

Java Microservices on Oracle Cloud

Marek Kratky

Cloud Solution Architect, Oracle

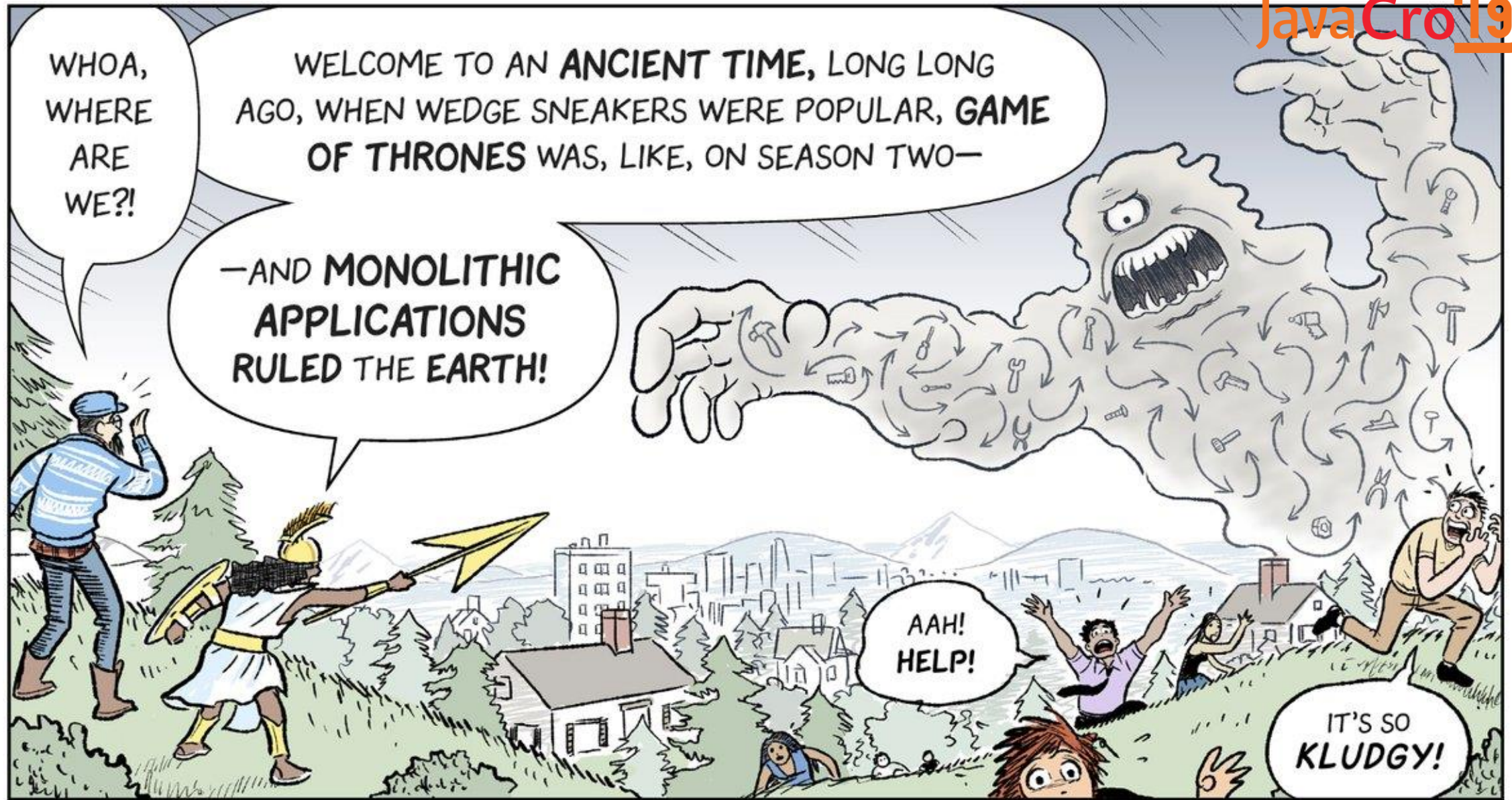
May, 2019

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Agenda

- Concepts: Monolithic vs. Microservices, Containers, Kubernetes
- Trends: Automation, APIs, Cloud
- Building Java apps on Oracle Developer Cloud
- Running Java apps on OCI Container Clusters
- Demo

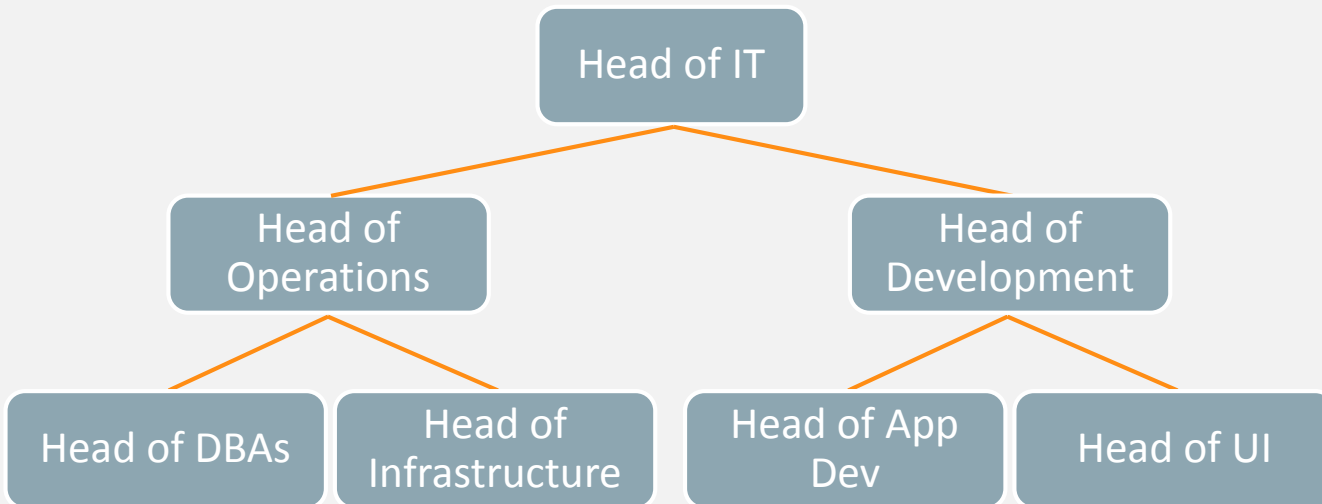


Source: <https://cloud.google.com/kubernetes-engine/kubernetes-comic/>

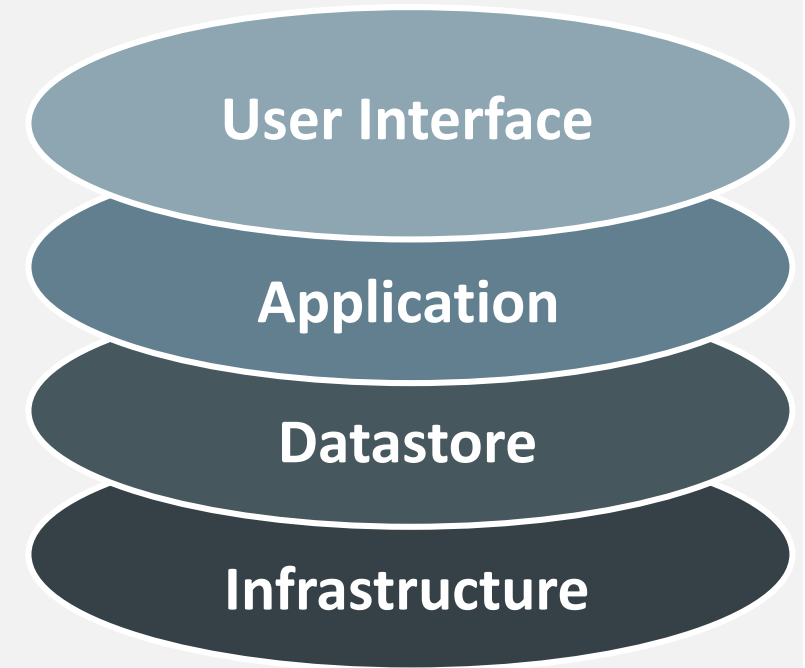
Horizontally Tiered Enterprises == Horizontally Tiered Apps

Conway's Law: Software reflects the structure of the organization that produced it

Typical Enterprise Organization Structure



Resulting Software



Multitier (Monolith) Architecture

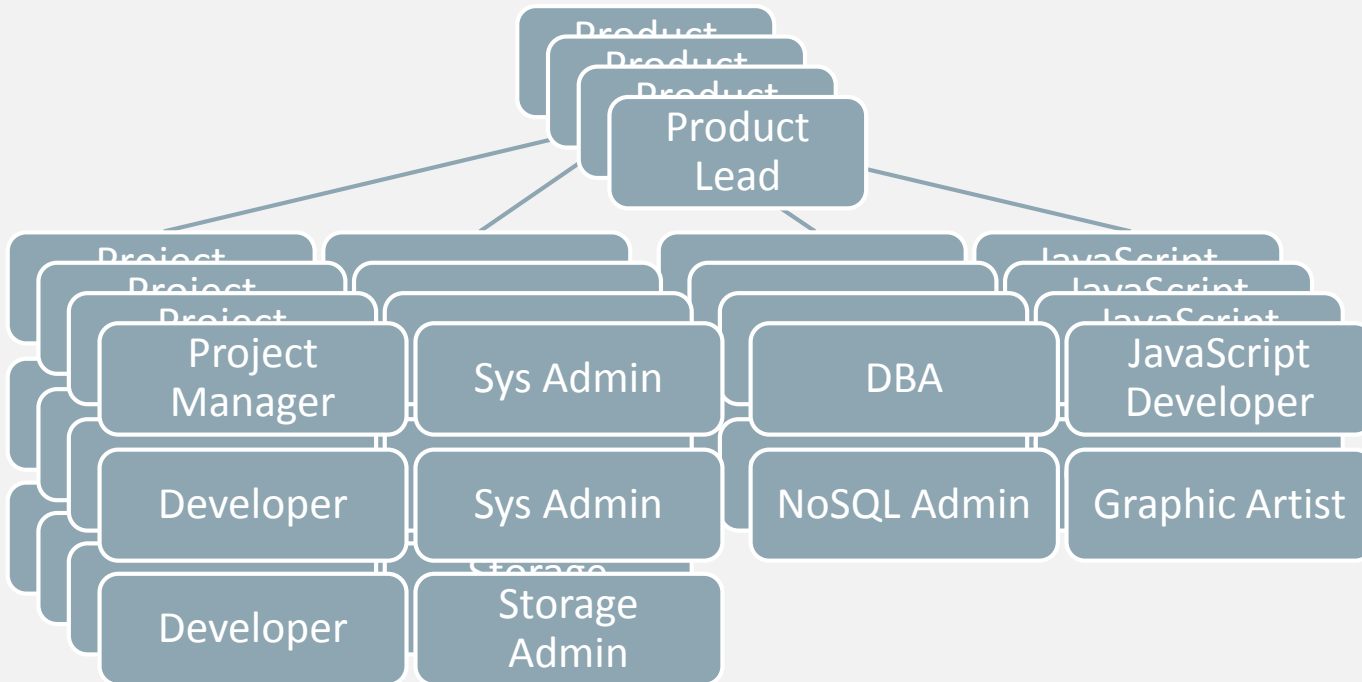


Source: <https://cloud.google.com/kubernetes-engine/kubernetes-comic/>

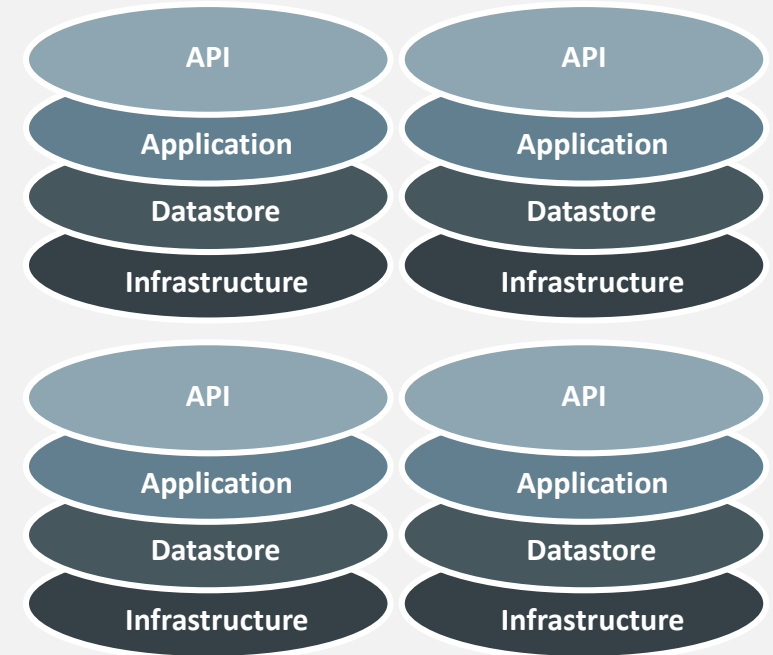
Re-structure Your Organization – Put Conway’s Law to Work

