

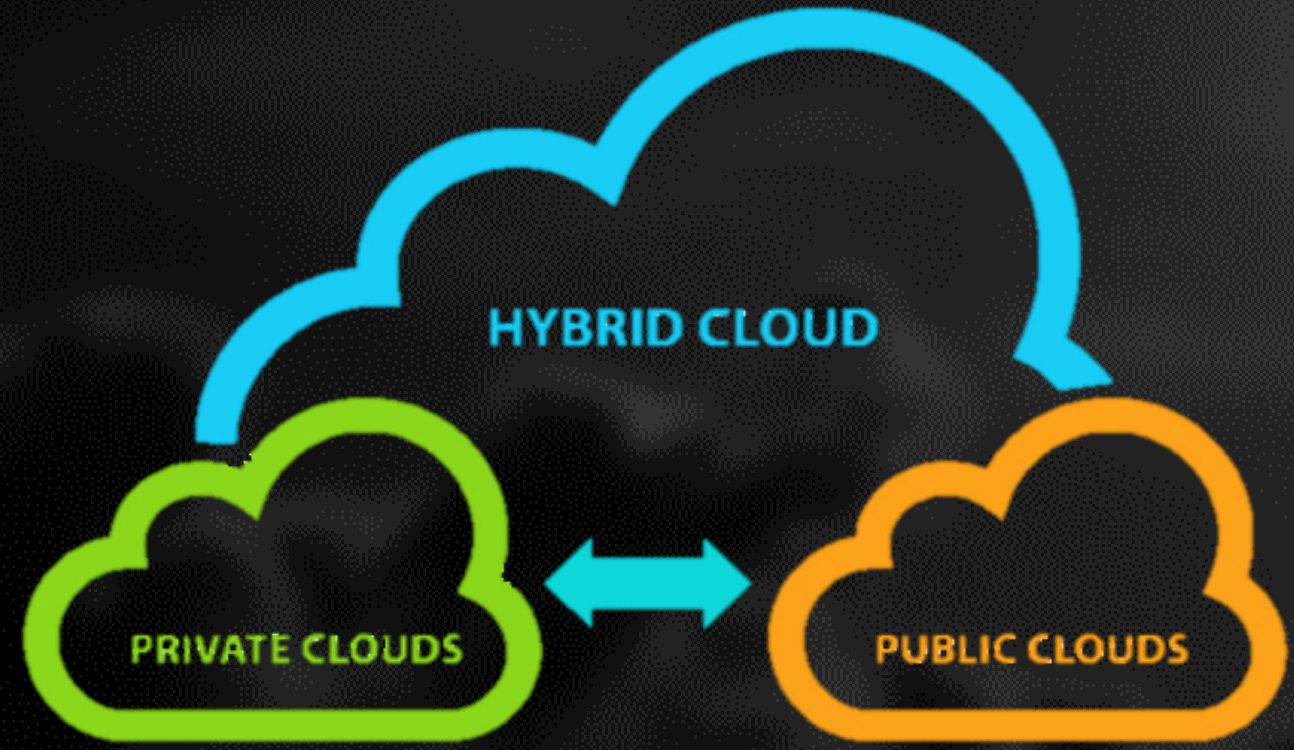


# Cloud.Next goes Hybrid !!

## The emergence of hybrid clouds

Eran Raichstein  
eranra@il.ibm.com

The 15th Cloud Control Workshop

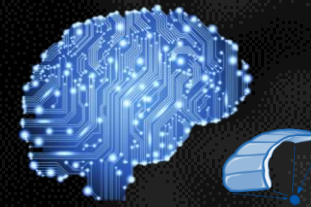


# Eran Raichstein

@ IBM Research - Haifa Israel

IBM Cloud architect → Networking


2nd time in Cloud Control Workshop



CogNETive



Sodalite

- **Cloud computing**
- **VM's → Containers → Kubernetes**
- **Hybrid clouds** 
- **Hybrid cloud networking**
- **Cloud native vs. Hybrid cloud native applications (No time for this 😞 )**



# What is cloud computing?



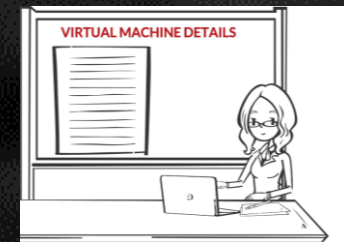


# What is cloud computing (e.g. IBM Definition) ?

Cloud computing is the delivery of **on-demand** computing resources. Everything from **applications to data centers** — over the internet on a **pay-for-use** basis.

**Or Intranet**

- **Elastic resources:** Scale up or down quickly and easily to meet changing demand.
- Metered services: **Pay per use.**
- **Self-service:** Find all the IT resources you need, with self-service access.



<https://www.ibm.com/cloud/learn/cloud-computing>



# Public cloud vs. Private cloud



Google Cloud



IBM Cloud



OPENSTACK



OPEN SHIFT















IBM Cloud Private



# Public cloud vs. Private cloud



|  | Public cloud  | Private cloud   |
|--|---|---|
| <b>Cost Efficient (\$)</b>                       | Critical for Providers <br>Important to customer | Important              |
| <b>Isolation &amp; Secured</b>                   | Critical for Multi-tenancy                       | Project level          |
| <b>Self-service portal</b>                       | Required   | Required               |
| <b>Workload</b>                                  | Common use case (e.g. e-shop)   | Sometimes specific  |
| <b>Network</b>                                   | Isolation, flexibility & remote access  | Flexible  |
| <b>Operations – backup, billing, monitoring.</b> | Only provider cares                             | Needed                 |
| <b>Scalable Storage</b>                          | Required                                       | Required             |
| <b>Virtualization – VMs, Container</b>           | Only provider cares                            | Full responsibility  |

## A virtual machine



- Server level functionary
- Proprietary packaging
- Strong Isolation
- Stateful
  - Includes: Storage + Compute + Network
- All resources are Virtualized (mini – server)





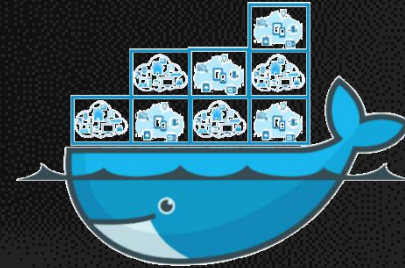
# From Virtual machines to Containers



## Private/Public VM cloud



## Containers



Process level functionality (microservice)  
Industry standard (docker)

Stateless  
Quick power on/off  
Scale-Out (elastic by nature)

```
C:\Users\sanjayd\source\repos\DockerTestApp\DockerWebTestApp>kubect1 get pods
NAME                READY    STATUS    RESTARTS   AGE
quotes-4213455183-1d0d5  1/1     Running  0          13h
quotes-4213455183-1r738  1/1     Running  0          13h
quotes-4213455183-33d52  1/1     Running  0          13h
quotes-4213455183-7w208  1/1     Running  0          13h
quotes-4213455183-vnxzk  1/1     Running  0          13h
```



