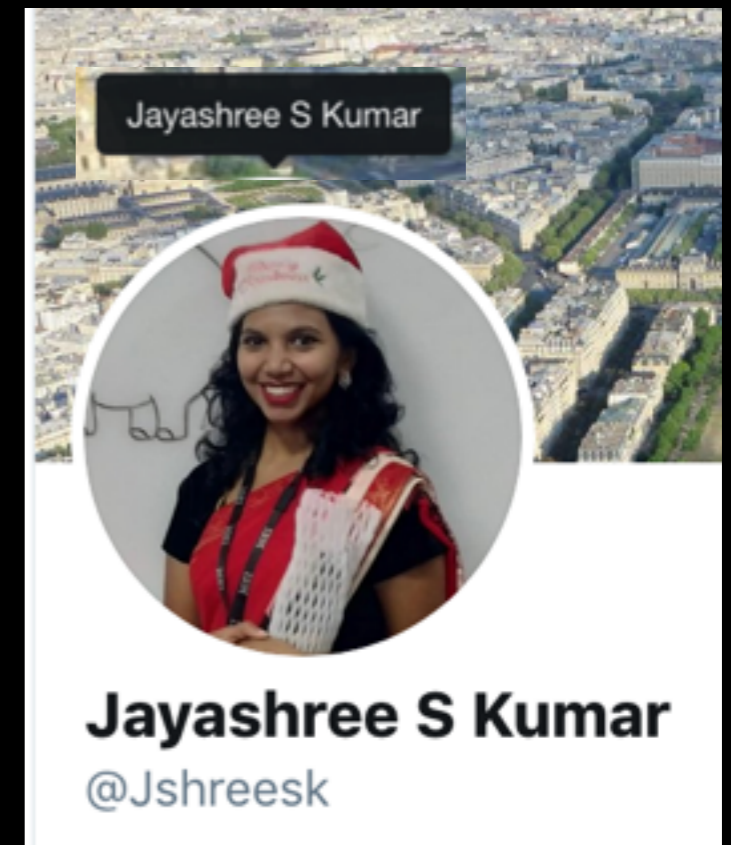


Complete Idiot's guide to Cloud Native

Jayashree S Kumar, IBM

About Me

- IBM-JAVA's Classes-Library developer
- Worked Extensively on JDK's Testing
- Evaluator in Invention Development Team
- Runtimes team @ IBM Software Labs



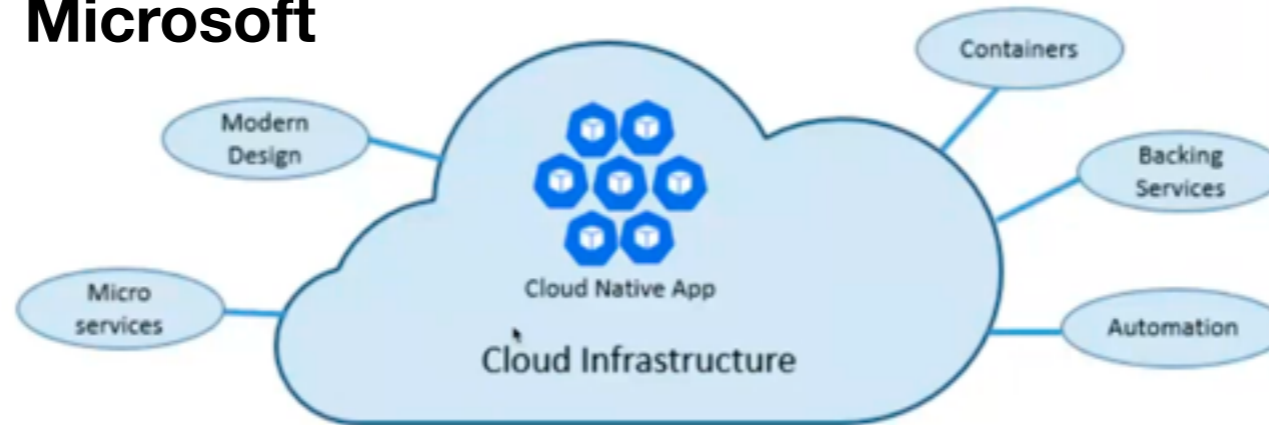
Agenda

- **What ? CLOUD-NATIVE**
- **Discuss (the very many) Terminologies**
- **Conclusion**

WHAT ?

varied Definitions..





Microsoft



VMWare



RedHat

 DevOps DevOps is an approach to culture, automation, and platform design intended to deliver increased business value and responsiveness. Understanding DevOps →	 Microservices A microservices architecture breaks apps down into their smallest components, independent from each other. Understanding microservices →
 APIs An application programming interface (API) is a set of tools, definitions, and protocols for building application software. They connect products and services without having to know how they're implemented. Understanding APIs →	 Containers Containers allow apps to be packaged and isolated with their entire runtime environment, making it easy to move them between environments while retaining full functionality. Understanding containers →

Build & Run- **Scalable** Applications in **Modern** & Dynamic environments- Public, Private, and Hybrid clouds.

It also lots more, impacting the design, implementation, deployment and operation of
your application.

CNCF



550 members...

Checkout @ <https://www.cncf.io/>

- Cloud Native Interactive Landscape
- Cloud Native Trail Map

Cloud Native Trail Map

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.



9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.



2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLOps



4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general-purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.



8. STREAMING & MESSAGING

When you need higher performance than JSON-Rest, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.



