



# When DevOps Services met Functions

Or

How deliver high-quality products with minimal costs by using serverless options



**Francisco Moreno**

<https://www.linkedin.com/in/fmorenod>

<https://github.com/fmorenod81/whendevopsmetfunctions>

28-Feb-2023



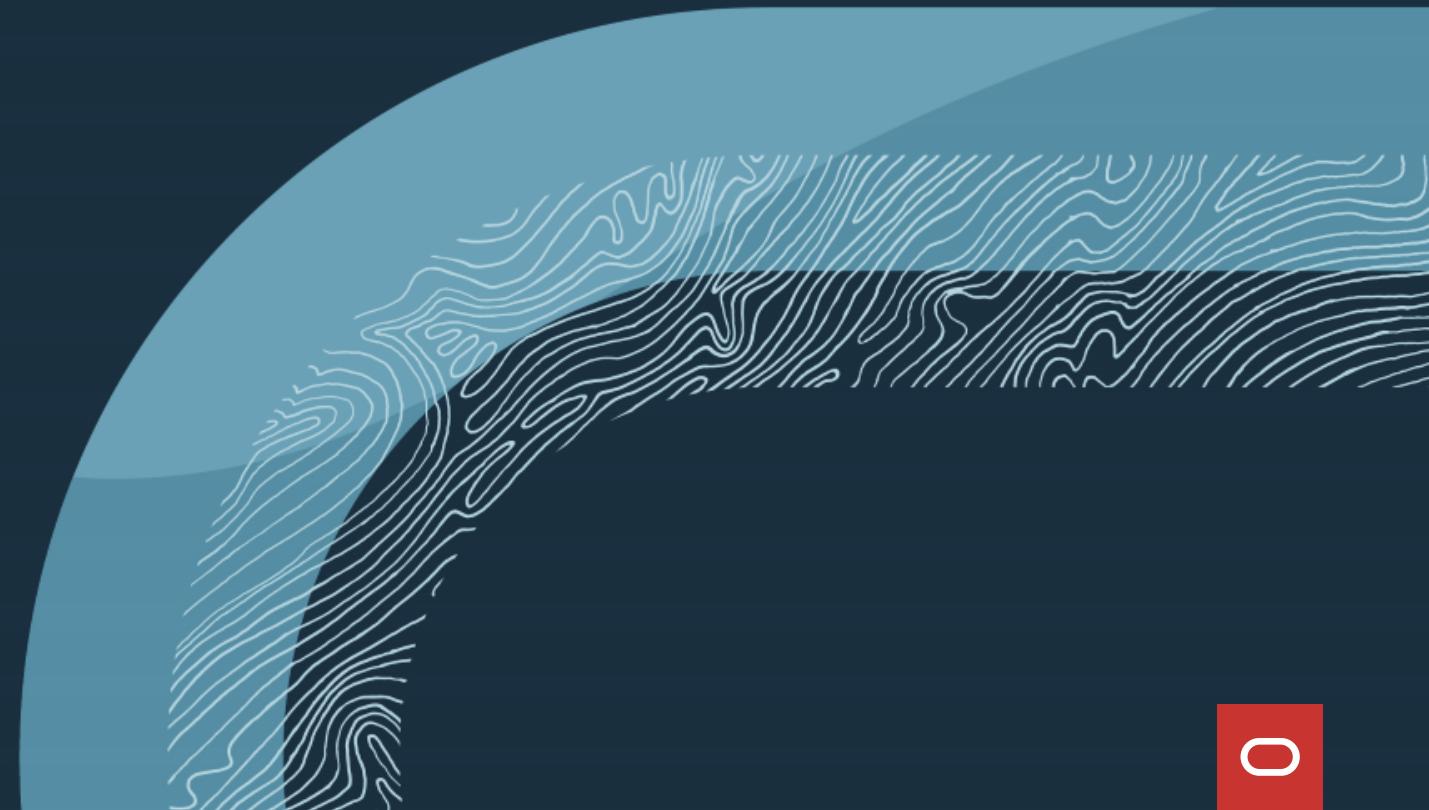
when  
harry  
met  
sally

# Agenda

50m

1. Conditions for Serverless services
2. Purpose of DevOps
3. Oracle's Vision for Cloud Native
4. CI/CD Tool: DevOps Services
5. Video DevOps with OKE
6. Serverless Compute: Functions
7. Demo DevOps with Functions
8. References

# Conditions for Serverless services

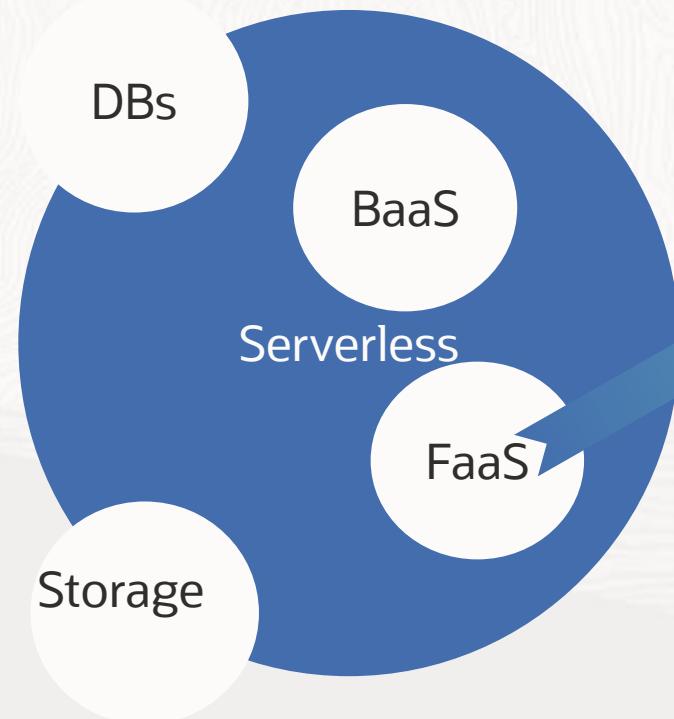


# Condition for Serverless Services

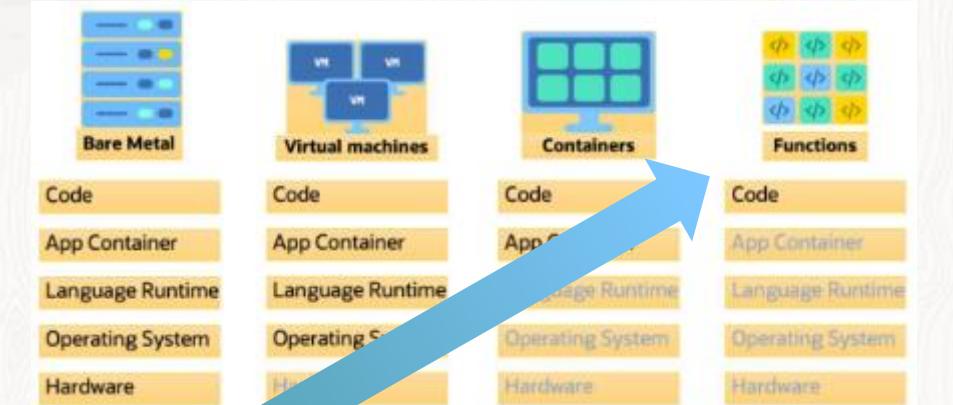


## Disadvantages

- Less control on servers
- Possible less performance (*Cold Start Problem*)
- Limited resources (Timeout, Memory > Complexity)



## FaaS is....



## Advantages:

Focus on Code



Quick development



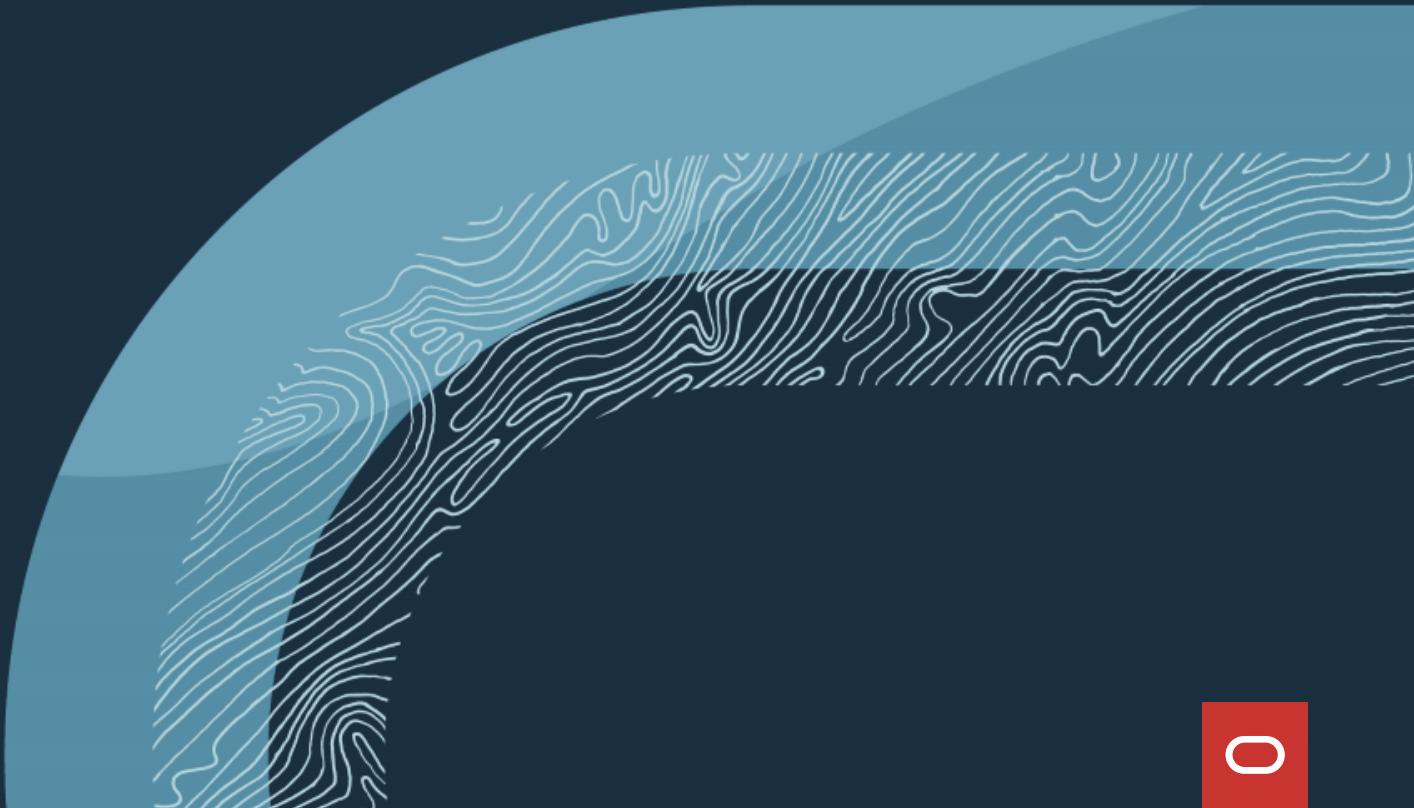
Unpredictable workload

- Unpredictable Costs, according with workload\*\*
- External Persistency

\*\* You can limit using Quotas or alert them using Budgets



# Purpose of DevOps



# DevOps Concepts for Dummies



The **business people** come to have some ethereal thing (Idea)

A **designer** create a template (**Code**)

Add raw material (Dependencies Injection: Libs/Framework)

Build it in a pipeline (**Build stage on Build Pipeline**)

A **line of production** deliver a product and ship it (**Artifacts**)

A **tester** probe its quality (**Test stage on Build Pipeline**)

