Swift on Linux

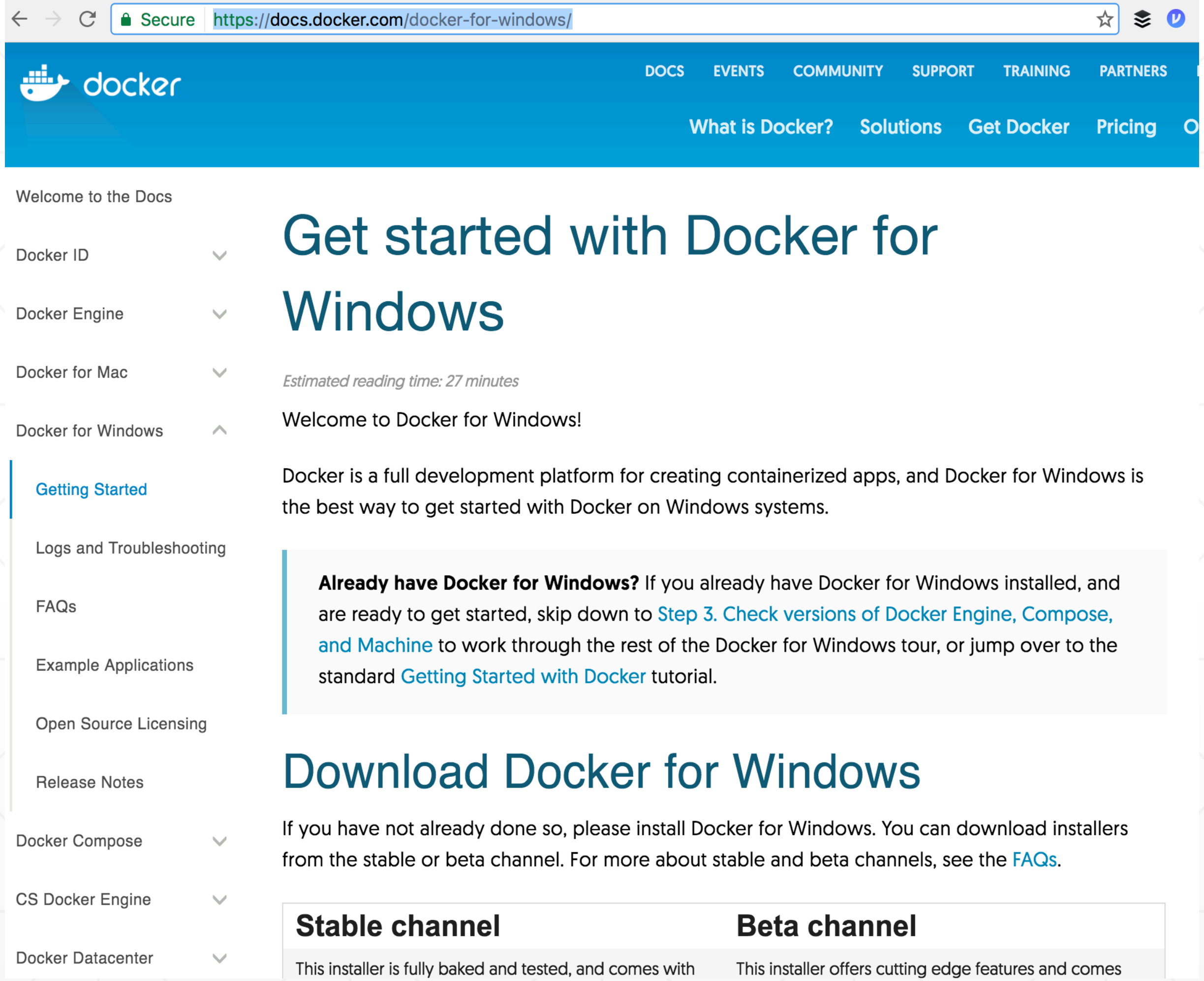
- What is Swift on Linux
  - compiler
  - REPL
  - Standard Library
  - Core Libraries
  - Package Manager

Swift on Linux

- What’s missing
  - Xcode
  - AppKit
  - UIKit
  - No Graphics

Docker image

- Docker on windows
  - http://docs.docker.com/docker-for-windows
- Download swiftfun
- Do swift –version to show it's working



Get started with Docker for Windows

Estimated reading time: 27 minutes

Welcome to Docker for Windows!

Docker is a full development platform for creating containerized apps, and Docker for Windows is the best way to get started with Docker on Windows systems.

> **Already have Docker for Windows?** If you already have Docker for Windows installed, and are ready to get started, skip down to Step 3. Check versions of Docker Engine, Compose, and Machine to work through the rest of the Docker for Windows tour, or jump over to the standard Getting Started with Docker tutorial.

