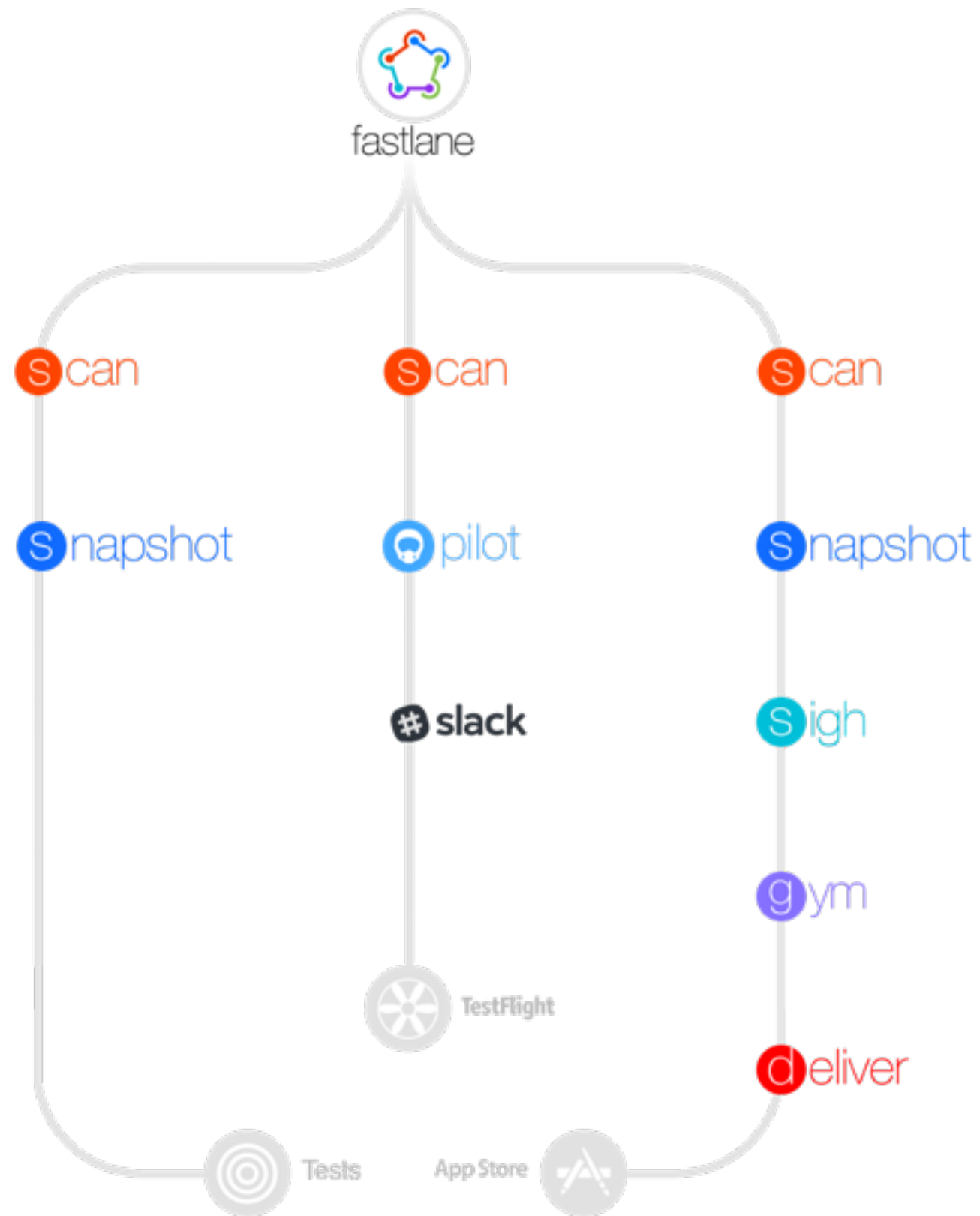


# We ❤️ Swift



@somkiat.cc



<https://fastlane.tools/>



Upload screenshots, metadata and your app to the App Store using a single command.



