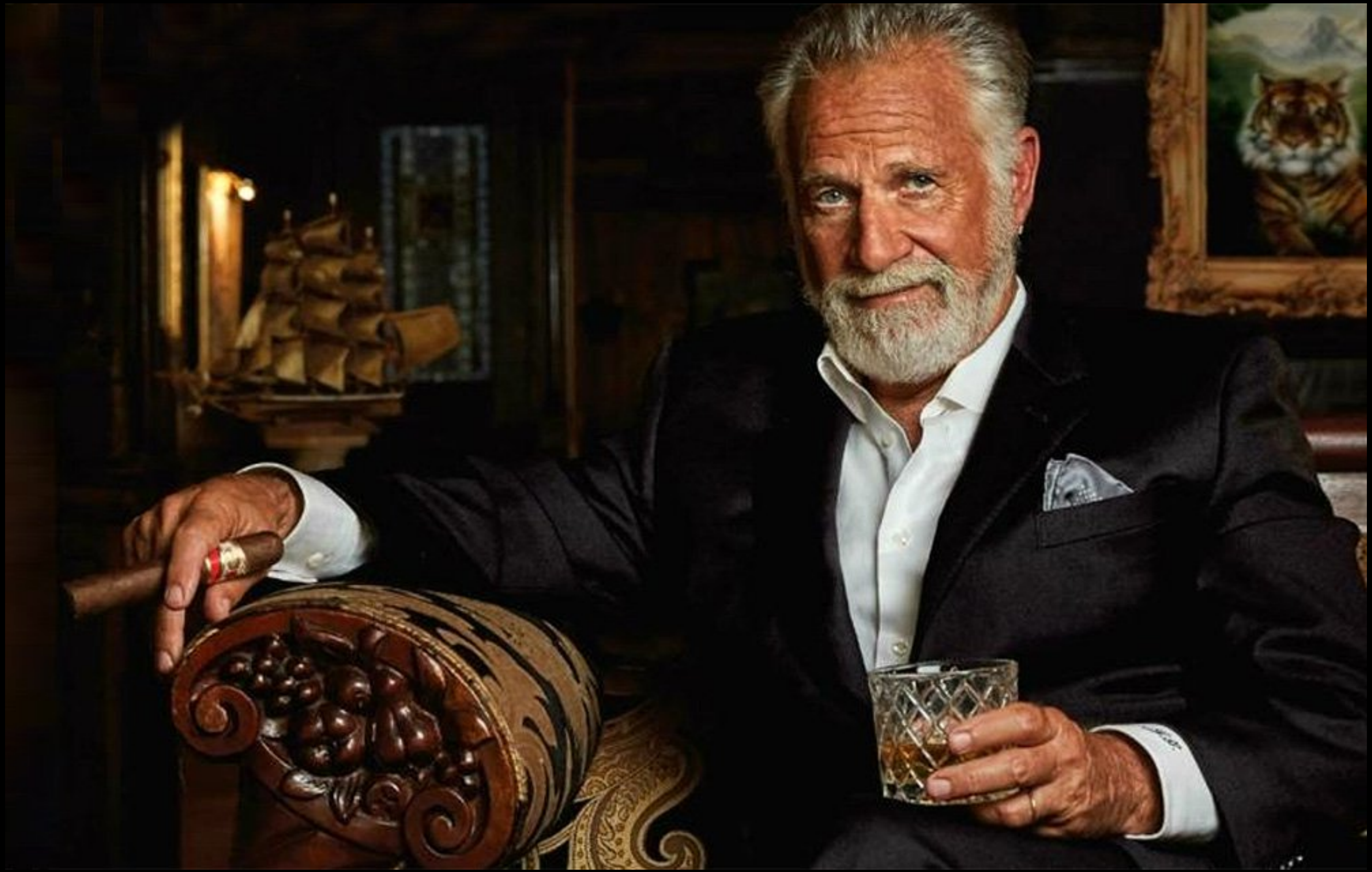
DEPLOYMENTS!

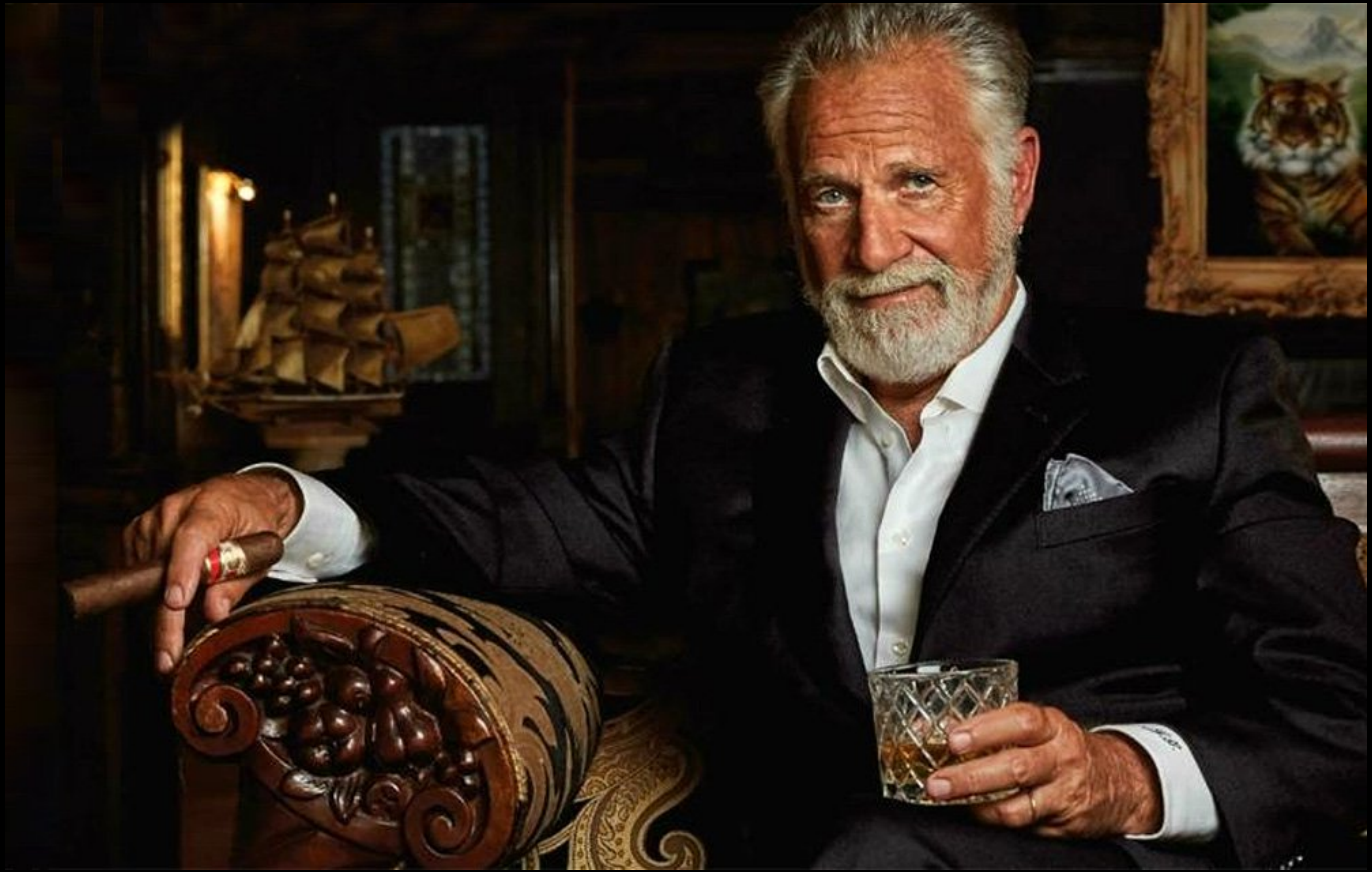
WHAT COULD POSSIBLY GO WRONG?

A LOT!

DUH.



When I _____________ … I do it in production.



When I **deploy to prod**… I do it in production.

CANARY DEPLOYS

WITH KUBERNETES AND ISTIO

JASON YEE

Technical Evangelist
Nomad & Travel Hacker
Whiskey Hunter
Pokemon Trainer

Tw: @gitbisect
Em: jyee@datadoghq.com

DATADOG

TW: @datadoghq
SaaS-based monitoring, tracing & logging

Trillions of points/day

We're hiring:
jobs.datadoghq.com

Note: We're running some production services on Kubernetes & have been implementing Istio.

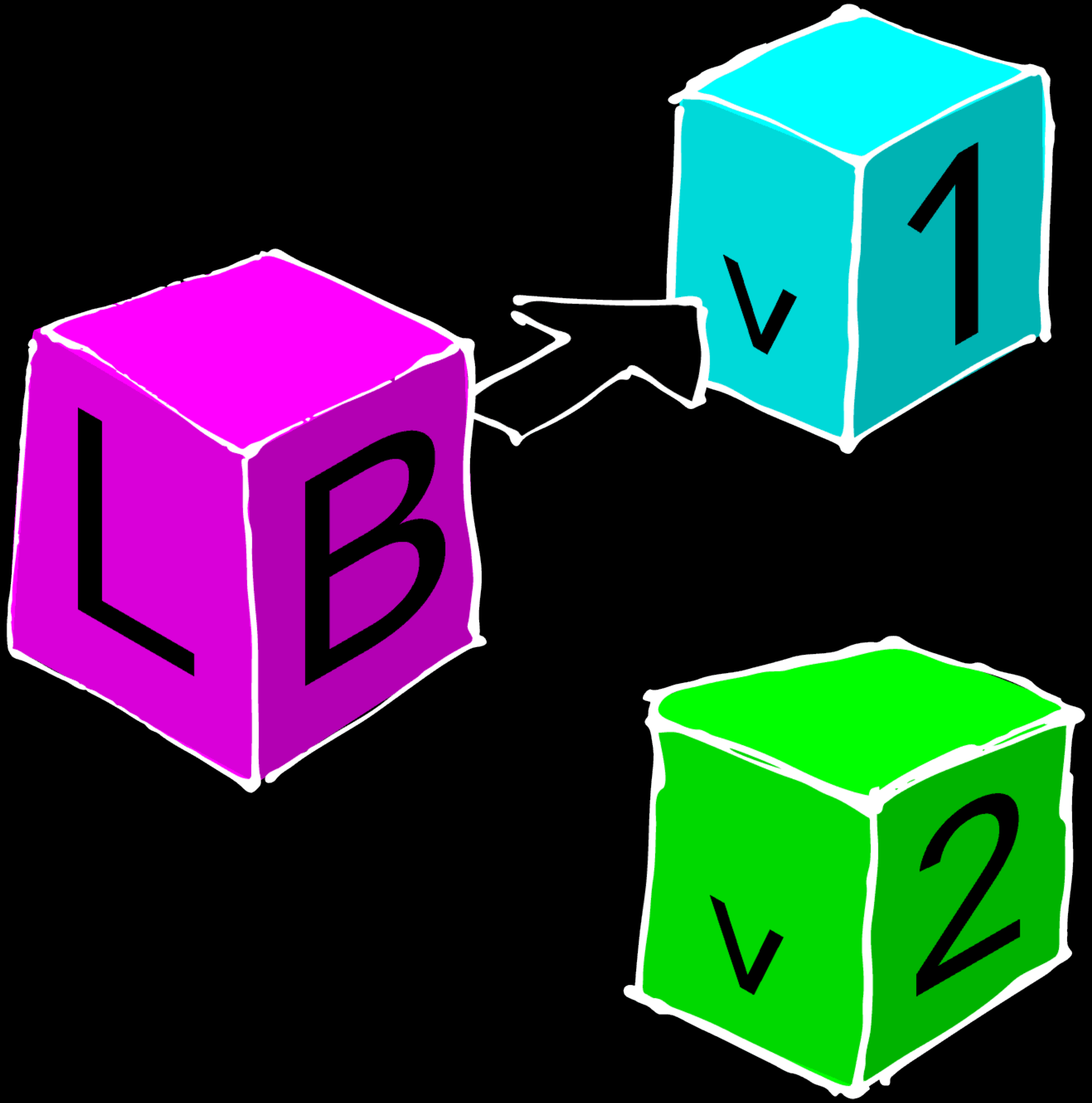
"ISTIO IS A VERY EARLY PROJECT. DON'T RUN OUT OF HERE AND DEPLOY IT IN PRODUCTION; YOU'LL BE ON THE NEWS."

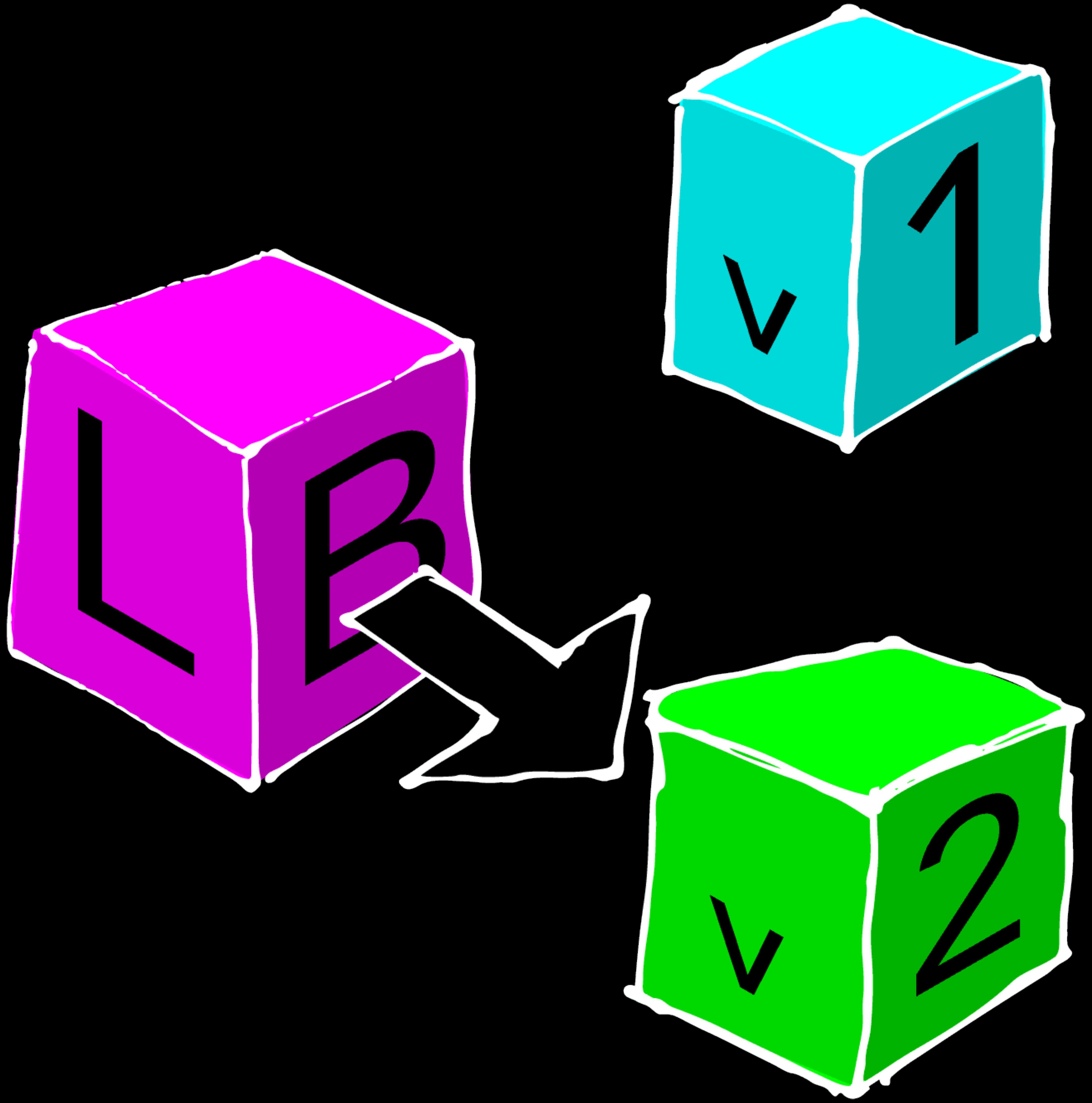
- KELSEY HIGHTOWER (FEB 22, 2018)

"ISTIO IS 1.0 AND APIS ARE STABLE.-ish GO TRY IT OUT!"

