



aircall

The phone system for modern business

Create multi-architecture docker images

Hassen Harzallah

February 2023

Hassen Harzallah



@hhassen

Site Reliability Engineer at Aircall

- Cloud Infrastructure
- Monitoring
- CI/CD
- Security

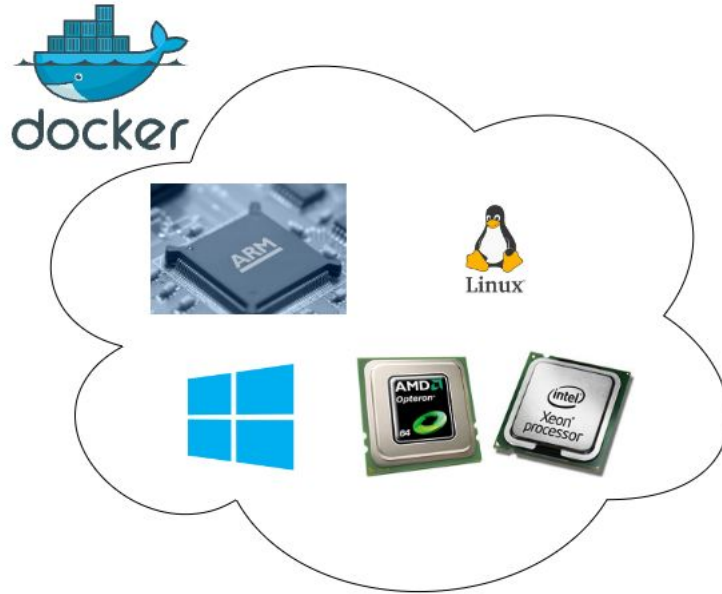


The phone system for modern business

Create multi-architecture docker images



But I only work on one architecture ?



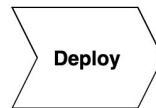
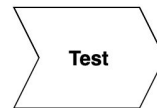
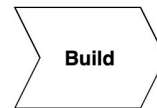
How we work in Aircall

In Aircall, we are using a lot of lambda functions for our applications.



AWS Lambda used to run on **AMD64 (== x86)** architecture, as well all their CI/CD workflows.

Developers use their **AMD64** computers to develop and test lambdas.



Dev team :
AMD64 arch

CICD + AWS lambda :
AMD64 arch



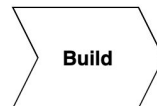
We started to need more architectures

AWS Lambda is proposing a new runtime powered by **Graviton2 Processors (ARM64)**:

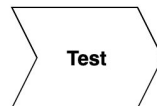
- **20 percent lower cost** (duration charge)
- Possible **performance improvements**



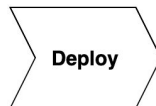
Code



Build



Test



Deploy



Lambda Function

Dev team :
AMD64 arch

CICD + AWS lambda :
ARM64 arch



We started to need more architectures

- We had to **change our workflows** to adapt newer architectures.
- We added new conditions, new environment variables ...
- This added **more complexity** to our workflow



What is a docker manifest ?

Each Docker image is represented by a **manifest**.

It's a JSON file containing these information :

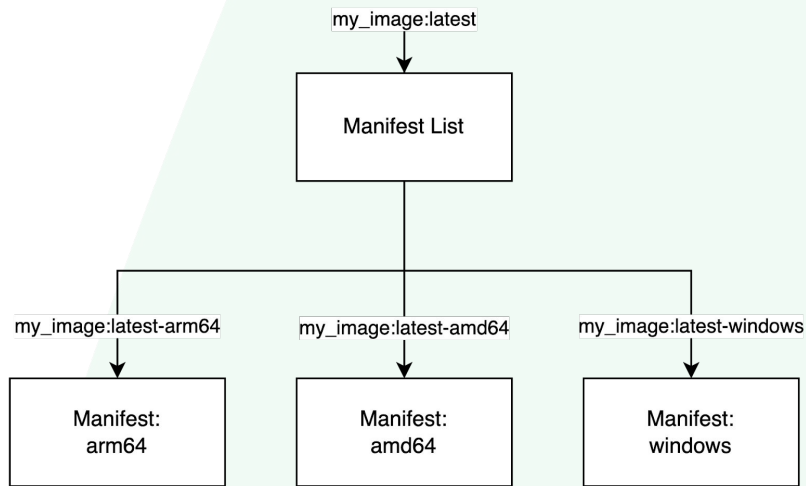
- Layers
- Layers size
- Hash of the image
- Configuration
- OS
- Architecture
- ...

```
$ docker inspect ubuntu:latest
[
  {
    "Id": "sha256:d63f752103bb93d846e17fa9996d3e708717c51b106382fe84d8527ee47a3547",
    "RepoTags": [
      "ubuntu:latest"
    ],
    "RepoDigests": [
      "ubuntu@sha256:35fb073f9e56eb84041b0745cb714eff0f7b225ea9e024f703cab56aaa5c7720"
    ],
    "ContainerConfig": {
      "Hostname": "bce4bc519485",
      "Env": [
        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
      ],
      "Cmd": [
        "/bin/sh",
        "-c",
        "#(nop) ",
        "CMD [\"bash\"]"
      ],
      "Image": "sha256:379e20d15303eb4e3c8df12fe357e89c9c31b5c923521a39eabd682d4c219755",
      "Entrypoint": null,
      "OnBuild": null,
      "Labels": {}
    },
    "Architecture": "arm64",
    "Variant": "v8",
    "Os": "linux",
    "Size": 69205455,
    "VirtualSize": 69205455,
    "RootFS": {
      "Type": "layers",
      "Layers": [
        "sha256:ede2ae06e2f45441ad8bfadd13b072d47e9f2d3adaadabf60e3b0f4a6b6b7723"
      ]
    },
    "Metadata": {
      "LastTagTime": "0001-01-01T00:00:00Z"
    }
  }
]
```



What is a multi-architecture image ?