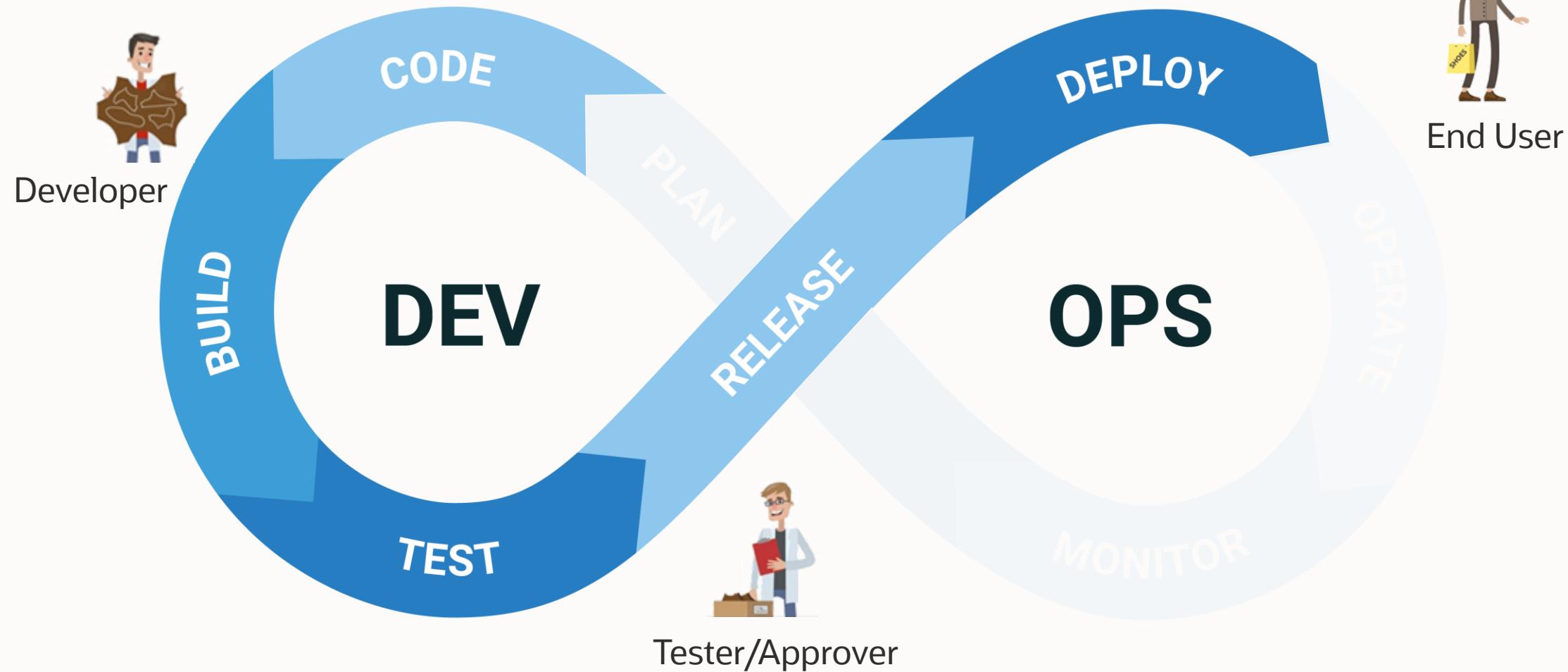
Assure circumstances for it (Configure on **Environments**)

A **line of production** deliver the product to several stores (**Deployment Pipeline**)

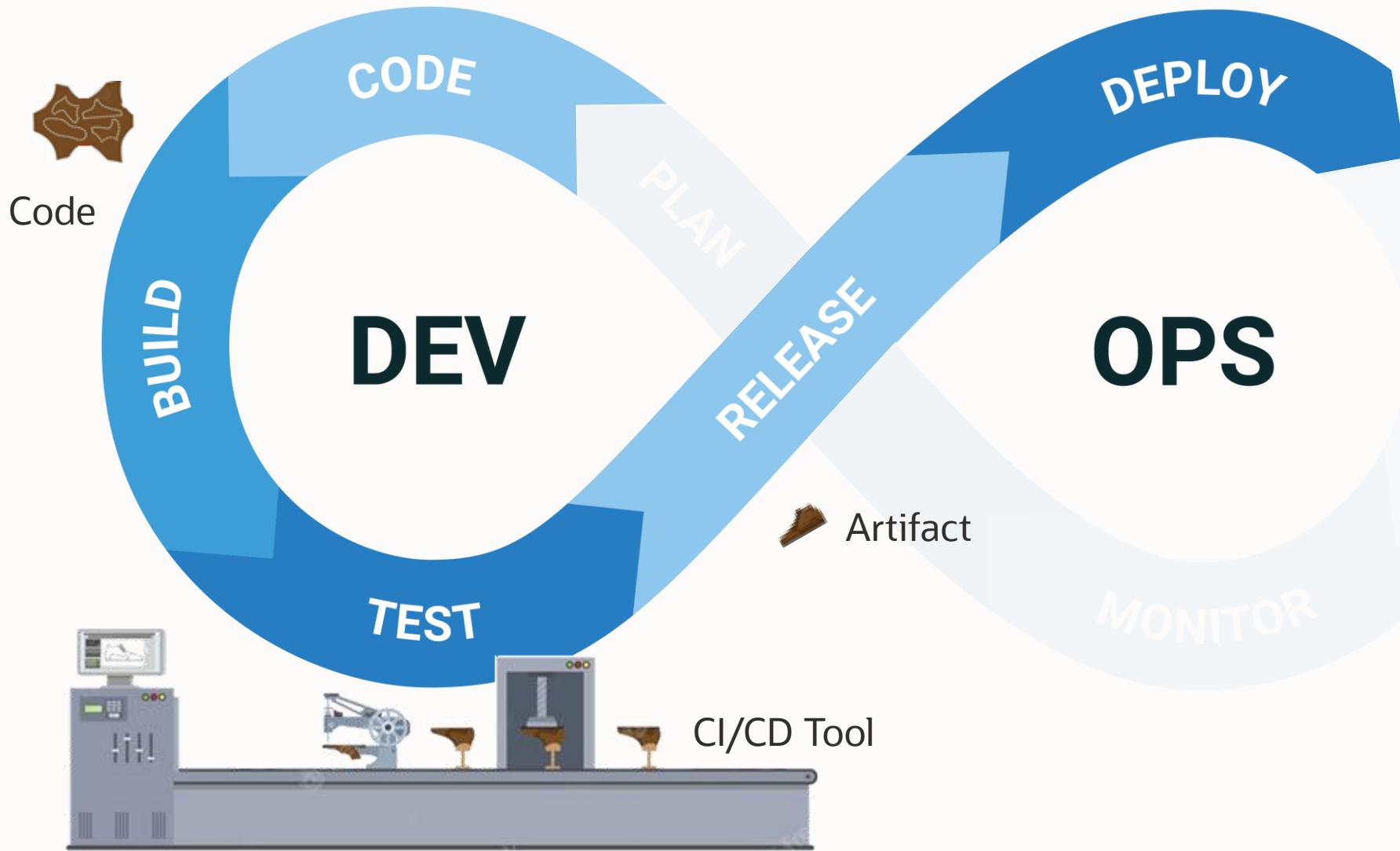
The store offer the **product** (API Exposition)

The **user** expects the product fulfils his needs, and the **user** only buys the amount that he needs and he uses it when he needs it only (**Serverless Computing**)

# Idea of DevOps



# Idea of DevOps



Final Product

Store

# What are cloud-native applications?

---

Modern applications designed as:

- Loosely-coupled **microservices**
- That communicate over **APIs**
- Often packaged as **Containers**
- Developed & released **using modern DevOps** patterns & tools

But -- **there's more...**



## Innovate Faster!

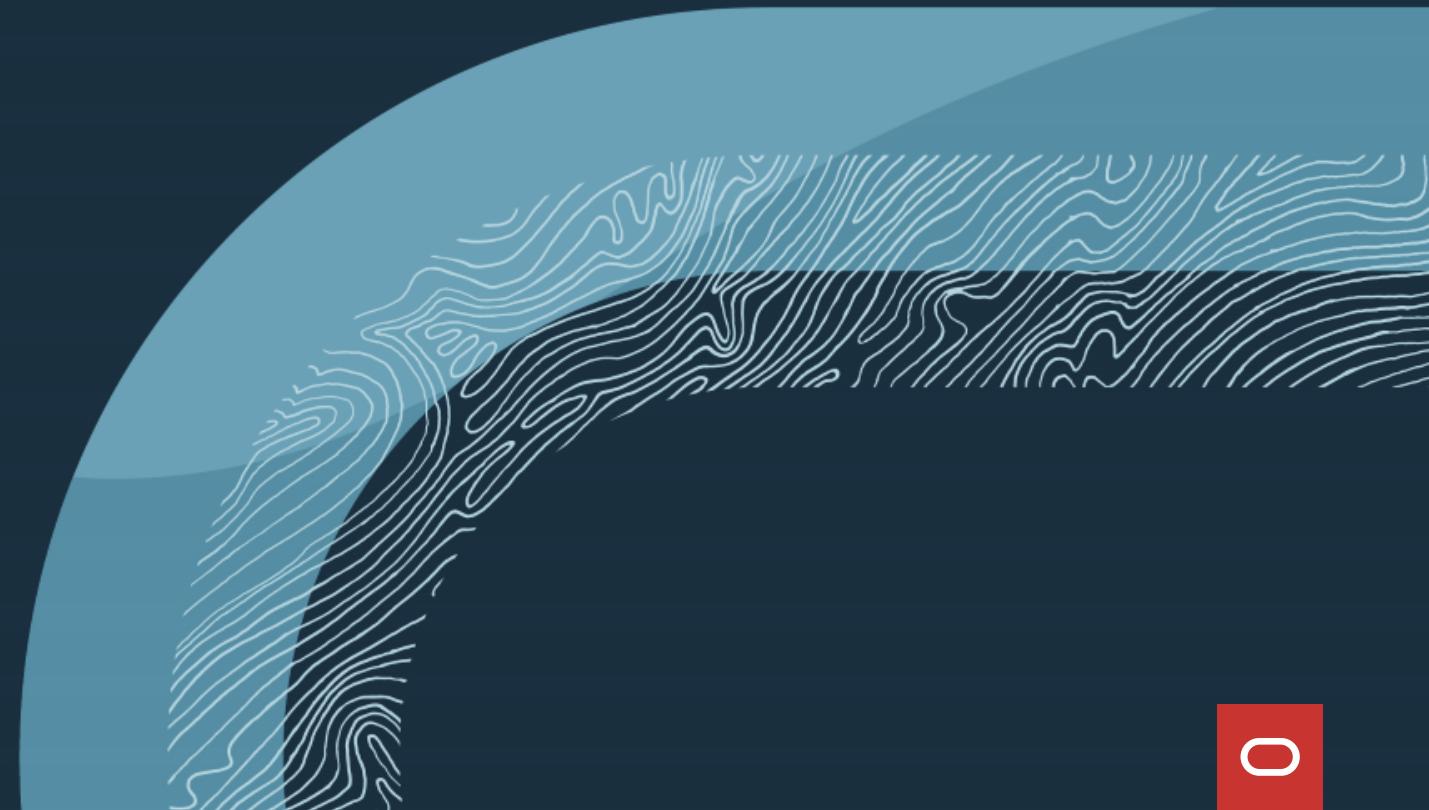
- Improve TTM and Lead-time
- Improve release cadence & reliability
- Improve developer productivity
- Ensure scaling + HA more easily
- Less management overhead



Loosely-coupled architectures are the **#1 predictor of DevOps success**.  
Each service is developed, deployed, and operated independently – so you can move faster!



