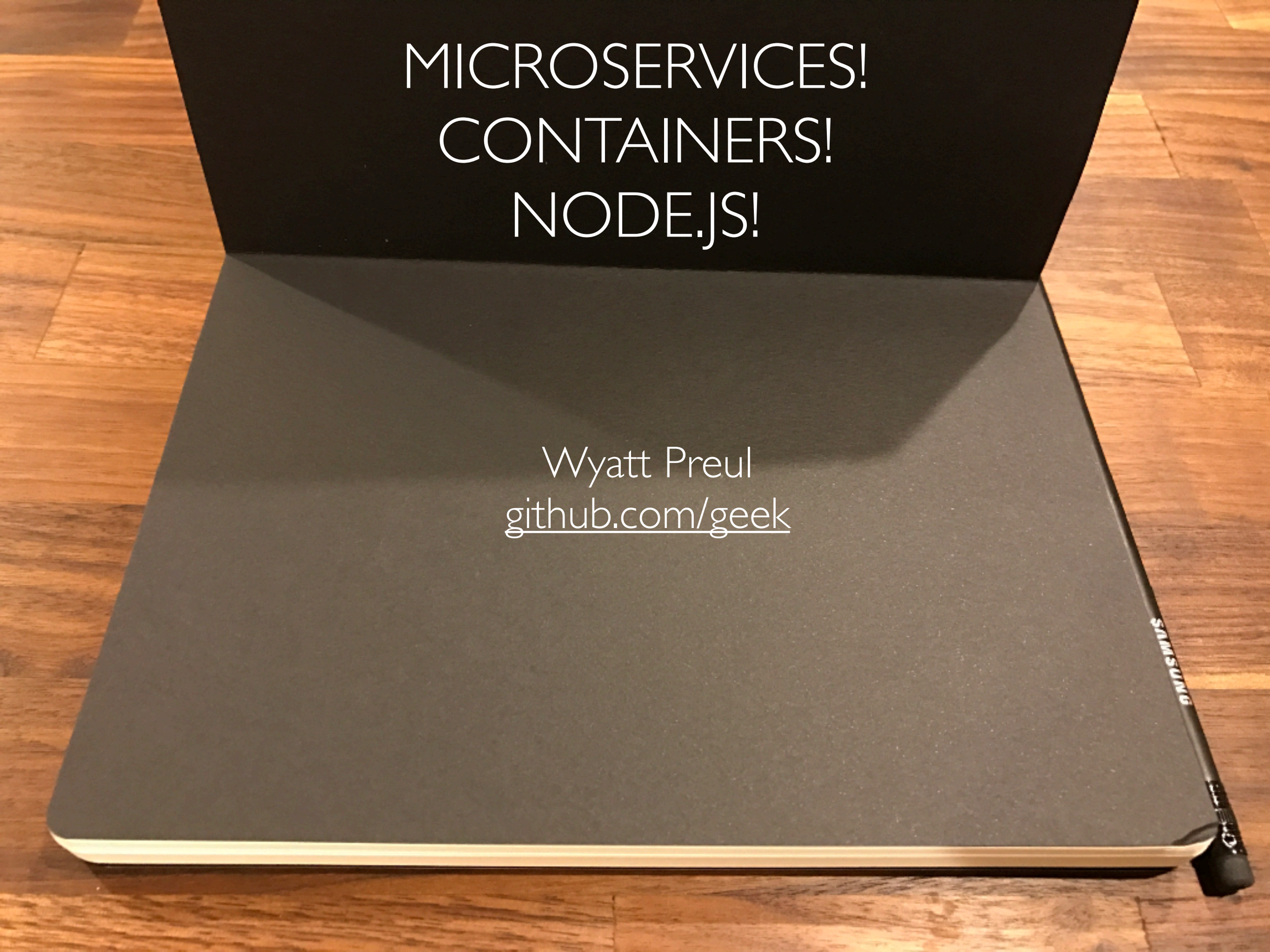
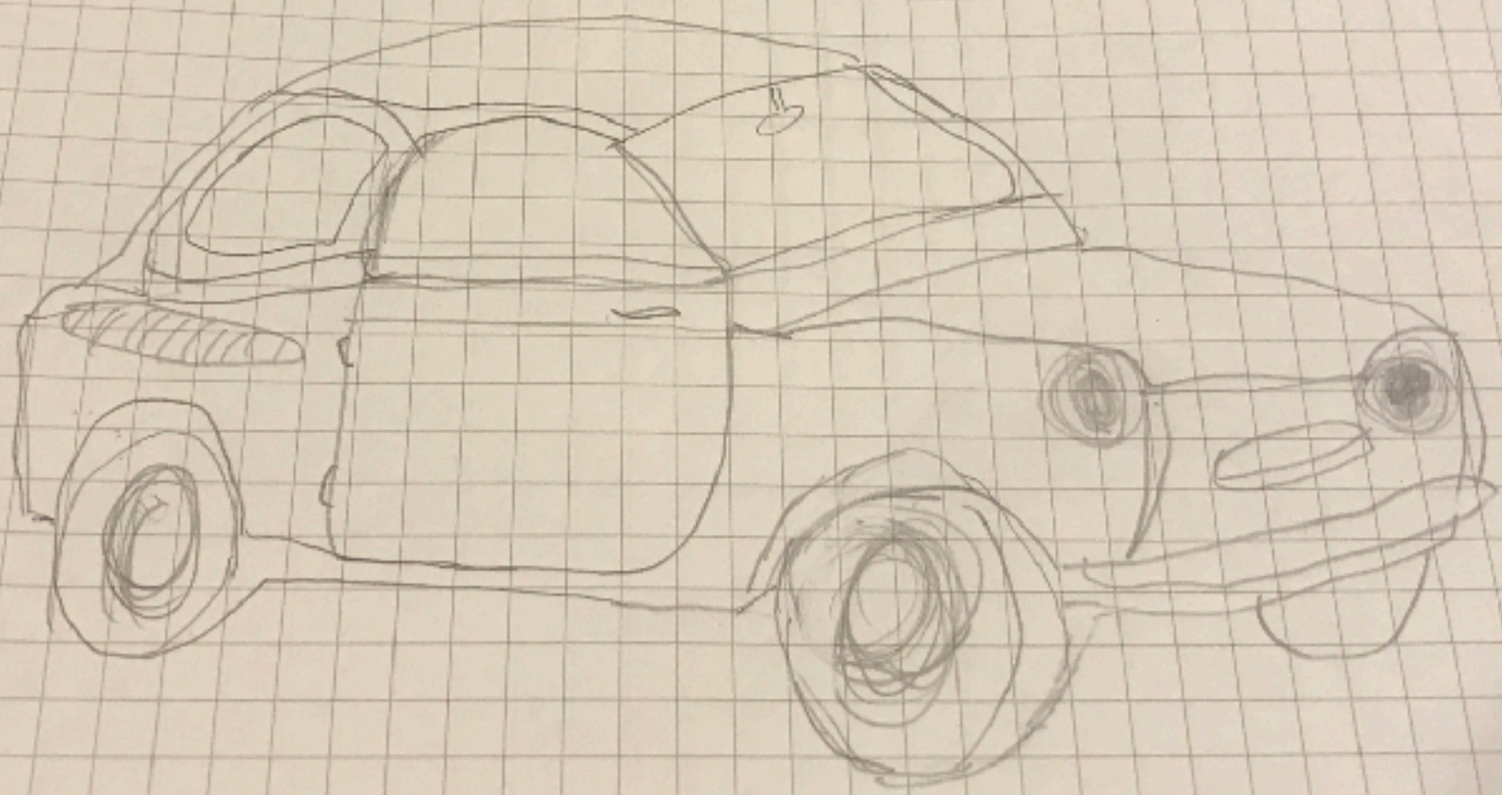
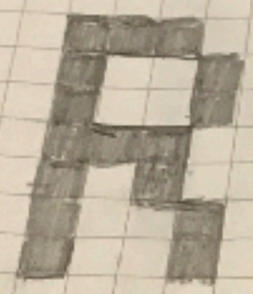

