



KubeCon



CloudNativeCon

Europe 2019

Do you really need on premises serverless ?

Igor Khapov
IBM

Who am I ?



KubeCon

CloudNativeCon

Europe 2019

- **Igor Khapov**

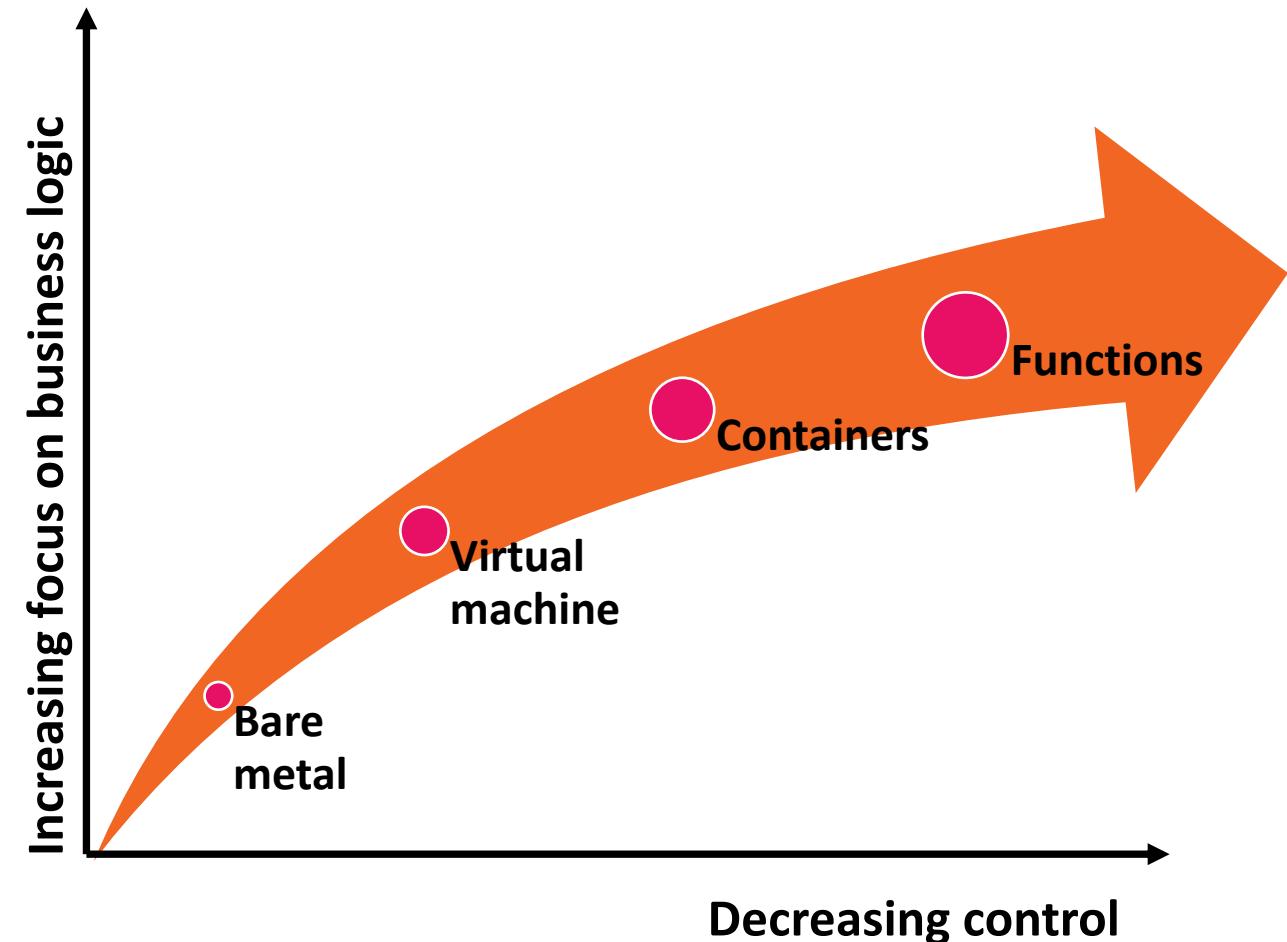
- #ibm #moscow_dev_lab
#developer #manager
#kubernetes #serverless
- #x86-64_ppc64le
#data_science_platform



What is serverless ?

Serverless architectures

are application designs that incorporate third-party “Backend as a Service” services, and include custom code run in managed, ephemeral containers on a “Functions as a Service” platform.*