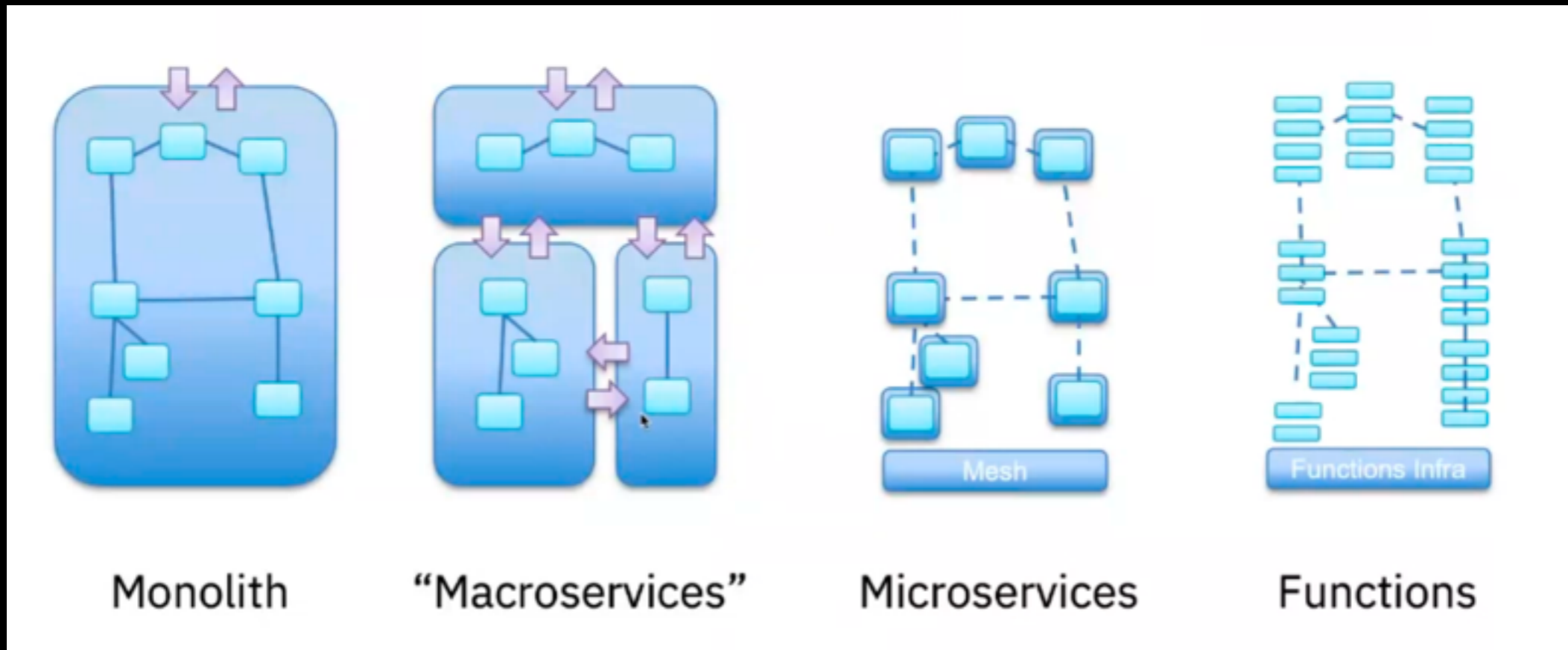
10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.



INFRASTRUCTURE

Architecture Styles



Browed: <https://techtv.bemyapp.com/#/conference/5f84cc4e9b8d68001b2cf93a>

Cloud Native Application



MicroServices only

Eg Cloud-native Application



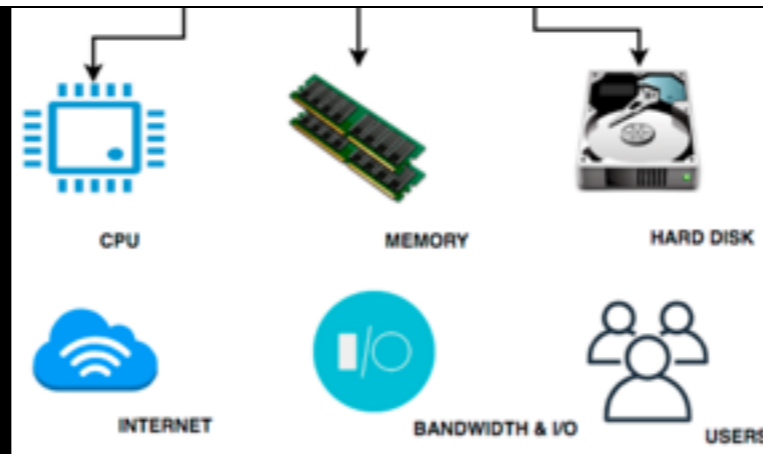
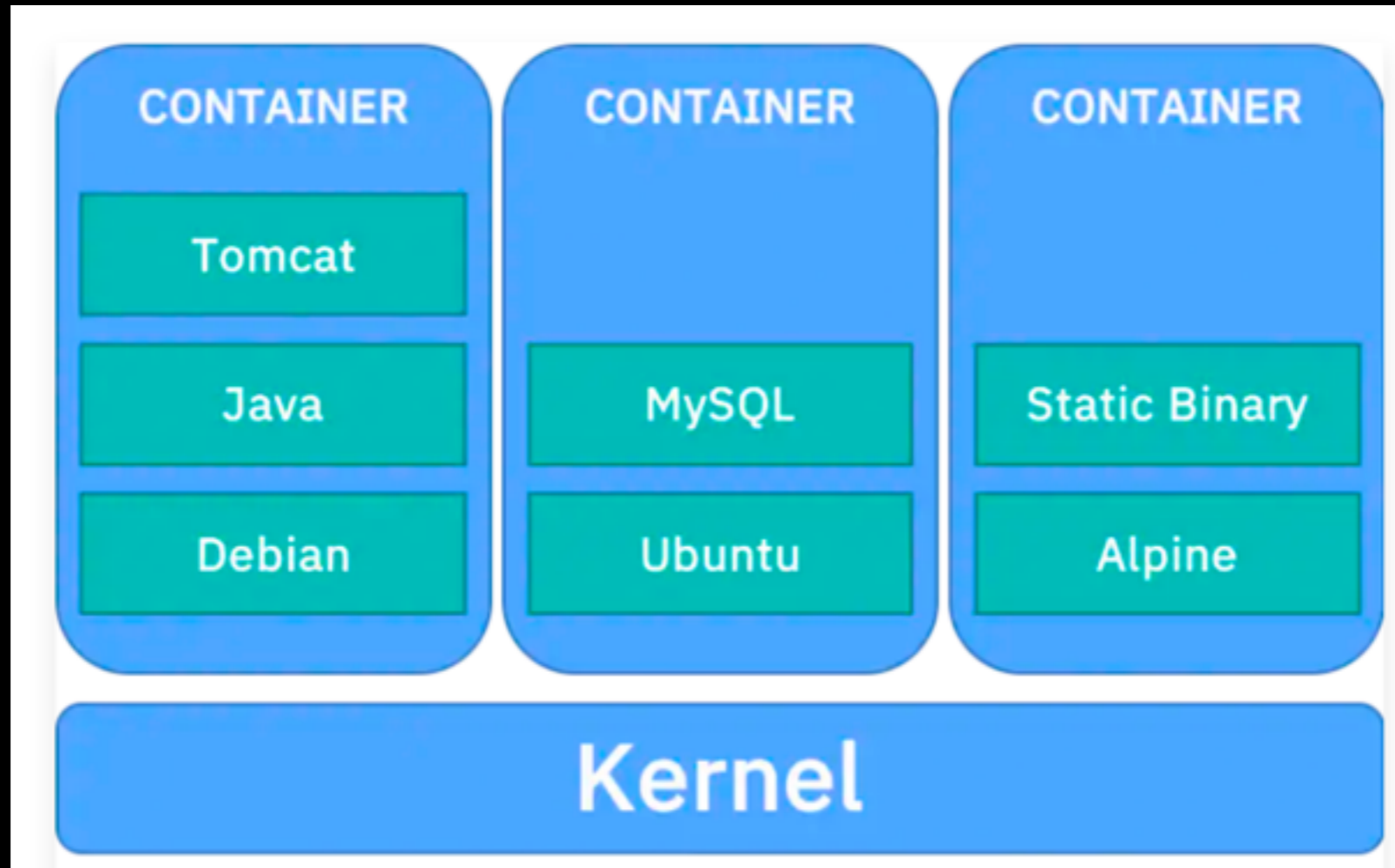
Containerization