# Oracle Portfolio for Cloud Native



# What is Cloud Computing?



"is a model for enabling **ubiquitous**, **convenient**, **on-demand** network access to a **shared pool** of **configurable computing resources** (e.g., networks, servers, storage, applications, and services) that can be **rapidly provisioned** and **released** with **minimal management effort** or service provider interaction"

NIST Definition. SP 800-145

- 1. Just user' needs**
- 2. Available**
- 3. Automatic options**

# OCI Regions – Global Footprint

January 2023: 41 Regions, 8 planned; 12 Azure Interconnect Regions

100%  
renewable  
**energy** used for  
Oracle Cloud  
data centers in  
Europe (today);  
All regions (by  
2025)



AWS: 31 Regions  
Azure: 46 (60?) Regions  
GCP: 35 Regions



# Alliance for Hosting in a New Public Region - Colombia

Consequences for a Public Region for a country and surroundings

**bnamericas**

NEWS

## Claro Colombia investing US\$25mn in datacenter for Oracle cloud

BnAmericas  
Published: Saturday, August 20, 2022

Claro is part of America Movil, 6th largest mobile network operator: 18 Countries in America and 7 in Europe.



TECNOLOGÍA

## Claro Colombia invertirá US\$200 millones para impulsar la productividad

Claro Colombia will invest US\$200mn to boost productivity

jueves, 18 de agosto de 2022

f    t    in     GUARDAR

### Press Release

## Oracle and Claro Partner to Expand Global Cloud Services in Colombia

Claro will offer Oracle Cloud Infrastructure as part of its managed cloud services

Collaboration will enable organizations to take advantage of Oracle Cloud Infrastructure's high performance, built-in security, and lower costs

Partnership is designed to modernize IT infrastructure and stimulate the economic recovery in Colombia

Austin, TX and Bogota, Colombia—July 14, 2022

Today Oracle and Claro announced a partnership to jointly offer Oracle Cloud Infrastructure (OCI) services to public and

or organizations and enterprises in Claro will become the host partner for loud region in Colombia, which will o to offer OCI platform services along sional and managed services to help adopt cloud solutions. In addition, to re its customer service and billings

Claro will migrate over 100 on-premises servers running mission-critical workloads to OCI.



100+ Claro' on-premises servers  
10 years agreement

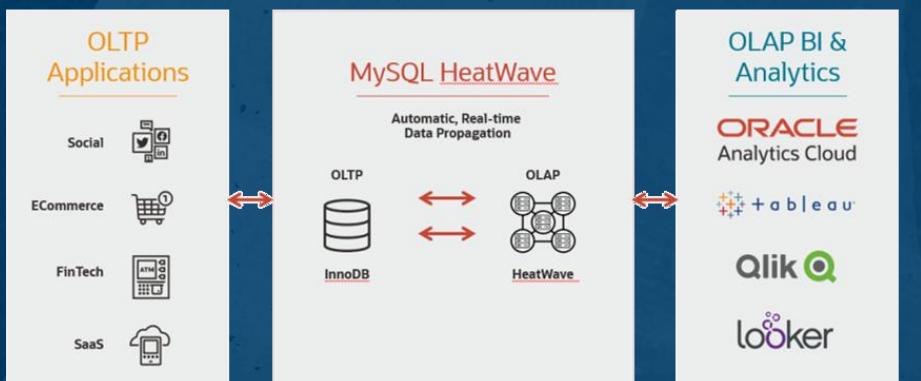


# Multicloud Options

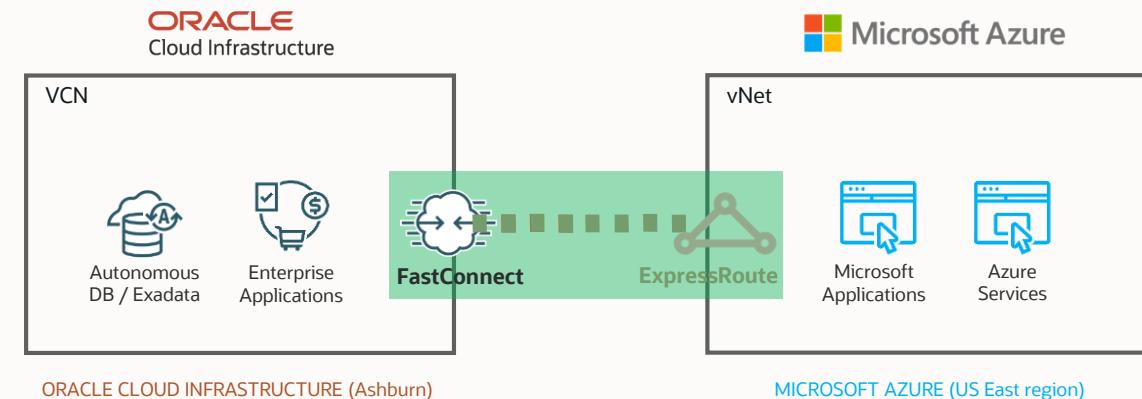
## Oracle Database Services for Azure

The screenshot shows the Oracle Database Service for Azure dashboard. It includes a left sidebar with navigation links like Home, Overview, Tags, Resources, and Backups. The main area displays metrics such as Activity log, Access control (IAM), Tags, and Diagnose and solve problems. A central panel shows Oracle services like Autonomous Database, Exadata Database, and Base Database, along with options to Create a resource, Navigate, and Useful links.

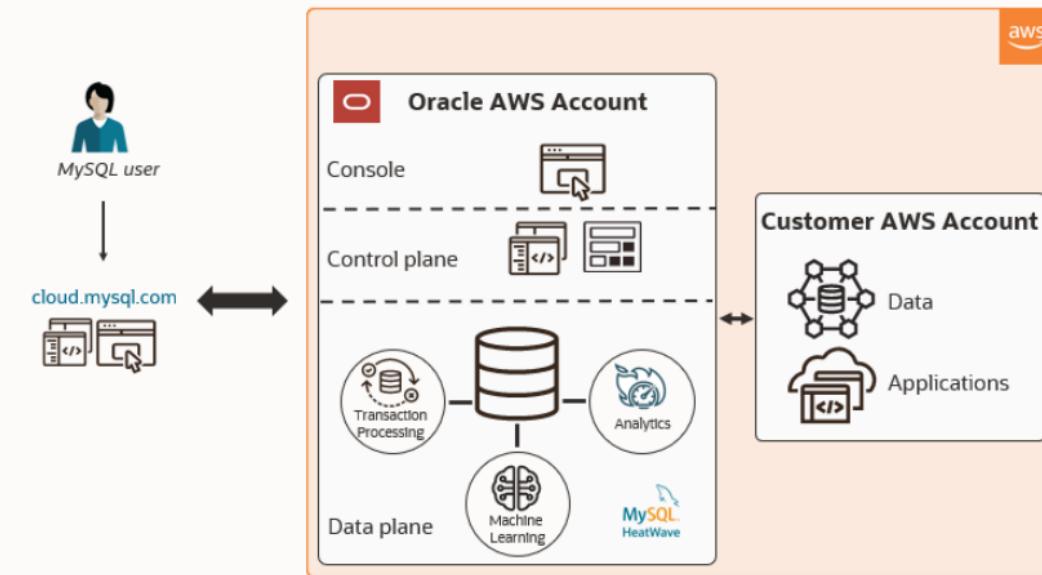
MySQL Heatwave is...



## Azure Interconnect



## MySQL Heatwave on AWS



# Gartner Magic Quadrant for Cloud Infrastructure and Platform Services

## Oracle moves from Niche to Visionary

*Oracle Cloud Infrastructure is mainly focused on lift-and-shift, HPC and hybrid workloads, though OCI endeavors to have broad capabilities outside of Oracle-focused applications (i.e. Cloud Native).*



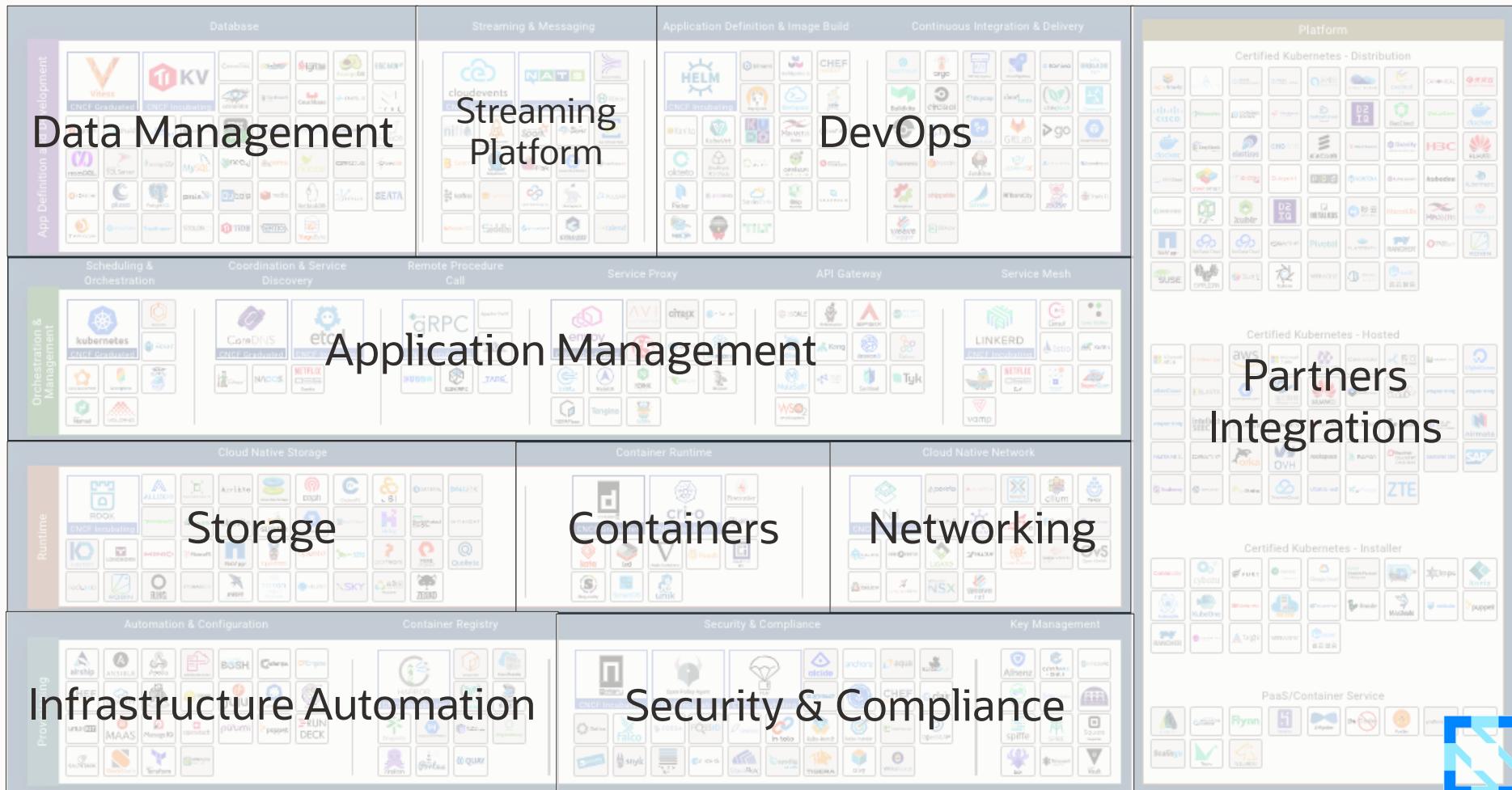
Report Oct/2022

# What does each CSP offer?

	Oracle	AWS	Microsoft	Google
Functions	OCI Functions	AWS Lambda	Azure Functions	Google Cloud Functions/ Google Cloud Run *
CI/CD	Oracle DevOps Service	AWS CodeBuild AWS CodeDeploy AWS CodePipeline	GitHub Azure DevOps Azure Pipelines Visual Studio App Center	Cloud Build Google Code Deploy Tekton



# Oracle vision for Cloud Native is open source, however...



# Cloud Native Landscape

CLOUD NATIVE COMPUTING FOUNDATION



# Open and flexible: Choice matters

Full support for OSS, 3rd party technologies, and ecosystem tools enables innovation, portability, and ensures optimal TCO for *anything* running on OCI