MICROSERVICES!
CONTAINERS!
NODE.JS!
Wyatt Preul
github.com/geek
SAMSUNG


MEMES ARE OUT
Move Your Karma Now
PRESENT WITH DRAWINGS



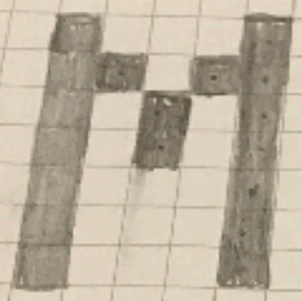
Failure

- familiar territory
- bug free isn't possible
- accustomed to disposable code



Reduce Fail

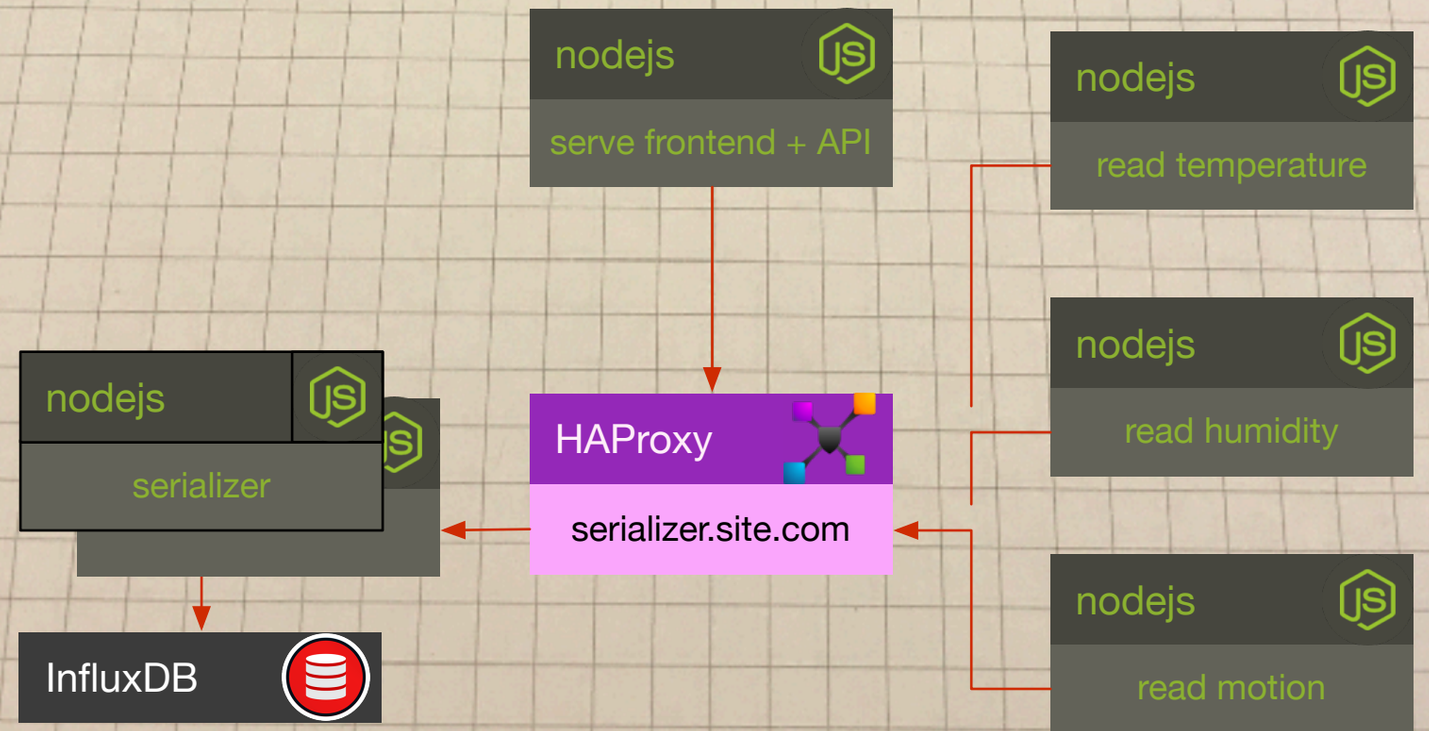
- break assumptions into functions
- separate functional units into services
- we recently name these microservices



