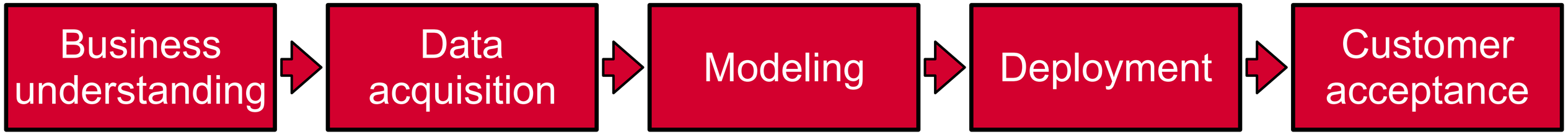
O'REILLY®
oscon

Serverless deep learning

Rustem Feyzkhanov
18 July 2018

oscon.com
#oscon

Data science process



- Define objectives
- Identify data sources

- Ingest data
- Explore data
- Update data

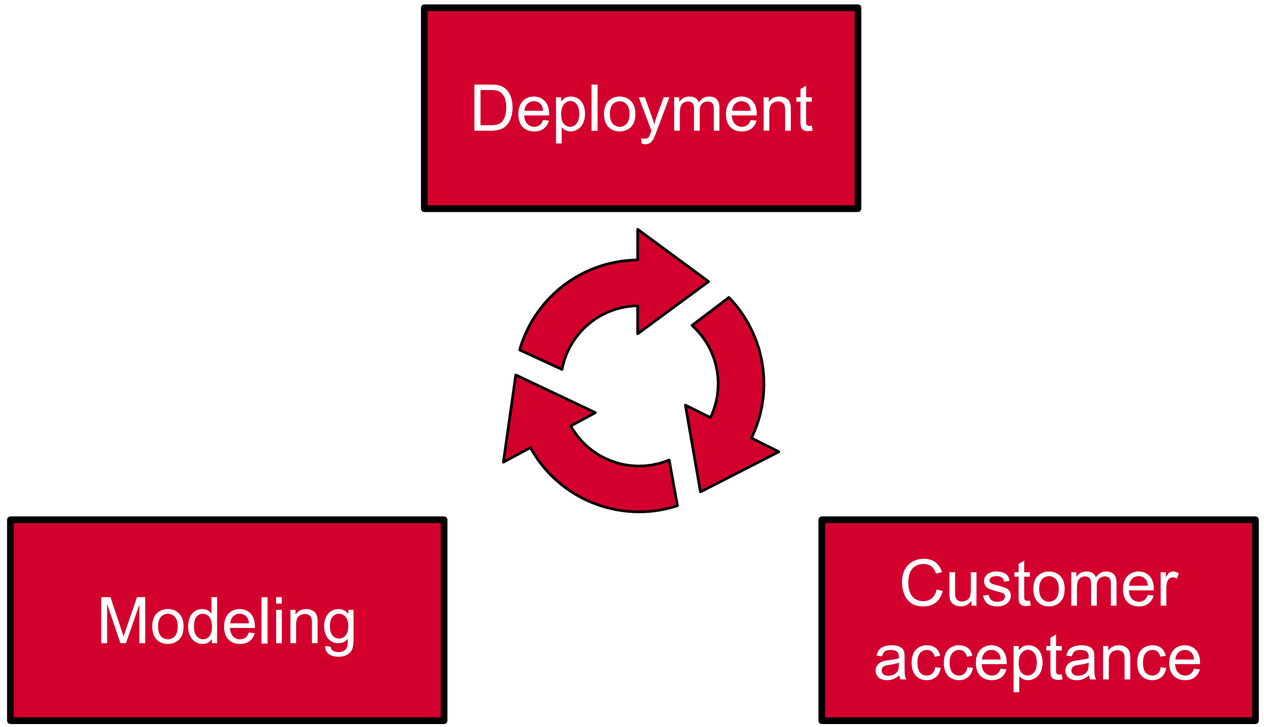
- Feature selection
- Create model
- Train model

- Operationalize

- Testing and validation
- Handoff
- Re-train and re-score

from https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview

Data science process



Challenges:

