Using Docker Hub's public registry for CI/CD builds - An ONAP use-case

Paul Vaduva
Cristina Pauna
Jessica Wagantall

ARM, ENEA, Linux Foundation

Agenda

1. Brief introduction on Docker Hub

2. Using Docker Hub for CI/CD in LF projects

3. ONAP use-case
   - Limitations and challenges
   - Possible solutions
   - Questions and concerns

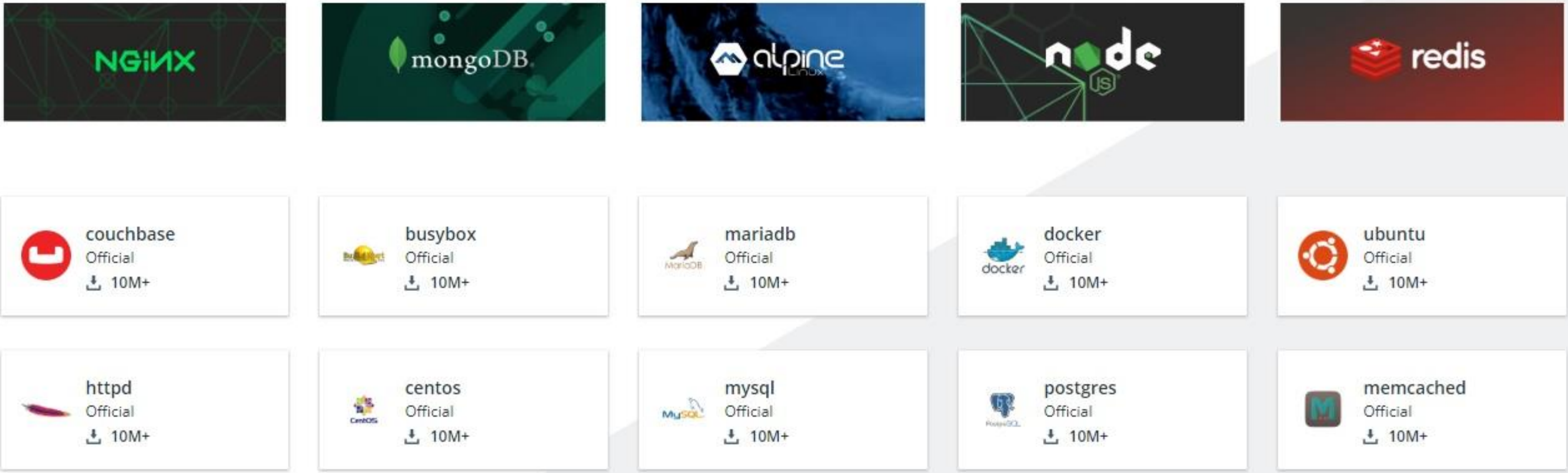
1. Brief introduction on Docker Hub

What is Docker Hub ?

- Docker Hub is the world's largest library and community for container images
- It offers a huge repository for storing container images
- It is available world-wide
- It can automatically build container images from GitHub and Bitbucket and push them to Docker Hub
- It offers seamless support for multi-arch images through fat manifest

Who uses Docker Hub ?

- All major open source projects use Docker Hub to release their software including LF projects OPNFV and ONAP



