWS Device Farm (purple cube), and Previous Build (green arrow). The main content area is titled "Artifacts of Hello World App #19". It features a folder icon next to the text "AWS Device Farm Results /" followed by a search input field and a green arrow. Below this, there is a list of device folders: Amazon Kindle Fire HDX 7 (WiFi), Motorola DROID Ultra (Verizon), Samsung Galaxy Note 4 (AT&T), Samsung Galaxy S5 (AT&T), and Samsung Galaxy Tab 4 10.1 Nook (WiFi). At the bottom right of the artifact list, there is a zip folder icon and the text "(all files in zip)".

セキュリティ



- AWS Device Farm とのすべての通信は SSL コネクション上で行われる。
- テストが終われば、インストールされたアプリはすべてアンインストールされる。
- 電話番号、追加アカウント、外部データを含むデバイス上のすべてのデータは削除される。
- AWS 上に一時保存されたアプリとファイルは自動的に 30 日経過すると削除される。
- テスト中、各デバイスはユーザーに専有される。
- デバイスは他のデバイスから独立しており、WiFi、Bluetooth、その他のワイヤレステクノロジーは他のデバイスにアクセスできない。
- 外部ネットワーク接続はインターネットに限定されており、他のデバイスへ接続できるネットワークはない。
- すべてのデバイスを利用できるように認証がとれるまでは、数個の試用デバイスの利用のみに限定される。（すべてのデバイスを利用するには要申請）


その他

- アップロードできるアプリは最大 4GB まで。
- 利用できるデバイス数に上限はないが、並列実行数は最大 5（緩和可）
- WiFi によるインターネット接続ができるが、キャリアのコネクションは持っておらず、電話の発信や SMS メッセージの送信はできない。
- フロントおよびリアのカメラで写真を撮影できるが、デバイスが設置されている環境に依存して暗くピンぼけした写真となる。
- Google Play サービスはインストールされておりサポートされているが、アクティブな Google アカウントは持っていない。
- 製造メーカーやキャリアによる最低限のアプリのみプリインストールされている。
- PhoneGap, Titanium, Xamarin, Unity, その他フレームワークで作成されたハイブリッドアプリもサポート。

アジェンダ

- 解決したい課題と背景
- AWS Device Farm の紹介
- **ユースケース**
- 利用料金
- まとめ
- Q&A

あらゆるモバイルアプリに向けたソリューションである

- 
- モバイルバンキング
 - オンライン証券
 - 保険
 - ヘルスケア
 - 小売
 - 旅行
 - メディア&エンタメ
 - ビデオ/テレビ
 - トレーニング
 - 教育
 - ゲーム
 - メッセージ
 - ソーシャル
 - 経費
 - 飲食
 - クーポン
 - 運送
 - フィットネス
 - 石油&ガス
 - エンタープライズ
 - 支払い
 - 政府
 - モバイル SDK
 - CRM

ユースケースごとの比較

エンタープライズ／大規模のビジネス

- すべてのモバイルビジネスに活用できるプラットフォームである。
- 自前でオートメーションソリューションを構築／運用しなくてよい。
- いままで利用したことのないようなデバイスも利用できる。