**Nobody ever got  
fired for buying IBM.**

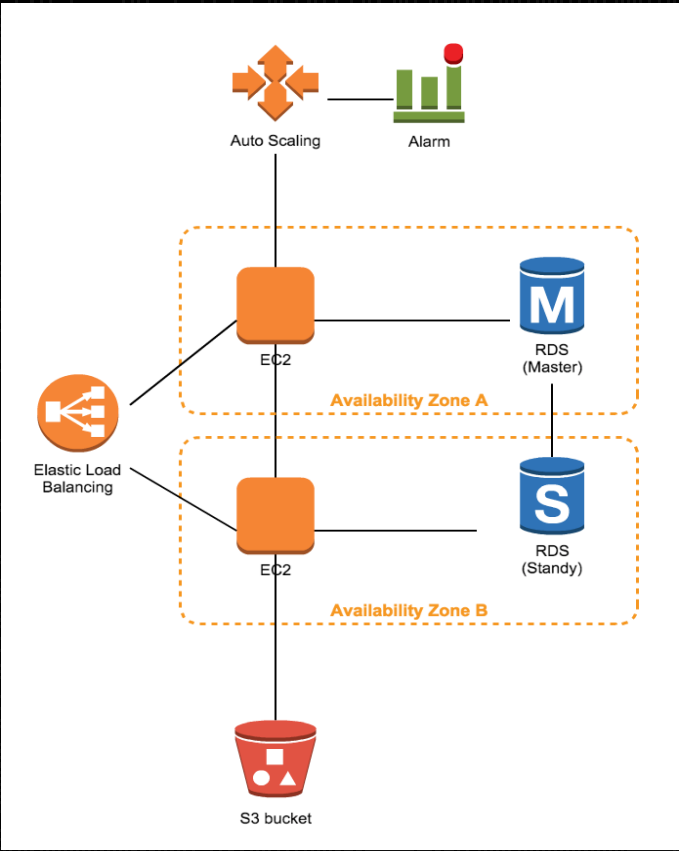
Single option  
Simple & efficient



# Cloud (As we know it today)



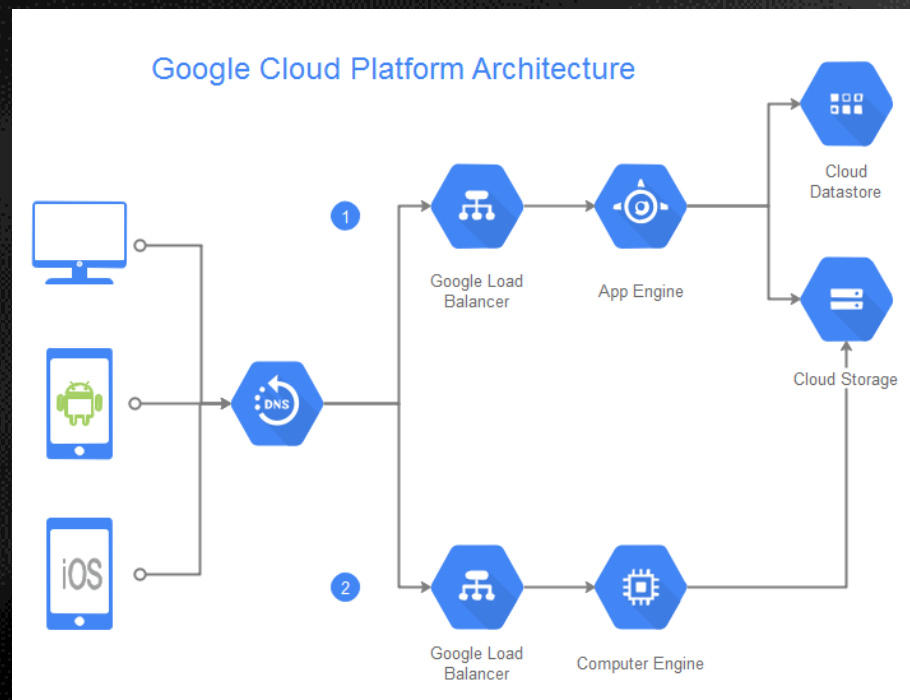
## AWS Style



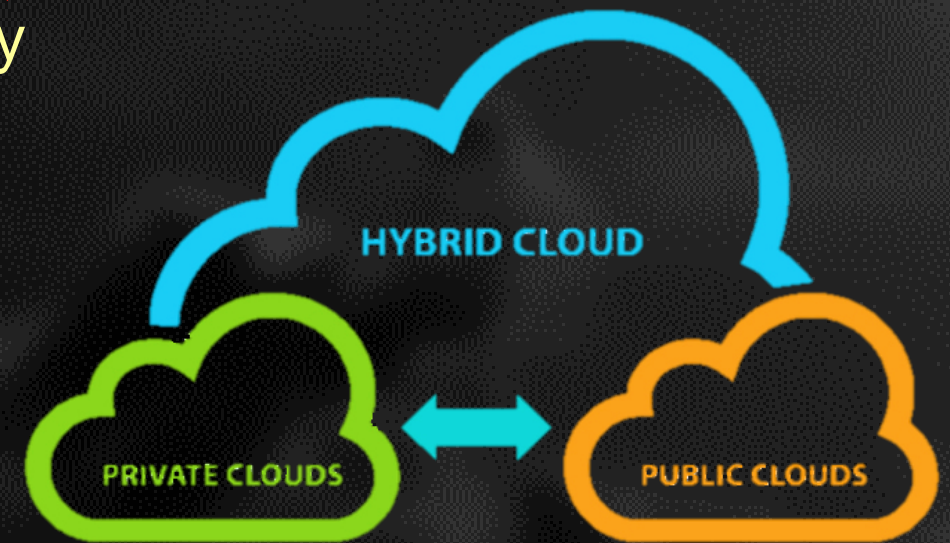
## VMWARE Style



## Google Style



- Becoming common for large enterprise customers
- Built on top of containers/microservices architecture (always?)
  - No vendor locking (and a lot of Open Source projects e.g. Kubernetes)
- **Scale-out** (Public & Private, Multiple locations) NEW
- Consumed from multiple vendors NEW
  - More than a single vendor for same functionality
- Massive networking NEW
  - Intra-site, internet, cross sites, cross vendors NEW
- Services and Storage through APIs



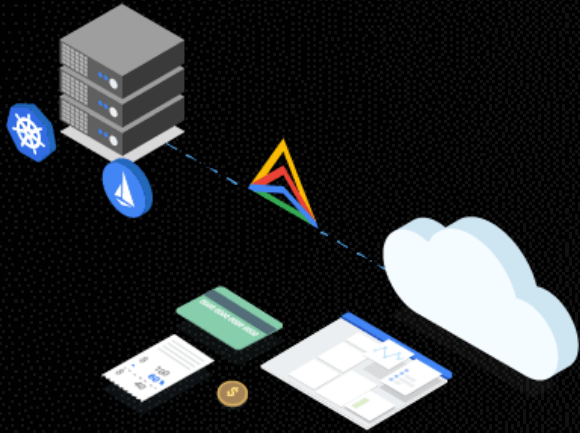
# Life in 2020 – Hybrid cloud - What the heck?



