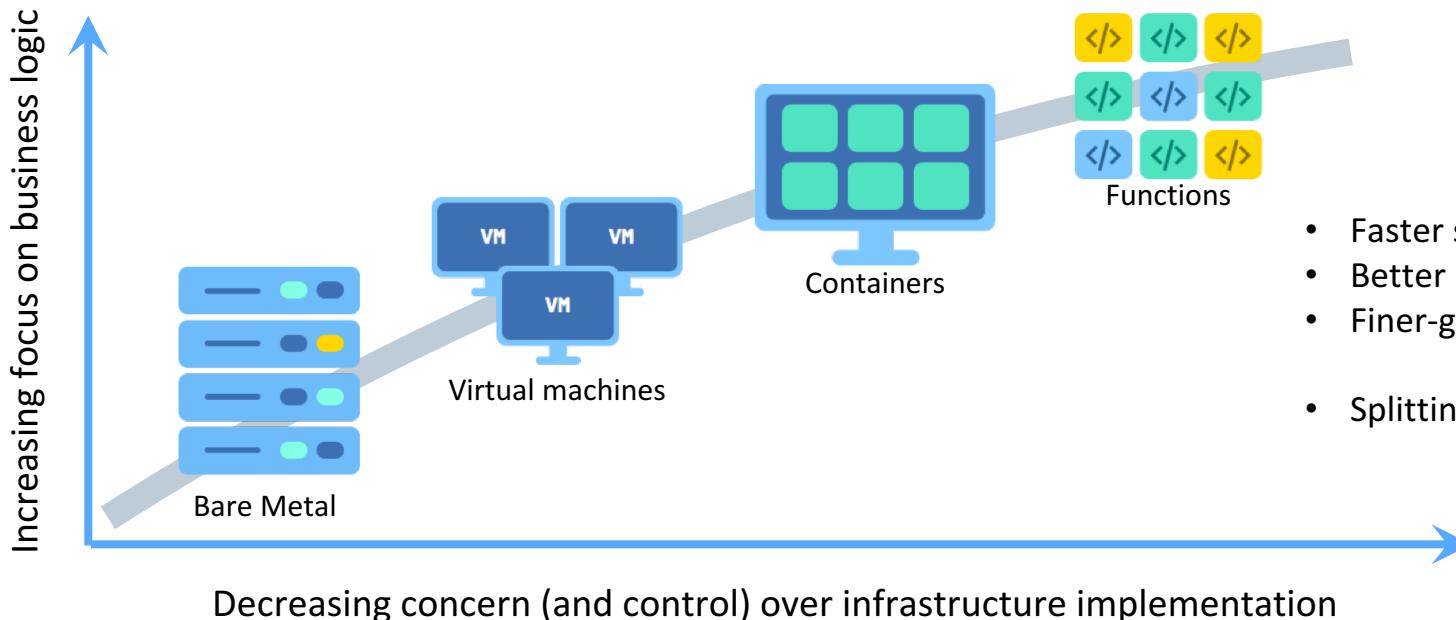# State of Serverless

Mark Peek, Principal Engineer, VMware (markpeek@vmware.com)
Doug Davis, STSM, IBM (dug@us.ibm.com)

Agenda

- What is Serverless?

- Why and when to use Serverless?

  - Serverless vs …

  - Use cases

- CNCF Serverless Working Group

- Future of eventing

But first...Functions as a Service (FaaS)

Increasing focus on business logic

Functions

Containers

Virtual machines

VM

VM

VM

Bare Metal

- Faster start-up times
- Better resource utilization
- Finer-grained management

- Splitting up the monolith

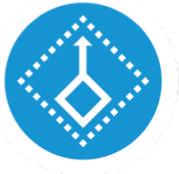Decreasing concern (and control) over infrastructure implementation

What is a Function?

EVENT DRIVEN

SHORT DURATION

STATELESS

AUTO-SCALED

LOWER COST

What is a Function?

