

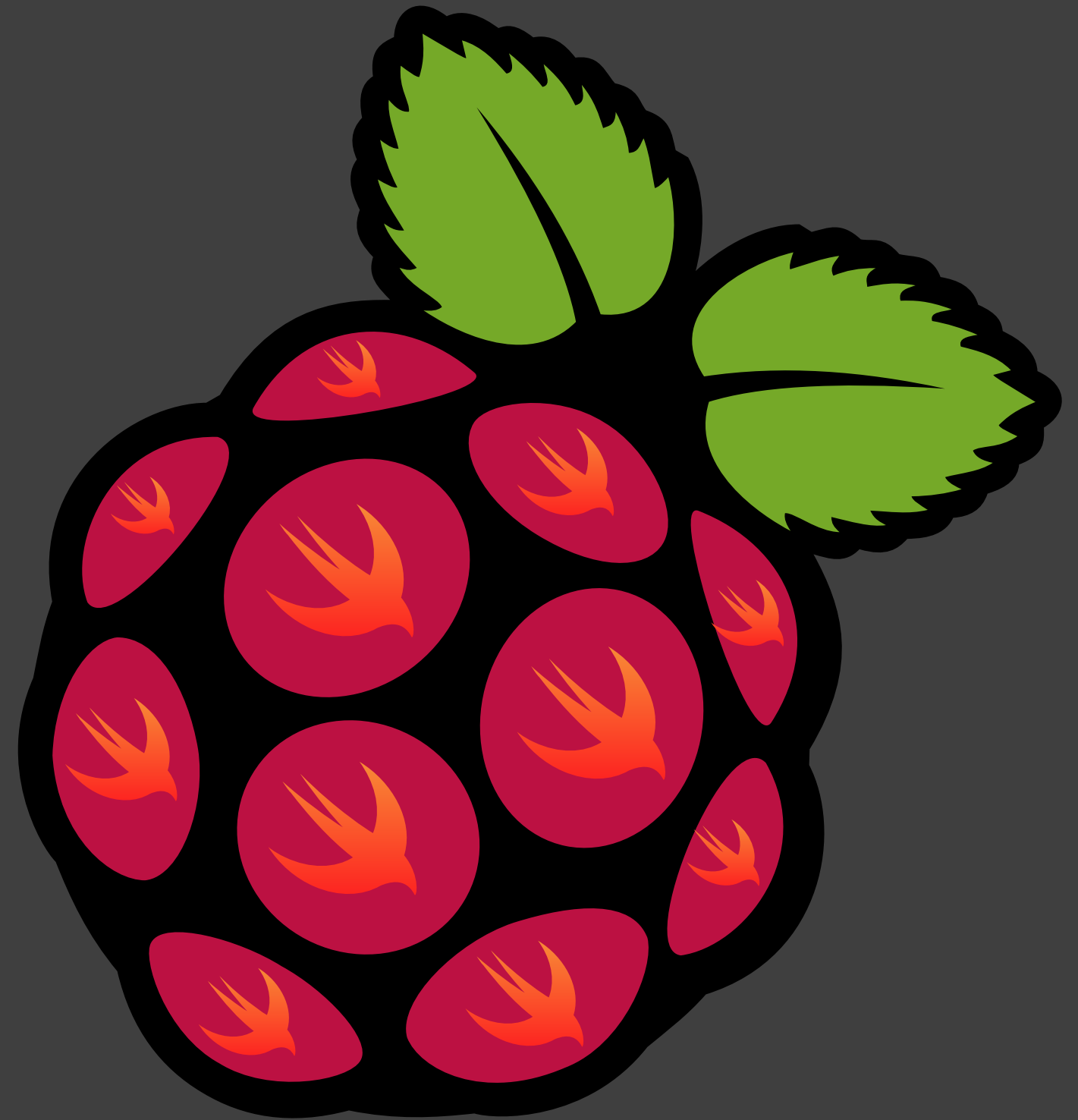
# SwiftNIO on Raspberry Pi

---

Running Swift on small hardware  
(and VERY LARGE hardware).

---

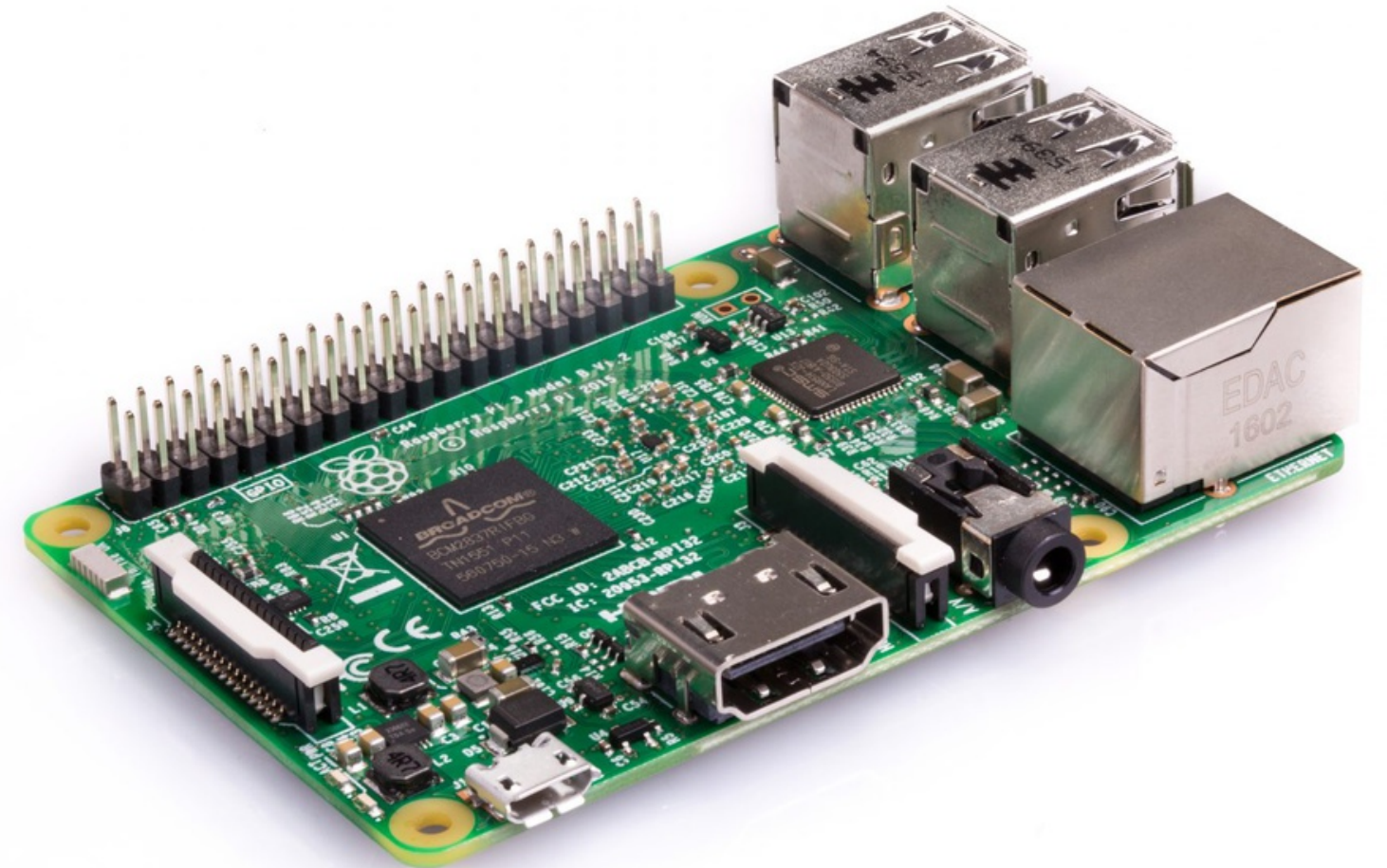
Helge Heß @ ZeeZide / Always Right Institute



# What is a Raspberry Pi?

A fast phone w/o the cellular modem, battery and touchscreen.

- 4-Core ARMv8 CPU @ 1.4Ghz
- 1GB RAM
- Ethernet, WLAN, HDMI, Bluetooth 4.2, 4x USB
- GPIO Pins



# What is a Raspberry Pi?

---

Nice cases available!

---

Runs faster w/ Swift sticker on it





# What is a Raspberry Pi?

... or 3-D print your own



# What is a Raspberry Pi?

---

CCU3 - HomeMatic Smart Home Basestation



## CCU 3 ★ HomeMatic Smart Home Basestation

RaspberryPi €35 - CCU 3 €150 📍 Profit.





# SwiftNIO is intended for The Clown

---

## iCloud Datacenter in Iowa

---

Built to serve hundreds of millions  
of devices people.





# Datacenter Node

Could be:  
32 Xeon Cores?  
128 GB RAM?

**UNLIMITED RESOURCES!!!**



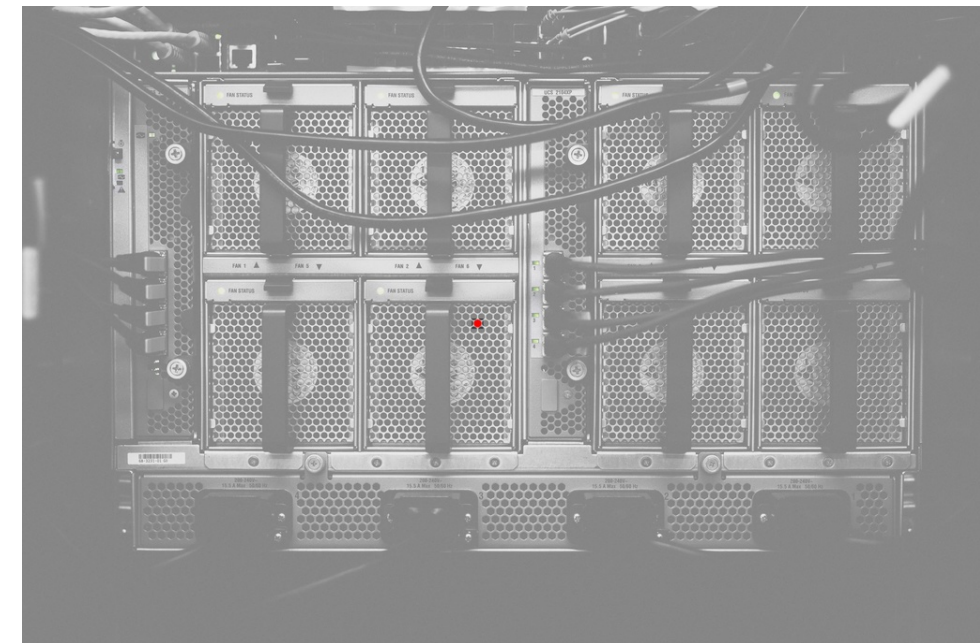


# Datacenter Node

Unlimited Resources. **Not.**

128 GB RAM / 10k-100k sessions:  
1..10MB each. CPU the same.

SwiftNIO is optimised for speed  
because it runs on tiny slices of  
big machines.

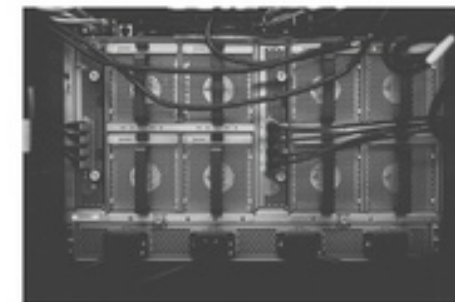
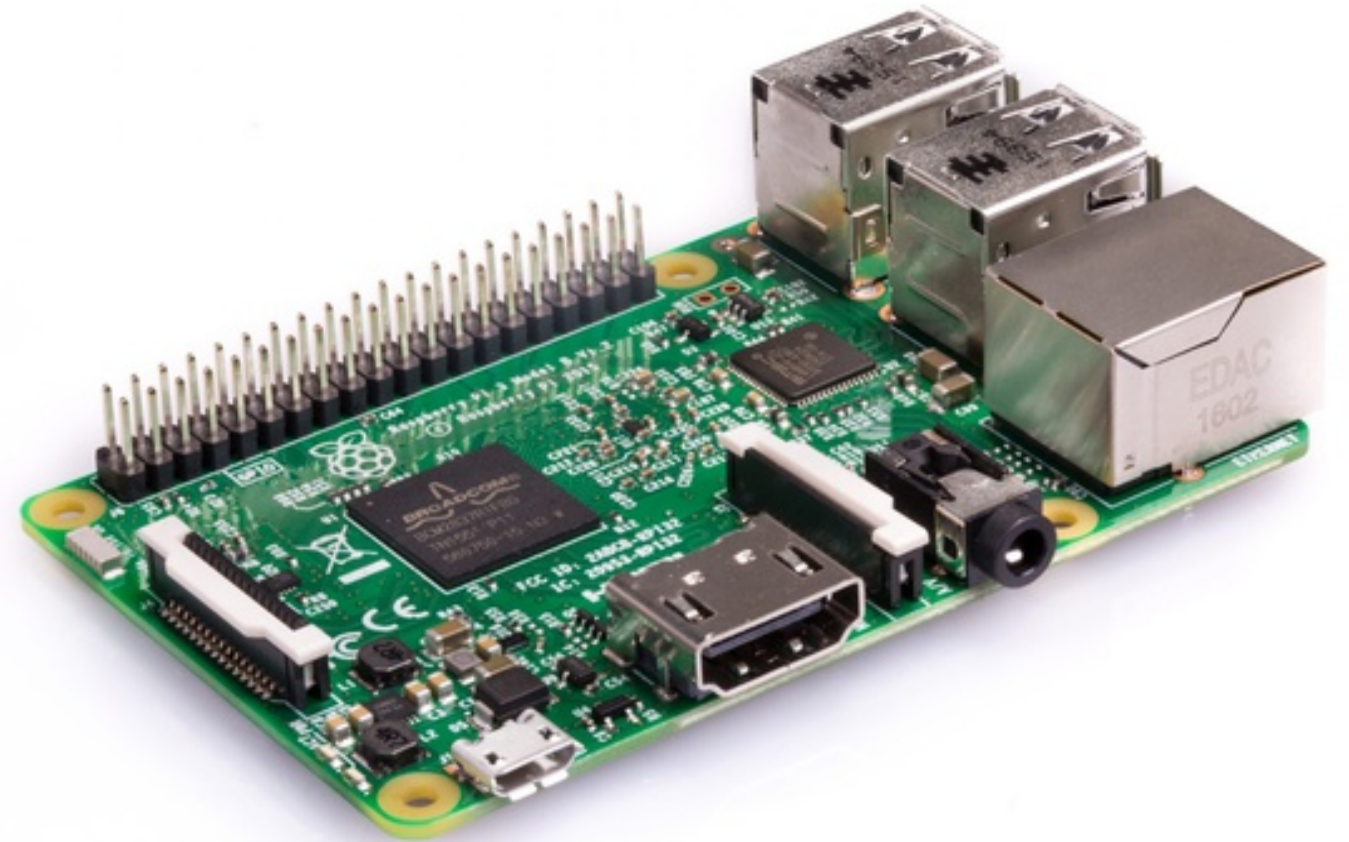


