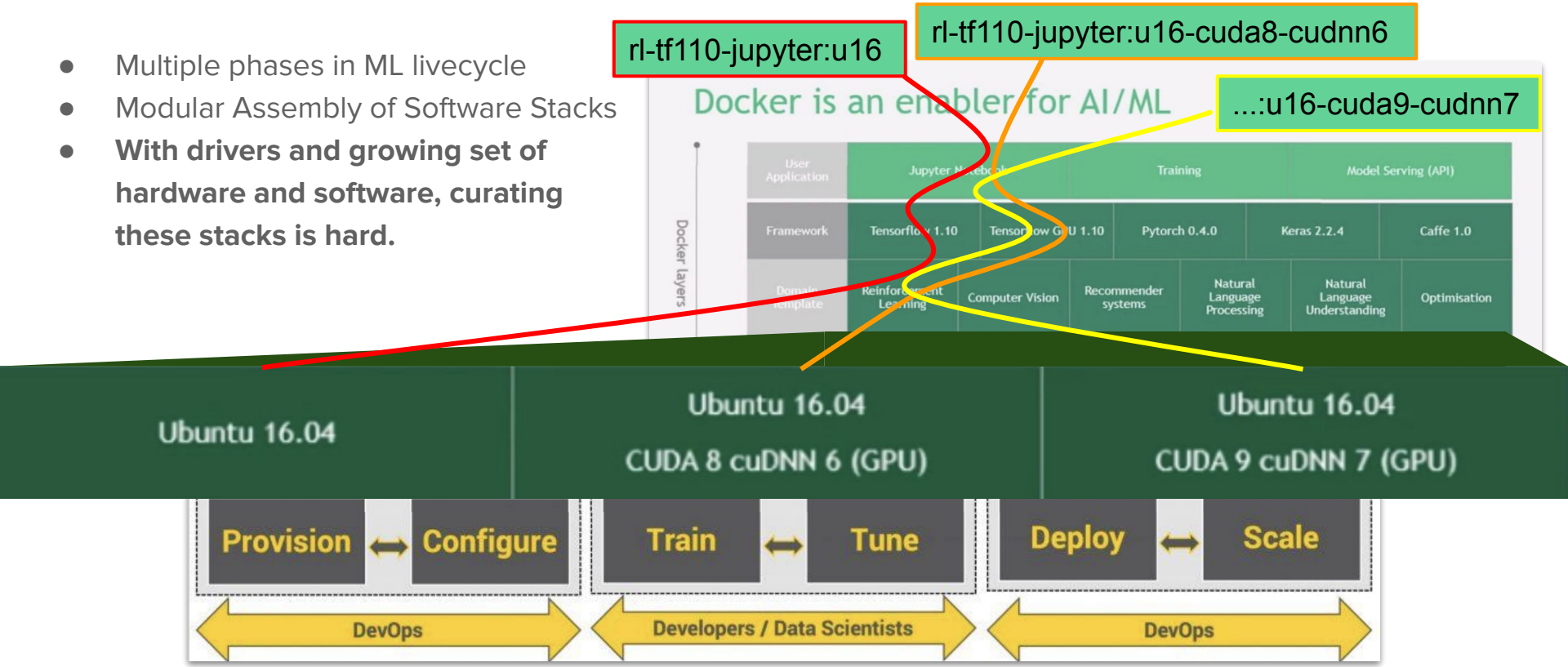
Hardware Optimized OCI Images via MetaHub Registry Proxy

High Performance Container Workshop - ISC19

Christian Kniep, QNIB Solutions
christian@qnib.org

Image Lifecycle needs Composability and Workflow

- Multiple phases in ML livecycle
- Modular Assembly of Software Stacks
- **With drivers and growing set of hardware and software, curating these stacks is hard.**



source: https://thenewstack.io/an-introduction-to-the-machine-learning-platform-as-a-service/

Current State of Affairs [cont]

Images optimized for different CPU μ-arch are picked by name.

As described in the previous slides.


```
$ docker run -ti --rm qnib/bench:cpu-broadwell
>> This container is optimized for: cpu:broadwell
```

```
$ docker run -ti --rm qnib/bench:cpu-skylake
>> This container is optimized for: cpu:skylake
```

```
$ docker run -ti --rm qnib/bench:generic
>> This container is not optimized for a specific microarchitecture
```

```
$ docker run -ti --rm qnib/bench:cpu_broadwell-nvcap_52
>> This container is optimized for: cpu:broadwell,nvcap:5.2
```

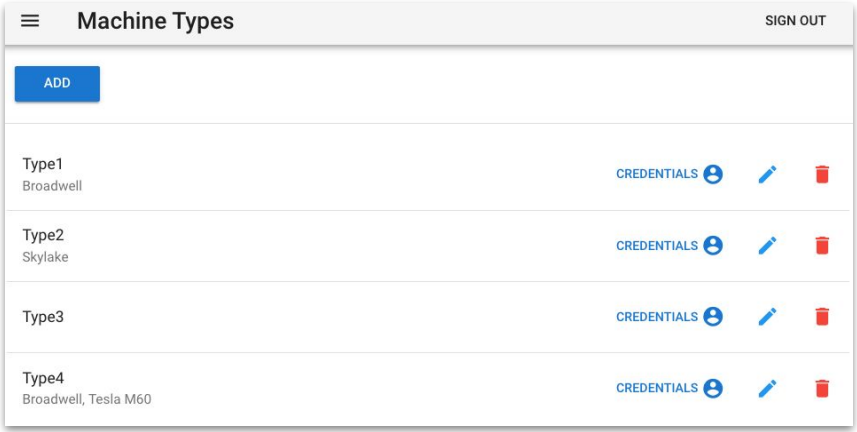
Dynamic Manifest List FTW!