- Example:

```javascript
  /* Javascript example */
function main(params) {
    var name = params.name || 'World';
    return {payload: 'Hello, ' + name + '!'};
}
```
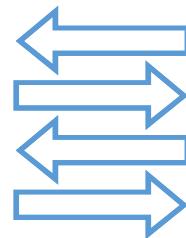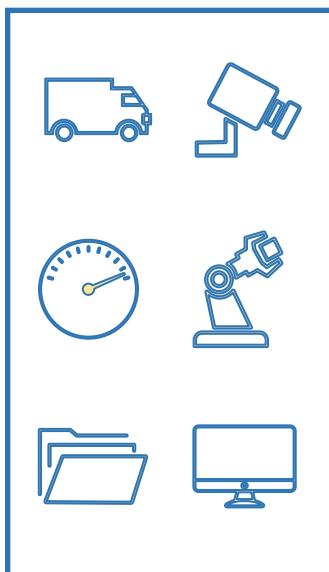
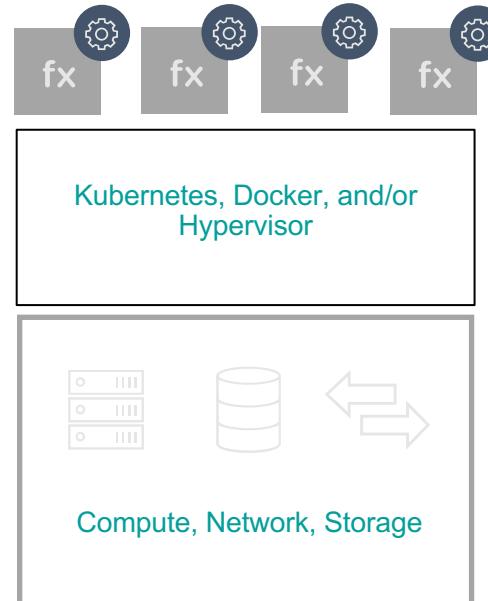e.g. https://openwhisk.ng.bluemix.net/api/v1/web/dug/default/test

- Framework handles hosting and infrastructure to deal with incoming messages and response
  - Provide access via an HTTP(s) API
  - Connect to a set of "Actions"
  - Chaining functions to orchestrate

Back to serverless...

- Serverless takes FaaS and adds the notion of:
  - Infrastructure manages the **auto-scaling of the functions based on demand**

  - Infrastructure manages the **scaling down to zero instances when not being invoked**

  - Fine grained, pay just for what you use, cost model
  - **Zero cost when not being executed (more on this later)**
  - Except for persistent storage type of resources

- "Serverless" means not needing to worry about managing the server

Serverless vs PaaS / CaaS

- Very similar
  - Especially if PaaS / CaaS has auto-scaling feature

- Similar mind-shift for VM -> PaaS / CaaS
  - Remove the OS and just deploy your app
  - Remove the "app" and just deploy a "set of functions (APIs)"

- Biggest difference is the scope of the code being deployed

- Function vs Application
  - E.g. can scale just "GETs" vs "Entire App" based on demand
  - Decompose monolithic app to individual function endpoints

Serverless/FaaS use cases

	micro-service	Easily implement fine-grained, micro-service APIs.
	IoT	Power various mobile, web and IoT app use cases by scaling and simplifying the programming model of orchestrating various services.
	Batch and Stream Processing	Automate and control batch and stream processing
	DevOps	Automate DevOps pipeline based on events triggered from successful builds or completed staging or a go-live event.
	IT/Ops	Allow an easier deployment model for administrative functions (bots) to run for IT/Ops.

CNCF Serverless Working Group

CNCF Serverless WG

- Formed in June 2017 at the request of CNCF Technical Oversight Committee (TOC)

- Asked for state of tech/community & recommendations for possible involvement

- Most key Serverless players involved
  - IBM, VMWare, Google, Red Hat, Huawei, Microsoft, SolarWinds, Docker, iguazio, Amazon, MasterCard, Pivotal, Serverless Inc., Clay Labs, The New Stack, A Cloud Guru, Platform9, Bitnami, Auth0, and Hyper

CNCF Serverless WG: White Paper

- Describes Serverless platforms as they exist today using a common vocabulary

  - Zero cost when idle (except e.g. stateful storage costs)
  - Public vs Private Cloud