Automate taking localized screenshots of your iOS app on every device



Quickly put your screenshots into the right device frames



Automatically generate and renew your push notification profiles



Because you would rather spend your time building stuff than fighting provisioning



Create new iOS apps on iTunes Connect and Developer Portal using the command line



Automatically create and maintain iOS code signing certificates



The easiest way to run tests of your iOS and Mac app



Building your iOS app has never been easier



Easily sync your certificates and profiles across your team using Git



<https://fastlane.tools/>

# Software requirement

- Ruby 2.0+
- Xcode CLI
- Paid Apple Developer Account

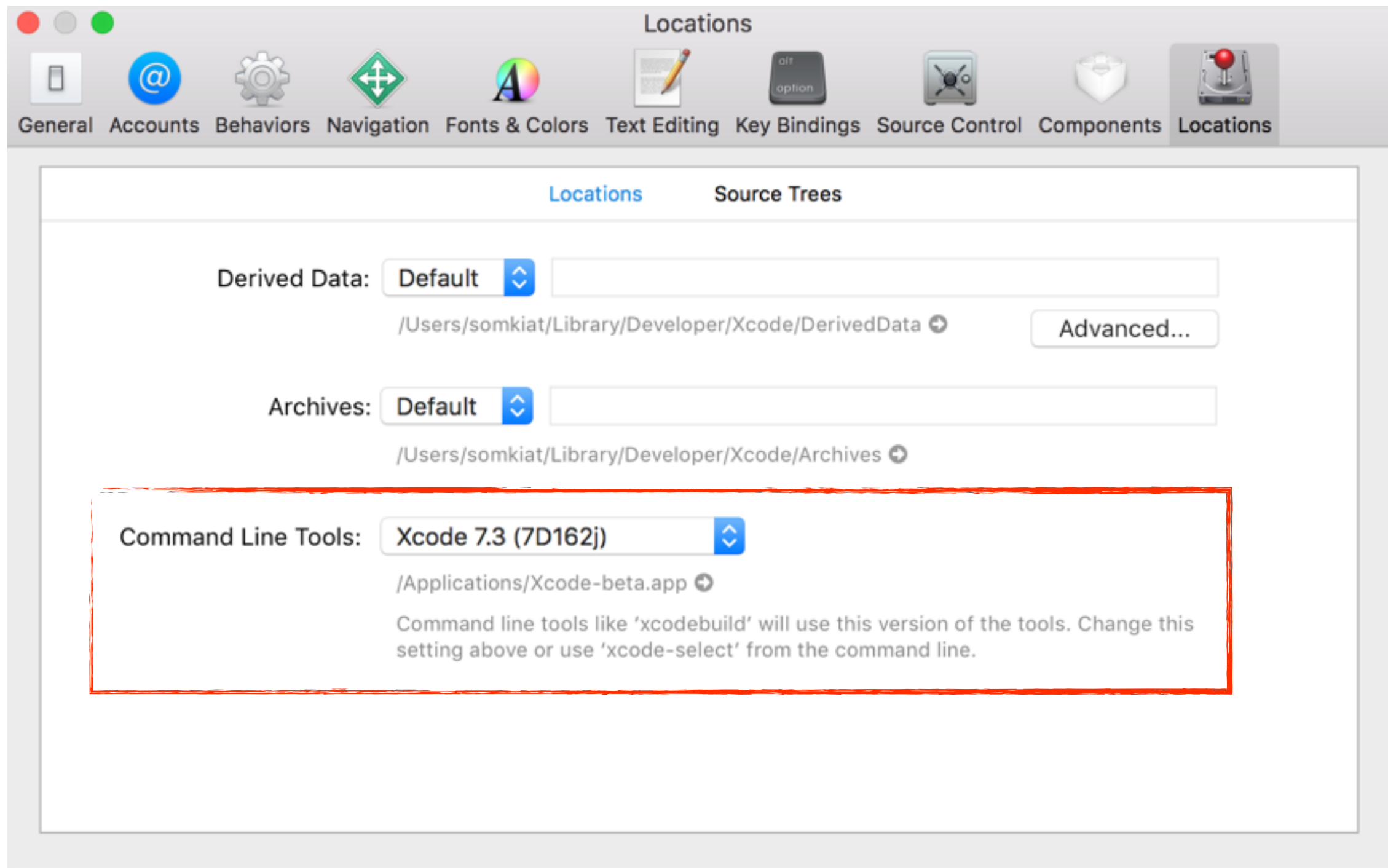
# Check Ruby 2.0+

```
$ruby -v
```

# Check Xcode CLI

```
$xcode-select --install
```

# Check Xcode CLI



# Installation

```
$sudo gem install fastlane
```



# Let's start

\$fastlane init

```
[00:03:17]: Writing to 'fastlane/metadata/secondary_second_sub_category.txt'
[00:03:17]: Successfully created new configuration files.
[00:03:17]: Downloading all existing screenshots...
[00:03:18]: Successfully downloaded all existing screenshots
[00:03:18]: Successfully created new Deliverfile at path 'fastlane/Deliverfile'
[00:03:18]: xcrun xcodebuild -list -project './01-Calculator.xcodeproj'
[00:03:19]: 'snapshot' not enabled.
[00:03:19]: 'cocoapods' not enabled.
[00:03:19]: 'carthage' not enabled.
[00:03:19]: Created new file './fastlane/Fastfile'. Edit it to manage your own deployment lanes.
[00:03:19]: fastlane will send the number of errors for each action to
[00:03:19]: https://github.com/fastlane/enhancer to detect integration issues
[00:03:19]: No sensitive/private information will be uploaded
[00:03:19]: You can disable this by adding `opt_out_usage` to your Fastfile
[00:03:19]: Successfully finished setting up fastlane
```



<https://github.com/fastlane/fastlane/tree/master/sCan>



ทำการ run test ได้ง่ายขึ้น

# Run scan

## \$scan

```