- **NOT** KELSEY HIGHTOWER (OCT 3, 2018)

BLUE-GREEN DEPLOYMENTS


LB
v1
v2


LB
v1
v2

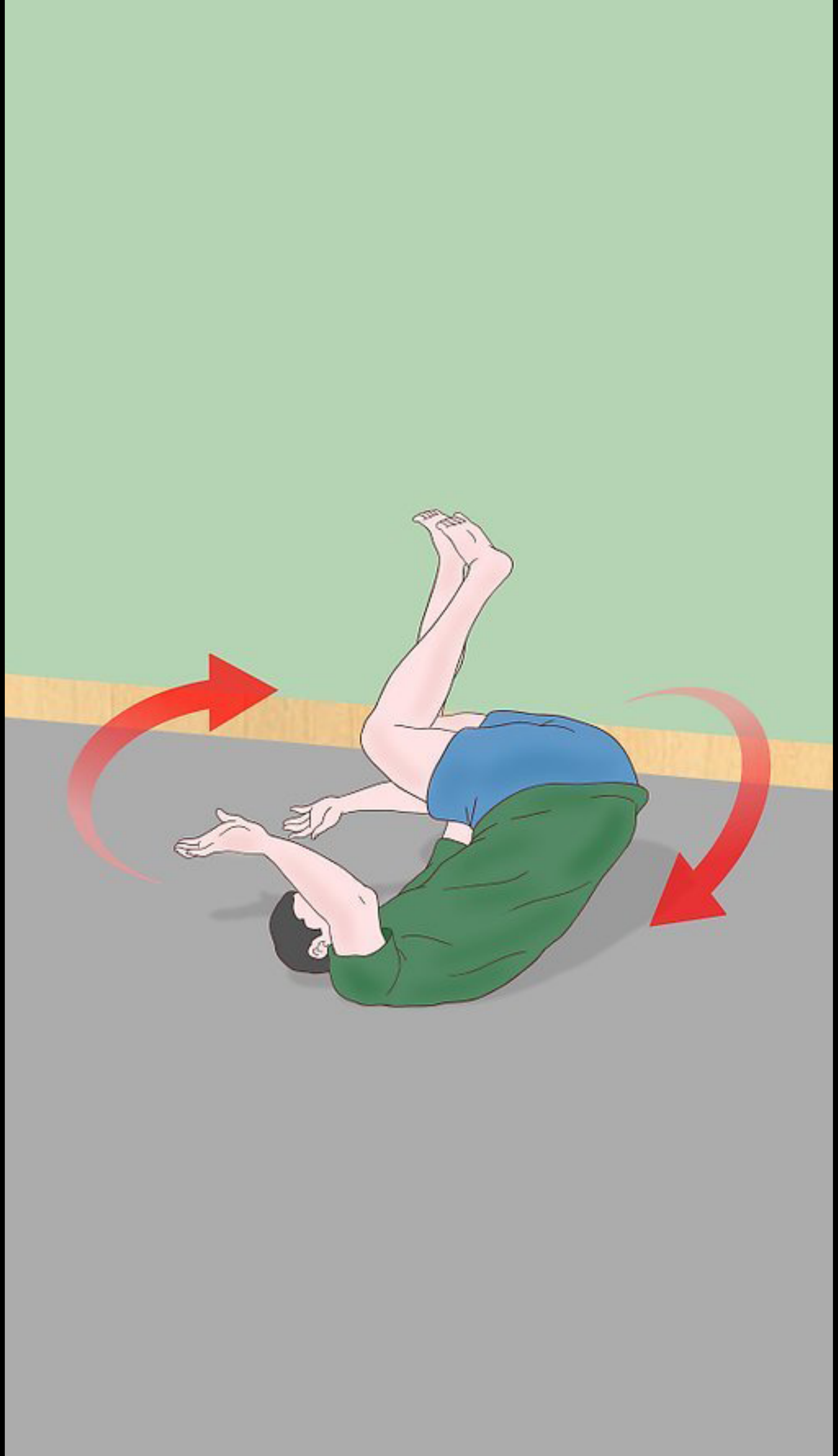
BLUE-GREEN DEPLOYMENTS

- Pros:
  - Zero-downtime deploys
  - Easy rollbacks

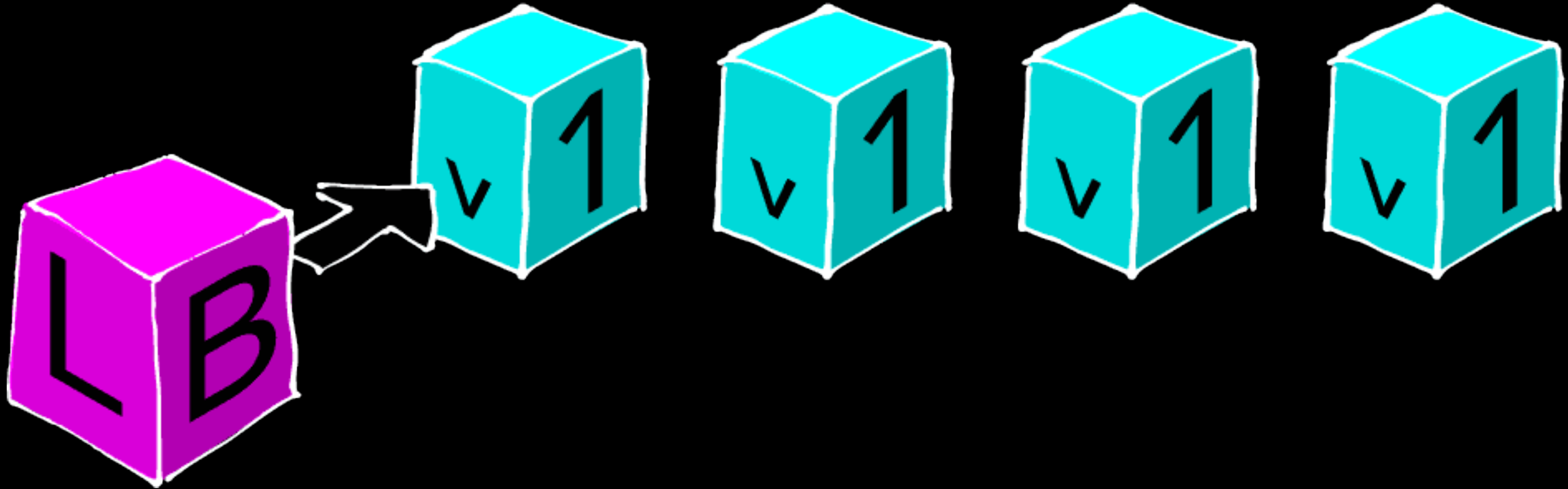
CTRL+Z

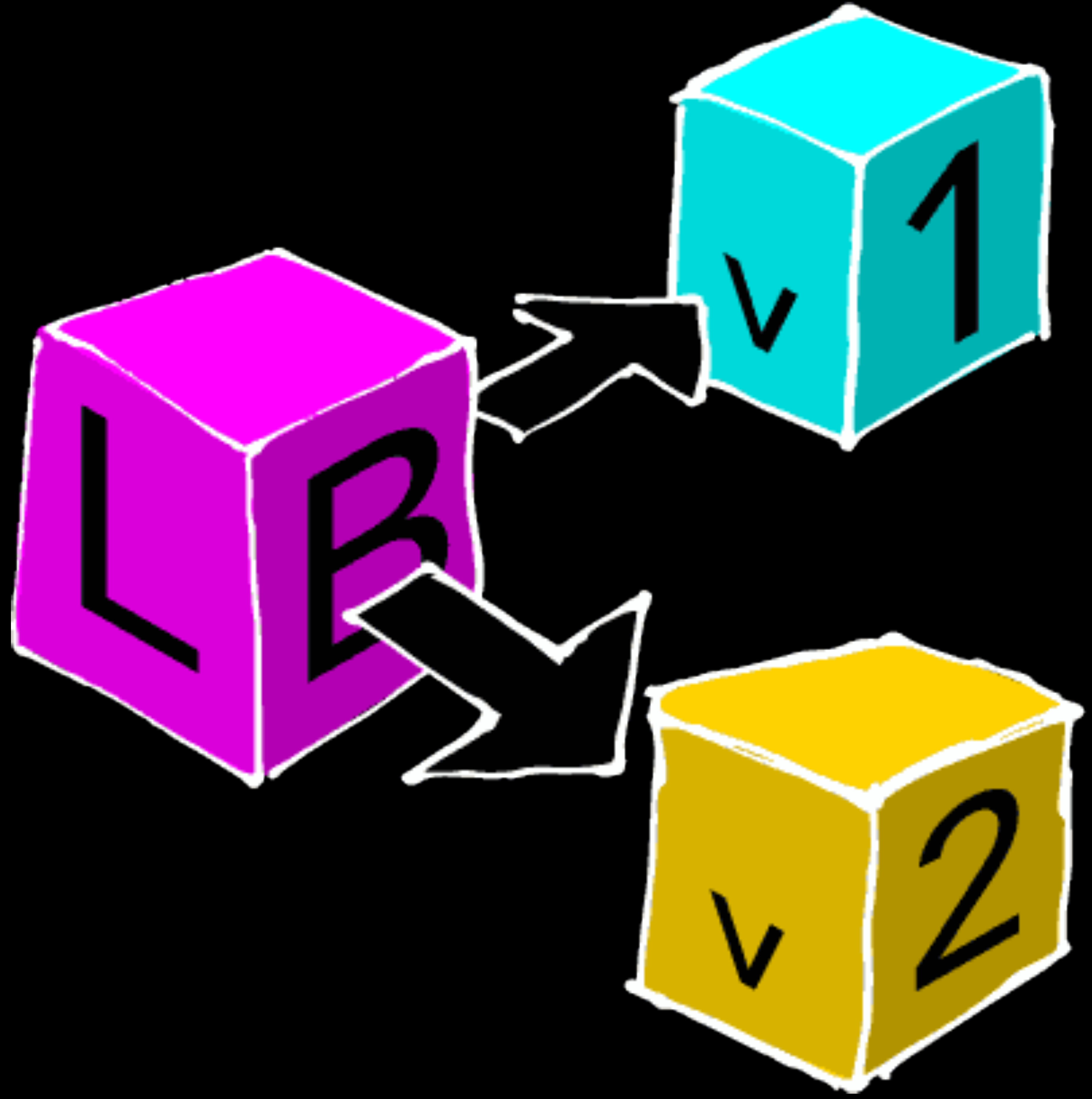
BLUE-GREEN DEPLOYMENTS

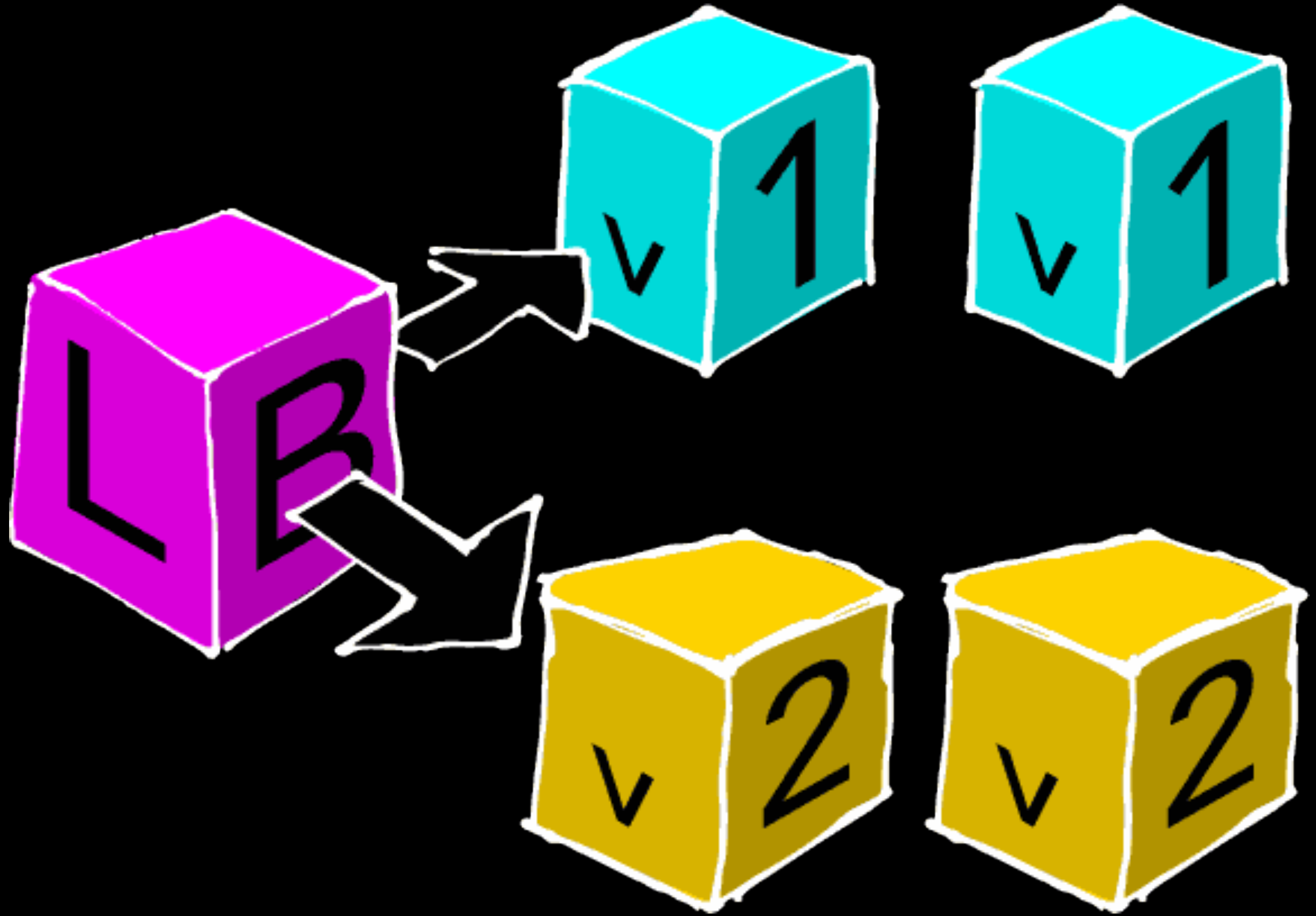