Build small product-focused teams – strict one team to one service mapping

DevOps Organization Structure



Resulting Software



(Micro)Services Architecture

Monolith vs. Microservices

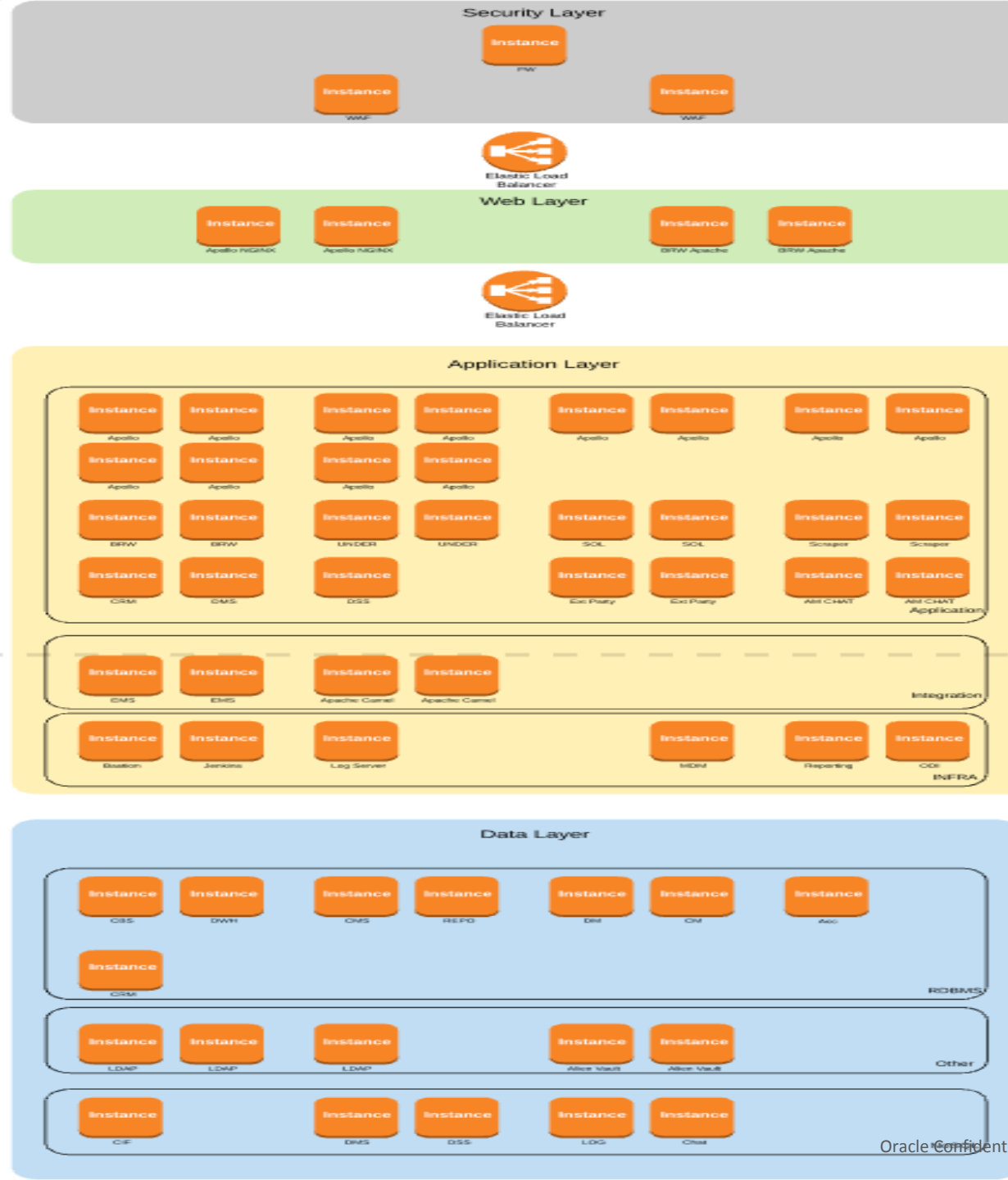


Example:

A Core Banking Application

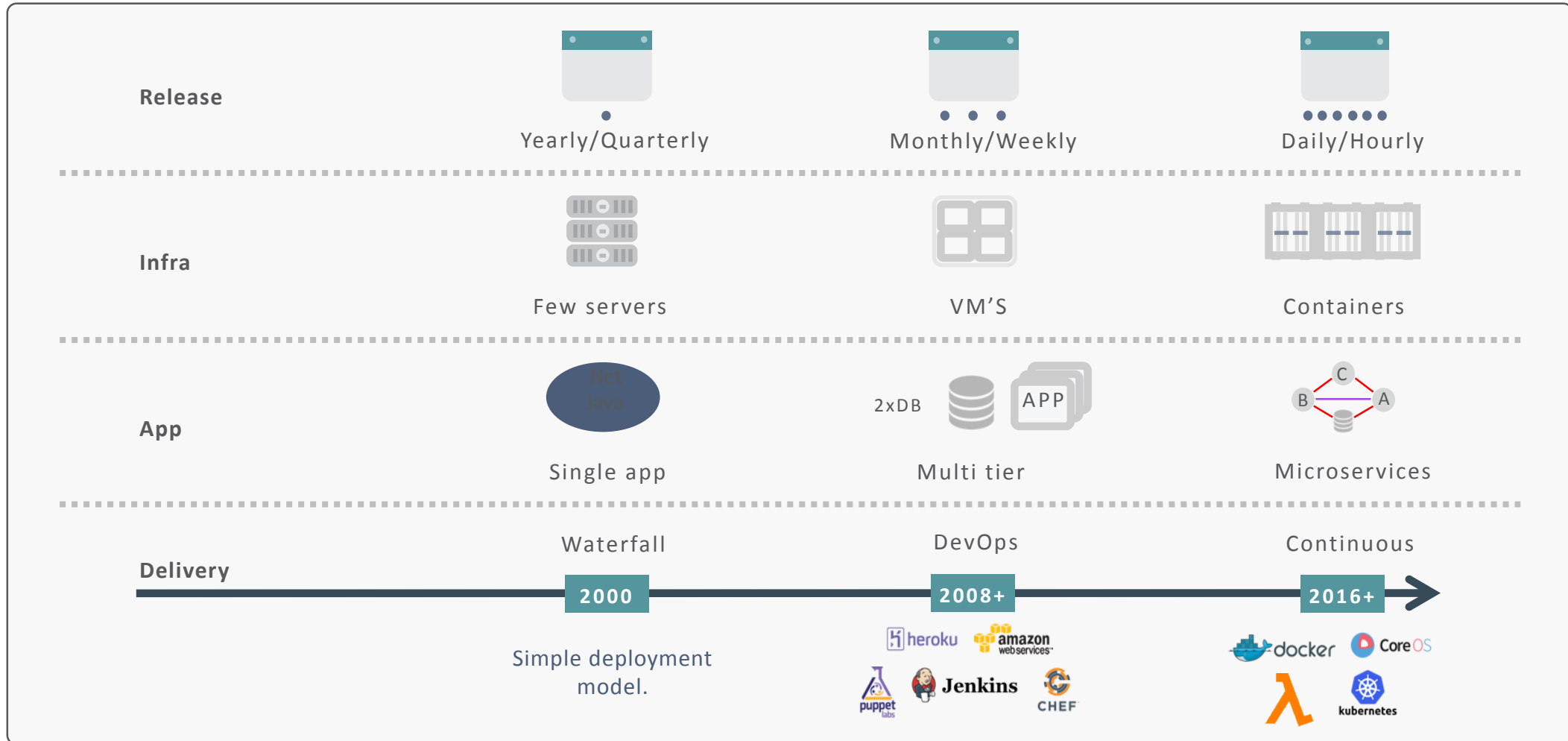
- 276 vCPUs/1496 GB of (Linux) virtual machines
- 128 vCPUs of Oracle DB

Is it Monolithic or (Micro)services?



Modern Development in Context

The Last 15 Years has been about driving increased development velocity

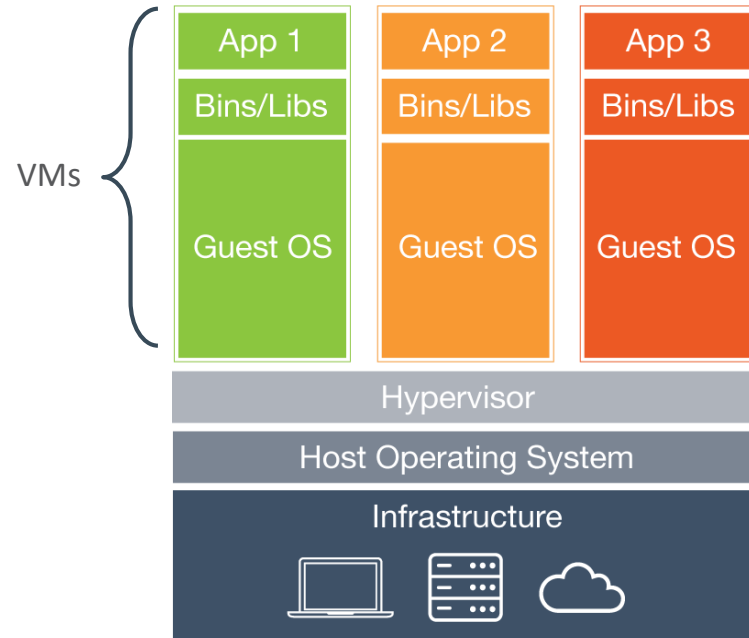


How do we treat the application infrastructure

Cattle vs Pets

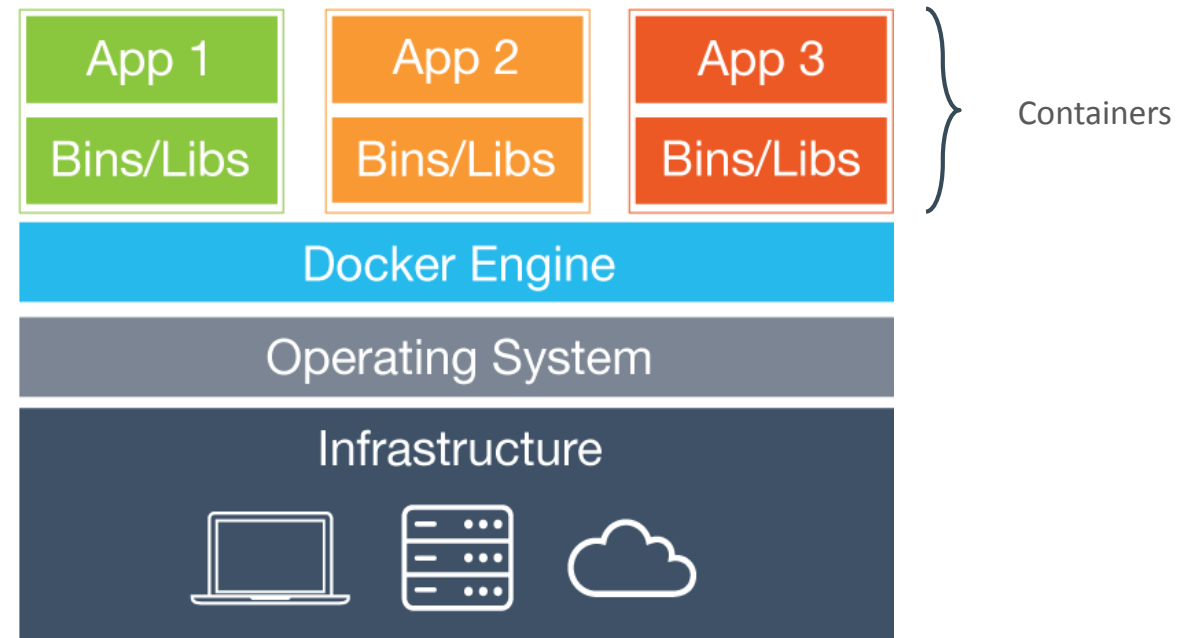


Virtual Machines vs. Containers



Virtual Machines

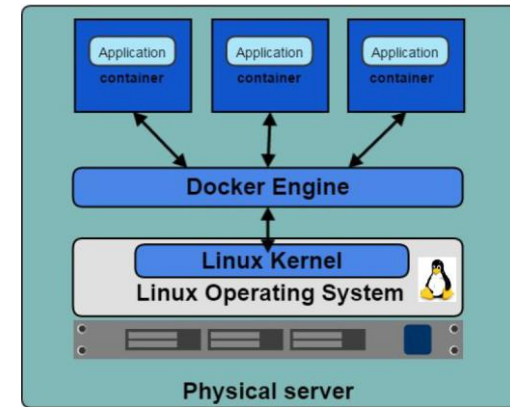
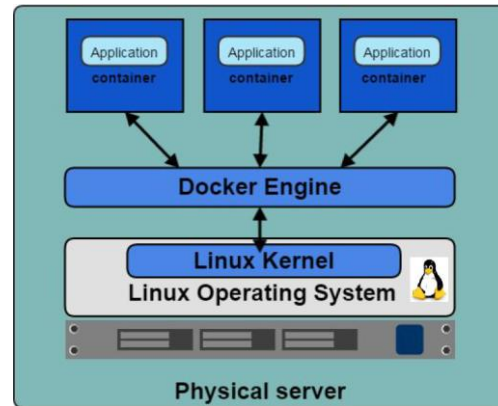
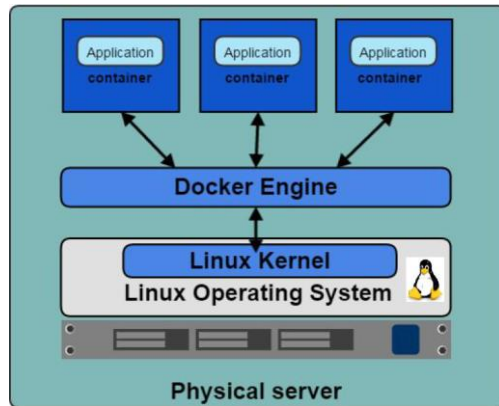
- Each virtual machine (VM) includes the app, the necessary binaries and libraries and an **entire guest operating system**



Containers

- Containers include the app & all of its dependencies, but **share the kernel** with other containers.
- Run as an isolated process in userspace on the host OS
- **Not** tied to any specific infrastructure – containers run on any computer, infrastructure and cloud.