[00:41:05]: ▶ All tests
[00:41:05]: ▶ Test Suite 01-CalculatorUITests.xctest started
[00:41:05]: ▶ _1-CalculatorUITests
[00:41:08]: ▶ ✓ testExample (3.599 seconds)
[00:41:08]: ▶ Executed 1 test, with 0 failures (0 unexpecte
[00:41:08]: ▶
[00:41:08]: ▶ Test Succeeded
```

Test Results	
Number of tests	3
Number of failures	0

```
[00:41:09]: Successfully generated report at '/Users/somkiat/d
ulator-unit-test/test_output/report.html'
[00:41:10]: Successfully generated report at '/Users/somkiat/d
ulator-unit-test/test_output/report.junit'
```

# Run scan with device

```
$scan --device "iPhone 6s (9.3)"
```

# List of devices

## \$instruments -s

### Known Devices:

```
MacBook Pro [13CA0FB5-F6C7-5FC4-B3CF-E0572816CA3B]
Apple TV 1080p (9.2) [C6576318-0B48-4002-A4D5-C697299BDB2B] (Simulator)
iPad 2 (9.3) [C61DBED4-F796-4793-A680-E30A29B1B5DE] (Simulator)
iPad Air (9.3) [5990D523-C34C-496D-9FE2-47093B6839FE] (Simulator)
iPad Air 2 (9.3) [16ED67C3-FDA3-48EA-81DD-5CCA100F0897] (Simulator)
iPad Pro (9.3) [27938C50-FE02-410F-A497-C04C20CC20C5] (Simulator)
iPad Retina (9.3) [129D2D6A-E4A0-4960-8D1A-10490123DE71] (Simulator)
iPhone 4s (9.3) [3F5E47A3-0089-4F9A-8836-FD20EB74C7B4] (Simulator)
iPhone 5 (9.3) [04CFFD66-03B8-4DC2-B9B6-E9737BA09D21] (Simulator)
iPhone 5s (9.3) [F20271AC-8920-4966-82B5-D1B35F776EE1] (Simulator)
iPhone 6 (9.3) [45283668-558D-407F-9604-266367396B82] (Simulator)
iPhone 6 Plus (9.3) [6C67B0E9-8A4A-481E-BFB9-AD3D6BC74260] (Simulator)
iPhone 6s (9.3) [EC2B7BB4-8B1F-485D-8D98-A86A0384F5C2] (Simulator)
iPhone 6s (9.3) + Apple Watch - 38mm (2.2) [0885C8DF-0CC6-4B2F-AADB-EA1018EB3197]
iPhone 6s Plus (9.3) [46FCF730-E4E7-4922-AA74-2F85FBDD78A3] (Simulator)
iPhone 6s Plus (9.3) + Apple Watch - 42mm (2.2) [D5898E61-AF78-42E3-8754-6089CDB7]
```



# See report file

## Test Results

3 tests

All

Failing

Passing

### \_1\_CalculatorTests.\_1\_CalculatorTests

0.001s testExample

0.343s testPerformanceExample

### \_1\_CalculatorUITests.\_1\_CalculatorUITests

3.599s testExample

Report generated with [xcpretty](#)

