# Serverless:
## the next major shift in cloud computing

HELLO!

# I am Doug Vanderweide

MCSE, MCSD, CTT+

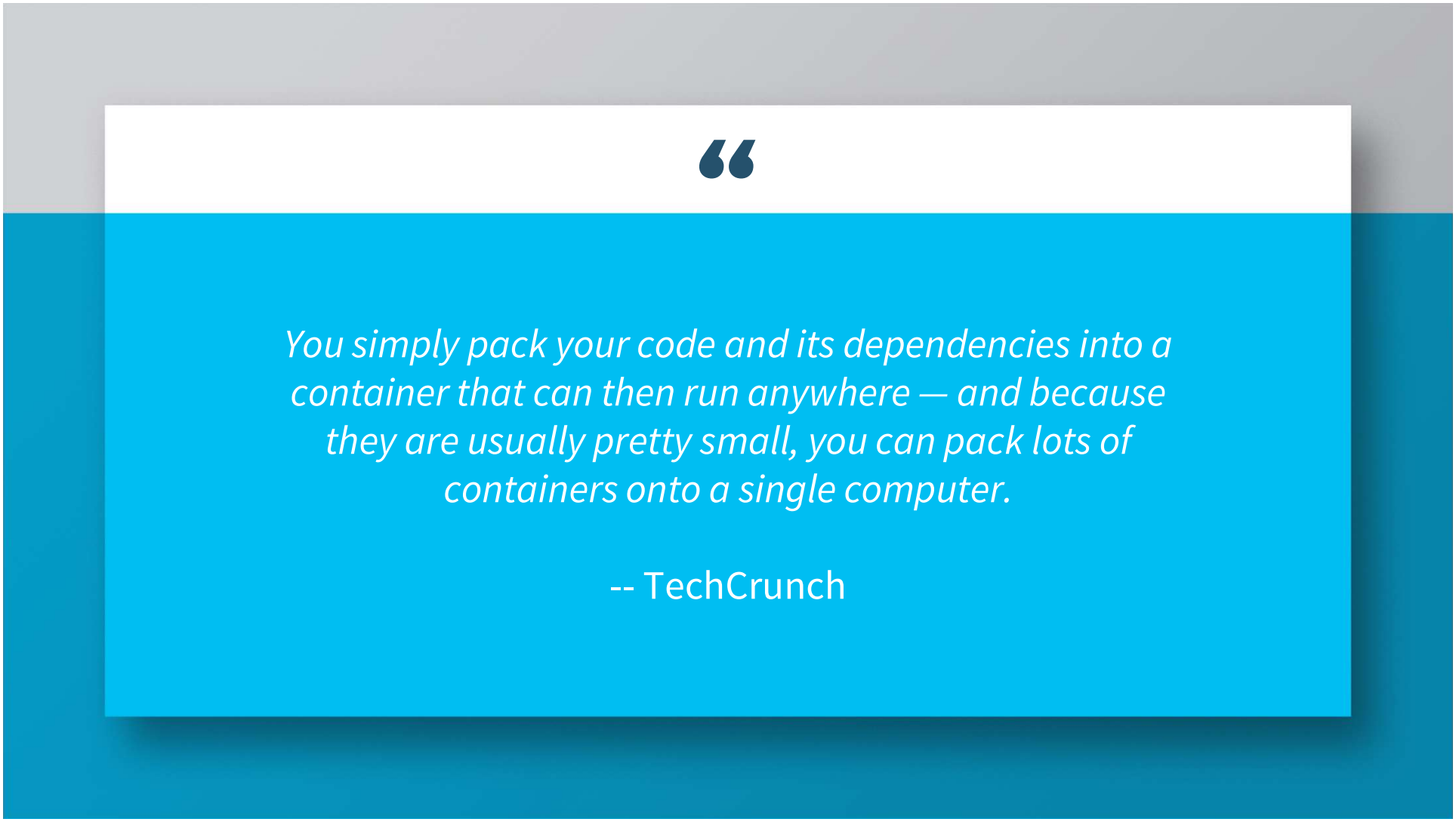
20+ years in software architecture, development and DevOps
Azure SME and instructor for LinuxAcademy.com