History



KubeCon

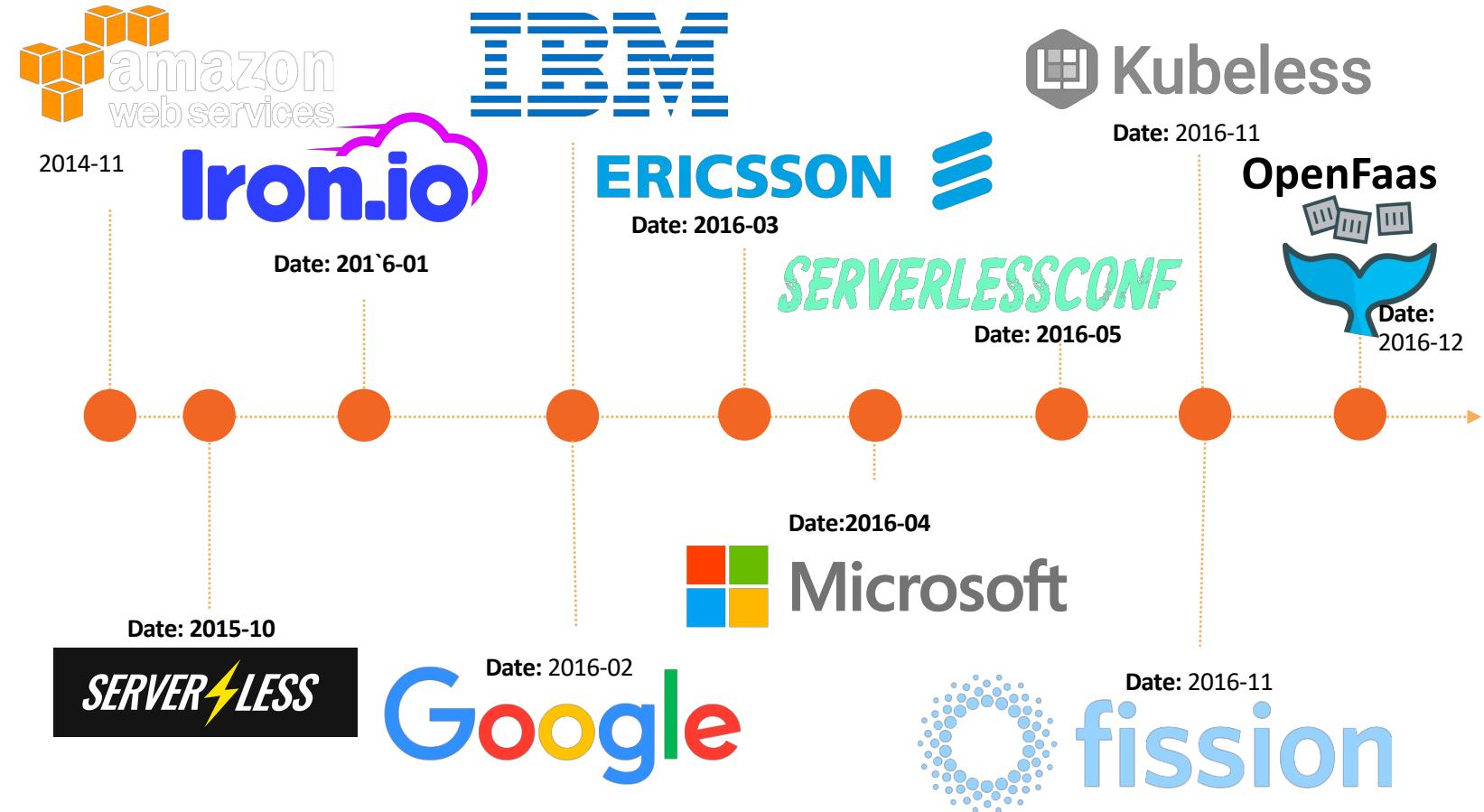
CloudNativeCon

Europe 2019

Launch Timeline



#open_source #serverless
#platforms #trend #history



Main use cases

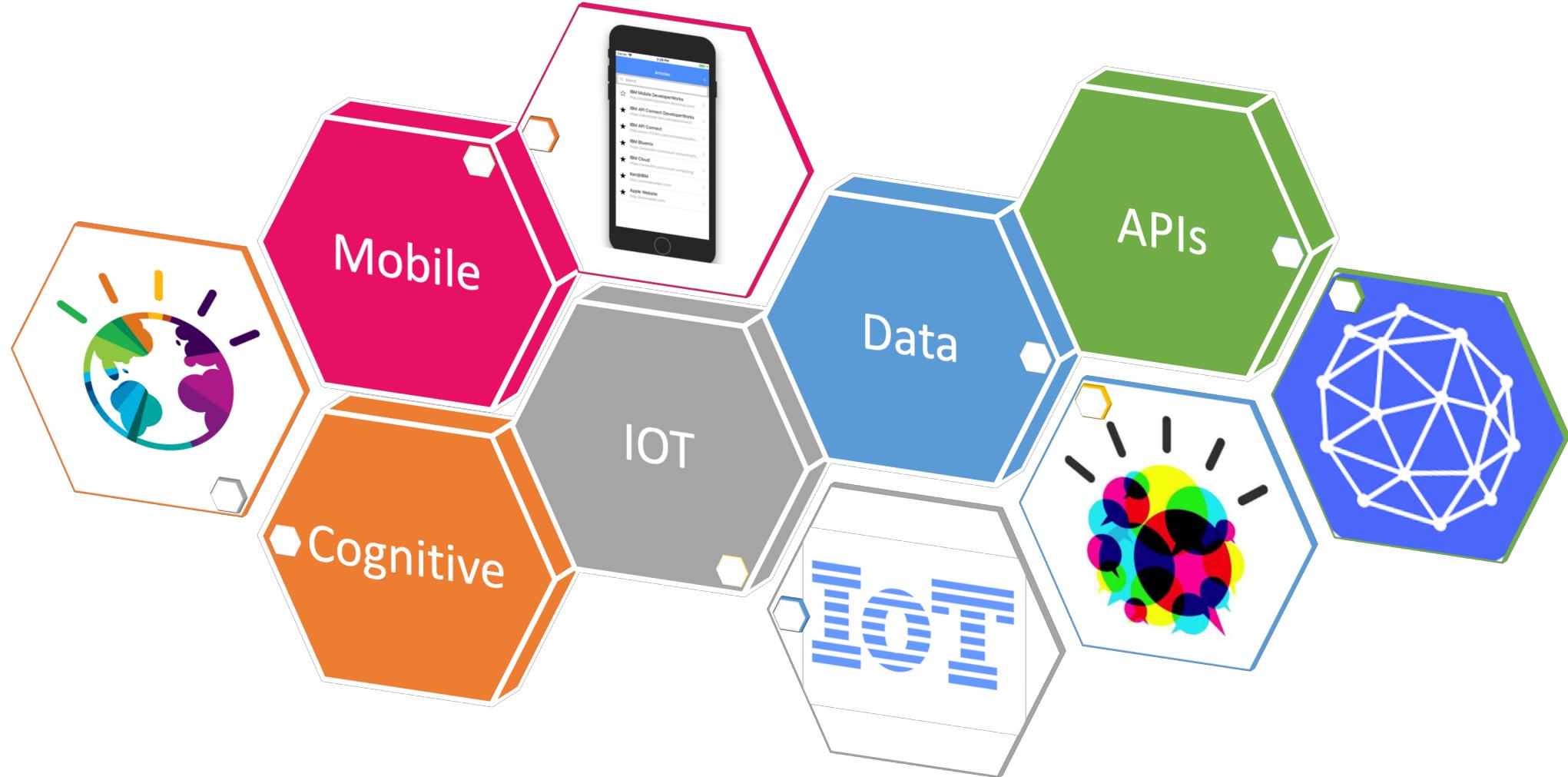


KubeCon



CloudNativeCon

Europe 2019



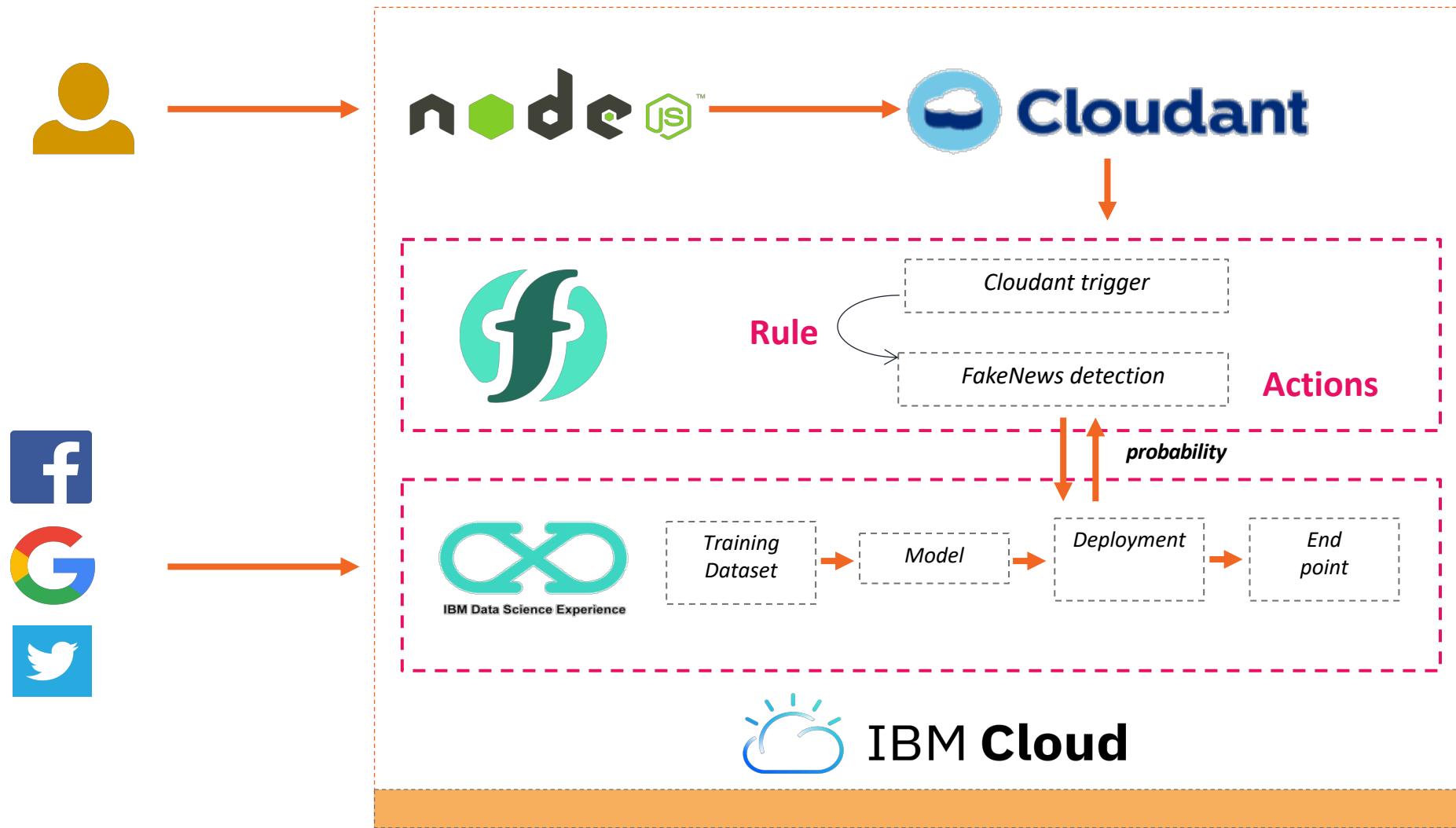
My first use case



KubeCon

CloudNativeCon

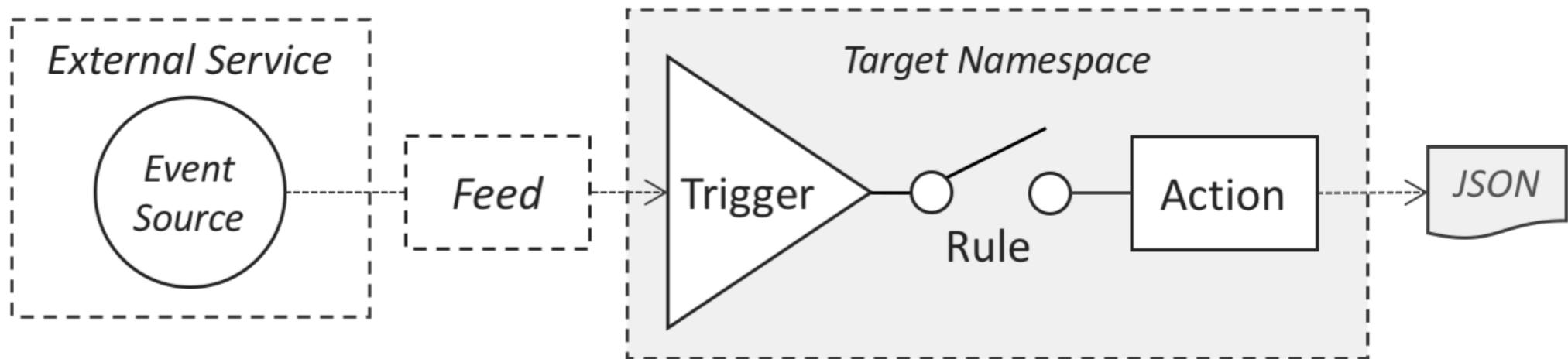
Europe 2019



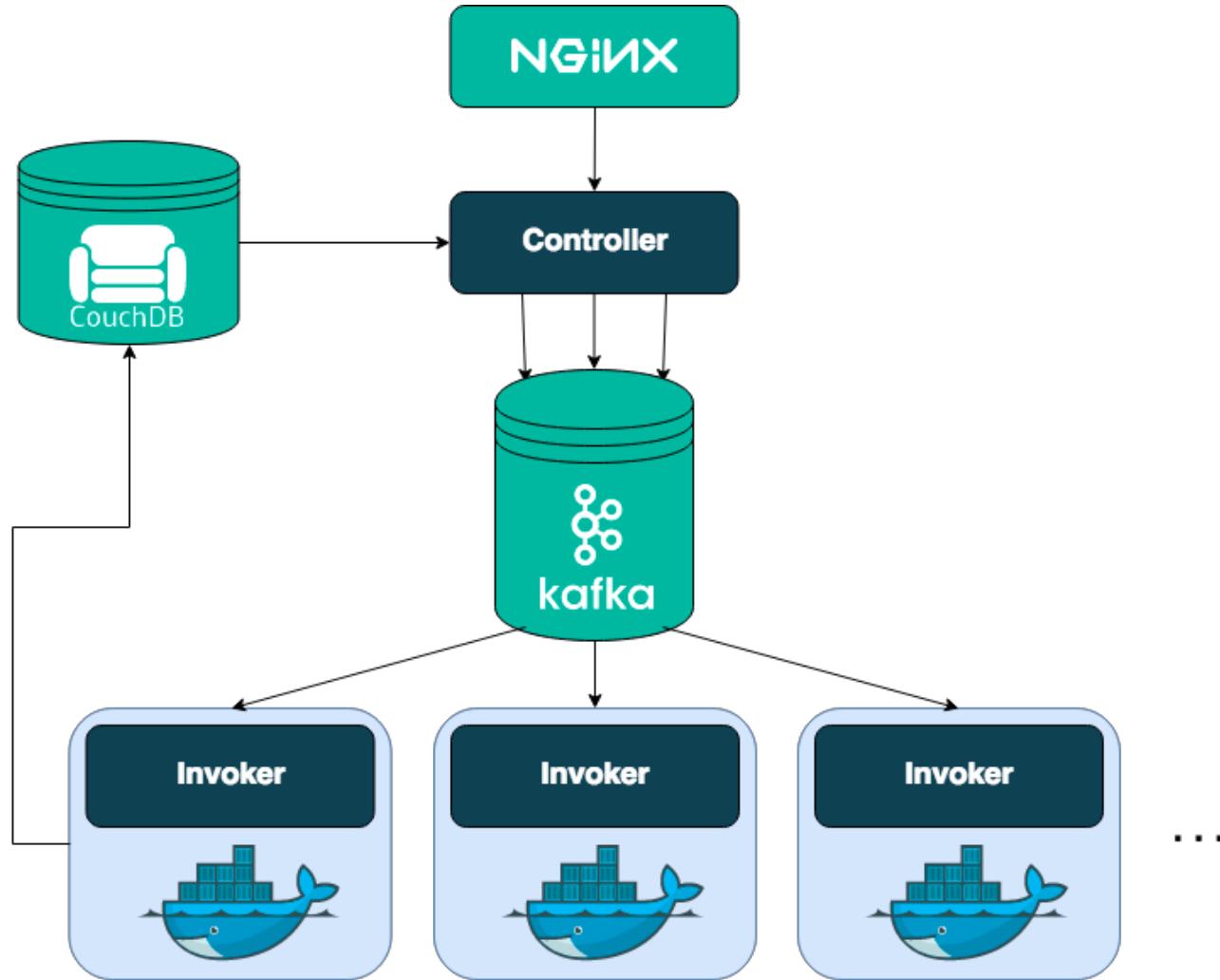
OpenWhisk flow



Europe 2019



OpenWhisk architecture



OpenWhisk git projects



KubeCon

CloudNativeCon

Europe 2019