- starting fast
- being flexible
- integrating in current infrastructure

from https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview

Takeaways from this talk

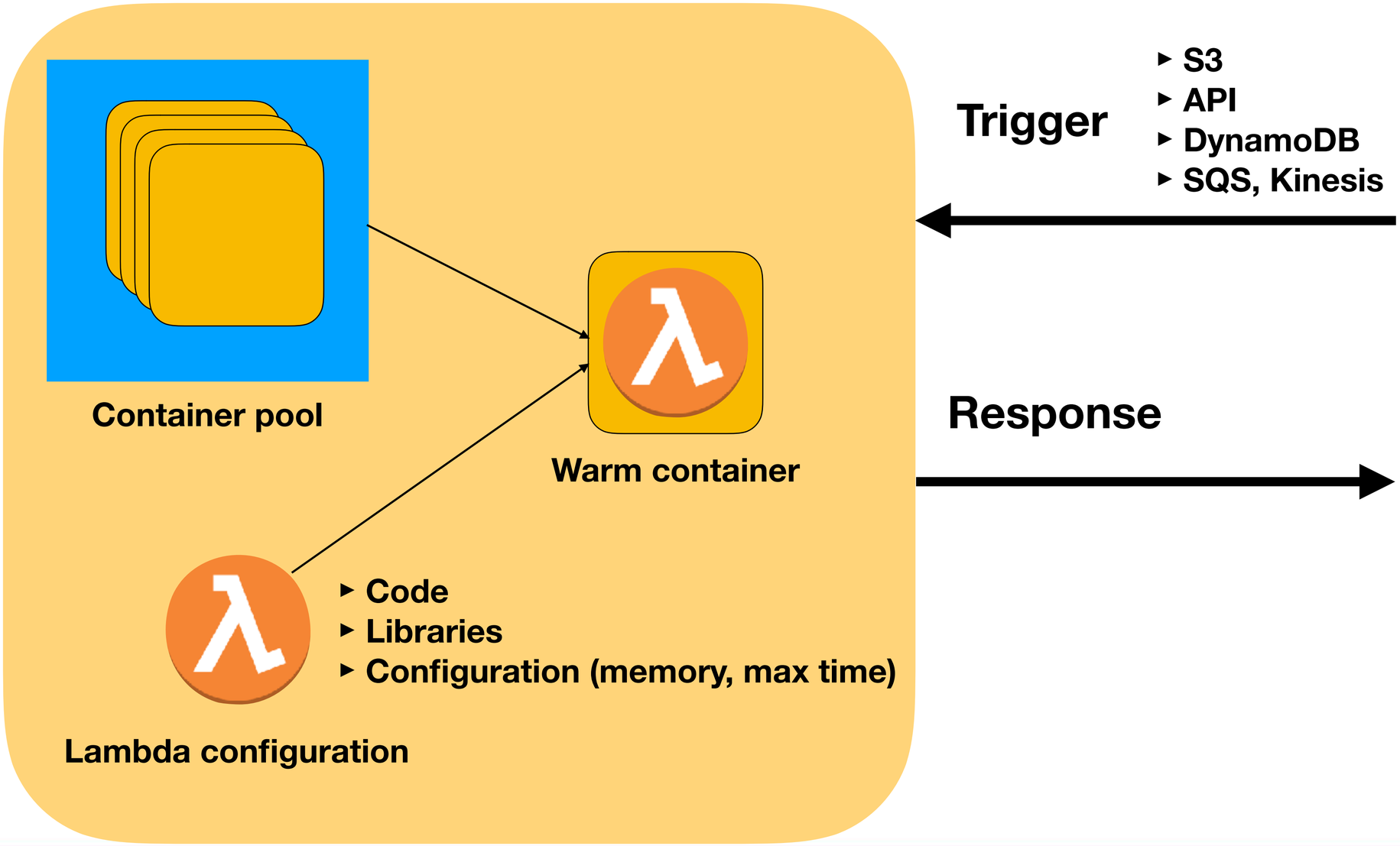
- How serverless deep learning works
- Serverless deep learning architecture
- Serverless deep learning use cases, do’s and don’ts
- Serverless deep learning simplicity of code

Function as a service (FaaS)

On premise	IaaS	PaaS	FaaS	SaaS
Functions	Functions	Functions	Functions	Functions
Application	Application	Application	Application	Application
Runtime	Runtime	Runtime	Runtime	Runtime
Operating system	Operating system	Operating system	Operating system	Operating system
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Networking	Networking	Networking	Networking	Networking
Storage	Storage	Storage	Storage	Storage
Hardware	Hardware	Hardware	Hardware	Hardware