# Raspi > Datacenter Node

Pi: **UNLIMITED RESOURCES!!!**

SwiftNIO is a perfect fit for tiny servers too, it is built to be fast and consume little resources.

Swift + NIO uses less resources than Node, Rails or Java.  
Still a reasonably nice language.





# Demo Time!

---

## Create NIO webservice

---

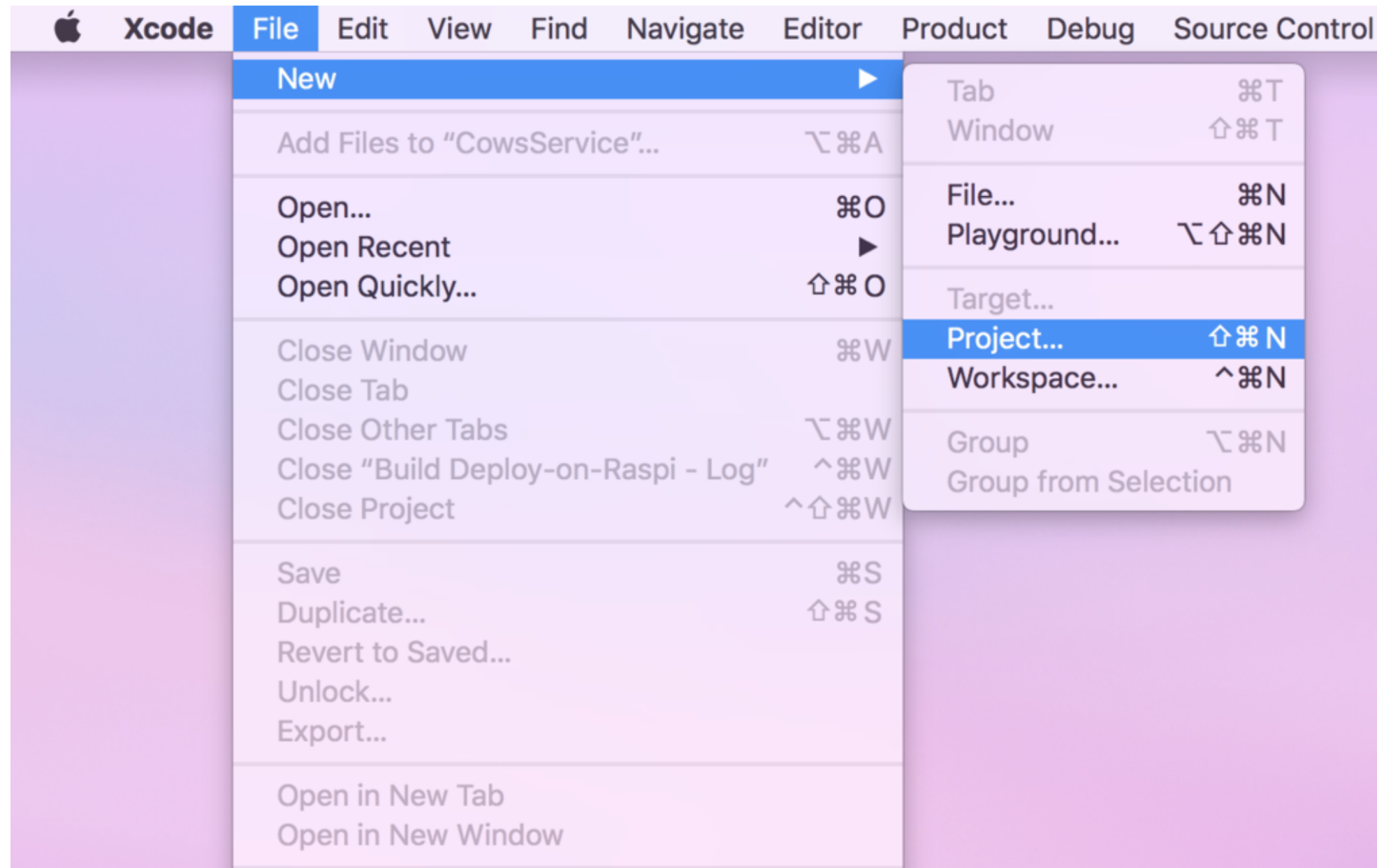
## Test on Intel

---

## Deploy on RasPi

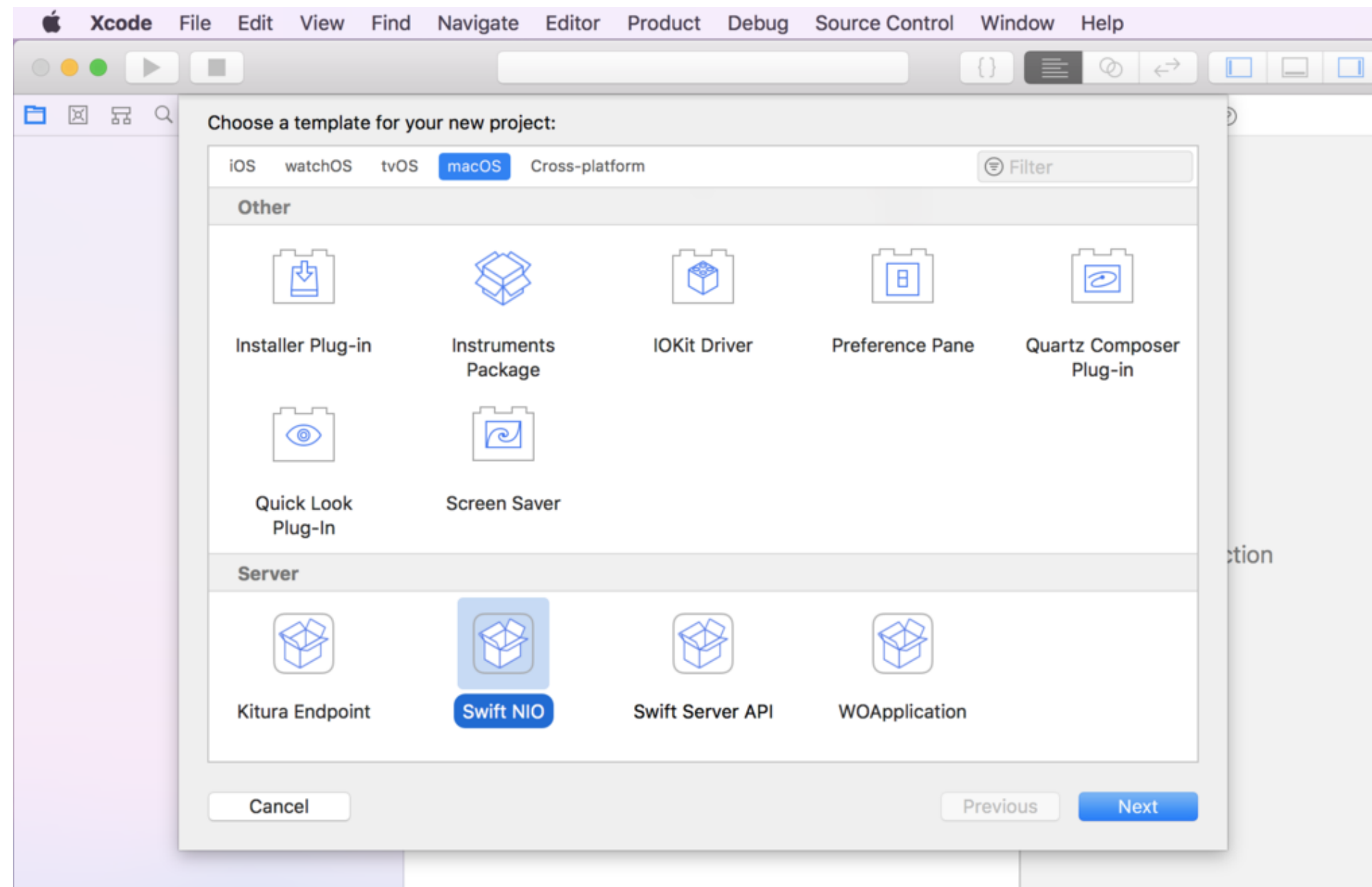


# Demo: Create New Project in Xcode





# Demo: Select Swift NIO Xcode Template



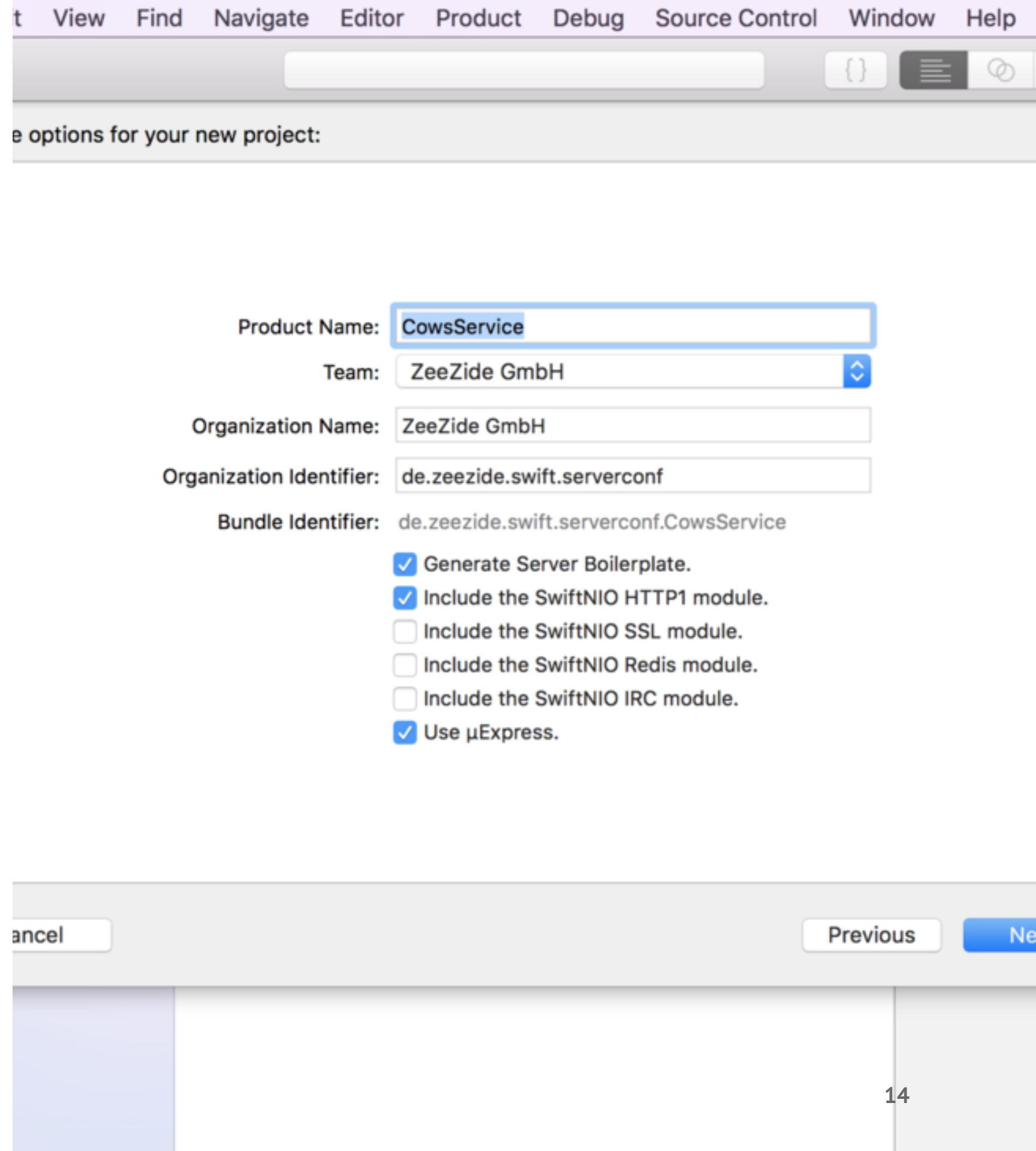
## Demo: Select Dependencies

Add **μExpress**, a tiny framework which wraps around Swift NIO.