How do you manage all these running containers on a single host, and, more importantly, across your whole infrastructure?



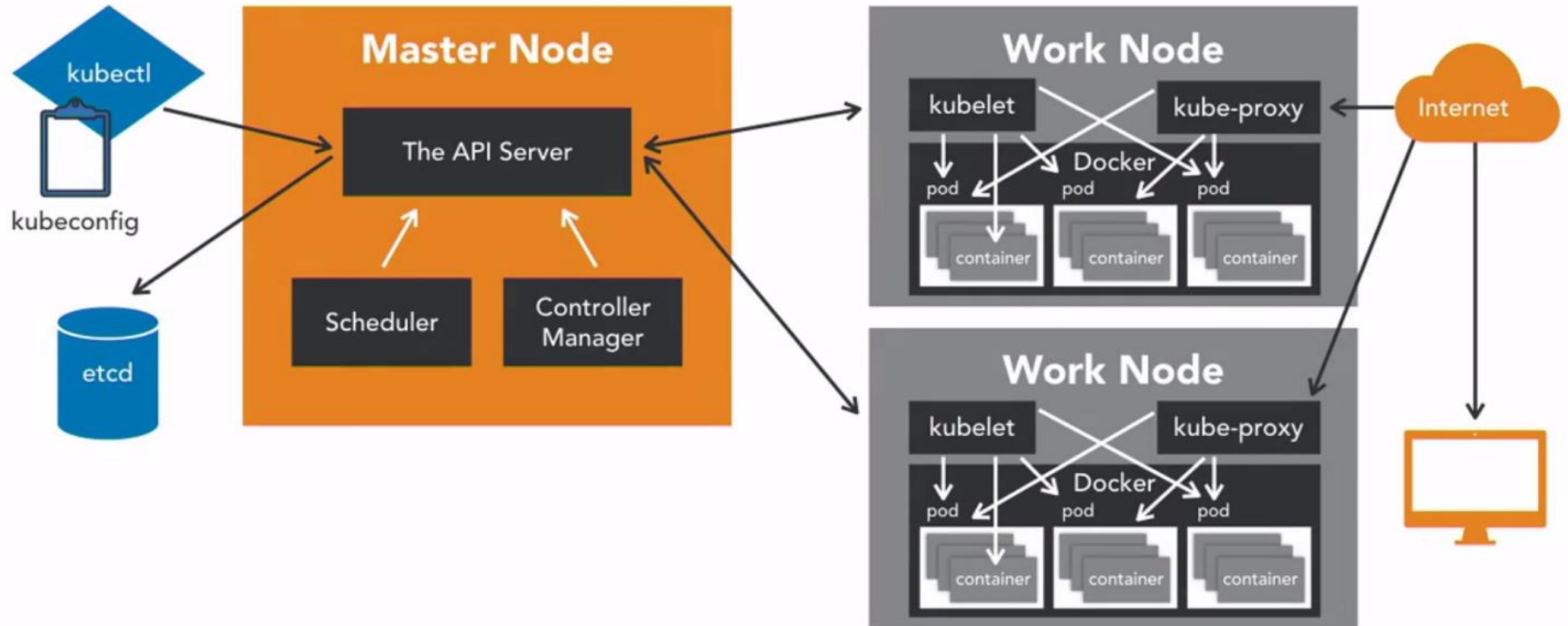
Orchestrator Features

- Provision hosts
- Instantiate containers on a host
- Restart failing containers
- Expose required containers as services outside the cluster
- Scale up or down the cluster

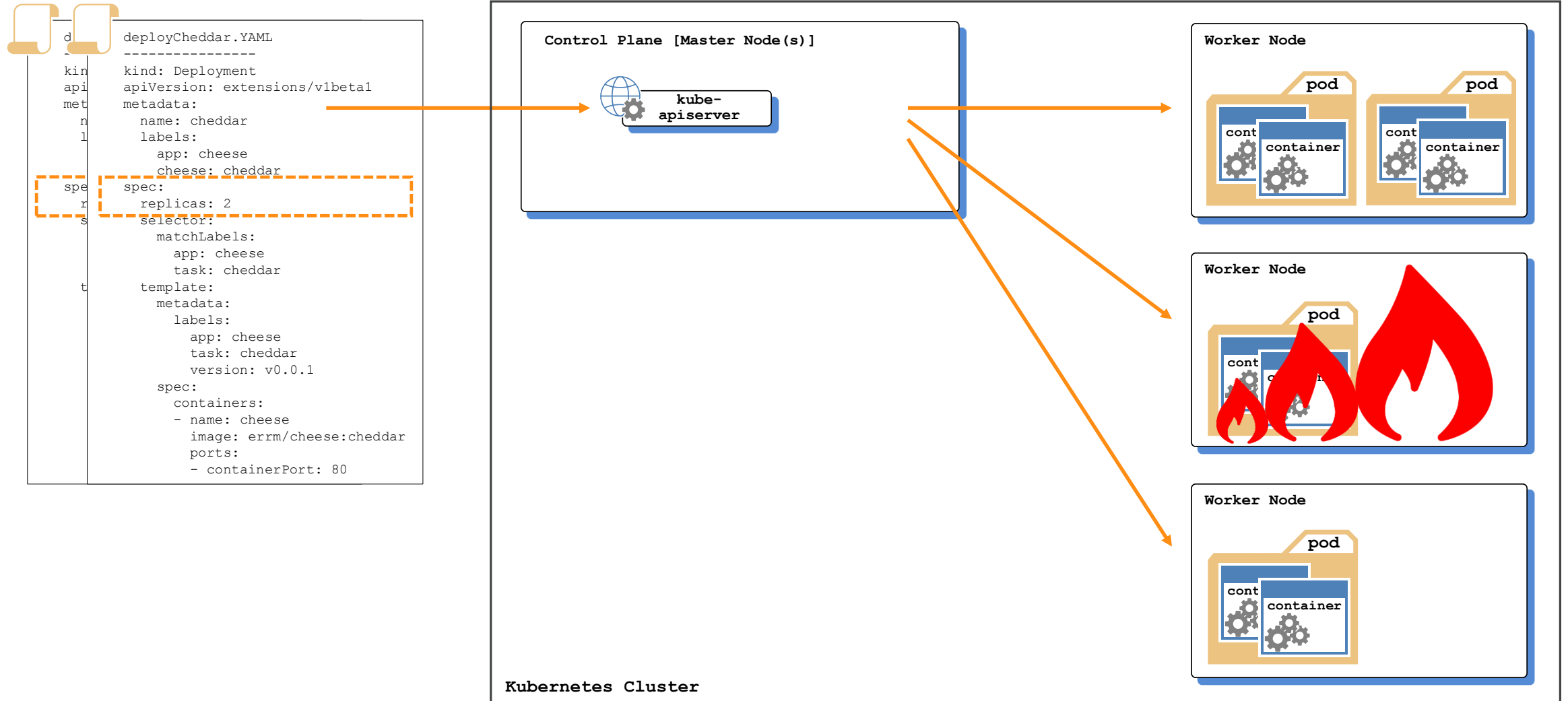




Architecture



Kubernetes – In Motion



What are Container Native Applications?

- Software that treats the container as the first-class unit of infrastructure
- Software that does not just “happen to work” in, on or around containers, but rather is purposefully designed for containers
- Represents a paradigm shift that enables microservices and serverless architectures



@ContainerXinc www.containerx.io



Helidon MP

JAX-RS

CDI

JSON-P

Cloud
Integrations

Helidon SE

Web Server

Netty

Config

Security

Landscape



Full-Stack

 Spring Boot  Dropwizard

MicroProfile Based

 Open Liberty  payara **THORNTAIL**  **helidon MP**

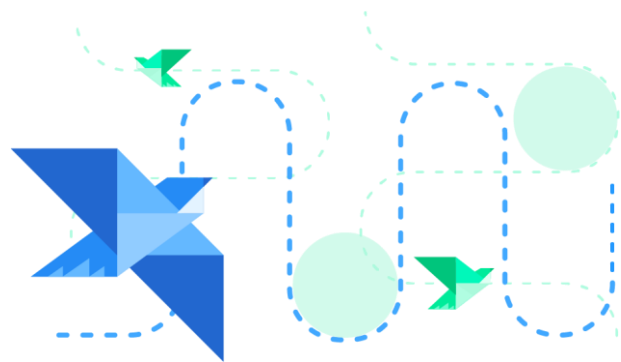
Microframeworks

 **javalin**  **Spark**  **MICRONAUT™**  **helidon SE**



helidon SE

- Microframework
- Functional style
- Reactive
- Transparent



helidon MP

- MicroProfile
- Declarative style
- CDI, JAX-RS, JSON-P



Helidon SE

```
Routing routing = Routing.builder()  
    .get("/hello", (req, res) -> res.send("Hello World"))  
    .build();
```

```
WebServer.create(routing)  
    .start();
```

Helidon MP

```
@Path("hello")
@ApplicationScoped
public class HelloWorld {
    @GET
    public String hello() {
        return "Hello World";
    }
}
```

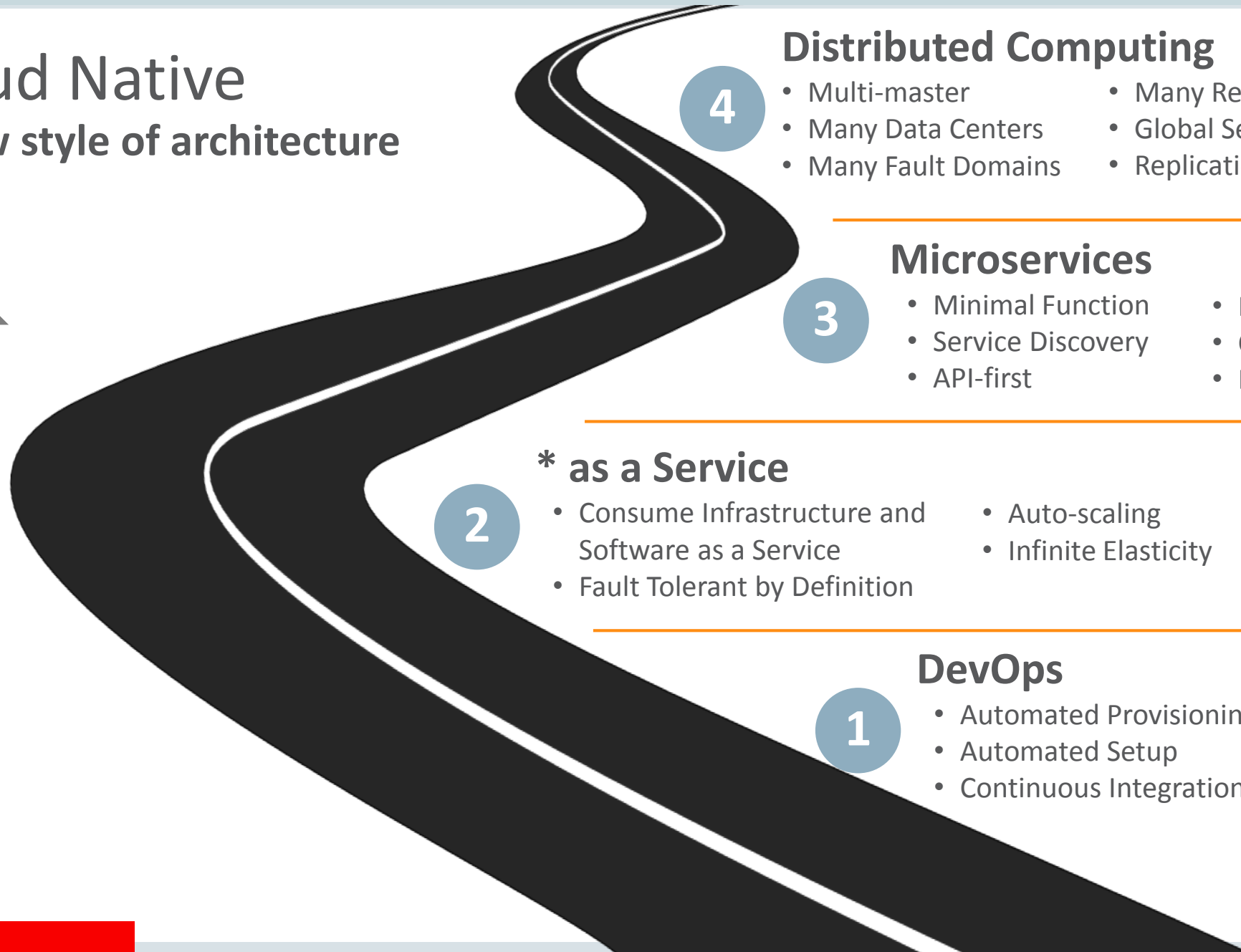
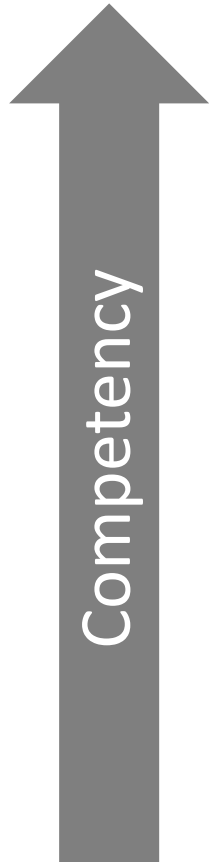
```
java -cp ... io.helidon.microprofile.server.Main
```

