THE DEVELOPER'S CONFERENCE

Using a Serverless Architecture to Deploy (and consume) Machine Learning models

Rafael Zotto
Senior Software Architect, HP Inc.

Short Bio

Rafael Zotto

Holds a master degree in Computer Science focused in high performance computing. Specialized in parallel and distributed computing with special interest in mobile and web technologies. Works for HP Inc. for the past decade acting as senior software architect for print firmware and wearable technologies. Recently joined the Data Science research team in Porto Alegre, Brazil.

Agenda

- Problem Statement
- Serverless Overview
- Deploy and Consume ML: A practical Use Case

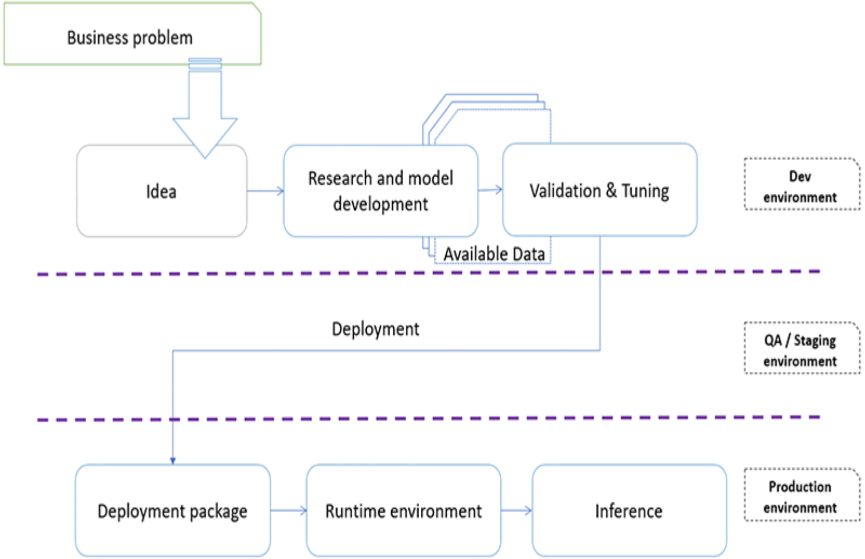
Problem Statement

- The model is trained;
- The results are acceptable;

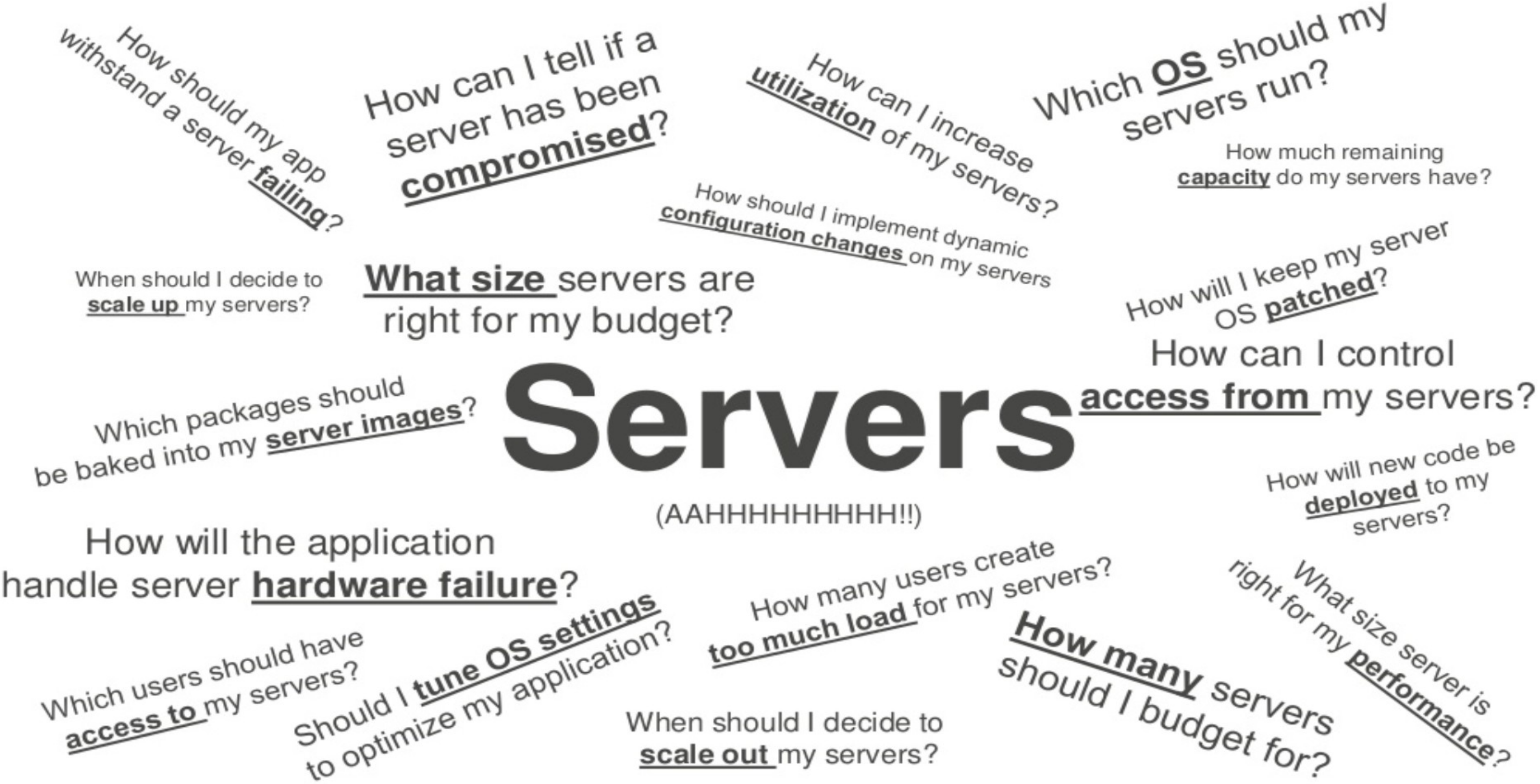
- How to share it ‘*with the world*’?
  - Talk goal: to share a ***Serverless Approach***

