# Akraino Enables Multi-Architecture Support for Edge Workloads through Docker Hub

Case Study

## Organization Snapshot

**LF Edge**

LF Edge is an umbrella organization that aims to establish an open, interoperable framework for edge computing independent of hardware, silicon, cloud, or operating system.

**lfedge.org**

## Goal

Offer developers seamless support for creating and pushing multi-architecture (multi-arch) container images by implementing a multi-arch parallel CI/CD pipeline for edge to cloud applications.

## Solution

The manifest list (a.k.a. "fat manifest") functionality in the Docker Hub registry is to introduce support for multi-arch images giving developers ability to use the same name when pulling images built on different hardware architectures. Prior to introduction of manifest list, developers had to either pull the image by its arch specific tag to indicate the platform for which an image was built or had to create Docker Hub namespaces for each architecture type creating fragmentation and non-standardization.

## Benefits

- True multi-architecture workload portability
- Increases developer's productivity
- Improves end-user experience

**Akraino Edge Stack** is one of the five main projects under LF (Linux Foundation) Edge, an umbrella organization under Linux Foundation, tasked to create open interoperable framework for edge computing devices. Using containers is gaining popularity amongst developers to develop cloud native software for edge to cloud applications. **Docker** is the leading container platform technology used by millions of developers and combined with **Docker Hub**, the world's largest cloud-based public repository (a.k.a registry) for sharing packaged container images, tremendously increases developer productivity for deploying services for edge applications like drones, autonomous vehicles, AR/VR, IIoT, smart home devices, healthcare, etc.

This case study shows how the Akraino Edge Stack project successfully integrated Docker Hub in one of its blueprints, and used the manifest list functionality to enable multi-arch support enabling developers to seamlessly port and maintain edge workloads for Arm Neoverse servers.

## Integrating Docker Hub Multi-arch Support in CI

Following the Akraino Edge Stack project example, one can integrate Docker Hub in a CI environment in their own project with the following two steps:

1. Install Docker on the system running CI software (Jenkins slave in Akraino example)
2. Replicate the JJB (Jenkins Job Builder) scripts in project's CI environment for:
   a. Building and pushing multi-arch images (for example: *akraino/validation:k8s-amd64- latest and akraino/validation:k8s-arm64-latest*)
   b. Building and pushing the manifest list based on those images with the same names removing the architecture type name from the tag (for example: *akraino/validation:k8s-latest*).

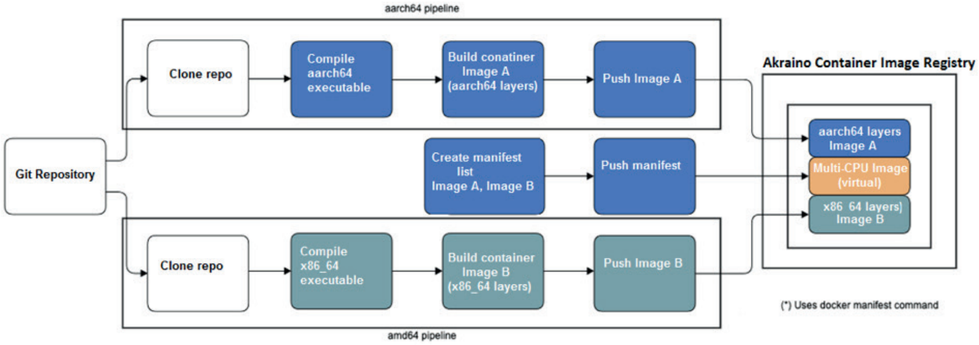
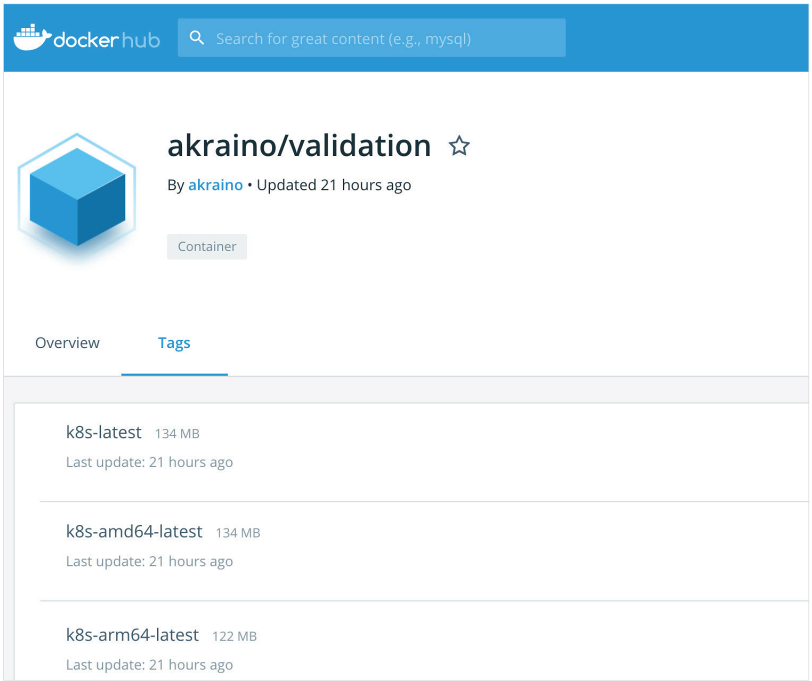