- Cons:
  - ✌️Easy✌️ rollbacks… 🤞😬

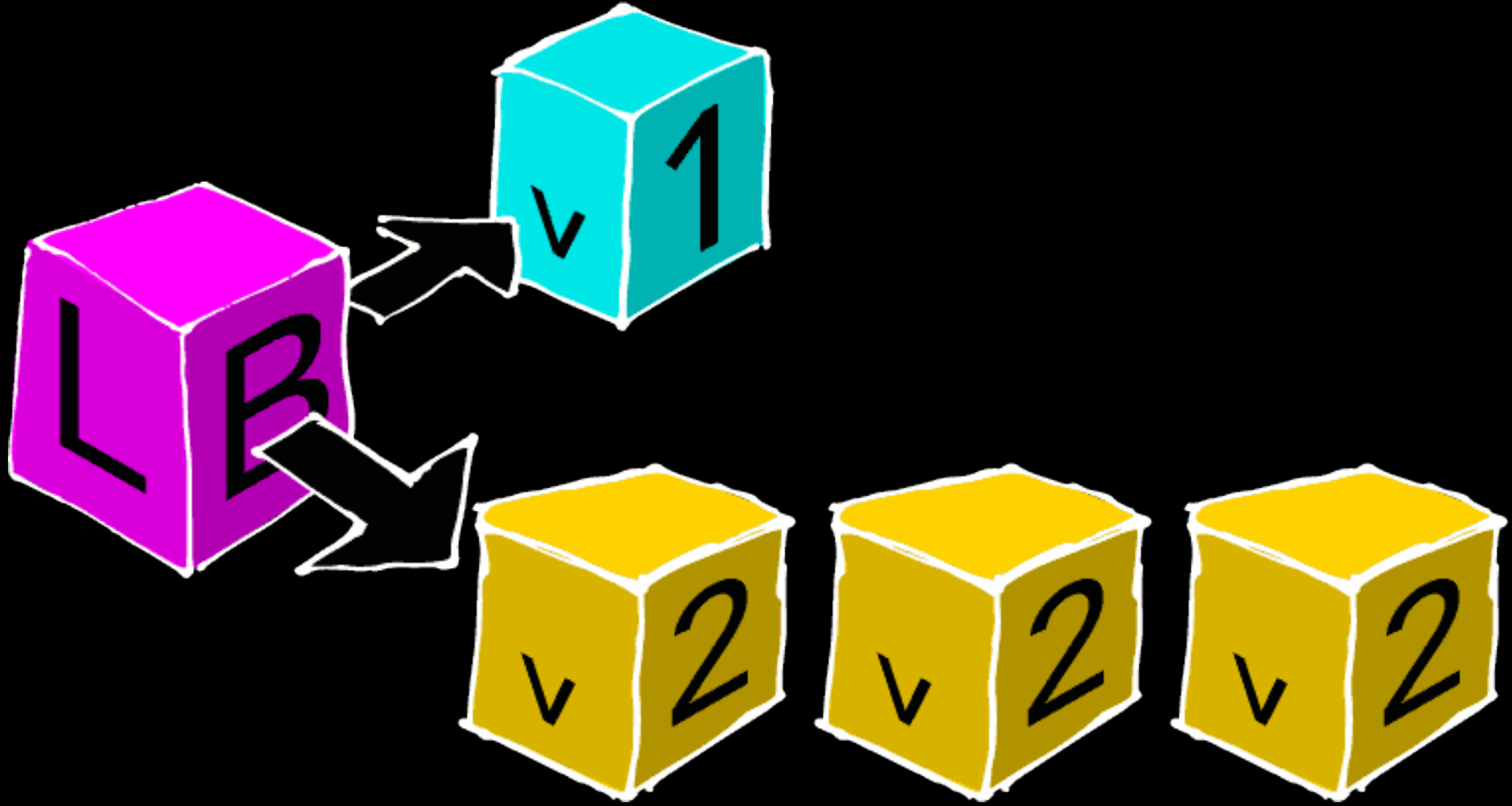


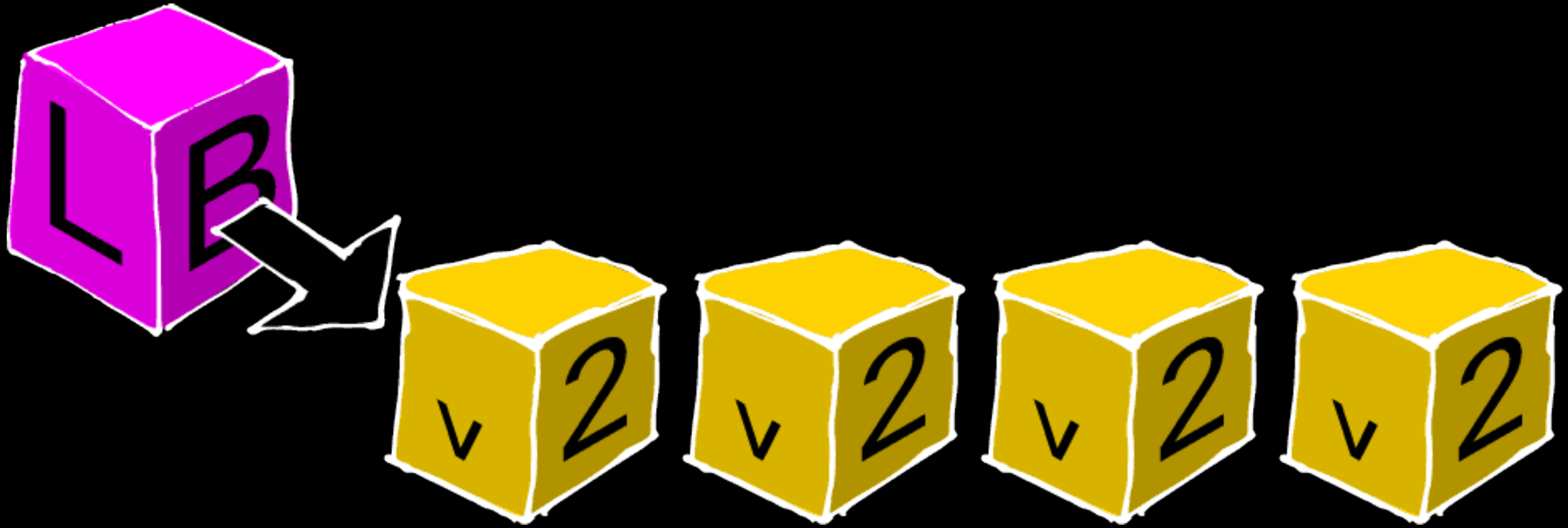
ROLLING DEPLOYMENTS

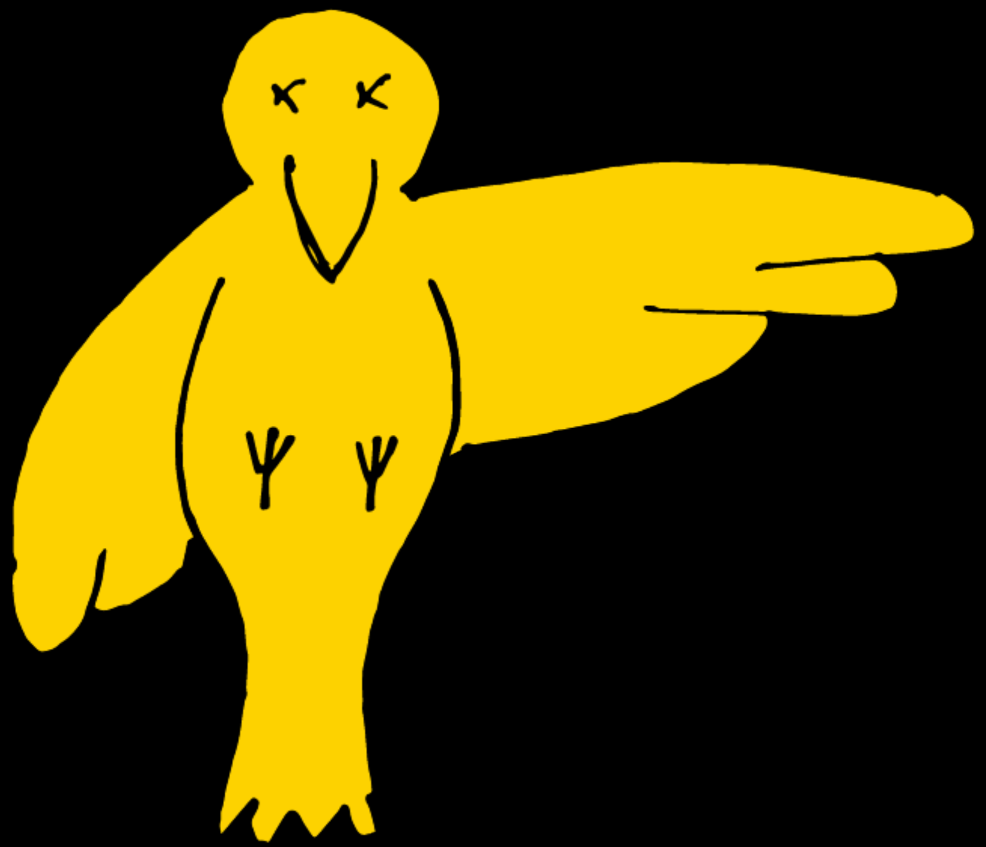

LB
v1
v1
v1
v1


LB
v1
v1
v1
v2


LB
v1
v1
v2
v2

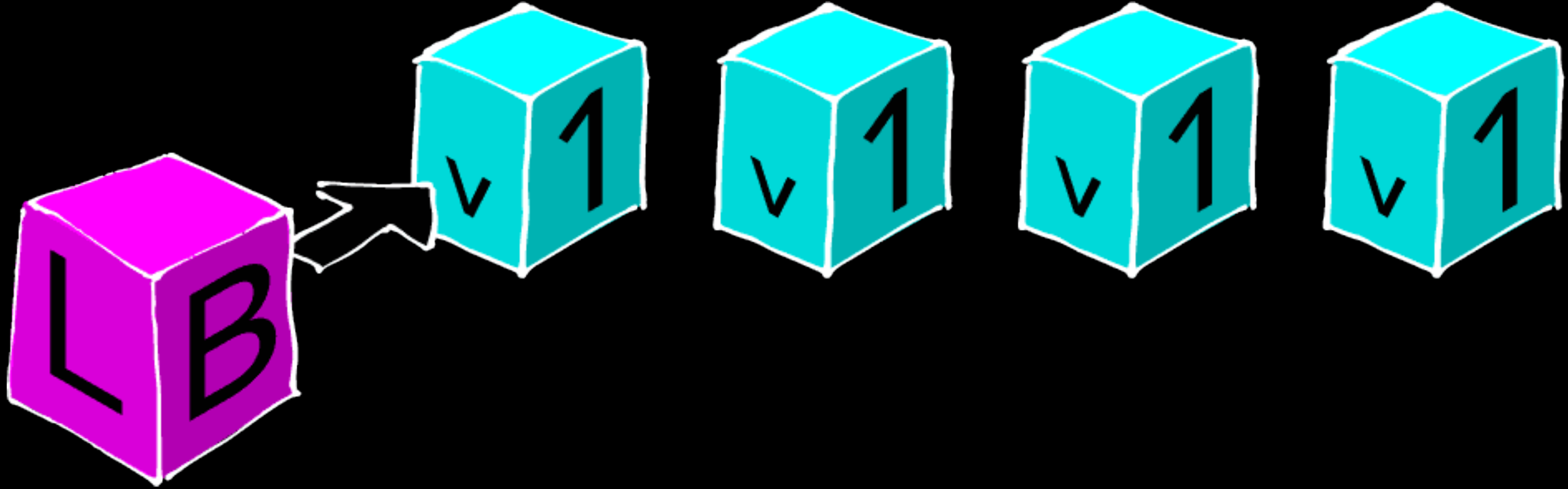

LB
v1
v2
v2
v2

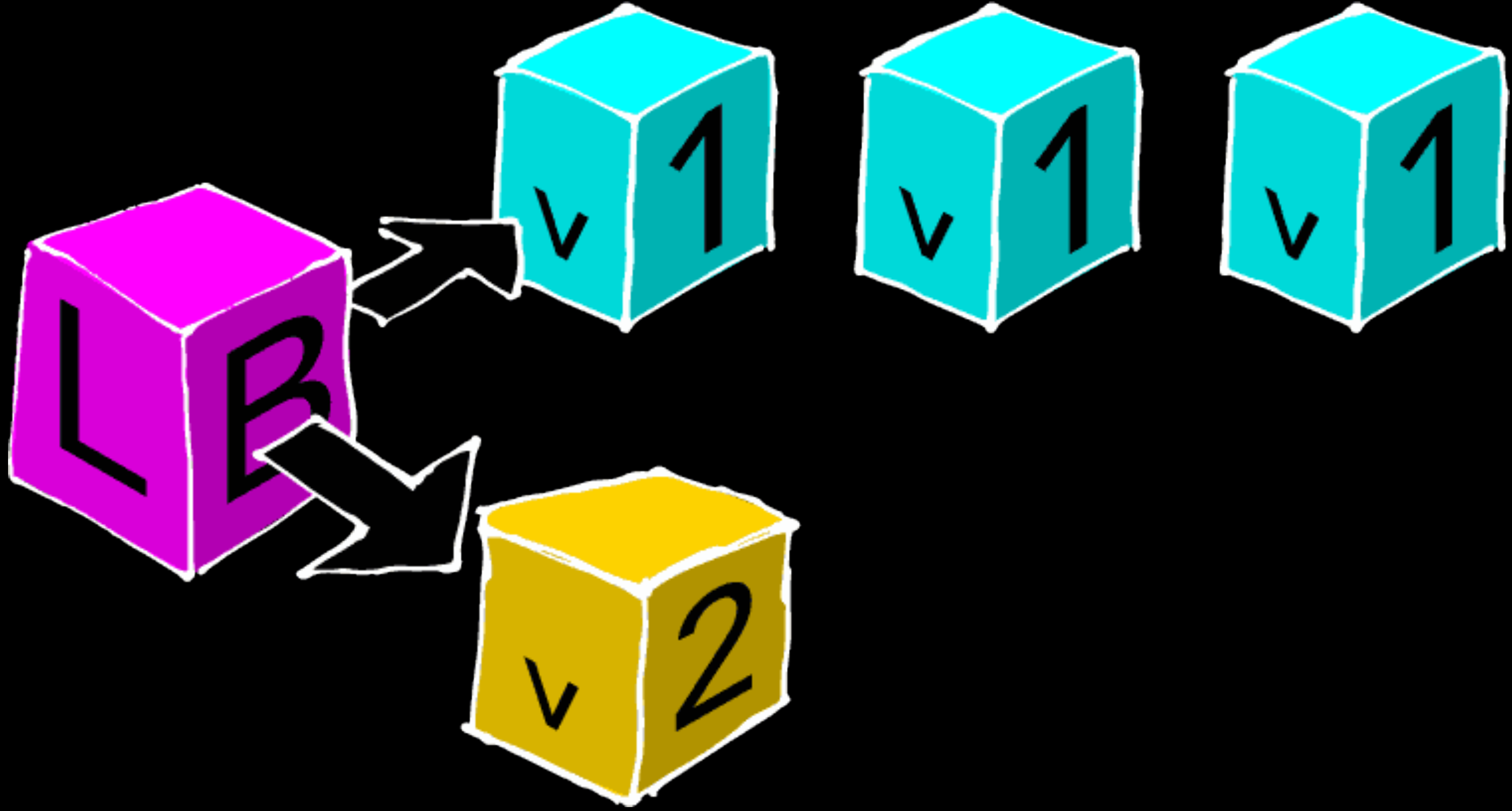