Generate The Project

```
mvn archetype:generate
-DinteractiveMode=false \
-DarchetypeGroupId=io.helidon.archetypes \
-DarchetypeArtifactId=helidon-quickstart-mp \
-DarchetypeVersion=1.0.3 \
-DgroupId=io.helidon.examples \
-DartifactId=helidon-quickstart-mp \
-Dpackage=io.helidon.examples.quickstart.mp
```

Cloud Native

A new style of architecture



4

Distributed Computing

- Multi-master
- Many Data Centers
- Many Fault Domains
- Many Regions
- Global Server Load Balancing
- Replication

3

Microservices

- Minimal Function
- Service Discovery
- API-first
- Polyglot
- Choreography
- Loose Coupling

2

* as a Service

- Consume Infrastructure and Software as a Service
- Fault Tolerant by Definition
- Auto-scaling
- Infinite Elasticity

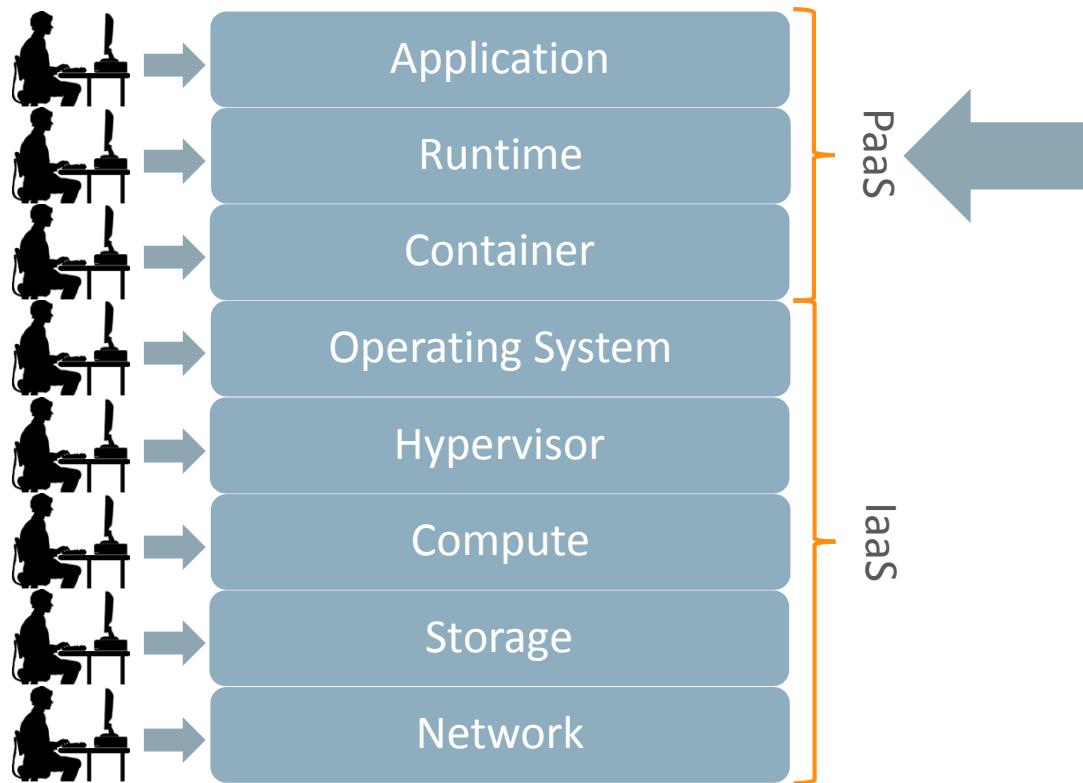
1

DevOps

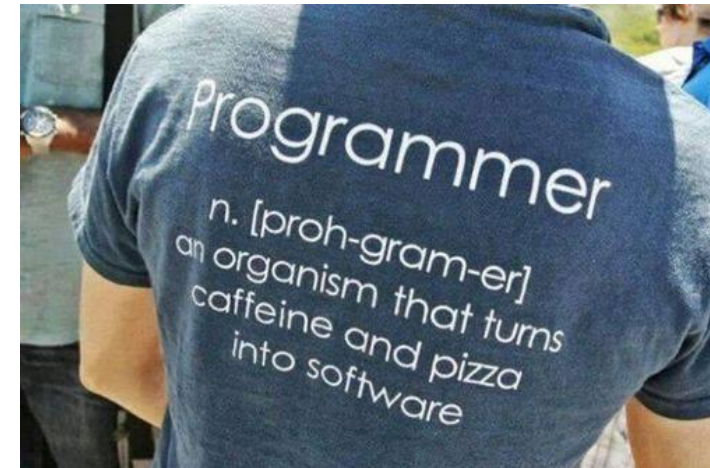
- Automated Provisioning
- Automated Setup
- Continuous Integration
- Continuous Delivery
- Automated Testing
- Agile
- Culture Change

Cloud REST API

Past: Ops manually provisioned each layer



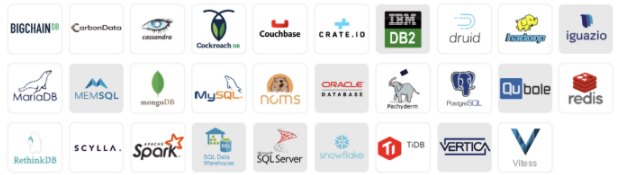
Today: Developers can provision entire stacks of hardware + software through REST API