@dougvdotcom
linkedin.com/in/dougvdotcom

# 1.

## Containers are the hot new thing

And for really great reasons


“
You simply pack your code and its dependencies into a container that can then run anywhere — and because they are usually pretty small, you can pack lots of containers onto a single computer.
-- TechCrunch

## THE BENEFITS OF CONTAINERS

- Fast deployment
- Cheap to run
- Quick to scale
- Automated
- Versioning
- Easily orchestrated

## CONTAINERS HAVE THEIR PROBLEMS

- Rapidly changing ecosystem
- Difficult security management
- Sprawl
- Broken dependencies
- Networking difficulties

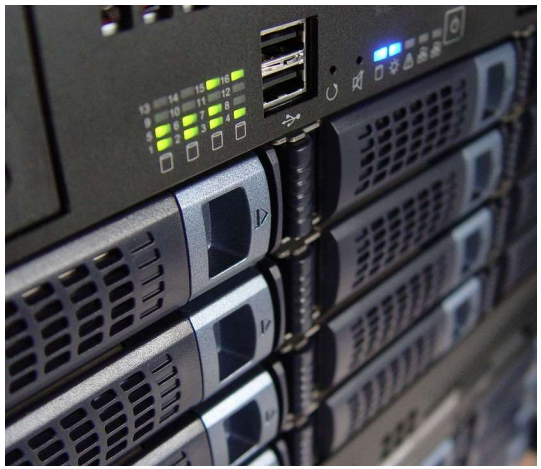
# 2.

## Serverless is the future

And for even better reasons

# Yes, there's a server!

But you don't manage it -- you just create code that will run on it.



## WHAT IS SERVERLESS COMPUTING

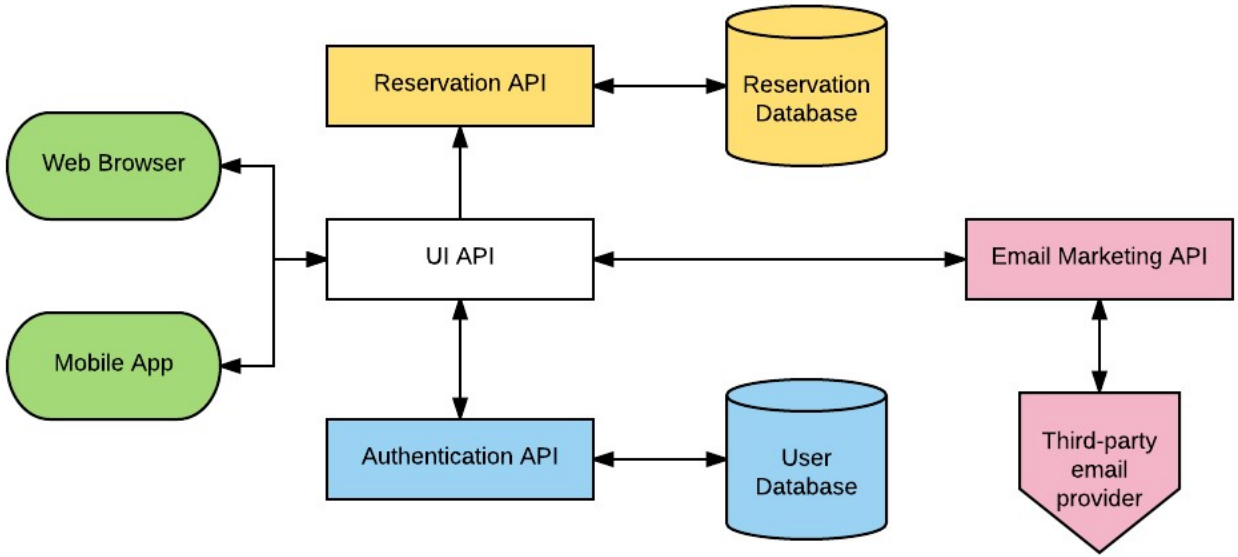
» Anonymous, generalized virtual machine instances
» Completely managed by the cloud provider
» Provisioned when you need them, deprovisioned when you're done
» Billed based on executions and resource consumption, not an hourly rate

