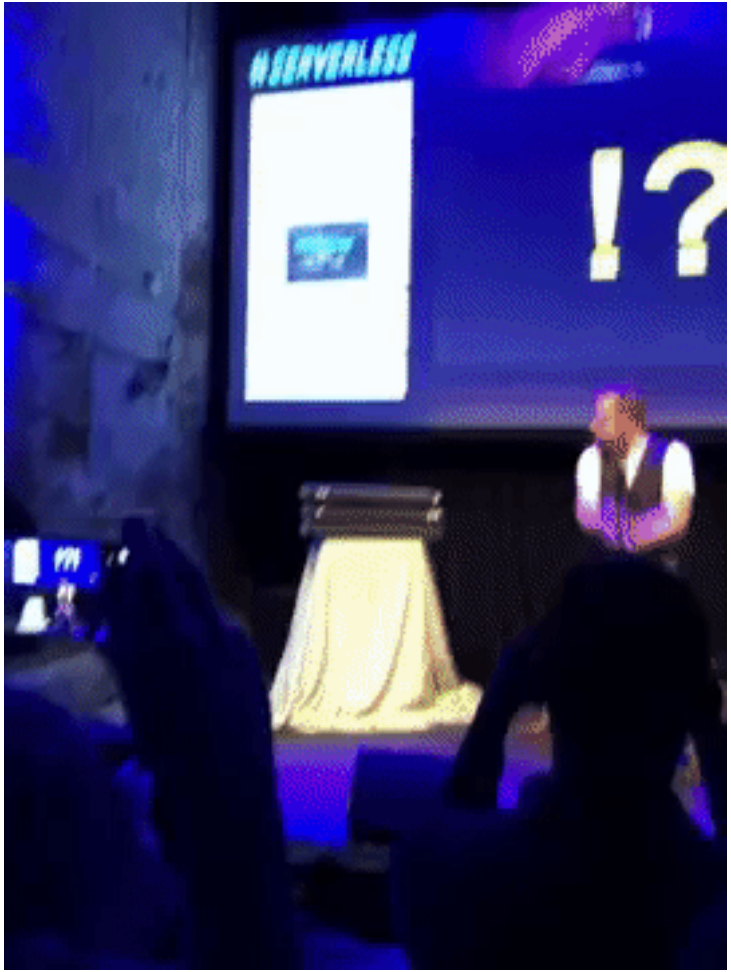
The Truth Behind Serverless

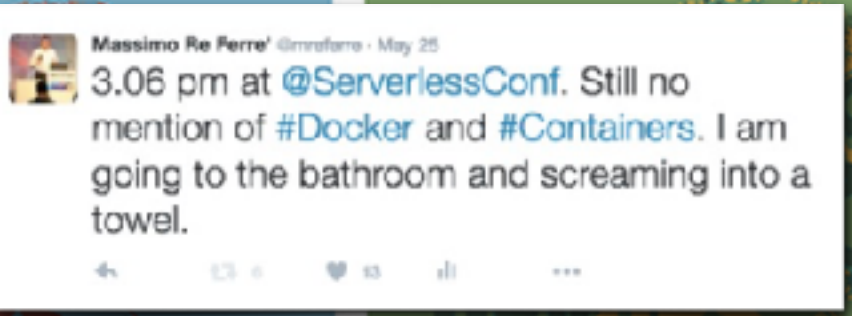
Erica Windisch

CTO & Co-founder, IOpipe

IO|


#SERVERLESS
!?


inspiring the future of cloud
SERVERLESSCONF


Massimo Re Ferre' @mreferre · May 25
3.06 pm at @ServerlessConf. Still no mention of #Docker and #Containers. I am going to the bathroom and screaming into a towel.


SERVERLESSCONF

How to draw an owl

1.

2.