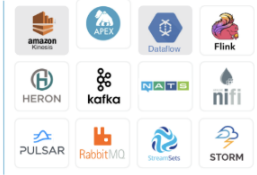
Cloud Native Landscape

v1.0

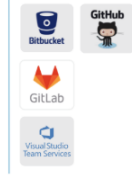
Database & Data Analytics



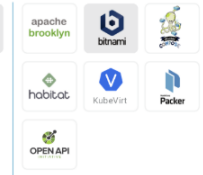
Streaming



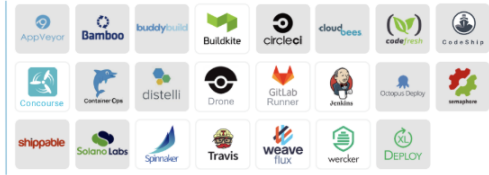
SCM



Application Definition



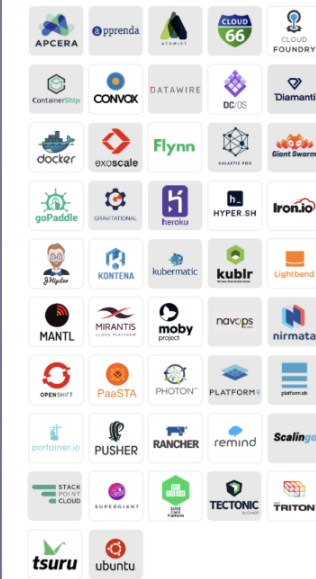
CI/CD



App Definition & Development

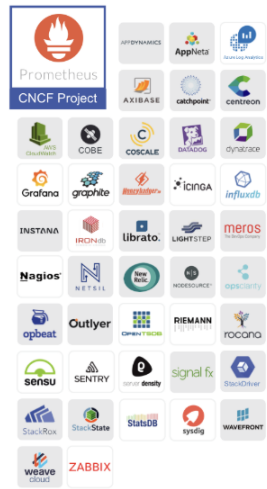
Platforms

PaaS / Container Service



Observability & Analysis

Monitoring



Logging

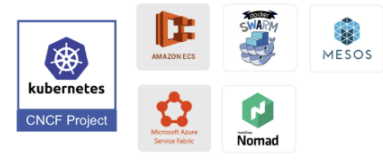


Tracing

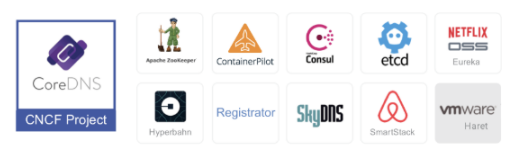


Orchestration & Management

Scheduling & Orchestration



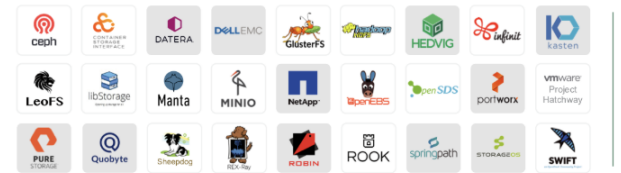
Coordination & Service Discovery



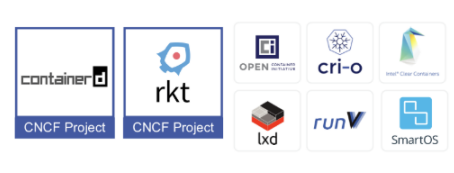
Service Management



Cloud-Native Storage



Container Runtime

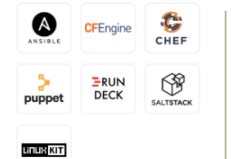


Cloud-Native Network

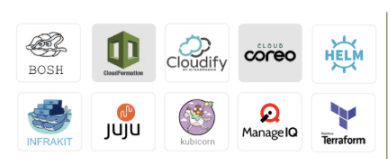


Runtime

Host Management / Tooling



Infrastructure Automation



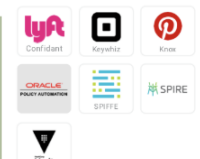
Container Registries



Secure Images

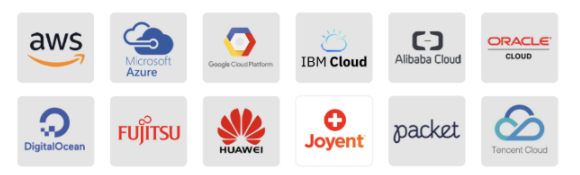


Key Management

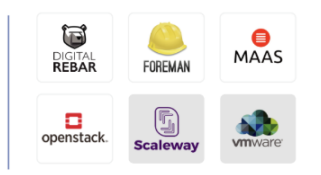


Provisioning

Public



Private

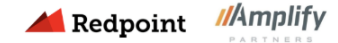


Cloud



github.com/cncf/landscape

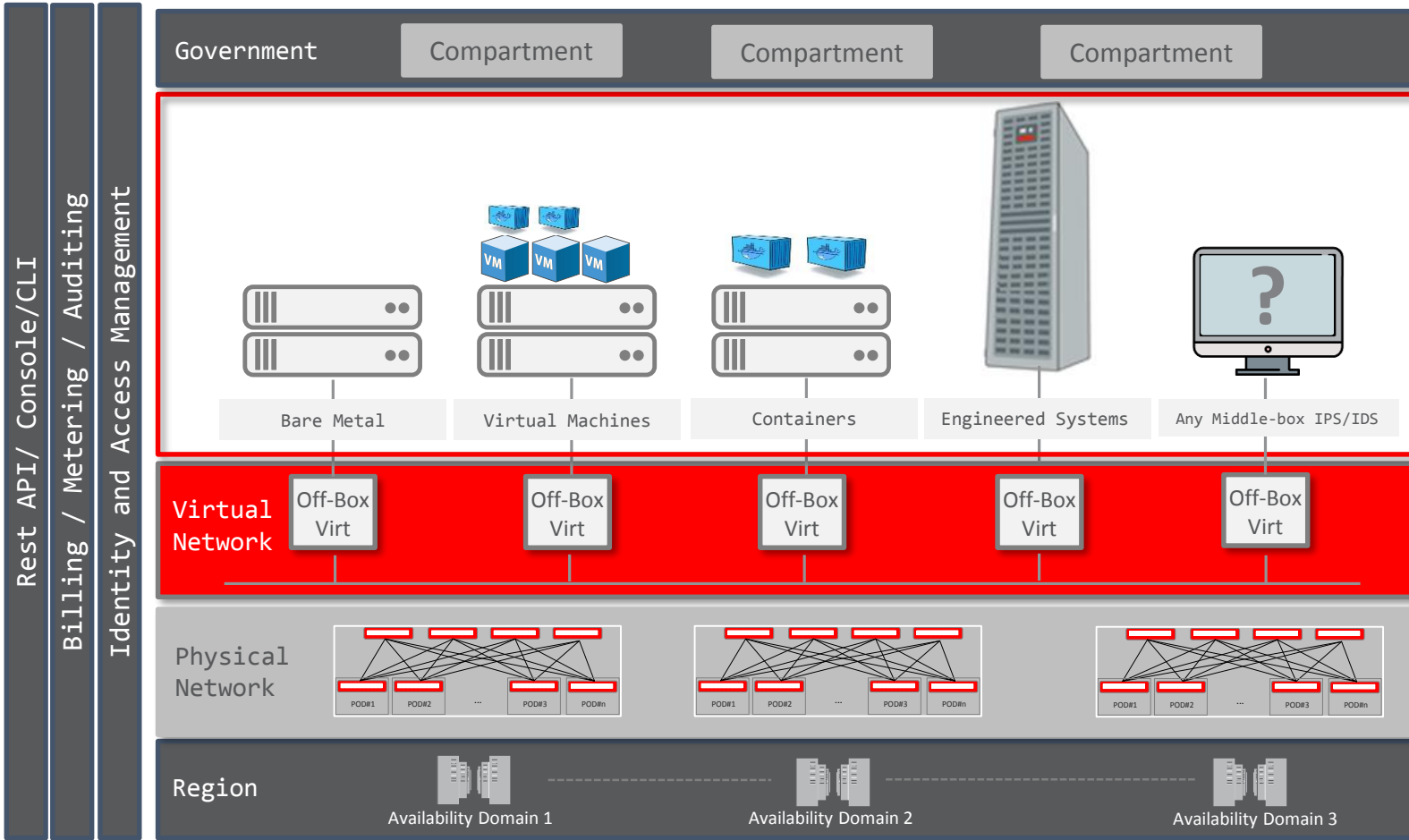
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.



Greyed logos are not open source

Oracle Cloud Infrastructure

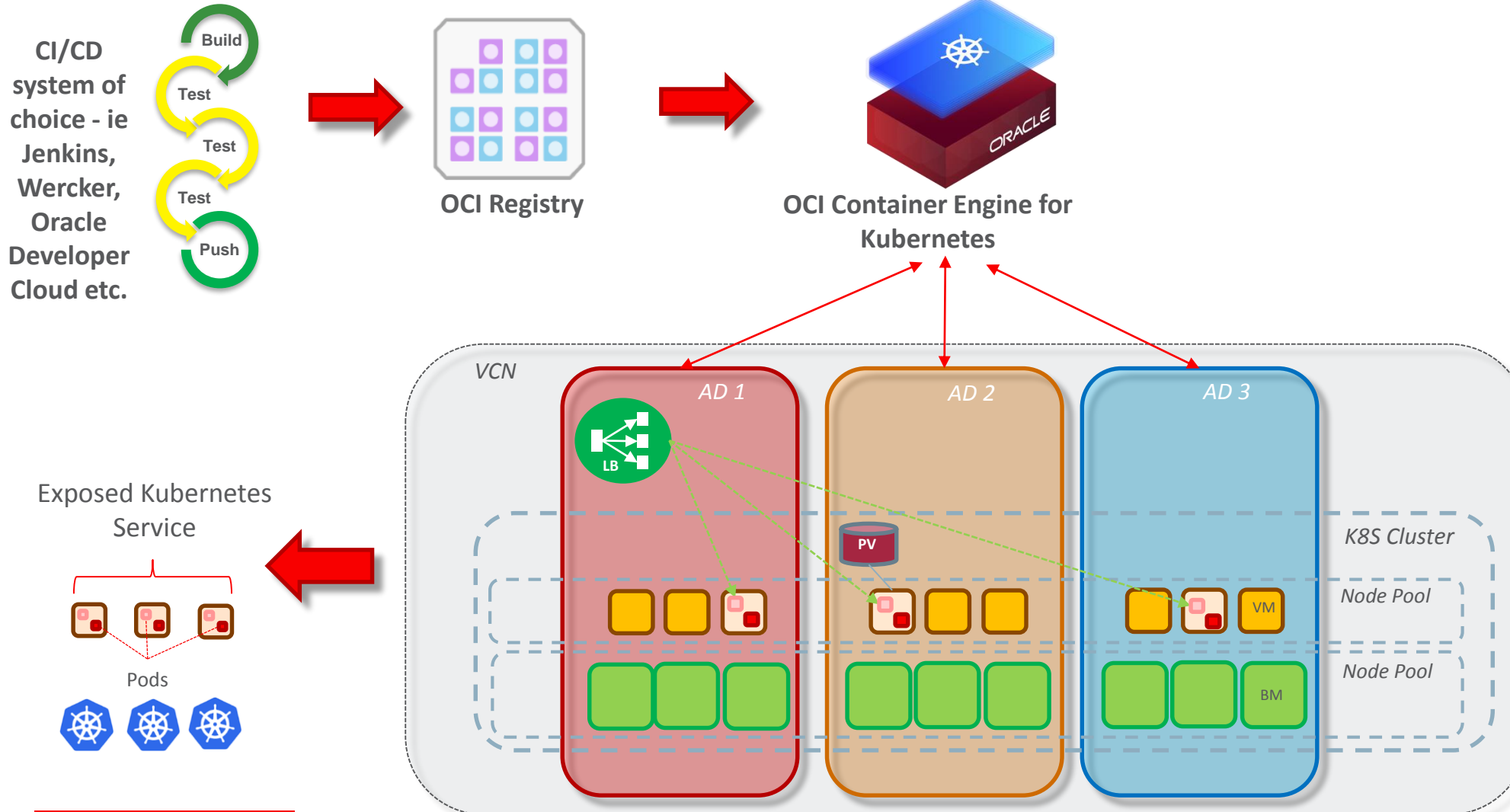
Dedicated resources and Performance-first approach



- **Non-oversubscription** CPU and network
- Single or multi-tenant with **same set of APIs**
- **NVMe** flash drives and super-fast SSD block volumes
- **IOPS** that scale **linearly**
- Easier **governance** with Compartments capabilities
- **First Enterprise SLAs** (Availability, Manageability, Performance)

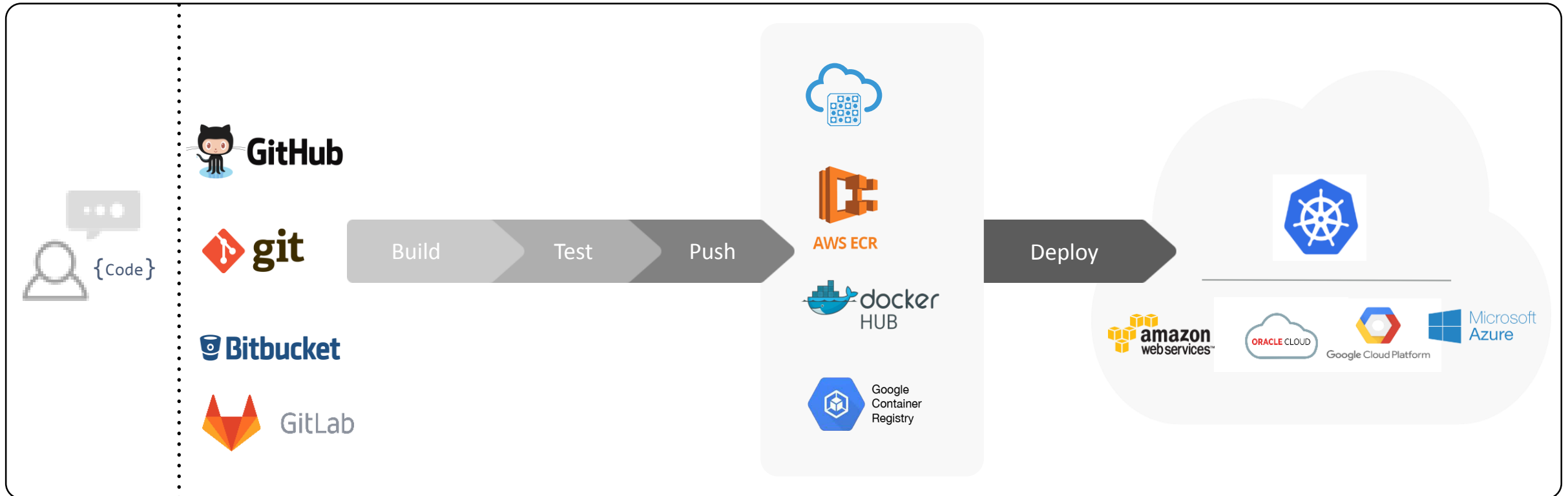
OCI Container Engine for Kubernetes and Registry

An Open, Fully-Managed Kubernetes Platform & Private Registry

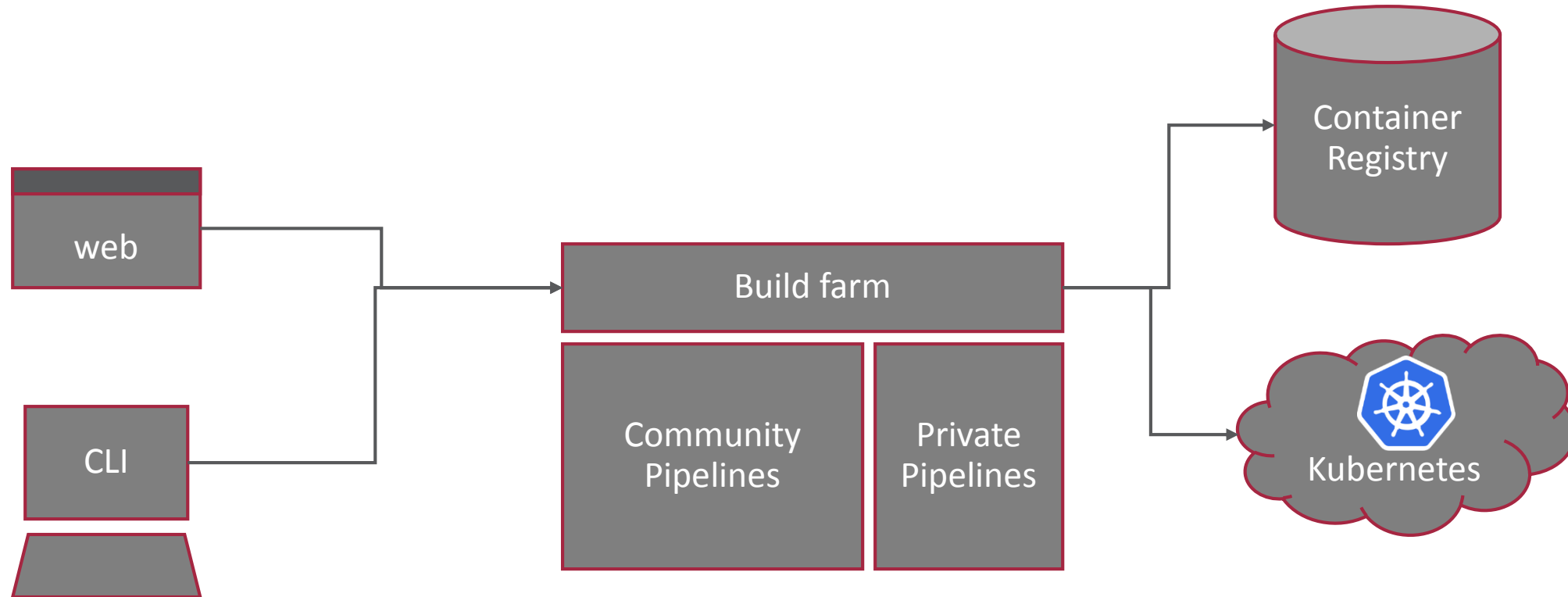


CI/CD with Oracle Wercker

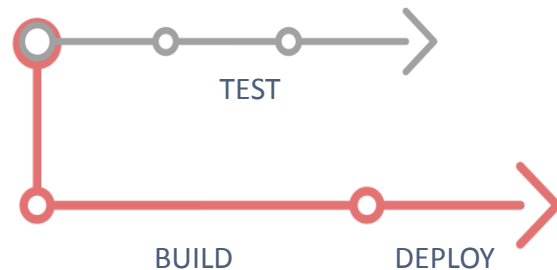
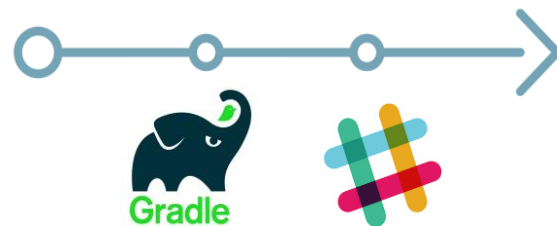
Deploy to any orchestration tool on any IaaS



Wercker Architecture



Wercker features



Pipelines

Pipelines are a series of steps that are triggered on a git push or the completion of another pipeline. Pipelines result in an artefact and are executed inside containers.