  - Roles: Provider vs Developer
  - Serverless vs Serverless Technology

CNCF Serverless WG: White Paper

- Describes & defined Serverless

- Highlights promising use cases and areas where they've already proven value

- Differentiates Serverless from PaaS and Container Orchestration

- Describes the mechanics of a generic Serverless system

- Identifies potential future "harmonization" the WG could look at

CNCF Serverless WG: Landscape

Project Name (Serverless/FaaS)	Sponsors	Homepage	Orchestration	Languages
AWS Lambda	Amazon	https://aws.amazon.com/lambda/		Node.js (JavaScript), Pyth
Google Cloud Functions	Google	https://cloud.google.com/functions/		Node.js
Hyper Func	Hyper.sh	https://docs.hyper.sh/Feature/container/func.html		Any language, Docker im
IBM Cloud Functions	IBM	https://console.bluemix.net/openwhisk/		Node.js, Swift, Python, Ja
Iguazio Data Platform	iguazio	https://www.iguazio.com/		Go/C (native), Python/Jav
Microsoft Azure Cloud Functions	Microsoft	https://azure.microsoft.com/en-us/services/functions/		Node.js, C#, F#, Python,
Huawei Function Stage	Huawei	https://www.huaweicloud.com/product/functionstage.html	Kubernestes	Node.js, Python, Java, go
Apache OpenWhisk	OSS	https://github.com/apache/incubator-openwhisk	Kubernetes, Standalone, Docker	Node.js, Python, Java, PH
fission.io	OSS	https://github.com/fission/fission	Kubernetes	NodeJS, Python, Go, Rub
OpenFaaS	OSS	https://github.com/openfaas/faas	Docker Swarm, Kubernetes, any other	Python, Ruby, C#, Node.j
Iron.io functions	OSS	https://github.com/iron-io/functions	Any that supports Docker images	Any language, Docker im
kubeless	OSS	https://github.com/kubeless/kubeless	Kubernetes	Python, Node.js, Ruby
microcule	OSS	https://github.com/Stackvana/microcule	Any Node.js HTTP middleware	Over 20 languages
Nuclio (by iguazio)	OSS	https://github.com/nuclio/nuclio	Docker, Kubernetes, Single binary	Go/C (native), Python/Jav

| + | ≡ | **1** Projects ▾ | **1** Dev Tools ▾ | Services ▾ | | Explo |

CNCF Serverless WG: Recommendations

- Maintain the landscape of Serverless implementations and features

- Potential collaboration / harmonization on:
  - Event format
  - Function definition / packaging & deployment / workflow

  Developer Interop

- Produce additional documents and samples that educate community
- Document integration with other CNCF projects, such as how to monitor and observe

CNCF Serverless WG: Events

- Have agreement from CNCF TOC to work on Events

- Creating a common model for event data, similar to CNI and CSI
  - Useful across entire Cloud Native deployments, beyond just Serverless

- Considering a few proposals as a starting point:
  - OpenEvents – (Serverless, Inc.)
  - ~~Cloud-Native Event Mapping (CNEM) – (iguazio)~~
  - ~~Cloud Auditing Data Federation (CADF) – (DMTF, IBM)~~

- Come join in!

Additional Information

- CNCF Serverless Working Group
  - https://github.com/cncf/wg-serverless
  - Weekly calls on Thursdays at 8am PT

- CNCF Serverless Working Group White Paper
  - https://docs.google.com/document/d/1UjW8bt5O8QBgQRILJVKZJej_IuNnxl20AJu9wA8wcdI

- CNCF Serverless Landscape
  - https://docs.google.com/spreadsheets/d/10rSQ8rMhYDgf_ib3n6kfzwEuoE88qr0amUPRxKbwVCk

Schedule for Serverless Track

Meeting Room 9AB - Level 3

11:10am - 11:45am	State of Serverless [B] - Mark Peek, VMware & Doug Davis, IBM
11:55am - 12:30pm	Building Serverless Application Pipelines [A] - Sebastien Goasguen, Bitnami
2:00pm - 2:35pm	FaaS and Furious - 0 to Serverless in 60 Seconds, Anywhere - Alex Ellis, ADP
2:45pm - 3:20pm	Fission: Serverless Functions for Kubernetes [B] - Soam Vasani, Platform9 Systems
3:50pm - 4:25pm	Pontoon: An Enterprise Grade Serverless Framework Using Kubernetes – As Used in VMware Cloud Services - Kumar Gaurav & Mageshwaran Rajendran, VMware
4:35pm - 5:10pm	Building and Running an Enterprise-grade Serverless Platform on Kubernetes - Ying Huang & Quinton Hoole, Huawei

Questions

Thank you!

Mark Peek
Principal Engineer, VMware (markpeek@vmware.com)
markpeek   @markpeek

Doug Davis
STSM, IBM (dug@us.ibm.com)
duglin   @duginabox