BERLIN, 4.MAR 2019

MICHAEL PERLIN

Run ML as Cloud Function

INNOQ

INNOQ

www.innoq.com

SERVICES

Strategy & technology consulting
Digital business models
Software architecture & development
Digital platforms & infrastructures
Knowledge transfer, coaching & trainings
Big data & machine learning

FACTS

~130 employees
Privately owned
Vendor-independent

OFFICES

Monheim
Berlin
Offenbach
Munich
Hamburg
Zurich

CLIENTS

Finance
Telecommunications
Logistics
E-Commerce
Fortune 500
SMBs
Startups

We are sponsoring today 🌯 🍺 🍕 🍩 🍹

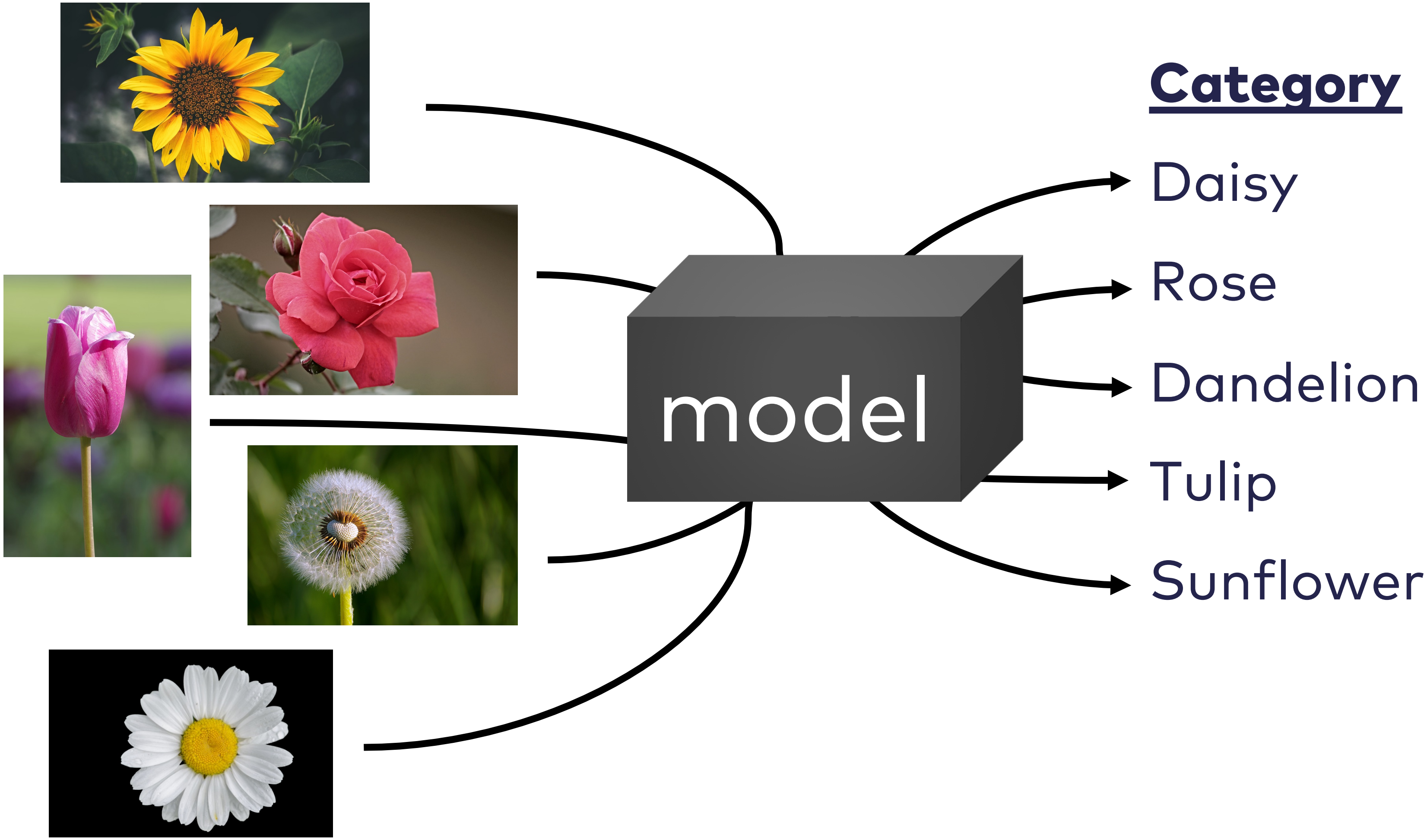
- consultant at INNOQ
- 15 years sortware development / architecture
- 2 years infrastructure / cloud
- 1 year machine learning / deep learning 💜

Agenda

- Problem we‘re solving
- The serverless solution
- Limitations, problems and how to deal with them
- Alternatives on the cloud
- Takeaways and best practices

Problem we‘re solving

Models



model

Category

Daisy

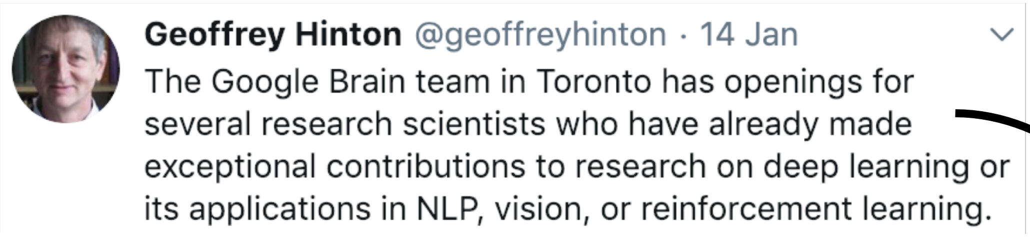
Rose

Dandelion

Tulip

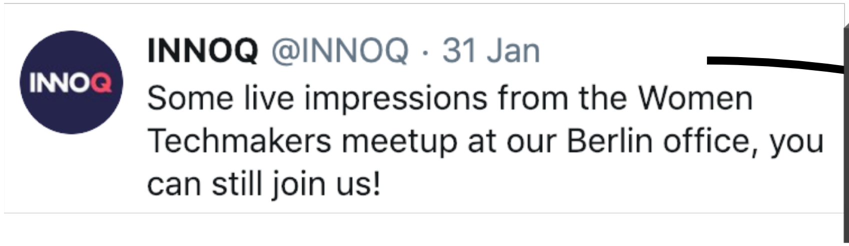
Sunflower

Models



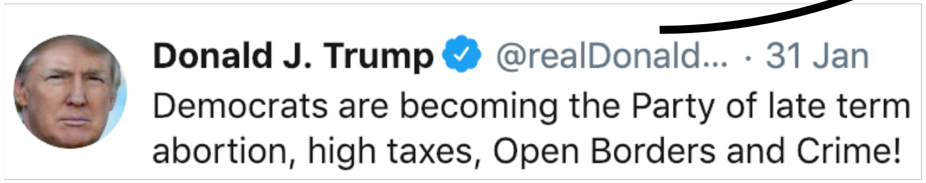
Geoffrey Hinton @geoffreyhinton · 14 Jan

The Google Brain team in Toronto has openings for several research scientists who have already made exceptional contributions to research on deep learning or its applications in NLP, vision, or reinforcement learning.