## Managed services based on upstream open source



kubernetes



docker



Terraform

## Deploy what you want, ensuring seamless operations and lowest TCO



Redhat,  
Ubuntu,  
CentOS,  
Debian,  
SUSE, Oracle



Windows  
Server



## Native integrations with the dev tools you're used to



GitHub



ANSIBLE



ANSIBLE



redis



mongoDB



kubernetes



HELM



Terraform



ATLASSIAN



## Communities we contribute to



CLOUD NATIVE  
COMPUTING FOUNDATION



Java



OPEN  
CONTAINER  
INITIATIVE



THE  
LINUX  
FOUNDATION



CD  
FOUNDATION



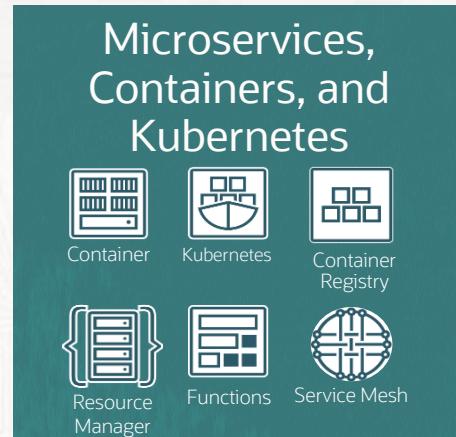
ECLIPSE  
FOUNDATION



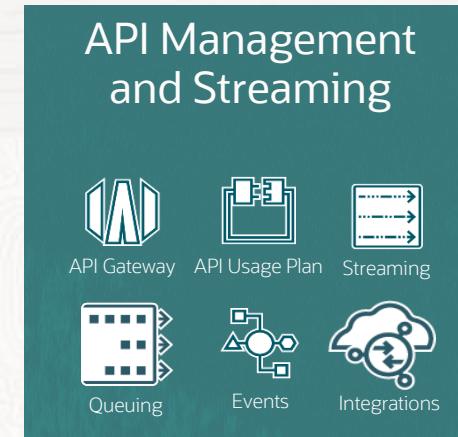
# Application Modernization on OCI



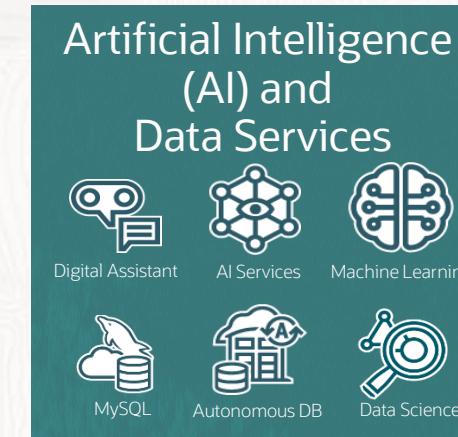
Dev  
Productivity



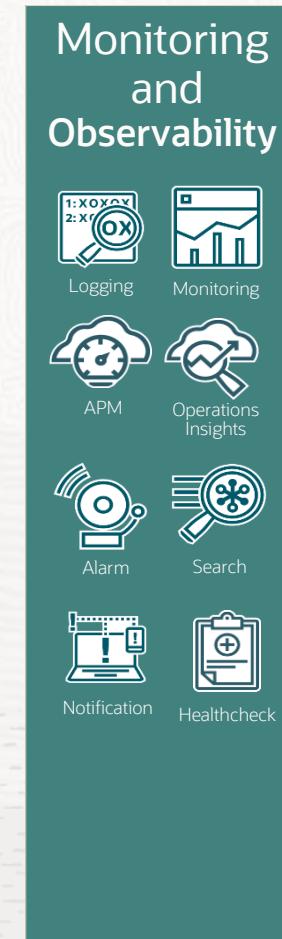
## Microservices, Containers, and Kubernetes



## API Management and Streaming



## Artificial Intelligence (AI) and Data Services



## Monitoring and Observability



OCI DevOps + Freedom of Choice with Integrated Ecosystem

Open Frameworks, 100% Upstream Compatible



Operations  
Reliability

Flexible Infrastructure – Configurable CPU / Memory Ratio



IAM, Security & Governance



# Oracle's Modern App Dev Framework



**Core requirements**



**Design principles**



**Architecture patterns**



**Technology recommendations**



**Security and compliance**



**Availability**



**Scalability**



**Performance**



**Agility**



**Observability**



**Resiliency**



**Cost Optimization**



**Portability**

## Common requirements that apply to any modern application

- Security Policies & Best Practices
- 24/7/365, no downtime.
- Scale gracefully
- Low latency and high throughput to improve UX
- Digital Enablers: DevOps/Cloud Native Services
- Metrics to improve Reliability
- Recover gracefully
- Total Costs
- Easy migration

# Modern Application Design Principles

We're your partner on your path to modernization



## Design principles

Use lightweight open-source frameworks and mature programming languages



## Architecture patterns

Build apps as services that communicate through APIs



Functions

Package and ship apps as containers



## Technology recommendations

Automate build, test, and deployment



DevOps

Use fully managed services to eliminate complexity across application development, runtimes and data management

Keep application tier stateless

Use converged databases with full featured support across all data

Instrument end-to-end monitoring and tracing

Eliminate single points of failure through automated data replication and failure recovery

Implement a defense-in-depth approach to secure the app lifecycle

# Oracle Cloud Infrastructure Reference Architectures

Access to OCI best practices framework

Cloud Architecture Center 

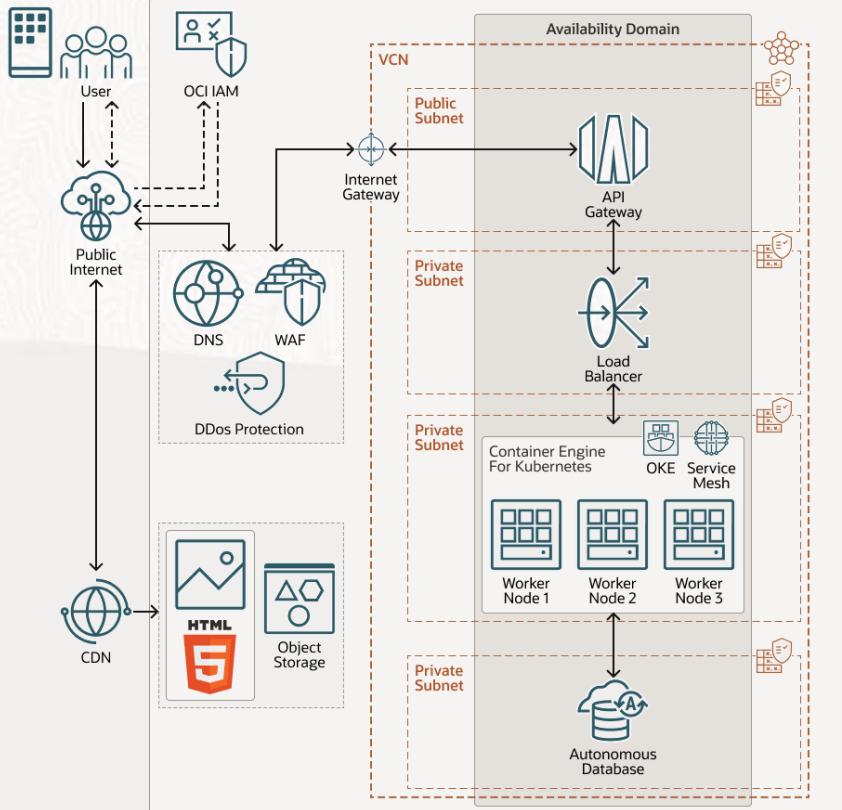


Image: Modern App Development - Web and Mobile



## Architecture patterns

Access OCI reference architecture for:

- Cloud best practices including availability, performance, security, and costs
- Access to OCI best practices framework
- Migration automation to OCI
- Latest deployments updates from customers and partners who are using OCI to drive innovation
- 200+ references architectures.



## Technology recommendations

### Automation Available

You can deploy this pattern using downloadable code or automated provisioning, as described in the Download or Deploy section.

[Learn more](#)

 [Deploy to Oracle Cloud](#)

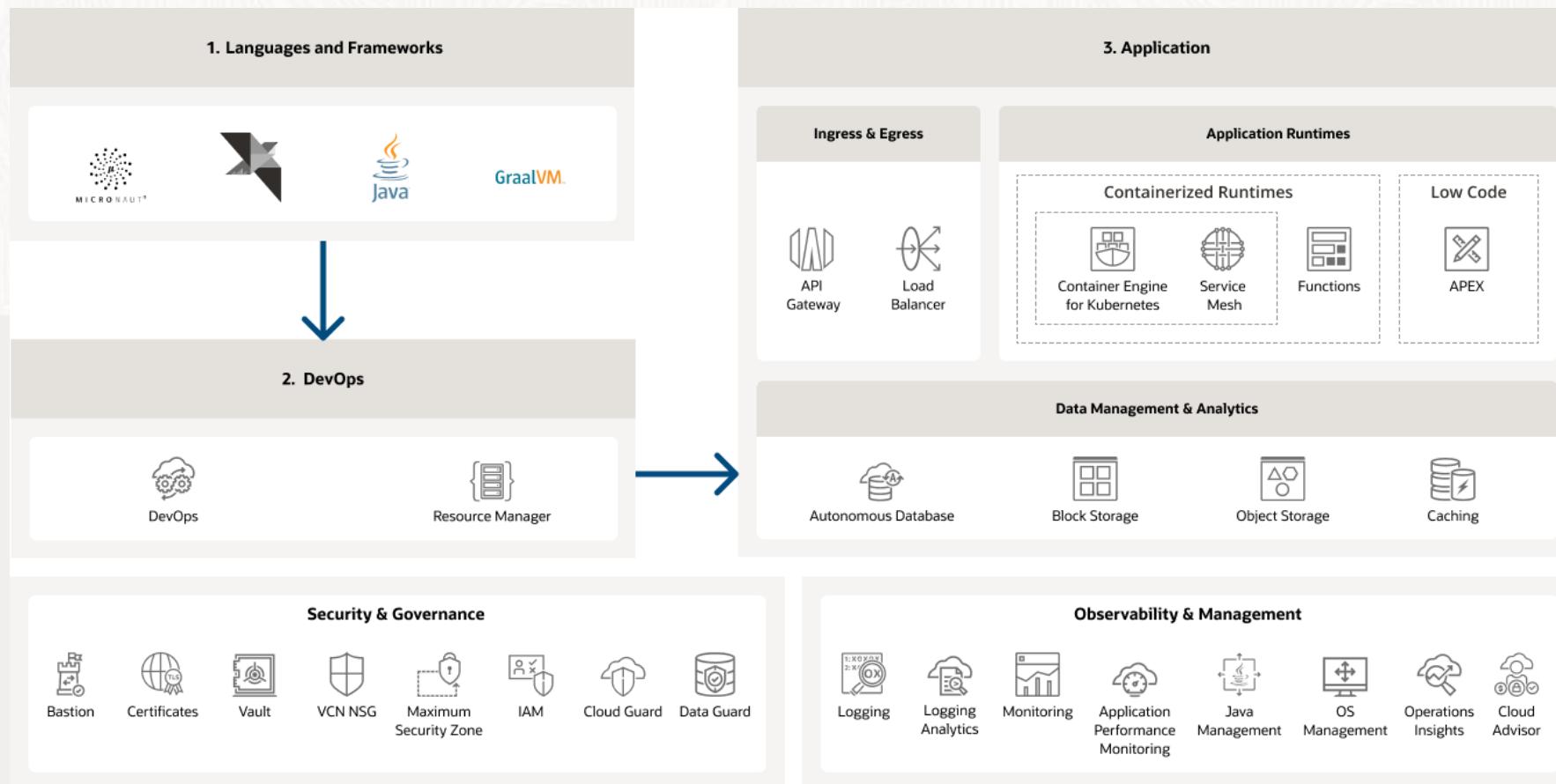
 [Go to GitHub](#)