1. Draw some circles

2. Draw the rest of the [redacted] owl

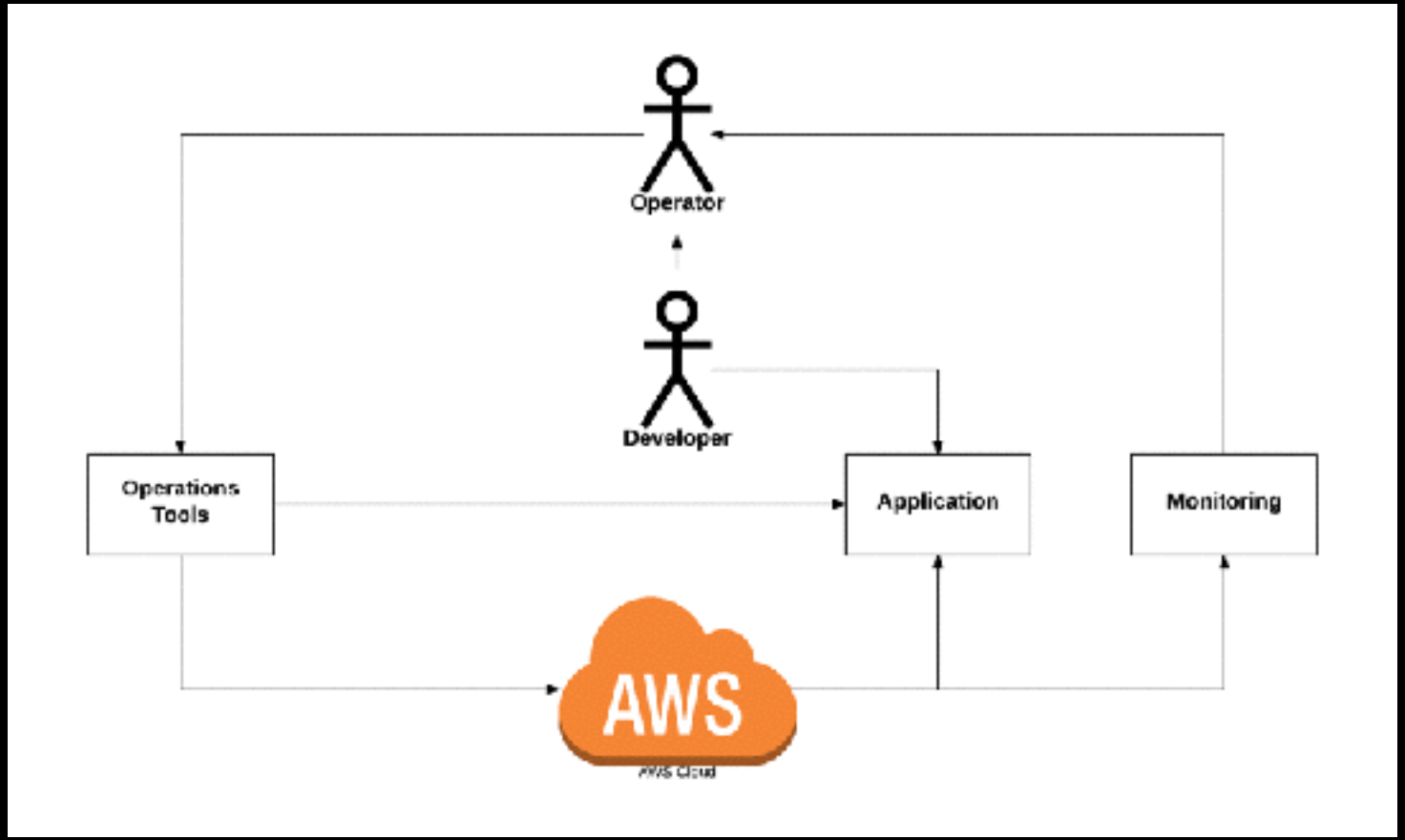
