

AWS Device Farm Mobile Testing

What Is AWS Device Farm?

Device Farm is an app testing service that enables you to test and interact with your Android, iOS, and Web apps on real, physical phones and tablets that are hosted by Amazon Web Services (AWS). There are two main ways to use Device Farm:

- Automated testing of apps using a variety of available testing frameworks
- Remote access to devices onto which you can load, run and interact with apps in real-time

Automated App Testing

Device Farm allows you to upload your tests or use built-in, script-free compatibility tests. Because testing is automatically performed in parallel, tests on multiple devices begin in minutes.

A test report containing high-level results, low-level logs, pixel-to-pixel screenshots, and performance data is updated as tests are completed.

Device Farm supports testing of native and hybrid Android, iOS, and Fire OS apps, including those created with PhoneGap, Titanium, Xamarin, Unity, and other frameworks. It supports remote access to Android apps for interactive testing.

Supported Test Types and Built-in Tests

Device Farm currently provides support for the following test types:

For Android:

- Appium Java JUnit
- Appium Java TestNG
- Appium Python
- Calabash
- Instrumentation (JUnit, Espresso, Robotium, or any instrumentation-based tests)

- UI Automator
- Explorer

For iOS:

- Appium Java JUnit
- Appium Java TestNG
- Appium Python
- Calabash
- UI Automation
- XCTest (including KIF)
- XCTest UI

For Web Apps:

- Appium Java JUnit
- Appium Java TestNG
- Appium Python

If you do not have your tests, you can use a built-in fuzz test. For more information, see Built-in: Fuzz (Android and iOS).

Remote Access Interaction

Remote access allows you to swipe, gesture, and interact with a device through your web browser in real-time. There are several situations where real-time interaction with a device is useful. For example, customer service representatives can guide customers through how to use or set up their devices. They can also walk customers through how to use apps running on a specific device. You can install apps on a device running in a remote-access session and then reproduce customer problems or reported bugs.

During a remote access session, Device Farm collects details about actions that take place as you interact with the device. Logs with these details and video capture of the session are produced at the end of the session for your review.

Initially, a limited number of Android and Fire OS devices are supported for remote access. However, the list of devices will grow during the beta period and as new devices enter the market.

Test Types in AWS Device Farm

AWS Device Farm provides many different built-in and custom test types for Android, iOS, and Web applications. Built-in tests enable you to test your apps without writing scripts. Custom tests allow you to test specific flows and business logic within your app. For more information, see [Working with Test Types in AWS Device Farm](#).

Android Test Types

Device Farm provides the following built-in and custom test types for Android devices.

- Built-in: Explorer (Android)
- Built-in: Fuzz (Android and iOS)
- Appium Java JUnit
- Appium Java TestNG
- Appium Python
- Calabash
- Instrumentation
- UI Automator

iOS Test Types

Device Farm provides the following built-in and custom test types for iOS devices.

- Built-in: Fuzz (Android and iOS)
- Appium Java JUnit
- Appium Java TestNG
- Appium Python
- Calabash
- UI Automation
- XCTest
- XCTest UI