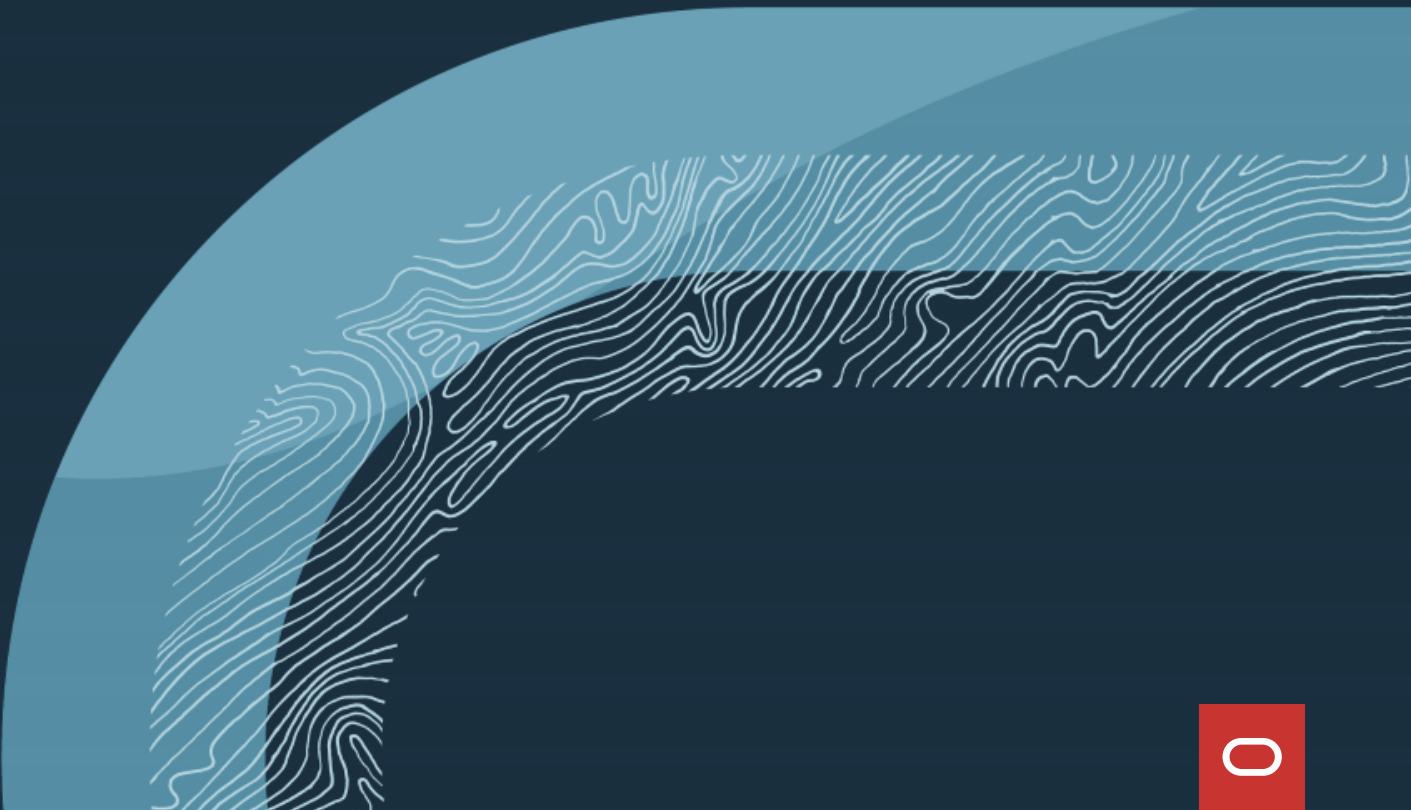
# Oracle's Modern App Dev Framework



## Technology recommendations



# DevOps Services



O

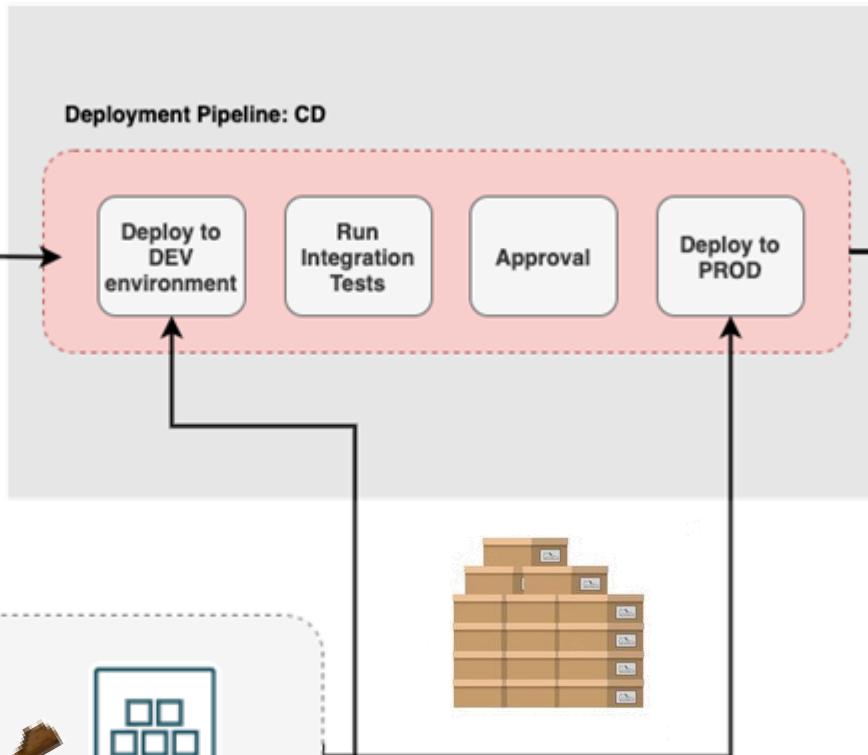
# Complete CI/CD Platform



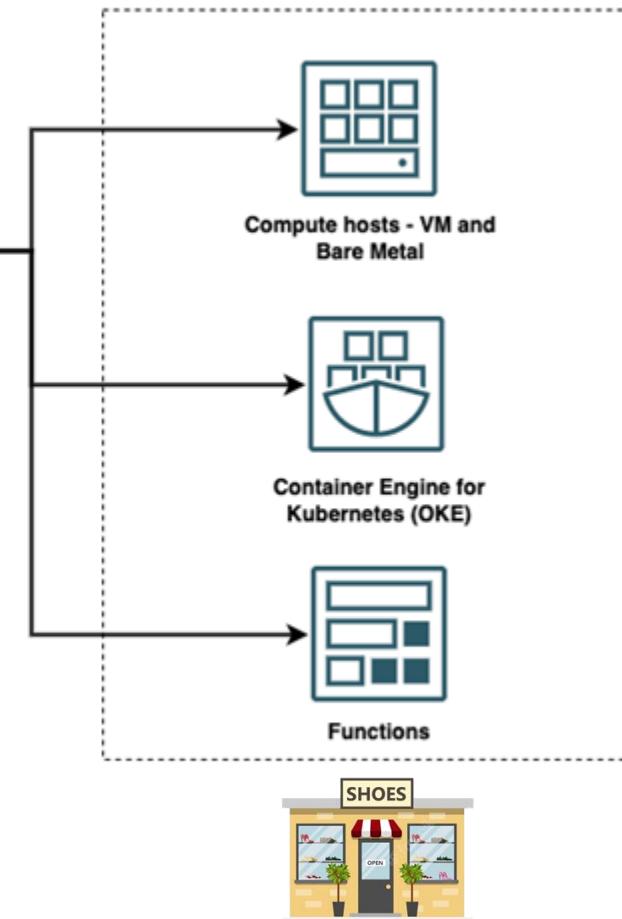
Jenkins Pipeline



OCI DevOps service



Customer Tenancy



OCI Artifact Repositories



# Integrations: Jenkins, GitHub, GitLab, Spinnaker

## For Code Repository:

Mirror a GitHub, GitLab or Bitbucket repo to your OCI Code Repository to speed up builds

## For CI Tools:

Easier to migrate existing pipelines – keep your current Jenkins file, GitHub Actions

- Deliver Artifacts to OCI Artifact Registry
- Trigger a Deployment Pipeline

## For CD Tools:

Spinnaker: open source multi-cloud orchestration for Kubernetes and VMs

- Deploy to OCI platforms



**Jenkins**



## Features of General DevOps



### Automation

Simplify and speed up software development



### Security

Take advantage of Cloud Security



### Governance

End to end Visibility



### Flexible

Integrate existing CI/CD tooling.



## Differences with DevOps Services

### Cloud Native

Using Serverless (cheap) or Free Services to simplify

### Cloud Security Services

IAM on DevOps. Scan and sign images

### OCI Integration

Observability, logging and Governance.