Steps (*pipelines consists of steps*)

Isolated bash script or compiled binary for accomplishing specific automation tasks.

They can be created from scratch or consumed from our open steps marketplace.

Workflows (*workflows consists of pipelines*)

Workflows is a set of chained and branched pipelines that allow you to form multi-stage, multi-branch complex ci/cd flows that take your project from code to production.

mkratky / angular-node-creditscore

Runs

Workflows

Access

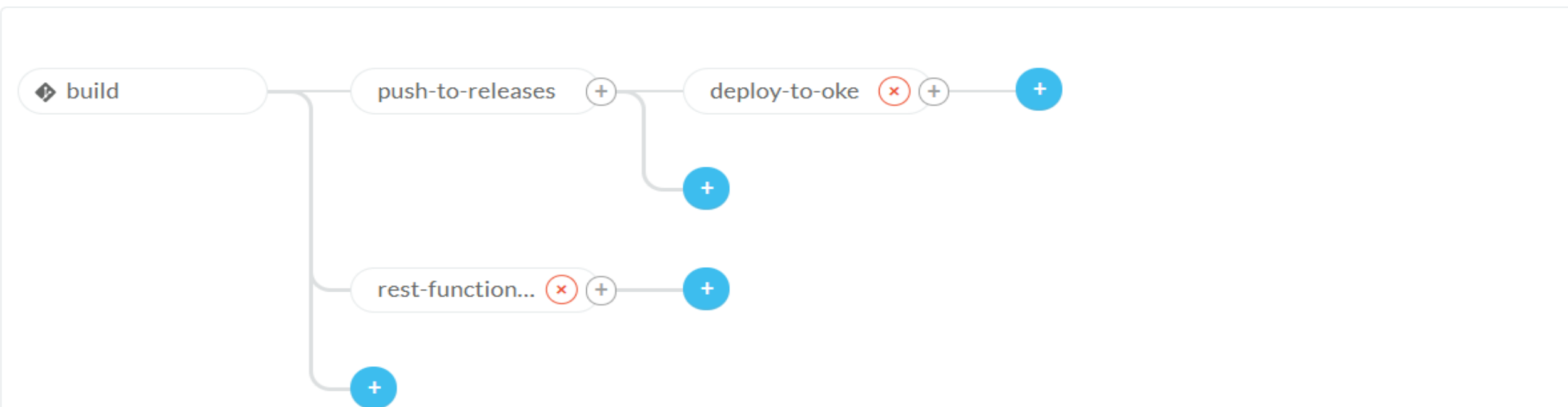
Environment

Options

Editor

Workflows are a way to [manage automation pipelines](#).

You can use them to chain pipelines together and configure on which git branch they should run



 mkratky / angular-node-creditscore 

 Runs

 Workflows

 Access

 Environment

 Options

Application environment variables

Settings and passwords defined here will be available to all pipelines

Key

Value

KUBERNETES_MASTER

https://c3tkyrygvtg.eu-frankfurt-1.clusters.oci.oraclecloud.com:6443

Delete

KUBERNETES_TOKEN

Protected

Delete

DOCKER_REGISTRY

https://fra.ocir.io/v2

Delete

DOCKER_REPO

fra.ocir.io/oraseemeaceeociworkshop/mkratky

Delete

mkratky / angular-node-creditscore

Runs

Workflows

Access

Environment

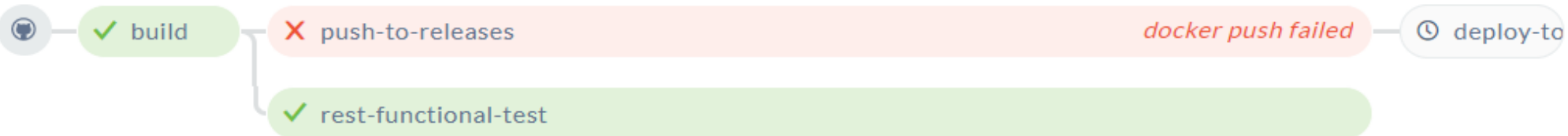
Options

✓ build now

master

build

Actions



Details

Author

mkratky

Branch

master

Commit

#89eb333

Created

3 days ago

Started

3 days ago

Duration

1 minute, 3 seconds

Steps

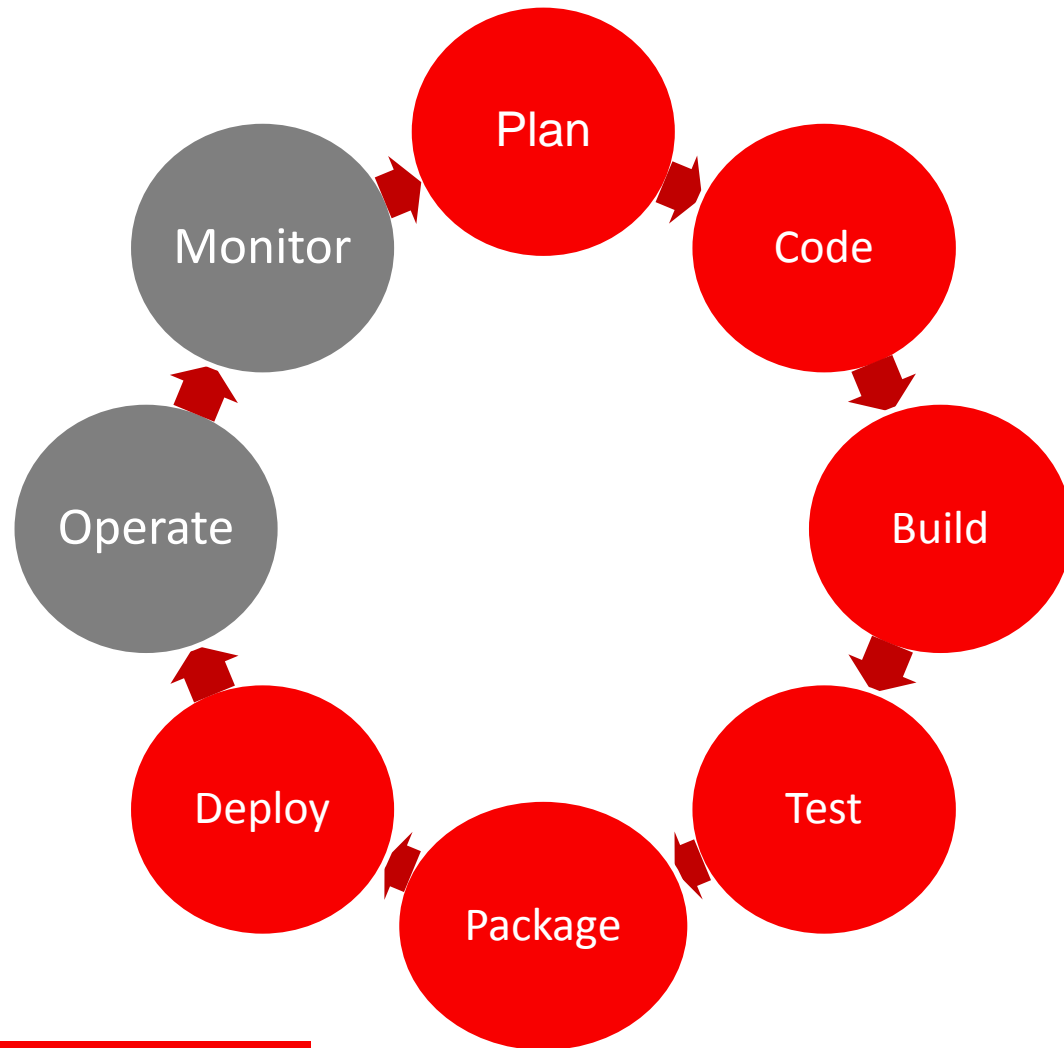
✓ get code 1 second

✓ setup environment 54 seconds

✓ wercker-init 0 seconds

Pipeline [Edit](#)

CI/CD with Oracle Developer Cloud Service



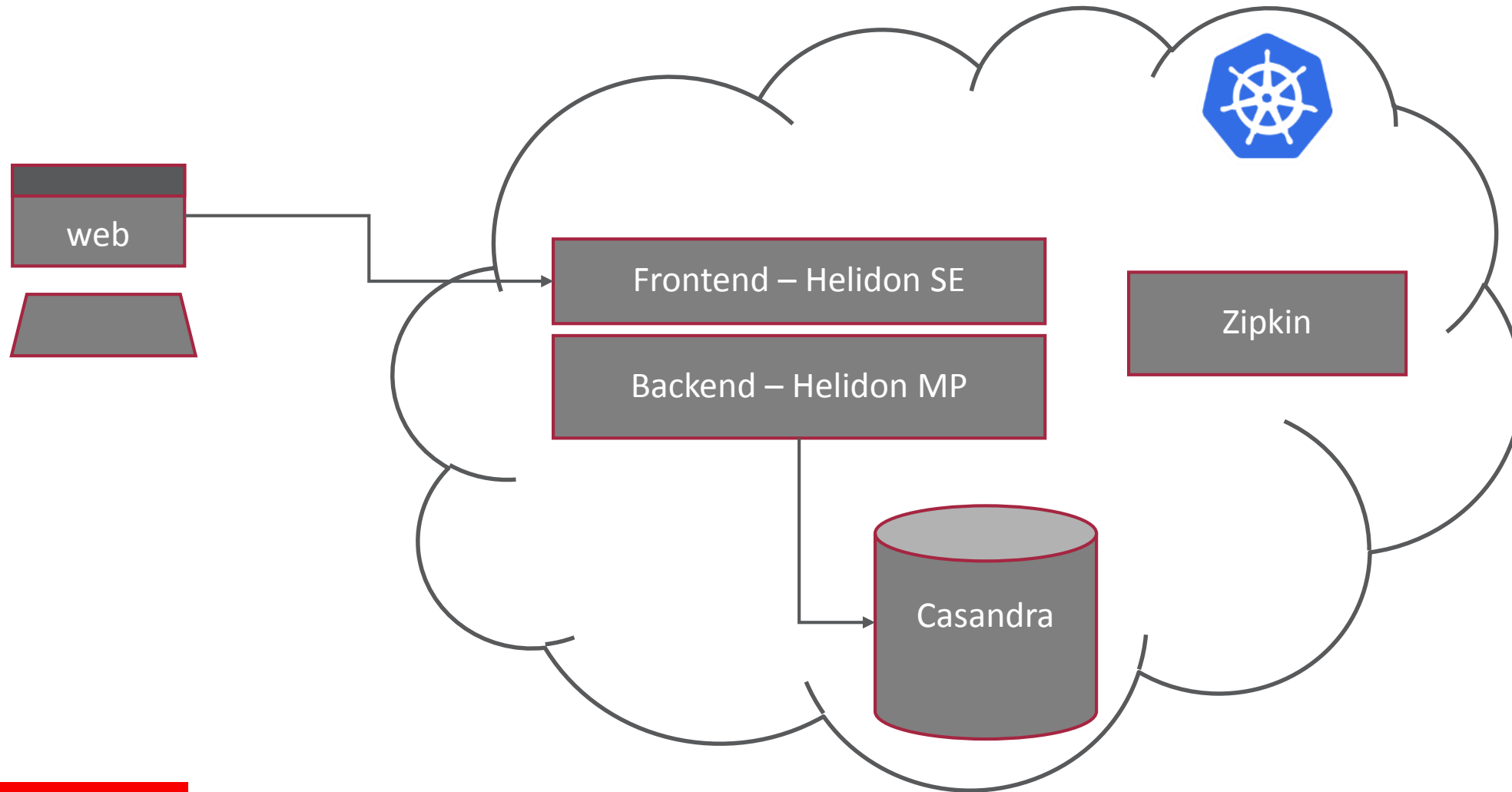
- Track Issues
- Agile Project Management
- Wikis
- Git Repositories
- Code Review
- Build Frameworks
- Orchestration and Dependencies
- Build Reports and Notifications
- Junit, Selenium, FindBugs
- QA Deployments
- Create packages
- Push to Docker Registry



Demo



Sample TODO application with Helidon



helidon-todo-app | Git

helidon-todo-app.git | master

Files | Logs | Refs | Compare

Clone

Click to add description of this repository.