Web Application Test Types

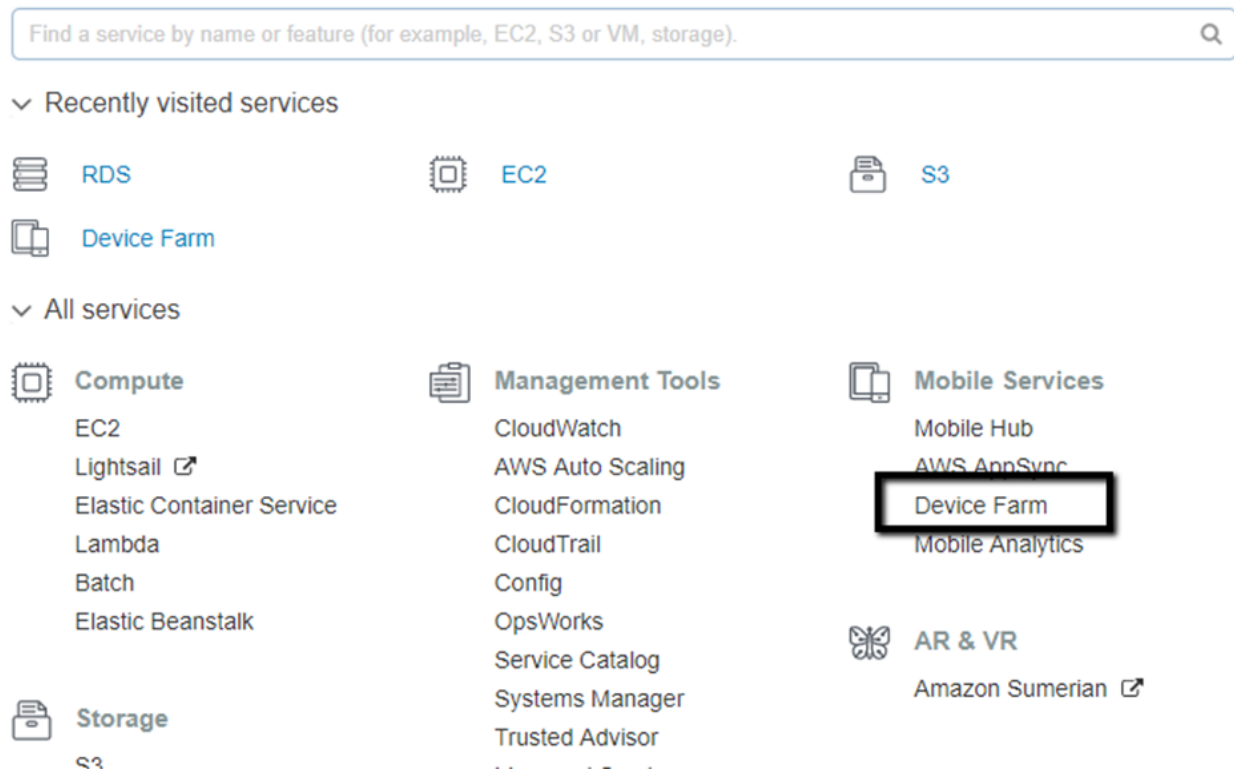
Device Farm provides the following custom test types for Web applications.