couchbase
Official
10M+

busybox
Official
10M+

mariadb
Official
10M+

docker
Official
10M+

ubuntu
Official
10M+

httpd
Official
10M+

centos
Official
10M+

mysql
Official
10M+

postgres
Official
10M+

memcached
Official
10M+

2. Using Docker Hub for CI/CD in LF projects

How LF projects use Docker Hub

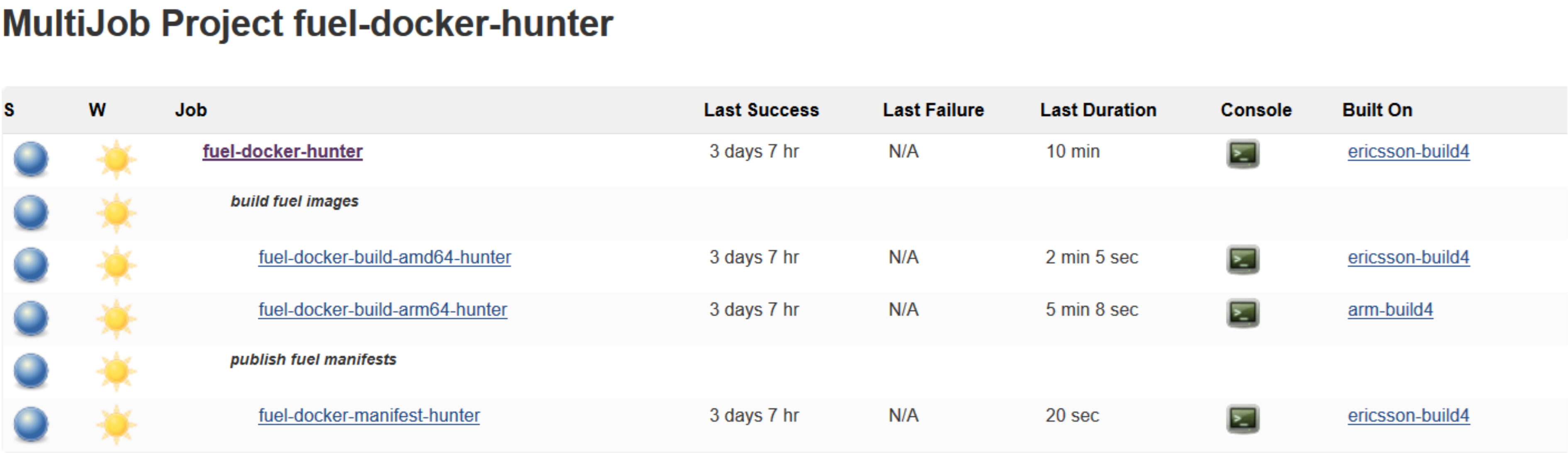
- LF projects have multiple labs and their own infrastructure when it comes to handle docker images. Therefore, out of the multiple features that Docker Hub offers, CI pipelines in LF focuses on these three:
    - Build docker images with CI jobs and push them to the Docker Hub registry
    - Pull docker images in CI jobs to run certain features or perform testing
    - Provide release images

- Integration of Jenkins with Docker Hub is handled by LF and it’s quick and easy to configure

- The jobs that run in Jenkins can afterwards call push and pull commands to docker

Building images for Docker Hub

- When building the docker images in CI, complex relationships can be defined:
    - build images in a certain order
    - build multiple images in parallel

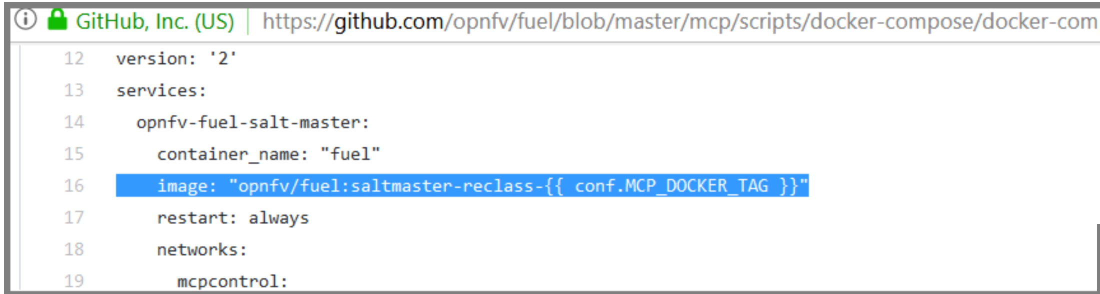


MultiJob Project fuel-docker-hunter

S	W	Job	Last Success	Last Failure	Last Duration	Console	Built On
●	☀	fuel-docker-hunter	3 days 7 hr	N/A	10 min	>_	ericsson-build4
●	☀	*build fuel images*					
●	☀	fuel-docker-build-amd64-hunter	3 days 7 hr	N/A	2 min 5 sec	>_	ericsson-build4
●	☀	fuel-docker-build-arm64-hunter	3 days 7 hr	N/A	5 min 8 sec	>_	arm-build4
●	☀	*publish fuel manifests*					
●	☀	fuel-docker-manifest-hunter	3 days 7 hr	N/A	20 sec	>_	ericsson-build4

