

</ Kubernetes na Prática! >

Bluetalks

</ Gerson ITIRO Hidaka >

IBM Certified Architect - Expert Level



youtube.com/user/itirohidaka



itirohidaka



@itirohidaka



itirohidaka



+55 (11) 9.9956-1887



itiro@br.ibm.com

Agenda

| Horário | Tempo | Agenda | Tipo | Palestrante |
|---------------|------------|---|----------|--------------|
| 19:00 - 19:15 | 15 minutos | Boas Vindas e Criação do Cluster | Lab 0 | Itiro |
| 19:15 - 19:30 | 15 minutos | Red Hat e os Containers | Palestra | Gabriel Erom |
| 19:30 - 19:45 | 15 minutos | Conceitos Rápidos sobre Kubernetes | Palestra | Itiro |
| 19:45 - 20:15 | 30 minutos | Lab 1 - Criando o Primeiro Aplicativo no K8s | Lab 1 | Itiro |
| 20:15 - 20:45 | 30 minutos | Lab 2 - Brincando com a escalabilidade do K8s | Lab 2 | Itiro |
| 20:45 - 21:15 | 30 minutos | Lab 3 - Deploy de aplicações Multi-Tier | Lab 3 | Itiro |
| 21:15 - 21:45 | 30 minutos | Lab 4 - Liveness e Readiness Probe | Lab 4 | Itiro |
| 21:50 | 1 minutos | Fechamento | Palestra | Itiro |
| 22:00 | - | Pizza Time! | - | - |

</ Pré Requisitos >

Lab0: Pré Requisitos

<https://github.com/itirohidaka/kubebr/tree/master/workshop/Lab0>

1. Criação de uma Conta na IBM Cloud.
2. Instalação do IBM Cloud CLI no Laptop (ibmcloud ou bx).
3. Instalação dos Plugins do Container Services e Container Registry no Laptop.
4. Instalação do Kubernetes CLI no Laptop (kubectl).
5. Download do source code para o Laptop.
6. Criação de um Kubernetes Cluster Lite.



</ Introdução aos Containers >

Developer Productivity, Choice, Control, & Consistency

Speed

Portability

Performance & Control



Cloud Functions

“Serverless” / Event Driven Apps



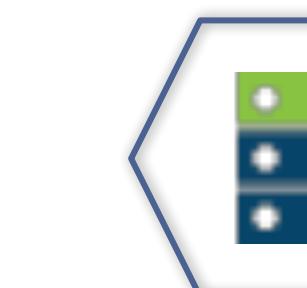
Cloud Foundry

Open PaaS Environment



Container

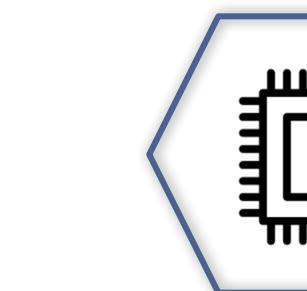
Maximum Portability



Virtual Server

or VMware

Leverage Existing Images & Tools



Bare Metal

Maximum Performance & Control

Language/
Framework

.js .java
liberty

.py .php

.go 

.rb 

express

 spring

Containers

A standard way to package an application and all its dependencies so that it can be moved between environments and run without changes.

Containers work by isolating the differences between applications inside the container so that everything outside the container can be standardized.

</ Container Stack >

Layer 6

**Development Workflow
Opinionated Containers**



Layer 5

**Orchestration/Scheduling
Service Model**



Layer 4

Container Engine



Layer 3

Operating System



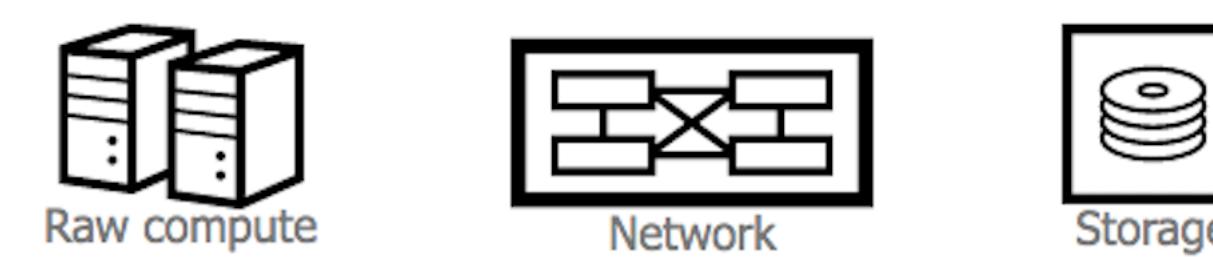
Layer 2

Virtual Infrastructure



Layer 1

Physical Infrastructure



Container Orchestration

Allows users to define how to coordinate the containers in the cloud when the multi-container packaged application is deployed.