Serverless Culture

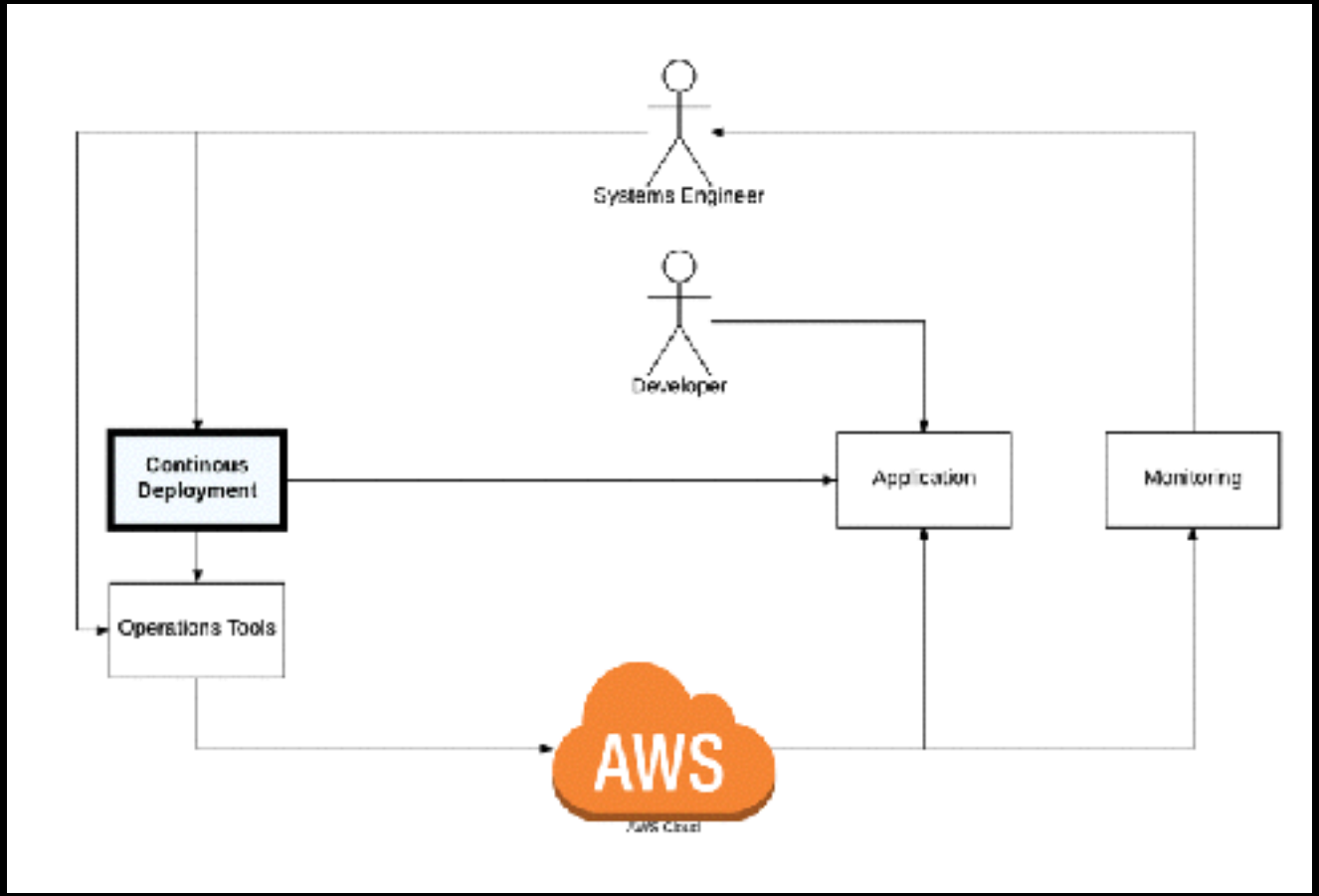
Don't build, when you can buy.