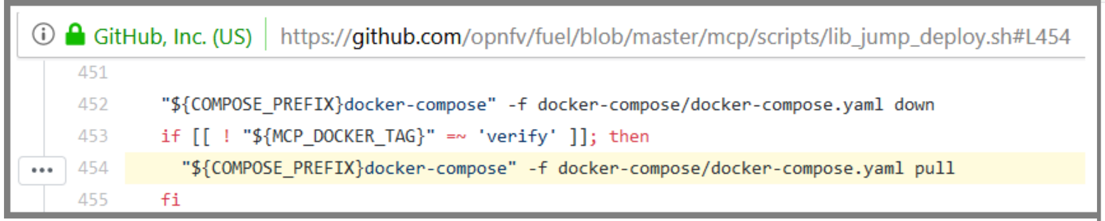
Using images from Docker Hub

- When using the docker images in CI, just pull and use



https://github.com/opnfv/fuel/blob/master/mcp/scripts/docker-compose/docker-compose.yaml.j2

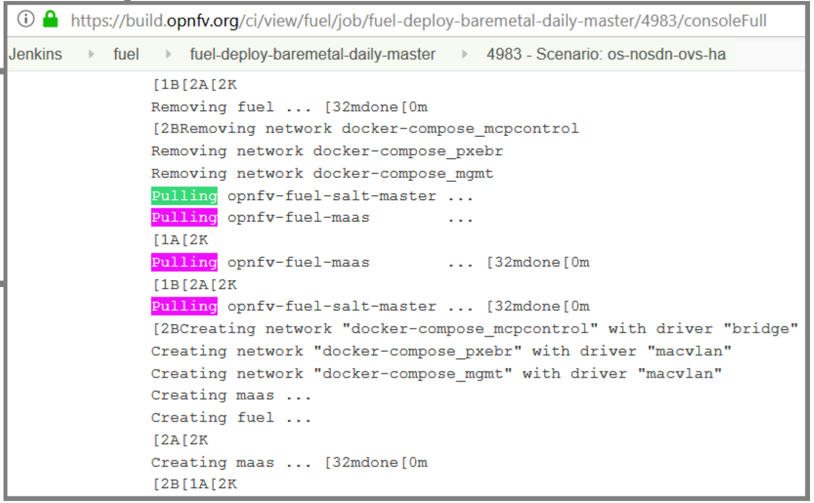
```
version: '2'
services:
  opnfv-fuel-salt-master:
    container_name: "fuel"
    image: "opnfv/fuel:saltmaster-reclass-{{ conf.MCP_DOCKER_TAG }}"
    restart: always
    networks:
      mcpcontrol:
```