Interested?

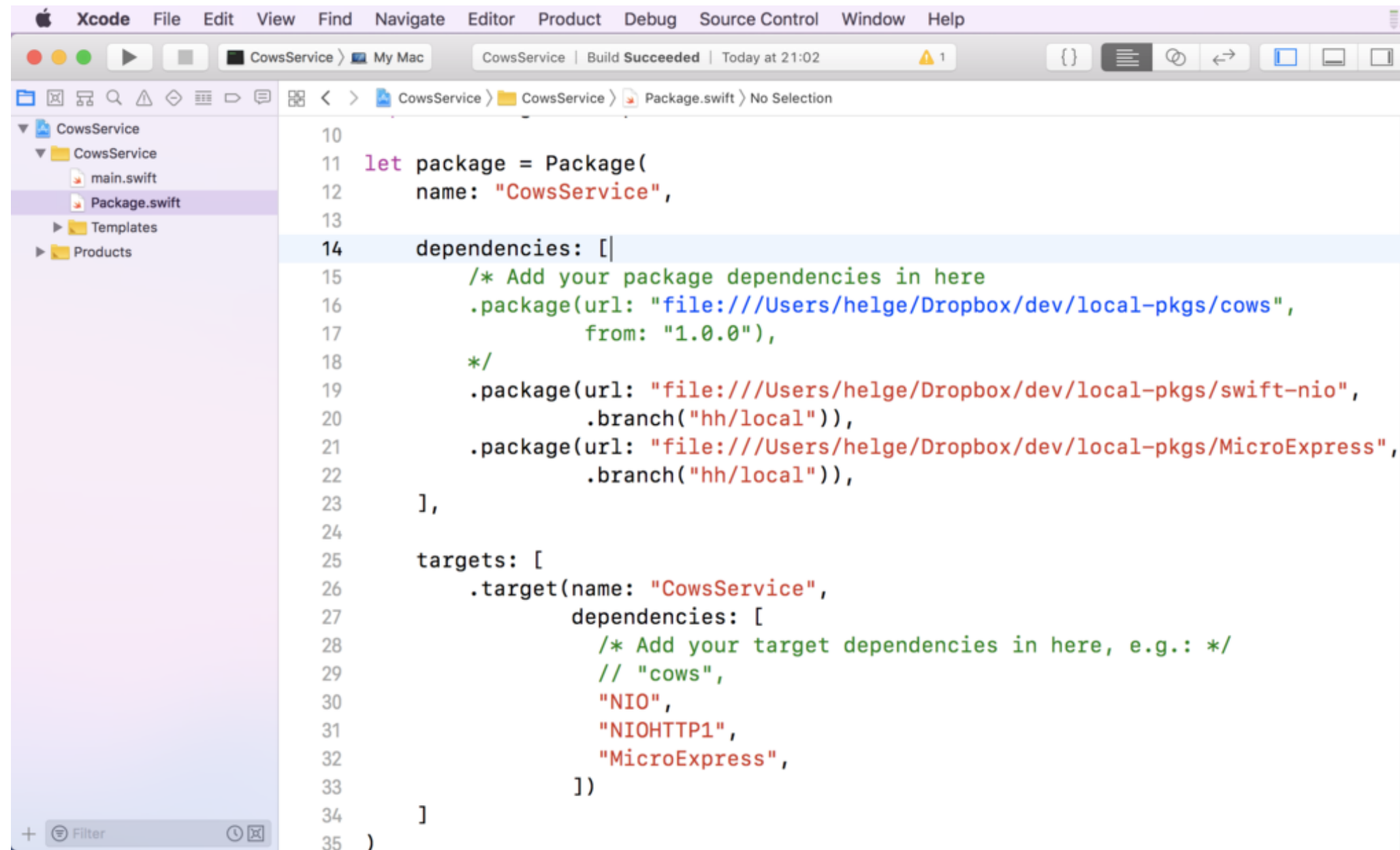
See: **A μTutorial on Swift NIO**

Only provides basics, doesn't compete w/ Kitura or Prefect.



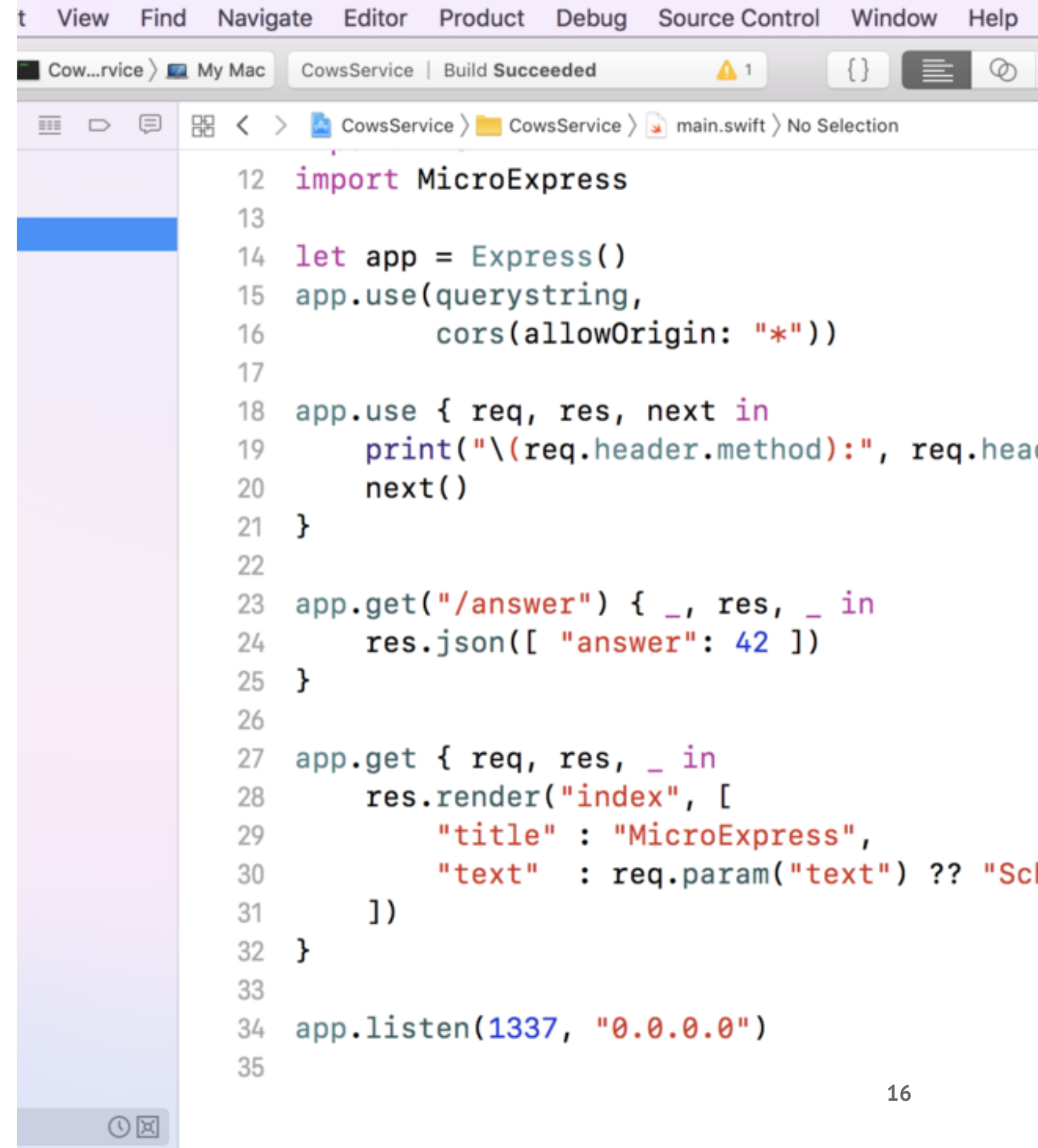


# Demo: Generated Package.swift



# Demo: Generated main.swift

1. Setup Swift NIO Server
2. Some middleware
3. Start listening on port



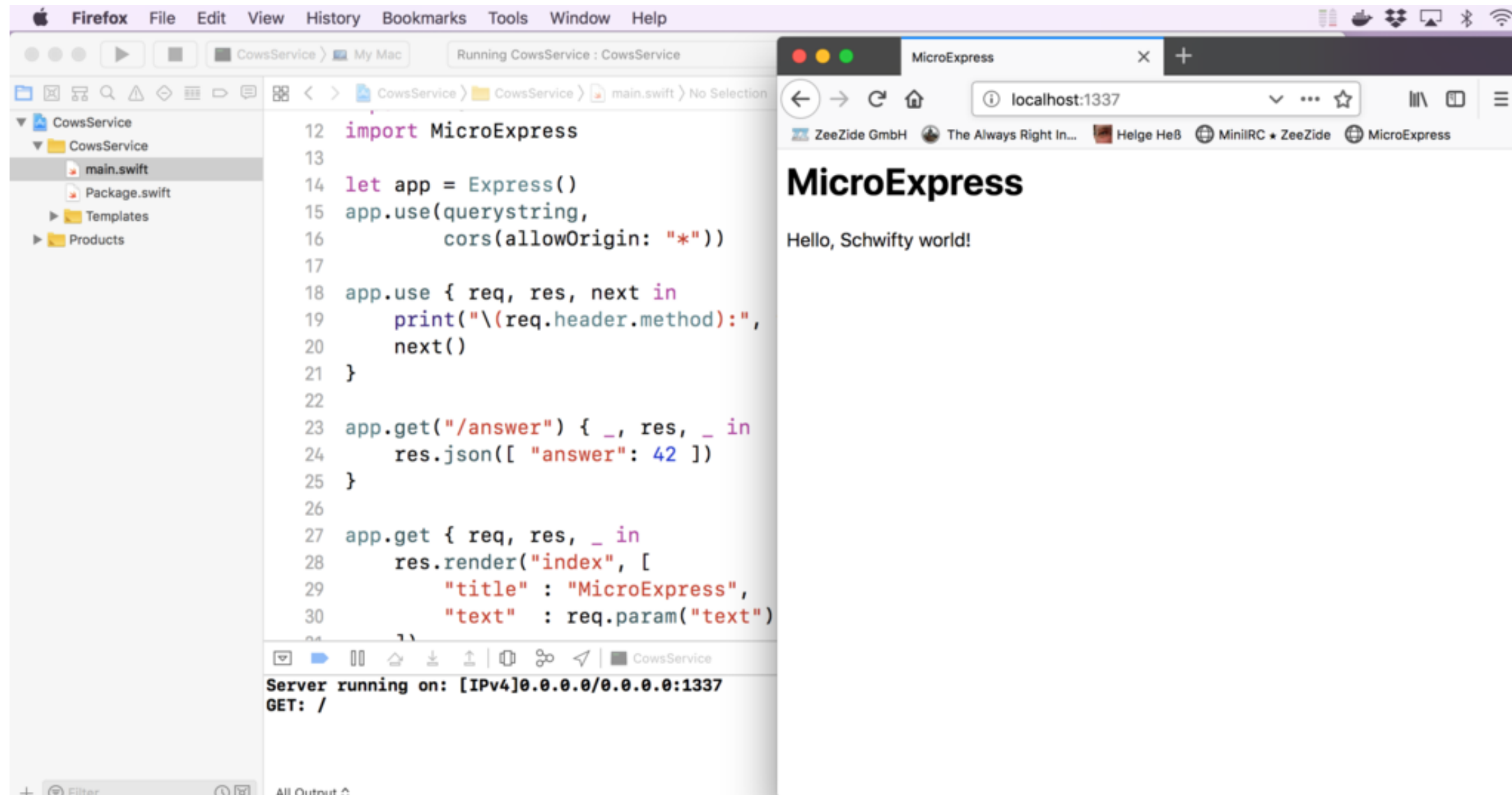
The screenshot shows an IDE window with the following elements:

- Top Bar:** Contains menu items: View, Find, Navigate, Editor, Product, Debug, Source Control, Window, Help.
- Tab Bar:** Shows 'Cow...rvice' and 'My Mac'. The active tab is 'CowsService | Build Succeeded' with a warning icon and the number '1'.
- Breadcrumb:** Shows the file path: 'CowsService > CowsService > main.swift > No Selection'.
- Code Editor:** Displays the content of 'main.swift' with line numbers 12 through 35. The code is as follows:

```
12 import MicroExpress
13
14 let app = Express()
15 app.use(querystring,
16         cors(allowOrigin: "*"))
17
18 app.use { req, res, next in
19     print("\(req.header.method):", req.header.method)
20     next()
21 }
22
23 app.get("/answer") { _, res, _ in
24     res.json([ "answer": 42 ])
25 }
26
27 app.get { req, res, _ in
28     res.render("index", [
29         "title" : "MicroExpress",
30         "text"   : req.param("text") ?? "Some text"
31     ])
32 }
33
34 app.listen(1337, "0.0.0.0")
35
```



# Demo: Run locally on Mac



# Demo: How to get it on the Raspi?

The Raspi is running **Docker**.