Platform

Primary source code repositories including platform code, run books, tests and more.

[openwhisk](#)
[openwhisk-cli](#)
[openwhisk-apigateway](#)
[openwhisk-catalog](#)

Runtimes

OpenWhisk supports several languages via Docker runtime containers.

[openwhisk-runtime-nodejs](#)
[openwhisk-runtime-docker](#)
[openwhisk-runtime-python](#)
[openwhisk-runtime-go](#)
[openwhisk-runtime-swift](#)
[openwhisk-runtime-php](#)
[openwhisk-runtime-java](#)
[openwhisk-runtime-ruby](#)

Deployments

OpenWhisk can be deployed and configured on variety of platforms.

[openwhisk-deploy-kube](#)
[openwhisk-devtools/docker-compose](#)
[openwhisk-deploy-mesos](#)
[openwhisk-deploy.openshift](#)
[openwhisk/ansible](#)
[openwhisk/vagrant-setup](#)

Tooling

OpenWhisk provides variety of tools around deployment and development.

[openwhisk-wskdeploy](#)
[openwhisk-devtools](#)
[openwhisk-debugger](#)
[openwhisk-playground](#)
[openwhisk-vscode](#)
[openwhisk-xcode](#)

Packages

Several common service integrations are made available as packages. By default they are registered in the OpenWhisk catalog, under the `/whisk.system/` namespace, and include:

[openwhisk-package-alarms](#)
[openwhisk-package-cloudant](#)
[openwhisk-package-kafka](#)
[openwhisk-package-deploy](#)
[openwhisk-package-pushnotifications](#)
[openwhisk-package-rss](#)
[openwhisk-package-jira](#)
[openwhisk-package-template](#)

Clients and SDK

Here are the clients to access to OpenWhisk API:

[openwhisk-client-go](#)
[openwhisk-client-js](#)
[openwhisk-client-swift](#)
[openwhisk-client-python](#)

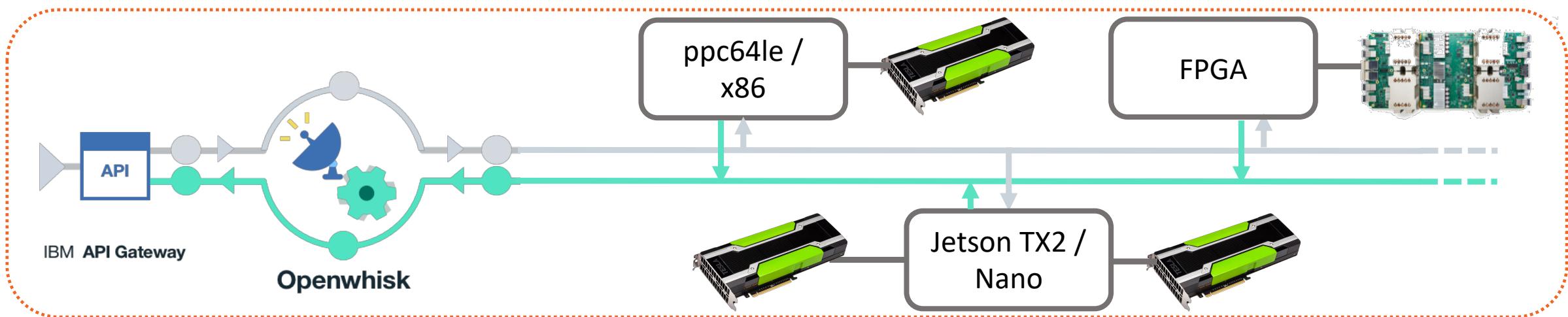
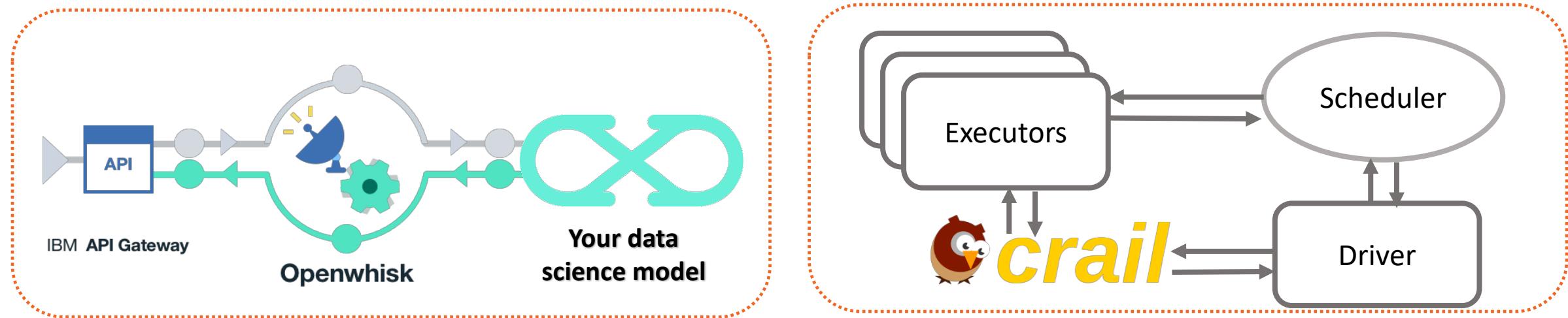
Serverless and data science