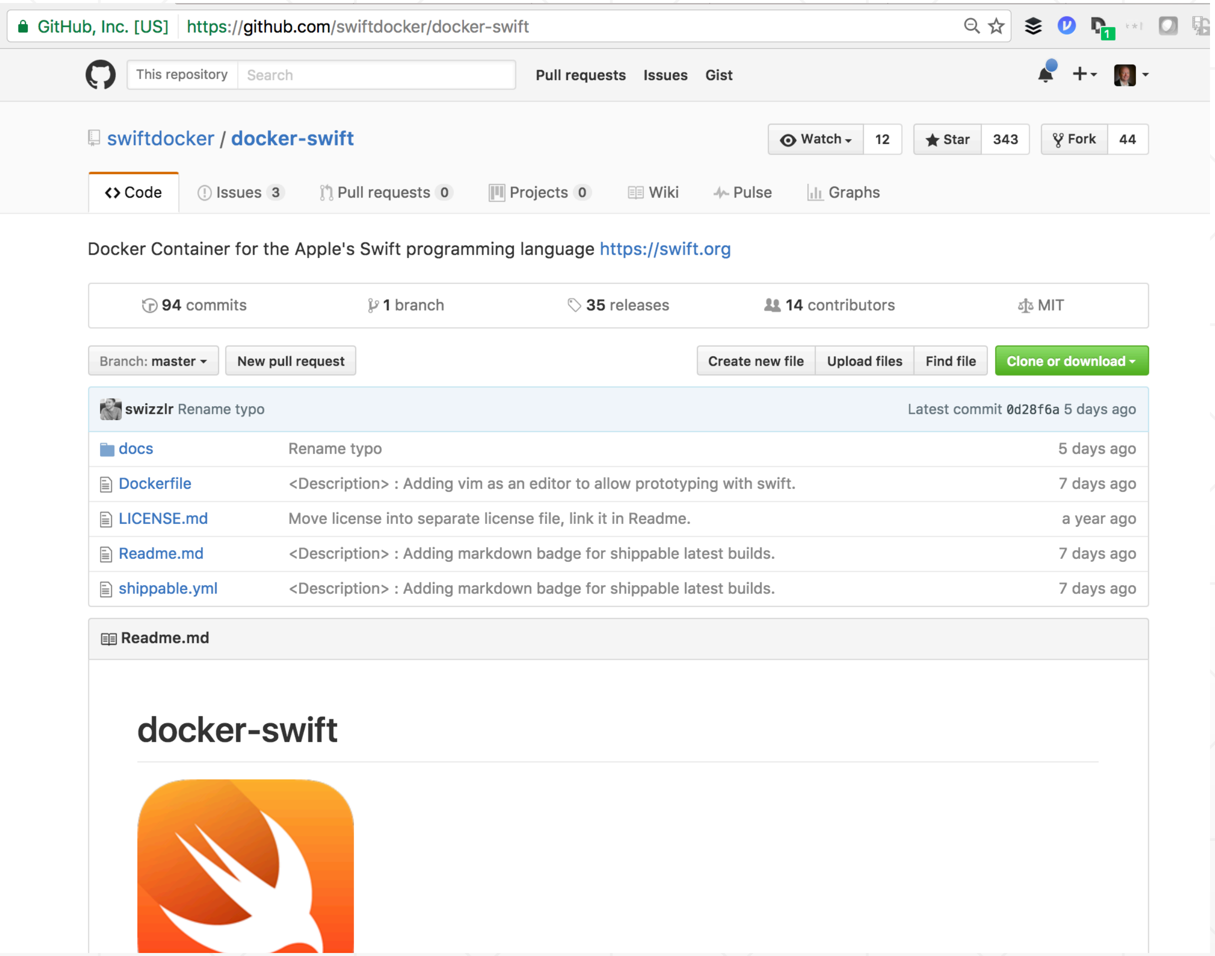
Download Docker for Windows

If you have not already done so, please install Docker for Windows. You can download installers from the stable or beta channel. For more about stable and beta channels, see the FAQs.