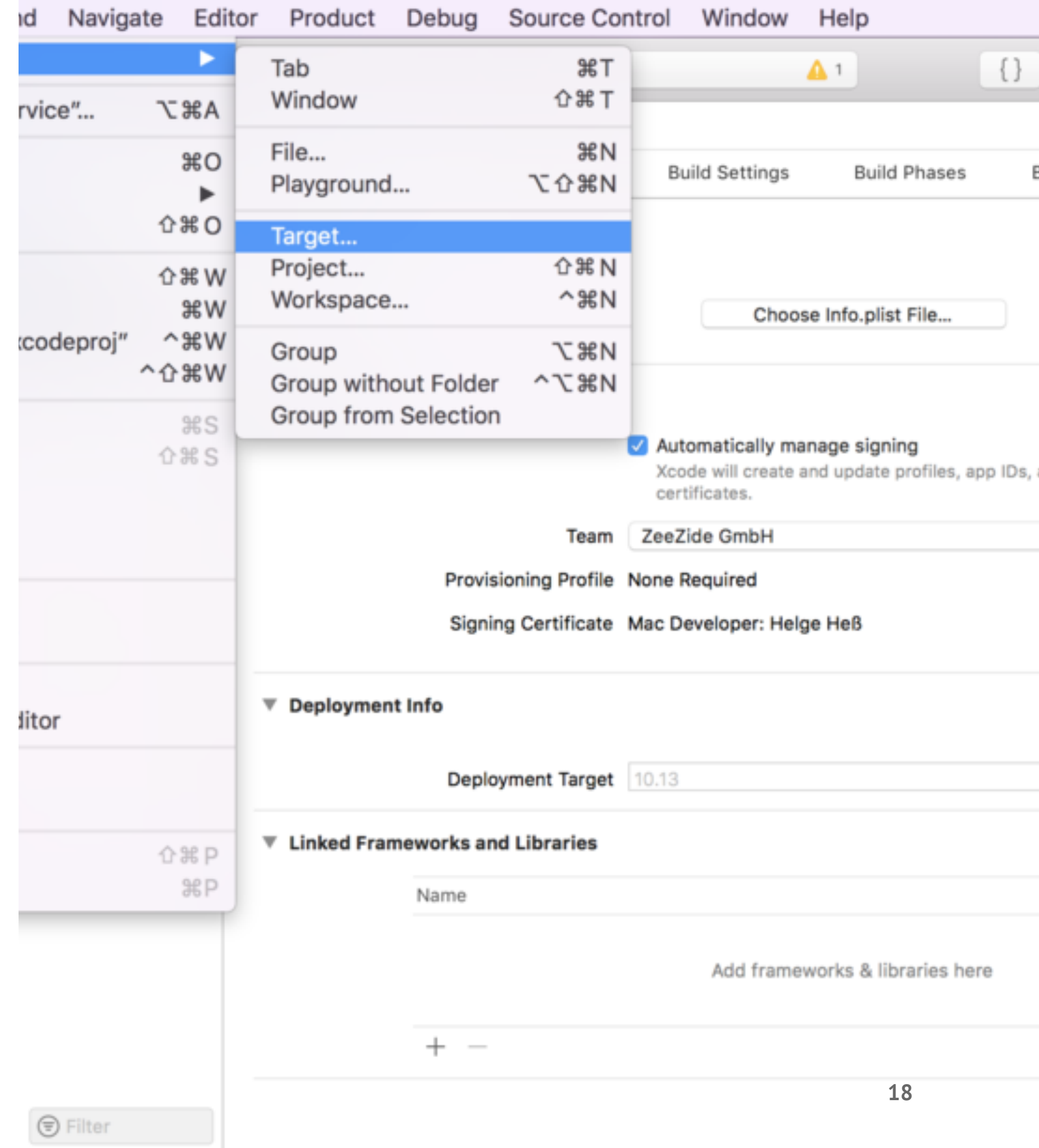
1. Create a new target

2. Add a script build phase

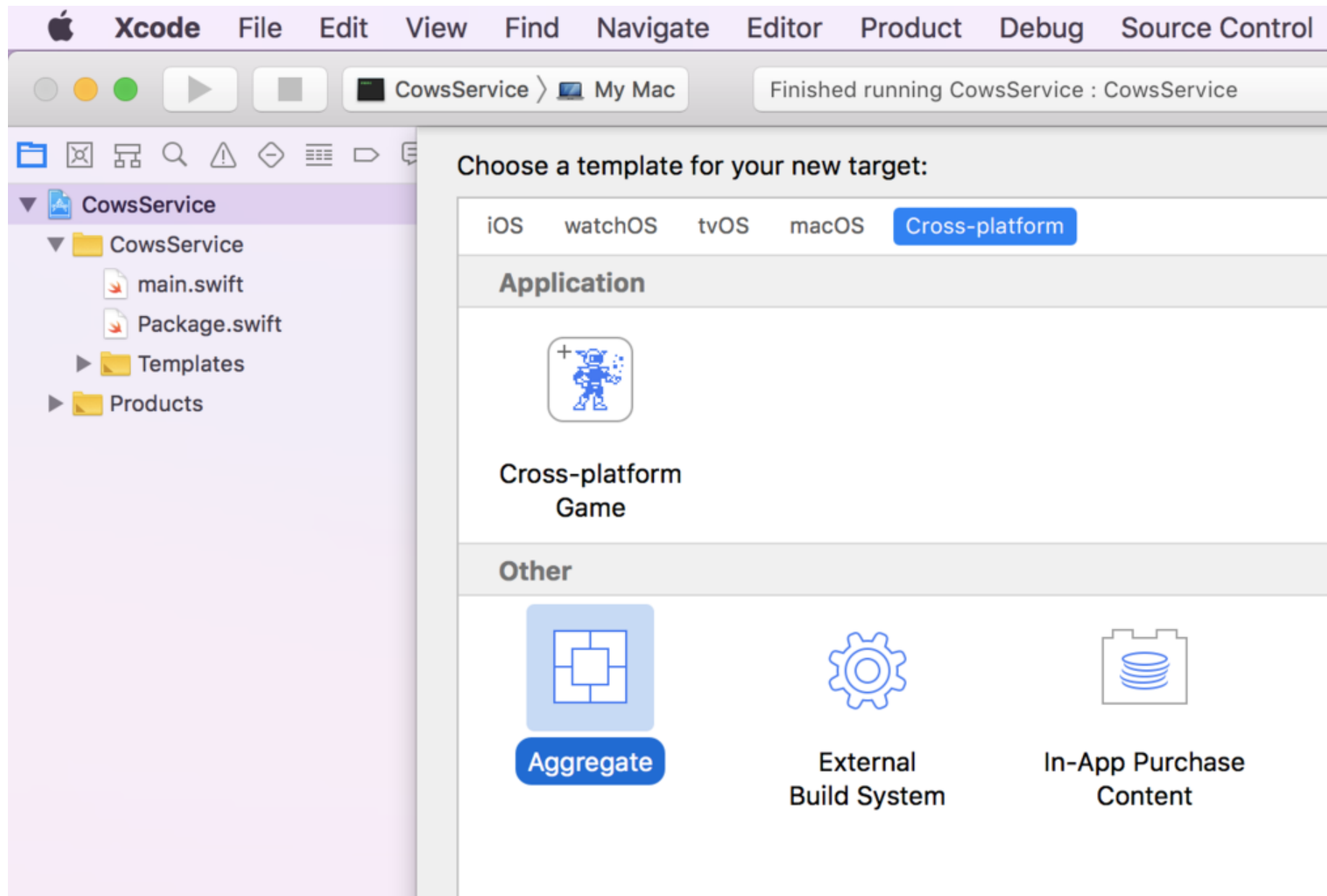
3. 🕒 Build

4. 🕒 Wait

5. 🎉 Success

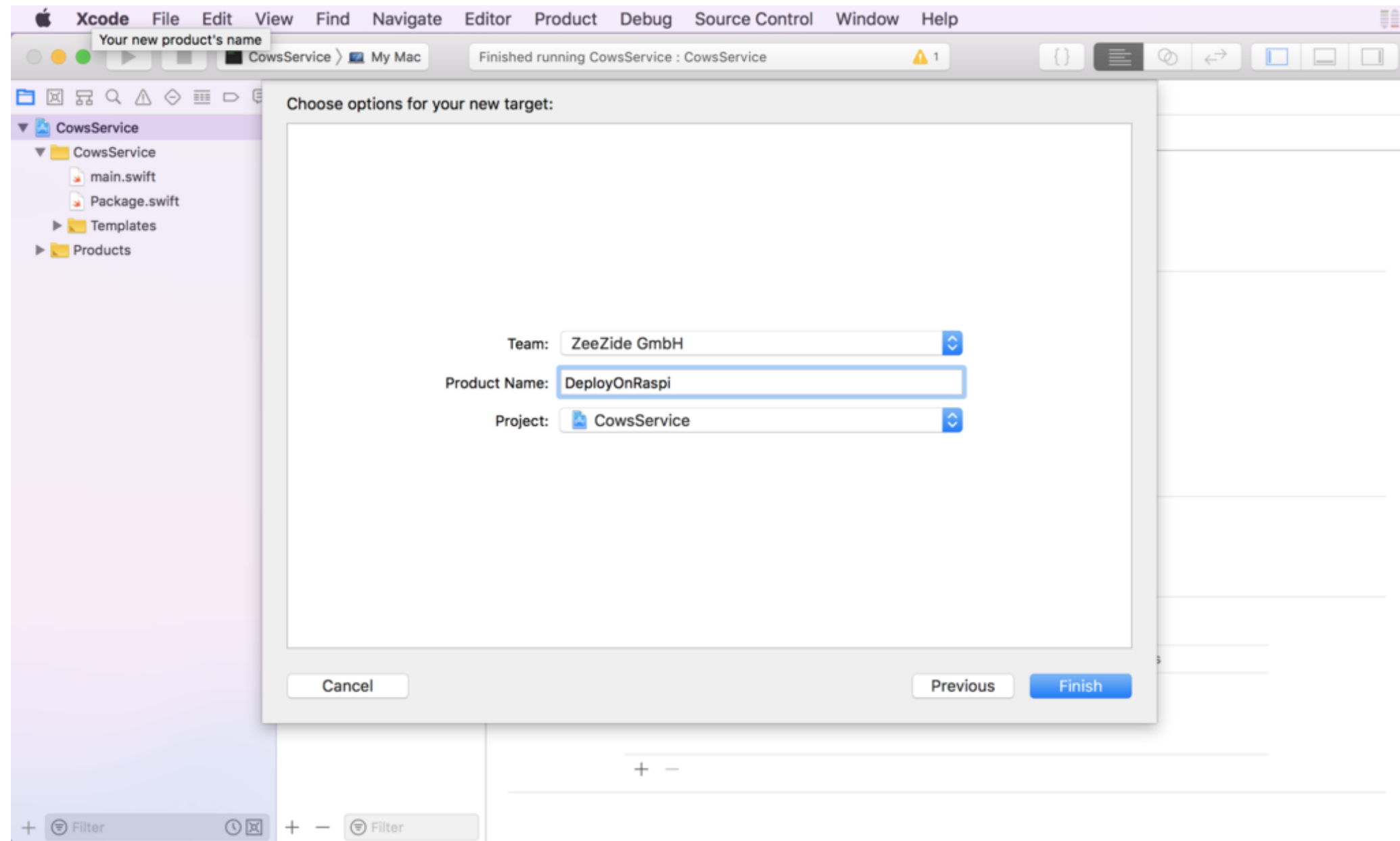


# Demo: Create aggregate target

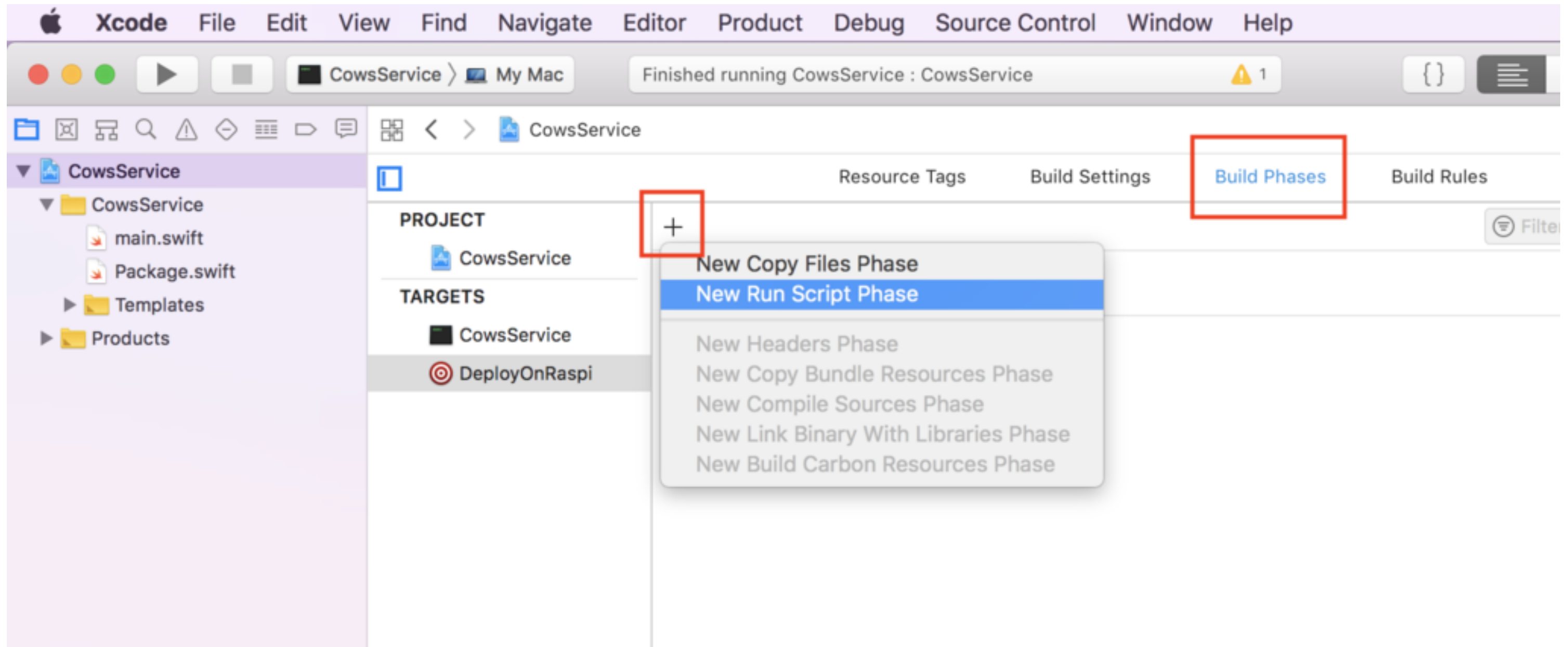




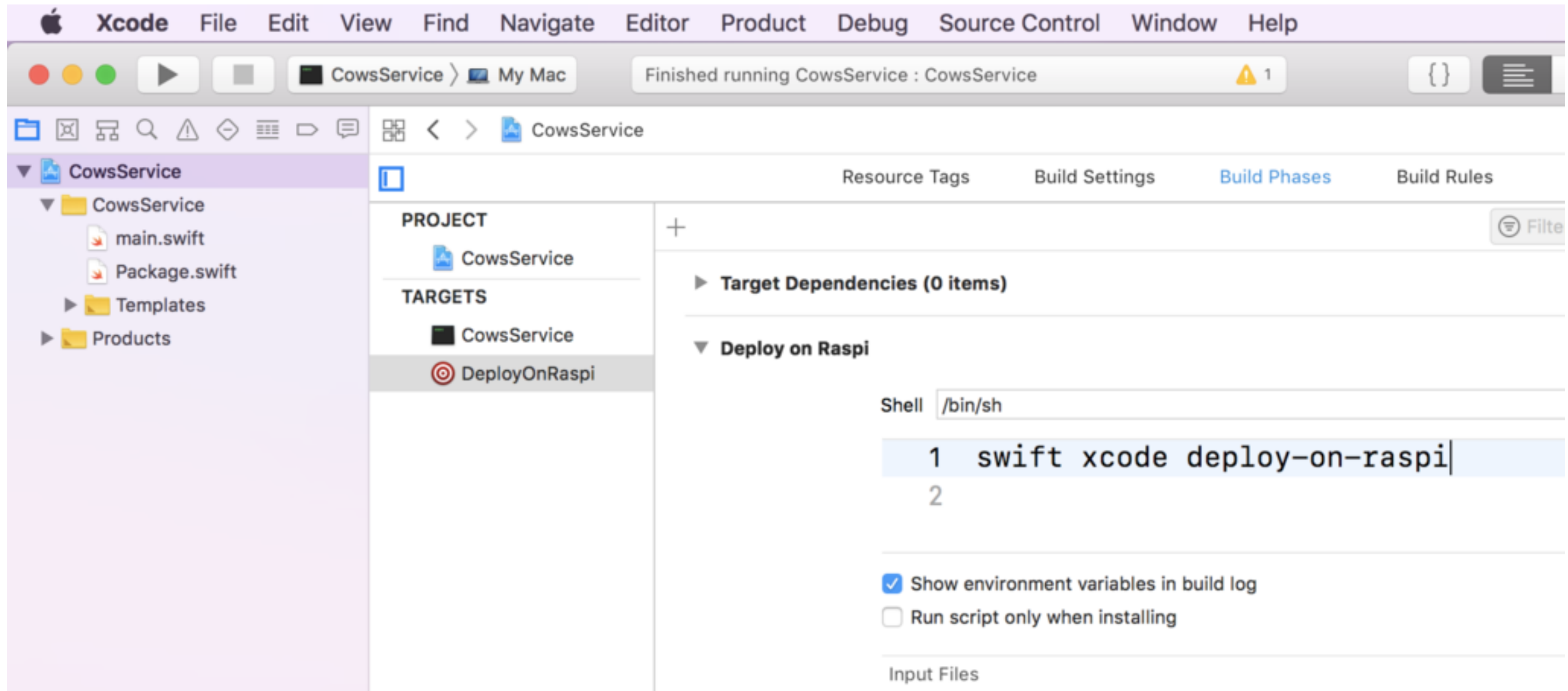
# Demo: Create aggregate target



# Demo: Add script phase to aggregate target



# Demo: swift xcode deploy-on-raspi





# Demo: Simple Deployment Script

```
#!/bin/bash

PROJECT_DIR=${PROJECT_DIR:=$(pwd)}
PROJECT=${PROJECT:=$(basename ${PROJECT_DIR})}
PROJECT_TARGET_DIR="${PROJECT_DIR}/${PROJECT}"

