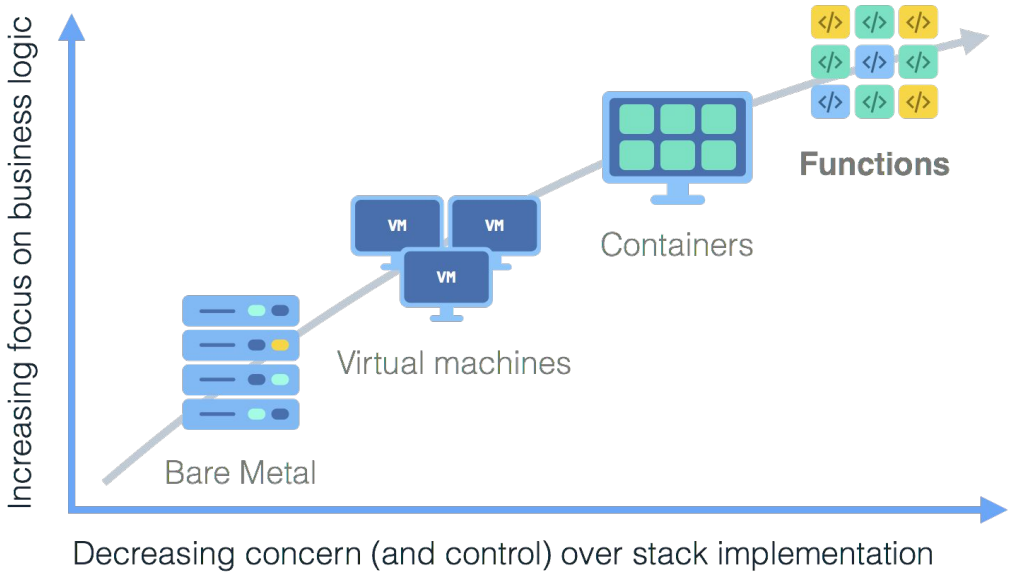
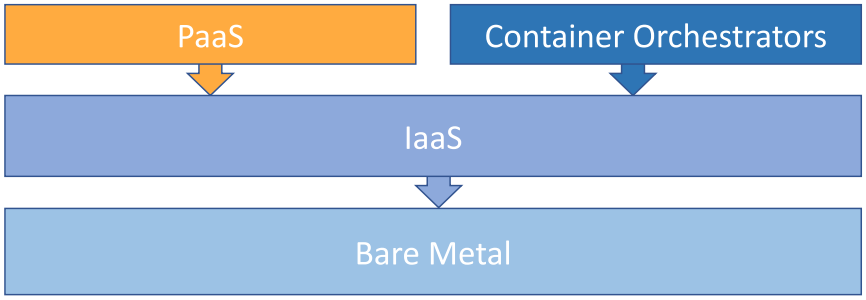
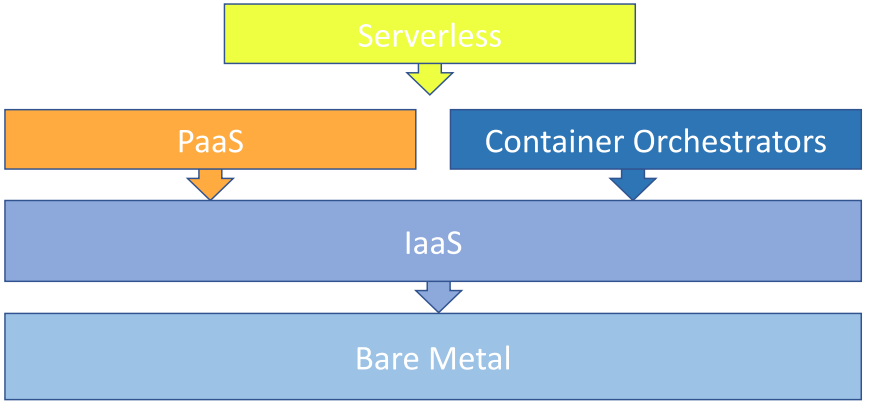
Serverless Computing

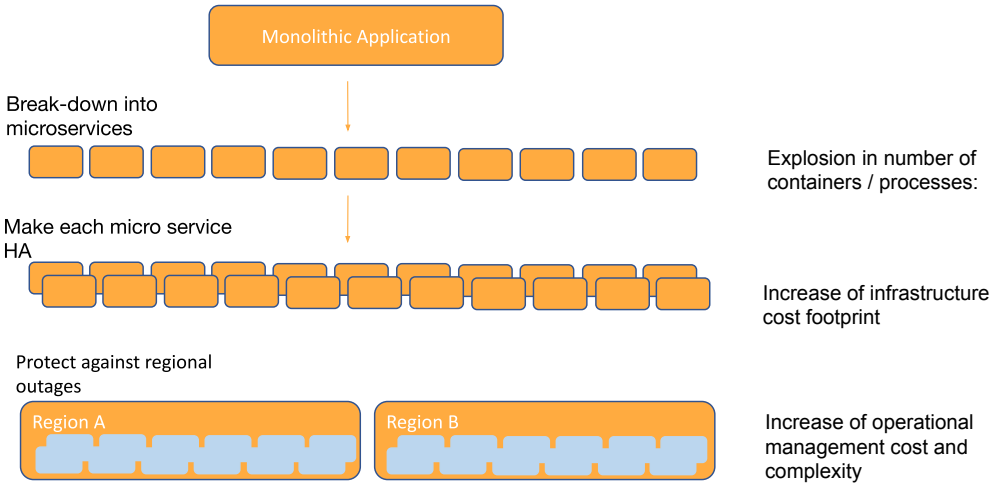


Evolution Of Serverless



Enter Serverless





Explosion in number of containers / processes:

Increase of infrastructure cost footprint

Increase of operational management cost and complexity

What is Serverless?

a cloud-native platform

for

short-running, stateless computation

and

event-driven applications

which

scales up and down instantly and automatically

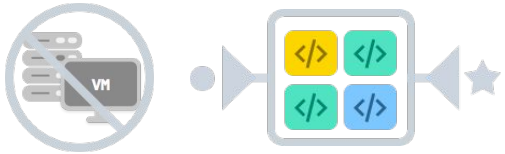
and

charges for actual usage at a millisecond granularity

Server-less means no servers?
Or worry-less about servers?