“It was working fine in my system”

Application

Runtime

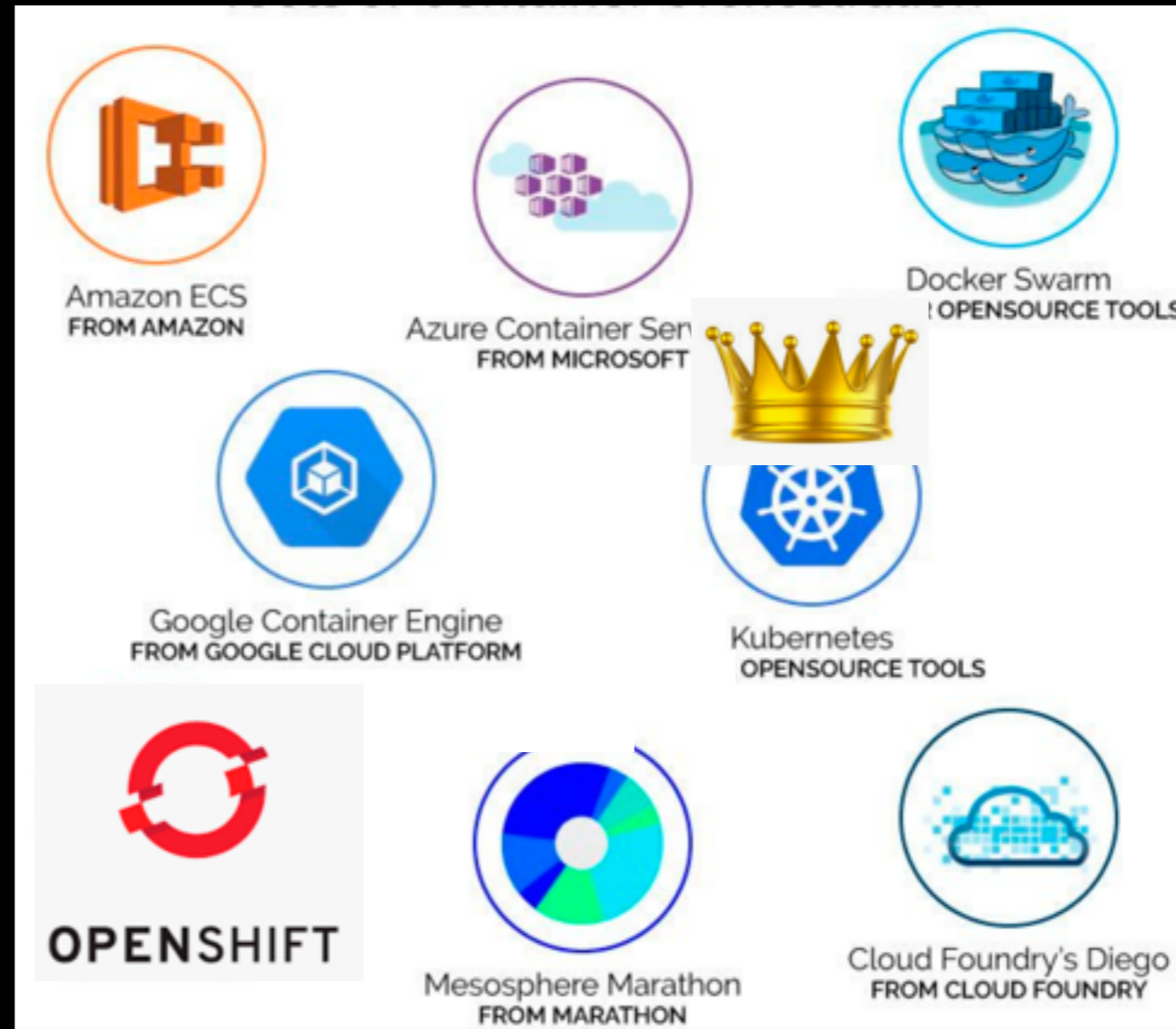
OS



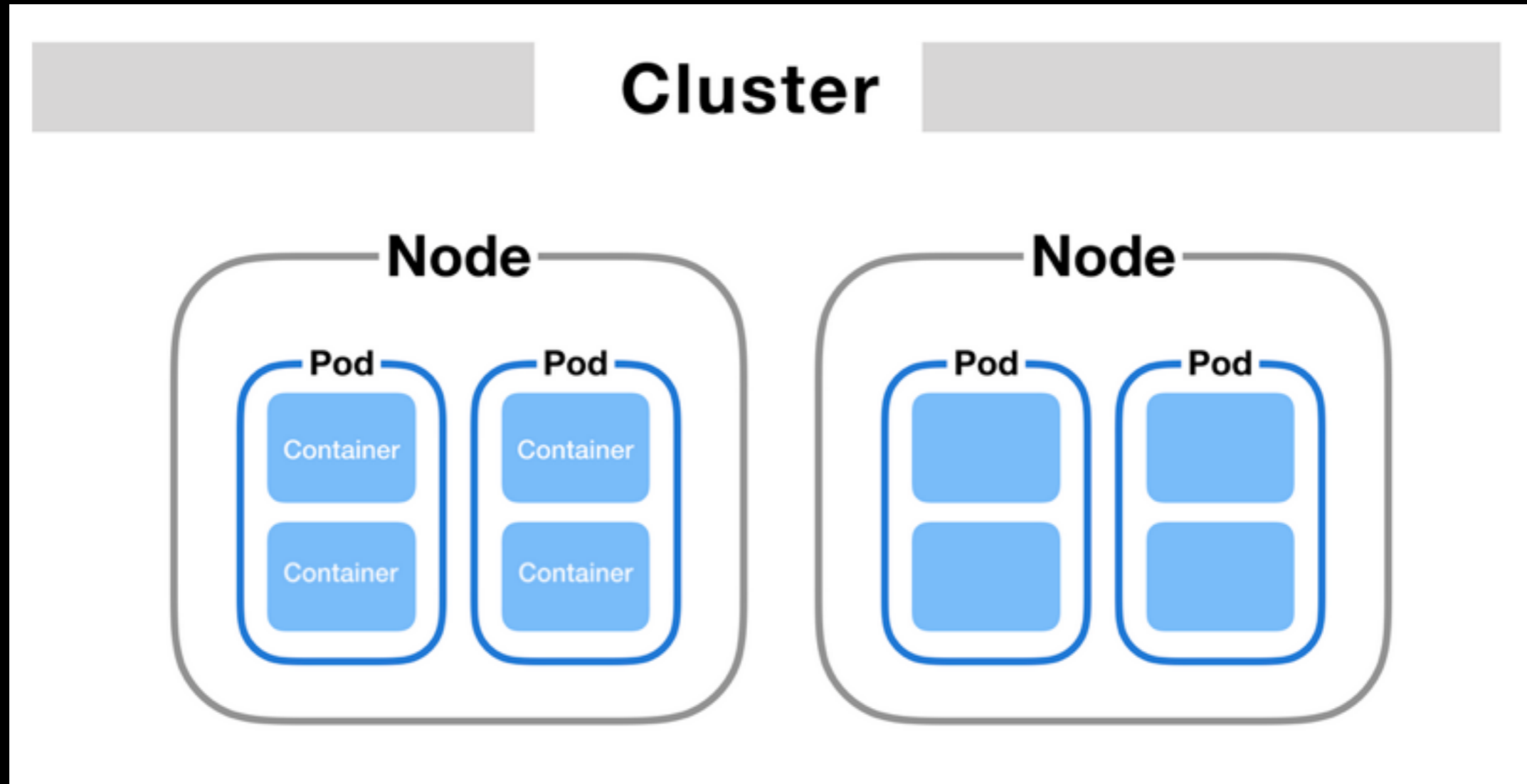
Container Orchestration

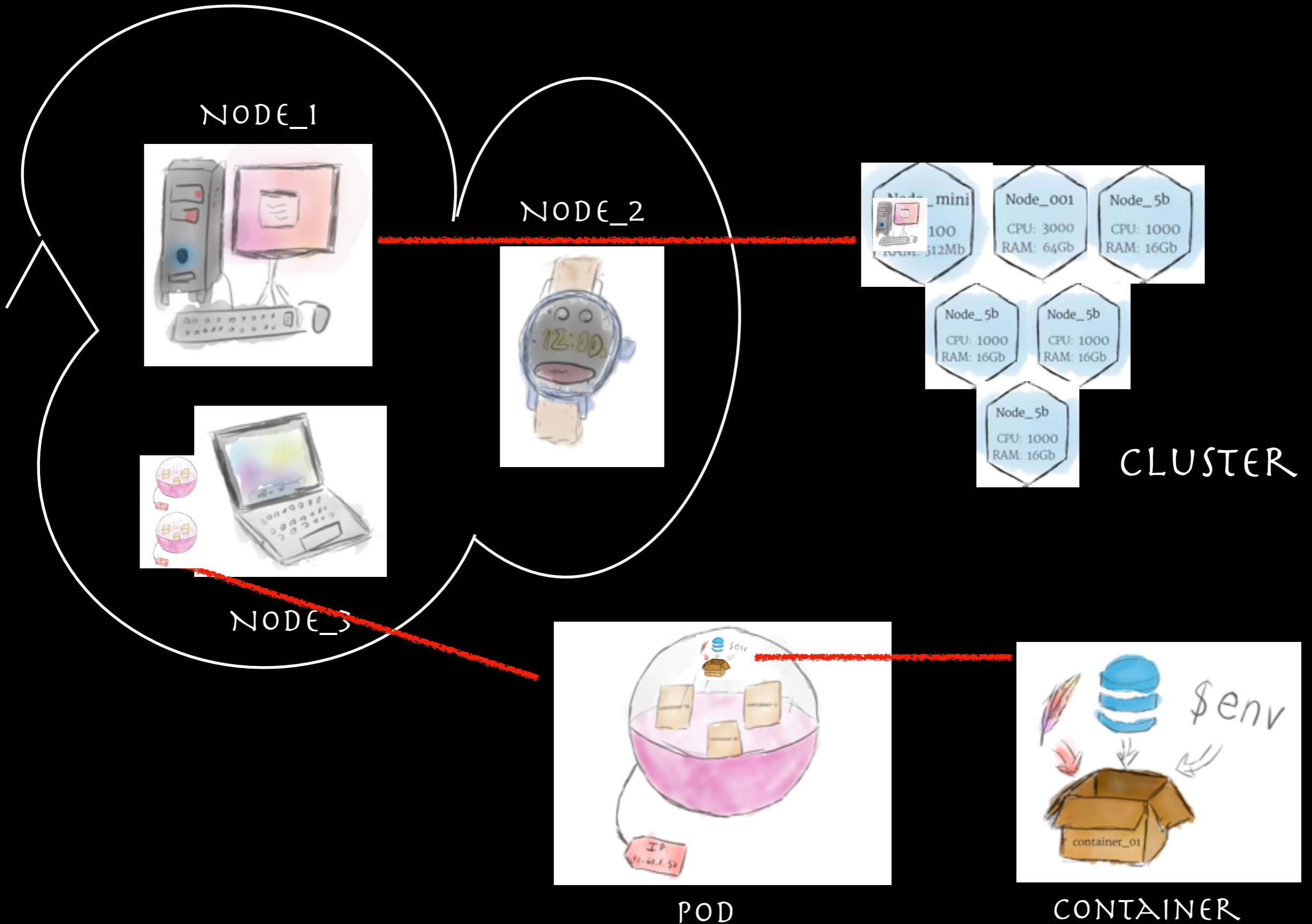
Automate:

- Configuration
- Provisioning
- Availability
- Scaling
- Resource allocation
- Load balancing
- Health monitoring



Cluster. Node. Pod





Service Mesh

Mark Runyon Analogies..



The goal of the mesh is to guarantee secure communications between each independent container application/MS and be able to redirect traffic in the event of failures.