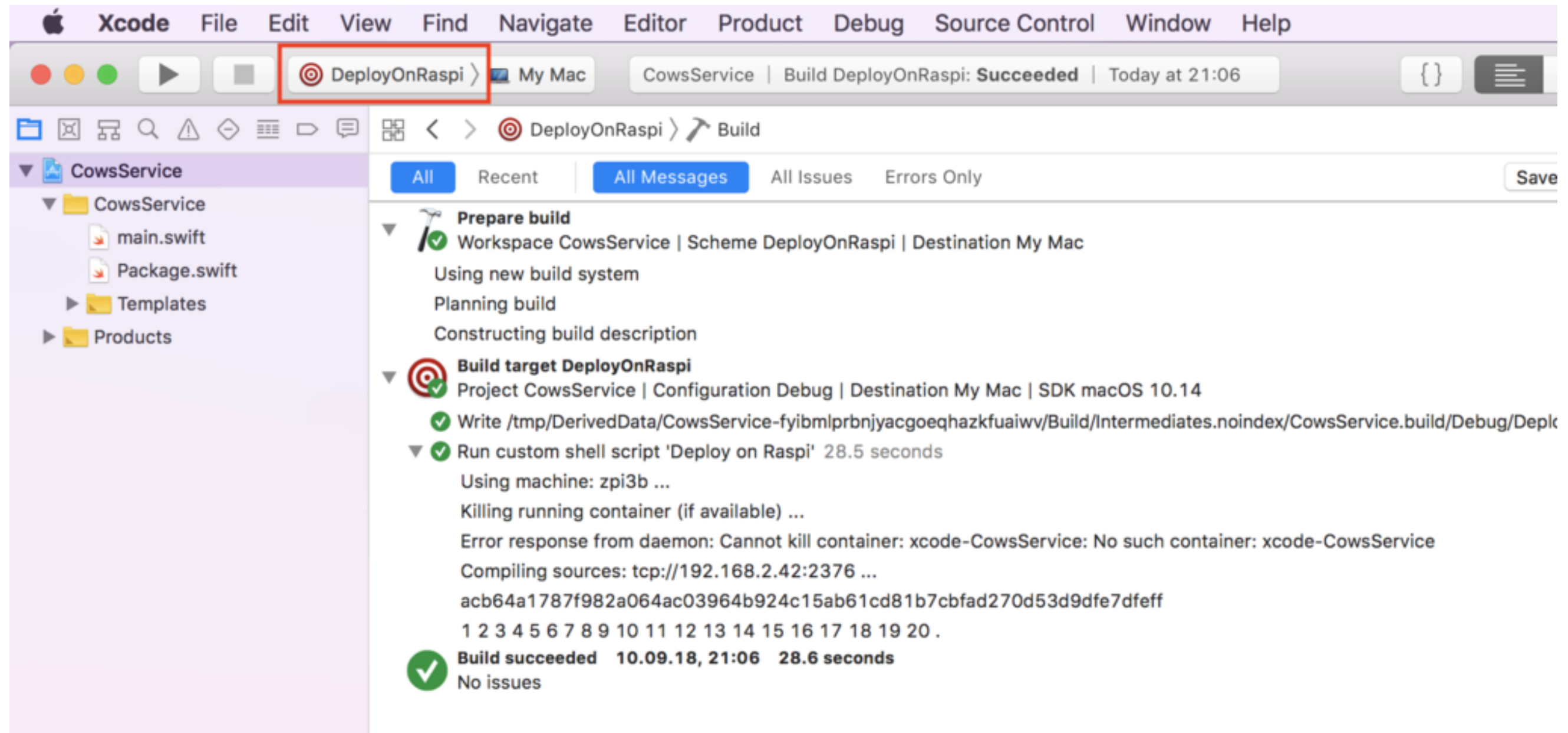
DOCKER_MACHINE_HOST=${DOCKER_MACHINE_HOST:=zpi3b}
DOCKER_IMAGE="helje5/arm64v8-swift-dev:latest"

CONTAINER_NAME="xcode-${PROJECT}"

eval $(docker-machine env ${DOCKER_MACHINE_HOST})

docker rm -f "${CONTAINER_NAME}"
docker run -d -p "1337:1337" --name "${CONTAINER_NAME}" \
    -v "${PROJECT_TARGET_DIR}:/src" \
    "${DOCKER_IMAGE}" \
    bash -c "cd /src && swift build && \
        exec $(ls $(swift build --show-bin-path)/${PROJECT})"
```

# Demo: swift xcode deploy-on-raspi



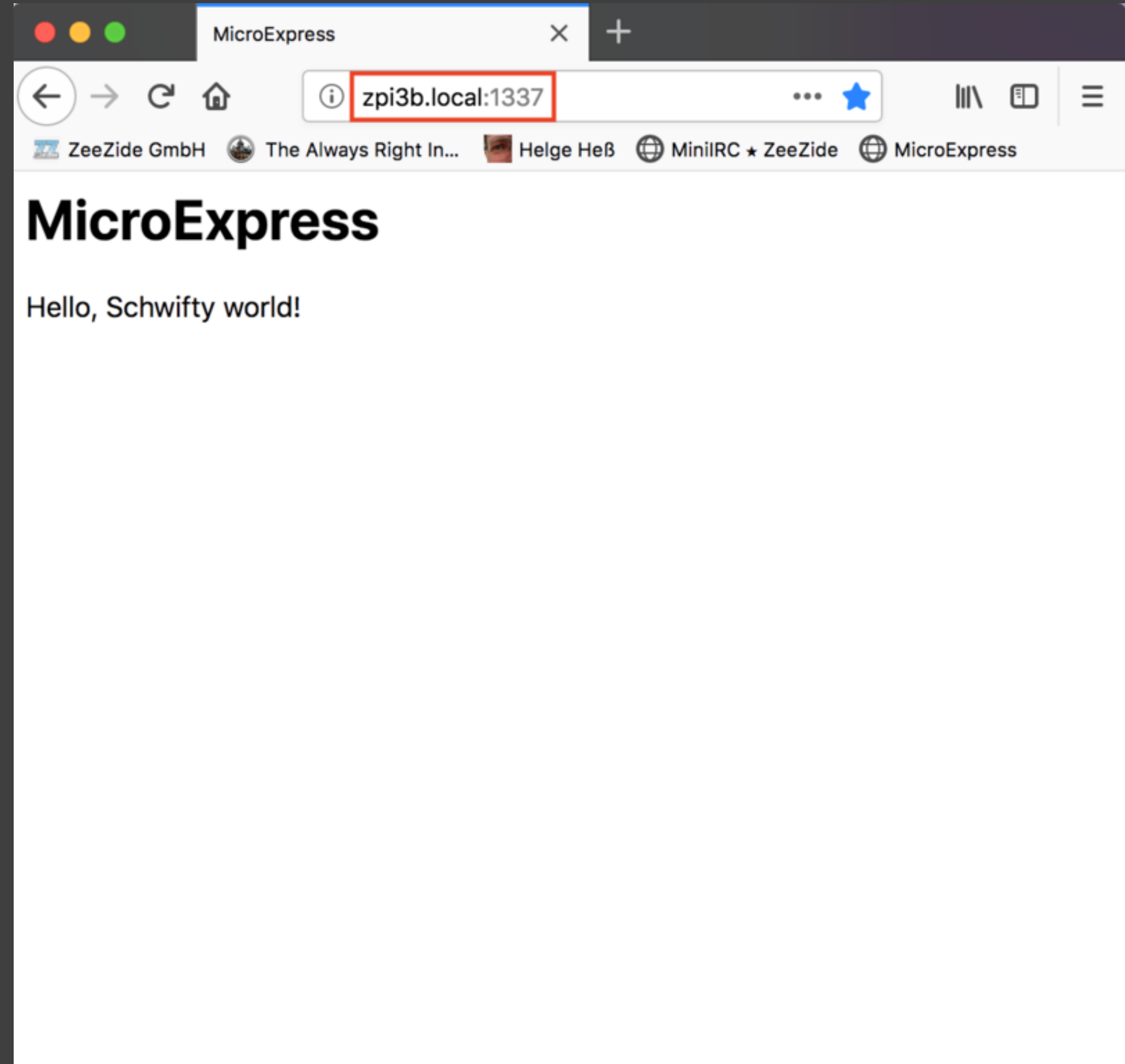
# Demo: Success

---

Building on and  
Deploying to Raspi is  
trivial

---

Thanks Docker!