Runs code **only** on-demand on a per-request basis

Serverless deployment & operations model



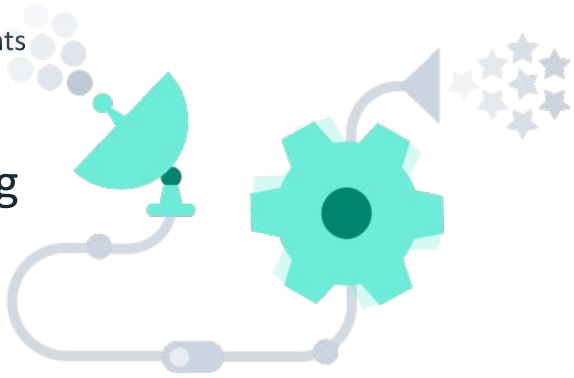
No servers

Just code

What triggers code execution?

Runs code **in response** to events

Event-programming model



Why is Serverless attractive?

- Making app development & ops dramatically faster, cheaper, easier
- Drives infrastructure cost savings

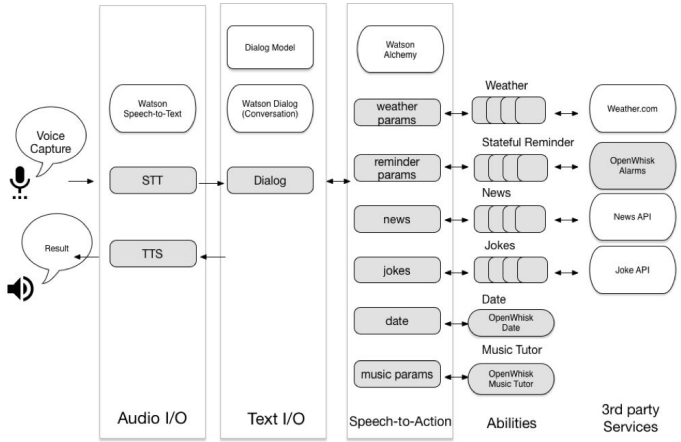
	On-prem	VMs	Containers	Serverless
Time to provision	Weeks-months	Minutes	Seconds-Minutes	Milliseconds
Utilization	Low	High	Higher	Highest
Charging granularity	CapEx	Hours	Minutes	Blocks of milliseconds

Source: Jason McGee, IBM; Serverless Conference 2017.

Key factors for infrastructure cost savings

	Traditional models (CF, containers, VMs)	Serverless
High Availability	At least 2-3 instances of everything	No incremental infrastructure
Multi-region deployment	One deployment per region	No incremental infrastructure
Cover delta between short (<10s) load spikes and valleys (vs average)	~2x of average load	No incremental infrastructure
Example incremental costs	2 instances x 2 regions x 2 = 8x	1x

Chatbots



Building a Chatbot with Serverless Computing, Yan, Mengting and Castro, Paul and Cheng, Perry and Ishakian, Vatche, Proceedings of the 1st International Workshop on Mashups of Things and APIs 2016

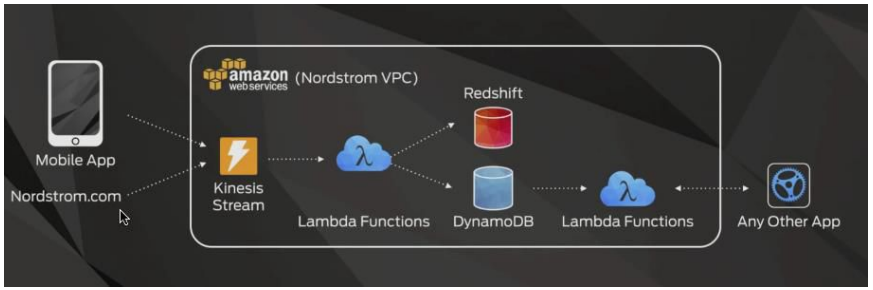
PyWren: a massive data framework for Lambda

- Open Source MapReduce framework using Lambda
- Word count job on 83M items is only 17% slower than PySpark running on dedicated servers.
- Sort 1TB data in 3.4 minutes (Spark 100TB in 23 min)

https://github.com/pywren/pywren
http://pywren.io/

Occupy the Cloud: Distributed Computing for the 99%, Eric Jonas, Qifan Pu, Shivaram Venkataraman, Ion Stoica, Benjamin Recht, https://arxiv.org/abs/1702.04024

Nordstrom Recommendations



15-20 minutes of processing → now in seconds
2x order of magnitude for cost savings

What is Serverless good for?

Serverless is **good** for

short-running
stateless
event-driven

- Microservices
- Mobile Backends
- Bots, ML Inferencing
- IoT
- Modest Stream Processing
- Service integration

Serverless is **not good** for

long-running
stateful
number crunching

- Databases
- Deep Learning Training
- Heavy-Duty Stream Analytics
- Numerical Simulation
- Video Streaming

Current Platforms for Serverless

APACHE OpenWhisk

OpenLambda

Iron.io

AWS Lambda

Azure Functions

fission

Kubernetes

IBM Cloud Functions

Red-Hat

Google Functions

fn