LB
v2
v2
v2
v2



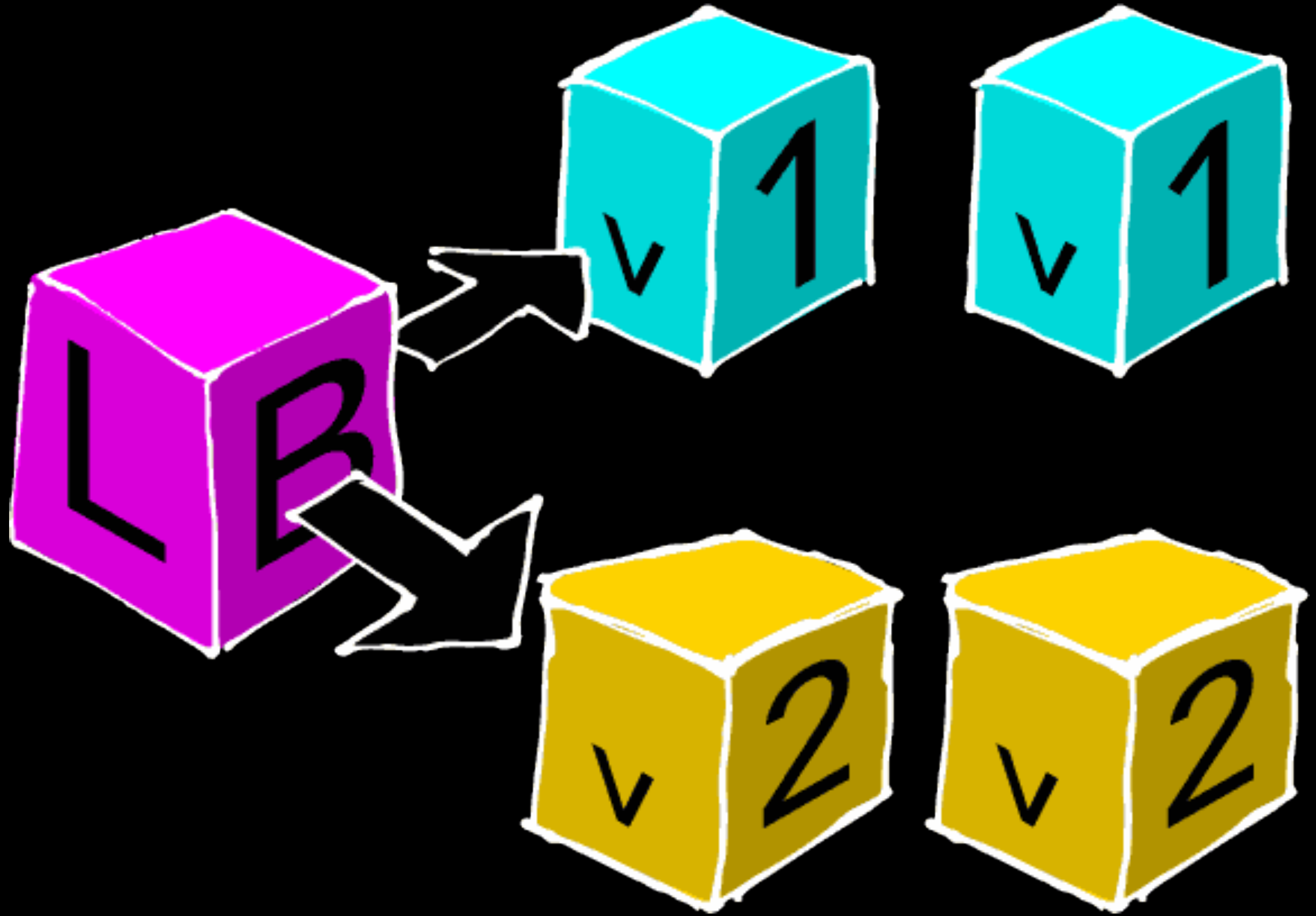
CANARY DEPLOYMENTS

or

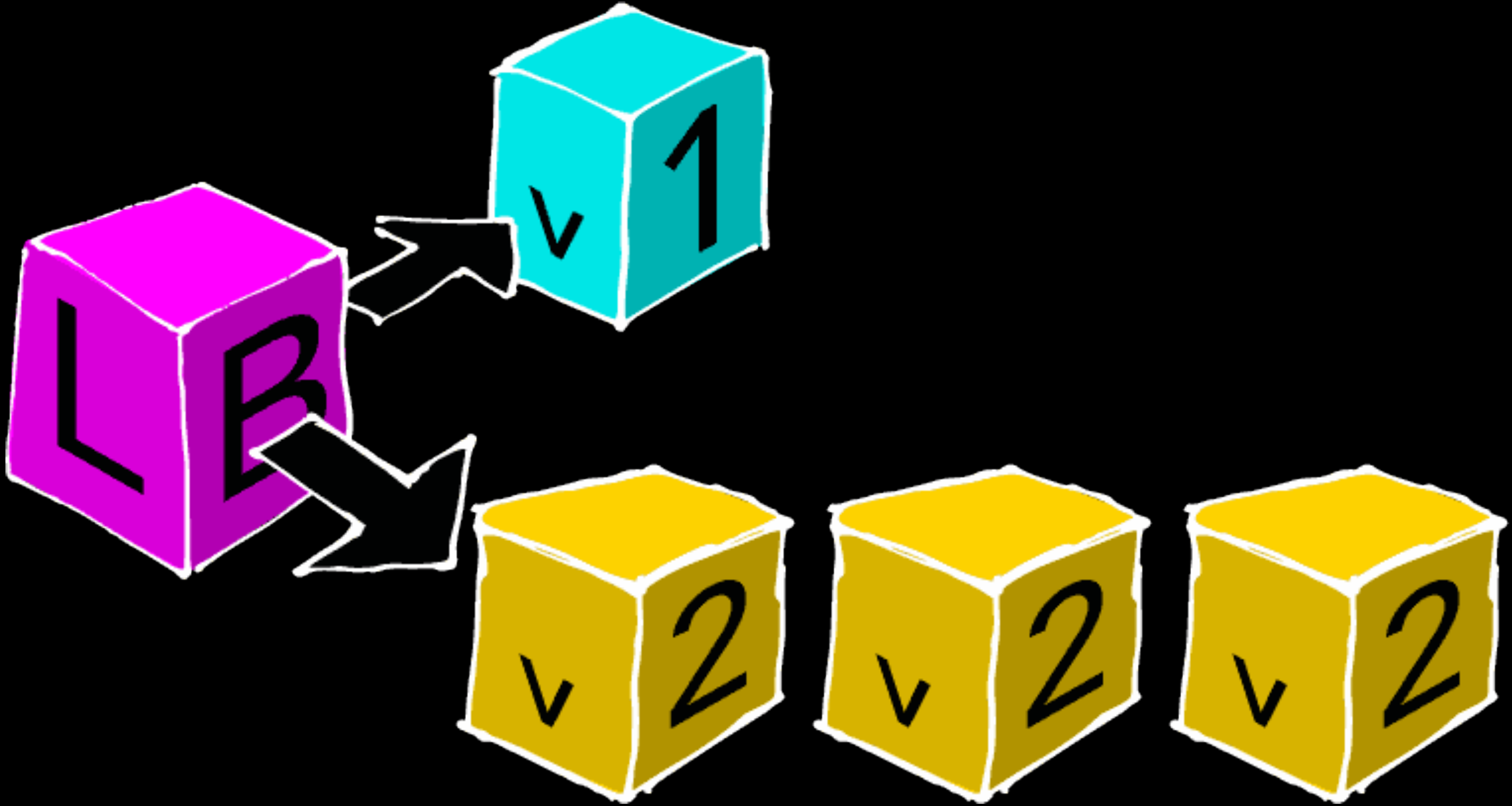

LB
v1
v1
v1
v1


LB
v1
v1
v1
v2

PAUSE. MONITOR.

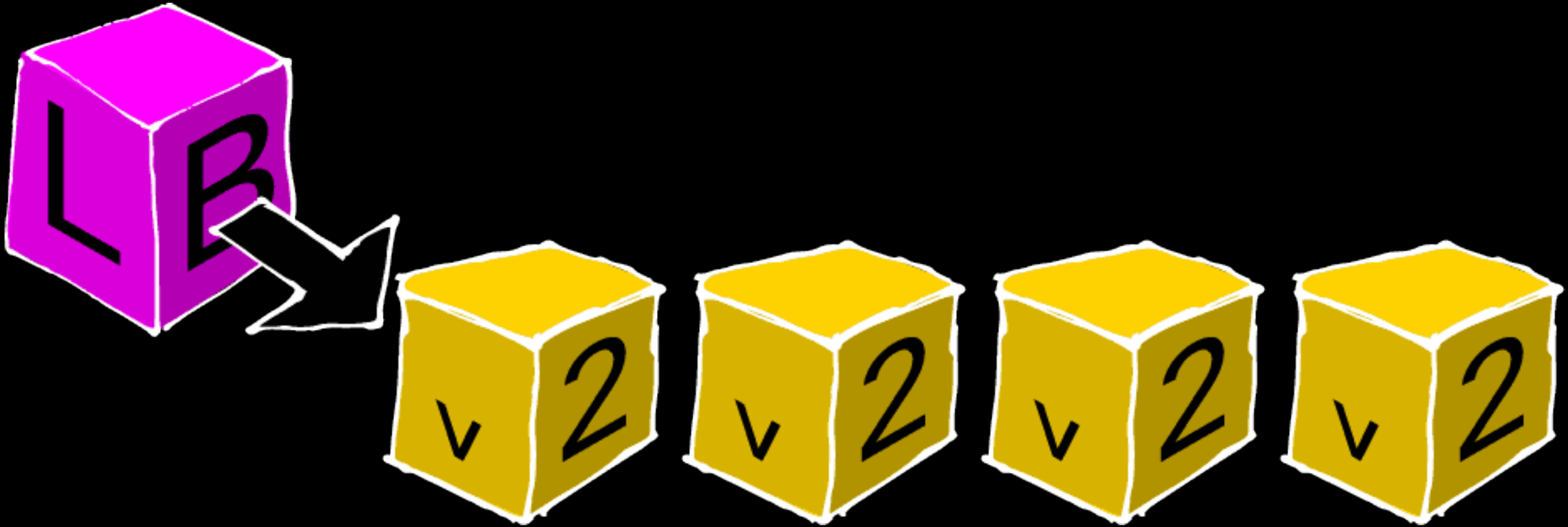

LB
v1
v1
v2
v2

DOUBLE CHECK.



LB
v1
v2
v2
v2

MAYBE ONE MORE TIME,
JUST TO BE CERTAIN.


LB
v2
v2
v2
v2

PARTY!