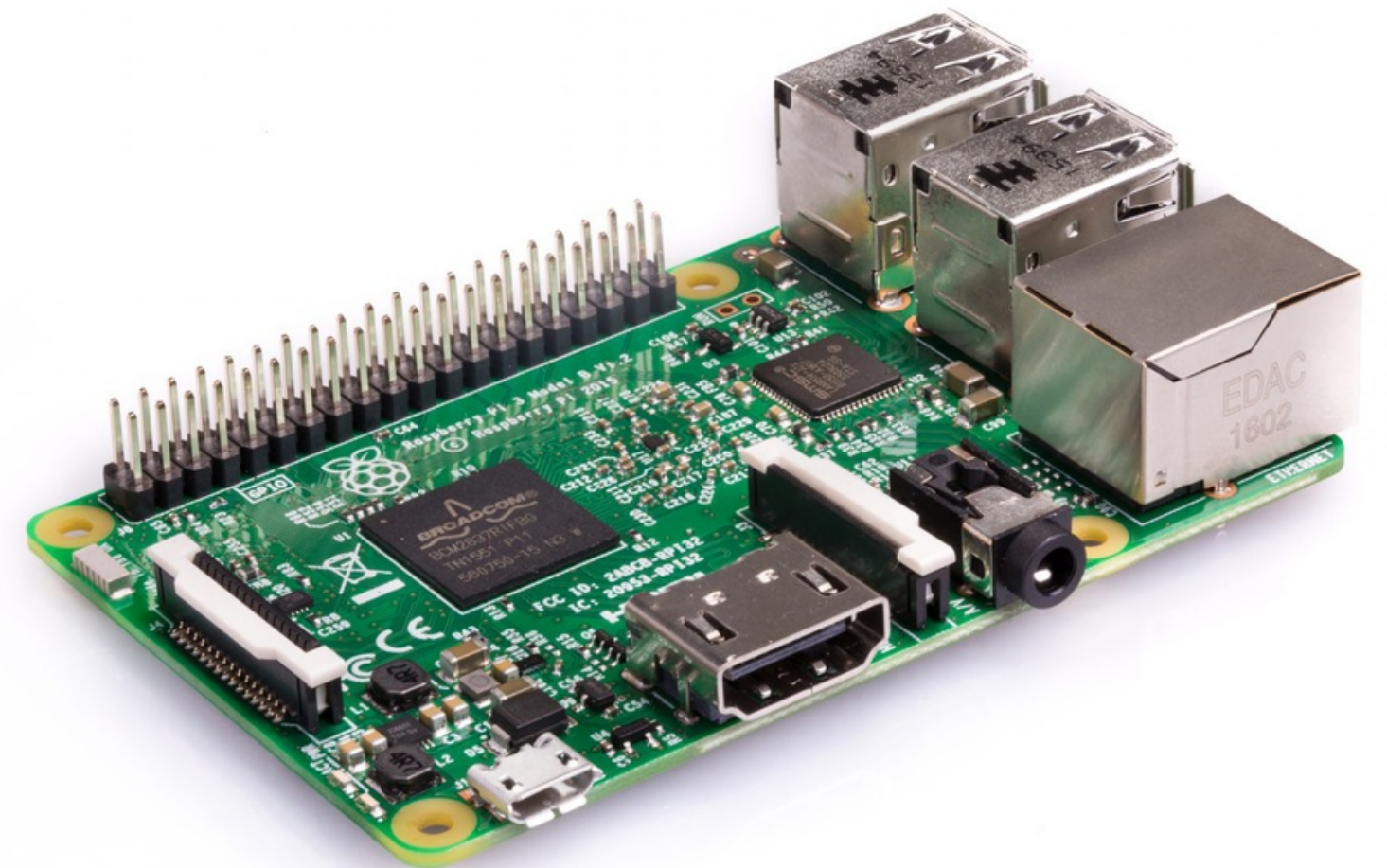


## Hardware required

- Raspberry Pi 3B/+ (35€)
- You bring:
  - USB Power
  - SD-Card (4GB+)

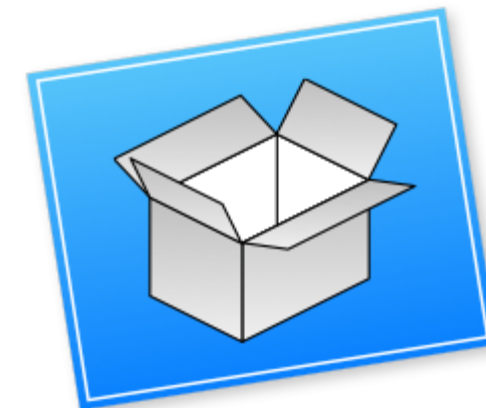
# That's it!

Got a USB-C MacBook? Extra 40€ for **SD Card dongle** (yes, more than the Pi ...).



# How to get Swift on the Pi

- build from source:  
**buildSwiftOnARM**
- community drops (tarballs,  
like on Swift.org)
- community **Debian** packages
- my unofficial **Docker** images



# HypriotOS

---

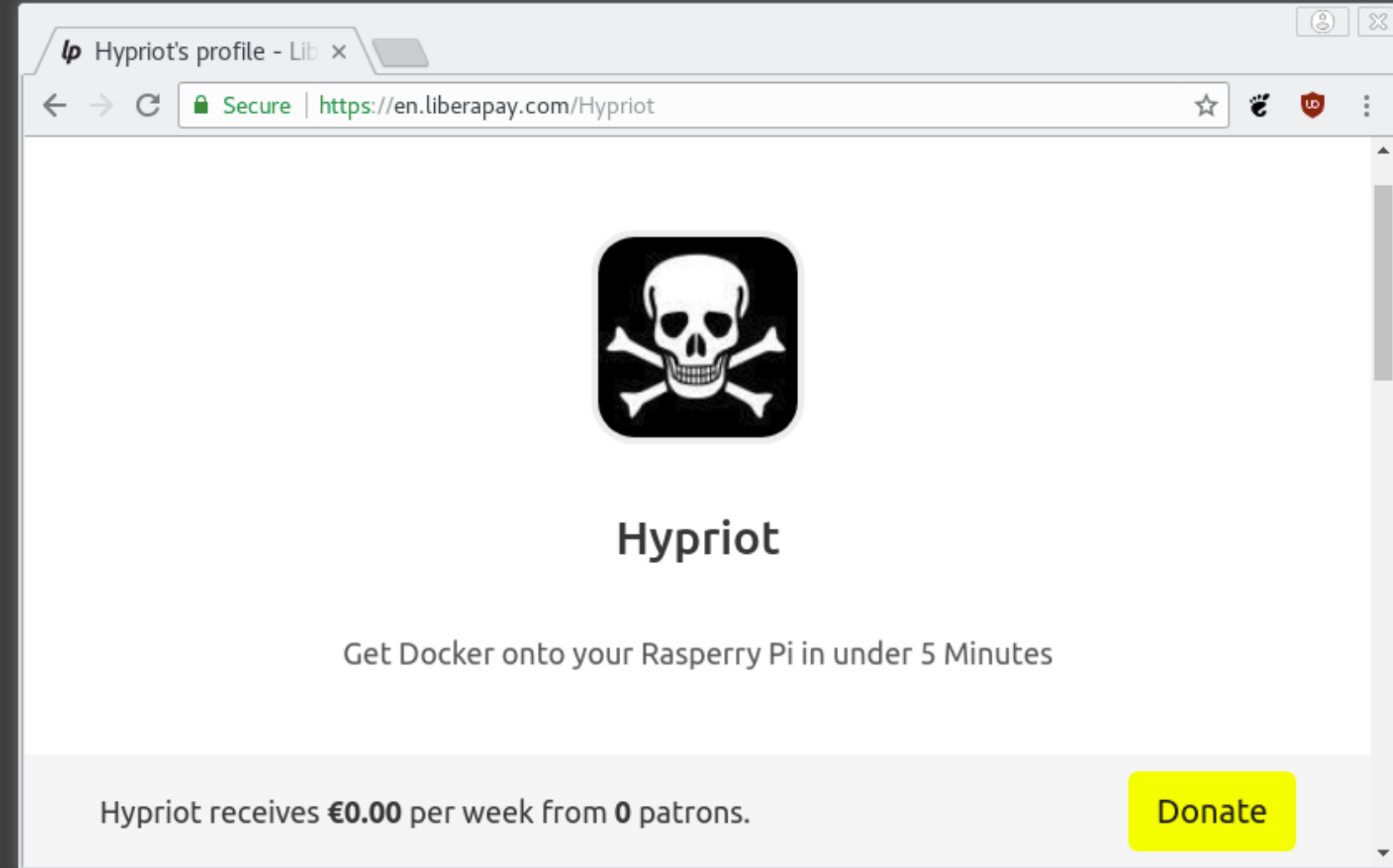
Small Docker host OS

---

Download tool ★ Flash to SD ★ Boot ★ Done.

---

[www.hypriot.com](http://www.hypriot.com)





# Swift-on-ARM Community

- Slack: [swift-arm](#), to stay up to date: [uraimo blog](#)
- No official ARM builds on [swift.org](#)
  - Funny, because 98% of the Swift deployments are on ARM (watch, phone, TV)
  - Swift.org [community CI](#)
- Takes time until releases are ported
  - Now we have 4.1, 4.2 may take some months

# Porting efforts 32-bit vs. 64-bit

- Not usually an issue for applications
- Had to port SwiftNIO ([PR #383](#), [PR #486](#))
  - Mostly because SwiftNIO is a little closer to the hardware than usual
- Most common issue: Int vs. Int64 vs. Int32