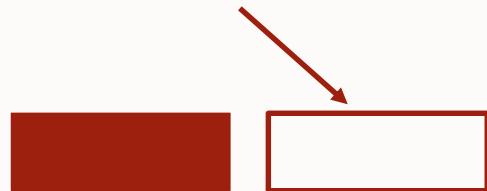


# DevOps Continuous Integration (CI)



## Serverless, Scalable

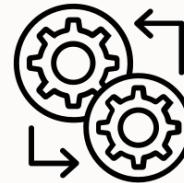
No instances to manage



## Release Strategies

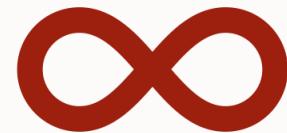
Reduce downtime, faster recovery

# DevOps Continuous Deployment (CD)



## Integrated

Works with your resources



## Integrated

Connect your workflows



## Complete

Native CI/CD Platform



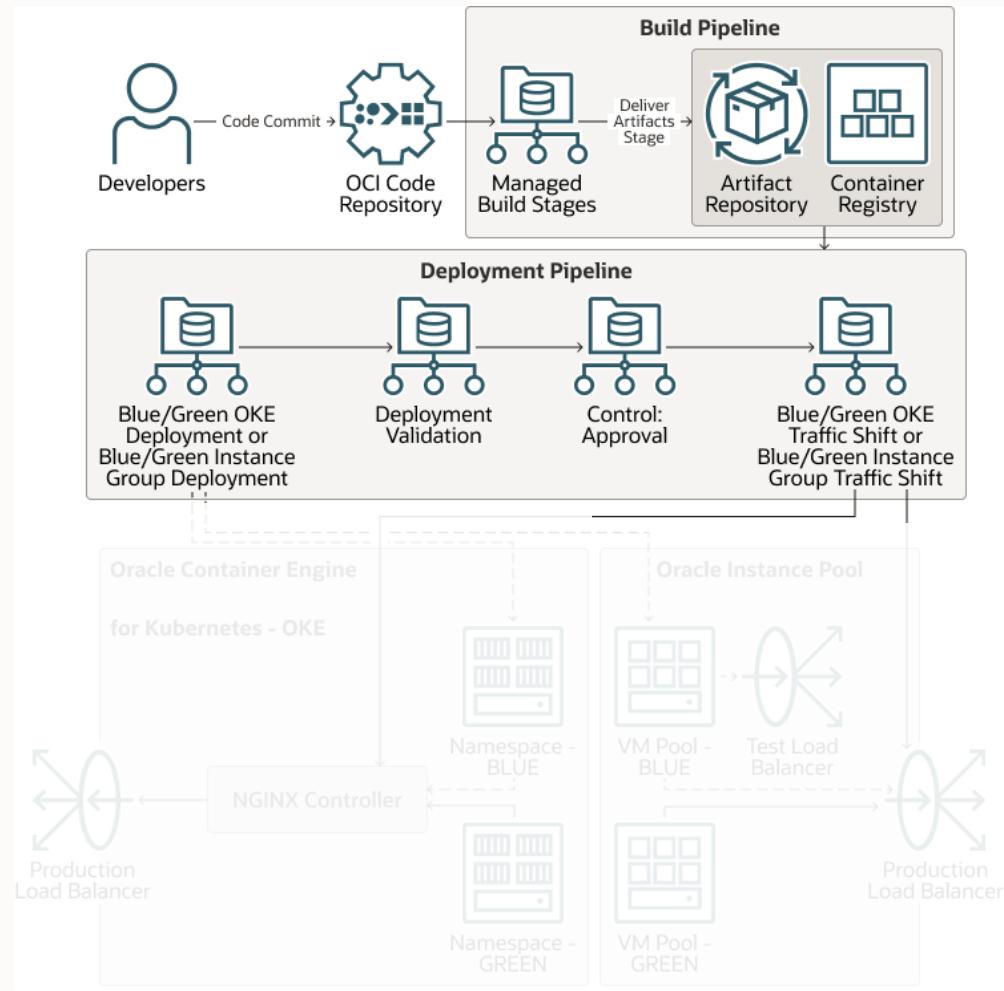
## Rollback

Recover from errors



# Deploy Strategies:

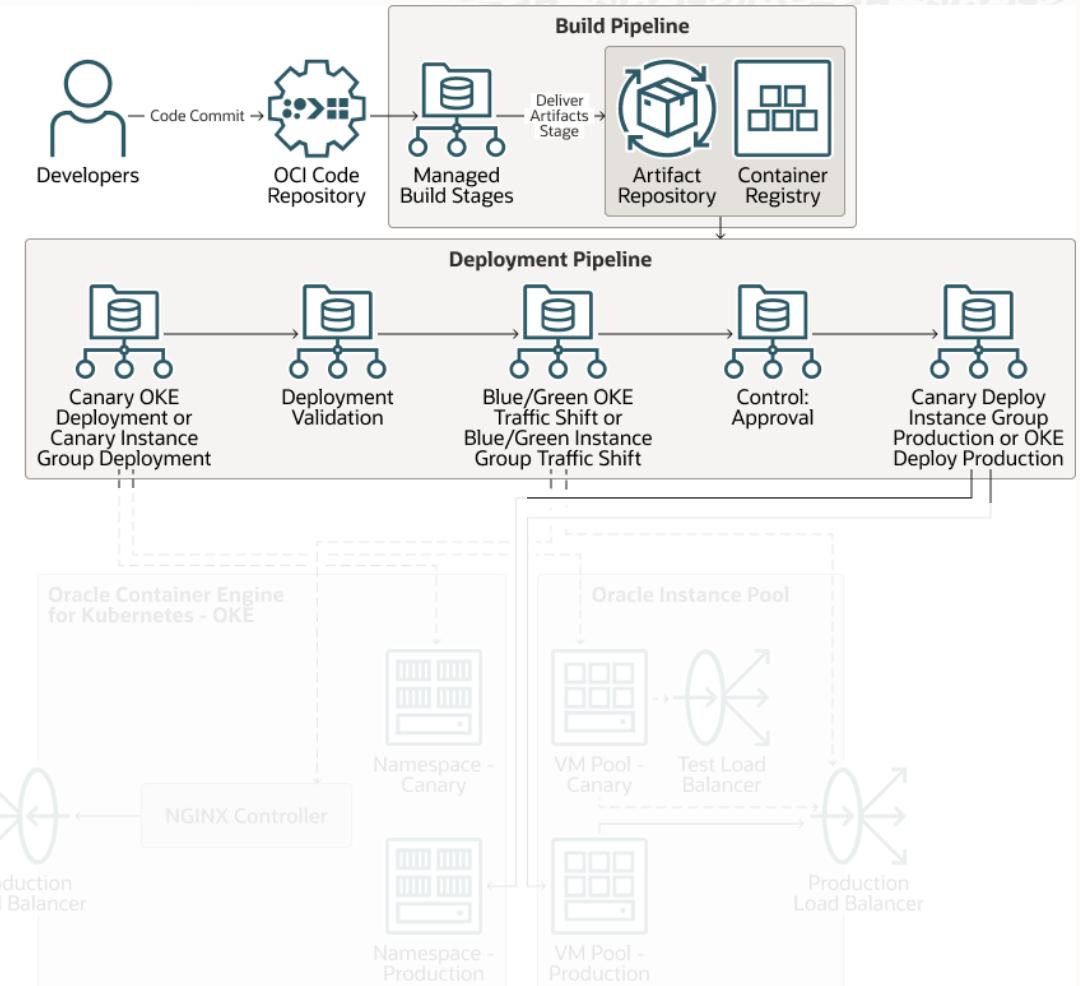
## Blue/Green Deploy



OKE

BM/VM

## Canary Release

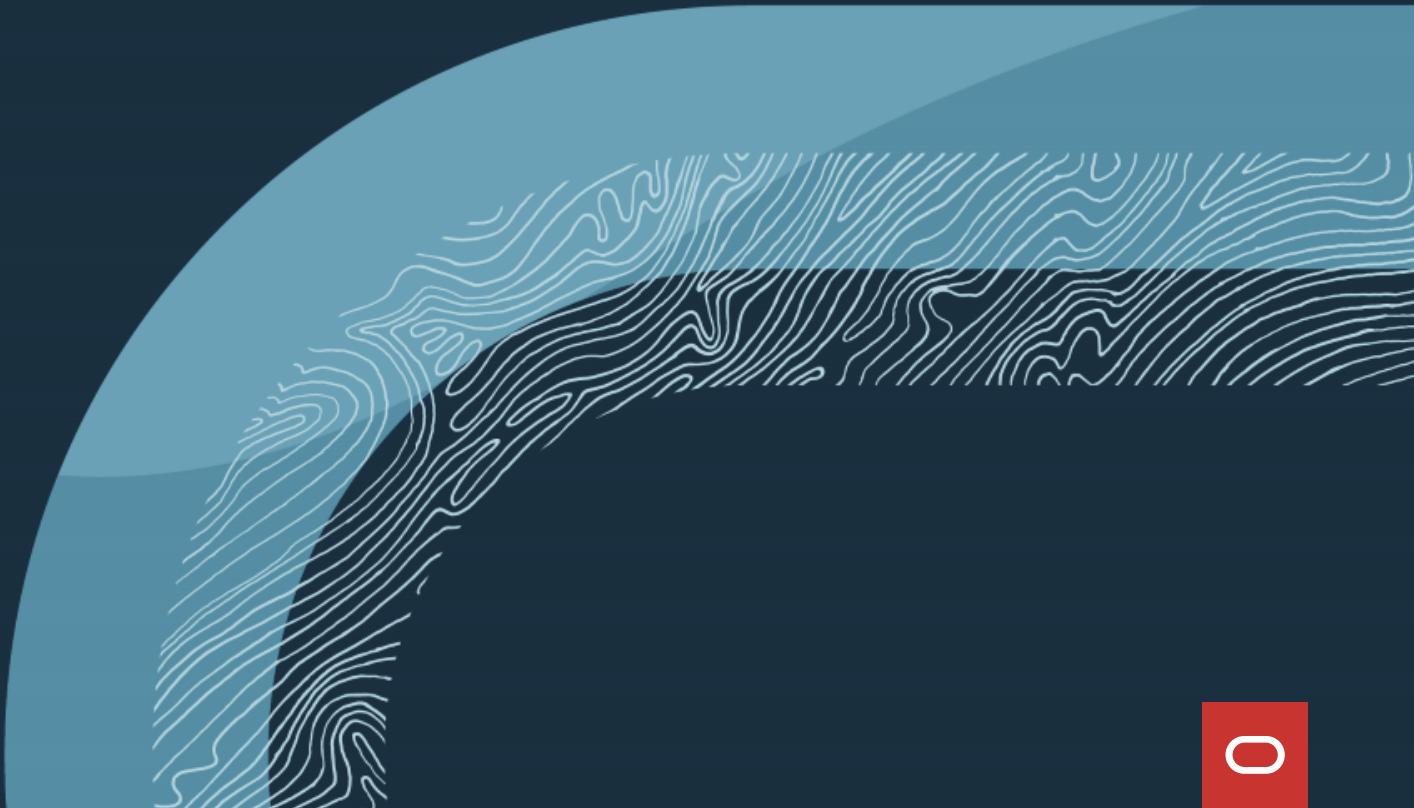


OKE

BM/VM

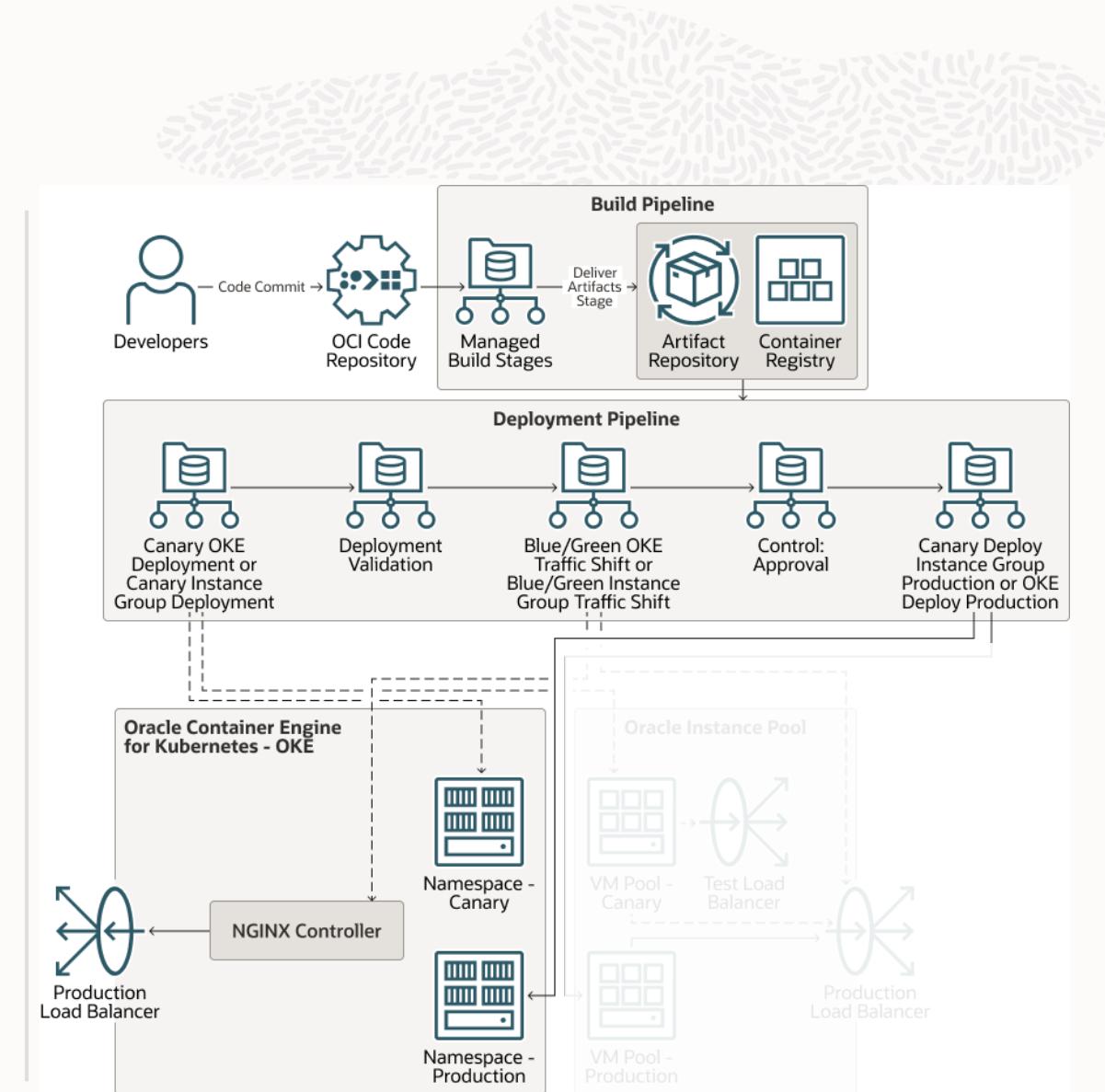