CANARY DEPLOYMENTS

- **Small scope**

CANARY DEPLOYMENTS

- Small scope
- **Limited ramifications**

CANARY DEPLOYMENTS

- Small scope
- Limited ramifications
- **Easier rollbacks**

CANARY DEPLOYMENTS

- Small scope
- Limited ramifications
- Easier rollbacks
- **Load tolerant**

CANARY DEPLOYMENTS

- Small scope
- Limited ramifications
- Easier rollbacks
- Load tolerant
- **Concurrency**

CANARY STRATEGY

CANARY STRATEGY

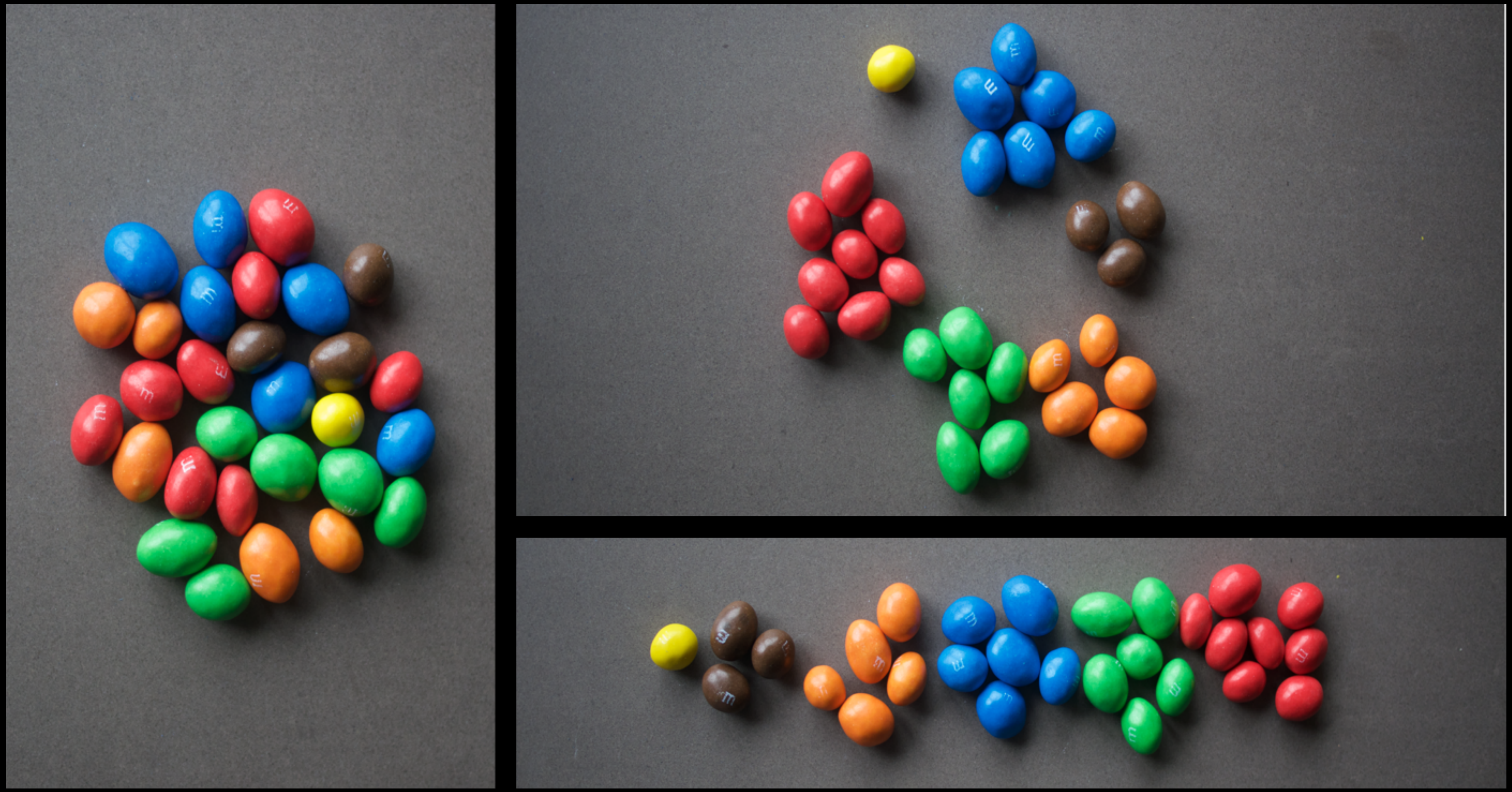
How do you choose your sample set?

- **Random**

CANARY STRATEGY

How do you choose your sample set?

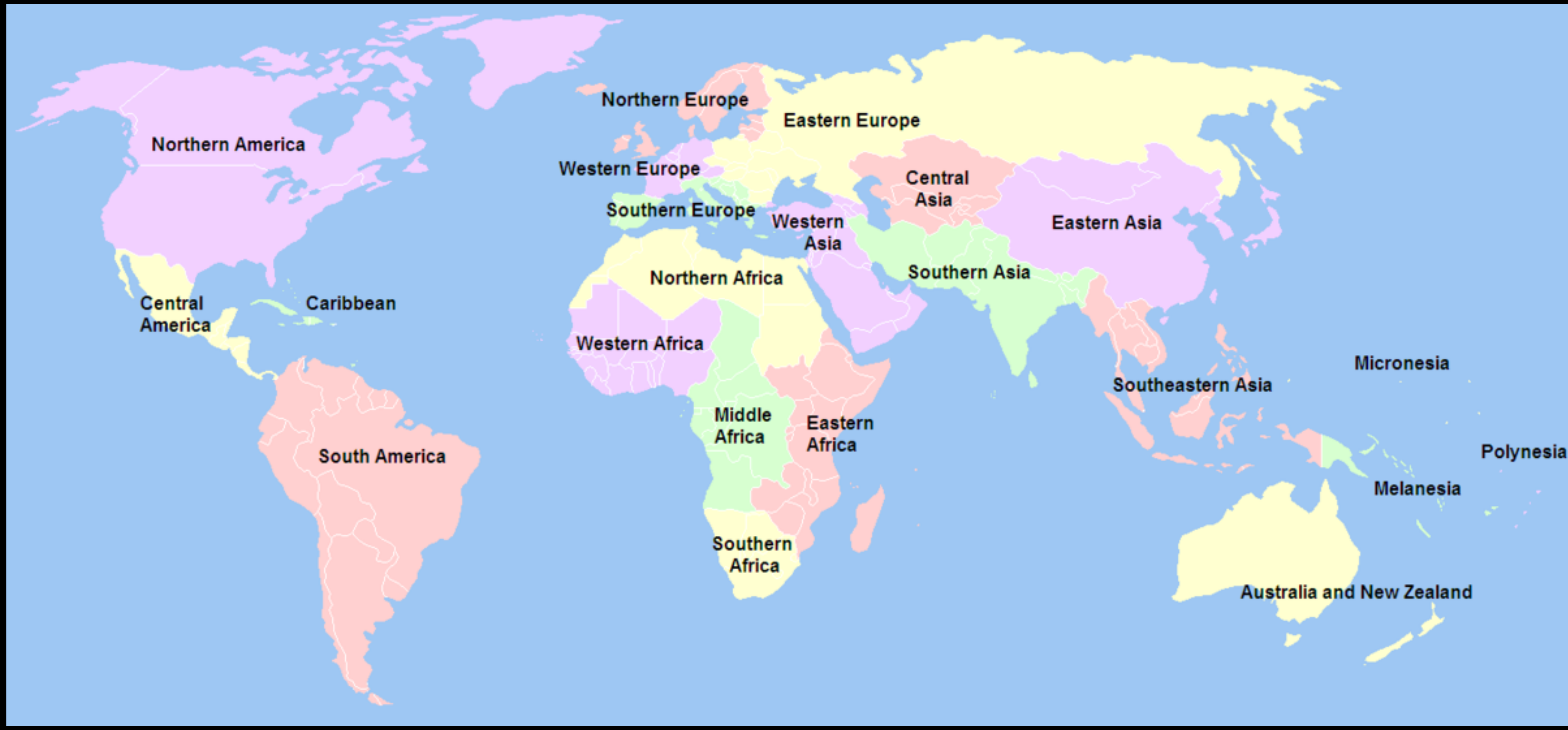
- Random
- **Representative**



CANARY STRATEGY

How do you choose your sample set?

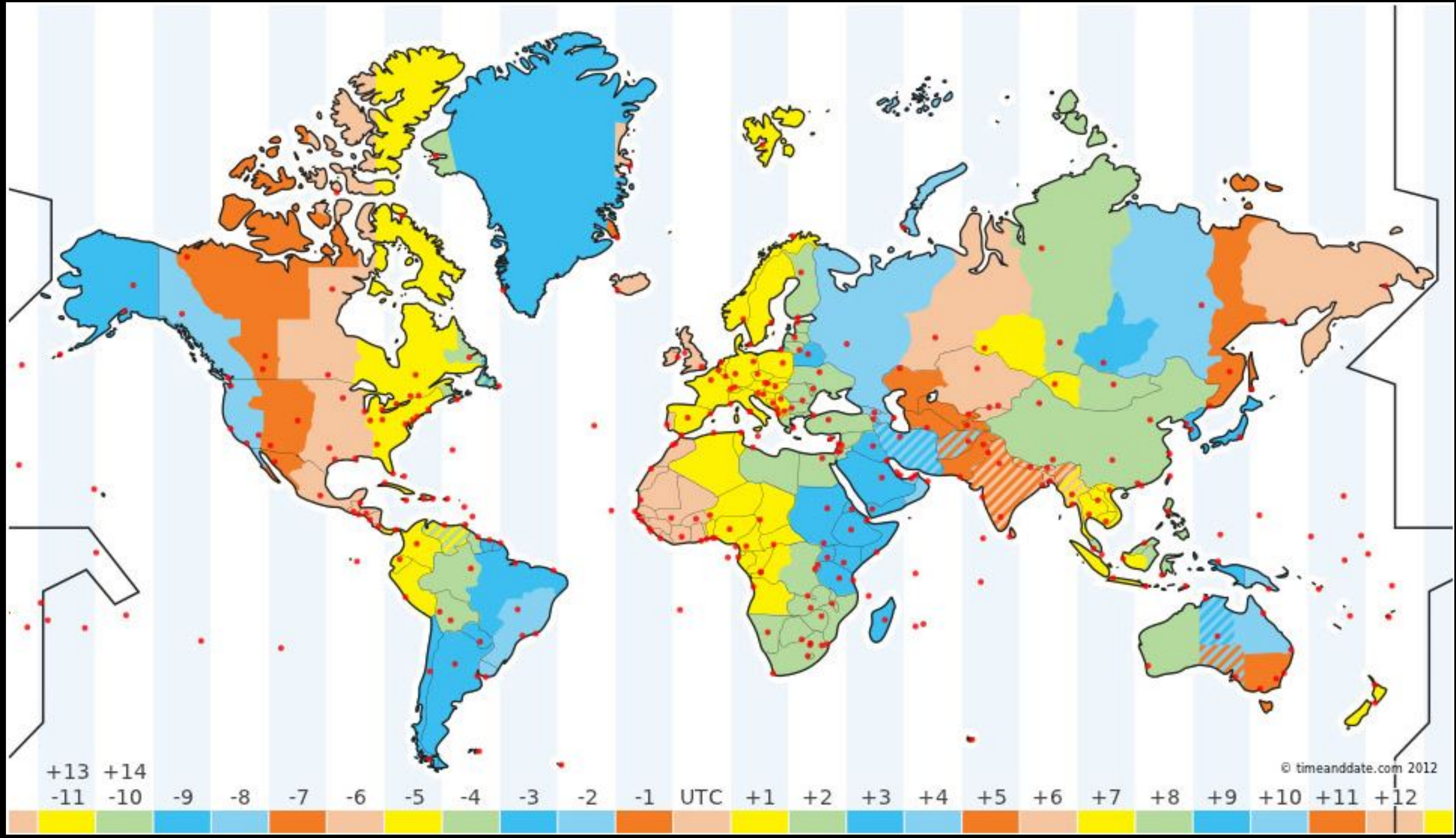
- Random
- Representative
  - **Geography**



CANARY STRATEGY

How do you choose your sample set?

- Random
- Representative
  - Geography
  - **Time**


+13 +14
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 UTC +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12
© timeanddate.com 2012

CANARY STRATEGY

How do you choose your sample set?

- Random
- Representative
  - Geography
  - Time
  - **Use patterns**

CANARY STRATEGY

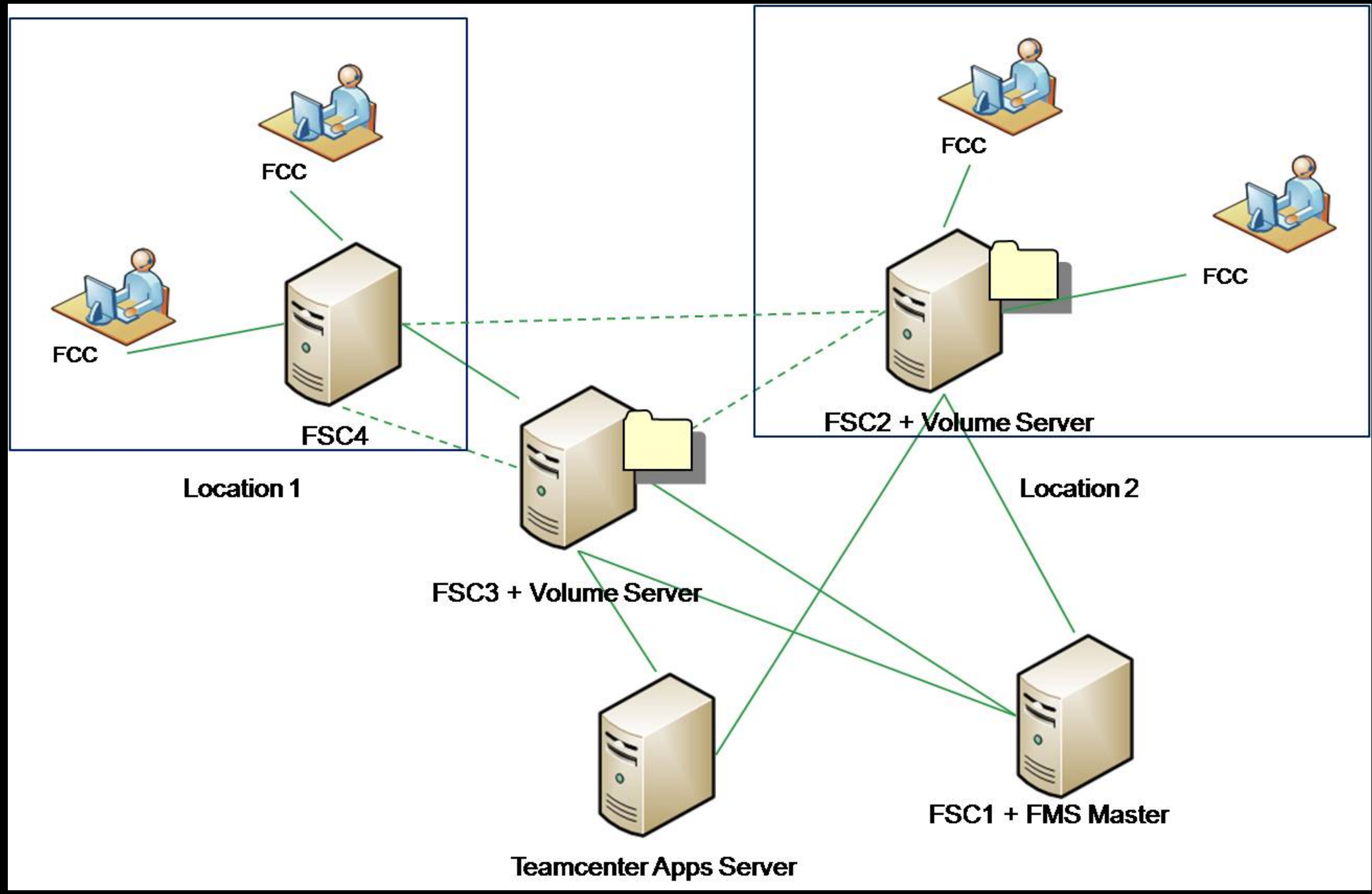
How do you choose your sample set?