Too many options  
Mix and Match



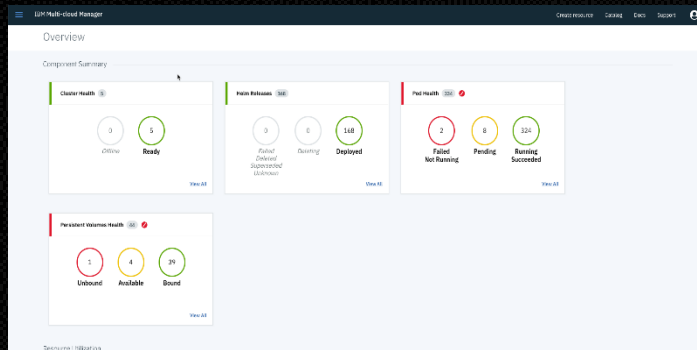
# (Some) Hybrid cloud projects



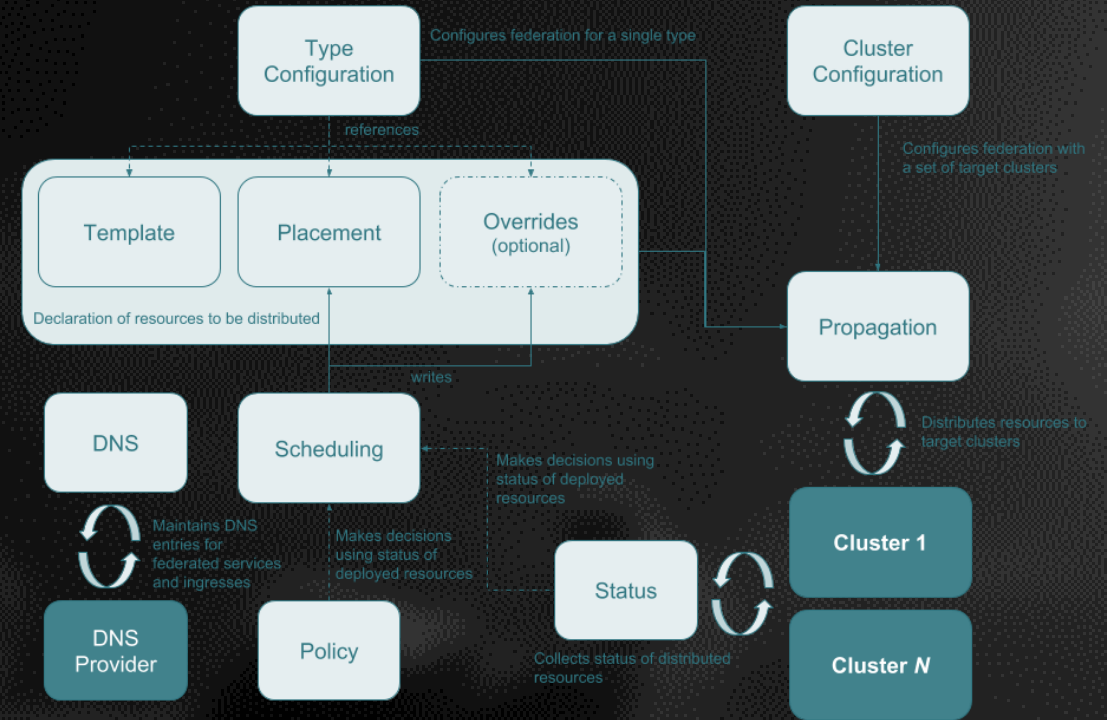
Google - Anthos



AWS - Outposts



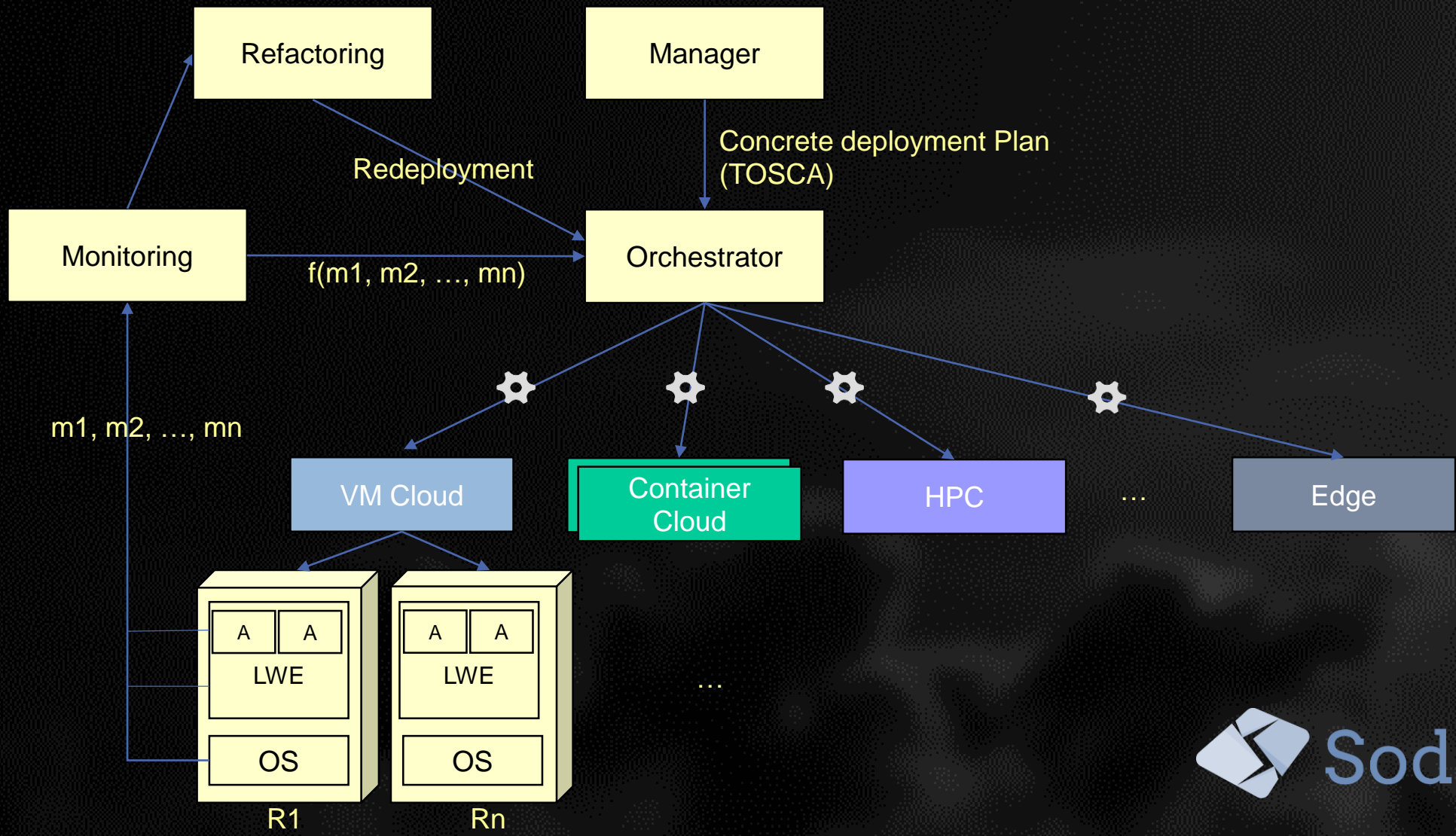