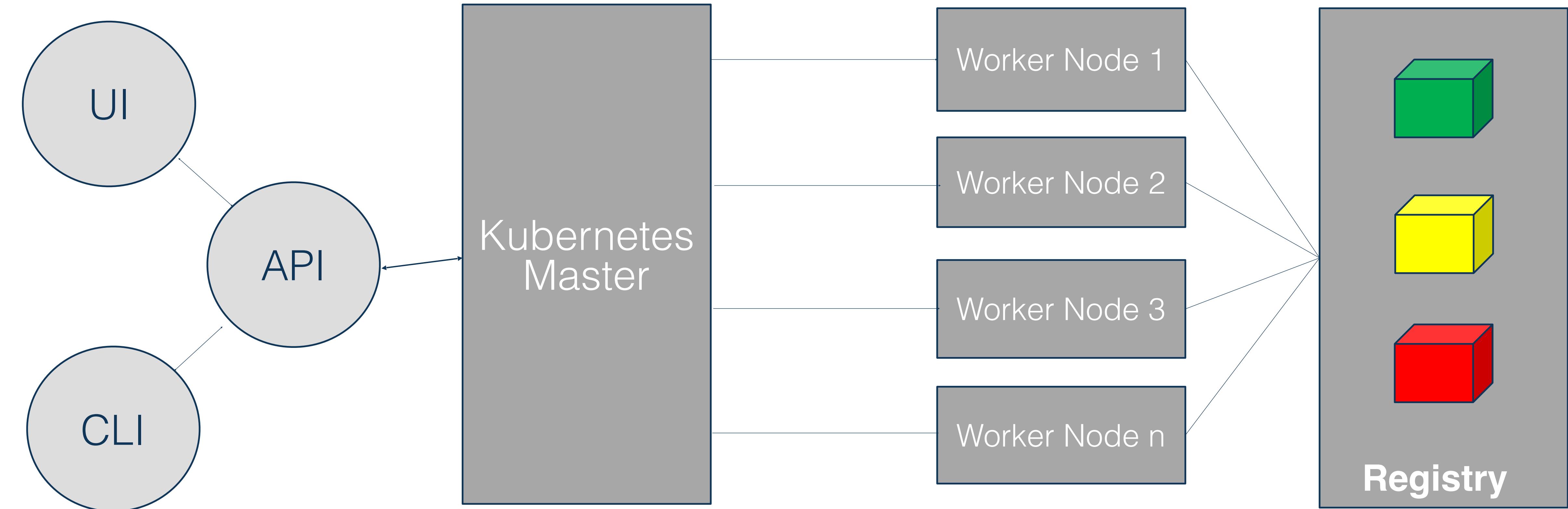
- Scheduling
- Cluster management
- Service discovery
- Provisioning
- Monitoring
- Configuration management