- Appium Java JUnit
- Appium Java TestNG
- Appium Python

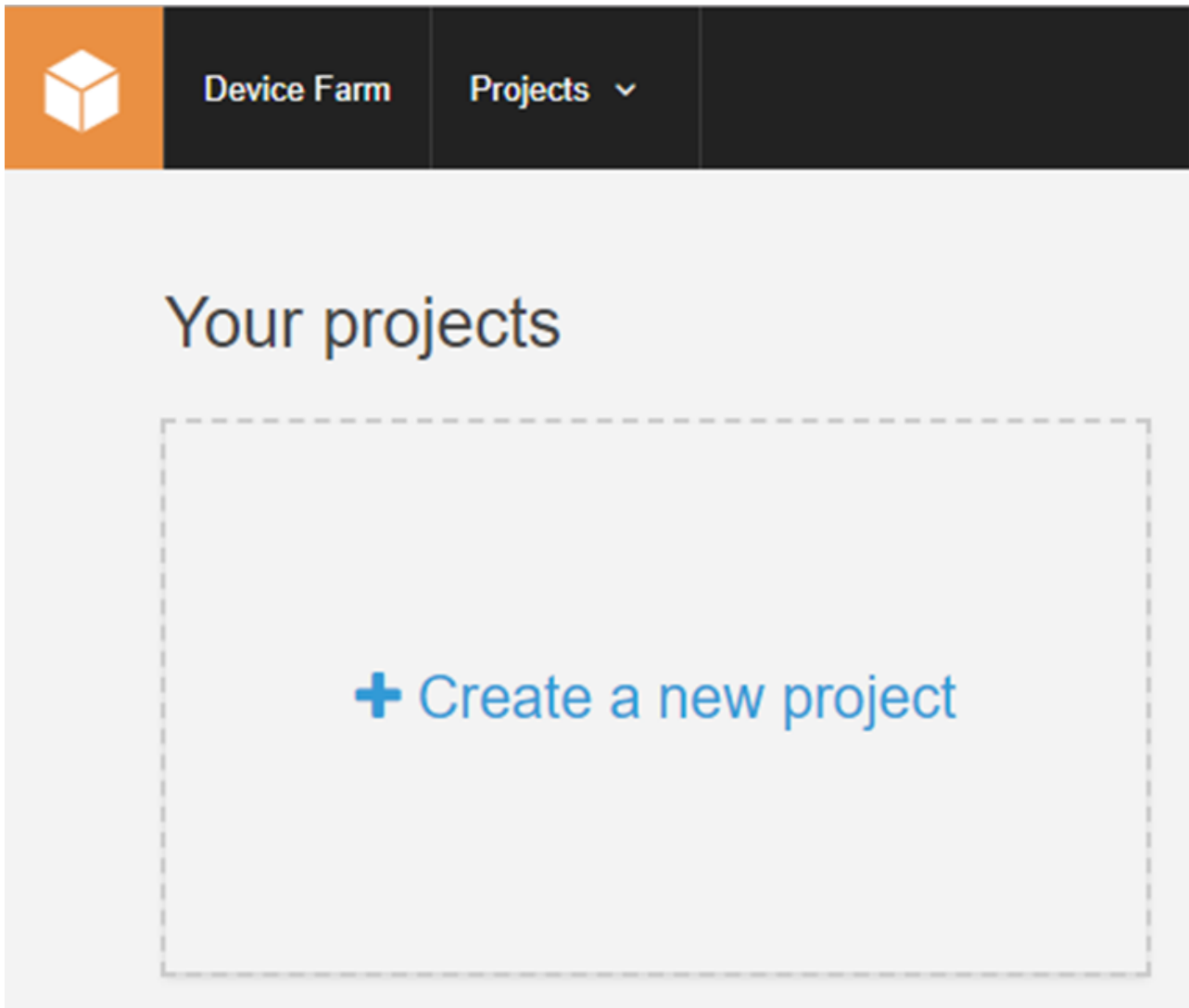
Working with AWS Device Farm

1. Log in to the AWS console using valid credentials and the user should be able to see the Home page of AWS as shown below