Lambda function - AWS implementation of FaaS

How Lambda works

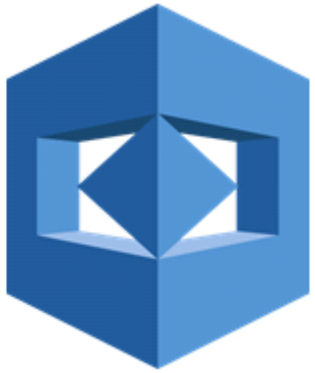


Lambda triggers

DynamoDB

S3

CloudWatch



Lex

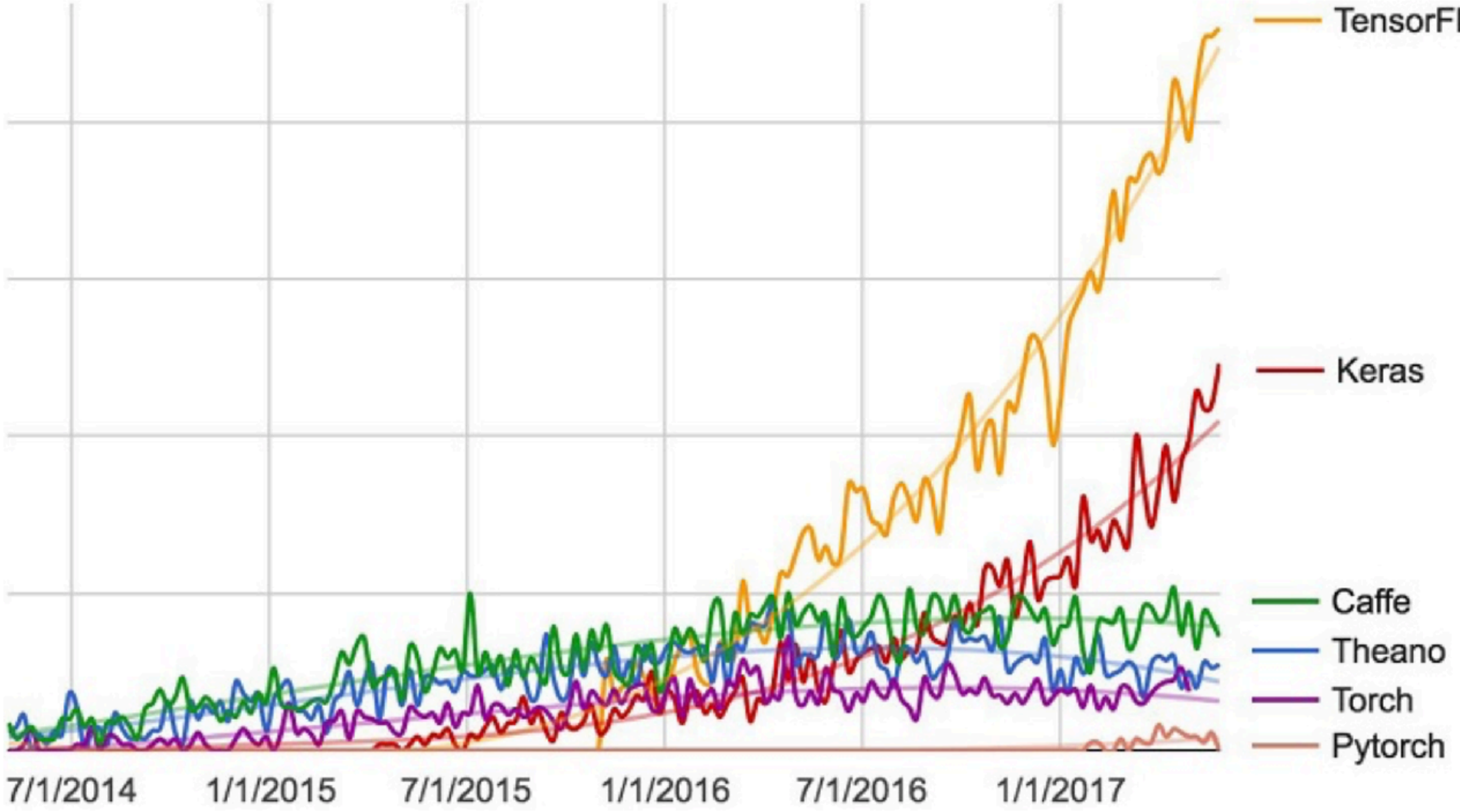
SQS

API gateway

Lambda pros/cons/limits

Pros	Cons	Limits
Easy to deploy (no docker) Easy to connect to triggers (API, S3, SQS, DynamoDB) Easy to scale Relatively cheap	Logging is not great No local debug Unpredictable warm containers	max 3 GB RAM max 500 MB disk max 5 min execution time CPU is proportional to provisioned memory

TensorFlow popularity



Google web search interest for different deep learning frameworks over time

Francois Chollet. “Deep Learning with Python MEAP.”

TensorFlow 1.*

- Keras in the core
- TF Boosted trees (!) + other ML algorithms
- Lots of other stuff:

  - https://github.com/tensorflow/tensorflow/blob/master/RELEASE.md

Why TF on Lambda?

- ☑ ~20000 runs for $1
- ☑ 1000 concurrent executions (up to 10000)
- ☑ Pay as you go model