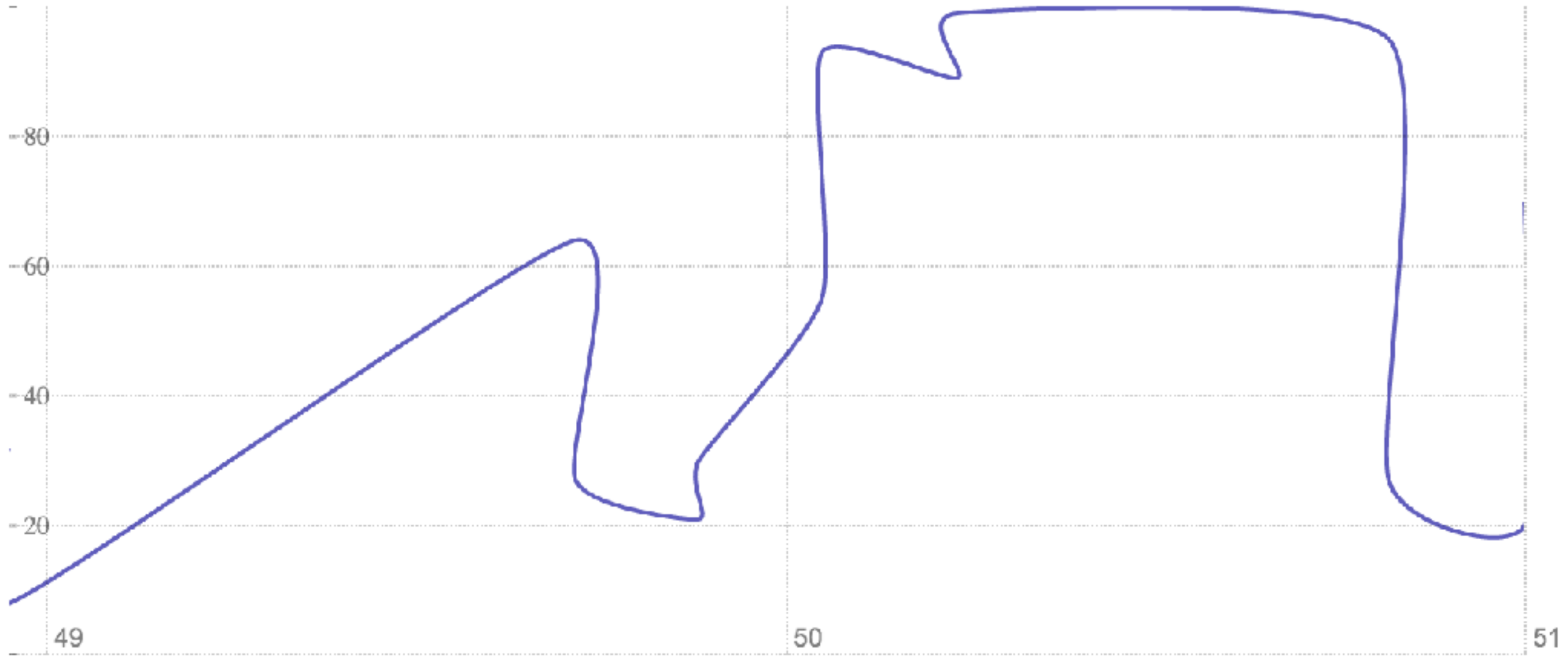
icroservice

- ought to embrace failure
- works in spite of outside failures
- can be disposed of

Microservice



Temperature Sensor



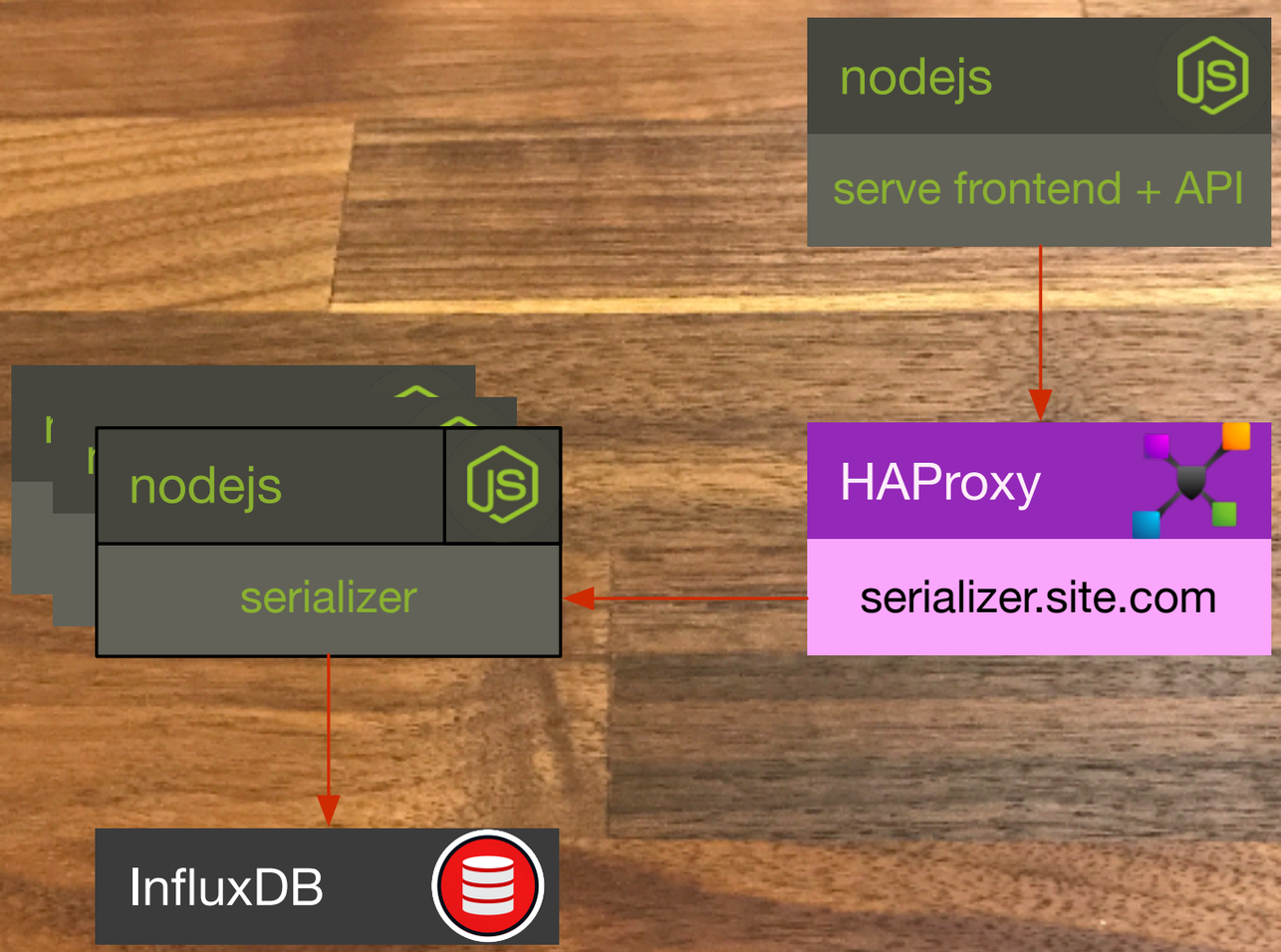
```
smartthings.act({
  role: 'smartthings',
  cmd: 'read',
  type: 'temperature',
  ago: 5
}, (err, data) => {

  serializer.act({
    role: 'serialize',
    cmd: 'write',
    type: 'temperature',
    value: point.value
  }, next);
});
```

```
const serializer = Seneca();
serializer.client({
  host: process.env.SERIALIZER_HOST,
  port: process.env.SERIALIZER_PORT
});
```