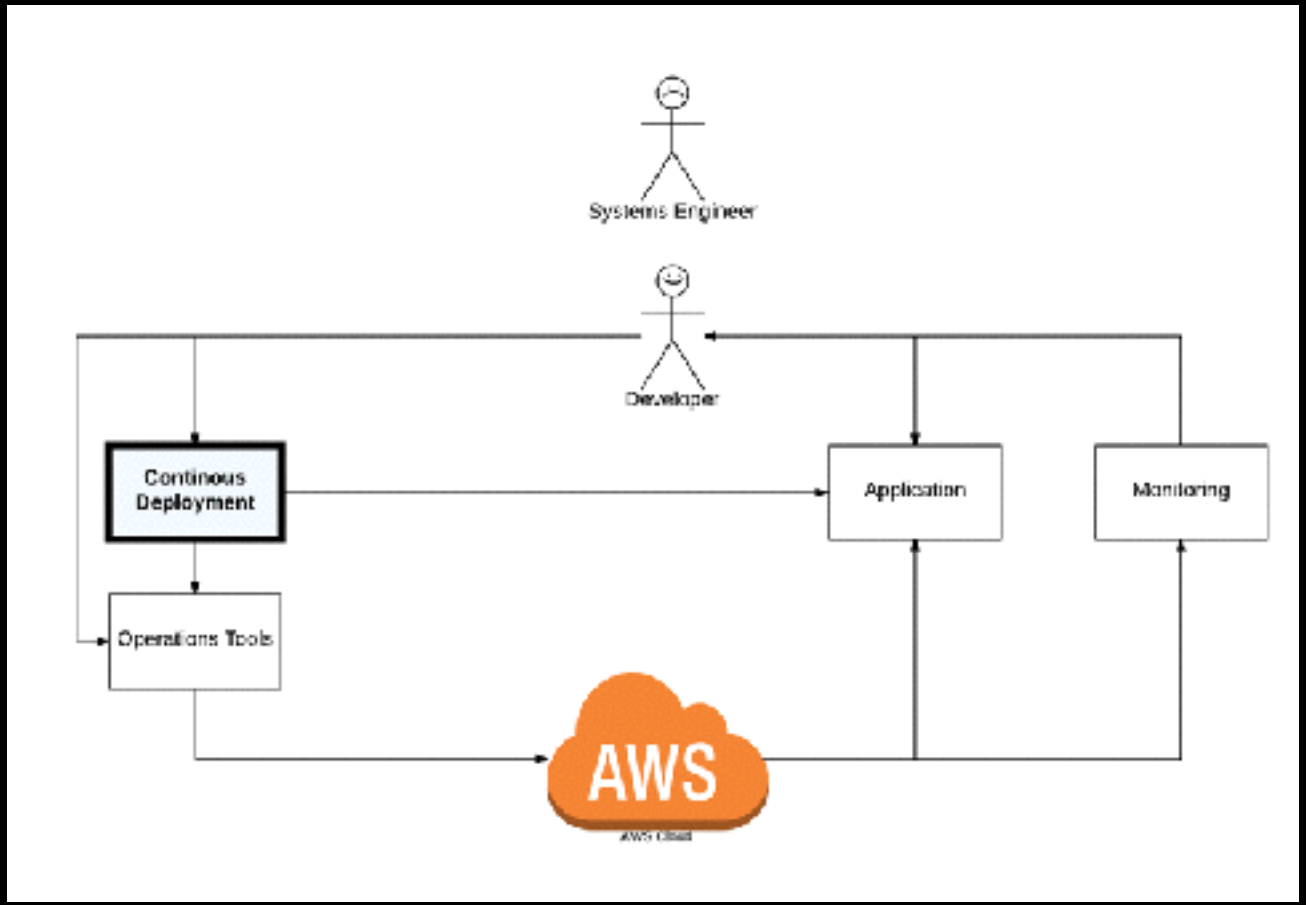
Build as little as necessary.

Build with the minimum operational complexity.