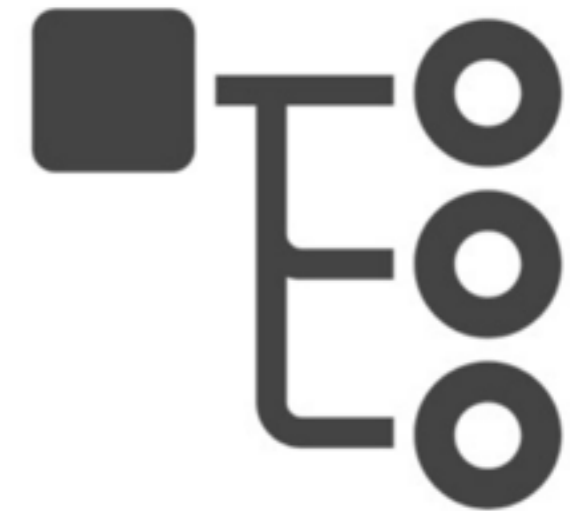
Observability



Metrics



Logs



Traces

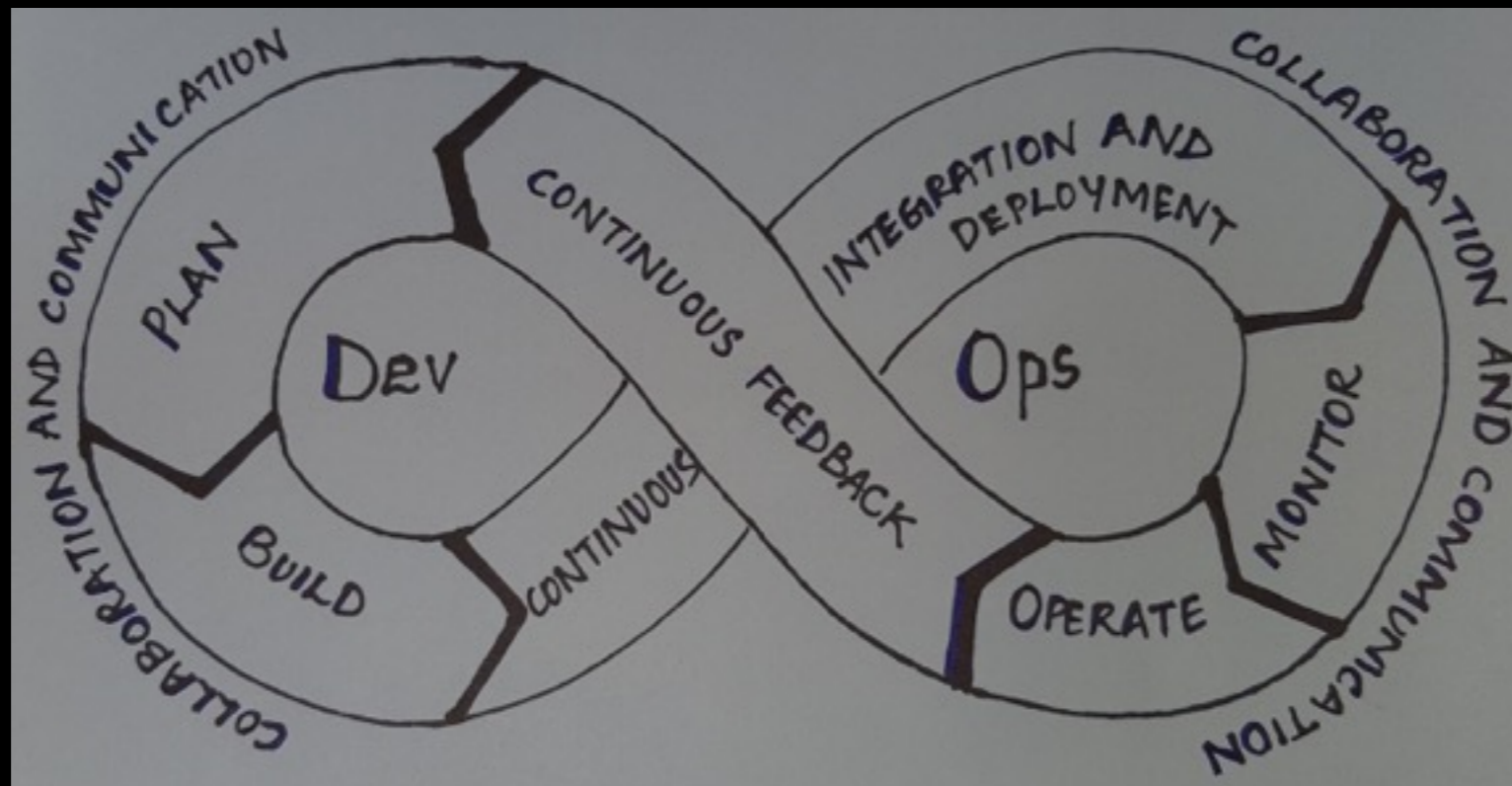


PROCESS

CI/CD : Continuous Integration & Delivery



DevOps : Development & Operations



PLAN



CODE



BUILD



TEST



RELEASE



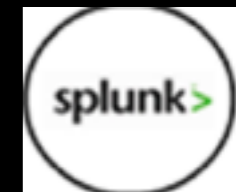