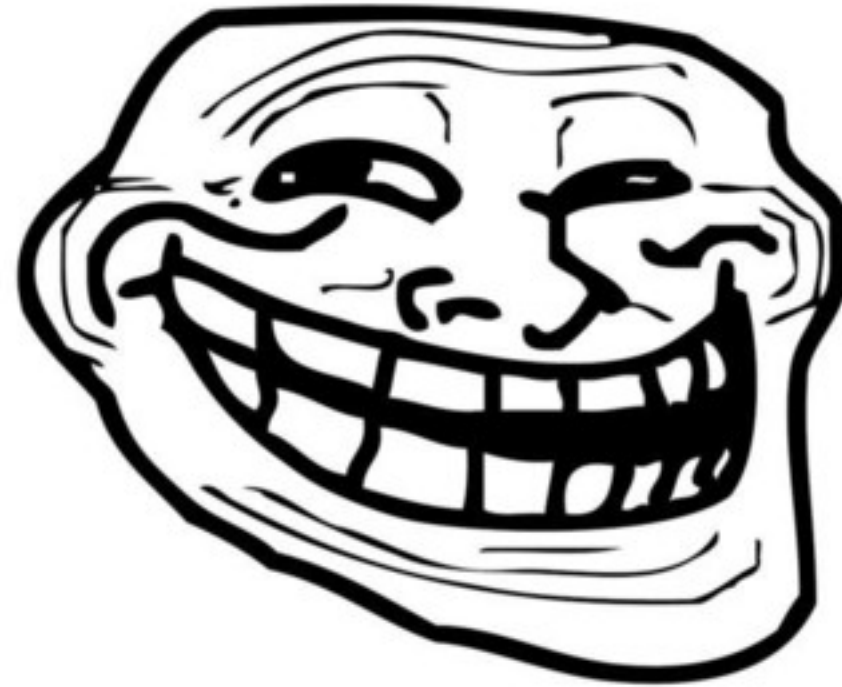
# test





<https://github.com/fastlane/fastlane/tree/master/snapshot>

# Xcode 7.3



**problem?**

# Init snapshot in project

## \$snapshot init

```
Successfully created SnapshotHelper.swift './SnapshotHelper.swift'  
Successfully created new Snapfile at './Snapfile'
```

---

Open your Xcode project and make sure to do the following:

- 1) Add the ./fastlane/SnapshotHelper.swift to your UI Test target  
You can move the file anywhere you want
- 2) Call `setupSnapshot(app)` when launching your app

```
let app = XCUIApplication()  
setupSnapshot(app)  
app.launch()
```

- 3) Add `snapshot("0Launch")` to wherever you want to create the screenshots

More information on GitHub: <https://github.com/fastlane/snapshot>

# Modify UI Test

```
override func setUp() {  
    super.setUp()  
  
    continueAfterFailure = false  
    let app = XCUIApplication()  
    setupSnapshot(app)  
    app.launch()  
}
```

# Modify UI Test

```
func testExample() {  
    //Arrange  
    snapshot("01CalculateScreen")  
  
    let app = XCUIApplication()  
    let first = app.textFields["first"]  
    first.tap()  
    first.tap()  
    first.typeText("2")  
    let second = app.textFields["second"]  
    second.tap()  
    second.typeText("5")  
  
    snapshot("02CalculateScreen")  
  
    //Act  
    app.buttons["calculate"].tap()  
  
    snapshot("03CalculateScreen")  
  
    //Assert  
    let result = app.staticTexts["result"].label  
    XCTAssertEqual(result, "7")  
}
```