demo-backend	Initial commit Oracle Cloud Infrastructure Client	April 26, 2019 5:32 PM +0200
demo-frontend	removed duplicate google login dependency marek.kratky@oracle.com	April 26, 2019 5:58 PM +0200
etc	Initial commit Oracle Cloud Infrastructure Client	April 26, 2019 5:32 PM +0200
k8s	Initial commit Oracle Cloud Infrastructure Client	April 26, 2019 5:32 PM +0200
docker-compose.yml	Initial commit Oracle Cloud Infrastructure Client	April 26, 2019 5:32 PM +0200
kubescrypt.sh	k8s/k8s-deployment.yml marek.kratky@oracle.com	April 26, 2019 7:07 PM +0200
pom.xml	Initial commit Oracle Cloud Infrastructure Client	April 26, 2019 5:32 PM +0200
README.md	Initial commit Oracle Cloud Infrastructure Client	April 26, 2019 5:32 PM +0200

+ File

README.md

TODOs Demo Application

If you want to run behind a proxy, you need to configure the following in application.yaml of both services (find appropriate existing google-login provider configuration):

```
providers:  
  - google-login:  
      proxy-host: "proxy_host"
```



helidon-todo-app.git

Files Logs Refs Compare

b2e6fca91fbf861af1efbdc5da04c6d1f1a7cd74



f16439d75ae059c6f009f99422bc0fb01939eb28

marek.kratky@oracle.com April 26, 2019 5:57 PM +0200



marek.kratky@oracle.com April 26, 2019 5:58 PM +0200

▶ **google login dependency**

▶ **removed duplicate google login dependency**

Changed Files Commits

pom.xml -4 demo-frontend



Filter



pom.xml -4 demo-frontend

Hide

```

@@ -39,10 +39,6 @@
39 39
40 40     <dependencies>
41 41         <dependency>
42         <groupId>io.helidon.security.providers</groupId>
43         <artifactId>helidon-security-providers-google-login</artifactId>
44         </dependency>
45         <dependency>
46 42         <groupId>io.helidon.common</groupId>
47 43         <artifactId>helidon-common</artifactId>
48 44         </dependency>

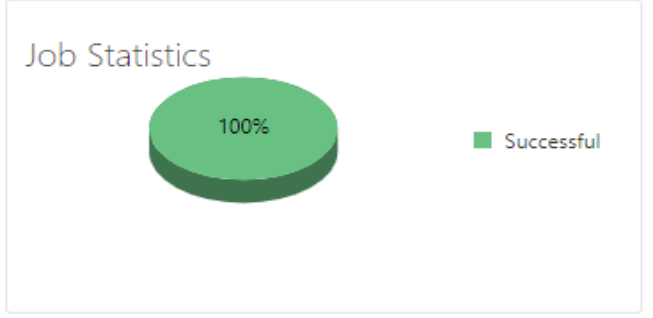
```

helidon-todo-app | Builds

Search Jobs

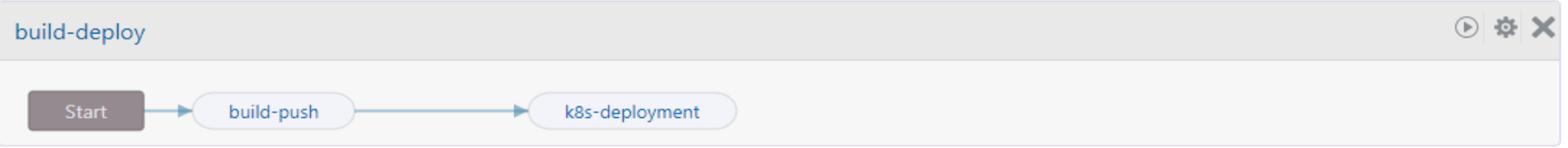
Build Queue
No build in progress

[View Recent Build History](#)



Jobs Pipelines

+ Create Pipeline



Jobs Overview > **build-push**

Build Now | Configure | Disable | Delete

Job Details

No description available

Changes | Artifacts | Javadoc | Tests | Build Log | Git Log | Audit

SonarQube | Vulnerabilities

Notifications

On | Off | CCme

Build History

Last | Successful | Unsuccessful | Failed | Test Failed

By	Status	Build	Started	Duration	Actions
	✓	#4	Apr 26, 2019 7:09 PM	1 m 5 s	
	✓	#3	Apr 26, 2019 6:14 PM	1 m 46 s	
	✓	#2	Apr 26, 2019 6:11 PM	1 m 16 s	
	✓	#1	Apr 26, 2019 5:52 PM	49 s	

Build Trend



Test Result Trend

- Organization
- Project Home
- Git
- Merge Requests
- Maven
- Docker
- Releases
- Builds**
- Deployments
- Environments
- Issues
- Boards
- Wiki
- Snippets
- Project Administration

Job Configuration

Cancel Save



- Git
- Parameters
- Before Build
- Steps**
- After Build

Configure Steps

Add Step ▾

Maven ✕

Goals

POM File

▶ Advanced Maven Settings

Docker login ✕

Docker logout will be performed automatically at the end of all build steps.

Registry Host

[Link External Registry](#)

* Username

* Password

Docker build ✕

* Registry Host

* Image Name

- Storage Classes
- Namespace
 - default
- Overview
- Workloads
 - Cron Jobs
 - Daemon Sets
 - Deployments
 - Jobs
 - Pods
 - Replica Sets
 - Replication Controllers
 - Stateful Sets
- Discovery and Load Balancing
 - Ingresses
 - Services
- Config and Storage
 - Config Maps
 - Persistent Volume Claims
 - Secrets
- Settings

Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age	
quickstart-mp	app: quickstart-mp	10.96.1.82	quickstart-mp:8080 TCP quickstart-mp:30479 TCP	-	5 minutes	
helidon-todos-frontend	-	10.96.241.38	helidon-todos-frontend:8080 TCP helidon-todos-frontend:30080 TCP	-	2 hours	
helidon-todos-backend	-	10.96.156.41	helidon-todos-backend:8854 TCP helidon-todos-backend:30451 TCP	-	2 hours	
zipkin	-	10.96.14.89	zipkin:9411 TCP zipkin:30011 TCP	-	2 hours	
helidon-todos-cassandra	-	10.96.71.133	helidon-todos-cassandra:9042 TCP helidon-todos-cassandra:30621 TCP	-	2 hours	
atp2	app: atp2	10.96.173.84	atp2:80 TCP atp2:30609 TCP	132.145.238.4:80	2 days	
kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP	-	a month	