- Random
- Representative
  - Geography
  - Time
  - Use patterns
- **Granularity**

CANARY STRATEGY

How do you choose your sample set?

- Random
- Representative
  - Geography
  - Time
  - Use patterns
- Granularity
- **Resource mapping**



MONITORING STRATEGY

MONITORING STRATEGY

How do you evaluate your deployment?

- **Tags! Tags! Tags! Tags! Tags!**

DATAPOINT

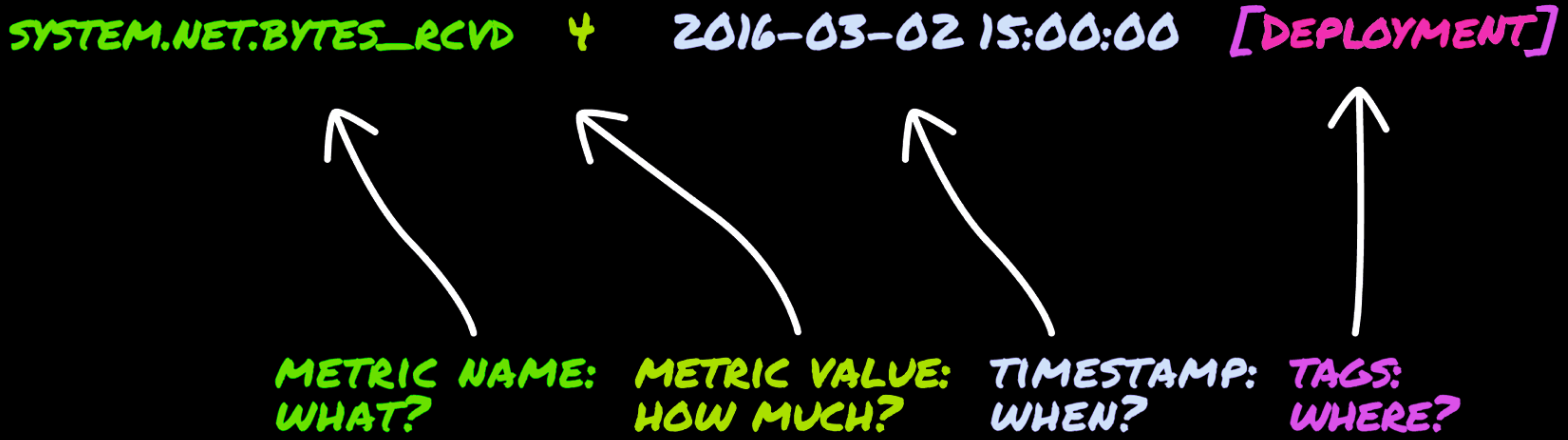
SYSTEM.NET.BYTES_RCVD 4 2016-03-02 15:00:00

METRIC NAME:
WHAT?

METRIC VALUE:
HOW MUCH?

TIMESTAMP:
WHEN?

Datapoint


system.net.bytes_rcvd
4
2016-03-02 15:00:00
[Deployment]
Metric name: what?
Metric value: how much?
Timestamp: when?
Tags: where?

MONITORING STRATEGY

How do you evaluate your deployment?

- Tags!
- **p90, p95, p99**

MONITORING STRATEGY

How do you evaluate your deployment?

- Tags!
- p90, p95, p99
- **Outliers**

SADANDUSELESS.COM
DAY 14
THEY STILL SUSPECT NOTHING

Outliers: one of these things is not like the others

58G
56G
54G
52G
50G
48G
46G
44G
42G
40G
38G
36G
34G
32G
30G
28G
26G
24G
22G
20G
18G
16G
role:canary, host:i-0b1ac7744a8253aac, org_id:2
48G

MONITORING STRATEGY

How do you evaluate your deployment?

- Tags!
- p90, p95, p99
- Outliers
- **Anomalies**

Anomalies: It wasn’t like this before

12.5
12
11.5
11
10.5
10
9.5
9
8.5
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3

Anomalies: It wasn't like this before

12.5
12
11.5
11
10.5
10
9.5
9
8.5
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3

Anomalies: It wasn't like this before

12.5
12
11.5
11
10.5
10
9.5
9
8.5
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3
M T W TH F
M T W TH F
M T W TH F
M T W TH F
NOT A NORMAL TUESDAY

SIGNALS TO WATCH

Latency

SIGNALS TO WATCH

Latency

Errors

SIGNALS TO WATCH

Latency

Errors

Traffic

SIGNALS TO WATCH

Latency

Errors

Traffic

Saturation

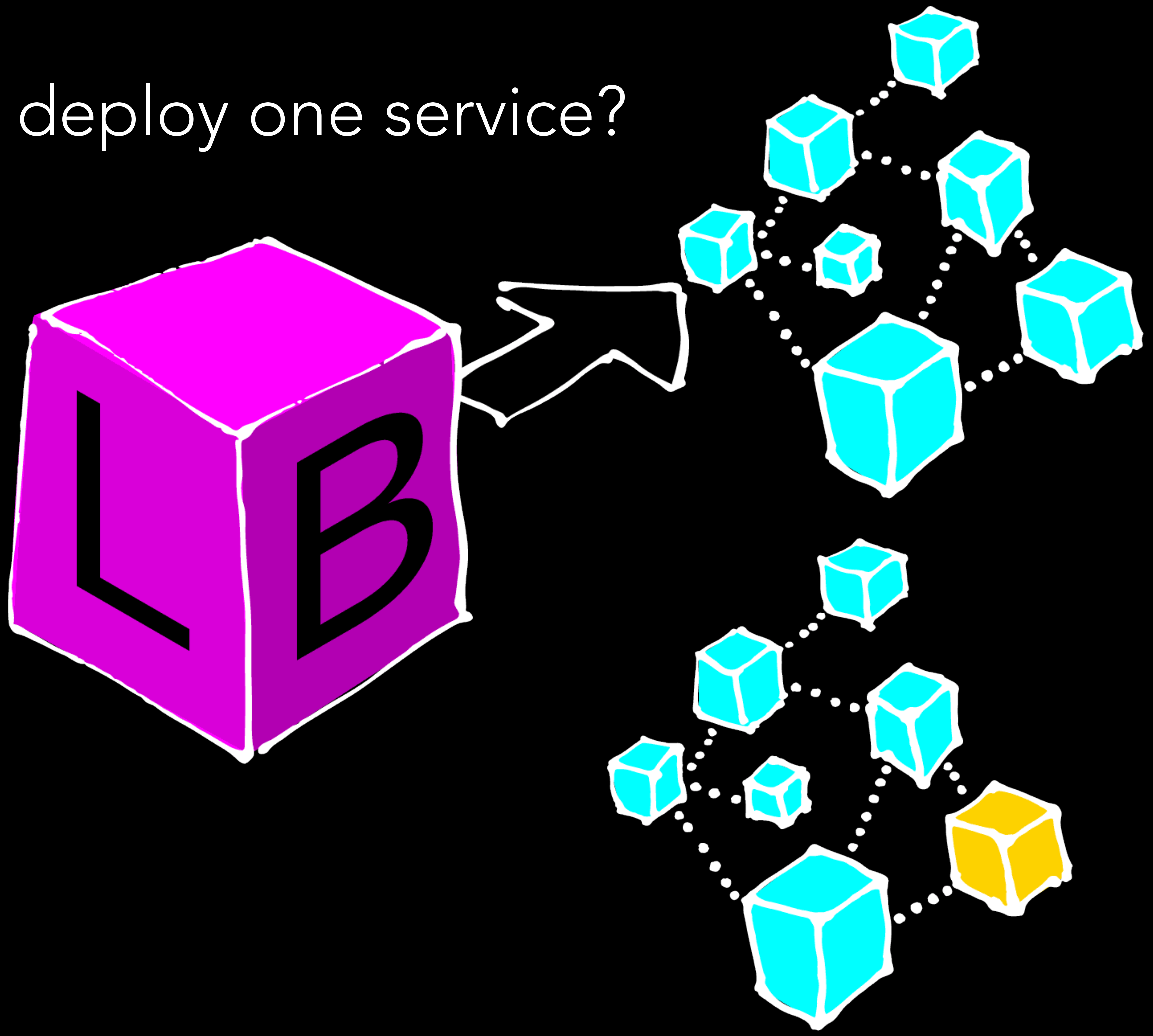
WHAT DOES KUBERNETES HAVE TO DO WITH ANY OF THIS?

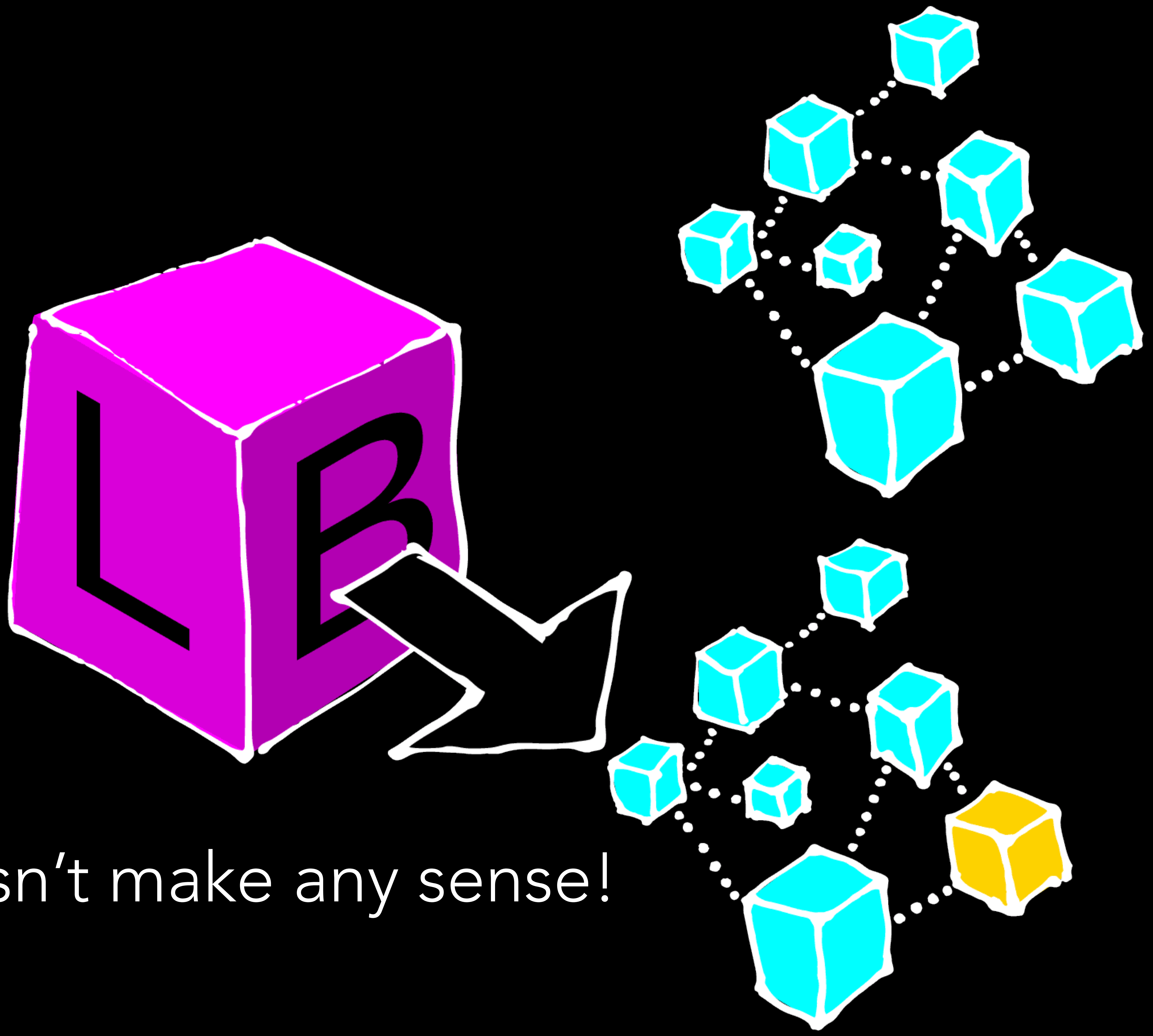
CONTAINER
ORCHESTRATOR

~~CONTAINER~~ SERVICE ORCHESTRATOR

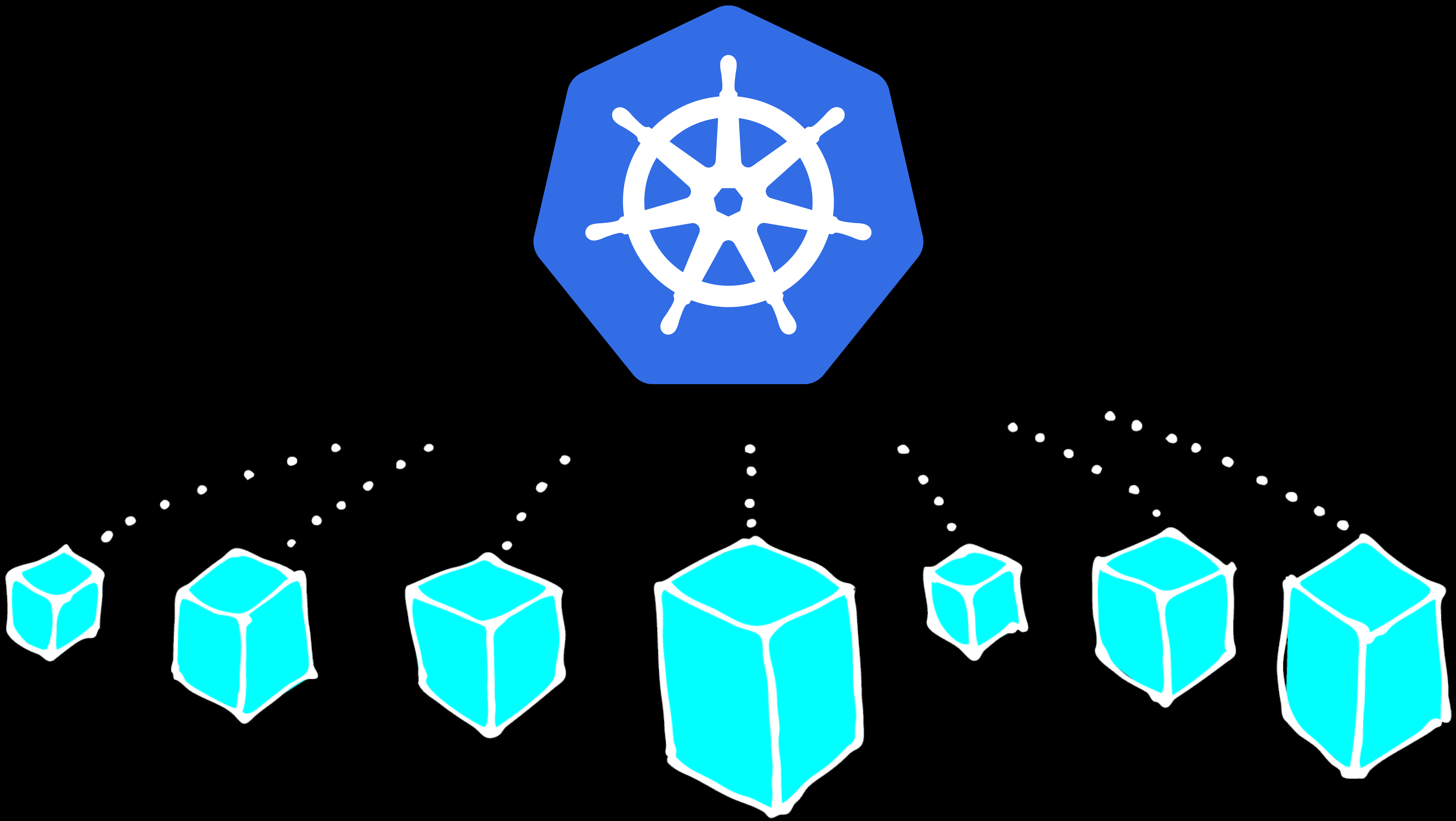
p.s. - Maybe a Squirtle orchestrator? Talk to me later if you want a sticker.