Process Perspective

- Development and Deployment of ML systems **should not be different** from traditional software solutions.



* Image from "*How to Deploy Deep Learning Models with AWS Lambda and Tensorflow*" article. Amazon Web Services Blogs.



How should my app withstand a server **failing**?

How can I tell if a server has been **compromised**?

How can I increase **utilization** of my servers?

Which **OS** should my servers run?

How much remaining **capacity** do my servers have?

How should I implement dynamic **configuration changes** on my servers

When should I decide to **scale up** my servers?

What size servers are right for my budget?

How will I keep my server OS **patched**?

How can I control **access from** my servers?

Which packages should be baked into my **server images**?

Servers

(AAHHHHHHHHH!!)

How will new code be **deployed** to my servers?

How will the application handle server **hardware failure**?

How many users create **too much load** for my servers?

What size server is right for my **performance**?

Which users should have **access to** my servers?

Should I **tune OS settings** to optimize my application?

When should I decide to **scale out** my servers?

How many servers should I budget for?

* Image from "*Getting Started with Serverless Architecture*" presentation. Amazon Web Services.

Serverless Definition

- Platform to develop, run and manage applications without the complexity of building and maintaining infrastructure.
- No free lunch!
  - You will pay for it.
  - Sub-second billing

Architect to be Serverless

- Fully Managed
  - No provisioning, zero administration, high-available
- Developer Productivity
  - Focus on what matters, innovate quickly
- Continuous Scaling
  - Up and Down automatically

Simple Use Case

- **Model** was previously trained;
- **Deploy** it to *a* cloud environment;
- **Execute** real-time predictions;

Simple Use Case

- **Model** – IRIS Data set
- **Deploy** – AWS
- **Execute** – AWS IoT Core Stack;

Model

- The **Iris flower data** set is a **multivariate data set**
- Introduced by the British statistician and biologist Ronald Fischer (1936)

Iris Setosa

Iris Versicolor

Iris Virginica

Deployment

- AWS Lambda functions for predictions
- Model saved in a S3 bucket

- The **Serverless Framework** might be your friend here.

Real-Time Prediction

- IoT Core stack (just because we want it ‘real-time’)
- MQTT Communication (Lambda ↔ Client)

ENOUGH TALKING…
SHOW ME THE CODE.

Tips and Issues

- Take advantage of AWS Lambda **container reuse**
- Keep you function **warm**!

THE DEVELOPER’S
CONFERENCE