=> perfect for early stage projects

Implementation Problem

Lambda limit - 50 MB

TensorFlow archive size - 43.1MB

Numpy archive size - 16.5 MB

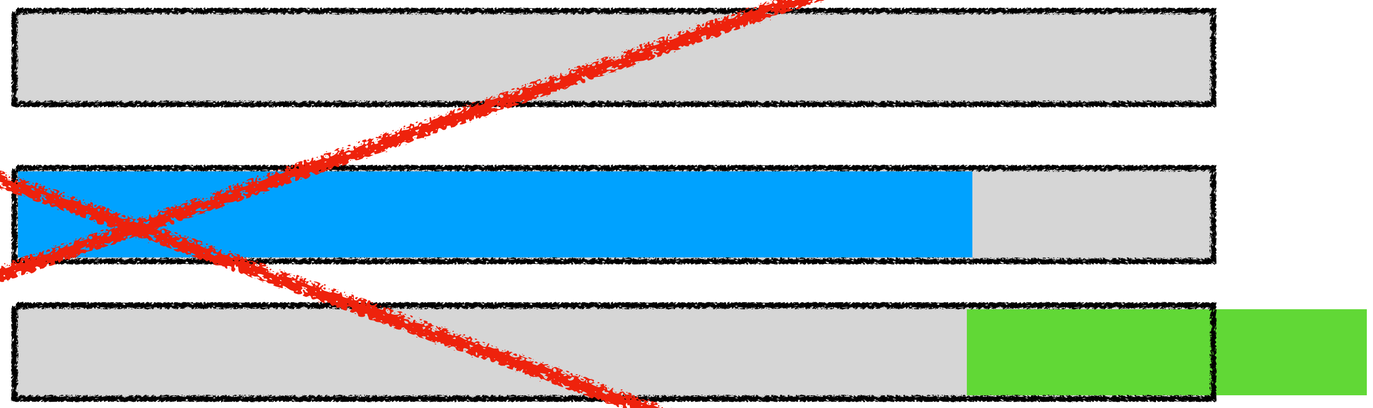
+ dependencies

Implementation Problem

Lambda limit - 50 MB

TensorFlow archive size - 43.1MB

Numpy archive size - 16.5 MB

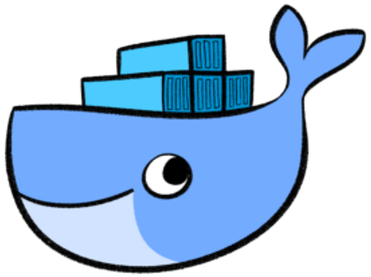


+ dependencies

250 MB unarchived

https://hackernoon.com/exploring-the-aws-lambda-deployment-limits-9a8384b0bec3

How to solve



Docker

+

Amazon Linux

+

PyPI wheels

+

Magic

Magic:

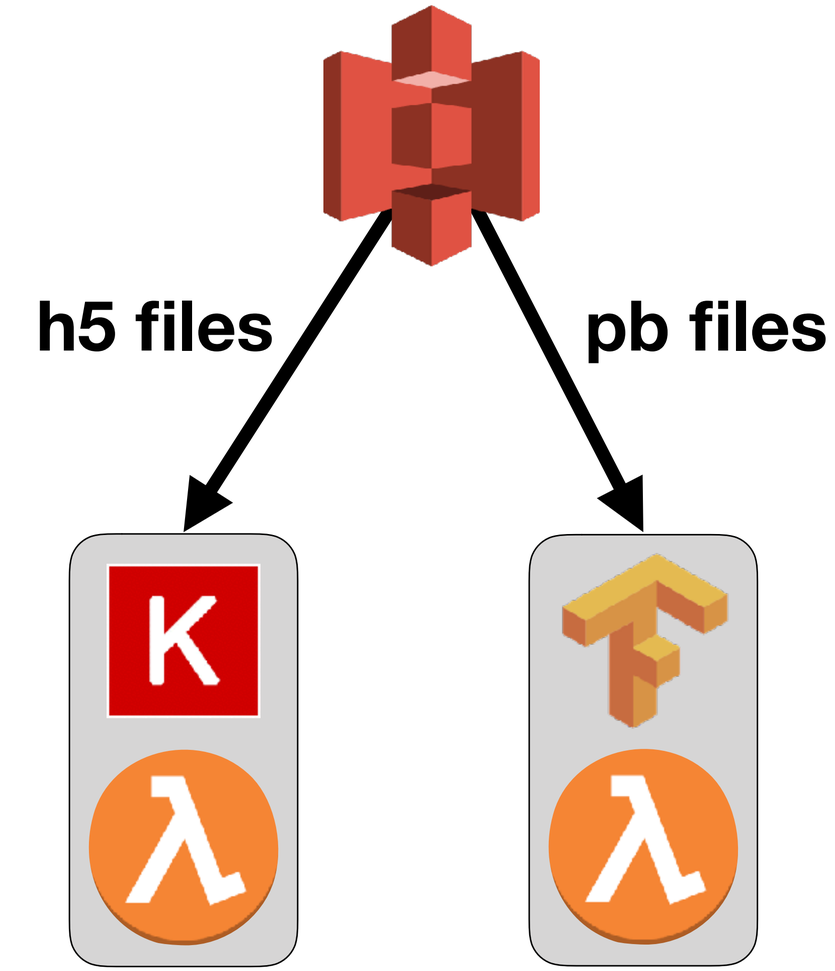
1. Compress so files
2. Delete .pyc files
3. Remove test folders, visualisation folders

Look up here: https://github.com/ryfeus/lambda-packs/blob/master/Tensorflow/buildPack.sh