So what if we want to deploy one service?

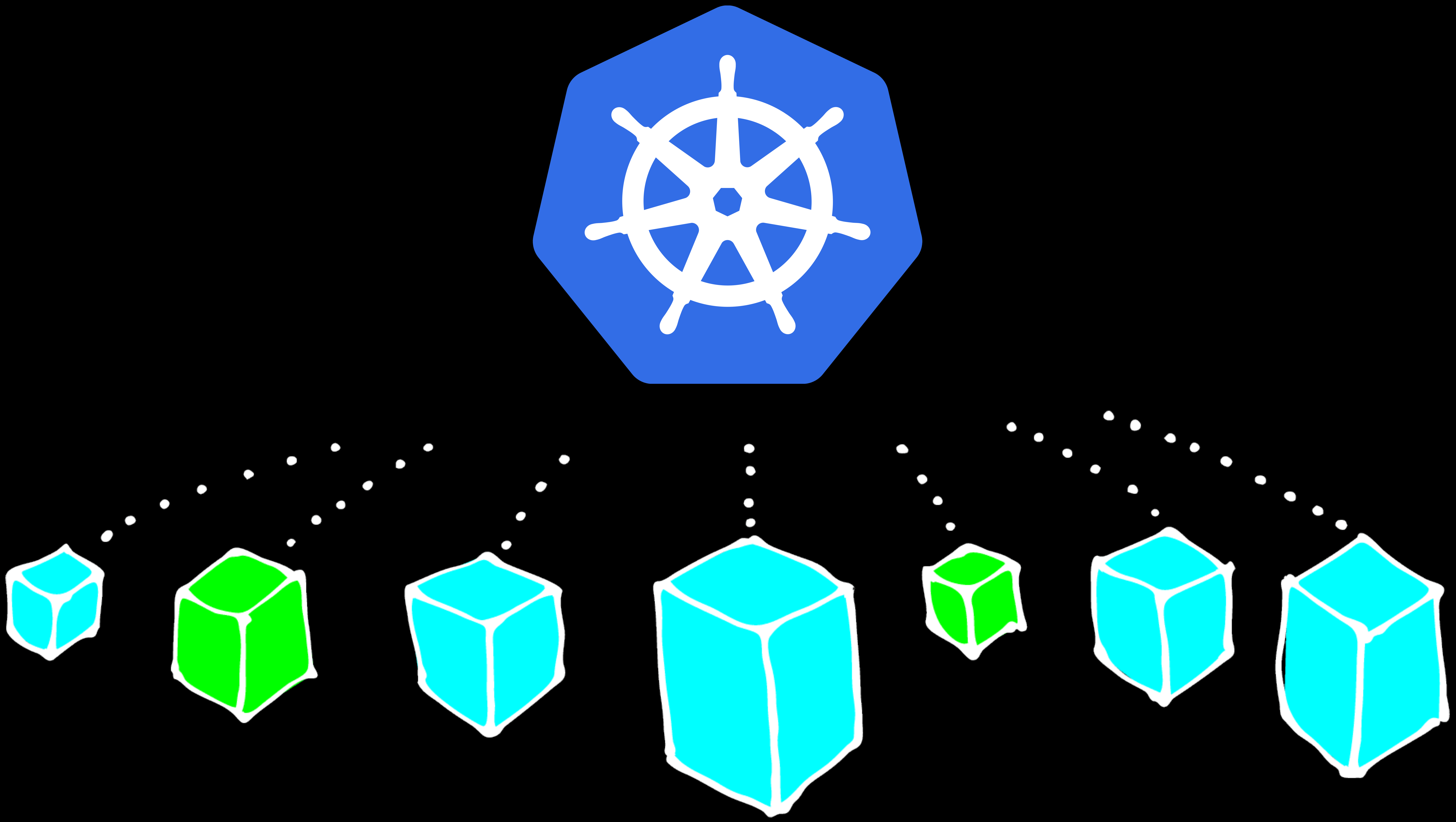




Blue-green doesn’t make any sense!

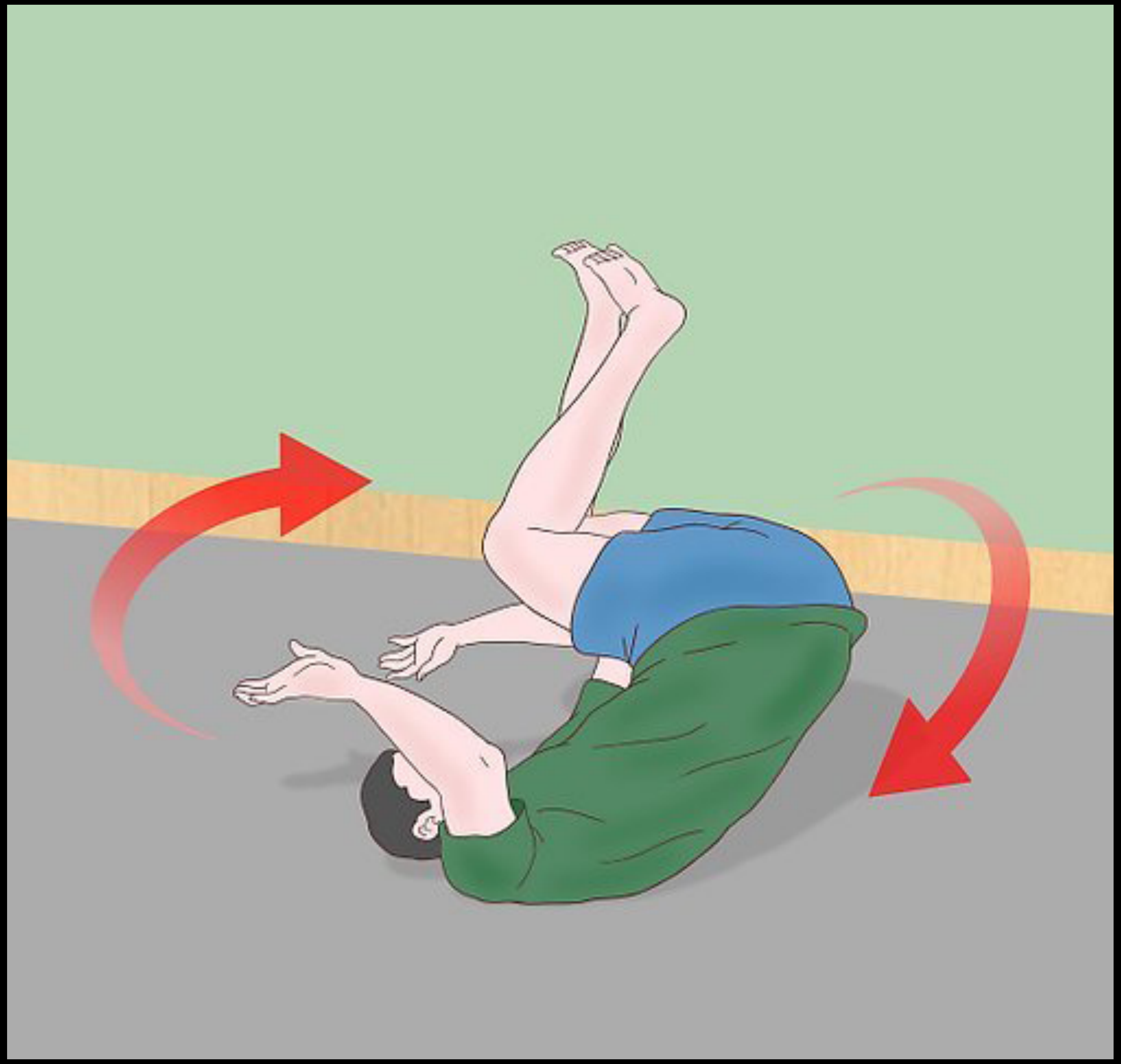


Kubernetes handles service deployments

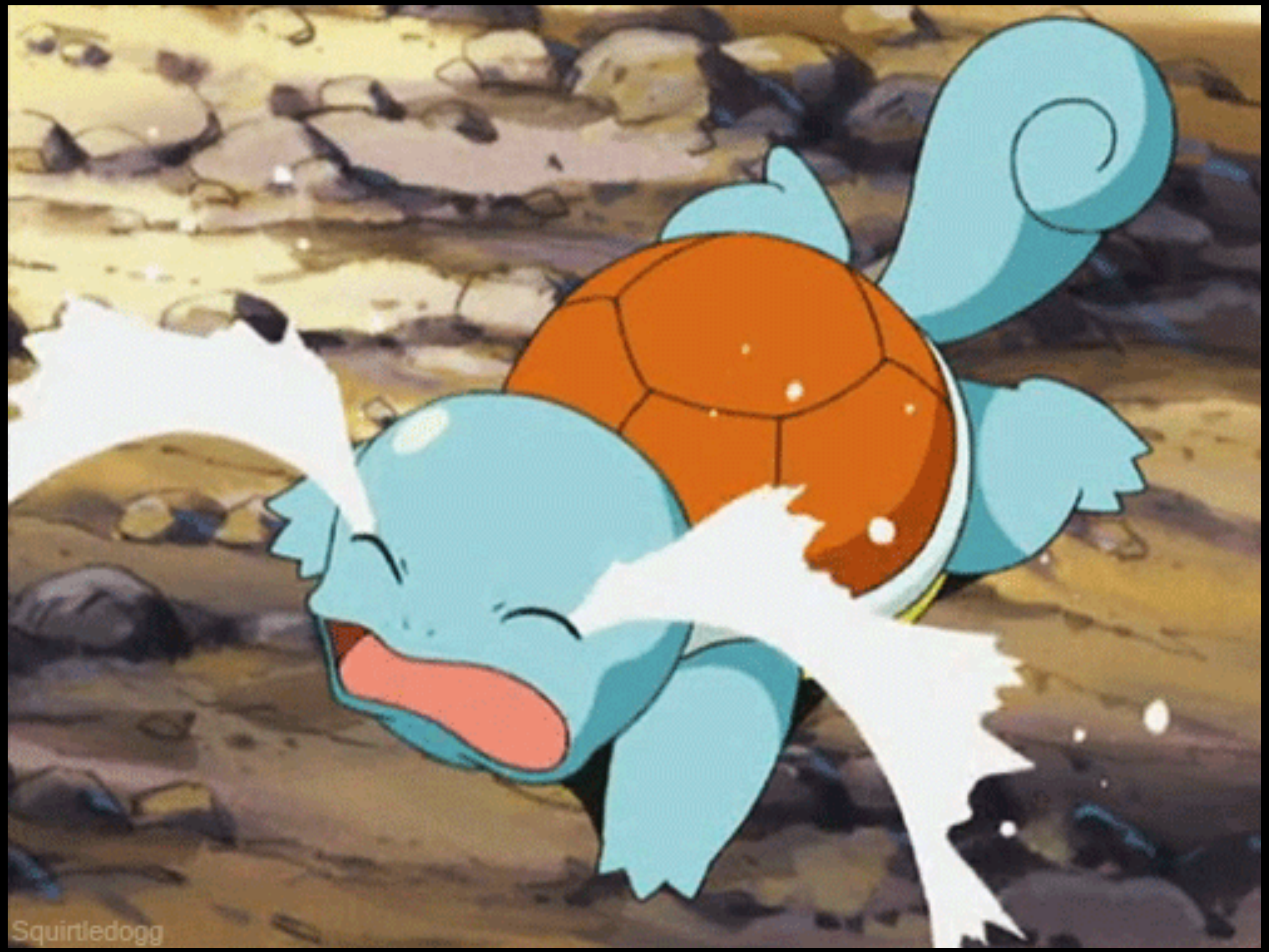


Kubernetes handles service deployments. YAY!

WHY DO I NEED A SERVICE MESH?



Kubernetes does rolling deploys really well!



Canary deploys,
not so much.

CANARY DEPLOYING WITH KUBERNETES

SERVICE

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  ports:
  - port: 80
    name: http
  selector:
    app: my-app
```

DEPLOYMENT

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v2
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v2
        imagePullPolicy: Always
```

CANARY?

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 9
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v2
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v2
        imagePullPolicy: Always
```

CANARY?

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 8
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v2
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v2
        imagePullPolicy: Always
```

CANARY?

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 7
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v2
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v2
        imagePullPolicy: Always
```

CANARY?

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 99
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v2
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v2
        imagePullPolicy: Always
```

WHAT DOES A SERVICE MESH GET YOU?