IBM – Multi Cloud Manager



Kubernetes Federation v2

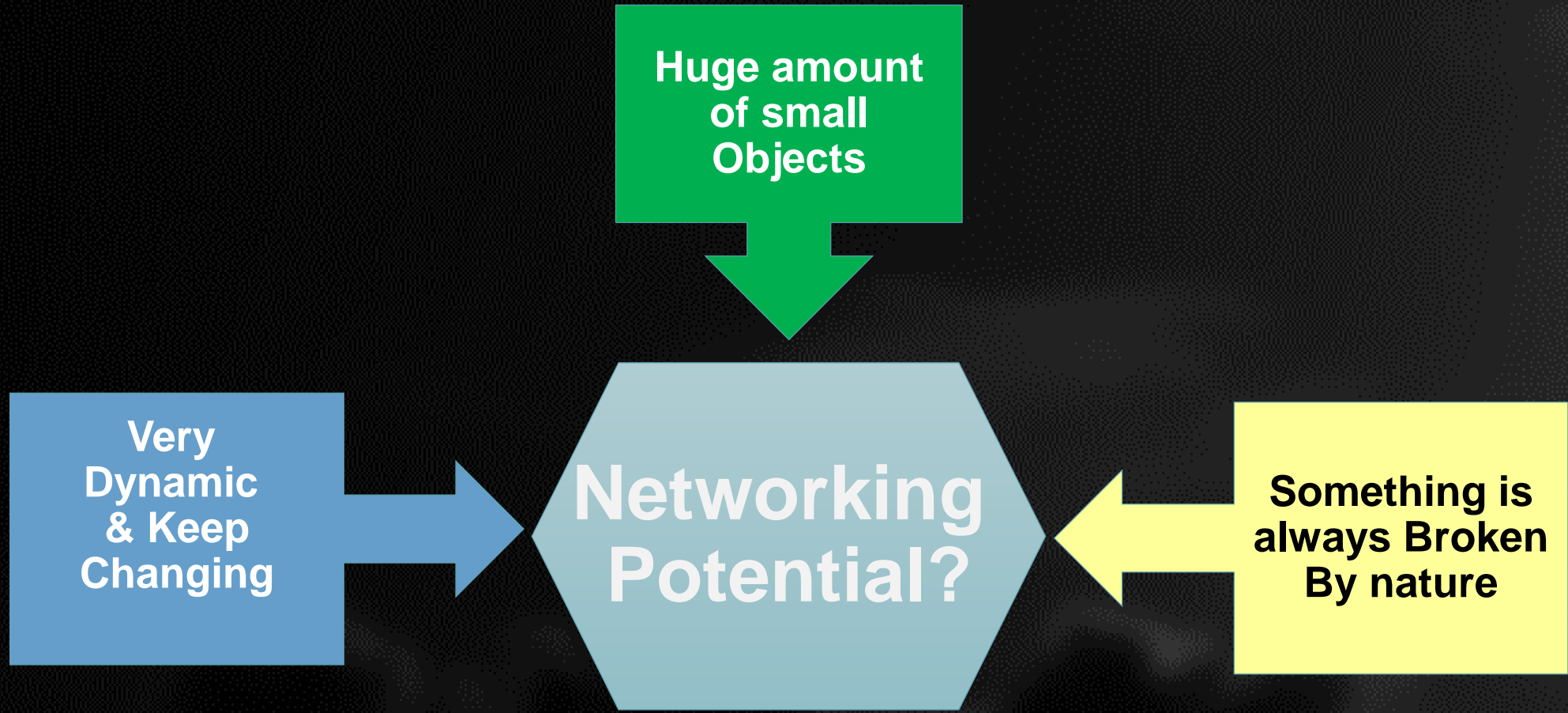


# Hybrid cloud - Sodalite Architecture





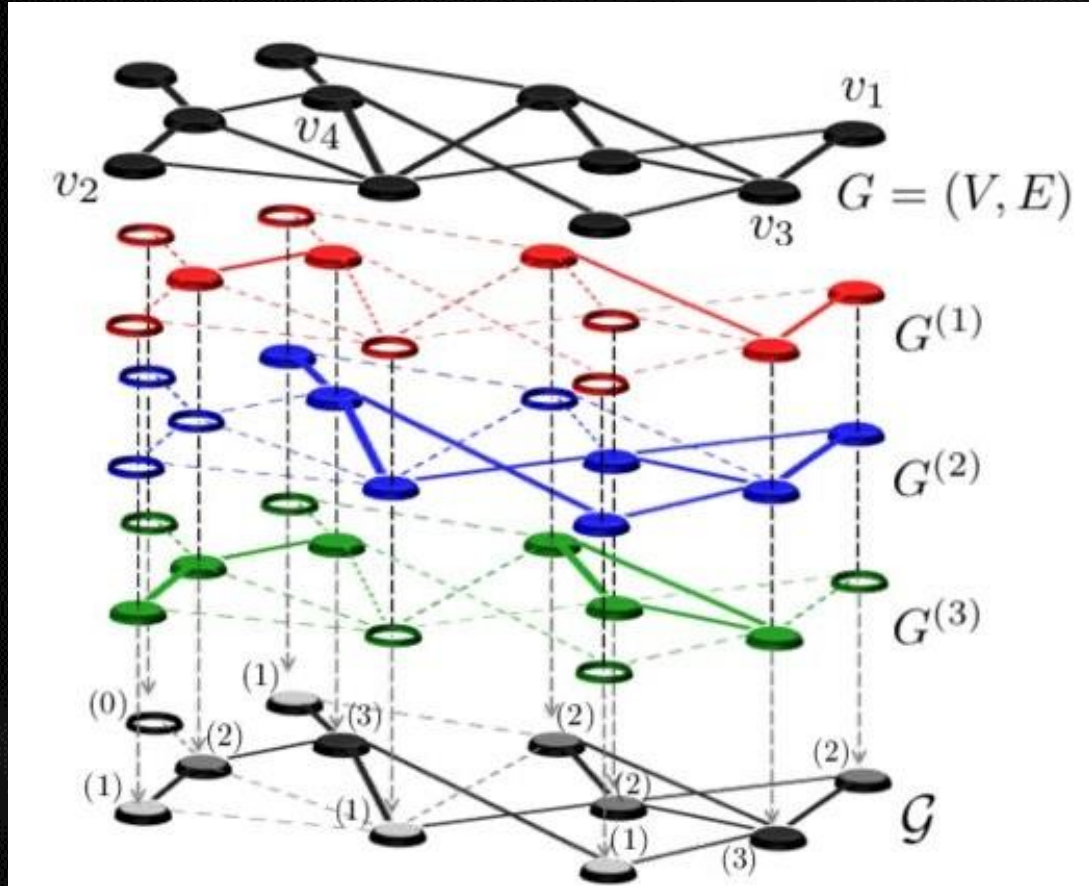
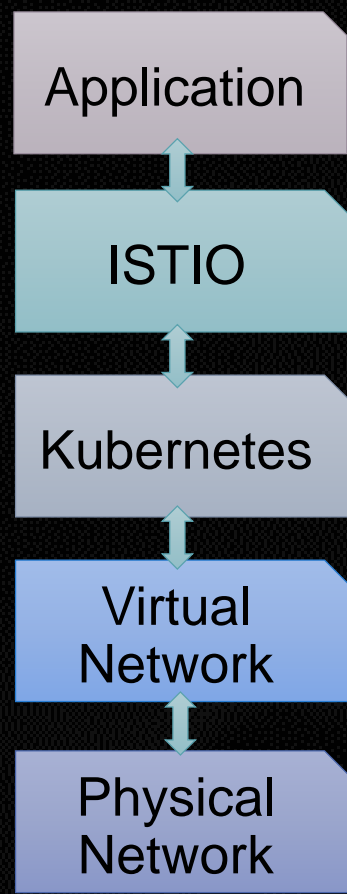
# Hybrid cloud challenges