Failure

- retry the connection
- if host configuration changes, must reload configuration, maybe even restart temperature service
- temperature service is unaware of health of its dependencies


nodejs
serve frontend + API
nodejs
serializer
HAProxy
serializer.site.com
InfluxDB

Load Balance

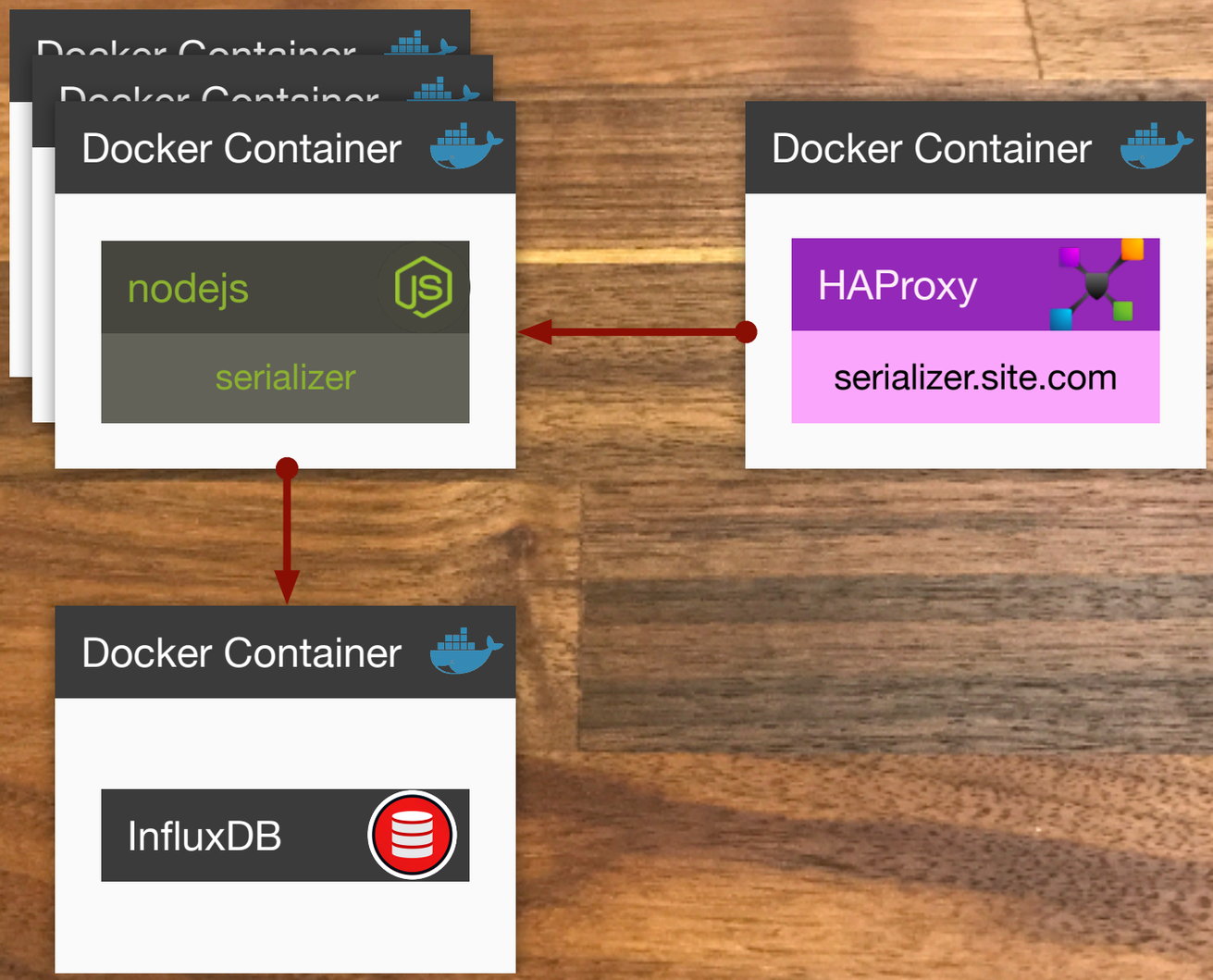
- subdomains setup for environment (qa, stg, prod)
- mistakes will happen, not uncommon for a prod service to point to a QA service, oops
- with lots of microservices and hosts, misconfiguration is likely more common