Multi-arch image = **manifest list** that points to different manifests



```
$ docker manifest inspect alpine:latest
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",
  "manifests": [
    {
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "size": 528,
      "digest": "sha256:93d5a28ff72d288d69b5997b8ba47396d2cbb62a72b5d87cd3351094b5d578a0",
      "platform": {
        "architecture": "amd64",
        "os": "linux"
      }
    },
    {
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "size": 528,
      "digest": "sha256:01a4cdaebc9c6af607753cc538c507d0867897cdf9a1caa70bbab2eb1506c964",
      "platform": {
        "architecture": "arm",
        "os": "linux",
        "variant": "v6"
      }
    },
    {
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "size": 528,
      "digest": "sha256:1c34b3cb760a98c23361d919357b99fa497074576c898e7289425d45ef67b46a",
      "platform": {
        "architecture": "arm",
        "os": "linux",
        "variant": "v7"
      }
    },
    {
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "size": 528,
      "digest": "sha256:41d876d4e44348d1c27445fdb0e64592e0eb926d4dbbcf09a3526dee7e628329",
      "platform": {
        "architecture": "arm64",
        "os": "linux",
        "variant": "v8"
      }
    }
  ]
}
```



Build multi-architecture images

There are different ways to build multi-architecture images:



The hard way :

docker manifest



The easy way :

docker buildx

docker



The Aircall way :

Kaniko on **Kubernetes**

docker in docker



Tip: add a new ARCH variable in Dockerfile



```
ARG ARCH=  
FROM ${ARCH}node:18-slim  
  
WORKDIR /app  
COPY ["package.json", "package-lock.json*", "./"]  
RUN npm install  
COPY . .  
CMD [ "node", "server.js" ]
```



docker manifest

myimage:latest-amd64

myimage:latest-arm64

myimage:latest

```
# Build & Push Docker images for each architecture
# AMD64
$ docker build -t ${MY_REPO}/myimage:latest-amd64 --build-arg ARCH=amd64/ .
$ docker push ${MY_REPO}/myimage:latest-amd64

# ARM64V8
$ docker build -t ${MY_REPO}/myimage:latest-arm64v8 --build-arg ARCH=arm64v8/ .
$ docker push ${MY_REPO}/myimage:latest-arm64v8

# Create the manifest list
$ docker manifest create \
  ${MY_REPO}/myimage:latest \
  --amend ${MY_REPO}/myimage:latest-amd64 \
  --amend ${MY_REPO}/myimage:latest-arm64v8

# Push the manifest list
$ docker manifest push ${MY_REPO}/myimage:latest
```



docker buildx

```
$ docker buildx build \  
--push \  
--platform linux/arm/v7,linux/arm64/v8,linux/amd64 \ --tag ${MY_REPO}/multiarch-example:buildx-latest .
```

Buildx allow to emulate different architectures on the same machine.



Multi-arch and Kaniko ?

In Aircall, we are using **Kaniko** to build Docker images inside Kubernetes

Kaniko doesn't build multi-architecture images

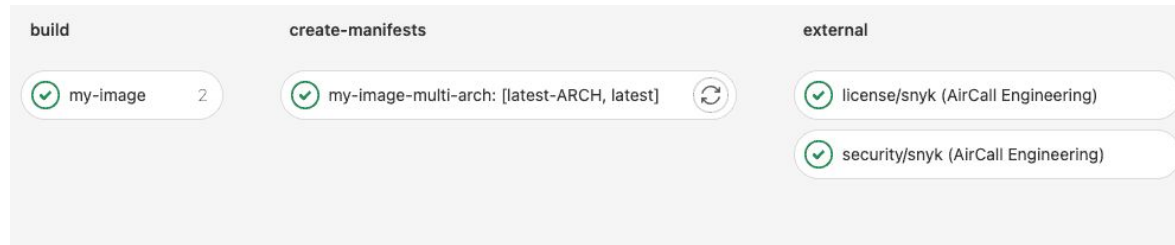
We used another tool : **manifest-tool**

(<https://github.com/estesp/manifest-tool>)



Our new process to build a docker image :

1. **Build docker images for each architecture** using Dockerfile and Kaniko
2. **Push docker image** of each architecture using different tags (eg. latest-amd64)
3. **Create & Push a manifest list** assembling the different images already pushed
4. **Security scans**



Takeaways

Multi-arch :

- Use **new generation processors** and profit from **lower prices**
- Have a **unique path** for all our docker images
- **Simplify** our CI/CD and development workflow



Useful articles:

- <https://aircall.io/blog/tech-team-stories/supporting-multi-architecture-container-images/>
- <https://aws.amazon.com/blogs/aws/aws-lambda-functions-powered-by-aws-graviton2-processor-run-your-functions-on-arm-and-get-up-to-34-better-price-performance/>
- <https://www.docker.com/blog/how-to-rapidly-build-multi-architecture-images-with-buildx/>





Thank you!