https://github.com/opnfv/fuel/blob/master/mcp/scripts/lib_jump_deploy.sh#L454

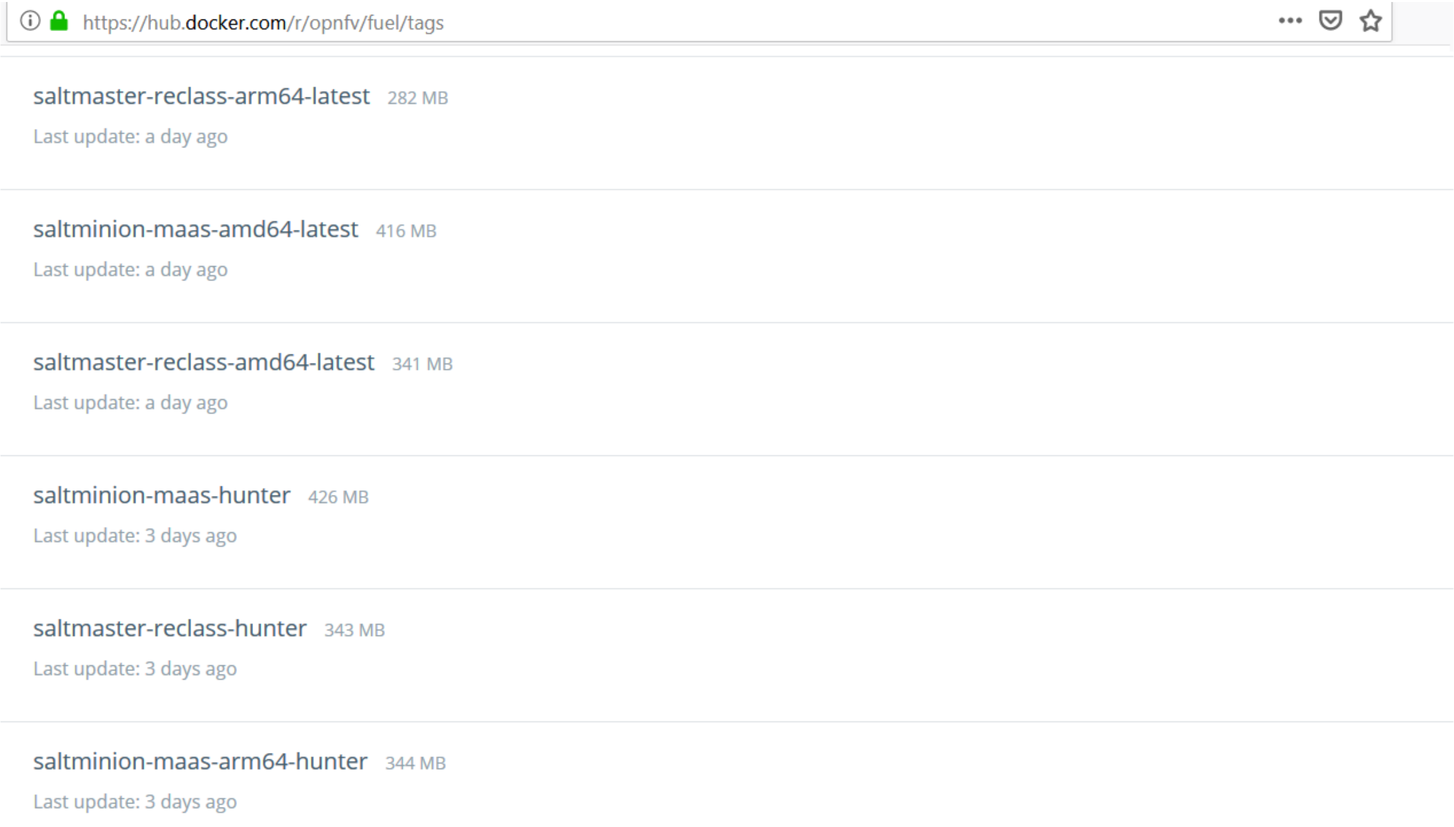
```

"${COMPOSE_PREFIX}docker-compose" -f docker-compose/docker-compose.yaml down
if [[ ! "${MCP_DOCKER_TAG}" =~ 'verify' ]]; then
  "${COMPOSE_PREFIX}docker-compose" -f docker-compose/docker-compose.yaml pull
fi
```


https://build.opnfv.org/ci/view/fuel/job/fuel-deploy-baremetal-daily-master/4983/consoleFull
Jenkins ▸ fuel ▸ fuel-deploy-baremetal-daily-master ▸ 4983 - Scenario: os-nosdn-ovs-ha

```
[1B[2A[2K
Removing fuel ... [32mdone[0m
[2BRemoving network docker-compose_mcpcontrol
Removing network docker-compose_pxebr
Removing network docker-compose_mgmt
Pulling opnfv-fuel-salt-master ...
Pulling opnfv-fuel-maas        ...
[1A[2K
Pulling opnfv-fuel-maas        ... [32mdone[0m
[1B[2A[2K
Pulling opnfv-fuel-salt-master ... [32mdone[0m
[2BCreating network "docker-compose_mcpcontrol" with driver "bridge"
Creating network "docker-compose_pxebr" with driver "macvlan"
Creating network "docker-compose_mgmt" with driver "macvlan"
Creating maas ...
Creating fuel ...
[2A[2K
Creating maas ... [32mdone[0m
[2B[1A[2K
```

Provide release images



https://hub.docker.com/r/opnfv/fuel/tags

saltmaster-reclass-arm64-latest 282 MB

Last update: a day ago

saltminion-maas-amd64-latest 416 MB

Last update: a day ago

saltmaster-reclass-amd64-latest 341 MB

Last update: a day ago

saltminion-maas-hunter 426 MB

Last update: 3 days ago

saltmaster-reclass-hunter 343 MB

Last update: 3 days ago

saltminion-maas-arm64-hunter 344 MB

Last update: 3 days ago

3. ONAP use-case

- Limitations and challenges

Current state of daily CI/CD in ONAP

- Currently ONAP Projects use Nexus3 docker registry implementation to store the intermediate images / Staging