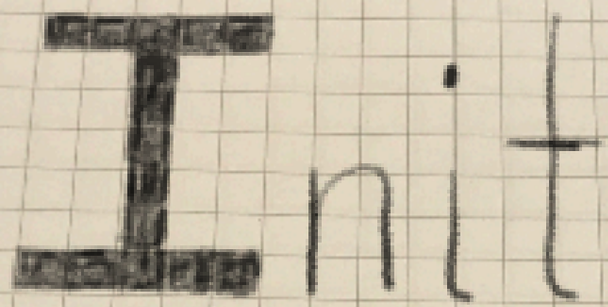
Health

- indicate issue with service, or at least an issue between the load balancer and the service
- often exposed as a path (/health)
- sometimes they do full checks, db connection, memory usage

Docker

- OS level virtualization, more performant than VM
- consistent environments, immutable
- operations that developers can do, speeding up delivery of software


Docker Container
Docker Container
Docker Container
nodejs
serializer
Docker Container
HAProxy
serializer.site.com
Docker Container
InfluxDB



- bring your own init (BYOI)

- container inits exist: tini, dumb-init, my_init

Lifecycle

- need setup and teardown hooks in container
- perform initialization before starting microservice
- perform cleanup (finish writes) before container is killed

TRITON

ContainerPilot

- addresses previous problems
- self-aware, self-operating containers, supports consul or etcd for discovery
- portable, works anywhere docker does
- open-source, free: joyent/containerpilot


Workshop Build
Docker Container
ContainerPilot
Consul
Docker Container
ContainerPilot
nodejs
frontend
Docker Container
ContainerPilot
nodejs
serializer
Docker Container
ContainerPilot
nodejs
read temperature
Docker Container
ContainerPilot
nodejs
serializer
Docker Container
ContainerPilot
nodejs
read humidity
Docker Container
InfluxDB
Docker Container
ContainerPilot
nodejs
read motion
CMS

```
"consul": "localhost:8500",
"services": [
  {
    "name": "frontend",
    "port": {{.PORT}},
    "health": "/usr/bin/curl -o /dev/null --fail -s http://localhost:{{.PORT}}/heartbeat",
    "poll": 3,
    "ttl": 10
  }
],
```

```
"services": [
  {
    "name": "mongodb-replicaset",
    "port": 27017,
    "health": "python /usr/local/bin/manage.py health",
    "poll": 5,
    "ttl": 25
  }
],
```

```
"coprocesses": [
  {
    "command": ["/usr/local/bin/consul", "agent",
                "-data-dir=/data",
                "-config-dir=/config",
                "-rejoin",
                "-retry-join", "{{ .CONSUL_HOST }}",
                "-retry-max", "10",
                "-retry-interval", "10s"],
    "restarts": "unlimited"
  }
],
```

```
"backends": [
  {
    "name": "serializer",
    "poll": 3,
    "onChange": "pkill -SIGHUP node"
  }
]
```

```
FROM node:6.9.1-alpine

# Install Consul
# curl && unzip package to /usr/local/bin

# Install ContainerPilot
# curl and untar to /bin

# Copy ContainerPilot configuration

# Install our application

CMD ["/bin/containerpilot", "node", "/opt/app/"]
```

```
$ docker-compose up -d
```

```
$ docker-compose scale serializer=2
```

consul	3 passing
frontend	2 passing
humidity	2 passing
motion	2 passing
serializer	4 passing
temperature	2 passing

```
const serializer = Piloted('serializer');
if (!serializer) {
  return;
}

const seneca = Seneca();
seneca.client({
  host: serializer.address,
  port: serializer.port
});
```


"telemetry": {
"port": 9090,
"tags": ["op"],
"sensors": [
{
"namespace": "containerpilot",
"subsystem": "frontend",
"name": "free_memory",
"help": "Frontend Free Memory",
"type": "counter",
"poll": 5,
"check": ["/bin/memory.sh"],
"timeout": "5s"
}
]
}

Prometheus Alerts Graph Status Help

Expression (press Shift+Enter for newlines)