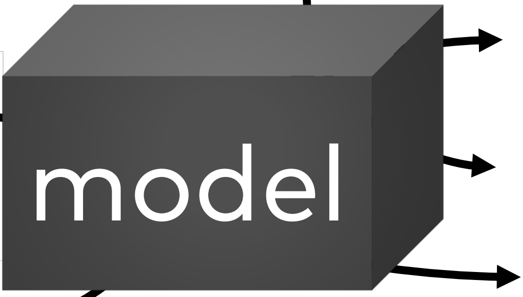
INNOQ @INNOQ · 31 Jan

Some live impressions from the Women Techmakers meetup at our Berlin office, you can still join us!



Donald J. Trump @realDonald... · 31 Jan

Democrats are becoming the Party of late term abortion, high taxes, Open Borders and Crime!



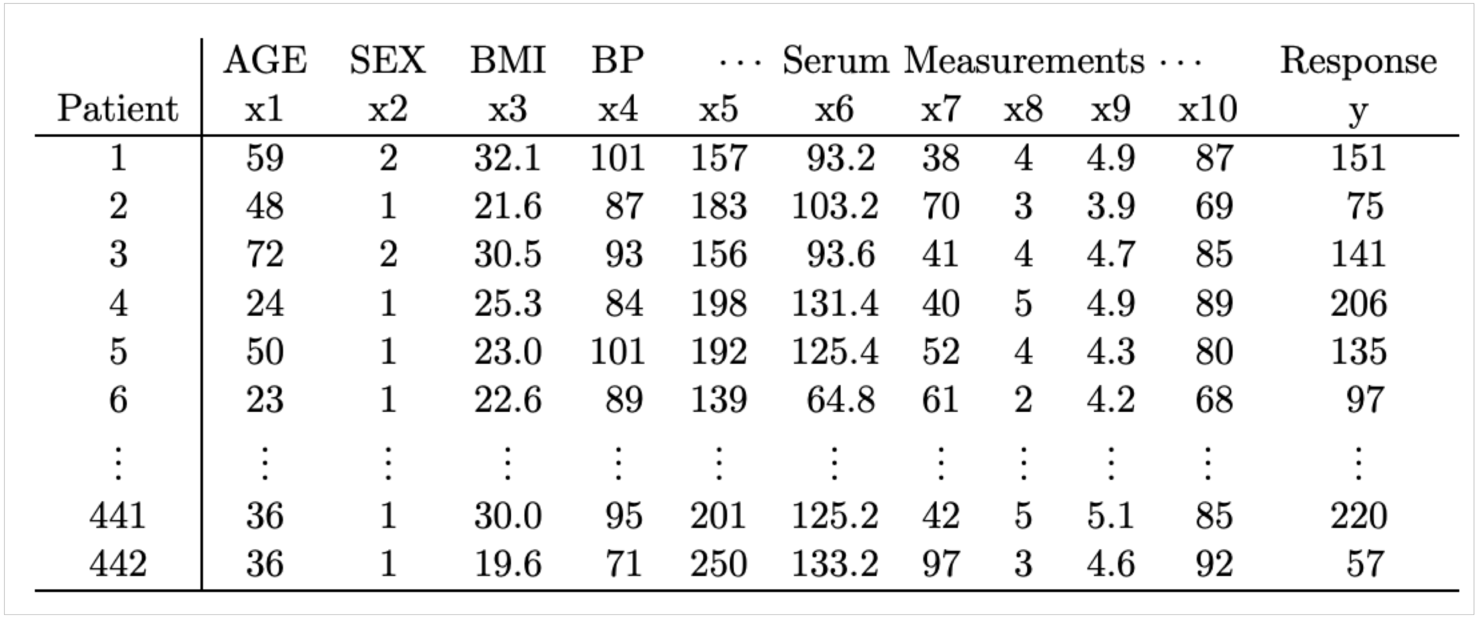
Category

Neutral

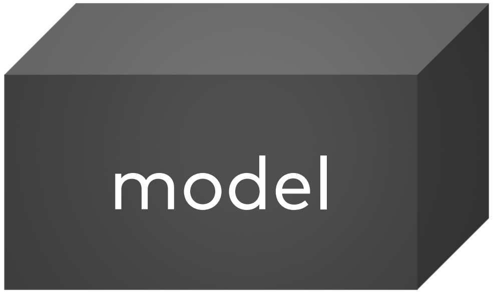
Offensive

Toxic

Models



	AGE	SEX	BMI	BP	··· Serum Measurements ···						Response
Patient	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	y
1	59	2	32.1	101	157	93.2	38	4	4.9	87	151
2	48	1	21.6	87	183	103.2	70	3	3.9	69	75
3	72	2	30.5	93	156	93.6	41	4	4.7	85	141
4	24	1	25.3	84	198	131.4	40	5	4.9	89	206
5	50	1	23.0	101	192	125.4	52	4	4.3	80	135
6	23	1	22.6	89	139	64.8	61	2	4.2	68	97
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
441	36	1	30.0	95	201	125.2	42	5	5.1	85	220
442	36	1	19.6	71	250	133.2	97	3	4.6	92	57



You built it, you run it?

You care for

- Model
- Hardware
- Network
- Runtime environment
- Security
- Packaging
- Deployment
- High availability
- Logging
- Monitoring / Alerting

You need friends...

Servers

Network

DevOps

Security

ML

Running ML service requires many people with different engeneering skills.

Can we run it with less effort?

The serverless solution

You built it, you run it?