## TRADITIONAL N-TIER WEB APP

## MICROSERVICES

- Break work into smaller steps
- Create APIs to handle each of these smaller steps

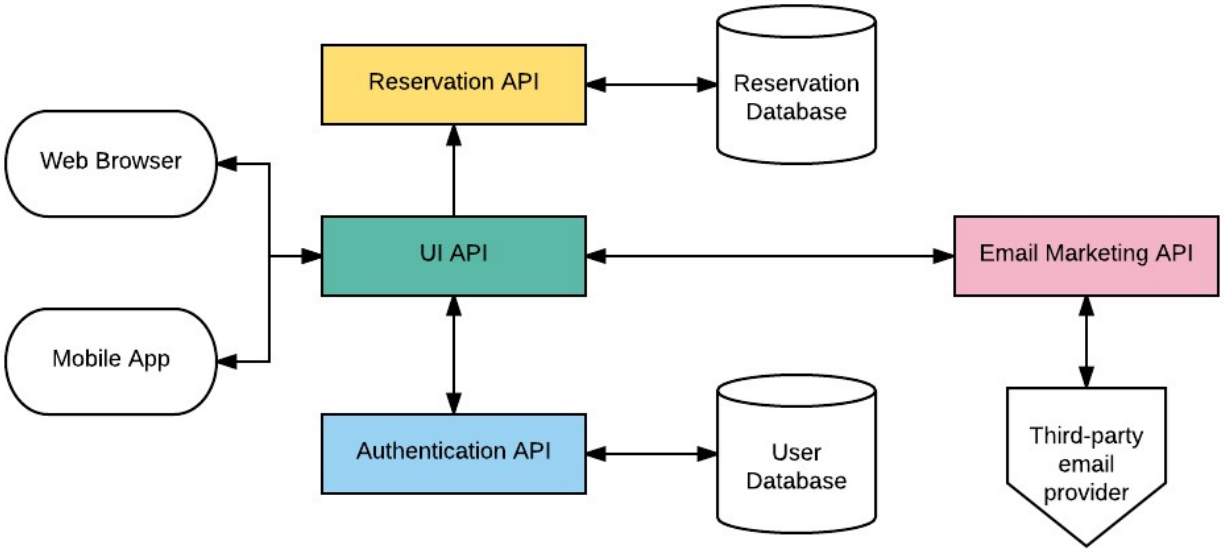
# MICROSERVICES ARCHITECTURE



## MICROSERVICE BENEFITS

» Manage functionality independently
» Streamlines development
» Makes code reusable among solutions
» Works best when it runs in small, virtualized environments like containers

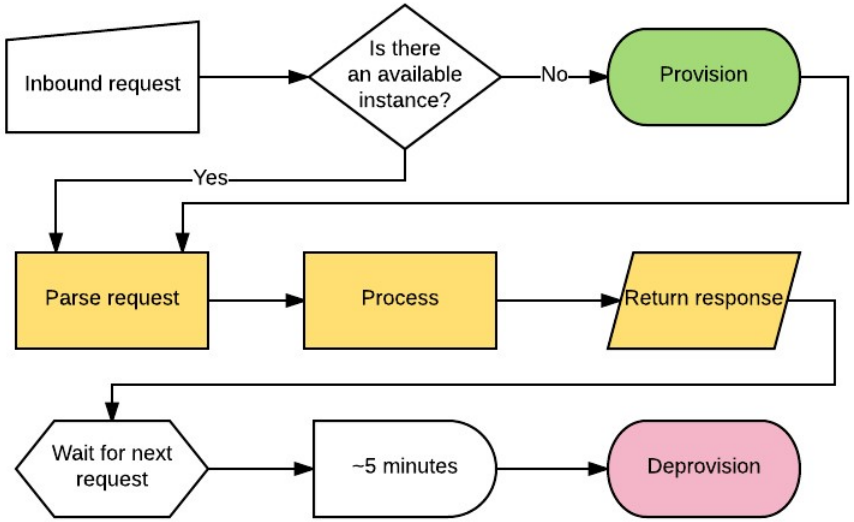
## CONTAINERS AND MICROSERVICES



## SERVERLESS IS MADE FOR MICROSERVICES

- Microservices != monoliths
- Focus on triggers, inputs and outputs
- Scale fast to demand
- Highly available