AWS services



2. Click Device Farm
3. The user would be navigated to the Projects page as shown below



4. Click New Project

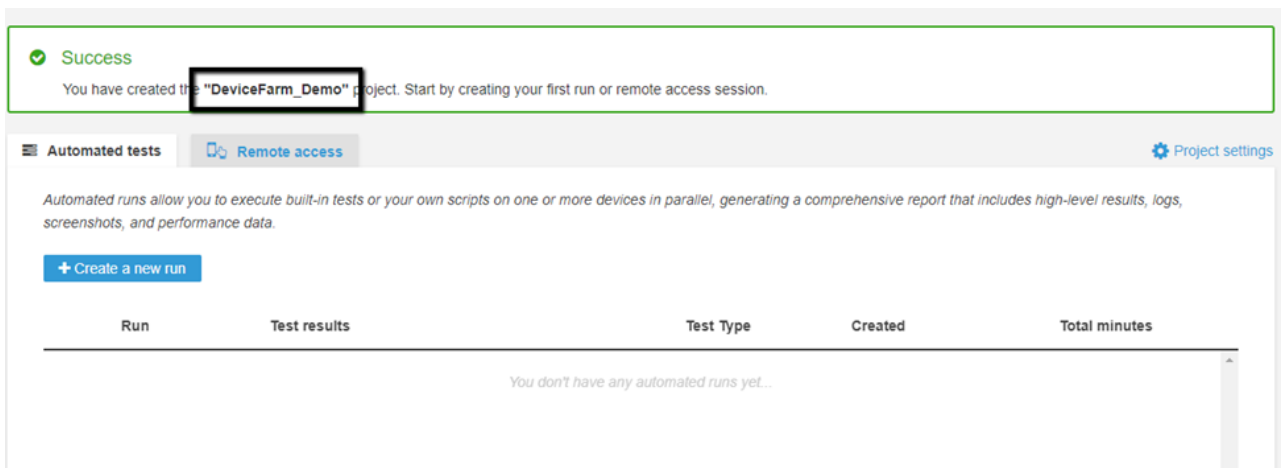
Create project ×

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

Project name

Cancel Create project

5. Enter the Project Name and click Create Project and the user will be navigated to the Run page



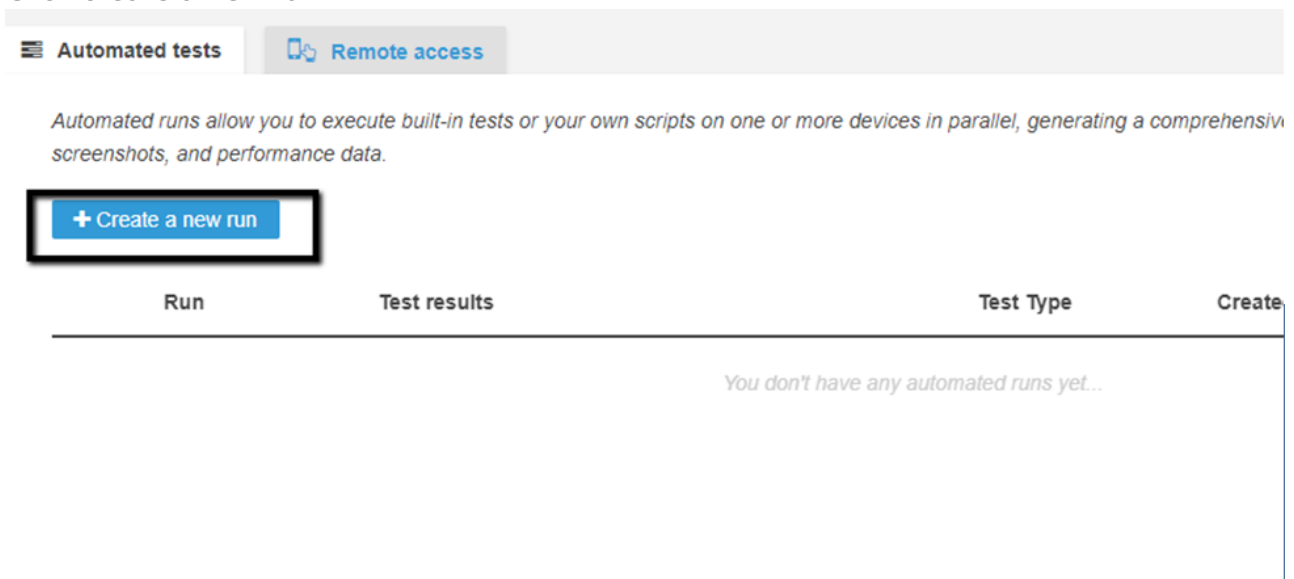
6. Now User has 2 options to choose from