# Run snapshot

\$snapshot

# cert sigh

<https://github.com/fastlane/fastlane/tree/master/cert>

<https://github.com/fastlane/fastlane/tree/master/sigh>

# Run snapshot

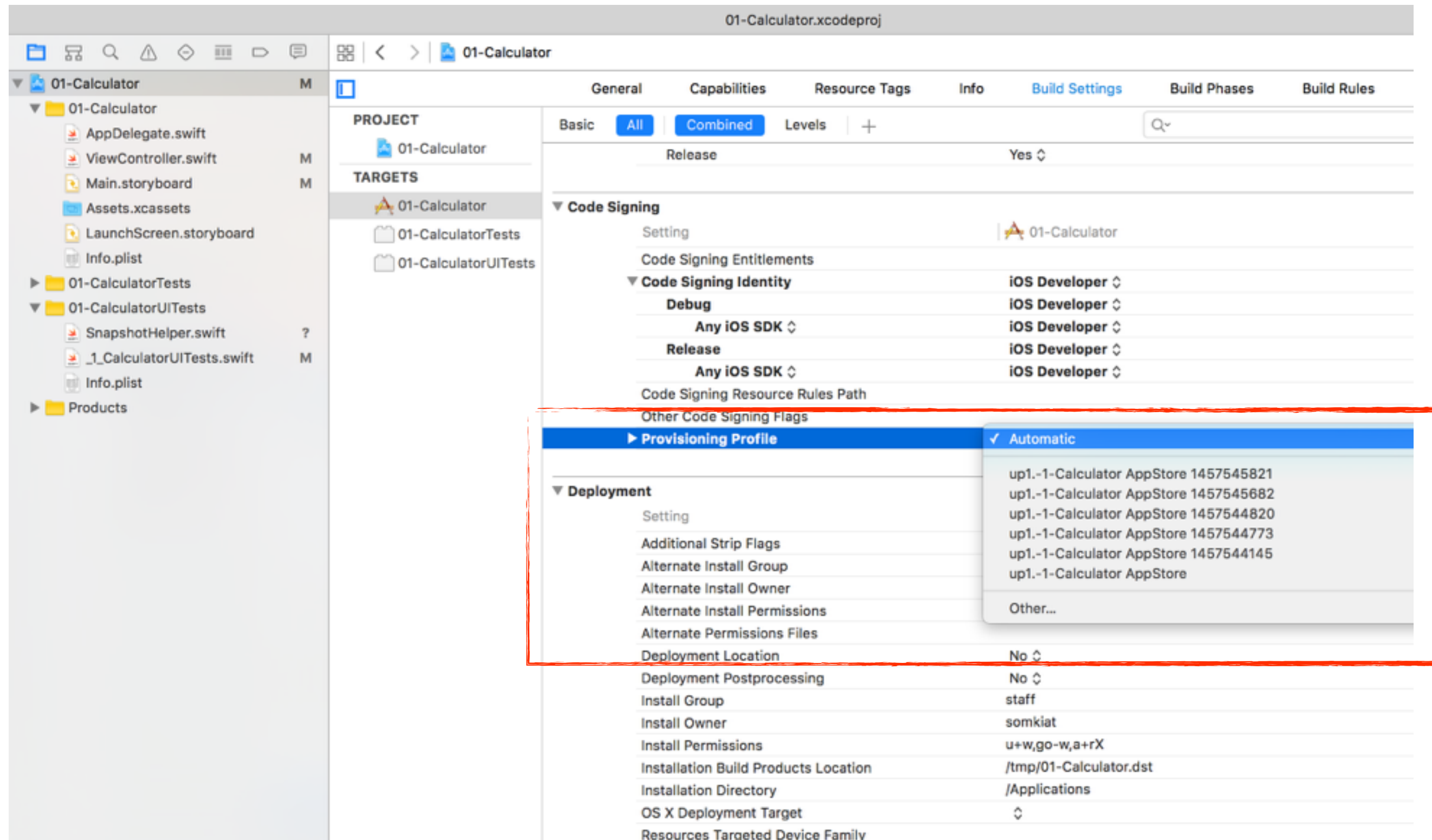
## \$cert & sigh



# Provisioning

```
desc "Creating a code signing certificate"
lane :provision do
  produce
  cert
  sigh(force: true)
end
```