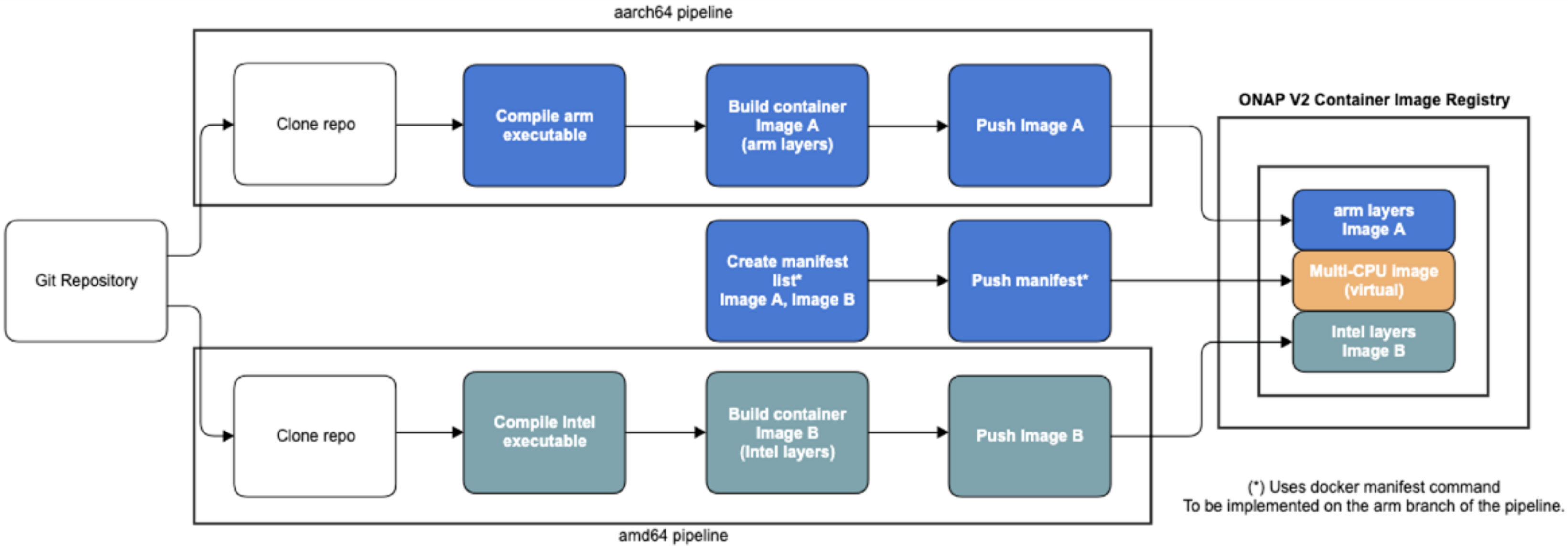
- ONAP Projects sometimes build docker images that have dependencies on other ONAP Projects docker builds. This multiple source dependencies issue is solved in LF by grouping together docker registries in Nexus3

- ONAP already uses Docker Hub to make available the release images, as it’s the largest library and community for docker images

- The intention is to move the registry from Nexus3 to Docker Hub

Multi-Arch Support: Parallel CI/CD Pipeline

- The main reason for doing the migration from Nexus3 to Docker Hub is multi-arch support



(*) Uses docker manifest command
To be implemented on the arm branch of the pipeline.

LF Infrastructure Upgrade Challenges

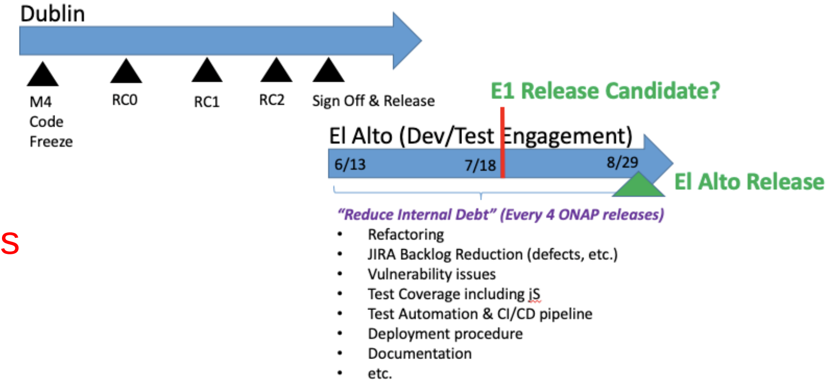
LF Infrastructure Team Priority:

1. CIMAN-234: Nexus2 - Migrate to Global-jjb gerrit-maven-stage job (opened 1/29/19)
2. CIMAN-229: Multi-cpu docker build jobs (opened 1/4/19)
3. CIMAN-239: Nexus3 - Migrate teams to start deploying images in DockerHub (opened 2/13/19)

The goal for the Dublin timeframe was to use Nexus 3 with the necessary changes requested by the ARM team. This effort was abandoned after efforts from both teams to make it work concluded it's not technically possible.