# FUNCTION WORKFLOW



## SERVERLESS FUNCTION FEATURES

» Base OS (Linux, Windows) with a generalized config
» Supports any code written in a given language: Node.js, Python, .NET Core, Java, etc.
» Provider can quickly provision these instances because they're all the same
» Instance started > code retrieved > code executed > instance deprovisioned

## PRICING DIFFERENCES

» VMs: Pay per CPU core, memory use, disk storage, software fees
» Containers: Also pay for VM use, but pack more work into the same VM
» Serverless: Pay for the resources you actually use

# 3.

## Serverless is best for TCO

Cheap to run, easy to manage

## VM vs FUNCTION PRICING

Assumption: 500,000 executions per month, 4 GB-seconds for each execution

| Azure VM D2v2 | AWS VM t2.large | Azure Function | AWS Lambda |
|---|---|---|---|
| **$104.16** | **$69.94** | **$25.60** | **$26.67** |

## VM vs FUNCTION PRICING

Assumption: 2 million executions per month, 4 GB-seconds for each execution

| Azure VM A4m v2 | AWS VM t2.2xlarge | Azure Function | AWS Lambda |
|---|---|---|---|
| **$220.97** | **$279.74** | **$121.80** | **$129.86** |

## VM vs FUNCTION PRICING

Assumption: 20,000 executions per month, 512 MB-seconds for each execution

| Azure VM A1v2 | AWS VM t2.small | Azure Function | AWS Lambda |
|---|---|---|---|
| **$31.99** | **$17.12** | **FREE** | **FREE** |

## THE LONG TAIL OF SERVERLESS

» Everything the same = dirt cheap to provide
» Each new instance is effectively profitable
» Only a small number of users need to exceed the free threshold periodically to turn a major profit
» Long-tail pricing model

## SERVERLESS VS VMS/CONTAINERS

» Similar workloads cost less on functions
» You don't pay for unused capacity
» No more zombies!

# Not quite NoOps, but close

Drastically reduce lead times and staffing requirements

## NOOPS

» Automation, abstraction and cloud vendor services eliminate several DevOps tasks (and positions)
» ~~Sprint develop, build, test and deploy~~
» Focus is shifted to rapid development
» Continuous integration / deployment

## SERVERLESS IS HIGHLY AVAILABLE AND SCALABLE

- Bad code downtime limited by microservices
- Functions scale automatically and quickly
- High availability is built in
- Regional outage is the only real threat

## SERVERLESS BENEFITS: RECAP

- Lower real infrastructure costs
- Easier SDLC via modular workloads/microservices
- No servers to manage
- Faster deployment via CI/CD/automation
- HA/DR built in
- Usual cloud-based business continuity strategies

# 4.

## Sold on serverless

They believe!


“
With AWS Lambda, we eliminate the need to worry about operations. We just write code, deploy it, and it scales infinitely; no one really has to deal with infrastructure management. The size of our team is half of what is normally needed to build and operate a site of this scale.
-- Tyler Love, CTO, Bustle

*In 5 years, every modern business will have a substantial portion of their systems running the cloud. But that's only the first step. The next step comes when you free your developers from the tedious work of configuring and deploying even virtual cloud-based servers.*