Maybe all you need are some circles


Operator
Developer
Operations Tools
Application
Monitoring
AWS
AWS Cloud


Systems Engineer
Developer
Continous Deployment
Application
Monitoring
Operations Tools
AWS
AWS Cloud


Systems Engineer
Developer
Continous Deployment
Application
Monitoring
Operations Tools
AWS

The 12 Factor App

I. Codebase One codebase tracked in revision control, many deploys

II. Dependencies Explicitly declare and isolate dependencies

III. Config Store config in the environment

IV. Backing services Treat backing services as attached resources

V. Build, release, run Strictly separate build and run stages

VI. Processes Execute the app as one or more stateless processes

VII. Port binding Export services via port binding

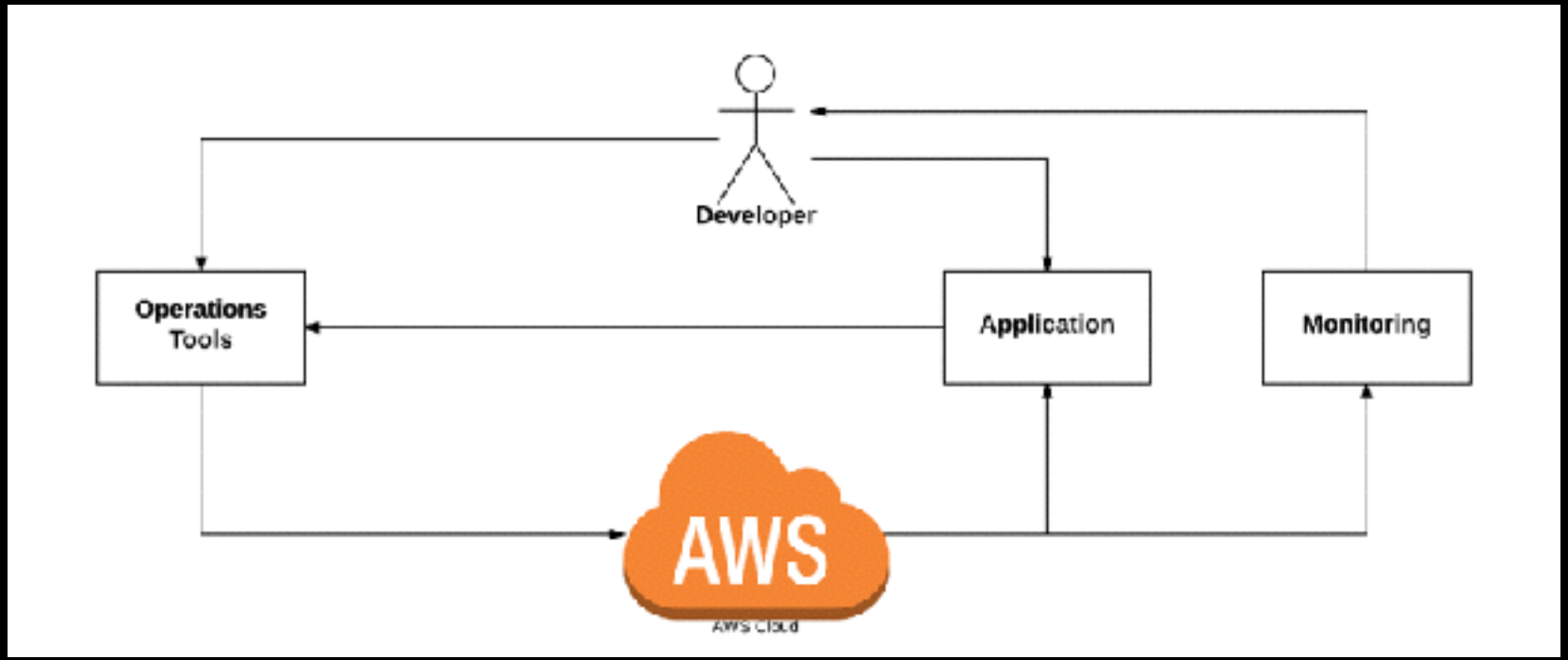
VIII. Concurrency Scale out via the process model

IX. Disposability Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity Keep development, staging, and production as similar as possible