DEPLOY



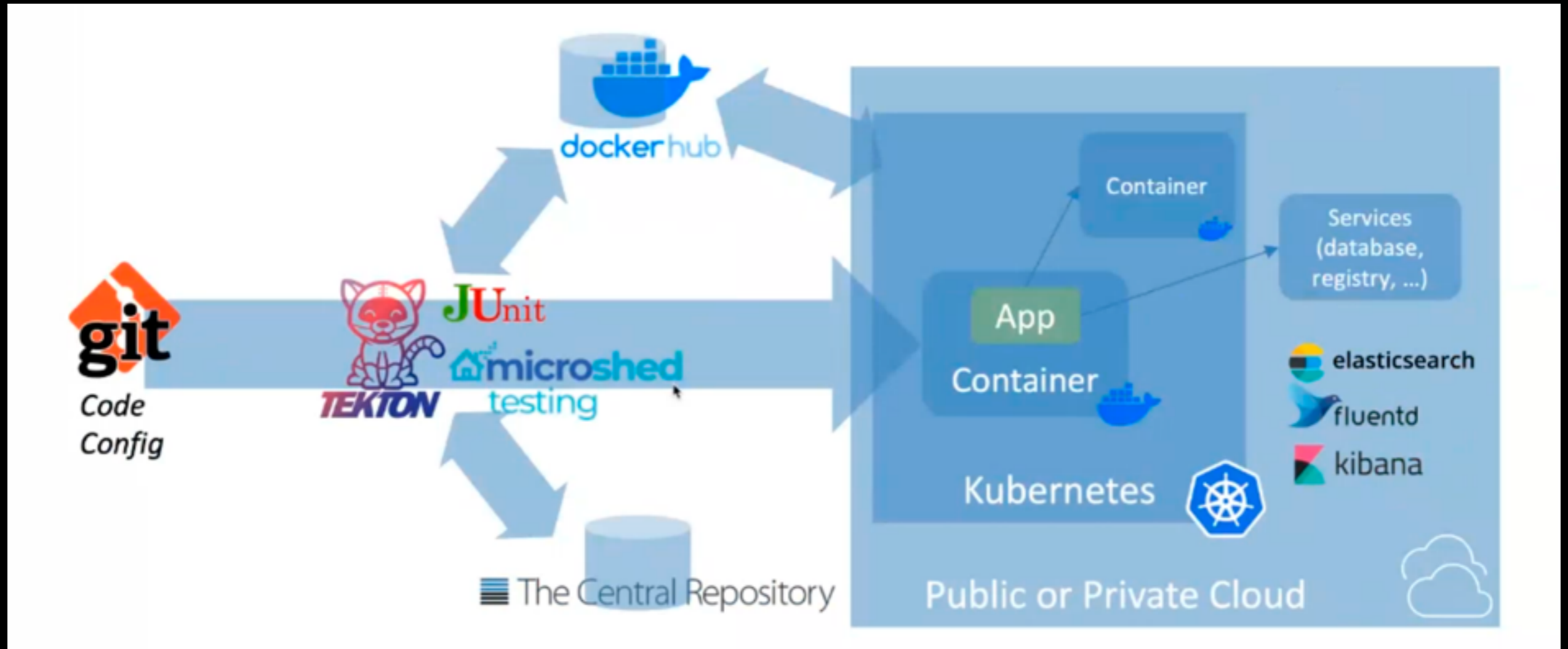
OPERATE



MONITOR

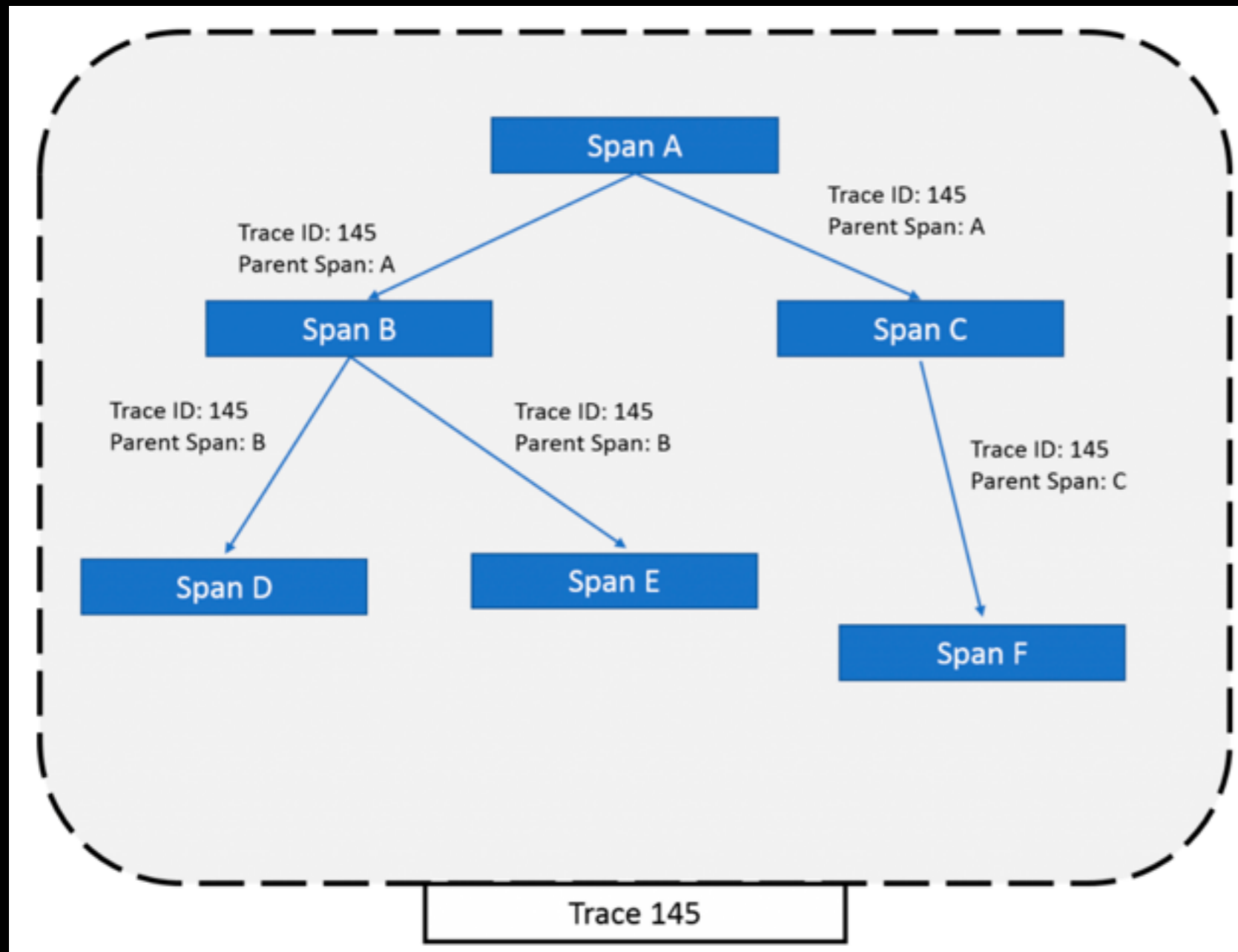


12-factor

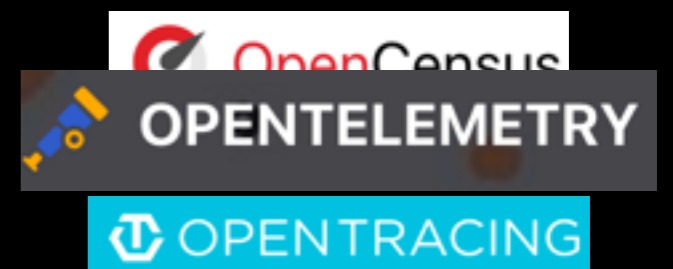


APP DESIGN

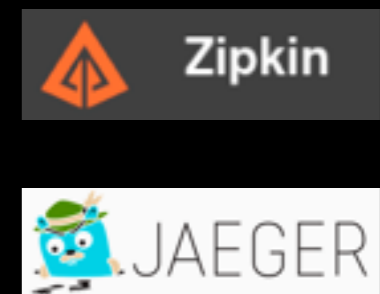
Distributed Tracing



Framework:



TOOLS:



Borrowed: <https://www.scalyr.com/blog/distributed-tracing-important-2019/>

Asynchronous Programming

Synchronous



Asynchronous



synchronous, single thread of control



synchronous, two threads of control



asynchronous



Borrowed: https://eloquentjavascript.net/11_async.html

Reactive Programming

`a = 10;`

`b = a + 1;`

`a = 11;`

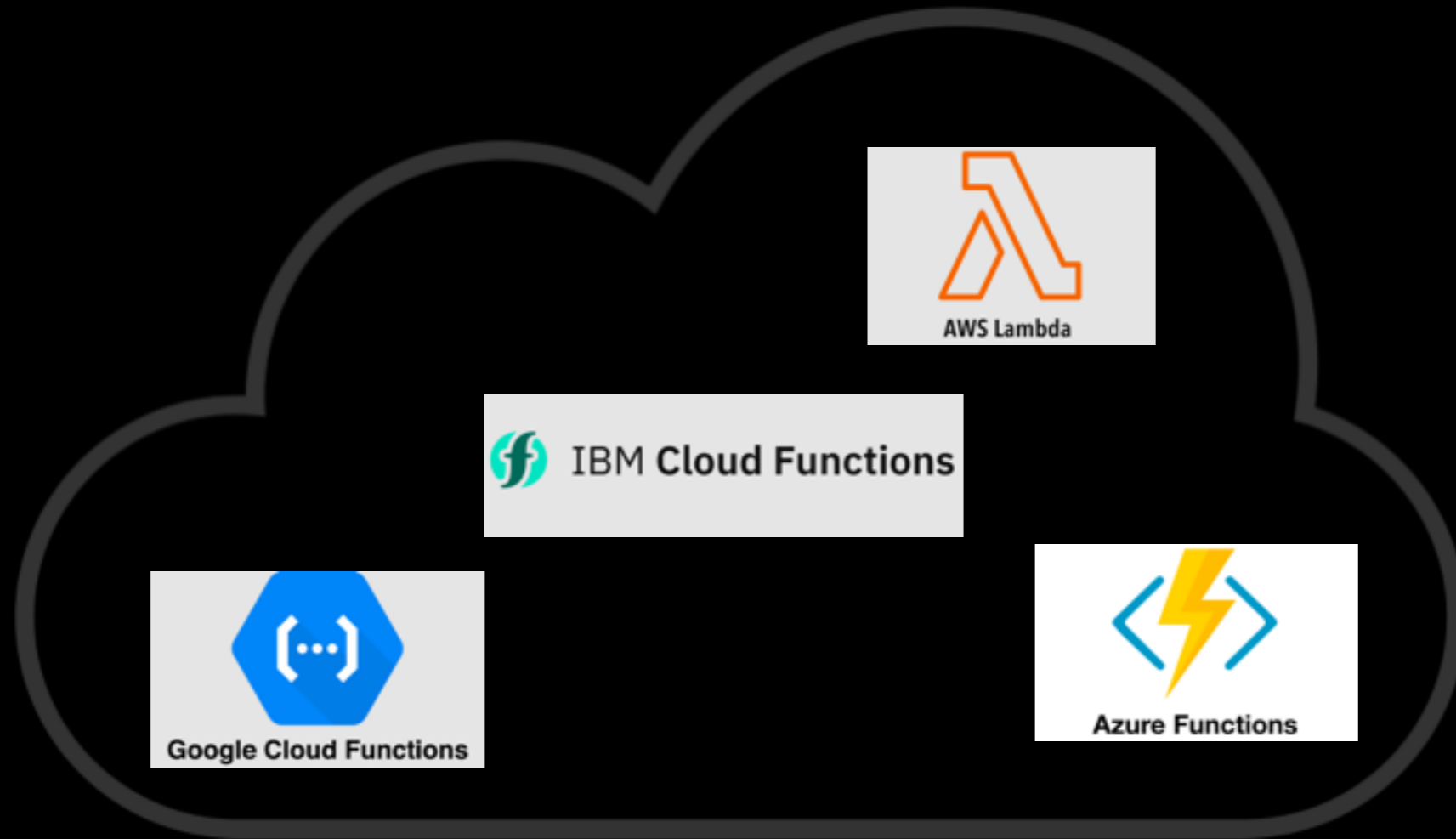
(Imperative $b = 11$) (Reactive $b = 12$)

$b = a + 1;$

$b \leq a + 1;$

Processing **Infinite streams of data** in an Asynchronous and Non-blocking manner.

Serverless



Dependancy Injection (IoC subtype)

```
class Company{  
    Employee employee;  
    Company(){  
        employee = newEmployee("111", "ABC");  
    }  
}
```

Constructor Injection

```
class Company{
    Employee employee;
    Company(Employee employee){
        this.employee = employee;
    }
}
```

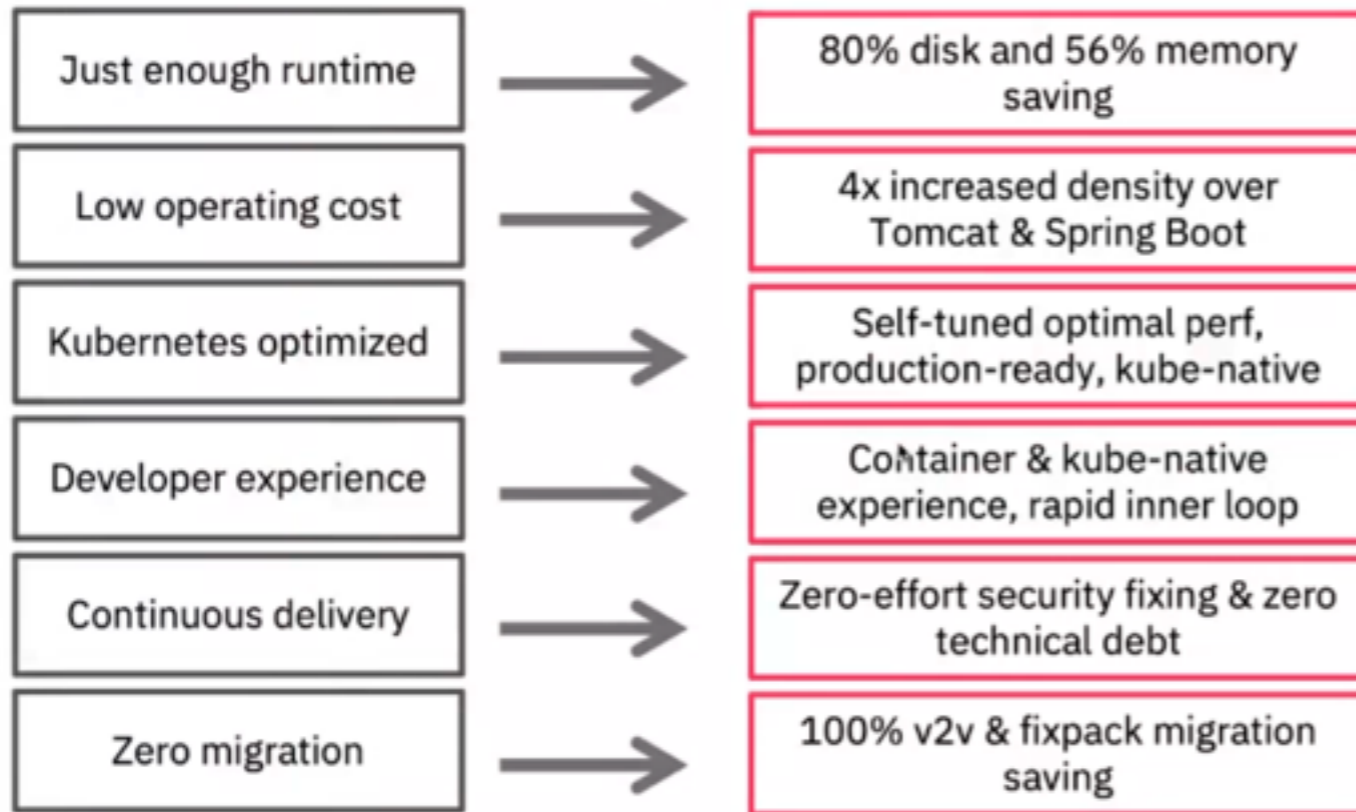
Setter Injection

```
class Company{
    Employee employee;
    Public void setEmployee(Employee employee){
        this.employee = employee;
    }
}
```