# Hybrid cloud Networking – cross layers





**Filter**

**Highlight**

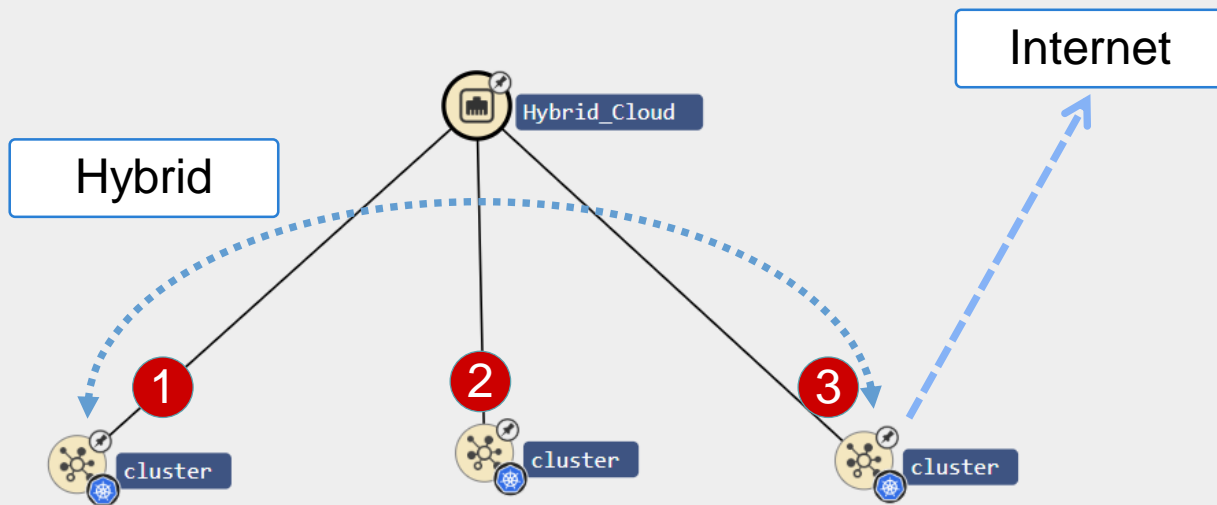
[Captures](#) [Generator](#) [Flows](#) [Alerts](#) [Workflows](#) [Topology rules](#)

Gremlin console

[Create](#) [All](#) [AIBPF](#)

### Metadata

**Name** : Hybrid\_Cloud  
**TID** : 38f95756-8989-59cc-7908-43af2ee988b7  
**Type** : cluster\_connector  
**source** : node-rule



**Topology view**  
 G.V().Has('Type',Regex('cluster|cluster\_conn...  
 Live



# Hybrid cloud Networking – Skydive



CogNETive powered by Skydive \_2019-06-02T14:54:05

[Preferences](#) [Documentation](#) [Status](#) [Logout](#)

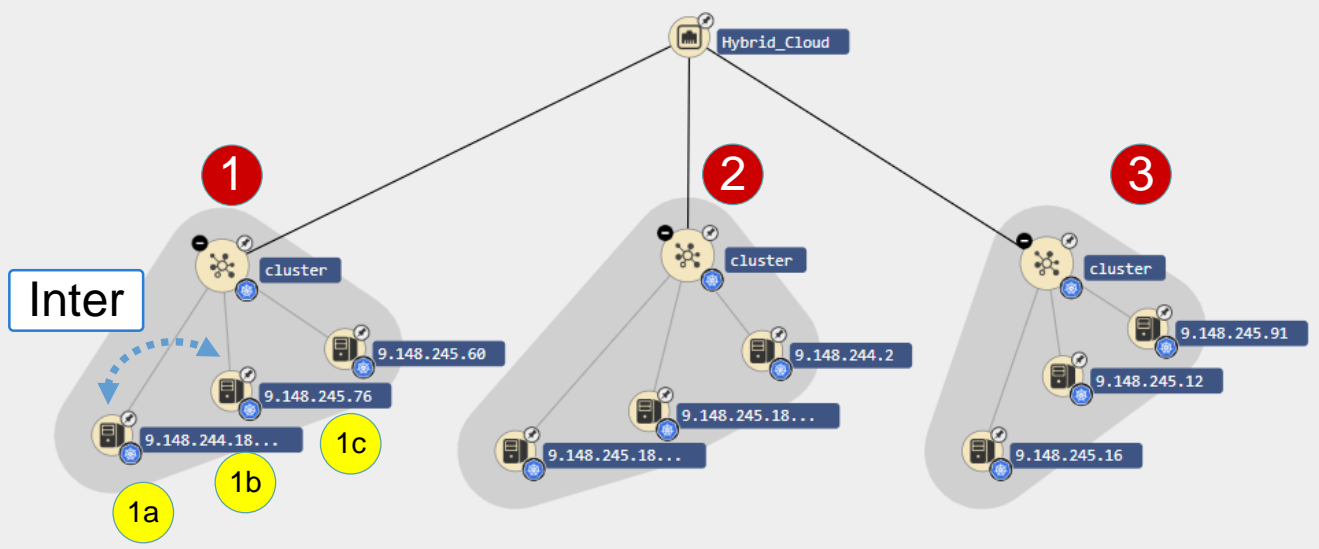
**Filter**

**Highlight**

[Captures](#) [Generator](#) [Flows](#) [Alerts](#) [Workflows](#) [Topology rules](#)

Gremlin console

[Create](#) [All](#) [AIBPF](#)



**Metadata**

**Name** : Hybrid\_Cloud  
**TID** : 38f95756-8989-59cc-7908-43af2ee988b7  
**Type** : cluster\_connector  
**source** : node-rule

Search, Filter, Full Screen, Zoom In, Zoom Out, Refresh, Live

**Topology view**  
 G.V().Has('Type',Regex('node|cluster|cluster\_...  
 Live



# Hybrid cloud Networking – Skydive



CogNETive powered by Skydive \_2019-06-02T14:54:05

Preferences Documentation Status Logout

Captures Generator Flows Alerts Workflows Topology rules

Gremlin console

Create All AIBPF

**Metadata**

K8s :