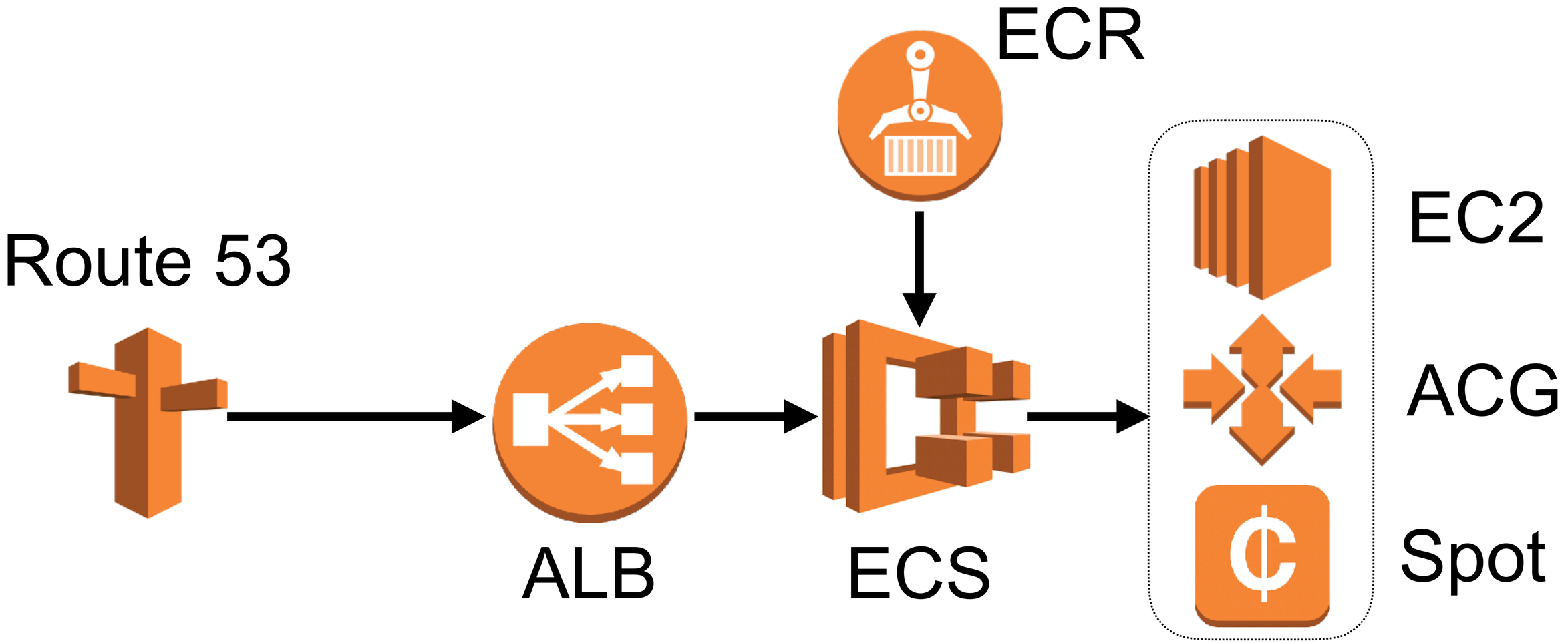
How to import models

Use Keras or TensorFlow for weights import:

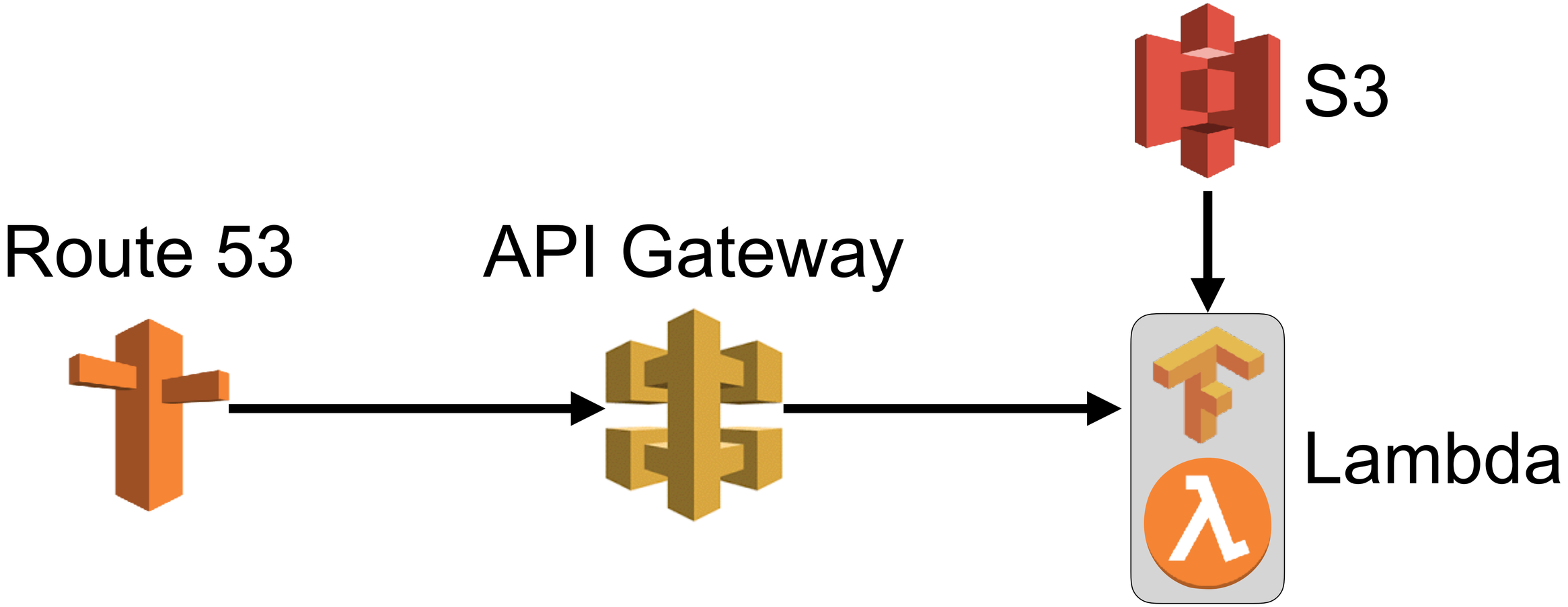
- Keras - h5 files
- TensorFlow - pb files