# Video DevOps with OKE

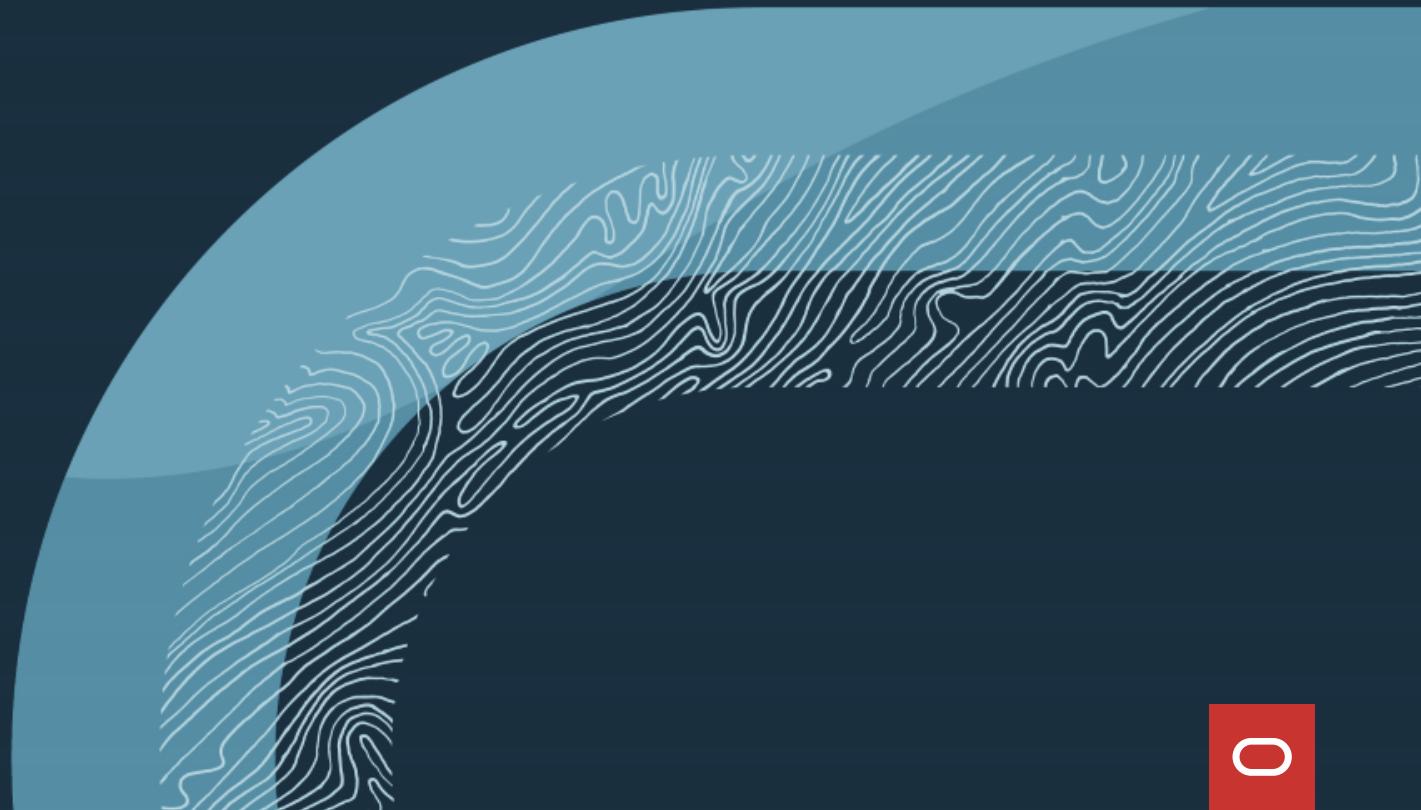


# Steps

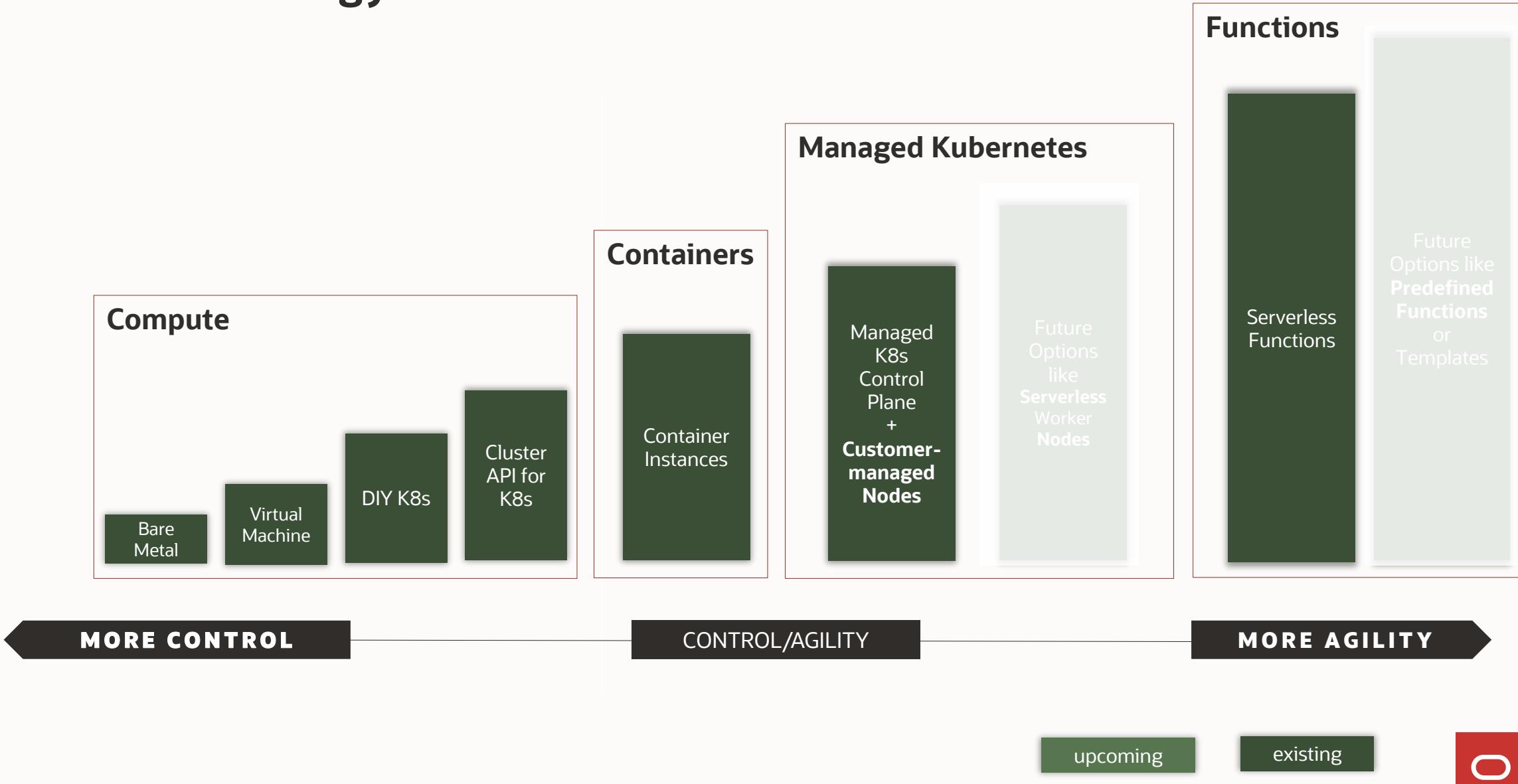
- New Deploy – Version 0.0
- Check canary release (%)
- Approve the change – Version 1.0
  
- New Deploy – Version 1.0
- Check canary release (%) – Version 1.1
- Reject the change - Version 1.0
  
- Manual Rollback
- Select deployment –Version 0.0



# Functions



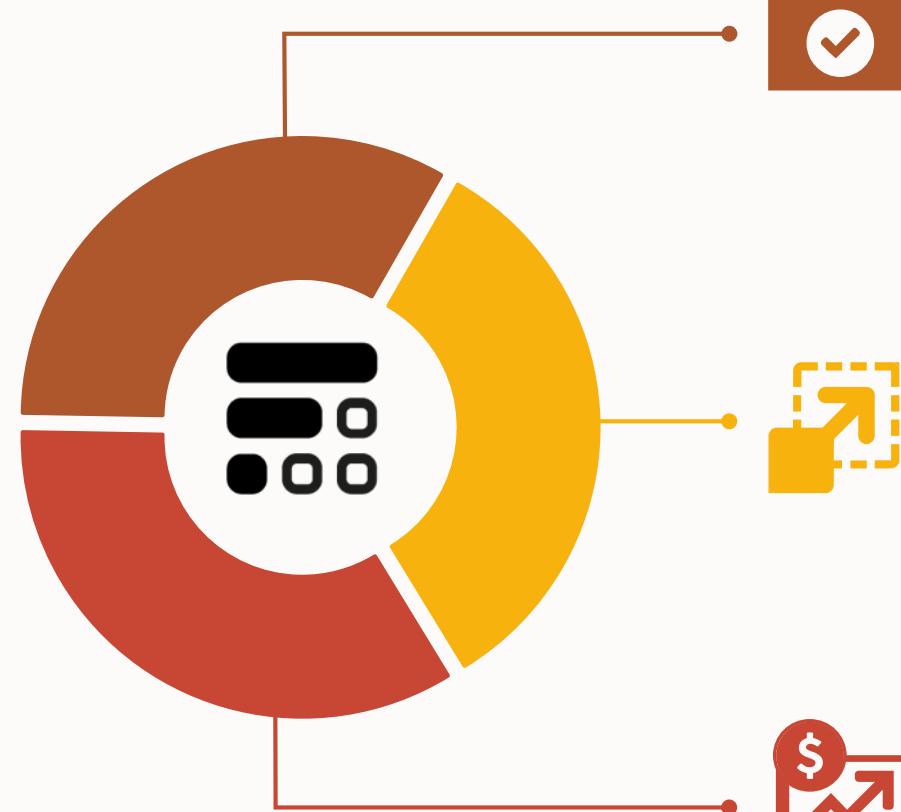
# Container Strategy: Flex Runtimes



# Oracle Functions

Functions-as-a-Service

Container-Base Open  
Source Initiative  
Cloud Integrated



**Pay per use**  
Pay for execution,  
not for idle time

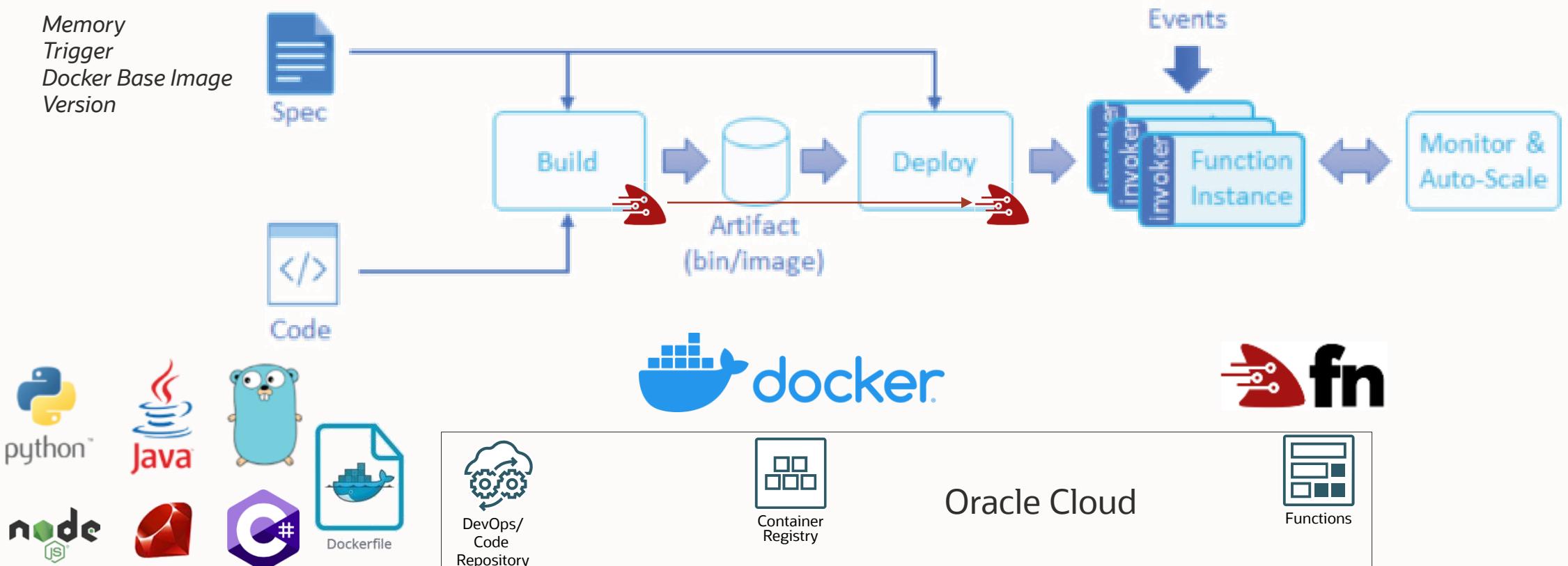
**Autonomous**  
Platform auto-scales functions  
No servers to provision, manage