You care for

- Model
- Hardware
- Network
- Runtime environment
- Security
- Packaging
- Deployment
- High availability
- Logging
- Monitoring / Alerting

Cloud

Google Cloud

amazon web services

ORACLE® CLOUD

HEROKU

IBM Bluemix

rackspace®

Azure

Cloud

Storage

- blob storage
- RDBMS
- NoSQL
- „File system“

Infrastructure

- VPN
- Content delivery
- Certificate authority

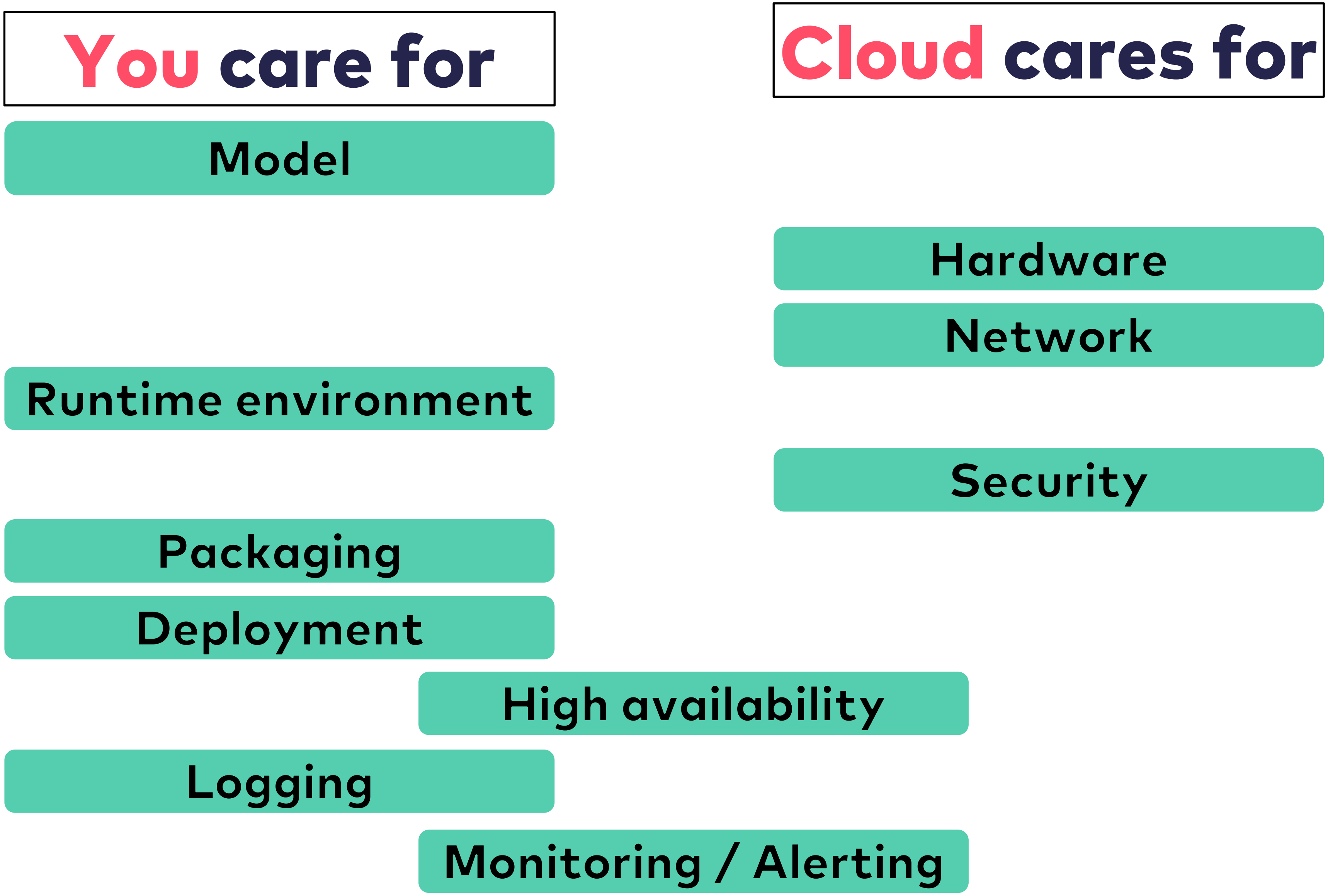
Computing power

- VM
- VM + Runtime
- Container platform
- Cloud function

Domain specific service

- Video transcoder
- Workflow
- Translation service

Cloud (VM)