KubeCon

CloudNativeCon

Europe 2019



Function and data science



Europe 2019



Jupyter nb flow process



KubeCon

CloudNativeCon

Europe 2019

localhost:8888/notebooks/Untitled3.ipynb?kernel_name=python

jupyter Untitled3 Last Checkpoint: a minute ago (unsaved changes)

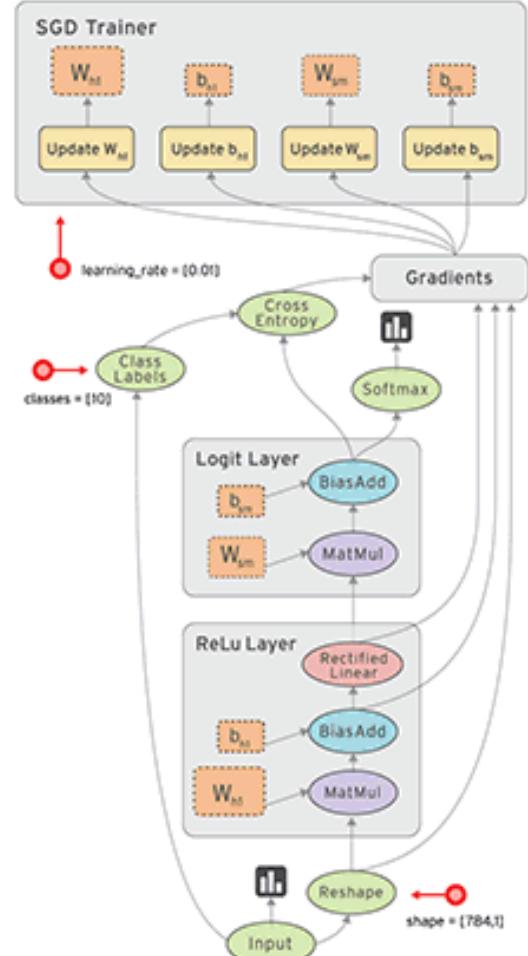
File Edit View Insert Cell Kernel Widgets Help

In [5]: `variable = 'jupyter not for developers'`

In [7]: `print(variable)`

code mesh

In [6]: `variable = 'code mesh'`



TF implementation

Save and restore a model



KubeCon

CloudNativeCon

Europe 2019

The image shows two side-by-side Jupyter notebook interfaces. The left notebook, titled 'save_and_restore_models.ipynb', demonstrates saving a Keras model to an HDF5 file. It includes code to create a model, fit it on training data, and save it. The right notebook, titled 'restore model', shows how to load the saved model and evaluate it on test data. Both notebooks run on Python 3.

Left Notebook (In [16]):

```
model = create_model()
model.fit(train_images, train_labels, epochs=5)

# Save entire model to a HDF5 file
model.save('my_model.h5')

Epoch 1/5
1000/1000 [=====] - 1s 862us/step - loss: 1.2003 - acc: 0.6520
Epoch 2/5
1000/1000 [=====] - 0s 480us/step - loss: 0.4355 - acc: 0.8770
Epoch 3/5
1000/1000 [=====] - 0s 414us/step - loss: 0.2833 - acc: 0.9310
Epoch 4/5
1000/1000 [=====] - 0s 423us/step - loss: 0.2186 - acc: 0.9410
Epoch 5/5
1000/1000 [=====] - 1s 602us/step - loss: 0.1548 - acc: 0.9680
```

Right Notebook (In [1]):

```
from __future__ import absolute_import, division, print_function
import os
import tensorflow as tf
from tensorflow import keras
tf.__version__

In [2]: new_model = keras.models.load_model('my_model.h5')
new_model.summary()
```