- Automated Tests
- Remote access

Automated Tests

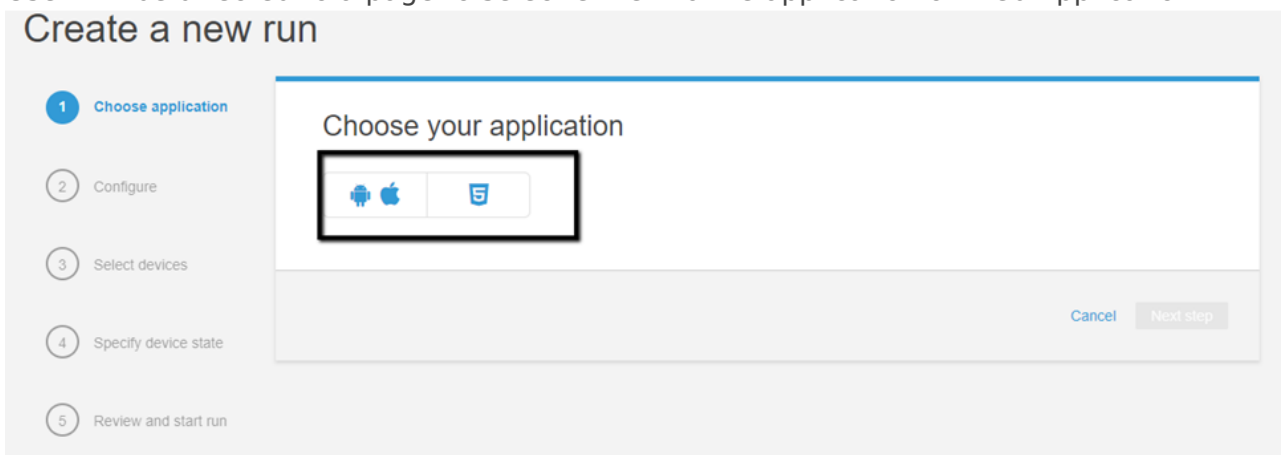
Step by step instructions on how to run automated tests in Device Farm

1. Click create a new run



2. User will be directed to a page to select either Native application or Web Application

Create a new run



3. Select Native application and upload the APK or IPA file depending on the OS

Choose your application



 Upload Or drop your file here 

Run name 

4. The user will upload the file

✓ Choose your application



Jira Cloud Official mobile app for Jira Software_v5.0.113_apkpure.com.apk 

Package	com.atlassian.android.jira.core
Activity	com.atlassian.android.jira.core.app.LauncherActivity
Minimum SDK	21
Native Code	arm64-v8a,armeabi,armeabi-v7a,mips,mips64,x86,x86_64
Screens	small,normal,large,xlarge
Target SDK	27
Version Code	5000113
Version Name	5.0.113

Run name 

Device farm will give us details about the package and the activity name. It also gives information on the Target SDK, Version code, and Version name.

The user clicks on Next Step

5. The user will be presented with Configure page where the user has to select the test type

✓ Configure

Setup test framework

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

○ Built-in: Explorer

No tests? No problem. We'll explore your app by analyzing each view with no scripts required.

○ Built-in: Fuzz

Username



○ Appium Java JUnit

○ Appium Java TestNG

Password



○ Appium Python

○ Calabash

○ Instrumentation

○ UI Automator

Other Configurations

☒ Enable Video Recording ⓘ

☒ Enable App Performance Data Capture ⓘ

AWS provides 2 built-in options for testing i.e Explorer and Fuzz. Rest needs to be developed by the User.

I will be selecting the Explorer option to test the application. Once that is selected click the Next step

6. The user will be navigated to the Select Devices page

This page will inform the user of the device which is compatible with the apk file or ipa file. Users can also create their device pool using the “Create a new device Pool” option

✓ Select devices

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

Device pool

Top Devices



Create a new device pool

⚠ Compatibility issues

Your app is incompatible with **3 out of 5** devices in the selected pool. Incompatible devices will be ignored at run time.

Device	OS	Reason
✓ Samsung Galaxy S6 (Verizon)	6.0.1	
✓ ASUS Nexus 7 - 2nd Gen (WiFi)	5.0.1	
⚠ Samsung Galaxy S5 (T-Mobile)	4.4.2	Android application requir...
⚠ LG G Pad 7.0" (AT&T)	4.4.2	Android application requir...
⚠ Samsung Galaxy Tab 4 10.1" (WiFi)	4.4.2	Android application requir...

Click Next

7. The user now will be presented to the Device state page

✓ Specify device state

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

Add extra data

Upload

Or drop your file here

Install other apps

Upload

Or drop your file here

or

Select a recent upload

Set radio states

- ☒ WiFi
- ☒ Bluetooth
- ☒ GPS
- ☒ NFC

Device location

47.6204

-122.3491

Paths to your files on the host machine and device

User can change settings like wifi, Blue-tooth, etc.