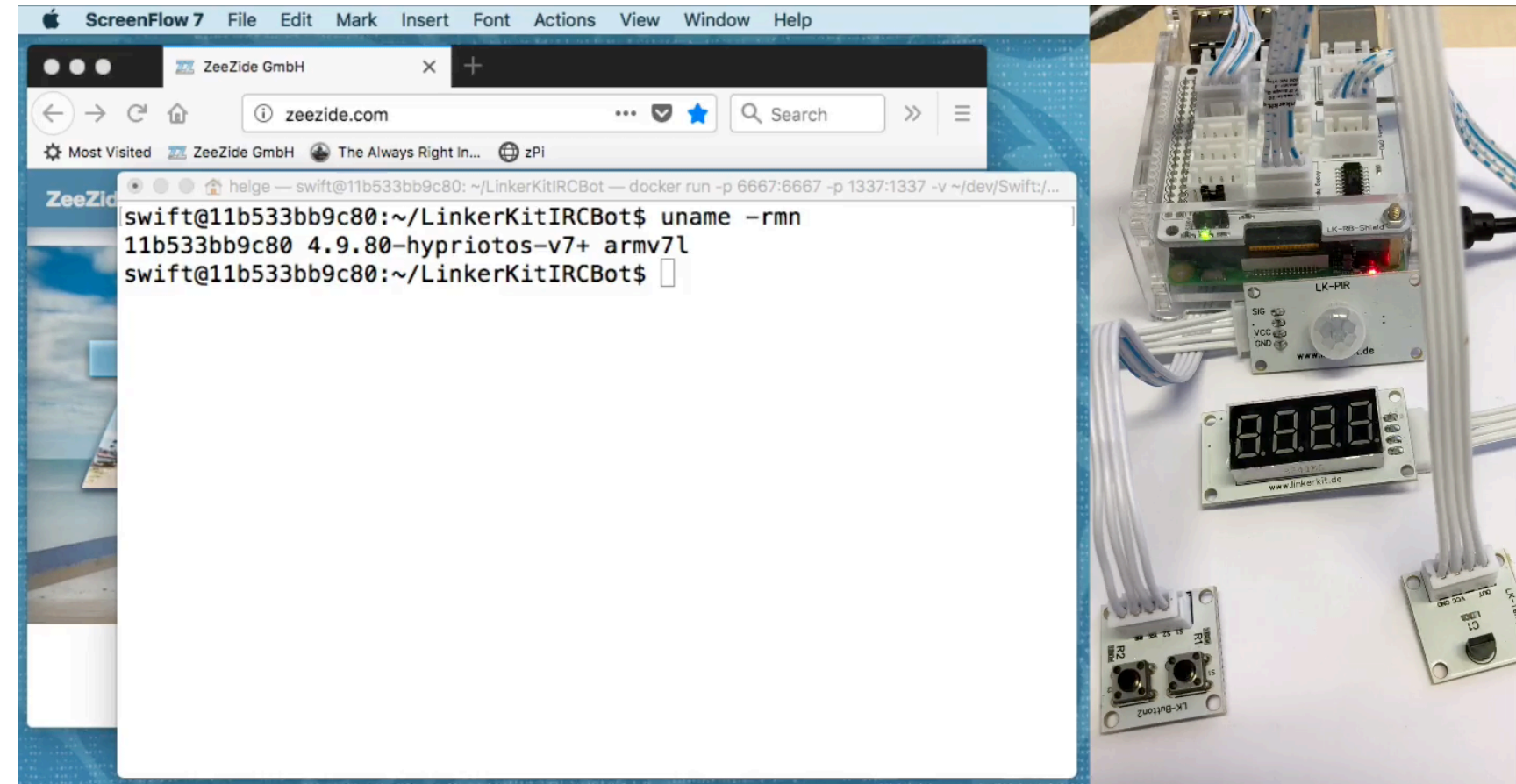
# Demo: SwiftyLinkerKit

The Internet of Things

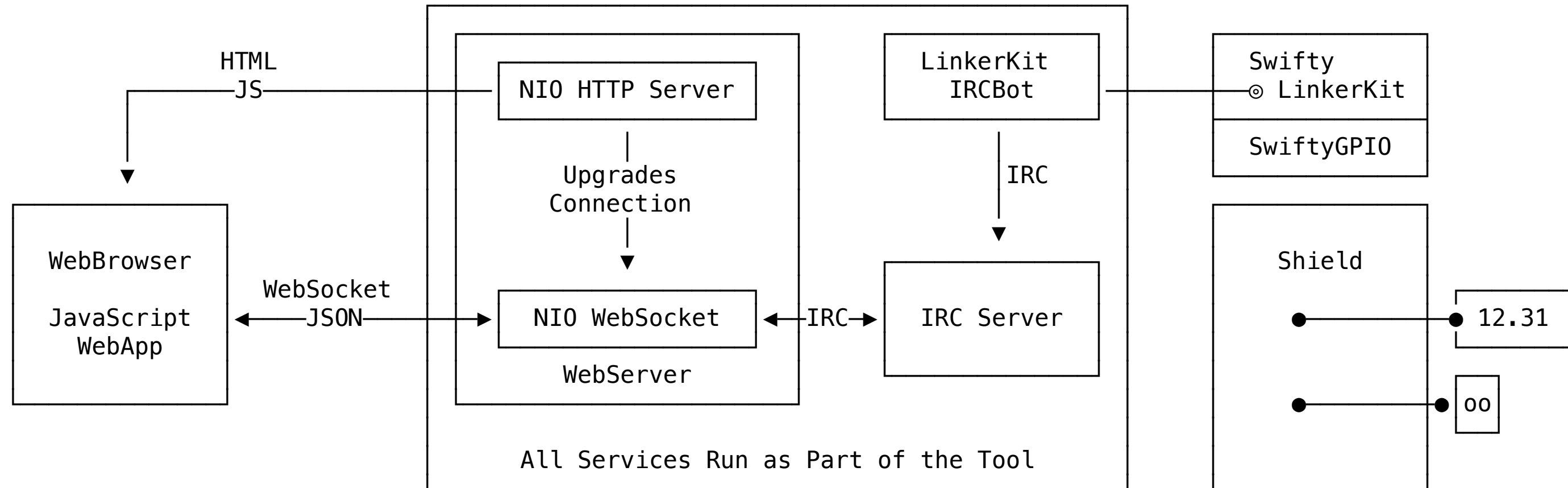
LinkerKitIRCBot

Blog: [Chat with your Raspi](#) (no need to read, has [videos](#) too!)

Purpose of Demo: Using Swift the Raspi can host complex software setups in very little memory.



# Demo: SwiftyLinkerKit



Try the IRC part yourself: <https://irc.noze.io>



# Why run Swift on ARM instead of Intel?

---

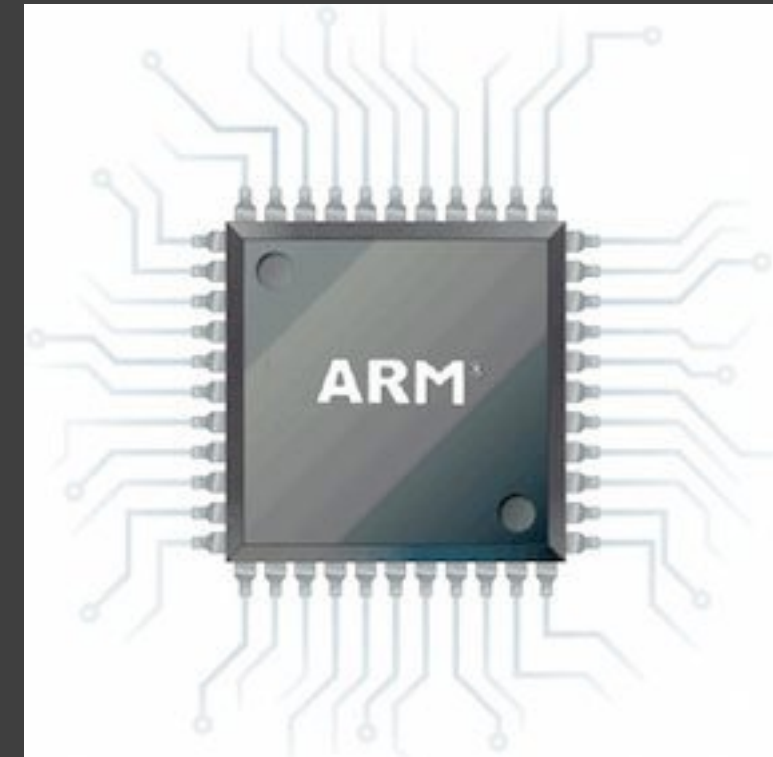
1. Price

---

2. Mobility

---

3. Power Efficiency



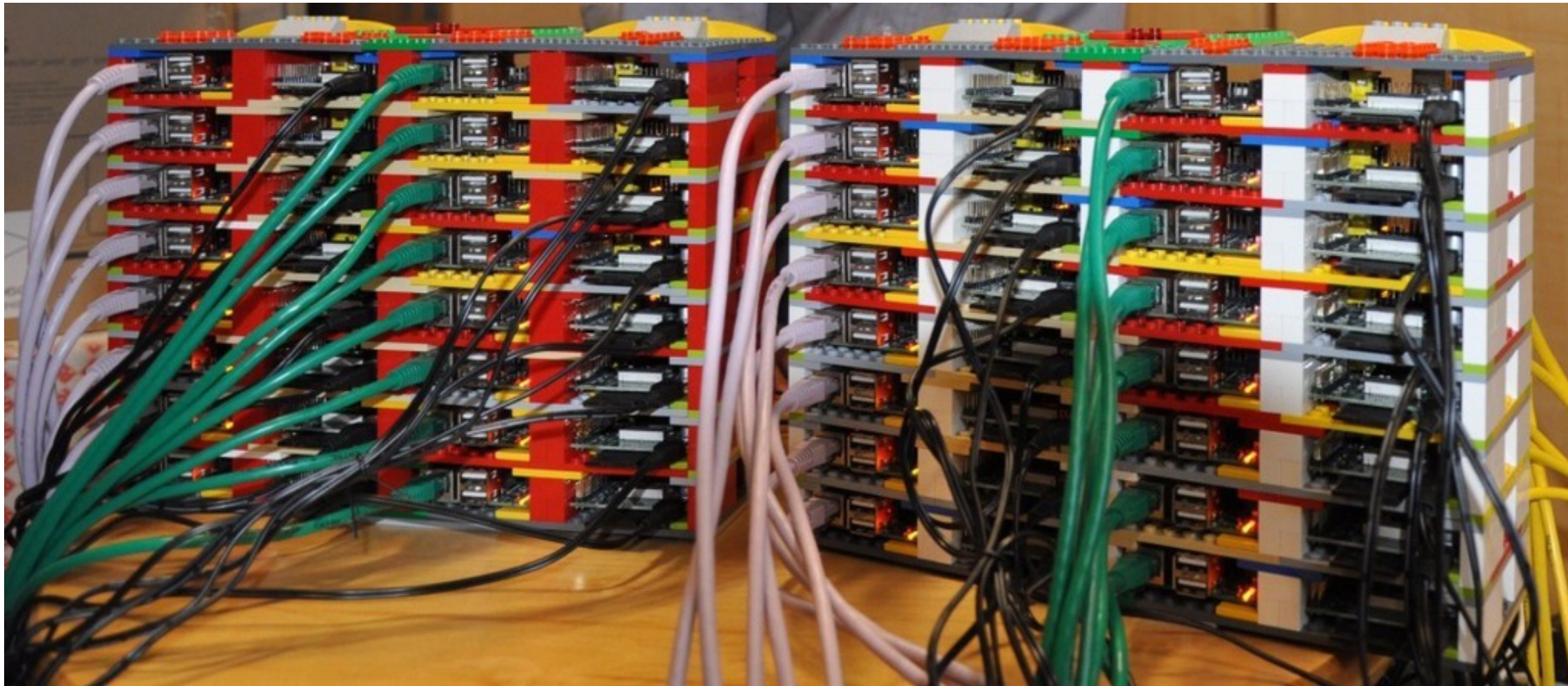
### 3. Power Efficiency

- Raspberry **Pi** 3B: idle: 1.3W, **load: 3.7W**, max 4.4W
  - Raspberry Pi Zero: max 0.7W (runs Swift!)
  - **MacPro idle: 80W**
- 👉 1 MacPro idle == 20 Pi 3B (80 Cores!) under load.

Who would run 20 Pis ... 🙈

### 3. Power Efficiency

Server devs! Docker Swarm / **K18s** Clusters. 64 Nodes.





## 2. Mobility

Runs on a power bank, anything USB, no power outlet needed!  
Or a small solar panel.

Can be used in places with fragile power infrastructure (🇺🇸...).

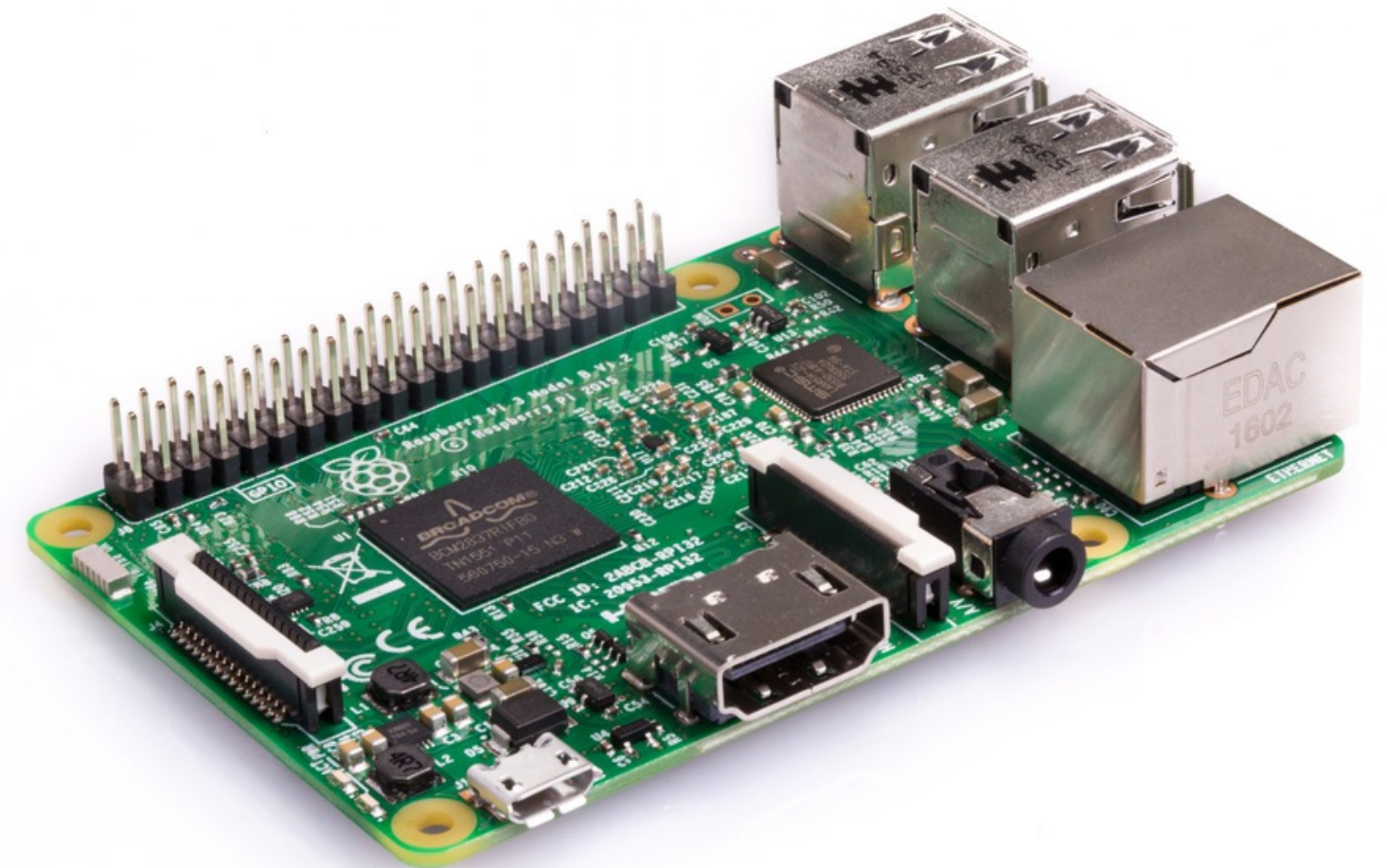
Take your server everywhere! (for example to a **Server-Side Swift Conference**).