Execute ✓ - insert metric at cursor -

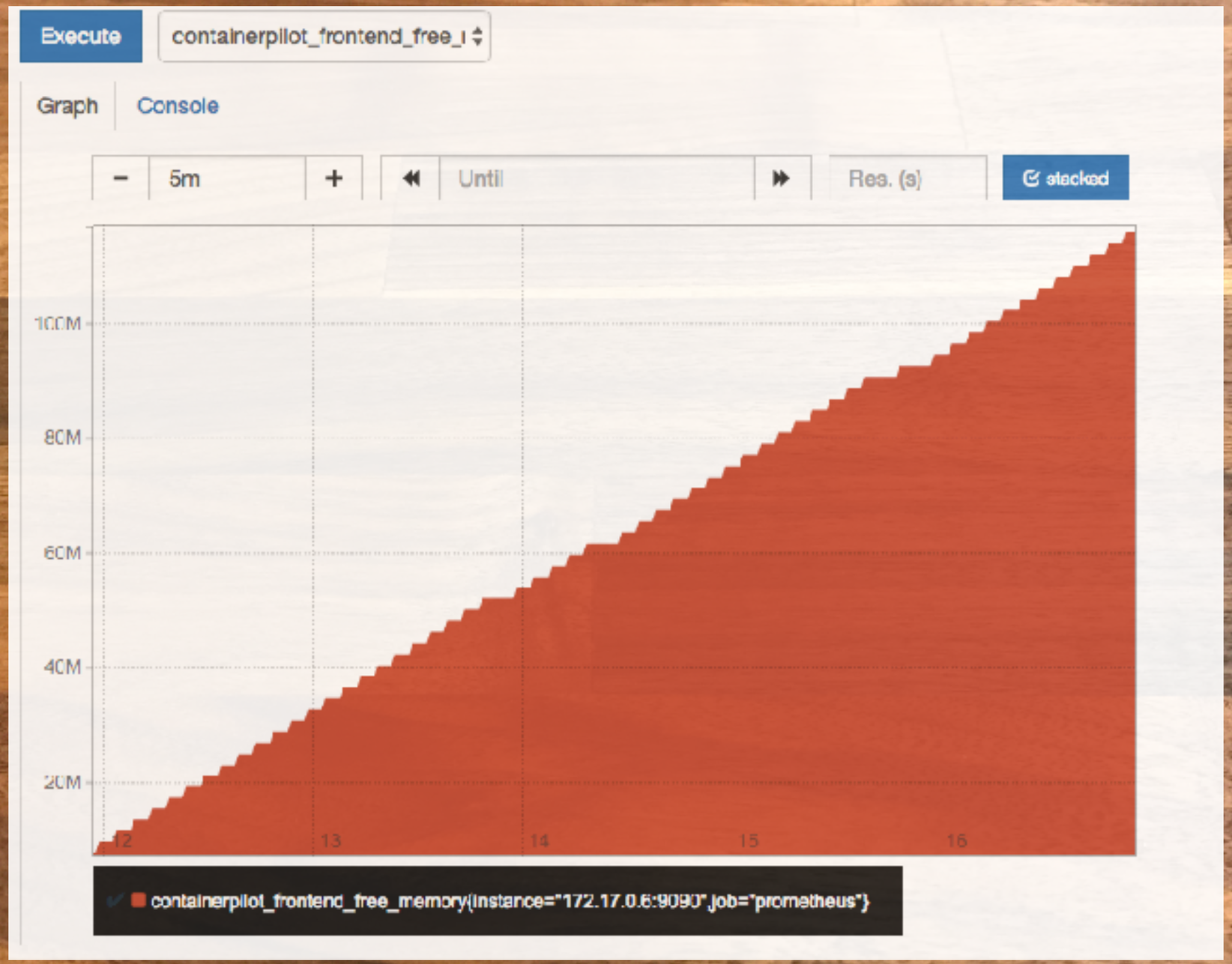
containerpilot_frontend_free_memory

go_gc_duration_seconds

go_gc_duration_seconds_count

go_gc_duration_seconds_sum

Graph


Execute
containerpilot_frontend_free_
Graph
Console
5m
Until
Res. (s)
stacked
10.0M
8.0M
6.0M
4.0M
2.0M
12
13
14
15
16
containerpilot_frontend_free_memory{instance="172.17.0.6:9090",job="prometheus"}