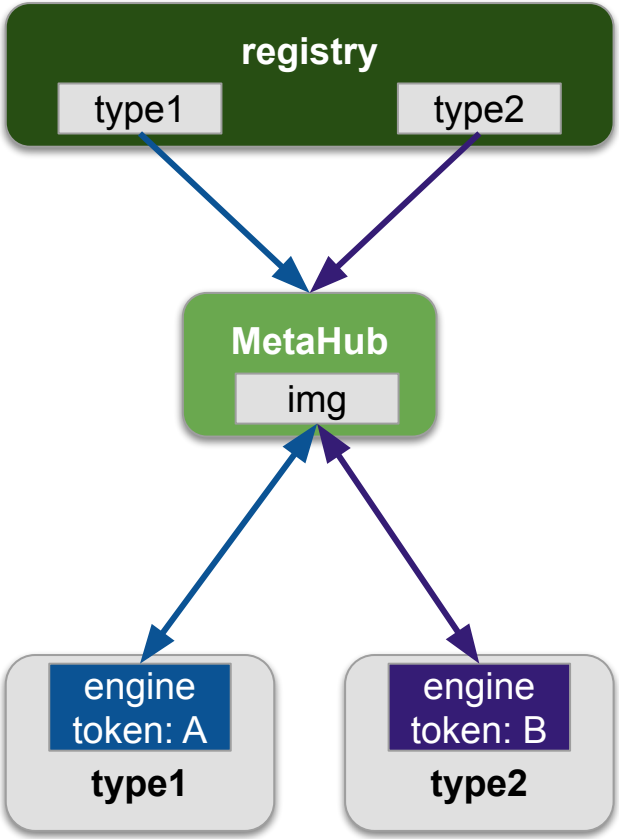
Manifest List provides choice via OCI “platform.features”.
Unfortunately this needs to be implemented by each runtime independently.


```
image: qnib/bench
manifests:
  - image: qnib/bench:cpu-broadwell
    platform:
      architecture: amd64
      features: [“cpu:broadwell”]
  - image: qnib/bench:cpu-skylake
    platform:
      architecture: amd64
      features: [“cpu:skylake”]
  - image: qnib/bench:cpu-coffelake
    platform:
      architecture: amd64
      features: [“cpu-coffelake”]
  - image:
qnib/bench:cpu_broadwell-nvcap_52
    platform:
      architecture: amd64
      features:
        - cpu-broadwell
        - nvcap-5.2
```

MetaHub to proxy normal Registry

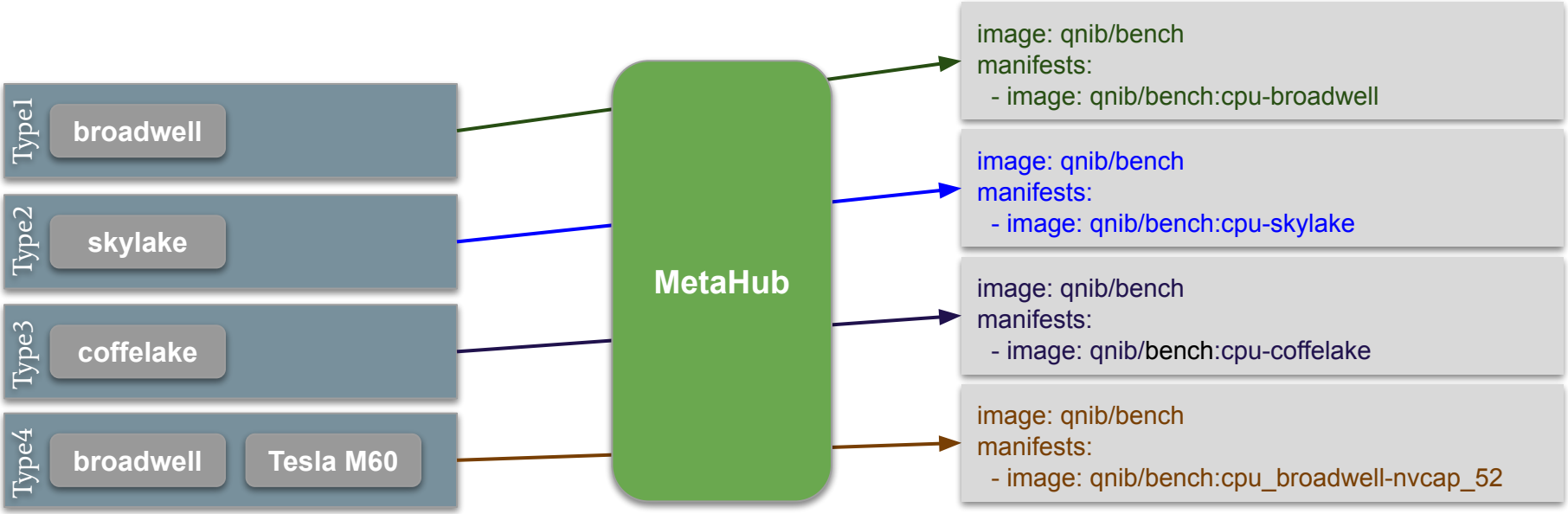
To be agnostic in terms of the runtime and supervision engine, MetaHub will dynamically create a Manifest List depending on the token used to authenticate.