XI. Logs Treat logs as event streams

XII. Admin processes Run admin/management tasks as one-off processes


Developer
Operations
Tools
Application
Monitoring
AWS
AWS Cloud

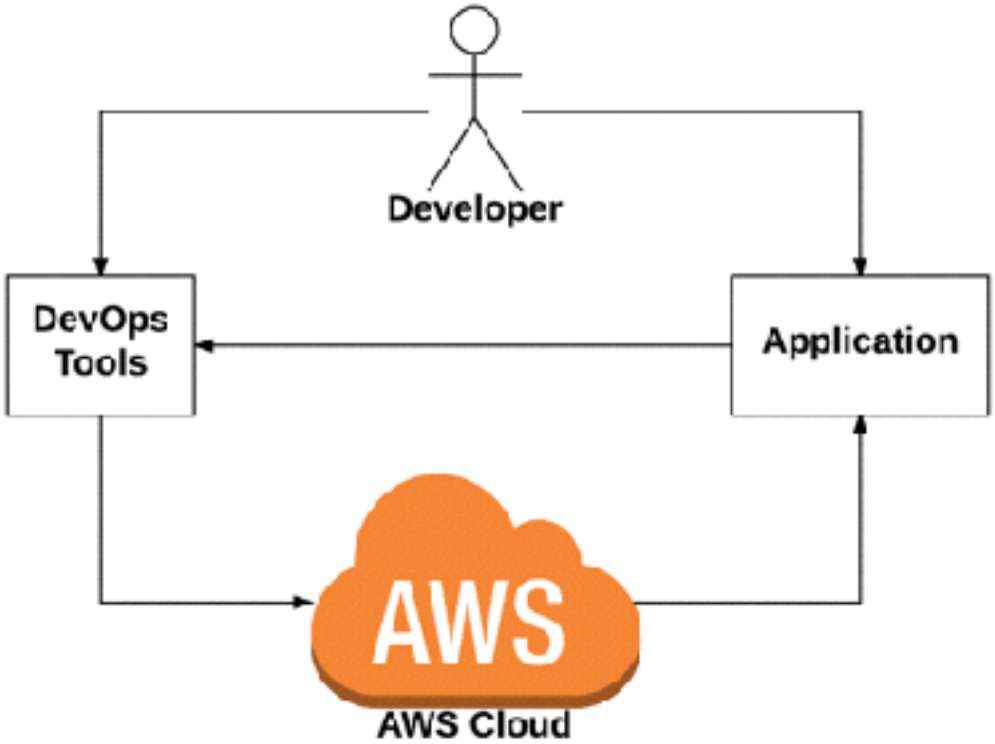
```
$ docker run --rm -it debian
…
# insmod veth.ko

insmod: ERROR: could not insert module
veth.ko: Operation not permitted

# whoami
root
```

Amazon's "Serverless (functions) Manifesto"

- Function are the unit of deployment and scaling.
- No machines, VMs, or containers visible in the programming model.
- Permanent storage lives elsewhere.
- Scales per request; Users cannot over- or under-provision capacity.
- Never pay for idle (no cold servers/containers or their costs).
- Implicitly fault-tolerant because functions can run anywhere.
- BYOC - Bring Your Own Code.
- Metrics and logging are a universal right.


Developer
DevOps Tools
Application
AWS
AWS Cloud

75 GB storage limit for all functions

5 minute maximum duration

~20ms for “hello world”

128MB to 3GB memory

1000 containers max (by default)

Functions lazy-loaded from s3

Containers + processes long-lived,
automatically scaled and garbage-collected.

4.5 minutes to 4.5 hours

non-root user

/ read-only

/tmp 512MB max

function mounted to /var/task

Pauses / unpauses containers between events

Pay only when processing an event.

V = 1
lim (1 + 1/n)^n = e
) = 1 − 1
cosα =

Serverless is...

event-driven

distributed

scalable

It's all about the events.

Events are always immutable.

Event processing can and should be immutable.

Event processing should be fast & low-latency.

Events generate events.

(it's turtles all the way down)

A function is... a nanoservice

Many nanoservices compose
a microservice.

a single microservice
could be written in many languages.

(implemented in multiple nanoservices)

IO|

www.iopipe.com

Erica Windisch
CTO & Founder IOpipe, Inc.
@ewindisch