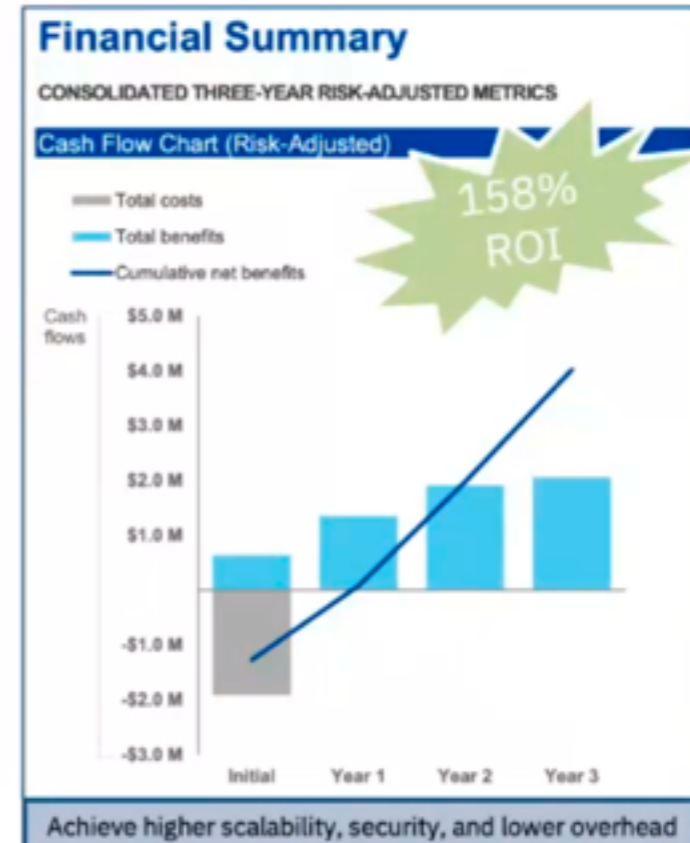
Method Injection

JAVA FRAMEWORK (SOME)

Open Liberty



<https://openliberty.io>



30

Others... Microservices



Quarkus

SUBATOMIC (10x)



Wait.
So you just save it,
And your code is running?
And it's Java?!



SUPERSONIC (100x)

I know, right?
SUPERSONIC JAVA, FTW!



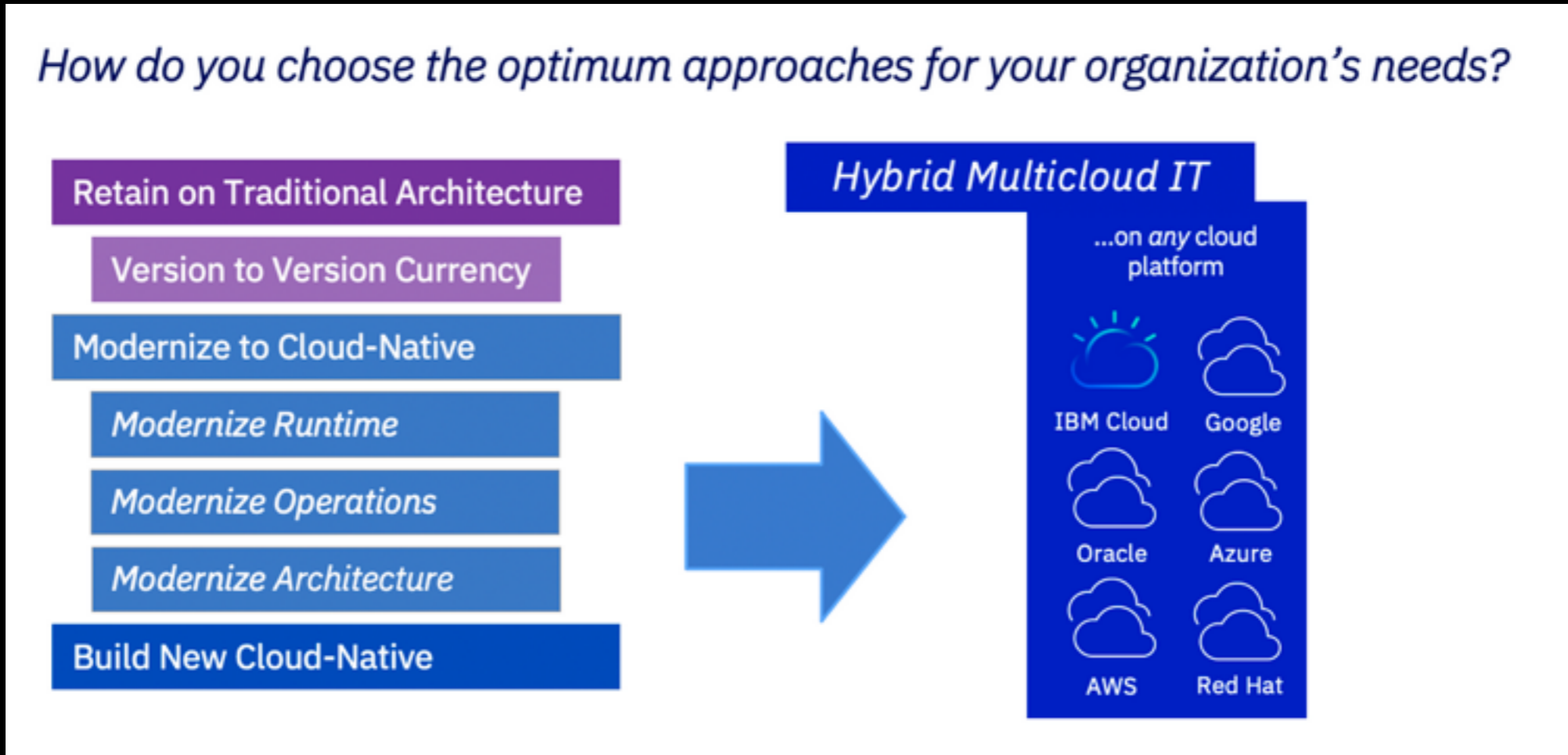
IMPERATIVE + **REACTIVE**



Conclusion

- Cloud-native is a Journey - here to stay.
- Can choose to Modernise Application
- Organisation's needs first - before fashion.
- Start by Knowing-It-All

IBM's Offers..



You Can Refer:

1. <https://developer.ibm.com/solutions/application-modernization/articles/modernize-your-valuable-java-applications/>
2. <https://ibm-cloud-architecture.github.io/modernization-playbook/getting-started/>

THANK U!

to
each
his
own