Manager : k8s

Name : cluster

Type : cluster

**Topology view**  
G.V().Has("Type",Regexp("host|node|cluster|clu...  
Live



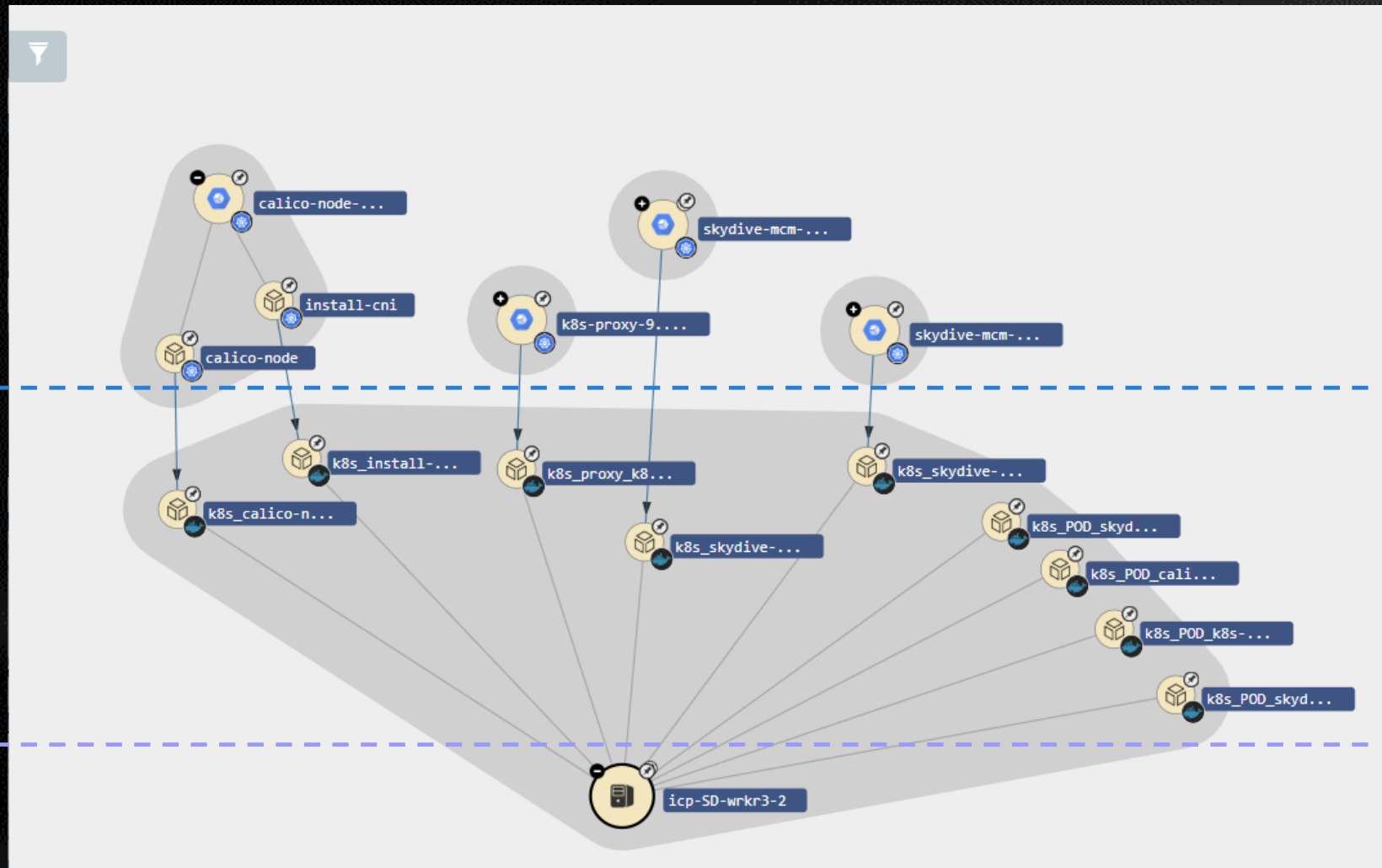
# Hybrid cloud Networking – Skydive



Kubernetes

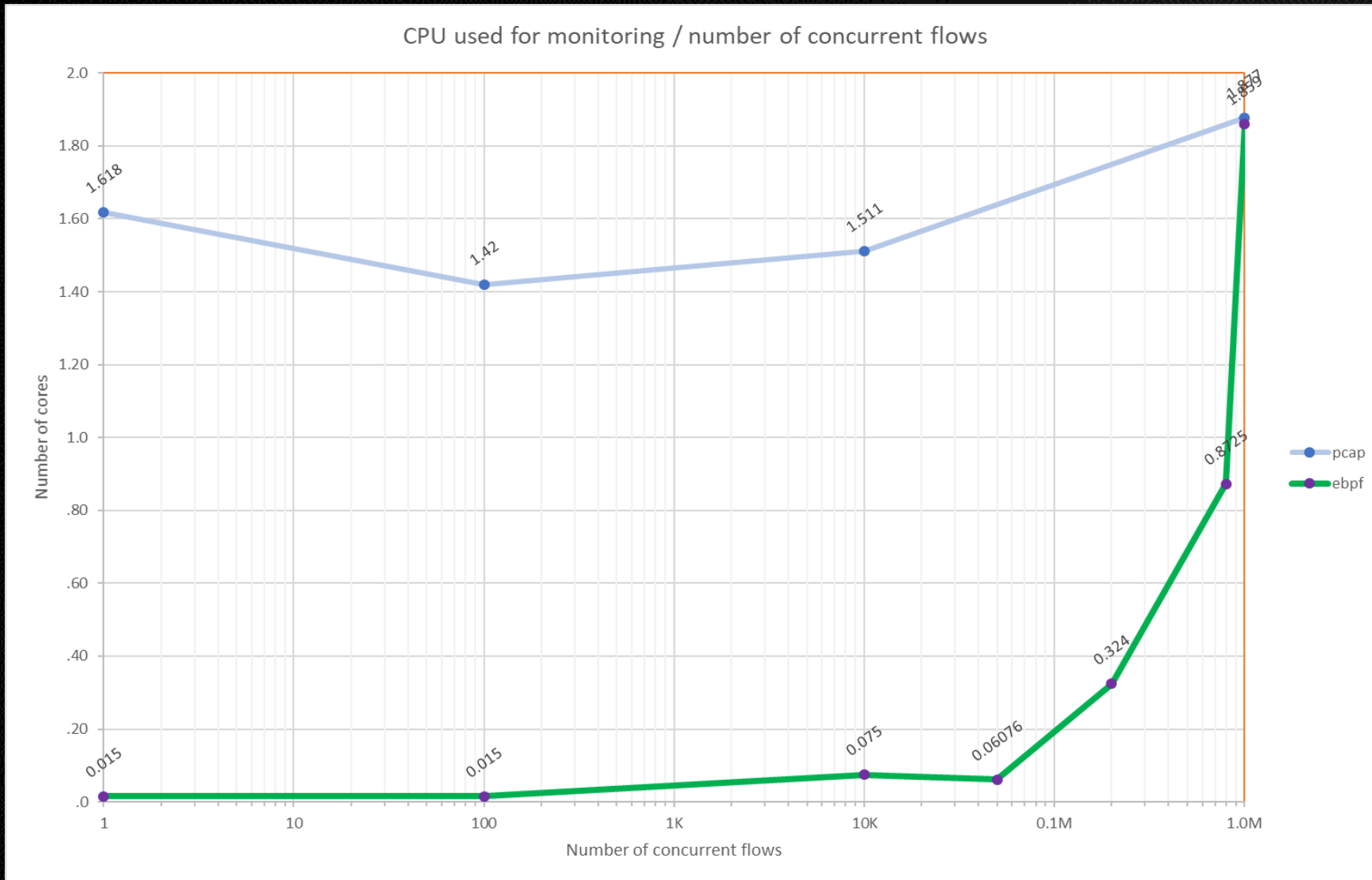
Virtualization (Docker)