You care for

Cloud cares for

Model

Hardware

Network

Runtime environment

Security

Packaging

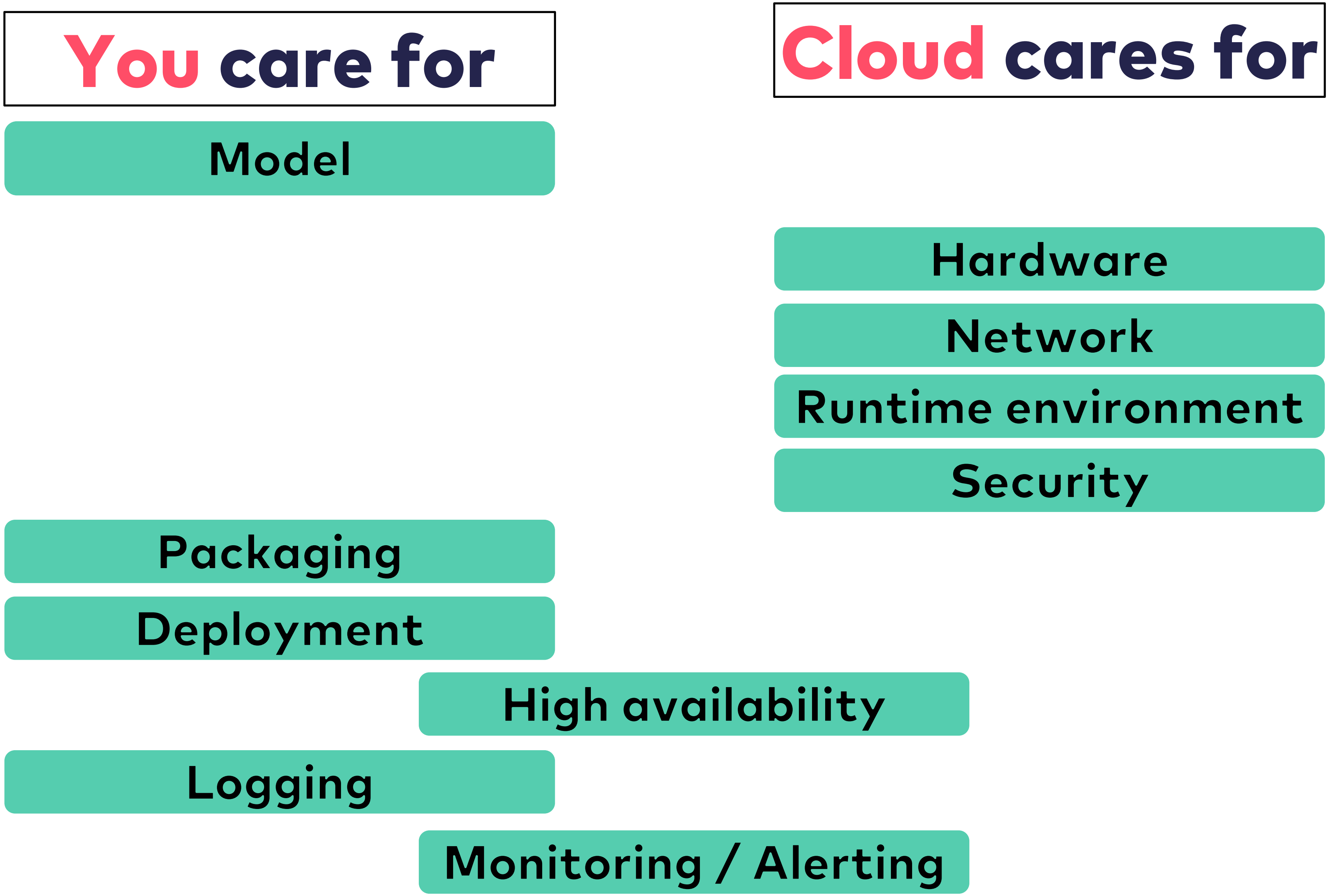
Deployment

High availability

Logging

Monitoring / Alerting

Cloud (VM + runtime)



You care for

Cloud cares for

Model

Hardware

Network

Runtime environment

Security

Packaging

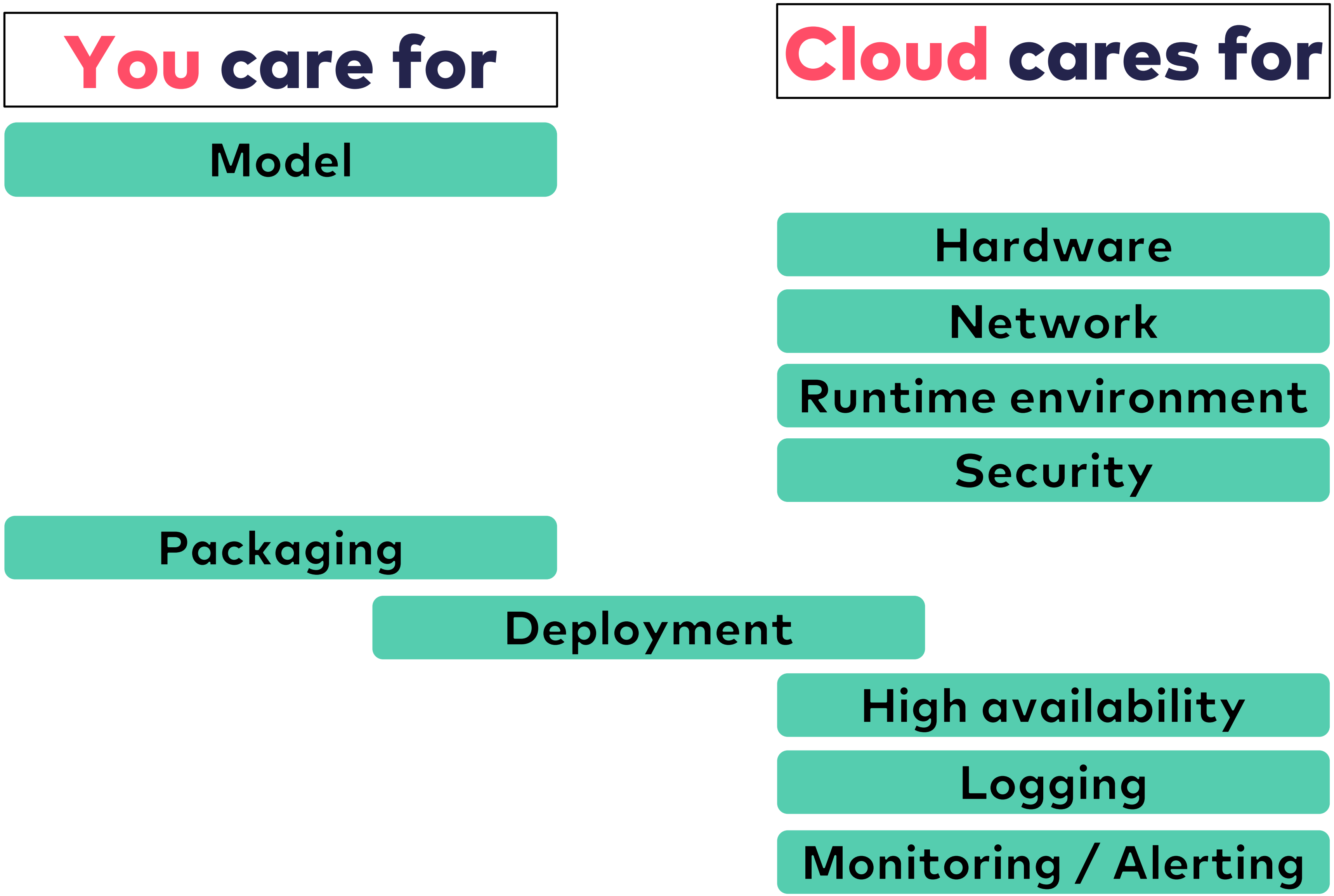
Deployment

High availability

Logging

Monitoring / Alerting

Cloud (cloud functions)



You care for

Cloud cares for

Model

Hardware

Network

Runtime environment

Security

Packaging

Deployment

High availability

Logging

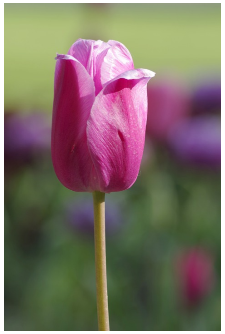
Monitoring / Alerting

Cloud functions

- highly pre-installed and pre-configured environments where you just put you code, and they care for the rest
- start on demand, fulfil the task, get terminated

Turn my inference code into cloud function (with AWS) – step 0

```
def run_inference_on_image(img):
    with tf.gfile.GFile('/tmp/retrained_graph.pb', "rb") as f:
        with tf.Session() as s:
            graph_def = tf.GraphDef()
            graph_def.ParseFromString(f.read())
            tf.import_graph_def(graph_def)
            inp_node = s.graph.get_tensor_by_name('import/input:0')
            out_node = s.graph.get_tensor_by_name('import/final_result:0')
            return s.run(out_node, feed_dict = {inp_node: img.eval()})[0]
```