Right Notebook (In [6]):

```
(train_images, train_labels), (test_images, test_labels) = tf.keras.datasets.mnist.load_data()

train_labels = train_labels[:1000]
test_labels = test_labels[:1000]

train_images = train_images[:1000].reshape(-1, 28 * 28) / 255.0
test_images = test_images[:1000].reshape(-1, 28 * 28) / 255.0
```

Right Notebook (In [7]):

```
loss, acc = new_model.evaluate(test_images, test_labels)
print("Restored model, accuracy: {:.2f}%".format(100*acc))

1000/1000 [=====] - 0s 159us/step
Restored model, accuracy: 86.10%
```

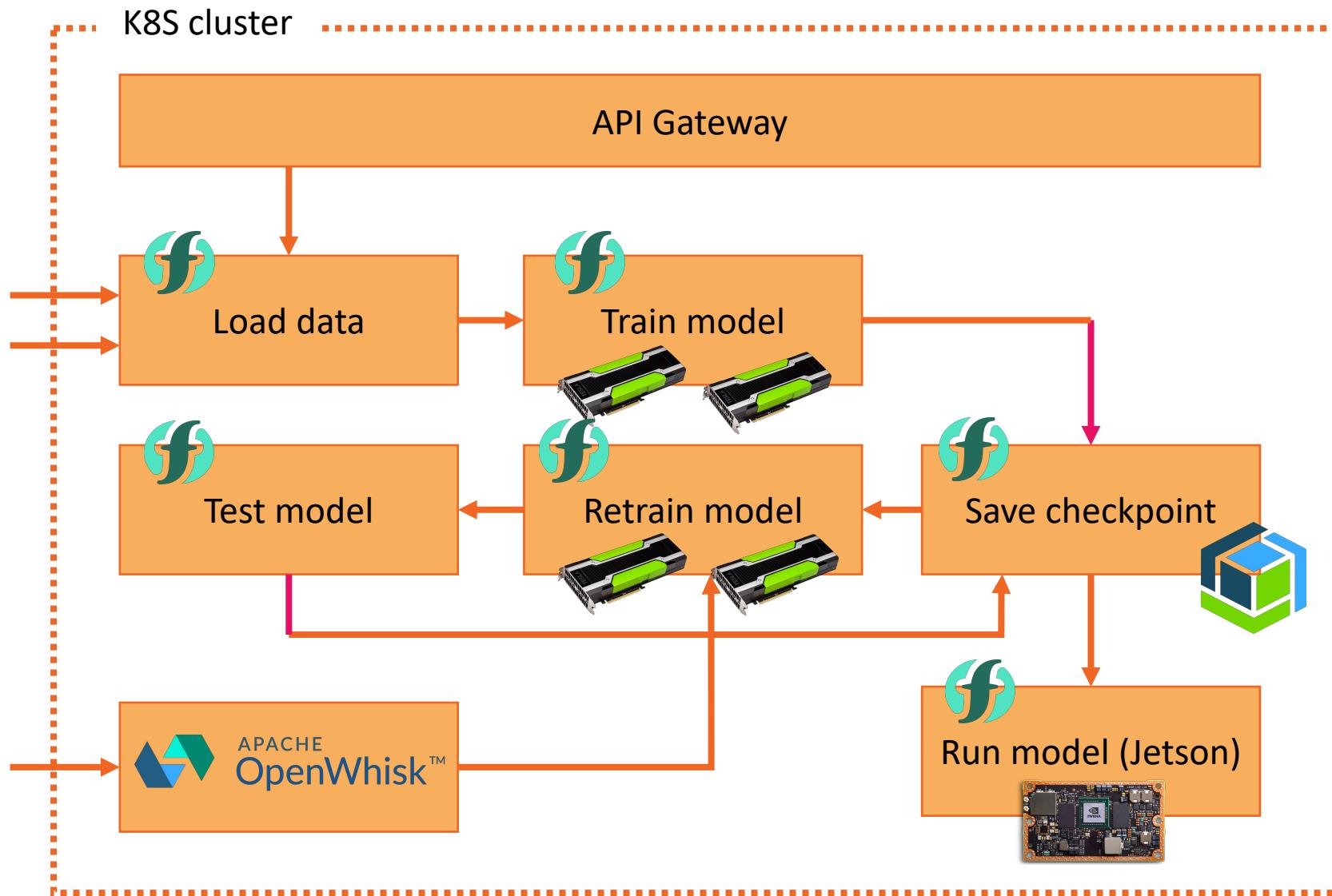
Target architecture



KubeCon

CloudNativeCon

Europe 2019



Docker for multiple architectures



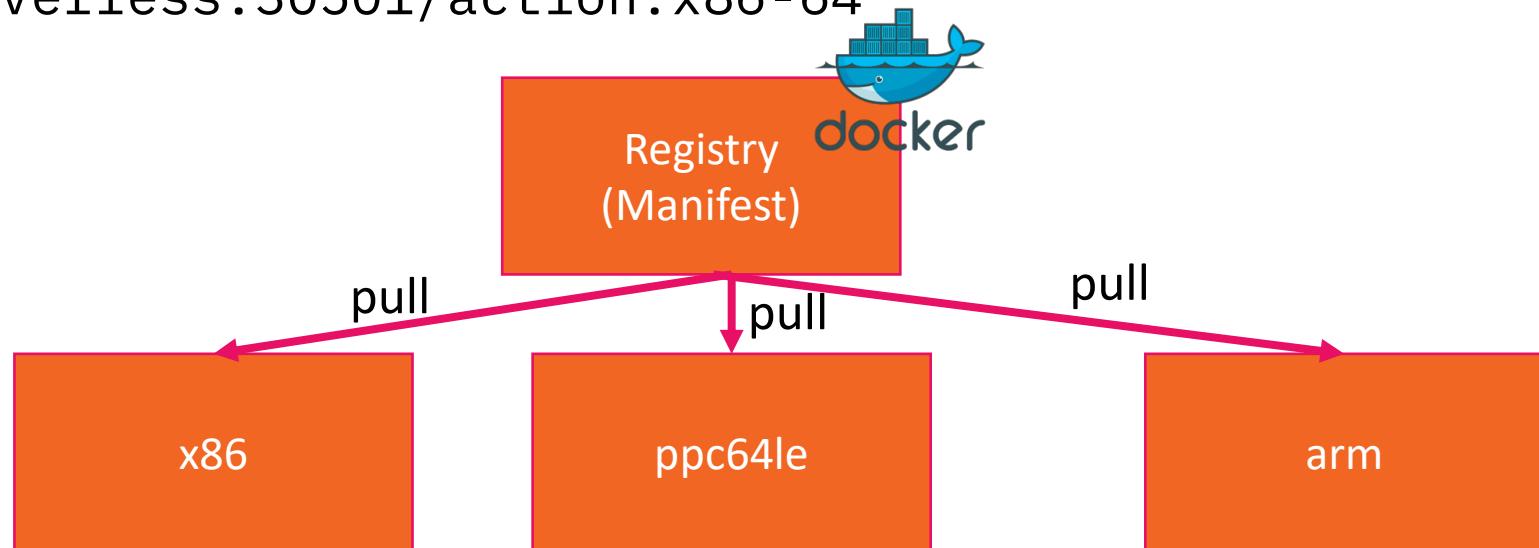
KubeCon



CloudNativeCon

Europe 2019

```
docker -D manifest create -insecure \
serverless:30501/action:latest \
serverless:30501/action:ppc64le \
serverless:30501/action:x86-64
```



```
root@serverless:~# docker images |grep ac1|grep -v 18 |grep -v none
serverless:30501/ac1                               ppc64le          1a9dd94f6deb      2 weeks ago     200MB
serverless:30501/ac1                               latest           cb82052802de      5 weeks ago     172MB
serverless:30501/ac1                               x86-64           cb82052802de      5 weeks ago     172MB
```

Scheduler customisation



Europe 2019

KubernetesClient.scala

```
127      .withRestartPolicy("Always")
128      if (config.userPodNodeAffinity.enabled) {