Physical (Linux)



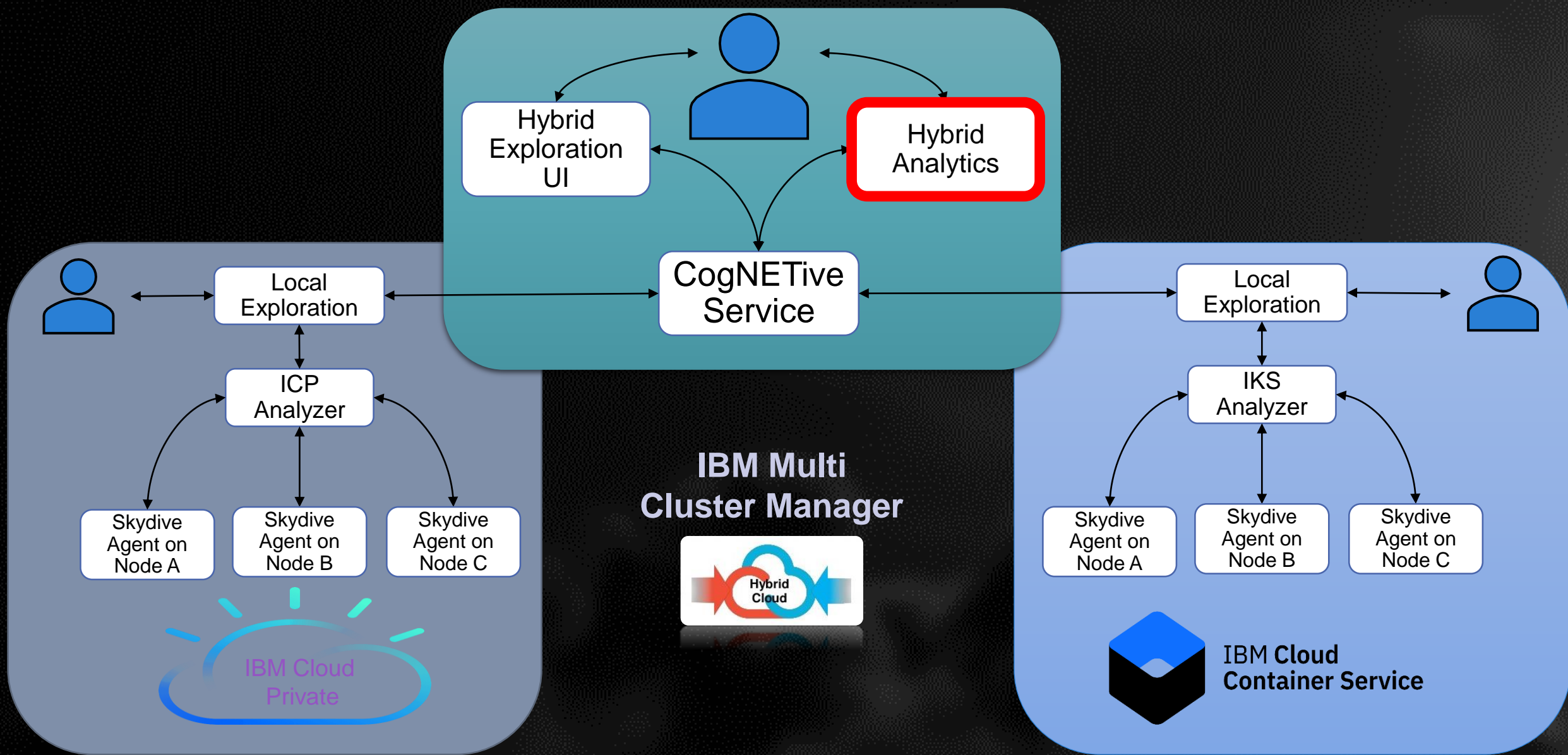


# Hybrid cloud Networking – eBPF Monitoring Overhead



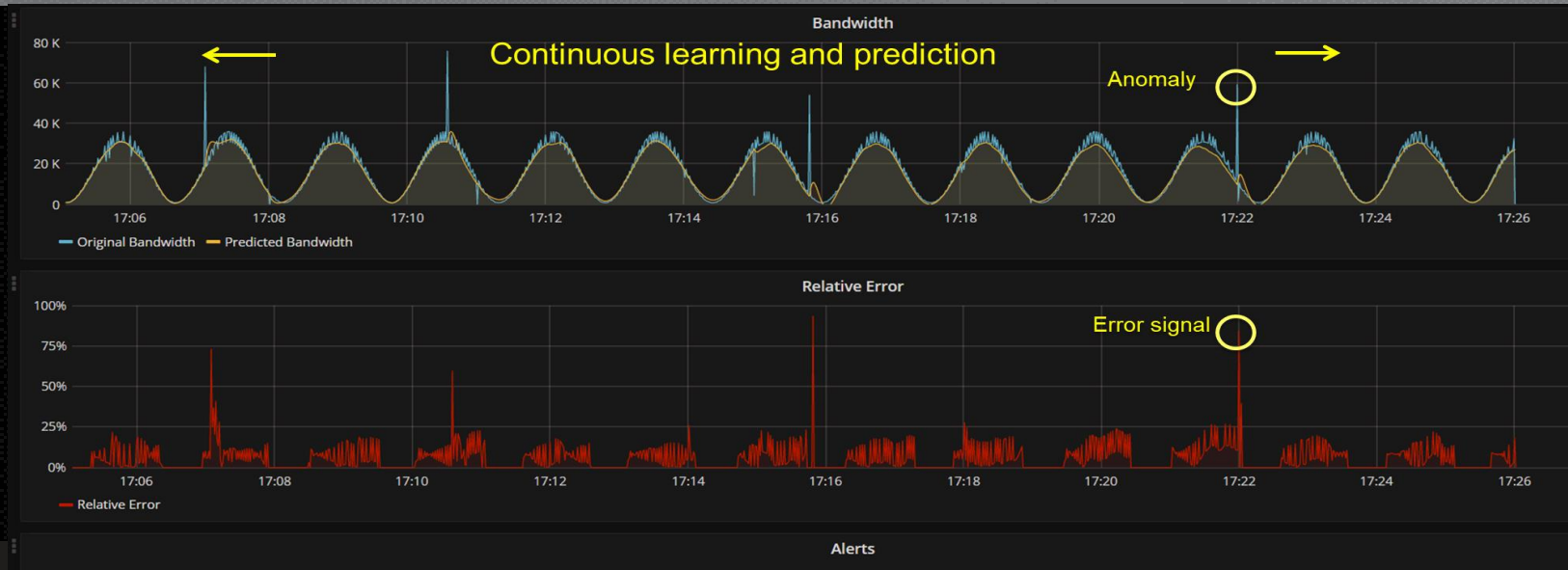


# Hybrid cloud Networking AIOPS

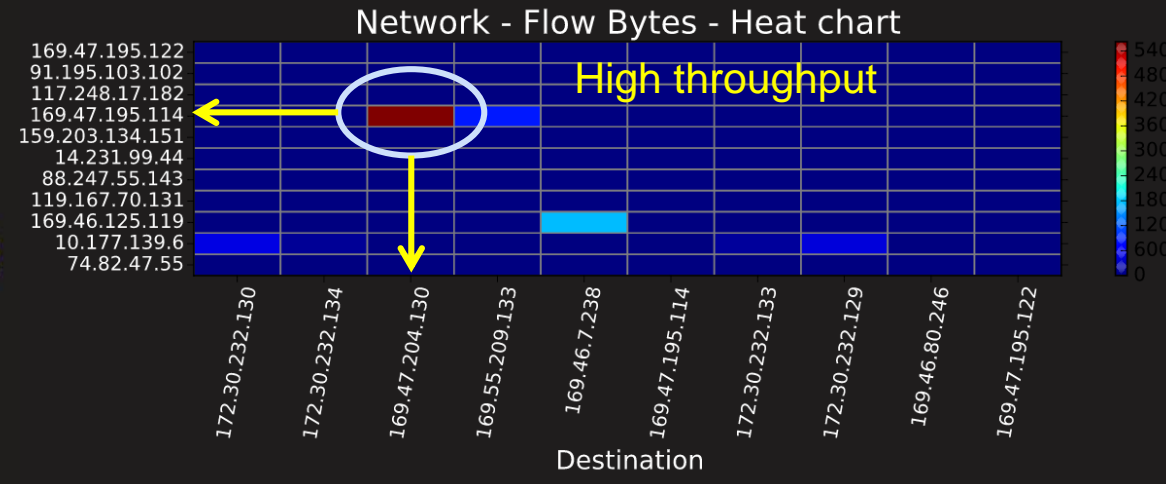
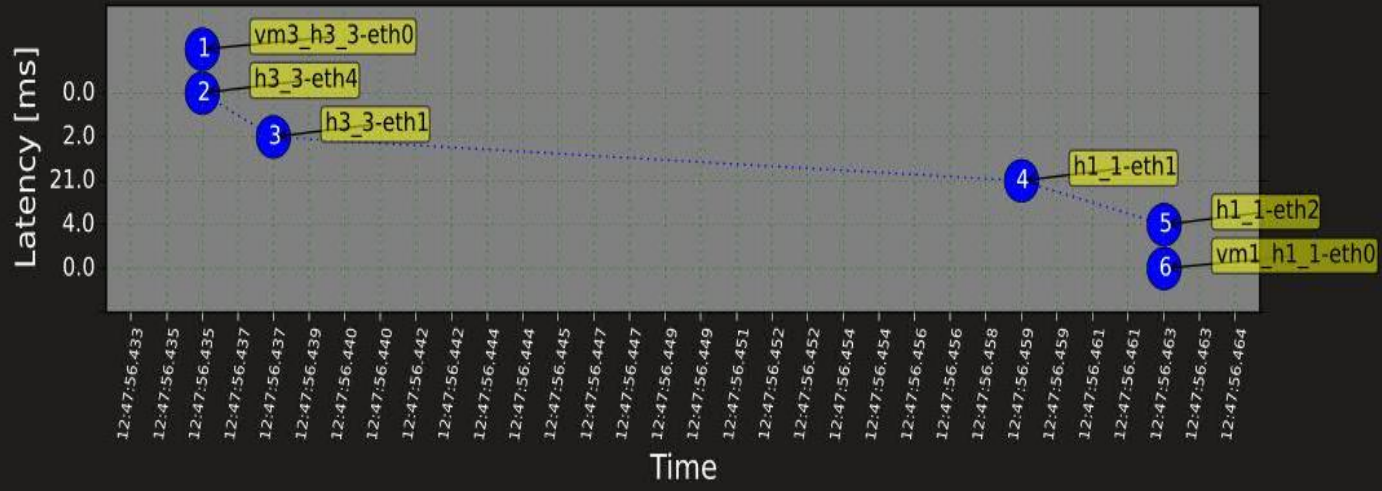




# Hybrid cloud Networking - Analytics



FPV for flow 10.0.0.27:47505==>10.0.0.1:80







<https://12factor.net/>

## THE TWELVE FACTORS

### **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**

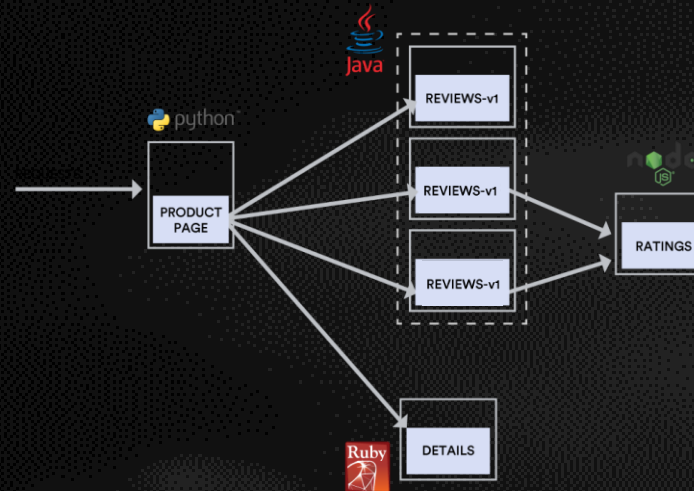
Run admin/management tasks as one-off processes



# Cloud Native Applications



- Loosely couple microservices
- Modelled around business capabilities
- Scalability
  - Build for failure
- Automation
- Hiding implementation
- Independently deployable
- Isolation



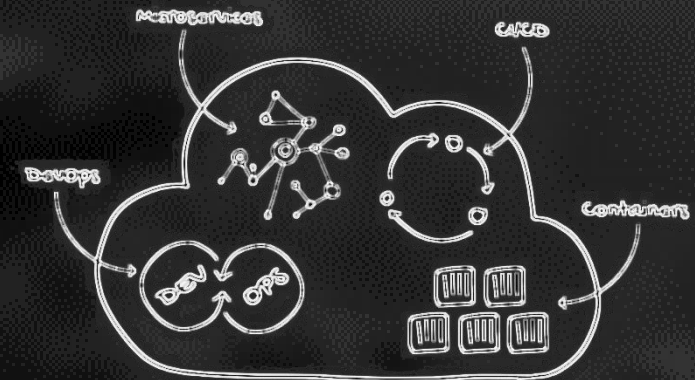
<https://12factor.net/>

**THE TWELVE FACTORS**

- 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



<https://kubernetes.io/>





- All cloud native application requirements still apply + :
- Latency awareness (e.g. cross vendors, storage)
- Operationally efficient
  - Same functionality from different solutions
- SREs/ DevOps are the “new developers”
- APIs and Industry standards are equal citizens



Next.Gen clouds will be Hybrid. Based on mixture of resources from multiple vendors public and on-prem private. Controlled by single federated orchestrator

## Kubernetes Federation

<https://github.com/kubernetes-sigs/kubefed>

## IBM Multicloud Manager

[https://www.ibm.com/support/knowledgecenter/en/SSBS6K\\_3.1.0/mcm/getting\\_started/introduction.html](https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.1.0/mcm/getting_started/introduction.html)

## Google Anthos

<https://cloud.google.com/anthos/>

## AWS Outposts

<https://aws.amazon.com/outposts/>





Thank  
you!!