小／中規模のビジネス

- いままで所有できなかったデバイスを利用できる。
- 手作業ではリソースを限定していたが、自動化することで効率的にテストできるようになる。

SIer／コンサルティングパートナー

- 自社サービスを提供するために、プラットフォームを活用できる。
- ビジネスを横展開するために、再利用できる技術と資産を作ることができる。

自前のソリューションとの比較

自前のソリューション

初期投資／運用が高コスト

新しいデバイスの追加とOSバージョンのアップデートは一日仕事である

いつでも利用できるようにデバイスをオンラインに保たなければならない

並列実行、データ収集、レポートニングなど、オートメーション機構を構築することは大変

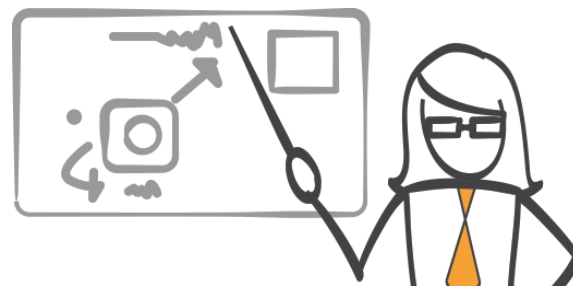
AWS Device Farm

従量課金により低コスト

ユーザーが利用しているデバイスとOSバージョンに遅れをとらない

AWS データセンターにホスティングされたデバイスの高い信頼性と可用性

高品質なアプリを提供することに集中できる



導入に際して

- すでにテストオートメーションを利用もしくは計画している。
- もっと多くのデバイスを利用したい。
- 自前でデバイスやインフラを管理したくない。

→すぐに AWS Device Farm に置き換え可能

- まだテストオートメーションを利用していない。
- テストスクリプトを作成するリソースが足りない。
- テストを実施するリソースが足りない。

→まずは Fuzz テストから



アジェンダ

- 解決したい課題と背景
- AWS Device Farm の紹介
- ユースケース
- **利用料金**
- まとめ
- Q&A

利用料金

従量課金

- テストを実行するのに必要となった Device Minutes によってつまり、デバイス数×時間(分) によって計測される。
 - 無料枠は : 250 Device Minutes
 - 以後は : 1 Device Minute あたり 0.17 ドル



定額オプション

- 1 デバイスあたり月額 250 ドル
 - 定額でデバイスを利用できる。
 - サブスクリプションを変更しなくても、いつでも数多くの Android デバイスから必要なデバイスを選択できる。

アジェンダ

- 解決したい課題と背景
- AWS Device Farm の紹介
- ユースケース
- 利用料金
- **まとめ**
- Q&A

AWS Device Farm のポイント

- クラウドにホスティングされた数多くの Android, Fire OS の**実デバイス**でテストを実施できる。
- **複数のフレームワーク**を利用できる。
- テストスクリプト不要のビルトイン **Fuzz テスト**を利用できる。
- テストを**並列実行**でき、数分で結果を得られる
- 実世界のシナリオを再現するために、**テスト環境**を設定できる（言語、ロケーション、デバイス状態など）
- **テスト結果**をすぐに取得できる（スクリーンショット、デバイスログ、パフォーマンスデータなど）
- **Jenkins** など継続的インテグレーション（CI）や他ワークフローの一部としてテストを実施できる。

アジェンダ

- 解決したい課題と背景
- AWS Device Farm の紹介
- ユースケース
- 利用料金
- まとめ
- **Q&A**

Q&A



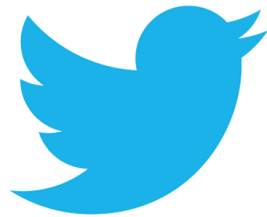
Webinar 資料の配置場所

AWS クラウドサービス活用資料集

- <http://aws.amazon.com/jp/aws-jp-introduction/>

プロダクト別：				
Amazon S3		AWSマイスターシリーズ Re:Generate Amazon Simple Storage Service (S3)	Slideshare	PDF
Amazon Glacier		AWSマイスターシリーズ Reloaded Amazon Glacier Amazon Glacierのご紹介 機能編	Slideshare (Reloaded) Slideshare (機能編)	PDF (Reloaded) PDF (機能編)
Amazon Route 53		AWSマイスターシリーズ Re:Generate	Slideshare	PDF

公式 Twitter / Facebook AWS の最新情報をお届けします



@awscloud_jp



検索

もしくは

<http://on.fb.me/1vR8yWm>

最新技術情報、イベント情報、お役立ち情報、お得なキャンペーン情報などを
日々更新しています！



WE'RE HIRING!

<http://aws.amazon.com/jp/careers/>

次回の AWS Black Belt Tech Webinar は、

7月22日 18:00～

～ CLI & AWS Tools for Windows PowerShell ～



ご参加ありがとうございました。