Turn my inference code into cloud function (with AWS) – step 1

index.py

```
def which_flower(event, context):
    url = event.get('queryStringParameters').get('url')
    img = tf.image.decode_jpeg(url)
    result = run_inference_on_image()
    return  { 'statusCode': 200, 'body': json.dumps({ "return": result }) }

def run_inference_on_image(img):
    with tf.gfile.GFile('/tmp/retrained_graph.pb', "rb") as f:
        with tf.Session() as s:
            graph_def = tf.GraphDef()
            graph_def.ParseFromString(f.read())
            tf.import_graph_def(graph_def)
            inp_node = s.graph.get_tensor_by_name('import/input:0')
            out_node = s.graph.get_tensor_by_name('import/final_result:0')
            return s.run(out_node, feed_dict = {inp_node: img.eval()})[0]
```

Turn my inference code into cloud function (with AWS) – step 2

template.yaml

```
AWSTemplateFormatVersion: '2010-09-09’
  Transform: 'AWS::Serverless-2016-10-31’
  Resources: WhichFlower:
  Type: 'AWS::Serverless::Function’
    Properties:
      Handler: index.which_flower
      Runtime: python3.6
      Events:
        Api:
          Type: Api
          Properties:
            Path: /which_flower
            Method: get
```

Turn my inference code into cloud function (with AWS) – steps 3,4

Set up an AWS Account.
Set up the AWS CLI .
Create S3 bucket for archive

```
> sam package \
      --template-file template.yaml \
      --output-template-file serverless-output.yaml \
      --s3-bucket which_flower_bucket

> sam deploy \
      --template-file serverless-output.yaml \
      --stack-name new-stack-name \
      --capabilities CAPABILITY_IAM
```

DONE

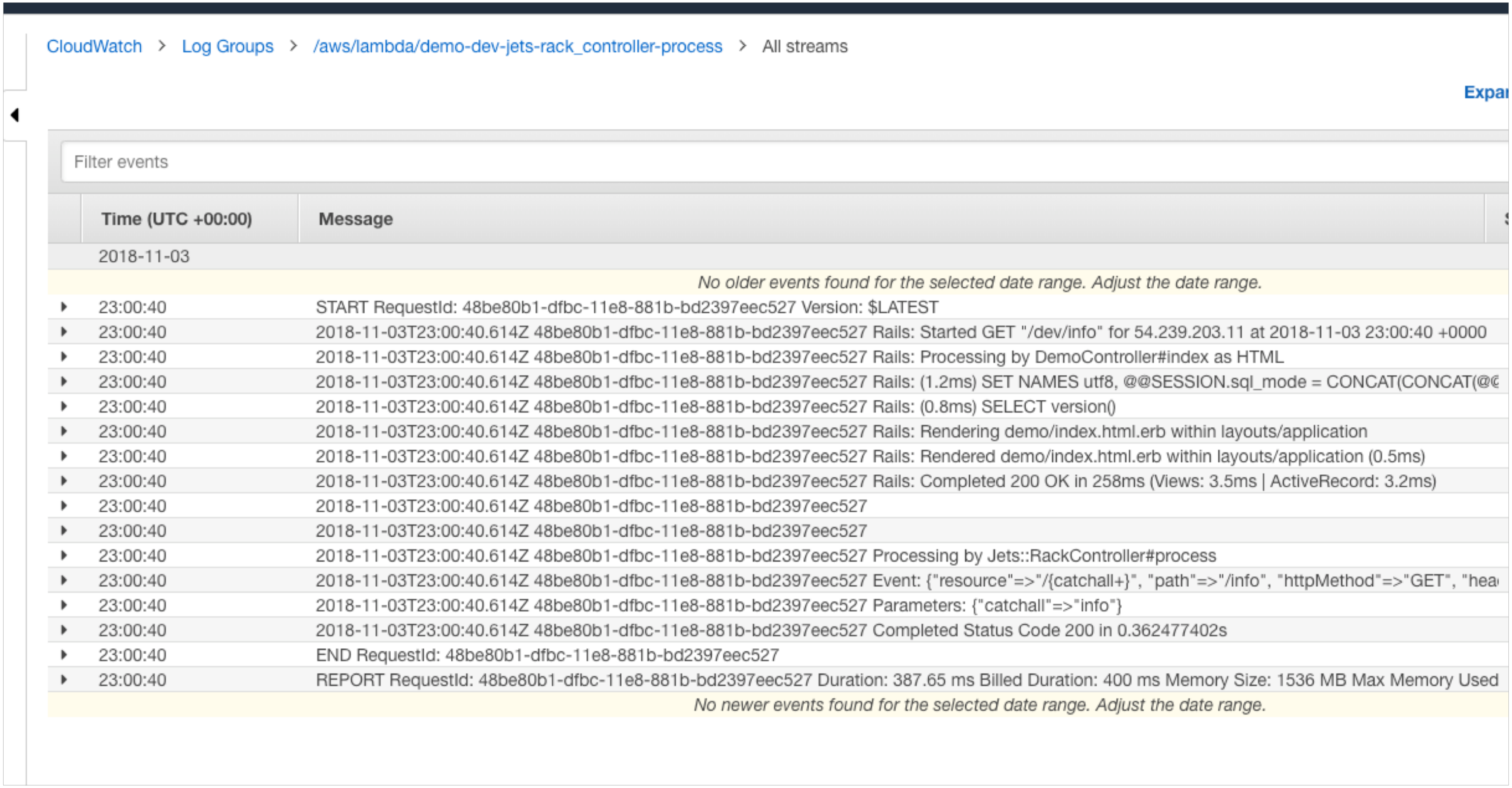
What you get: language support

AWS Lambda	Python (2.7, 3.6, 3.7) JavaScript (Node 6 & 8) Java 8, Go, PowerShell C# (.NET Core 1.0, 2.0, 2.1)
Google Cloud Functions	Python 3.7, Go JavaScript (Node 6 & 8)
Azure Functions	Python 3.6, Java 8 C#, F# (.NET Core 2) JavaScript (Node 8 & 10)

What I get: my function is always available

- start on demand, fulfil the tasks, get terminated => **any time so many instances as you need**