Execute | containerpilot_frontend_free_r

Graph | Console

- | 10s | + | « | Until | » | Res. (s) | stacked

Production

- only requires pointing to prod consul cluster

- setup database volume for persistence

Triton

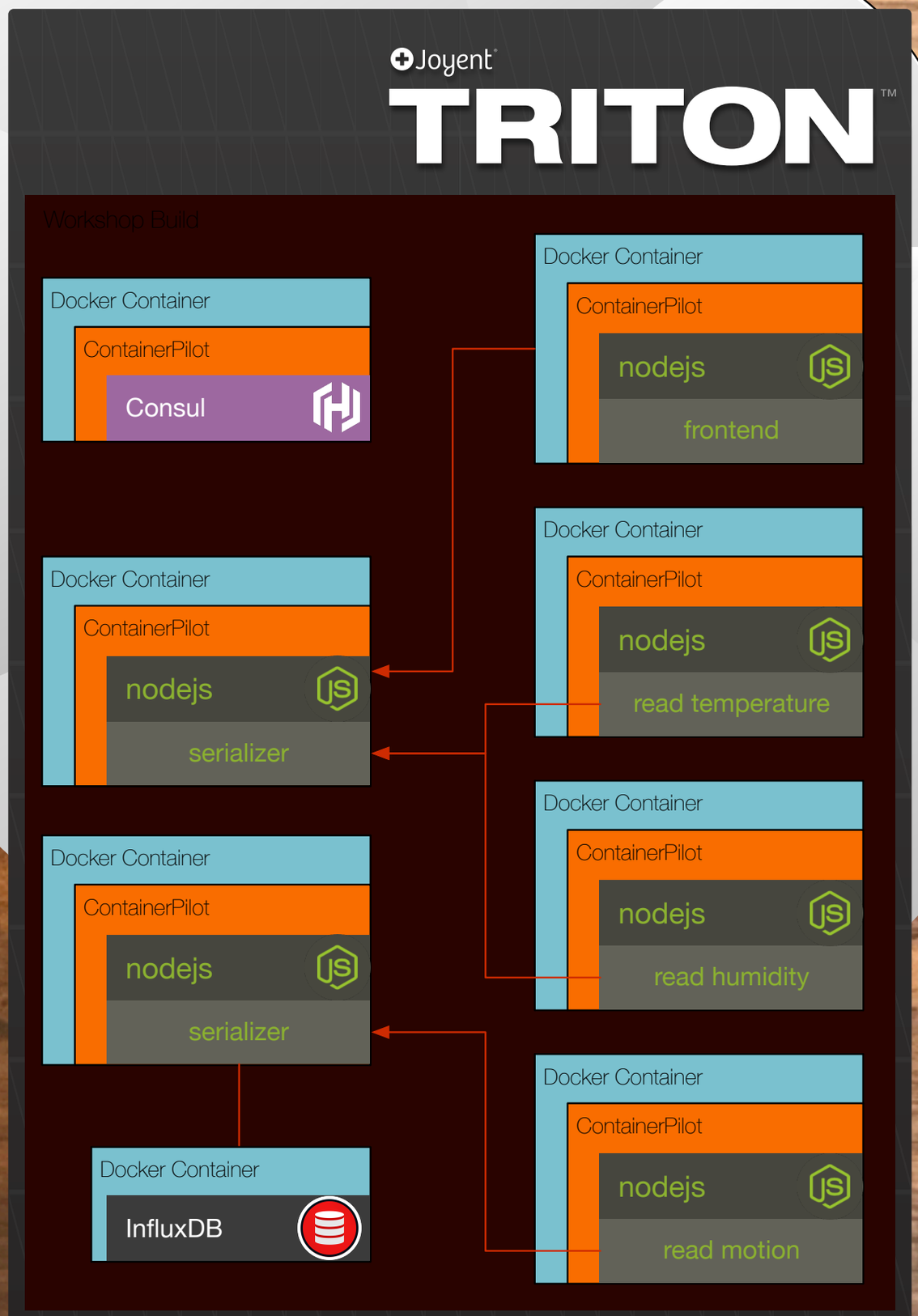
- the datacenter is the docker host
- docker container runs on the metal!
- get the debugging tooling from SmartOS!!!

```
$ eval $(triton env)
```

```
triton (master)* $ docker-compose up -d
Creating triton_temperature_1
Creating triton_frontend_1
Creating triton_influx_1
Creating triton_motion_1
Creating triton_humidity_1
```

```
triton (master)* $ docker-compose ps
         Name                        Command                State
-------------------------------------------------------------------------
triton_consul_1          /usr/local/bin/containerpi ...   Up      53/tcp, 53/udp
triton_frontend_1        /bin/containerpilot node / ...   Up      0.0.0.0:10001-
triton_humidity_1        /bin/containerpilot node / ...   Up
triton_influx_1          /run.sh                          Up      0.0.0.0:8083->
triton_motion_1          /bin/containerpilot node / ...   Up
triton_serializer_1      /bin/containerpilot node / ...   Up
triton_temperature_1     /bin/containerpilot node / ...   Up
```

```
triton (master)* $ triton ip triton_frontend_1
64.30.132.211
```


Joyent
TRITON
Workshop Build
Docker Container
ContainerPilot
Consul
Docker Container
ContainerPilot
nodejs
frontend
Docker Container
ContainerPilot
nodejs
serializer
Docker Container
ContainerPilot
nodejs
read temperature
Docker Container
ContainerPilot
nodejs
serializer
Docker Container
ContainerPilot
nodejs
read humidity
Docker Container
InfluxDB
Docker Container
ContainerPilot
nodejs
read motion

Thank

y'all

jsgeek.com/links