< What is Kubernetes? >



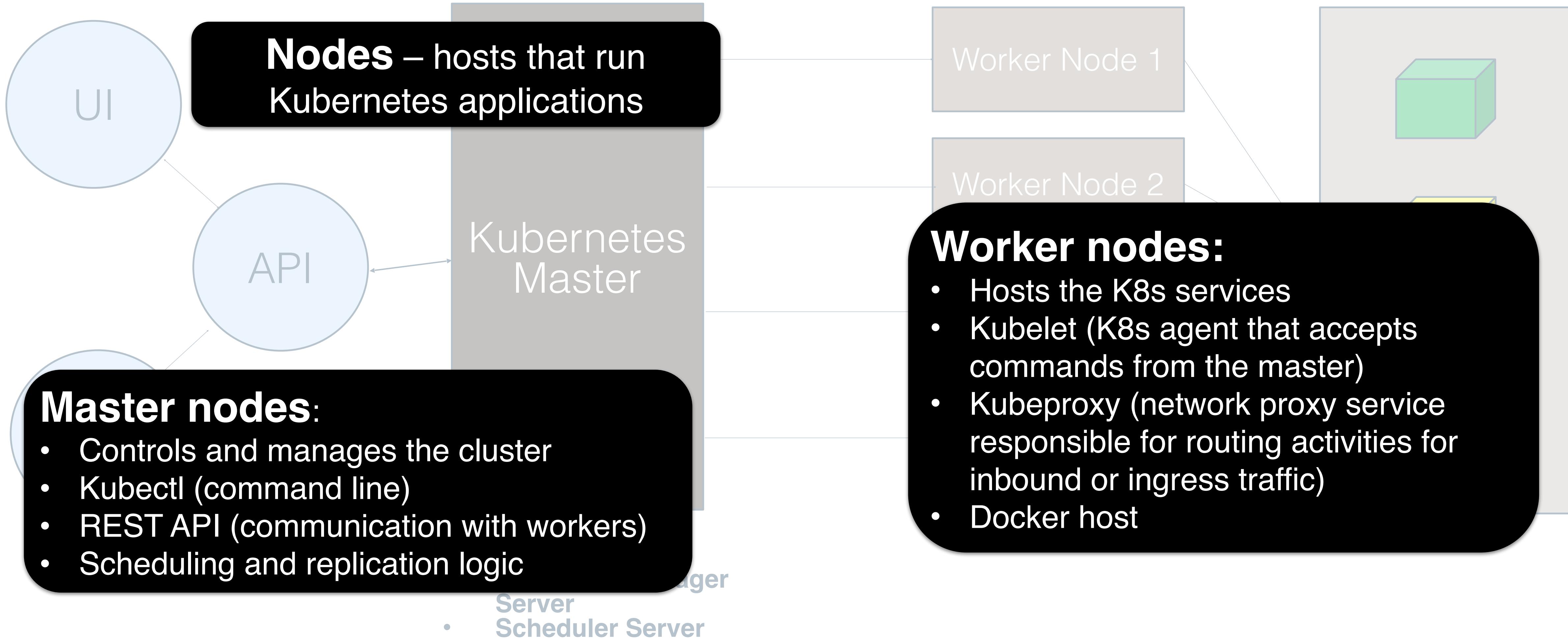
- Container orchestrator
- Runs and manages containers
- Supports multiple cloud and bare-metal environments
- Inspired and informed by Google's experiences and internal systems
- 100% Open source, written in Go
- Manage applications, not machines
- Rich ecosystem of plug-ins for scheduling, storage, networking

< Kubernetes Architecture >



- **Etcd**
- **API Server**
- **Controller Manager Server**
- **Scheduler Server**

</ Kubernetes Architecture >



</ Kubernetes Architecture >



UI

Pods:

- Smallest deployment unit in K8s
- Collection of containers that run on a worker node
- Each has its own IP
- Pod shares a PID namespace, network, and hostname

Replication controller:

- Ensures availability and scalability
- Maintains the number of pods as requested by user
- Uses a template that describes specifically what each pod should contain

Labels:

- Metadata assigned to K8s resources
- Key-value pairs for identification
- Critical to K8s as it relies on querying the cluster for resources that have certain labels

Worker Node 3

Pods:

- Collections of pods exposed as an endpoint
- Information stored in the K8s cluster state and networking info propagated to all worker nodes

Lab1: Criando o seu Primeiro App

<https://github.com/IBM/kube101/tree/master/workshop/Lab1>

The terminal window shows the following output:

```
total 64
drwxr--r-- 1 itiro staff 911B Jul 30 15:19 Makefile
drwxr--r-- 1 itiro staff 388B Jul 30 15:20 deployment-guestbook.yaml
drwxr--r-- 1 itiro staff 432B Jul 25 16:31 deployment-redismaster.yaml
drwxr--r-- 1 itiro staff 440B Jul 25 16:31 deployment-redisslave.yaml
drwxr--r-- 1 itiro staff 185B Jul 26 23:10 extservice-redis.yaml
drwxr--r-- 1 itiro staff 192B Jul 25 15:58 service-guestbook.yaml
drwxr--r-- 1 itiro staff 205B Jul 25 16:31 service-redismaster.yaml
drwxr--r-- 1 itiro staff 202B Jul 25 16:32 service-redisslave.yaml
-rw-r--r-- 1 itiro staff vim Makefile
```

The browser window displays a local development environment for a guestbook application. The page source code is visible, showing HTML, CSS, and JavaScript. The browser status bar indicates the URL is `http://127.0.0.1:8080`.