Stable channel	Beta channel
This installer is fully baked and tested, and comes with	This installer offers cutting edge features and comes

Docker image

- Docker on windows
- Download swiftfun
  - http://github.com/swiftdocker/docker-swift
  - docker pull swiftdocker/swift
  - docker run --privileged -i -t --name swiftfun swiftdocker/swift:latest /bin/bash
- Do swift --version to show it's working
- Restart
  - docker start swiftfun
  - docker attach swiftfun



swiftdocker / docker-swift

Docker Container for the Apple's Swift programming language https://swift.org

94 commits · 1 branch · 35 releases · 14 contributors · MIT

swizzlr Rename typo — Latest commit 0d28f6a 5 days ago

docs	Rename typo	5 days ago
Dockerfile	<Description> : Adding vim as an editor to allow prototyping with swift.	7 days ago
LICENSE.md	Move license into separate license file, link it in Readme.	a year ago
Readme.md	<Description> : Adding markdown badge for shippable latest builds.	7 days ago
shippable.yml	<Description> : Adding markdown badge for shippable latest builds.	7 days ago

docker-swift

Helloworld

- Create file to print (’Hello, World!”)
- swiftc helloworld
- ./helloworld

Helloworld

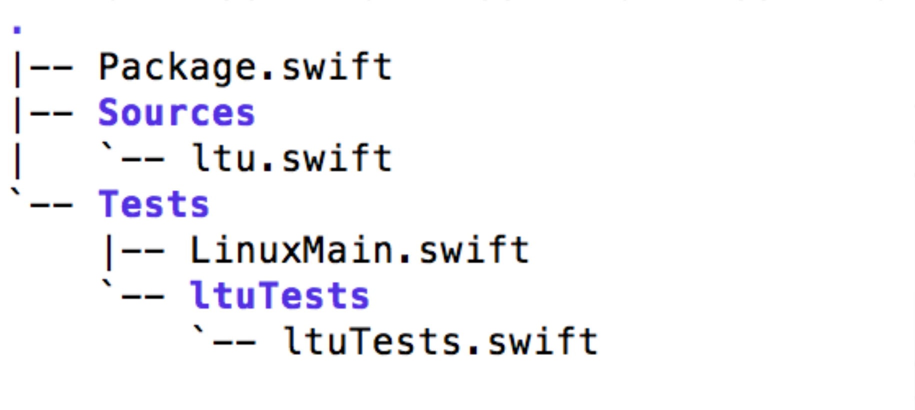
- swift package init --type executable
- swiftc helloworld
- ./helloworld

Swift Package Init

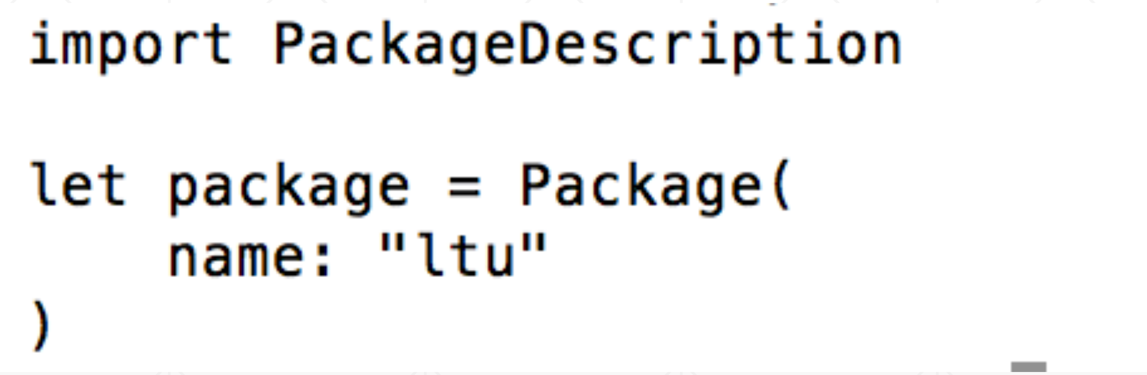
- Package manager file
- LinuxMain.swift file for testing
- Run swift package init
- Download tree to view structure
  - apt-get update
  - agt-get install tree
  - apt-get install vim
- Run tests

Package Manager

- A *package* consists of Swift source files and a manifest file. The manifest file, called Package.swift, defines the package’s name and its contents using the PackageDescription module.



```
.
|-- Package.swift
|-- Sources
|   `-- ltu.swift
`-- Tests
    |-- LinuxMain.swift
    `-- ltuTests
        `-- ltuTests.swift
```

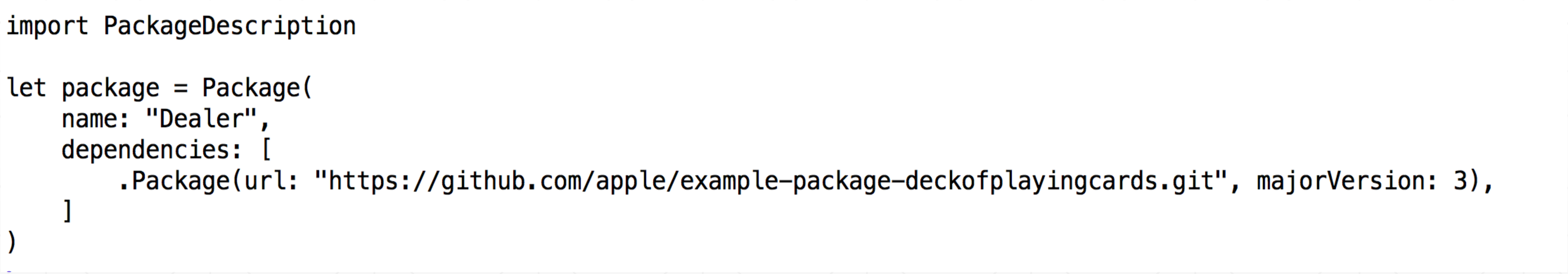


```
import PackageDescription

let package = Package(
    name: "ltu"
)
```