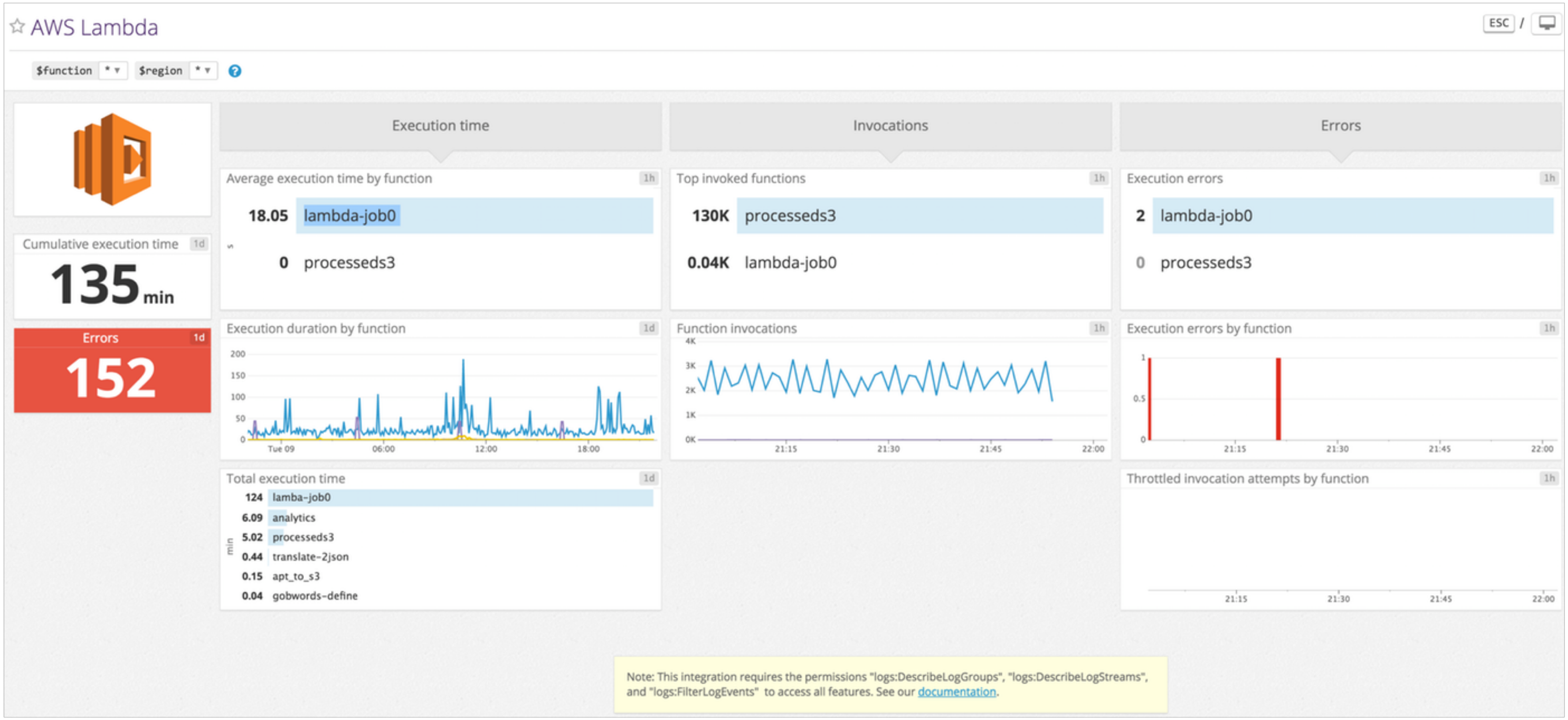
What I get: my function has logging



CloudWatch > Log Groups > /aws/lambda/demo-dev-jets-rack_controller-process > All streams
Expa
Filter events

```
Time (UTC +00:00)   Message
2018-11-03
                    No older events found for the selected date range. Adjust the date range.
23:00:40            START RequestId: 48be80b1-dfbc-11e8-881b-bd2397eec527 Version: $LATEST
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: Started GET "/dev/info" for 54.239.203.11 at 2018-11-03 23:00:40 +0000
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: Processing by DemoController#index as HTML
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: (1.2ms) SET NAMES utf8, @@SESSION.sql_mode = CONCAT(CONCAT(@@
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: (0.8ms) SELECT version()
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: Rendering demo/index.html.erb within layouts/application
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: Rendered demo/index.html.erb within layouts/application (0.5ms)
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Rails: Completed 200 OK in 258ms (Views: 3.5ms | ActiveRecord: 3.2ms)
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Processing by Jets::RackController#process
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Event: {"resource"=>"/{catchall+}", "path"=>"/info", "httpMethod"=>"GET", "hea
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Parameters: {"catchall"=>"info"}
23:00:40            2018-11-03T23:00:40.614Z 48be80b1-dfbc-11e8-881b-bd2397eec527 Completed Status Code 200 in 0.362477402s
23:00:40            END RequestId: 48be80b1-dfbc-11e8-881b-bd2397eec527
23:00:40            REPORT RequestId: 48be80b1-dfbc-11e8-881b-bd2397eec527 Duration: 387.65 ms Billed Duration: 400 ms Memory Size: 1536 MB Max Memory Used
                    No newer events found for the selected date range. Adjust the date range.
```

What I get: my function has monitoring/alerting



☆ AWS Lambda

$function * ▾ $region * ▾

Cumulative execution time 1d

135 min

Errors 1d

152

Execution time

Average execution time by function 1h

18.05	lambda-job0
0	processeds3

Execution duration by function 1d

Total execution time 1d

124	lamba-job0
6.09	analytics
5.02	processeds3
0.44	translate-2json
0.15	apt_to_s3
0.04	gobwords-define

Invocations

Top invoked functions 1h

130K	processeds3
0.04K	lambda-job0

Function invocations 1h

Errors

Execution errors 1h

2	lambda-job0
0	processeds3

Execution errors by function 1h

Throttled invocation attempts by function 1h

Note: This integration requires the permissions "logs:DescribeLogGroups", "logs:DescribeLogStreams", and "logs:FilterLogEvents" to access all features. See our documentation.

What I get: my function can be **triggered by other cloud services**

New entry in blob storage

HTTP call

Message from message queue

Scheduled

New/changed DB entry

Incoming Email

and many more…

What I get: my function can be **triggered from outside**

HTTP call

Cloud functions as microservices

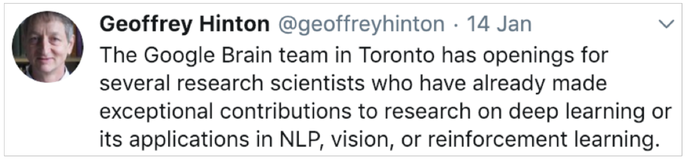
Microservice architectural style is an approach to developing a single application as a **suite of small services**, each running in its own process and communicating with lightweight mechanisms, often an **HTTP resource API**. These services are built around business capabilities and **independently deployable** by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in **different programming languages** and use **different data storage technologies**.