aarch64 pipeline
Git Repository
Clone repo
Compile aarch64 executable
Build conatiner Image A (aarch64 layers)
Push Image A
Create manifest list Image A, Image B
Push manifest
Clone repo
Compile x86_64 executable
Build container Image B (x86_64 layers)
Push Image B
amd64 pipeline
Akraino Container Image Registry
aarch64 layers Image A
Multi-CPU Image (virtual)
x86_64 layers) Image B
(*) Uses docker manifest command

Figure 1: Multi-arch CI/CD Pipeline Implementation Example

Source: Akraino Multi-Architecture Support Presentation on Akraino LF Edge Site

## Example: Akraino Validation Multi-arch project

The Akraino Blueprint Validation project is already pushing multi-arch images to the official Docker Hub registry as well as the manifest list. The Jenkins build jobs run daily creating these multi-arch images.



akraino/validation ☆

By akraino • Updated 21 hours ago

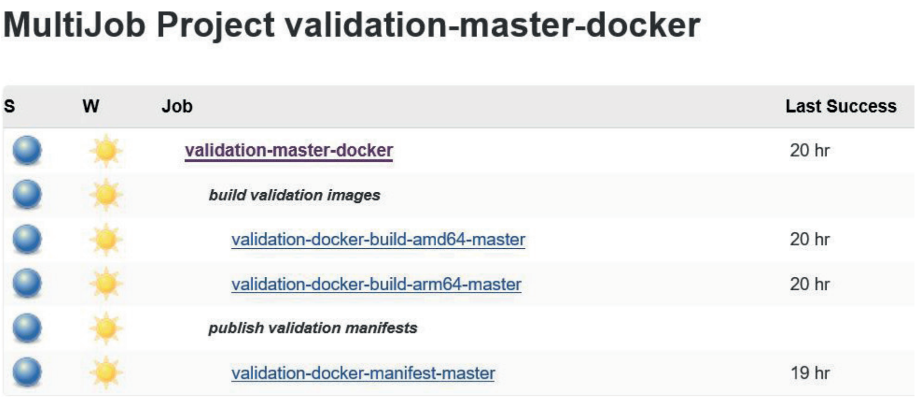
Container

k8s-latest 134 MB
Last update: 21 hours ago

k8s-amd64-latest 134 MB
Last update: 21 hours ago

k8s-arm64-latest 122 MB
Last update: 21 hours ago

Figure 2: Akraino Validation Project on Docker Hub



## MultiJob Project validation-master-docker

| S | W | Job | Last Success |
|---|---|---|---|
| ● | ☀ | validation-master-docker | 20 hr |
| ● | ☀ | *build validation images* | |
| ● | ☀ | validation-docker-build-amd64-master | 20 hr |
| ● | ☀ | validation-docker-build-arm64-master | 20 hr |
| ● | ☀ | *publish validation manifests* | |
| ● | ☀ | validation-docker-manifest-master | 19 hr |

Figure 3: Jenkins Build Jobs creating multi-arch images

When the developer wants to pull an image from Docker Hub, the name of the manifest is used (for example: akraino/validation:k8s-latest) and the correct image will be pulled based on the host architecture from which the pull command is initiated.

For more details on how this works please refer to the LF Edge blog post.

### Conclusion

True multi-platform workload portability has been the quintessential goal for edge to cloud computing applications. Any open source project, and not just projects under Linux Foundation, developing container images and using Docker Hub as the registry can achieve multi-arch support by following Akraino LF Edge Validation project described in this case-study.