Package Manager

- A package has one or more targets.
- Each target specifies a product and may declare one or more dependencies.

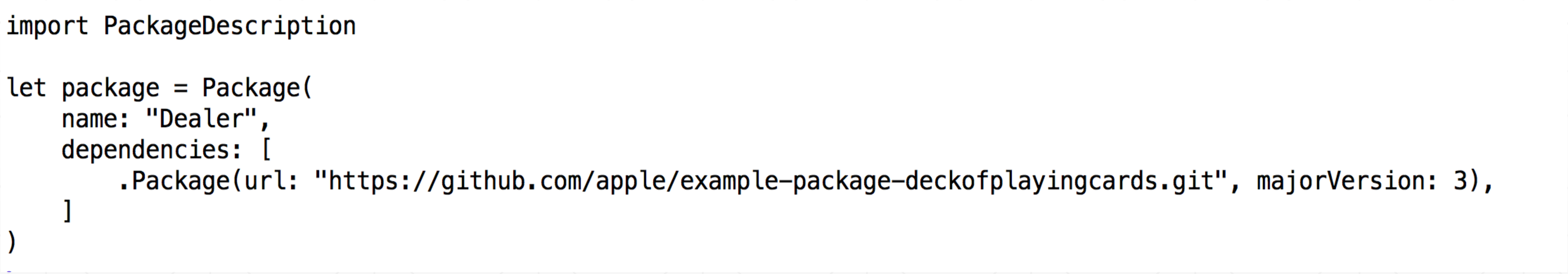


```
import PackageDescription

let package = Package(
    name: "Dealer",
    dependencies: [
        .Package(url: "https://github.com/apple/example-package-deckofplayingcards.git", majorVersion: 3),
    ]
)
```

Products

- A target may build either a library or an executable as its product.
- A *library* contains a module that can be imported by other Swift code.
- An *executable* is a program that can be run by the operating system.

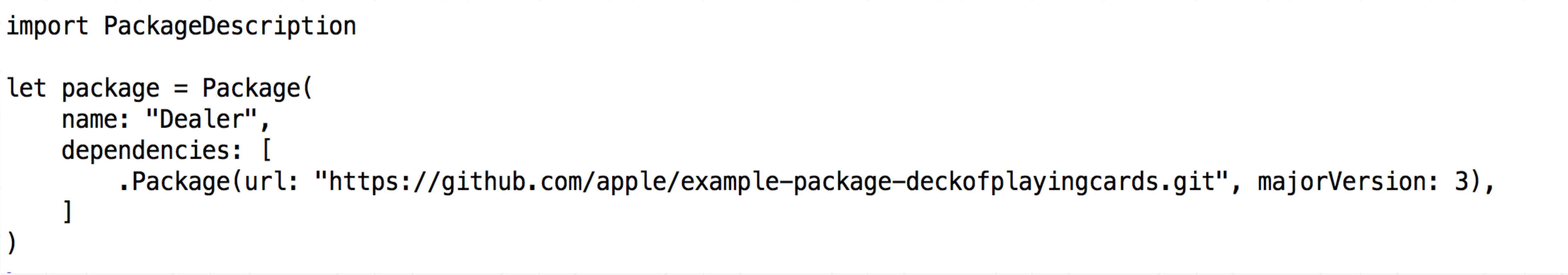


```
import PackageDescription

let package = Package(
    name: "Dealer",
    dependencies: [
        .Package(url: "https://github.com/apple/example-package-deckofplayingcards.git", majorVersion: 3),
    ]
)
```

Dependencies

- A target’s dependencies are modules that are required by code in the package.
- A dependency consists of a relative or absolute URL to the source of the package and a set of requirements for the version of the package that can be used.



```
import PackageDescription

let package = Package(
    name: "Dealer",
    dependencies: [
        .Package(url: "https://github.com/apple/example-package-deckofplayingcards.git", majorVersion: 3),
    ]
)
```

Deck of cards playing game

- git clone https://github.com/apple/example-package-dealer.git
- swift build
- Run .build/debug/Dealer