(Martin Fowler, James Levis)

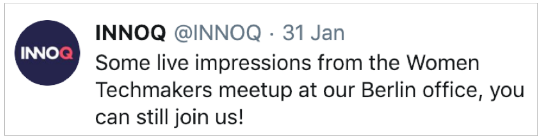
Limitations, problems and how to deal with them

How we checked: 3 typical use cases with 3 well known frameworks

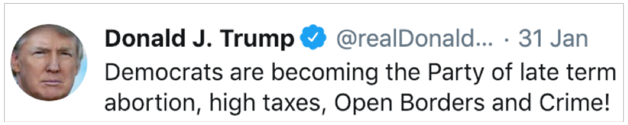
Sentiment analysis
SpaCy
10M values (word vectors)



Geoffrey Hinton @geoffreyhinton · 14 Jan
The Google Brain team in Toronto has openings for several research scientists who have already made exceptional contributions to research on deep learning or its applications in NLP, vision, or reinforcement learning.



INNOQ @INNOQ · 31 Jan
Some live impressions from the Women Techmakers meetup at our Berlin office, you can still join us!



Donald J. Trump @realDonald... · 31 Jan
Democrats are becoming the Party of late term abortion, high taxes, Open Borders and Crime!

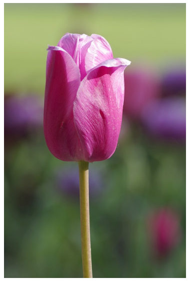
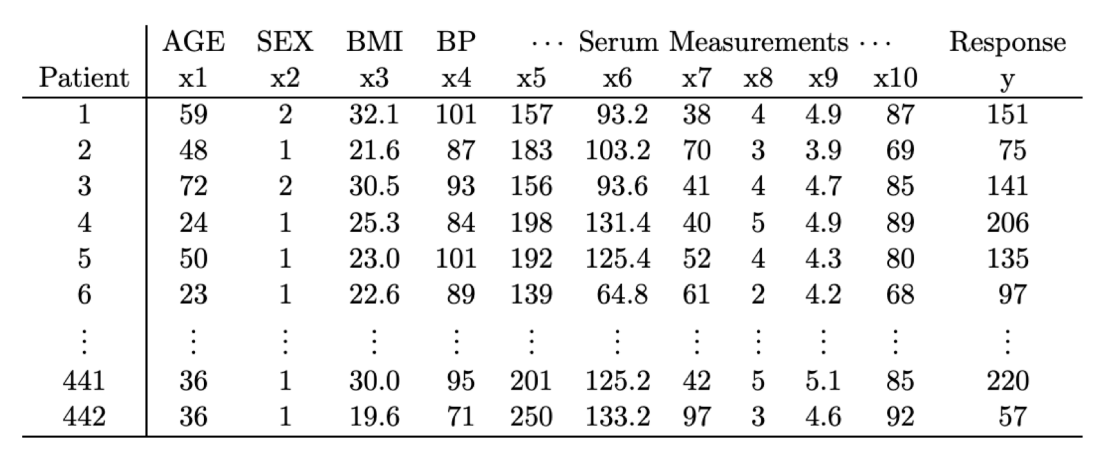


Image classification
Tensorflow
4M values (retrained MobileNet)

Structured data
Scikit-learn
1K values
(Random Forest)



	AGE	SEX	BMI	BP	· · · Serum	Measurements · · ·					Response
Patient	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	y
1	59	2	32.1	101	157	93.2	38	4	4.9	87	151
2	48	1	21.6	87	183	103.2	70	3	3.9	69	75
3	72	2	30.5	93	156	93.6	41	4	4.7	85	141
4	24	1	25.3	84	198	131.4	40	5	4.9	89	206
5	50	1	23.0	101	192	125.4	52	4	4.3	80	135
6	23	1	22.6	89	139	64.8	61	2	4.2	68	97
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
441	36	1	30.0	95	201	125.2	42	5	5.1	85	220
442	36	1	19.6	71	250	133.2	97	3	4.6	92	57

How we checked:
3 major cloud providers

amazon
web services

AWS Lambda

Azure

Azure Functions

Google Cloud

Google Cloud Functions

How we checked:
trying to make it run

https://github.com/innoq/ml_serverless

Limitation 1: no GPU

	GPU	CPU
AWS Lambda	X	128 to 3008 MB
Google Cloud Functions	X	128 to 2048 MB
Azure Functions	X	128 to 1536 MB

What can be done: nothing

Limitation 2:
deployable artifact size

AWS Lambda	256 MB uncompressed 😢
Google Cloud Functions	500 MB uncompressed
Azure Functions	No limit

Limitation 2:
deployable artifact size

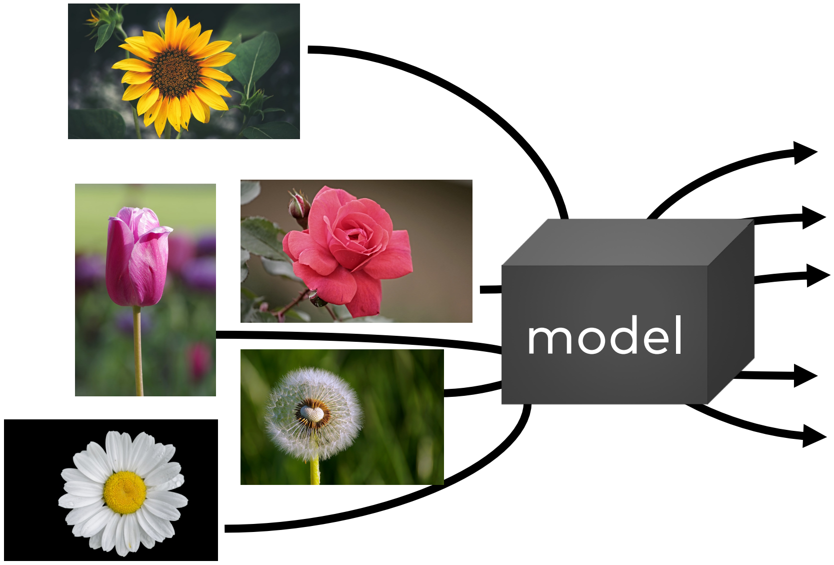
What can be done: **use pip/bash to package only what you need in minimal size**

https://github.com/Accenture/serverless-ephemeral/blob/master/docs/build-tensorflow-package.md

https://github.com/antonpaquin/Tensorflow-Lambda-Layer

Limitation 3: cold start

„start on demand, fulfil the task, get terminated“



Category

Daisy

Rose

Dandelion

Tulip

Sunflower

Init model – 20s

Evaluate image – 4s

(retrained MobileNet, 4M values,
AWS Lambda with 1GB memory)