129          val invokerNodeAffinity = new AffinityBuilder()
130          .withNewNodeAffinity()
131          .withNewRequiredDuringSchedulingIgnoredDuringExecution()
132          .addNewNodeSelectorTerm()
133          .addNewMatchExpression()
134          .withKey(config.userPodNodeAffinity.key) // Line 134
135          .withOperator("In")
136          .withValues(config.userPodNodeAffinity.value)
137          .endMatchExpression()
138          .endNodeSelectorTerm()
139          .endRequiredDuringSchedulingIgnoredDuringExecution()
140          .endNodeAffinity()
141          .build()
142      podBuilder.withAffinity(invokerNodeAffinity)
143 }
```

KubernetesContainerFactory.scala

KubernetesContainer.scala

KubernetesContainerFactory.scala

InvokerReactive.scala

KubernetesContainerFactoryProvider

Demo



KubeCon



CloudNativeCon

Europe 2019

Is all actions should be hardware agnostic ?



KubeCon
Europe 2019



CloudNativeCon
Europe 2019

- **Collocation to the data warehouse**
- **Selectors for GPU / TPU resources**
- **Selectors for resources (RAM, cores ...)**

You need on premise serverless if ...



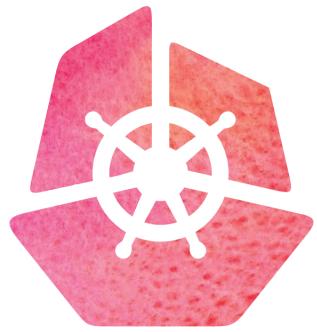
KubeCon



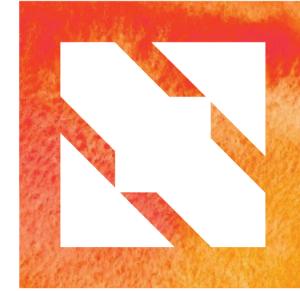
CloudNativeCon

Europe 2019

- A lot of in-company developers
- Functions which is in NOT hardware agnostic
- Increase utilization of your resources
- Split workflows into small steps and store temporary results
- You have some time to implement or adopt that



KubeCon



CloudNativeCon

Europe 2019