Click Next

8. Review and start the Run page

The user has to select the minutes carefully since if the time given by the user is less than the actual run then only partial results will be available to the user

Execution timeout

Limit the maximum number of device minutes that your run can use by setting a timeout on each device. If execution exceeds your timeout, execution on that device will be forcibly stopped. Partial results will be available if possible.

8

MAXIMUM MINUTES PER DEVICE

×

2

DEVICES

=

16

MAXIMUM DEVICE MINUTES FOR THIS RUN

5m

77m

150m

You should choose a value that is greater than the anticipated duration of your tests. For example, if your tests take 20 minutes to complete you should choose a timeout of 30 minutes. Need to run tests longer than 150 minutes? Please [contact us](#).

Application

[Edit](#)

Name Jira Cloud Official mobile app for Jira Software_v5.0.113_apkpure.com.apk

- Once the above setting is done user clicks on confirm and runs the button
9. The user will be presented to the Run page with a scheduled test in Progress

Automated tests

Remote access

Project settings

Automated runs allow you to execute built-in tests or your own scripts on one or more devices in parallel, generating a comprehensive report that includes high-level results, logs, screenshots, and performance data.

+ Create a new run

Run	Test results	Test Type	Created	Total minutes
 Jira Cloud Offi...		Built-In: Explorer	2018-04-19T17:45+0530	

Automated tests

Remote access

Project settings

Automated runs allow you to execute built-in tests or your own scripts on one or more devices in parallel, generating a comprehensive report that includes high-level results, logs, screenshots, and performance data.

+ Create a new run

Run	Test results	Test Type	Created	Total minutes
 Sample_App.apk		Built-In: Fuzz	2018-04-03T21:52+0530	00:04:45

Once the user clicks on the completed test, it will be navigated to the results page

 Jira Cloud Official mobile app for Jira Software_v5.0.113_apkpure.com.apk

Scheduled at: Apr 19, 2018 5:45:42 PM

Summary > Samsung Galaxy S6 (Verizon)

Video

Suites

Logs

Performance

Files

Screenshots

Video

Download Video

Where the user can check the video recording, Suites that were run, screenshots, and also the logs needed for debugging.

Remote Access

This feature allows the user to interact with the actual device which is launched in AWS

Steps to Start a new session are given below

1. Click Start new session button on the Remote access page. Users will be presented with a page to select the device name, Platform, and OS
2. Select the device and click the “Confirm and Start Run” button

Choose a device

Select a device for an interactive session. Interested in unlimited, unmetered testing? [Learn more](#) ▾.

Name ▲	Platform	OS	Form factor
samsung Galaxy S6			
☒ Samsung Galaxy S6 (Verizon)	Android	5.0.2	Phone

3. User will be presented with a page to indicate to the user that the device is being initialized in the back-end

Device requested

Please wait while we find an available device.

Waiting for **Samsung Galaxy S6 (Verizon)**



[Cancel request](#)

If you would like to keep working while you wait for your session to start, you can continue in a [new tab](#) [↗](#).

4. Device will be launched and users can perform actions as they would on a physical device

Samsung Galaxy S6 (Verizon)

Time left: 02:29:20

Stop session



Notice about sensitive data

AWS Device Farm captures video and logs of activity taking place during your Remote Access session. It is recommended that you avoid providing or entering sensitive information such as account info (E.g., Google Account, Apple ID), personal information, and other security-sensitive details during your remote access session.

Device information

Name: Samsung Galaxy S6 (Verizon)
OS: 5.0.2
Manufacturer: Samsung
Model: 5.0.2
Chipset: arm64-v8a 1500MHz
Memory: 3200000000
Heap: 512000000
Display: 1440 x 2560

Last updated: February 16, 2024

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #4

Created 2026-07-31 15:08:54 UTC by Admin

Updated 2026-10-14 13:31:12 UTC by AI_Mananger