Limitation 3: cold start

What can be done:

- **use hacks to keep your functions warm**
- **declare expensive resources as global variables so that they will be cached with a function**

https://mikhail.io/2018/08/serverless-cold-start-war/

Performance

			Warm	Cold
Image classification, Tensorflow, 4M values	Google, JavaScript	1GB	**4.3s**	**17s**
Image classification, Tensorflow, 4M values	AWS, Python	1GB	**4s**	**15s**
Sentiment analysis, SpaCy, 10M values	Google, Python	1GB	**0.15s**	**22s**
Structured data Scikit-learn, 1K values	Google, Python	256MB	**0.28s**	**0.38s**
Structured data Scikit-learn, 1K values	Microsoft, Python	256MB	**0.25s**	**0.7s**

Limitation 4: hard to test offline

- API which cloud uses to call your function
- API of cloud services your function calls

are available in the cloud only

Deployment takes 3-5 minutes

Limitation 4: hard to test offline

What can be done: try offline emulators

For AWS: https://www.npmjs.com/package/serverless-offline or https://github.com/localstack/localstack
For Google: https://cloud.google.com/functions/docs/emulator
For Azure: https://docs.microsoft.com/de-de/azure/azure-functions/functions-develop-local

Limitation 5: may get expensive

Spontaneous use	👌
Heavy load, multiple req/s	💰

https://www.trek10.com/blog/lambda-cost/

Limitation 6: cloud APIs are proprietary and not standardised

Plan some extra effort if you decide to move

Alternatives on the cloud

Alternative

docker

+

kubernetes

by AWS
by Google
by Azure
on premise

Giant Swarm

HEROKU

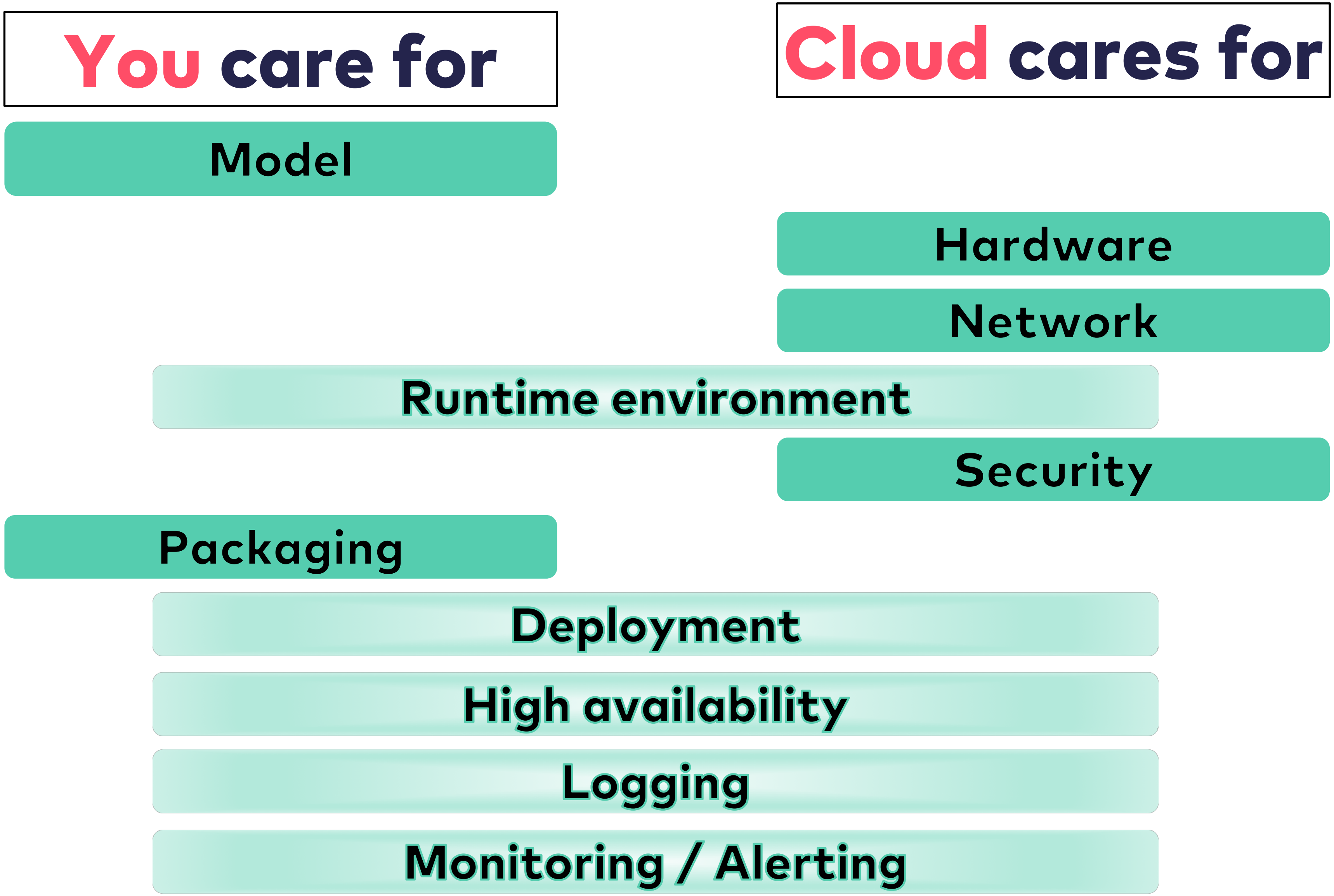
General purpose
container platforms

Amazon
SageMaker
aws

SELDON

ML focused
container platform

Docker-based approach



You care for

Cloud cares for

Model

Hardware

Network

Runtime environment

Security

Packaging

Deployment

High availability

Logging

Monitoring / Alerting

Takeaways and best practices

Takeaways

It works

Adds more value if you use other cloud services as well

Fits best for fast leightweight models

Fits best for unheavy load scenarios

Best practices

Declare you model as global variable

Try offline emulators

You can convert your model to run from other languages

Credentials

Michael Krämer, Leonardo Ramirez, Philipp Beyerlein, Phillip Ghadir, Christian Stettler (INNOQ)

Image sources

- Unsplash.com
- Wikipedia.org
- http://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf

Thank you!
Questions?

INNOQ
www.innoq.com

Michael Perlin
Michael.Perlin@innoq.com

☎ +49 178 7818063
🐦 @ttzt_mp

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin
Germany

Ludwigstr. 180E
63067
Offenbach
Germany

Kreuzstr. 16
80331
München
Germany

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
+41 41 743 01 11

Albulastr. 55
8048 Zürich
Switzerland