Lab2: Scale and Update Deployments

<https://github.com/IBM/kube101/tree/master/workshop/Lab2>



Lab3: Building Multi-Tier Apps

<https://github.com/IBM/kube101/tree/master/workshop/Lab3>



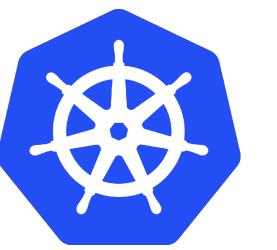
Lab4: App Health

<https://github.com/itirohidaka/kubebr>





Kubernetes Capabilities



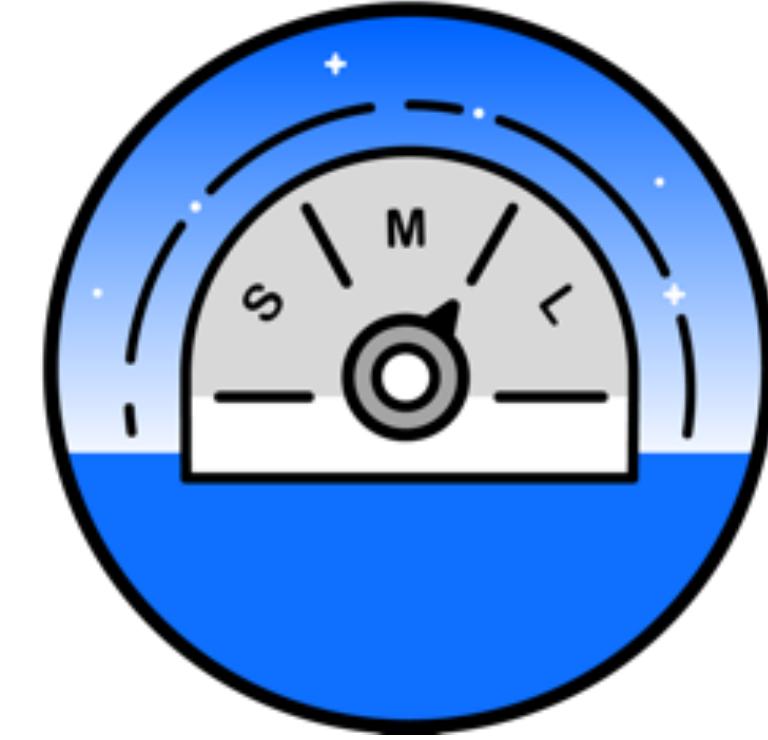
Kubernetes Capabilities



Intelligent Scheduling



Self-healing



Horizontal scaling



Service discovery & load balancing



Automated rollouts and rollbacks



Secret and configuration management

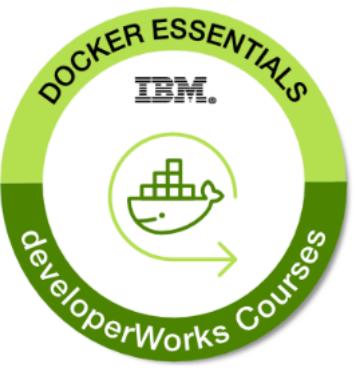
</ Referências e Links >

Links e Referências

- ★ **12 Factor:** <http://12factor.net>
- ★ **Martin Fowler:** <https://martinfowler.com>
- ★ **IKS Labs Github:** <https://github.com/IBM/kube101>
- ★ **Kubernetes Official Page:** <https://kubernetes.io>
- ★ **Kubernetes Online Courses:** <https://kubernetes.io/docs/tutorials/online-training/overview/>
- ★ **Cognitive Class:** <https://cognitiveclass.ai/courses/kubernetes-course/>
- ★ **Istio:** <https://istio.io>
- ★ **Istio Traffic Manager Sample:** <https://github.com/IBM/microservices-traffic-management-using-istio>
- ★ **Istio Labs Github:** <https://github.com/IBM/istio101>

Badges

<https://www.youracclaim.com/>



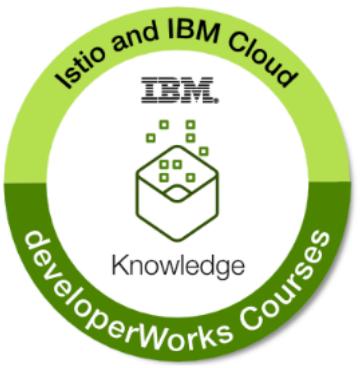
Docker essentials: Extend your apps with containers

<https://developer.ibm.com/courses/all/docker-essentials-extend-your-apps-with-containers/>



Get started with Kubernetes and IBM Cloud Kubernetes Service

<https://developer.ibm.com/courses/all/get-started-kubernetes-ibm-cloud-container-service/>



Get started with Istio and IBM Cloud Kubernetes Service

<https://developer.ibm.com/courses/all/get-started-istio-ibm-cloud-container-service/>