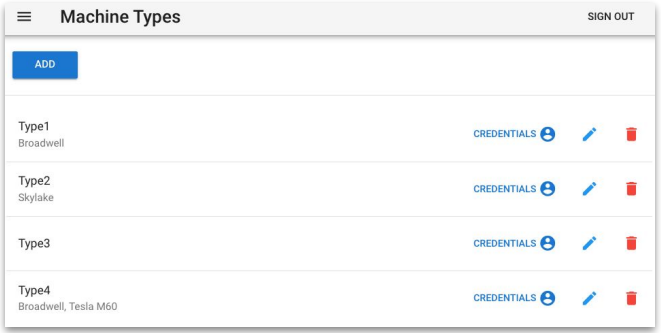
MetaHub to proxy normal Registry [cont #1]

Metahub reduces the Manifest List down to what the runtime ‘needs’.



MetaHub to proxy normal Registry [cont #2]

Each machine type has a special credentials to login to MetaHub.



Type1 broadwell

Type2 skylake

Type3 coffelake

Type4 broadwell Tesla M60

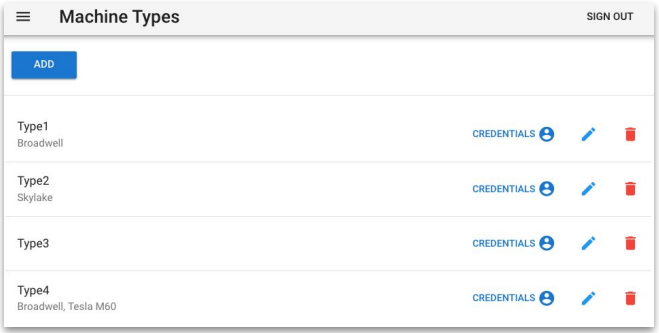
```
$ docker login -u qnib-type1 -p $PASSWD1 metahub.qnib.org
Login Succeeded
```

```
$ docker login -u qnib-type2 -p $PASSWD2 metahub.qnib.org
Login Succeeded
```

```
$ docker login -u qnib-type3 -p $PASSWD3 metahub.qnib.org
Login Succeeded
```

```
$ docker login -u qnib-type4 -p $PASSWD4 metahub.qnib.org
Login Succeeded
```

MetaHub to proxy normal Registry [cont #3]



Thus, the runtime will download the correct choice.


```
$ docker run -ti --rm metahub.qnib.org/qnib/bench
>> This container is optimized for: cpu:broadwell
```

```
$ docker run -ti --rm metahub.qnib.org/qnib/bench
>> This container is optimized for: cpu:skylake
```

```
$ docker run -ti --rm metahub.qnib.org/qnib/bench
>> This container is not optimized for a specific microarchitecture
```

```
$ docker run -ti --rm metahub.qnib.org/qnib/bench
>> This container is optimized for: cpu:broadwell,nvcap:5.2
```

FOOS FTW! MetaHub released.

https://qnib.org/2019/06/12/metahub/

Metahub: Dynamic Registry Proxy

Jun 12, 2019 • Christian Kniep

I won't say "Long time, no post" - but...

As I had some time at my hands the last couple of month, I was iterating on my idea on hardware optimization using Manifest List from the last post Match Node-Specific Needs Using Manifest Lists.

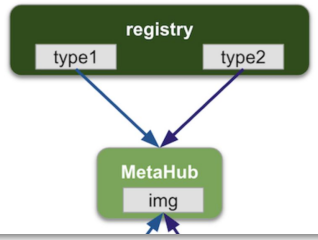
https://github.com/qnib/metahub

MetaHub - Dynamic OCI Registry Proxy

Announcement: qnib.org

The MetaHub project is meta-data registry, which serves images filtered via login so that a machine gets the image that fits the specifics of the host the image is going to run on.

That could be picking an image that not only fits the CPU Architecture (x86-64, ppcle, arm) but is optimized for the microarchitecture of the host (Broadwell, Skylake, ...). And it does not stop there - any host specific attribute can be use: Accelerators, network, configuration or the full depth of gcc options.



Thank you!