ONAP El Alto proposed release includes a shorter technical debt release

Request is to include Docker Hub Migration in El Alto as part of the "Test Automation & CI/CD Pipeline and Deployment procedure"



"Reduce Internal Debt" (Every 4 ONAP releases)

- Refactoring
- JIRA Backlog Reduction (defects, etc.)
- Vulnerability issues
- Test Coverage including jS
- Test Automation & CI/CD pipeline
- Deployment procedure
- Documentation
- etc.

See https://wiki.onap.org/display/DW/TSC+2019-04-18?preview=/60889264/63996004/2019-04-18%20ONAP%20Cadence%20Release_V5.pptx

So what's the problem with Nexus docker registry?

- Nexus3 docker registry implementation does not allow the creation of manifest-list (multiarch images)
    - https://issues.sonatype.org/browse/NEXUS-18546

- Nexus3 docker caching registry does not allow to pull and cache a multiarch docker image (caches only the x86_64 image).

3. ONAP use-case

- **Possible solutions**

Docker Trusted Registry

This storage solution is available but currently out of the picture as a possible solution:

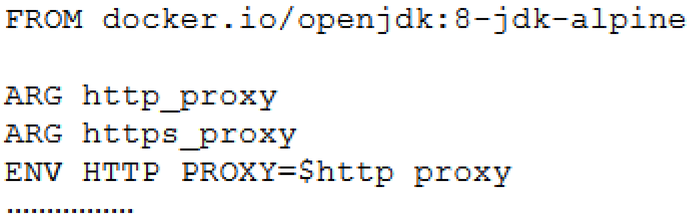
- Additional cost for ONAP which hasn’t been proposed yet
  - Will need additional planning and analysis to provide the exact numbers
- Extra cost for running operations to support it
- Availability of resources from the Infrastructure team will need to be acquired
- Requires about 6+ weeks for implementation if approved

ONAP Teams Tasks to Multi-arch CI/CD Pipeline

1. ONAP teams need to provide a component dependency chart to plan for an effective migration.
2. Tagging convention is still not consistent across components. Requires a fixing by tech teams.
3. Create additional jobs for building each container on new arch e.g. arm64 (under works to support it)
   a. The jobs would use the same jjb templates but be required to run on the arm64 docker build machines
   b. The result would be a docker image that would have the same microservices as the x86_64 but would run on arm64 infrastructure.
   c. Together with LF we can enable these jobs, but we will respect and involve the tech teams as much as they want to be involved.
4. Projects to migrate to global-jjb docker jobs.
5. Switching from Nexus3 docker registries to Docker Hub.
6. Rename all docker images to contain a suffix (a reference to arch of the image) to have different names for each architecture.
7. Create an additional job for building the manifest list (multi-arch image) out of the existent arch specific images e.g. aarch64 and amd64.

ONAP Teams: Component dependencies:

- Each ONAP Tech Team needs to provide docker components dependencies (e. g. Docker image deps):
  - SO (Service Orchestrator base image):



```
FROM docker.io/openjdk:8-jdk-alpine

ARG http_proxy
ARG https_proxy
ENV HTTP PROXY=$http proxy
................
```

  - SO base image depends on openjdk-8-jdk-alpine

Proposed migration plan and stakeholders

Activities happening or that can happen right now:

- Docker templates:
    - RELENG: Templates for Maven-Docker projects → COMPLETED
    - RELENG: Templates for Docker only projects → COMPLETED
    - ONAP: Define component dependency chart → PENDING (1 hr, maybe a PTL meeting task June 17?)