# 1. Price of Hardware

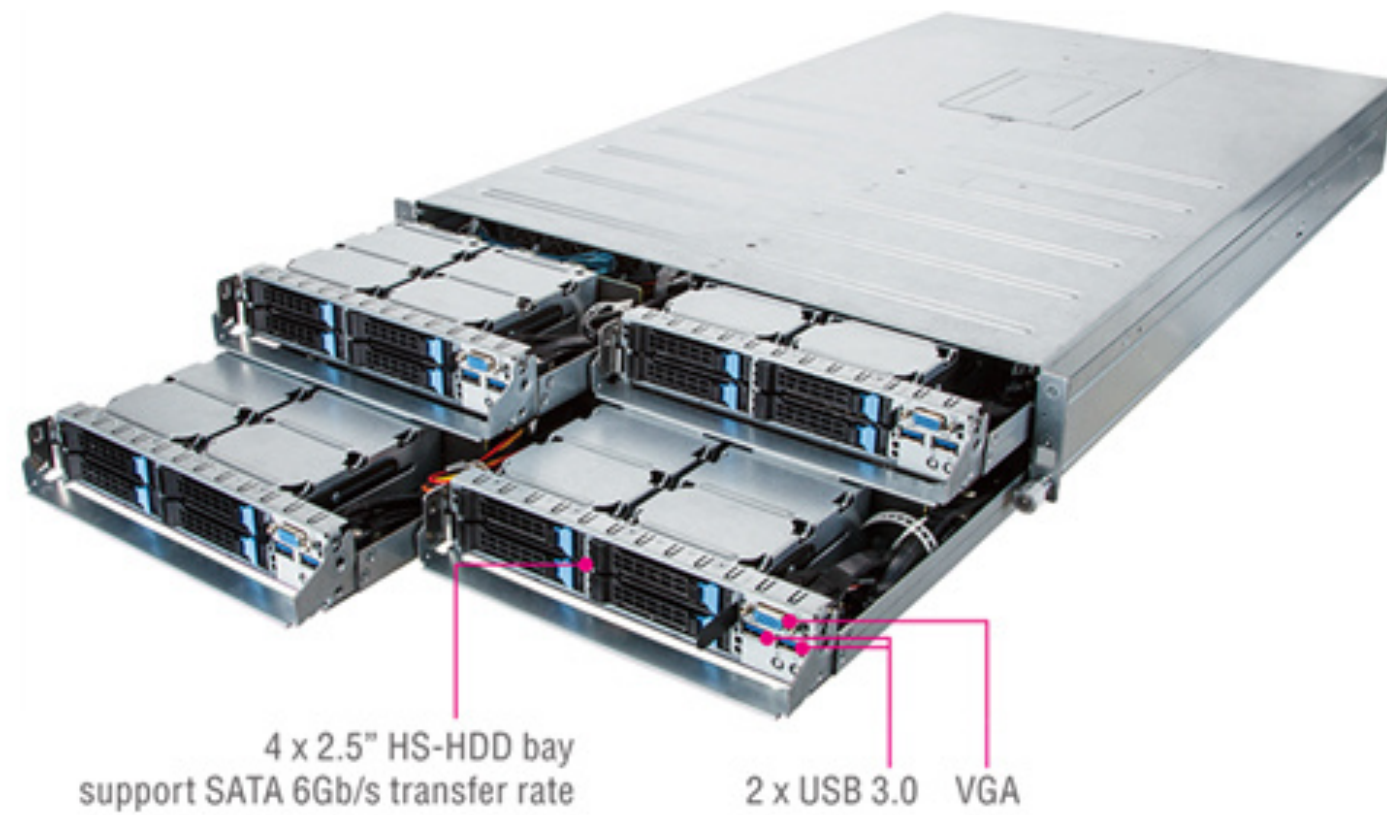
Model	Price
Pi 3B+	~35€
Pi Zero	~15€
Intel NUC	~120€



"Throw-away" device. Breaks?  
Replace it.

# 1. Price of Hardware "Datacenter Edition"

**Gigabyte Cavium ThunderX** ARM based server with **384 Cores** in a 2U case (2.5GHz). ~8k cores per rack!



# 1. Price of Hardware "Datacenter Edition" - Scaleway

Arm

ARM64-2GB

4 ARMv8 cores

2 GB memory

50 GB SSD disk

1 flexible public IPv4

200 Mbit/s unmetered bandwidth

€2.99

per month

Discover

BareMetal

C1

4 dedicated cores

2 GB memory

50 GB SSD Disk

1 flexible public IPv4

200 Mbit/s unmetered bandwidth

€2.99

per month

Discover

## Arm

Scaleway is the world's first cloud computing platform offering ARMv7 and ARMv8 SSD Cloud Servers built for developers.

Name	Cores	Memory	SSD Disk	Bandwidth	Price
ARM64-2GB	4 ARMv8	2 GB	50 GB	200 Mbits/s	€2.99/mo €0.006/hr
ARM64-4GB	6 ARMv8	4 GB	100 GB	200 Mbits/s	€5.99/mo €0.012/hr
ARM64-8GB	8 ARMv8	8 GB	200 GB	200 Mbits/s	€11.99/mo €0.024/hr
ARM64-16GB	16 ARMv8	16 GB	400 GB	500 Mbits/s	€34.99/mo €0.070/hr
ARM64-32GB	32 ARMv8	32 GB	600 GB	500 Mbits/s	€69.99/mo €0.140/hr
ARM64-64GB	48 ARMv8	64 GB	800 GB	1 Gbit/s	€139.99/mo €0.280/hr
ARM64-128GB	64 ARMv8	128 GB	1 TB	1 Gbit/s	€279.99/mo €0.560/hr

Our prices are VAT exclusive.

# Cross Compilation

- Build a binary for any target directly on your Mac
  - Like Xcode builds an iPhone binary
- Advantage: Build much faster on a Mac.
- Johannes Weiß added **toolchain support** to SwiftPM
- Added a **toolchain for Raspberry Pi** based on that
  - And a **macOS toolchain that runs on RasPi**
- BUT: b0rked in 4.1, fix coming in 4.2. works w/ 3.1



## Contact Helge ★ Q & A

Twitter: @helje5

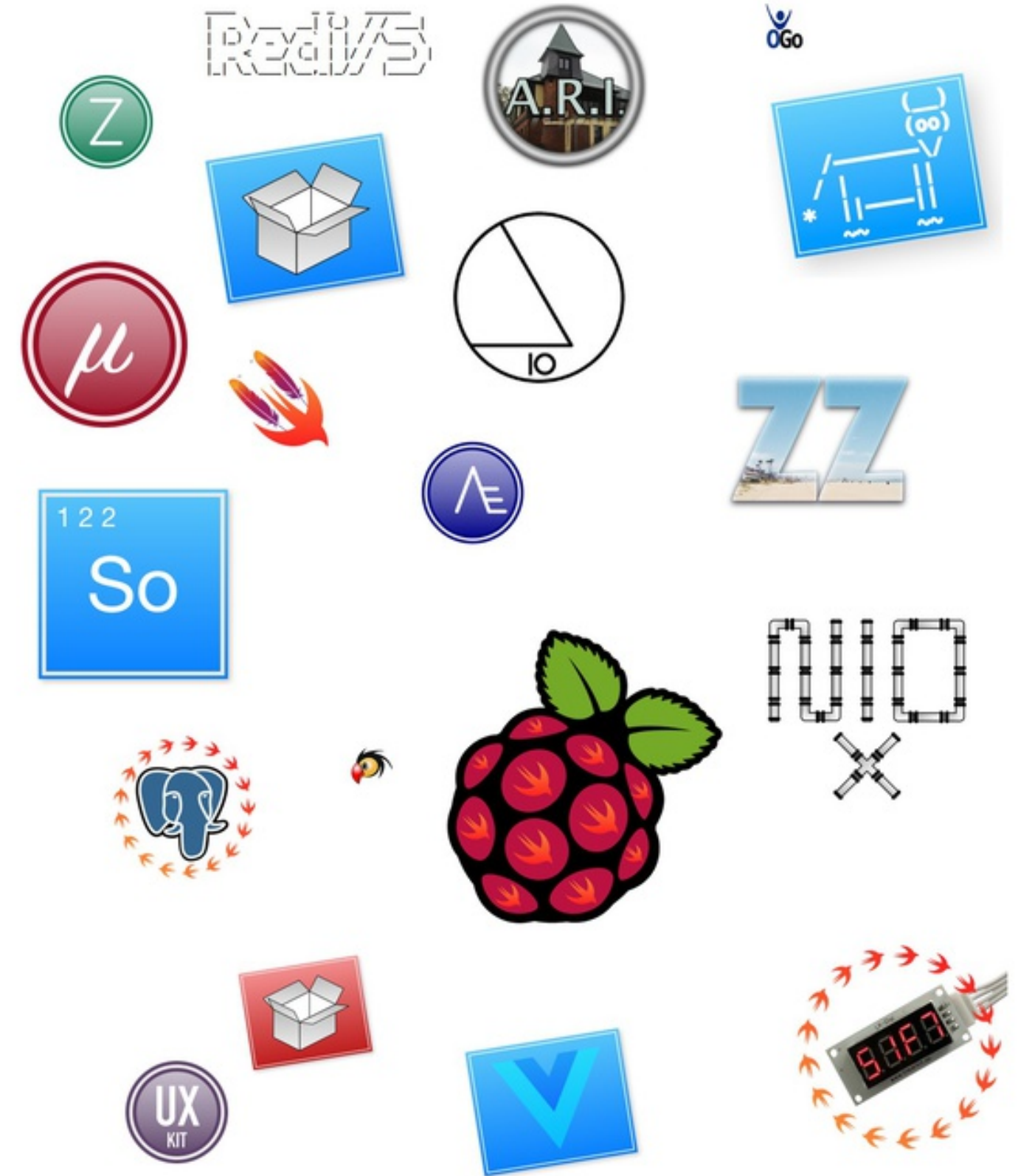
Mail: [me@helgehess.eu](mailto:me@helgehess.eu)

Web: [www.helgehess.eu](http://www.helgehess.eu)


GitHub: [helje5](#)

Business: ZeeZide GmbH

Blog: [AlwaysRightInstitute](https://www.alwaysrightinstitute.com/)



# OpenSource Projects

Redi/S,  $\mu$ Express, swift-nio-irc, SwiftNIO Extras,  
SwiftXcode, mod\_swift, SwiftObjects, ApacheExpress,  
Noze.io, ZeeQL, SwiftyLinkerKit, SwiftMon/S,  
SwiftSockets, UXKit, PL/Swift, SwiftWebResources,  
VueXcode, dockSwiftOnArm, macOS->RasPi, Cows ,  
WebpackMini/S, mustache, libFoundation, GNUstep,  
OGo, ScalableOGo, SOPE, swifter-lang, GroupDAV,  
mod\_objc1, GTKKit, StaticCMS, ...

# Image Sources

- Datacenter node: Photo by [Thomas Kvistholt](#) on Unsplash
- Apple iCloud Data Center: [Apple](#)
- CCU 3 Images by <https://technikkram.net>
- ARM CPU Image: <https://hothardware.com/news/arm-details-armv8-64bit-architecture>
- Raspi Lego Cluster: Simon Cox