-- Greg DeMichillie, Head of Developer Platform and Infrastructure, Adobe

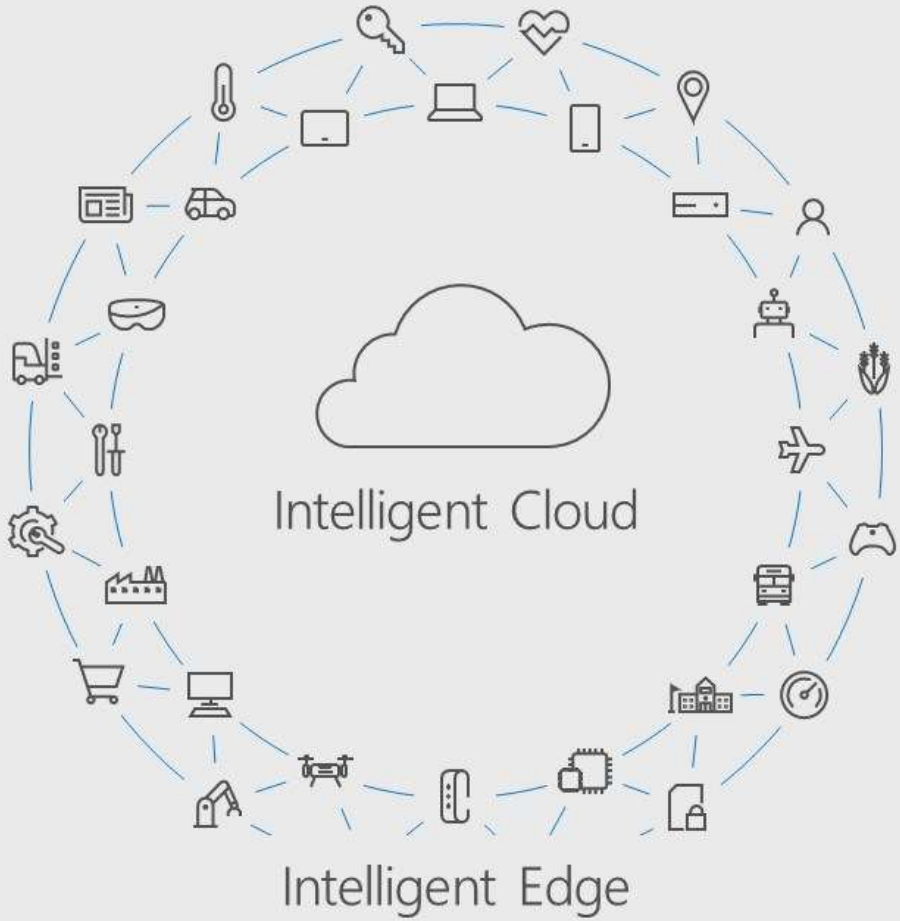
## WORKFLOWS FOR THE MASSES

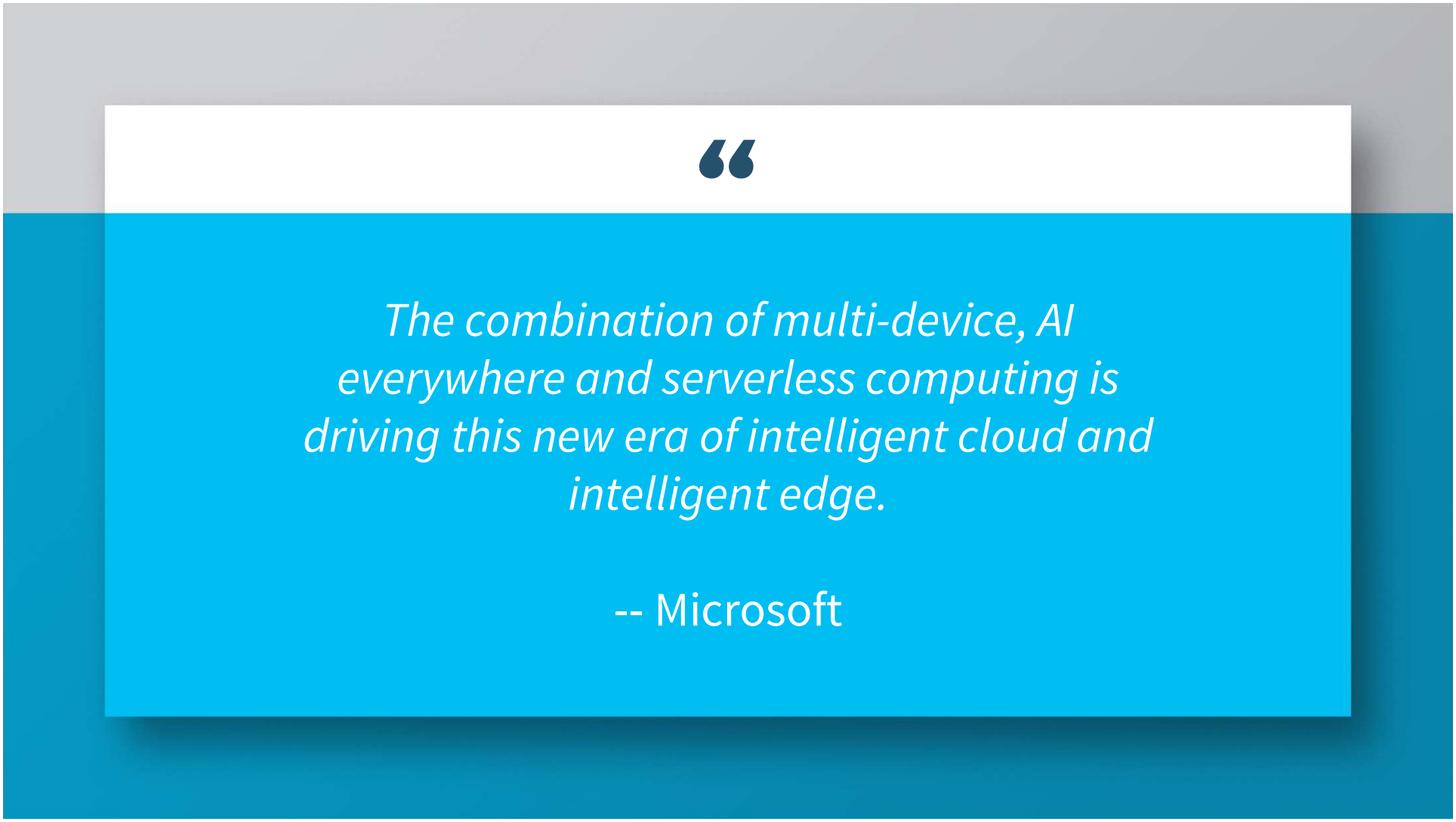
- What if everyone could program?
- Microservices are the building blocks of workflows
- AI/big data are already tacking semantics
- Orchestrate your vision, yourself

Multi-device

Artificial Intelligence

Serverless




“
The combination of multi-device, AI everywhere and serverless computing is driving this new era of intelligent cloud and intelligent edge.
-- Microsoft

# 5.

## Serverless isn't for everything

A wholesale change with big up-front costs

## BIG UP-FRONT COSTS

» Microservices mean rebuilding workloads
» Huge up-front costs
» Requires revisiting existing partnerships
» N-tier ports well to containers

## NOT FOR EVERYTHING

» Small, non-scaling workloads
» Solutions that depend on the environment/many services
» Massive, constant computing power requirements

## WEAKNESSES OF SERVERLESS

- Laggy startups for cold code
- Lag/drops in microservice communication
- Immature technology
- Somewhat wedded to the vendor
- Restrictions in code you can run
- Somewhat limited library access
- Event-input-output model might not work

## SUMMARY

- Serverless is the next wave in cloud computing
- Huge time and cost savings, low TCO
- Significant benefits to cloud vendors
- Built-in HA/DR, business continuity is simple
- Fast deployment and sensible architecture
- But it's not for every workload

THANKS!

# Any questions?

You can find me at:

- @dougvdotcom
- doug@linuxacademy.com
- linkedin.com/in/dougvdotcom

## CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash and Pixabay