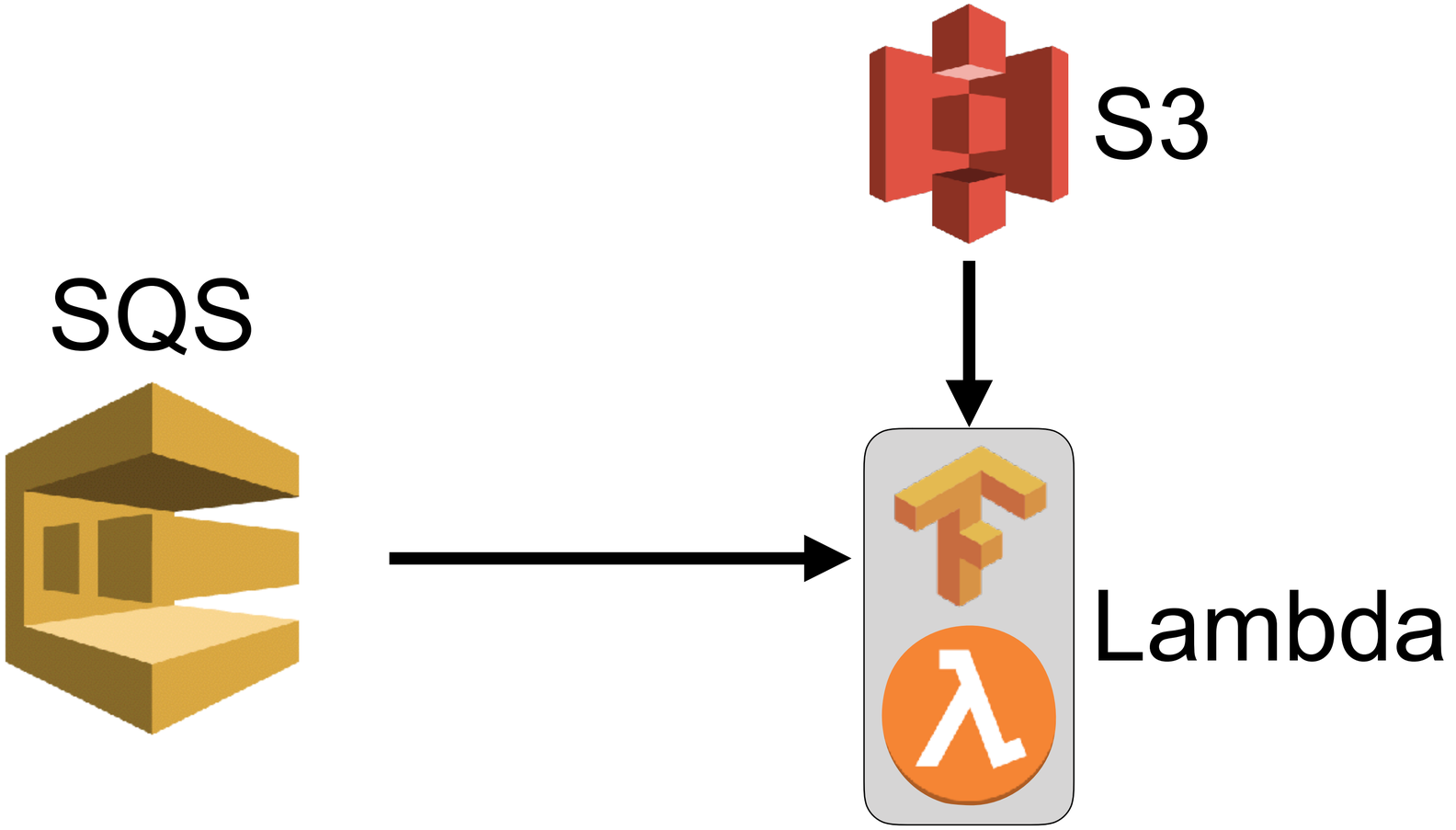
Usual AWS architecture for DL



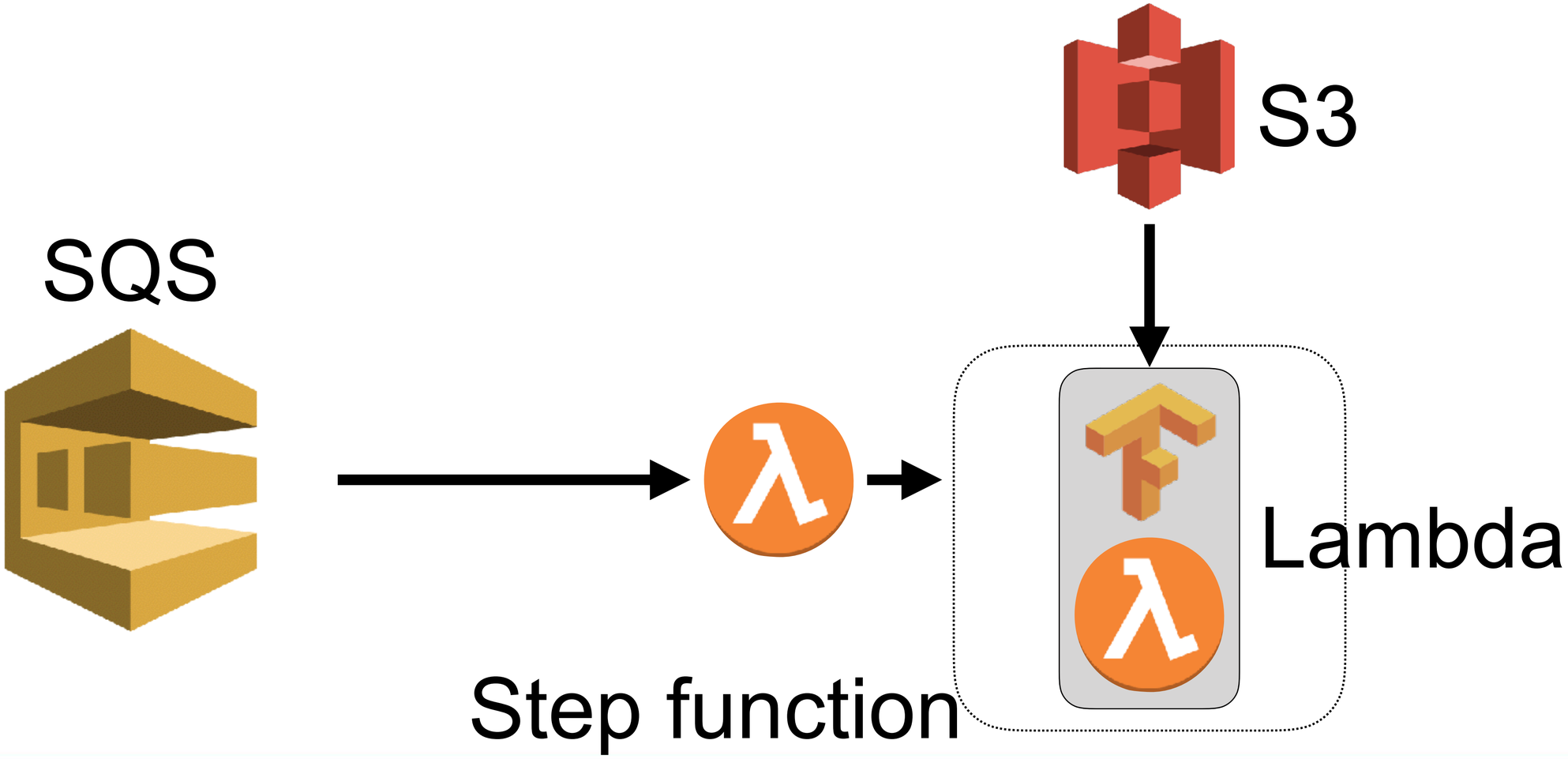
Architecture for DL using Lambdas



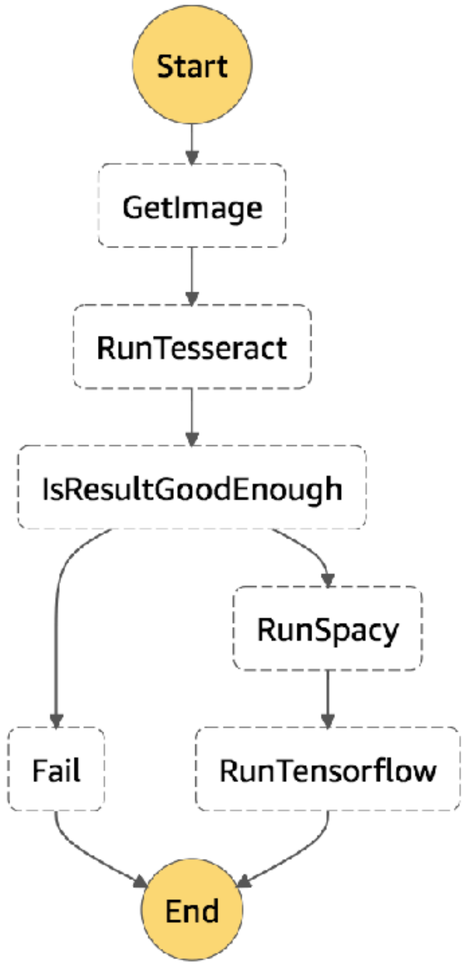
Architecture for DL using Lambdas



Architecture for DL using Lambdas



Architecture for DL using Lambdas



Step functions:

- allow modular approach
- enable to handle errors and special cases
- serverless functions => serverless pipelines

Where to get models

Train yourself

Keras:

https://github.com/fchollet/deep-learning-models

TensorFlow:

TensorFlow ZOO (https://github.com/tensorflow/models/tree/master/official)

TensorFlow.org (https://www.tensorflow.org/performance/performance_models)

Github projects (e.g. https://github.com/taehoonlee/tensornets)

Projects - Image recognition

API to recognize image using Inception-v3 - 0.00005$ / 1 image

giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.92276)

lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score = 0.00282)

earthstar (score = 0.00176)

racer, race car, racing car (score = 0.00062)

ndri, indris, Indri indri, Indri brevicaudatus (score = 0.00061)

https://github.com/ryfeus/lambda-packs/tree/master/Tensorflow

https://www.tensorflow.org/tutorials/image_recognition

Projects - accessible WEB

API to describe what happens on the picture - 0.0001$ / 1 image

Image

API Response


```
[
  - {
      url: "https://hack4impact.org/assets/images/photos/mayors-awards.jpg",
    - captions: [
        - {
            prob: "0.005999",
            sentence: "a group of people standing next to each other ."
          },
        - {
            prob: "0.002621",
            sentence: "a group of people posing for a picture ."
          },
        - {
            prob: "0.001902",
            sentence: "a group of people posing for a picture"
          }
      ]
  }
]
```


Abhinav Suri - https://medium.freecodecamp.org/making-the-web-more-accessible-with-ai-84598eebabdb

How do you know if this is for you

- You want to deploy your model for pet project
- You want to make s simple MVP for your startup/project
- You have simple model and this architecture will reduce cost
- You have peak loads and it is hard to manage clusters

How do you know if this is NOT for you

- You have very complex model (a lot of data as input/high CPU)
- You need to have real-time response

Some new stuff

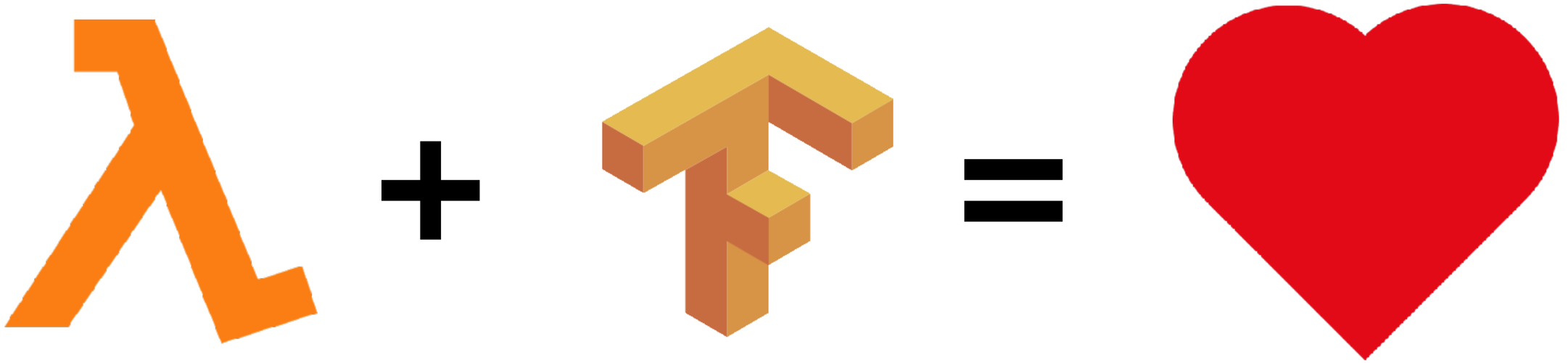
- LightGBM package - fast, distributed, high performance gradient boosting framework from Microsoft + Sklearn/Scipy/Numpy

https://github.com/ryfeus/lambda-packs/tree/master/LightGBM_sklearn_scipy_numpy

- Spacy package - natural language processing library

https://github.com/ryfeus/lambda-packs/tree/master/Spacy

Conclusions



Presentation: http://bit.ly/2L72P2y

Checkout here: https://github.com/ryfeus/lambda-packs (https://goo.gl/HQiHD7)