# Provisioning profile





<https://github.com/fastlane/fastlane/tree/master/gym>

# Run gym

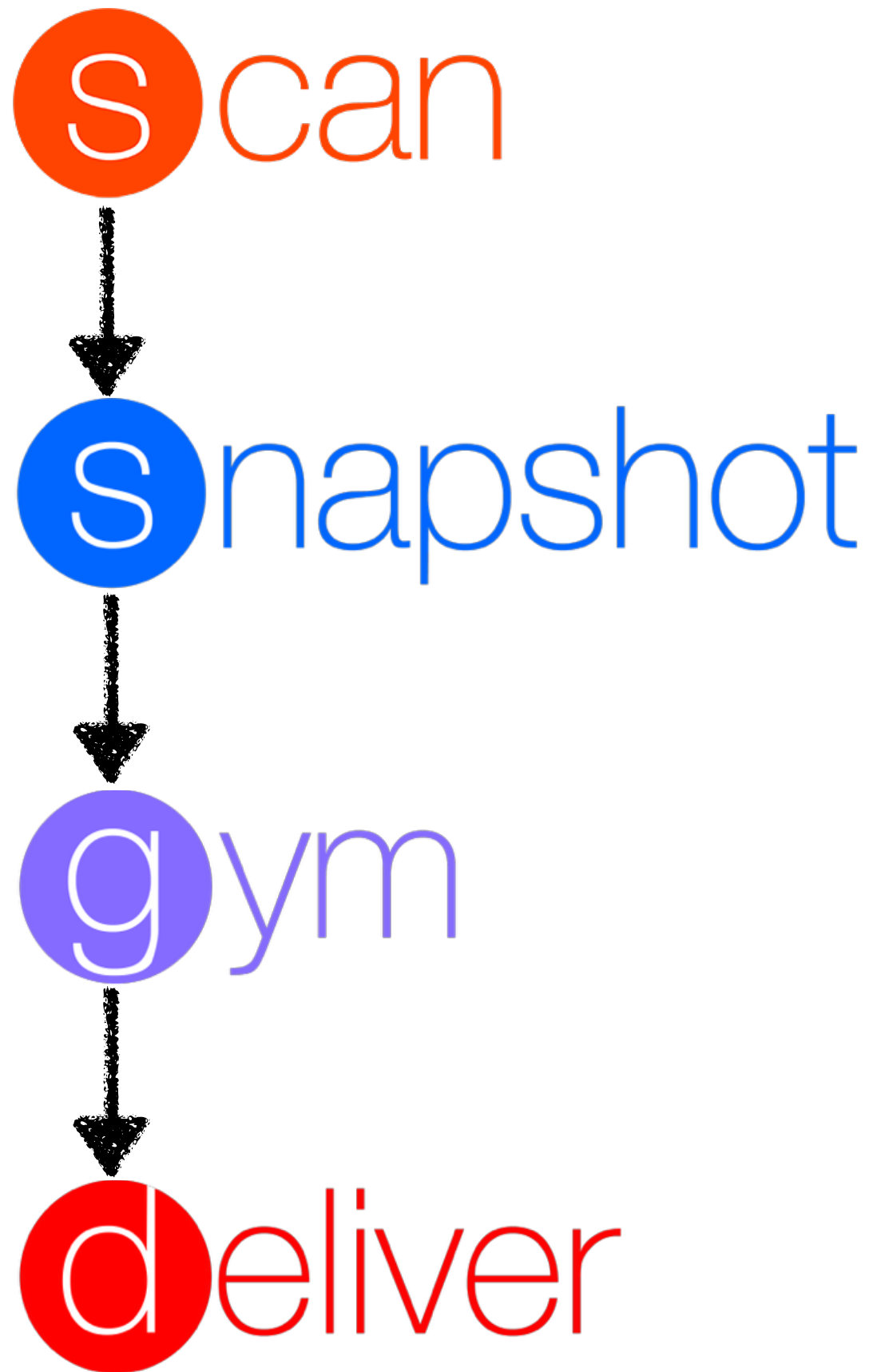
## \$gym



<https://github.com/fastlane/fastlane/tree/master/deliver>

# Run deliver

## \$deliver



# Edit Fastfile

```
desc "Runs all the tests"
lane :test do
  scan
end

desc "Take screenshots"
lane :screenshot do
  snapshot
end
```



# Edit Fastfile

```
desc "Creating a code signing certificate and provisioning profile"
lane :provision do
  produce
  cert
  sigh(force: true)
end

desc "Create ipa"
lane :build do
  gym
end
```

# Edit Fastfile

```
desc "Upload to App Store"  
lane :upload do  
  deliver  
end
```

# Edit Snapfile

```
devices([  
  .... "iPhone 6"  
])  
  
languages([  
  .. "en-US"  
])
```

# Run Fastlane

**\$fastlane test**

**\$fastlane screenshot**

**\$fastlane provision**

**\$fastlane build**

**\$fastlane upload**