todos

✓ *What needs to be done?*

Koupit dálniční známky

Výměna airbag směsi

Výměna oleje

3 items left

Double-click to edit a todo
Helidon implementation of TodoMVC

 Signed in



Investigate system behavior Find a trace View Saved Trace Dependencies

Try Lens UI

Go to trace

Search

Duration: 219.923ms

Services: 2

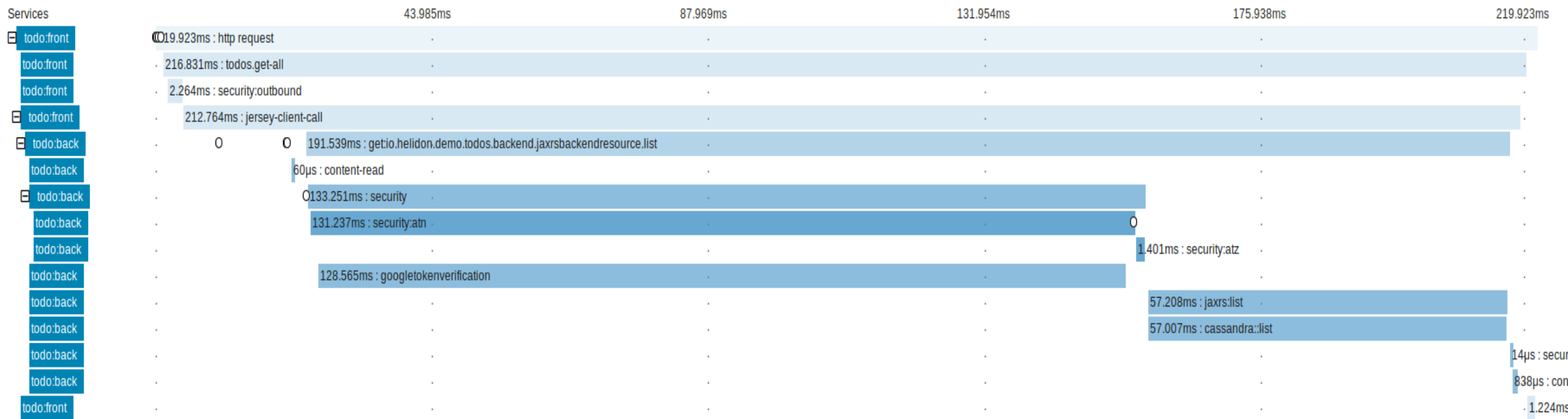
Depth: 5

Total Spans: 15

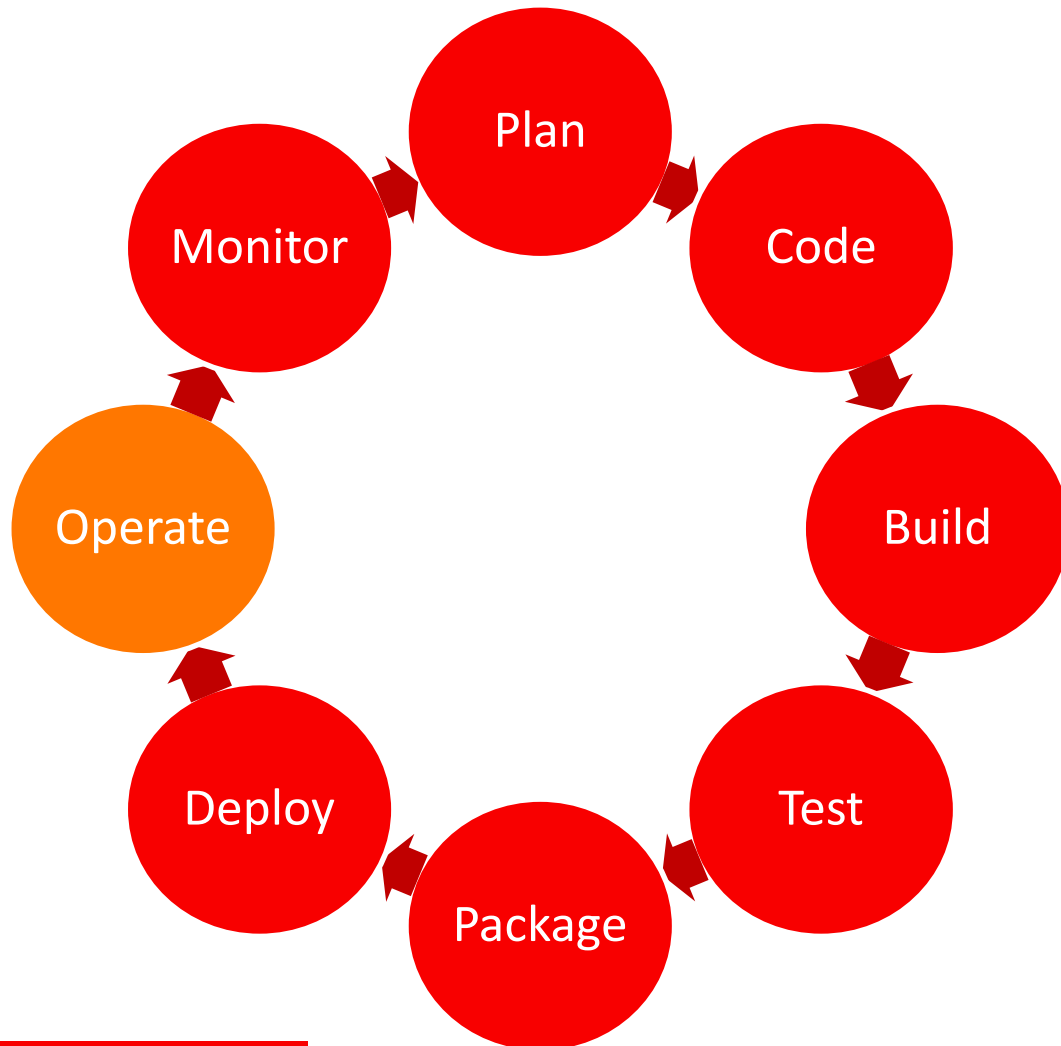
JSON

Expand All Collapse All

todo:back x10 todo:front x5

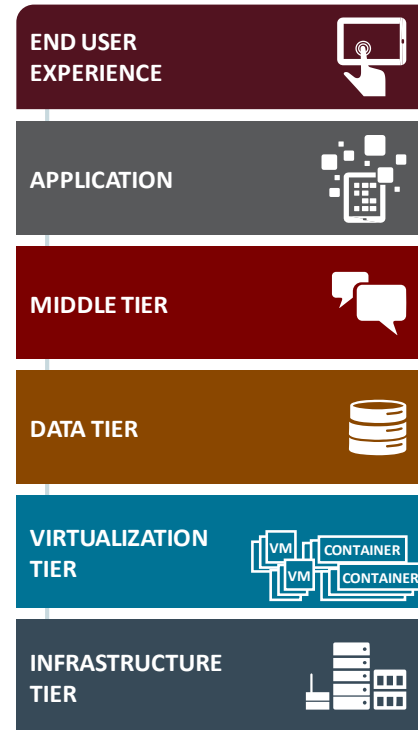
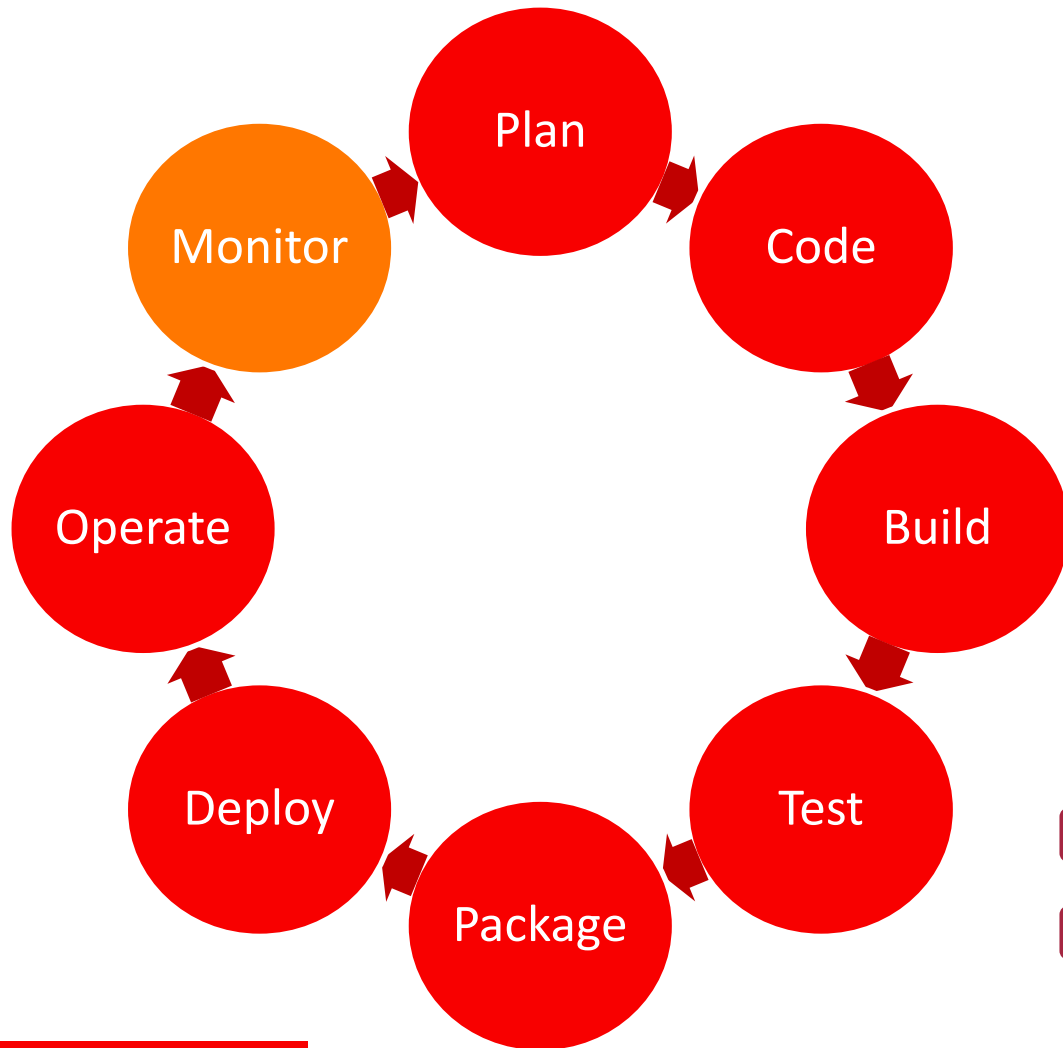


Oracle Cloud Dashboard/APIs



- (Auto)Scaling
- Patching/Upgrade
- Auditing
- Logging

Oracle Management Cloud



ANOMALY DETECTION

PREDICTION

CLUSTERING

CORRELATION



APM - Troubleshooting Across The Stack

ORACLE Management Cloud
Application Performance Monitoring

Server Request

/RideShare/checkout

Type: **SERVLET**
AppServer: oemwlsrv.oracleads.com:9001
Deployment: RideShareEar

REQUEST RESPONSE TIME

463.81 ms avg
Max 923 ms
Min 188 ms
↑ <1% prior

TIER AVERAGE RESPONSE

- External 1.88%
- AppServer 85.7%
- Database 12.42%

Last Message
Ajax Call: checkout Page: Shopping Cart has an average response time (over last hour) of 2124.55 ms; it is greater than expected value of 20.0 ms.

Normal application behavior and expected component and transaction performance are automatically learned by Oracle Management Cloud, ensuring intelligent alerting.

Diagram Metrics Links Callers Database Instances

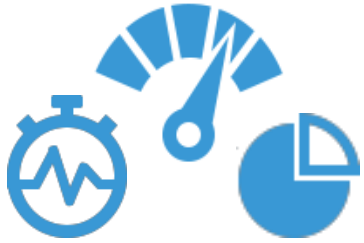
View Related Logs

Tier Time

- AppServer 87.38%
- Database 12.62%
- External 0%

Application topologies and cross-tier dependencies are automatically learned and kept up to date by Oracle Management Cloud, ensuring rapid troubleshooting.





Java Flight Recorder & Mission Control

Oracle Java Mission Control

web.1_2016_09_15_14_09_23.jfr

JVM Browser Event Types

- Flight Recorder
 - Java Application
 - Statistics
 - Allocation in new TLAB
 - Allocation outside TLAB
 - File Read
 - File Write
 - Java Error
 - Java Exception
 - Java Monitor Blocked
 - Java Monitor Wait
 - Java Thread End
 - Java Thread Park
 - Java Thread Sleep
 - Java Thread Start
 - Socket Read
 - Socket Write
 - Java Virtual Machine
 - Operating System

Overview

Interval: 1 min (all) Synchronize Selection

15/09/16 10:09:24 AM 15/09/16 10:10:24 AM

Producers

Filter Column: Producer

Producer	Total	Count
HotSpot JVM	2 min 19 s 234 ms	67

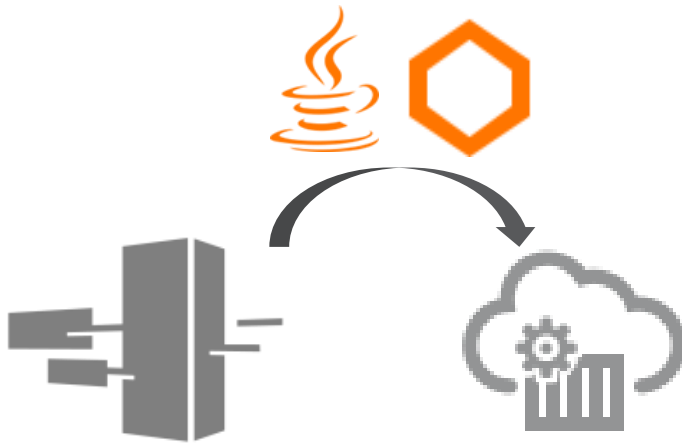
Event Types

Filter Column: Event Type

Event Type	Total	Count
Java Monitor Wait	58 s 991 ms	59
Java Thread Sleep	1 min 20 s	6
Java Monitor Blocked	241 ms 916 μs	2

Overview Log Graph Threads Stack Traces Histogram

Java applications on Oracle Cloud



Migrate Existing Apps to Cloud

- “rehosting” of existing apps to cloud
- Cloud benefits
- Connect to other Cloud services

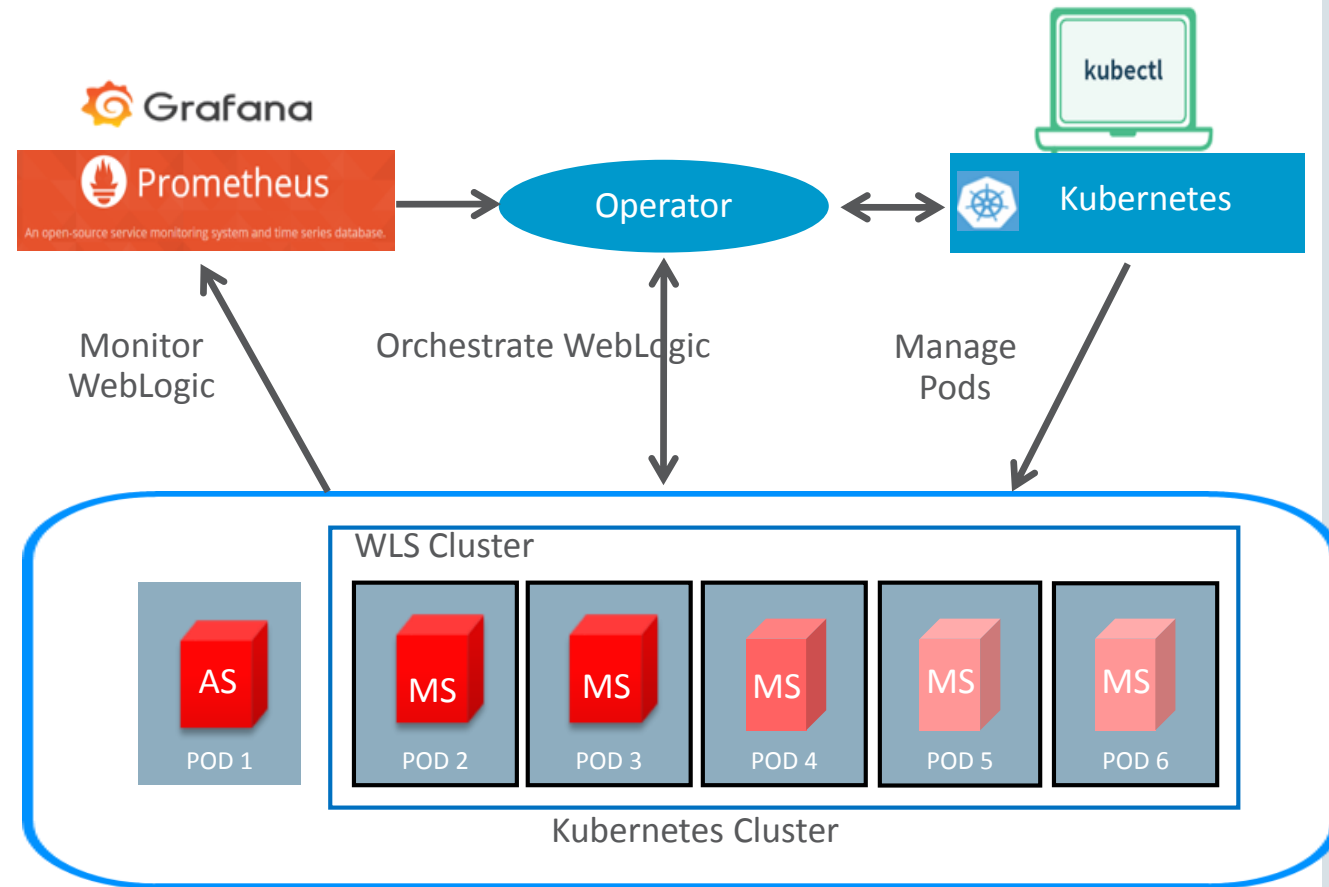


Container Native App Development

- Born-in-the-cloud apps
- Broad technology choice
- Light-weight, microservices

WebLogic Docker/Kubernetes Support

- Docker images, Dockerfiles, examples
- Helm charts to install the Operator
- Operator WebLogic best practices are followed
- Create overall WebLogic environment through Kubernetes APIs
- Manage a WebLogic domain in Docker image or PV/PVC
- Monitoring (MBean) metrics in Prometheus and Grafana
- Logs managed in the Elasticsearch and interacting with them in Kibana

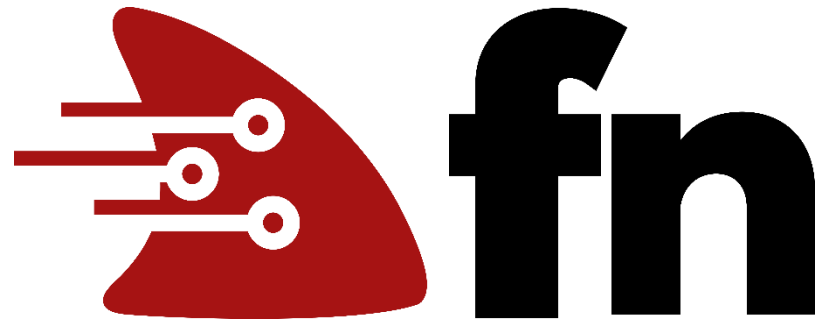


<https://blogs.oracle.com/weblogicserver/updated-weblogic-kubernetes-support-with-operator-20>

Evolution of Computing Abstractions

- Virtual Machines
 - Abstract the hardware
- Containers
 - Abstract the OS
- Serverless Functions
 - Abstract the language runtime

Fn—An open source Functions Platform



www.fnproject.io

Functions are packaged as containers—so any container can be deployed as a function

Q&A