SERVICE MESHES

- **Routing & load balancing**

SERVICE MESHES

- Routing & load balancing
- **Service discovery**

SERVICE MESHES

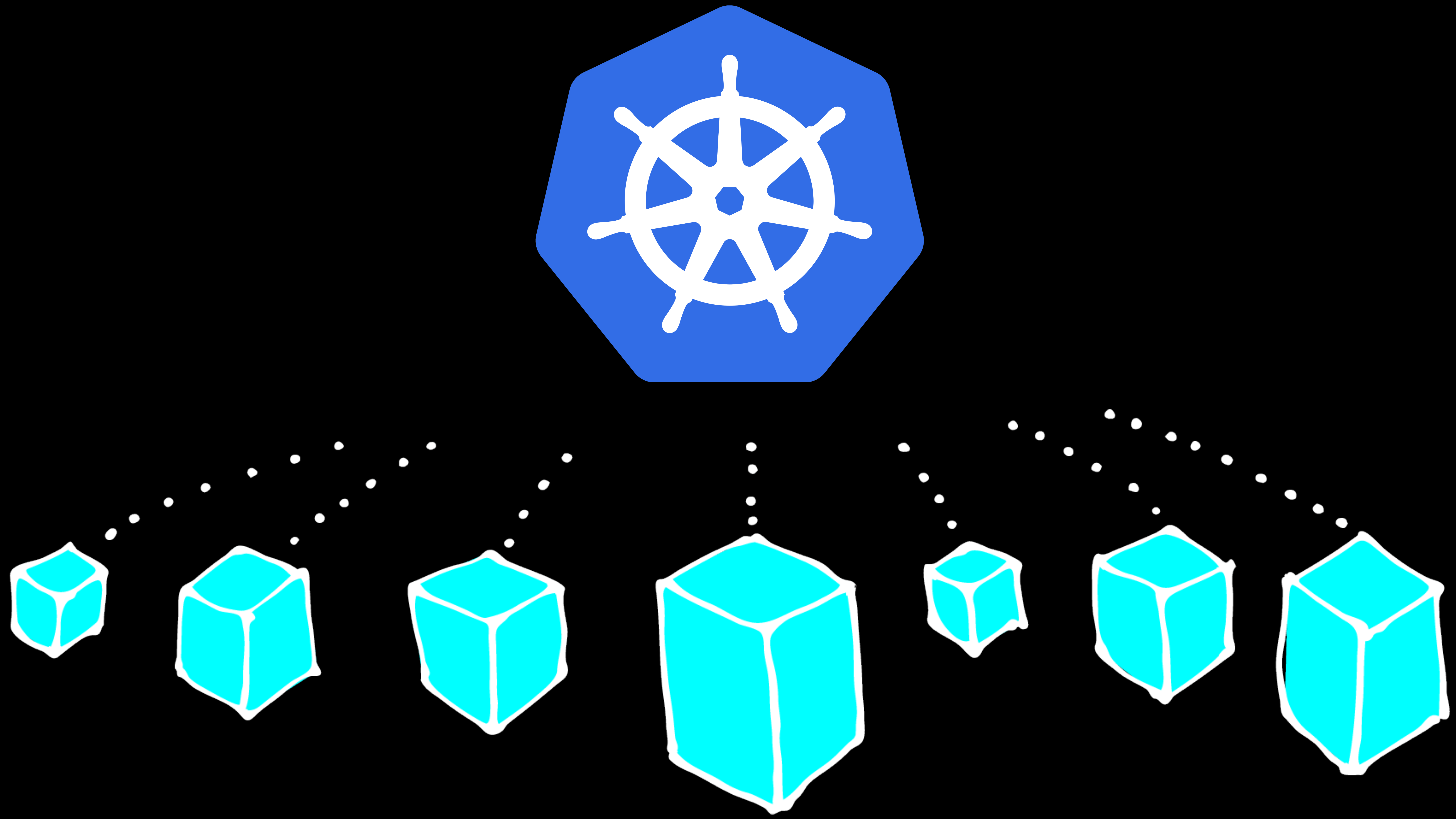
- Routing & load balancing
- Service discovery
- **Timeouts & retries**

SERVICE MESHES

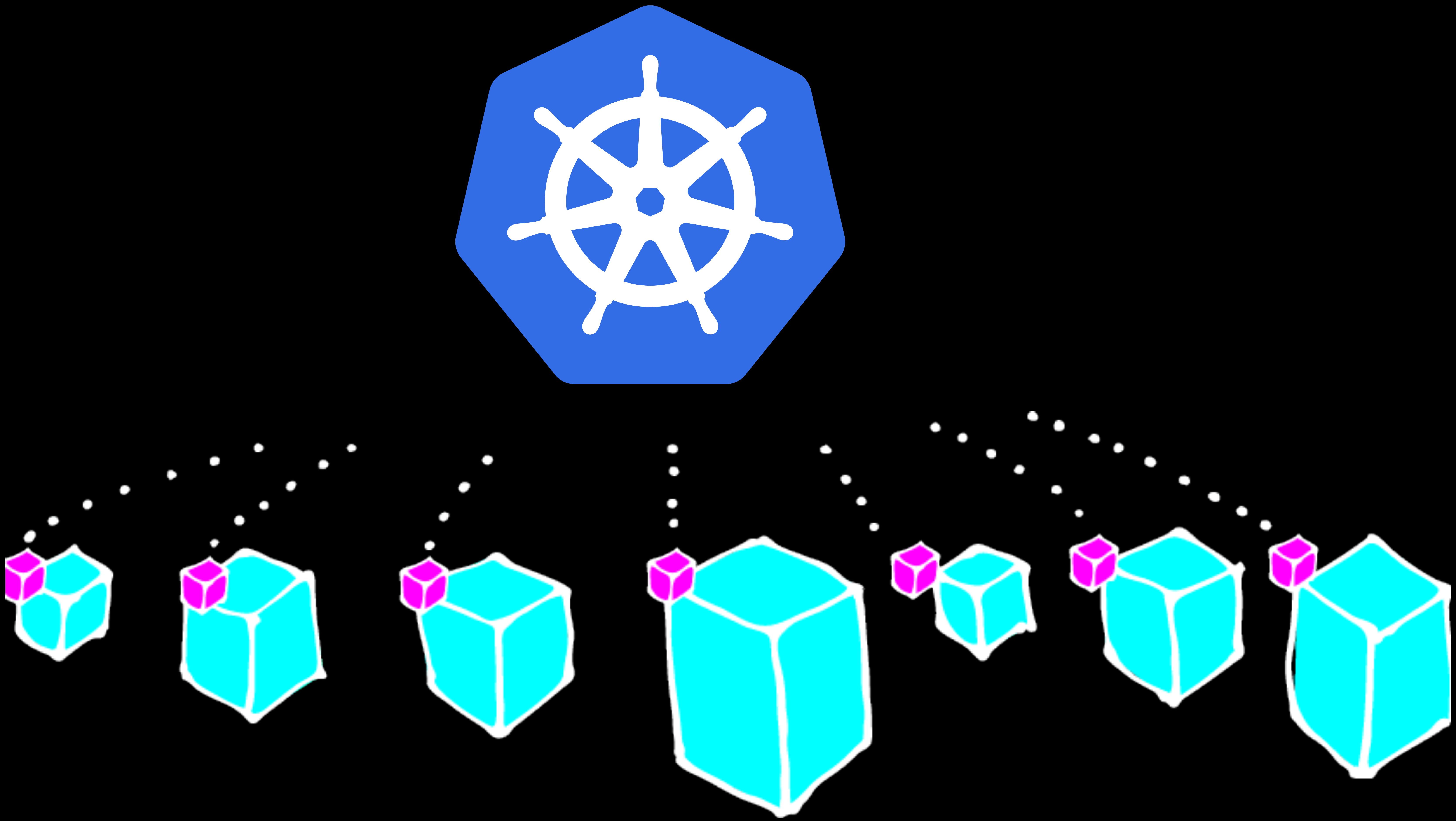
- Routing & load balancing
- Service discovery
- Timeouts & retries
- **Policy enforcement**

SERVICE MESHES

- Routing & load balancing
- Service discovery
- Timeouts & retries
- Policy enforcement
- **Monitoring & tracing**

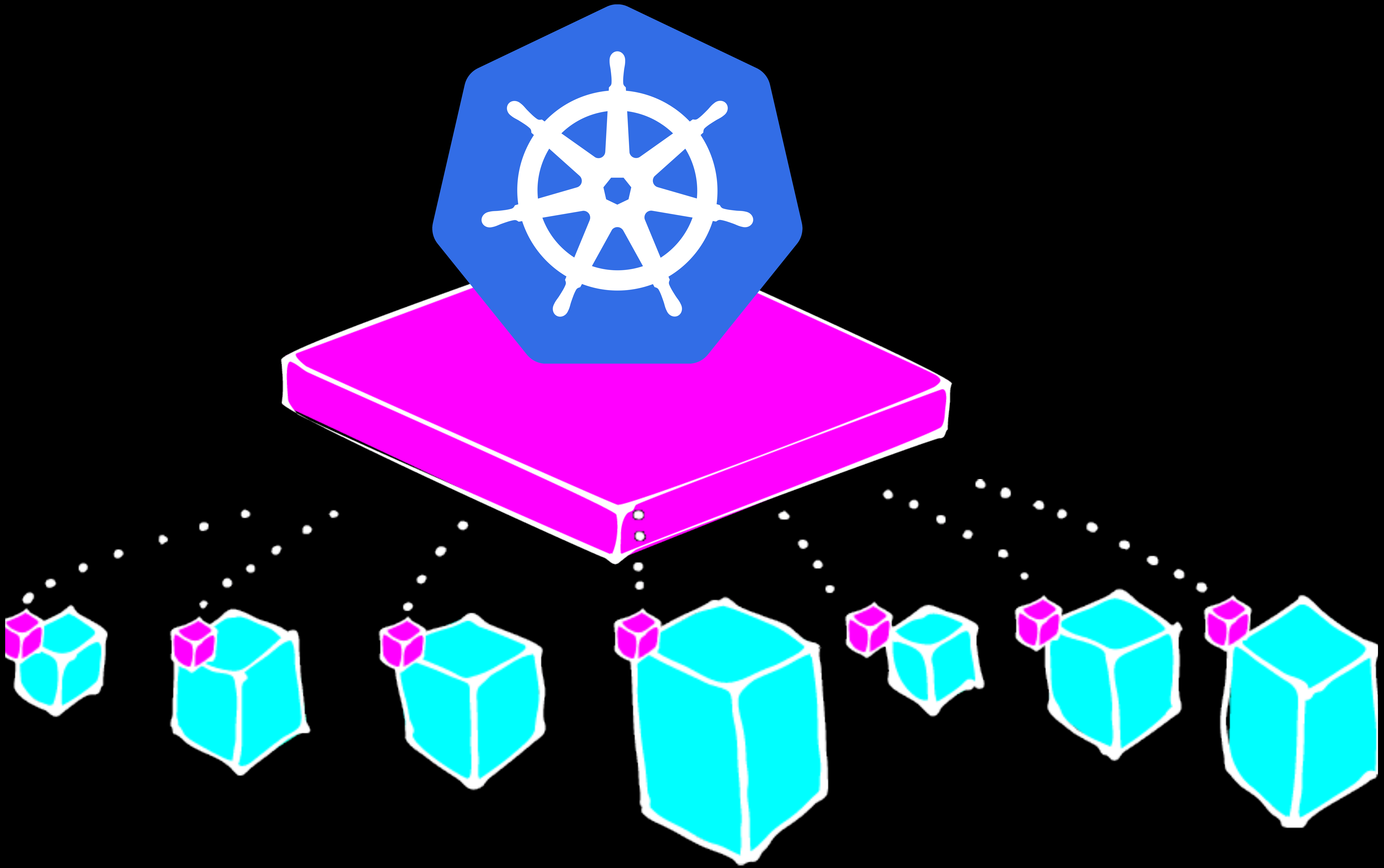


How does it work?



1. Add a data plane

envoy



2. Add a control plane

CANARY DEPLOYING WITH ISTIO

SERVICE

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  ports:
  - port: 80
    name: http
  selector:
    app: my-app
```

SERVICE

```
apiVersion: v1
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  ports
  - port: 8
    nam     tp
  selector:
    app: my-app
```

SAME!

DEPLOYMENT

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

DEPLOYMENT

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

SAME!

(sort of)

```
istioctl kube-inject -f my.yaml > mod.yaml
kubectl apply -f mod.yaml
```

TEENAGE MUTATING WEBHOOK ADMISSION CONTROLLERS!

AKA AUTO-SIDECAR INJECTION

ISTIO VIRTUALSERVICES

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: my-app-routing
spec:
  hosts:
    - my-app
  http:
  - route:
    - destination:
        host: my-app
        subset: v1
```

ISTIO DESTINATIONRULES

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: my-app-destination
spec:
  host: my-app
  subsets:
    - name: v1
      labels:
        version: v1
```

ISTIO

```yaml
kind: VirtualService
...
spec:
  hosts:
    - my-app
  http:
  - route:
    - destination:
        host: my-app
        subset: v1
```

ISTIO

```yaml
kind: DestinationRule
...
spec:
  host: my-app
  subsets:
    - name: v1
      labels:
        version: v1
```

K8S

```yaml
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      ...
```

DEPLOYMENT

```
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v1
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v1
        imagePullPolicy: Always
```

```
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        version: v2
    spec:
      containers:
      - name: my-app
        image: jyee/my-app:v2
        imagePullPolicy: Always
```

ISTIO DESTINATIONRULES

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: my-app-destination
spec:
  host: my-app
  subsets:
    - name: v1
      labels:
        version: v1
    - name: v2
      labels:
        version: v2
```

ISTIO VIRTUALSERVICES

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: my-app–routing
spec:
  hosts:
    - my-app
  http:
  - route:
    - destination:
        host: my-app
        subset: v1
      weight: 80
  - route:
    - destination:
        host: my-app
        subset: v2
      weight: 20
```

ISTIO VIRTUALSERVICES

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
...
  http:
  - match:
    - headers:
        cookie:
          user: my-logged-in-user
    route:
    - destination:
        host: my-app
        subset: v2
      weight: 20
```

WHAT ELSE CAN IT DO?

LOTS!

https://istio.io/docs/reference/config/

RECAP

- **Service meshes give you more control**

RECAP

- Service meshes give you more control
- **Canary deploys: Representative & Granular**

RECAP

- Service meshes give you more control
- Canary deploys: Representative & Granular
- **Monitoring: Tags, Outliers, Anomalies**

RECAP

- Service meshes give you more control
- Canary deploys: Representative & Granular
- Monitoring: Tags, Outliers, Anomalies
- **What to watch: Latency, Errors, Traffic, Saturation**

RECAP

- Service meshes give you more control
- Canary deploys: Representative & Granular
- Monitoring: Tags, Outliers, Anomalies
- What to watch: Latency, Errors, Traffic, Saturation
- **GO PLAY WITH ISTIO 1.0.2!!!**

QUESTIONS?

email: jyee@datadoghq.com
twitter: @gitbisect