My obsession with Arm — Entering Apple M series!

I bought the Mac Mini M1 in 2020 which was the transition time for several popular developer tools to support ARM64 architecture in macOS. Incredible to see that it did not take a year also to see most of the packages from Homebrew to the docker engine available for Apple Silicon. By 2022, Apple M series processors became so popular and widely used. Since then, Apple M Series Arm-based processor has become the de-facto one at my home. It has a butter-smooth performance, is noise-free, no heating issues and I can keep going with its pros. Just go for it!! No regrets!!

Search for Cloud Compute Instance

It started with my personal hobby project work. As a full-stack developer, I just wanted to explore, and learn modern cloud-native tools & patterns. With a basic web service ready locally, I was looking for a cloud-based compute instance. There are several trial services and free credits available from different providers such as Digital Ocean, AWS, Linode, etc. However, my personal hobby project is not just for learning, I wanted to host it and keep it alive for a long time. I didn't want to spend hundreds of dollars as I am not doing any business, neither there is any RoI. I am a certified AWS Solution architect too and hence my obvious first choice is to go with AWS EC2 which is comparatively expensive to run beyond the trial period.

While searching for the cheapest(~ $10 pm), yet reliable cloud compute instances, there were few popular options.

Choice1:

EC2 T3/T4 Micro -Very low resource, ok for learning, but really not usable for long-term deployment.

Choice2 — selected:

Other popular options like Linode, Upcloud, Digital Ocean, etc. were there. More or less the same fixed price, and it was ok to start with. So, I chose the most popular one — Digital Ocean **Droplets.**

Signed up! Added credit card, Happy purchase!!.

The elephant in the room: X86 vs Arm64

I had my container images generated locally and stored in GitLab's & docker hub. I used both kubectl (K8) & docker compose to deploy! Boom — Failed!! There revealed the obvious elephant in the room. I was happily creating docker images from my local Apple M1 system, which is Arm64 based. The Digital ocean droplet I purchased was Intel X86 based. Then, I went through the docker documentation and found the 'docker buildx', a tool to build multi-architecture docker images. I have made a separate blog on how to use it here. But, the availability of matured & verified base images built and published specifically for Arm64 is very less compared to X86 based.

This workaround was ok, but there were several instances where the buildx command failed, and I always had to google and fix it.

Back to AWS:

Digital ocean was ok, but as an AWS developer, I wanted to use different seamless integration options within AWS such as SES, Cognito, gateway, CF etc. Realized that the low-cost digital ocean is ok only for basic web projects. I wanted some solution for the long term and hence checked for a few savings general purpose instance plans in AWS.

t4g.small — 2GB, 2Vcpu, $0.0134/hr , 10$ / pm
t3.small — 2GB, 2 Core, $0.0166 / hr, 12.55$/ pm

Wow!! Graviton(t4G) made my life easy. The advantage is not about the negligible difference in cost per month. This graviton instance is arm64 based. Yaayyy!!!! Now, I can simply build from my Apple Mac M1 and deploy it straight into AWS EC2. No more workaround.

Benefits of Graviton Processor

Is cost the only deciding factor? No!! There are more reasons. One of the highlights of any Arm-based processor is its energy efficiency (same as why my Mac M1 is super silent and so our powerful smartphones).

As per AWS documentation,

- Up to 40% better price performance over comparable current generation x86-based instances.
- Graviton3-based instances use up to 60% less energy for the same performance than comparable EC2 instances.
- Reduce their carbon footprint
- Cost Efficient

How about a Performance Benchmark?

I found some interesting article that shows better performance while using arm64 instances. Eg. below is from, **Tigeria**

Image from Tigeria

Instana, an **IBM** company also talks about their support for arm64 and how the future will be arm64.

Mac Mac Mac. How about Windows? — Project Volterra

So, the usage and support of ARM-based processors are not just limited to smartphones and the Apple ecosystem. Even Microsoft knows the future and hence unveiled the Windows Dev Kit 2023 a.k.a. **Project Volterra** , an Arm-powered device built by Windows developers for Windows developers.Ref: **https://www.microsoft.com/en-us/d/windows-dev-kit-2023/94K0P67W7581?activetab=pivot:overviewtab**

Other popular ARM instances

- **Oracle** used Arm Ampere A1 Compute instance, which offers generous **free** hour usage.
- **Google** Cloud launched its ARM-based compute recently — **Tau T2A**
- **Azure** launched **Ampere Altra** Arm-based compute instance earlier this year and is now available generally.

It is very exciting to see the transition in the enterprise and clearly, with hybrid, edge computing, IoT, and 5g demands, the energy need is increasing rapidly. ARM-based cloud instances would definitely take over the traditional AMD & Intel X86 processors.

Let me know your thoughts:)