</ Próximos Passos >

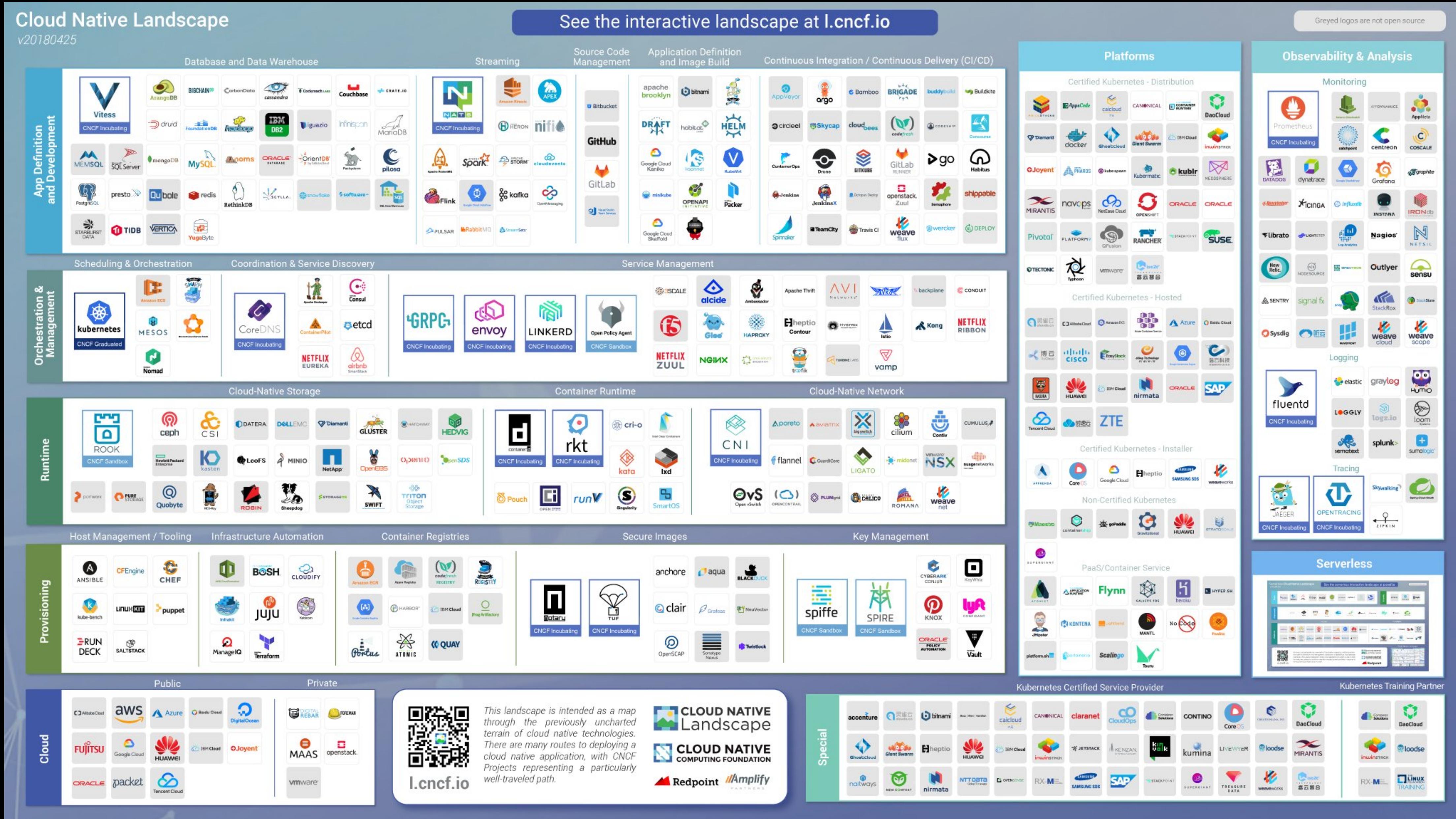
Container Ecosystem

Cloud Native Landscape

v20180425

See the interactive landscape at [l.cncf.io](https://landscape.cncf.io)

Greyed logos are not open source



Próximos Passos

- ▶ **Cloud Private!**
- ▶ **Persistência de Dados, PVC, Cloudant**
- ▶ **ConfigMap e Secrets**
- ▶ **Service Binding**
- ▶ **Vulnerability Assessment**
- ▶ **MZR cluster Design / IBM Cloud Internet Services**
- ▶ **Service Mesh / Istio**