Order of activities that will happen after TSC approval and after Nexus2 migration (starting July 22):

1. RELENG + ONAP + MULTIARCH: Schedule a call to plan the migration → 1 hr
2. ONAP + MULTIARCH: Modify any docker registries mentions in the code and any changes to the code to make it multi-ach friendly. → (2 hr min average, Time depends on # of components per project)
3. ONAP + RELENG: Add global-jjb jobs in project yaml files. → 20 mins
4. ONAP + RELENG + MULTIARCH: Test jobs and address any issues → Best case scenario 2 hr
    a. Verify that the right artifacts were produced and pushed into DockerHub -> 1 hr
    b. Attempt to run multi-architecture pulls and make sure DockerHub calls the right manifest -> 1 hr
    c. Functional tests? Scope and ETA need to be defined by tech team.
5. ONAP: Confirm dependencies and needed images appear in Docker Hub. → 20 min
6. ONAP + RELENG: Remove deprecated Nexus3 jobs → 20 min

Notice these are best case scenario situations. If any ONAP component requires an upgrade on the global-jjb jobs, such upgrade will need to be evaluated and developed by RELENG.
At all times (until #6 is executed), Nexus3 jobs could be running in parallel as long as the Nexus3 registries are still used.

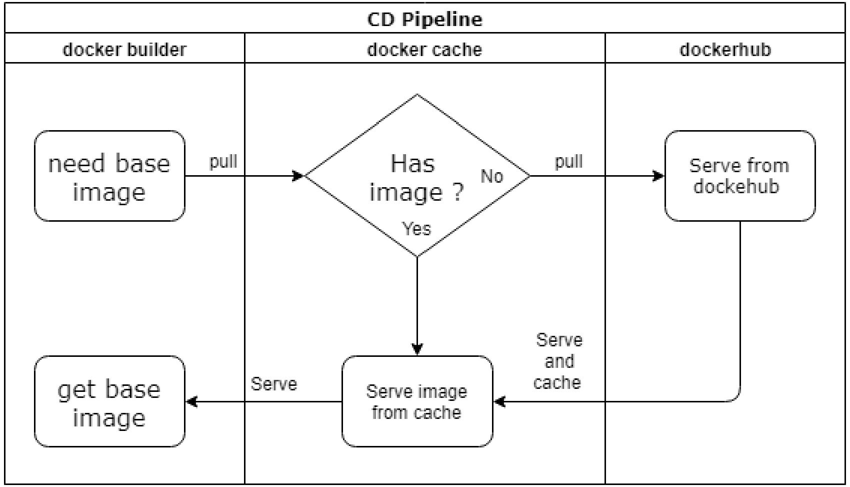
3. ONAP use-case

- **Questions and concerns**

Performance Concerns using Docker Hub

- Since Docker Hub registry is not in the proximity of the build infra, one could have concerns regarding bandwidth and performance for the docker build / push jobs.

- LF has given assurances that if such an issue arises they will provide docker images caching in the proximity of the build infrastructure.



Security Concerns using Docker Hub

- The recent security breach refers to a database of Docker Hub accounts whose account information have been stolen
  - https://motherboard.vice.com/en_us/article/7xgbzb/docker-hub-breach-hackers-stole-private-keys-tokens
- The impact of this incident is
  - Proprietary code of companies could have been hacked
    - Not applicable to ONAP as all code is open source
  - Images build with Docker autobuild system might contain injected code
    - Not applicable to ONAP as it uses its own build servers
- The conclusion is that this incident does not affect ONAP in any way, nor will possible future similar incidents

Docker Hub Availability

- The world’s leading service for finding and sharing container images with your team and the Docker community.

- Docker Hub is the world’s largest repository of container images with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers.*

- Incidents / downtime reported are related in the vast majority to their automated build system, not the registry itself.**

*https://www.docker.com/products/docker-hub
**https://cloudstatus.eu/status/docker

Nexus3 Availability/Performance Requirements

- There are no High Availability implementations for ONAP Nexus or Nexus3. They are individual services running on virtual machines in a shared cloud environment at Vexxhost.
  - nexus3.onap.org is running on a VM with 4x Intel Xeon 2.9GHz CPU, 15G of RAM, and 12G SSD disk.
  - nexus.onap.org is running on a VM with 4x Intel Xeon 2.5GHz CPU, 16G of RAM, and 10G SSD disk

- LF doesn’t have any Application Performance Monitoring (APM) in place for Nexus3 at this time, their monitoring is strictly at the system level, system load, number of processes, disk fill, network latency, open ports, total memory usage, etc.

Summary

ONAP Platform mandates support for any CPU architecture (e.g., Resource agnostic) to allow deployments on any hardware infrastructure

Nexus3 docker registry does not support multi-arch images

Request is to include Docker Hub Migration in El Alto as a Non-Functional Requirement

Questions

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Thank you