**Event-driven**  
Oracle Cloud Infrastructure  
triggers to run your code



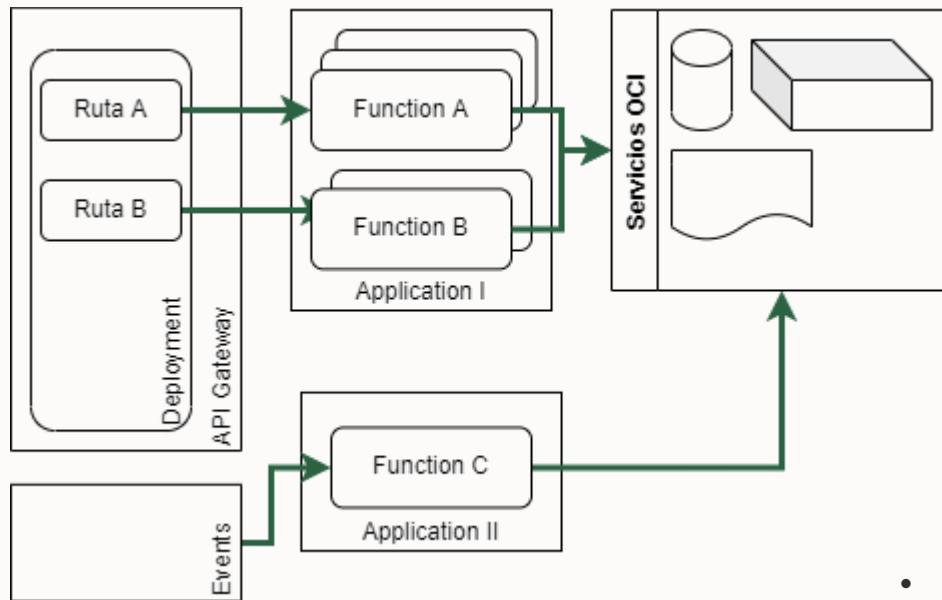
# How it does it work?

**Options:**  
Local  
DataCenter  
OCI

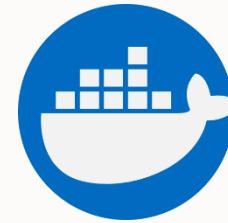


# Oracle Functions: Features and Architectures

1. Type of License
2. Scope
3. Description
4. Pricing
5. Input

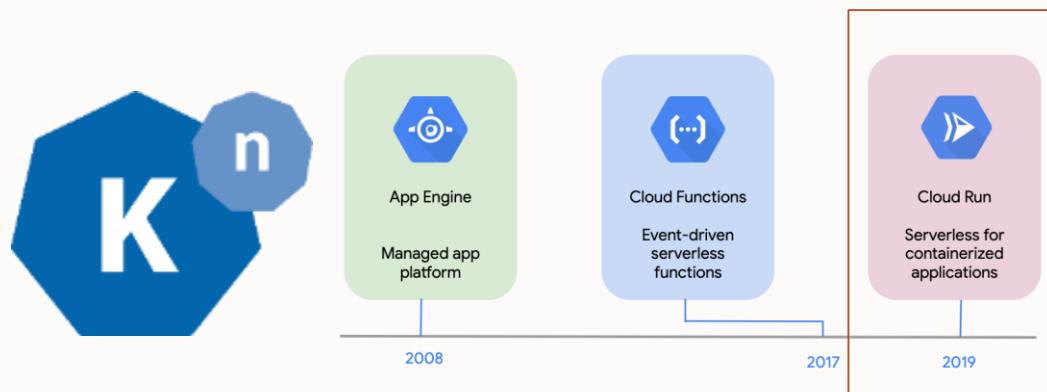


1. Open Source – Apache 2.0 License
2. Multisite - Multicloud
3. Improve DevOps experience and “On my machine works”
4. Container Native – Autoscaling – Pricing by invocations and GB-sec consume. Agnostic to orchestrator and language (hotwrap)
5. CNCF Events.



- **Application**  
Logical grouping of isolated functions, with a common context (cfg vars) and resources (networks, signature verification, metrics and logs)
- **Function**  
Imagen Docker (Almacenada en el Container registry)  
Metadata: Localización de la imagen, memoria, timeout, parámetros de configuración

# Comparative with other open source options



1. Open Source / CNCF Support.
2. Multisite – Multicloud.
3. Kubernetes (Container) Native, autoscale to 0.
4. Pricing due to Kubernetes cluster. GCP 2 models.
5. Cloud Events, HTTP(S) support.



1. Open Source / Commercial Support.
2. Multisite – Multicloud.
3. Kubernetes/OpenShift/Swarm (Container) Native, autoscale to 0. PLONK - Grafana
4. Pricing due to Kubernetes cluster.
5. HTTP/TLS.

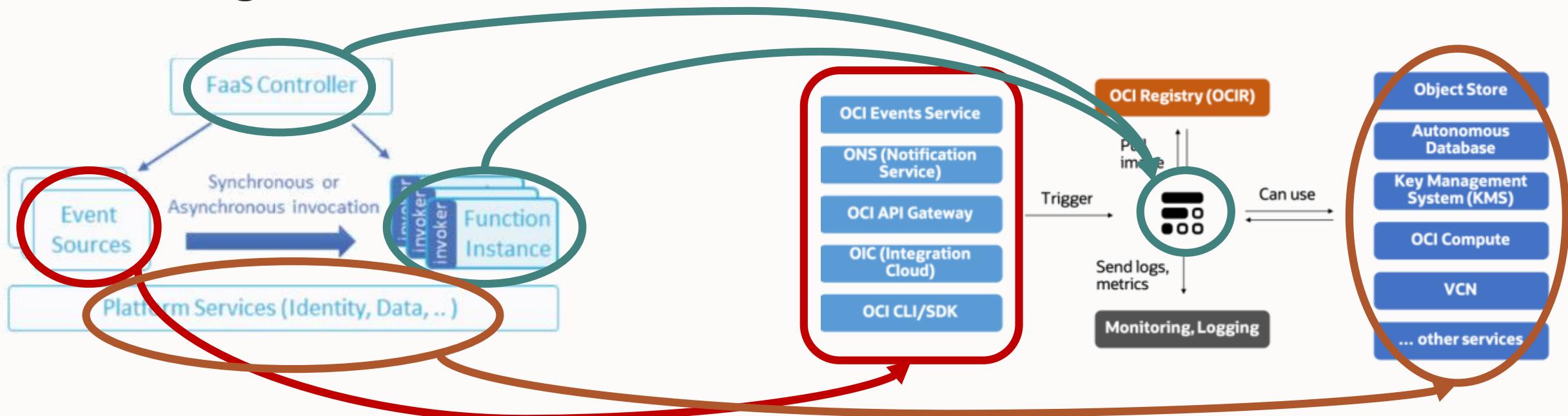


1. Open Source / Deprecated by VMWare (Dic/2021).
2. Multisite – Multicloud.
3. Kubernetes (Container) Native, autoscale to 0. Python, Node or Ruby.
4. Pricing due to Kubernetes cluster.
5. HTTP or Kafka Events.



1. Type of License
2. Scope
3. Description
4. Pricing
5. Input

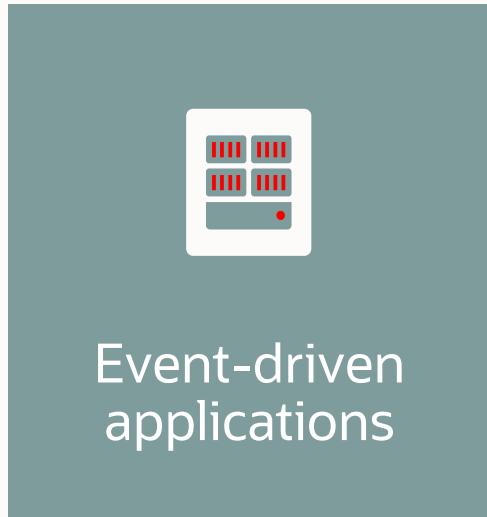
# Processing Model



**Events** are configured to make calls to Apps/Functions.

The **FaaS Controller** (Oracle Functions) is in charge of balancing and instantiating the Functions  
The integration is native from the Functions to **Cloud Services**.

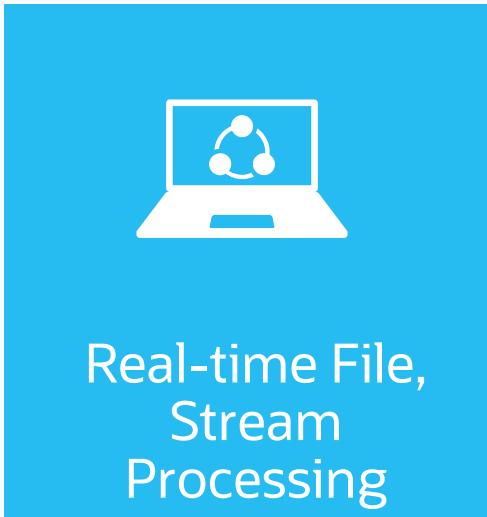
# Event-Driven Design Patterns: Common Use Cases



Event-driven  
applications



Web, Mobile, IoT  
Backends



Real-time File,  
Stream  
Processing



DevOps, Batch  
Processing



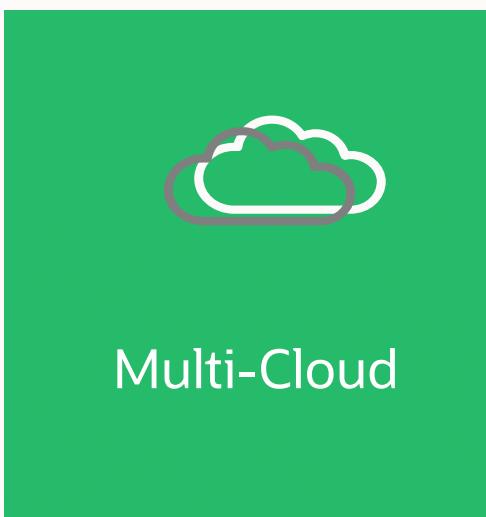
Glue Cloud  
Services, Event-  
driven



Security  
Operations

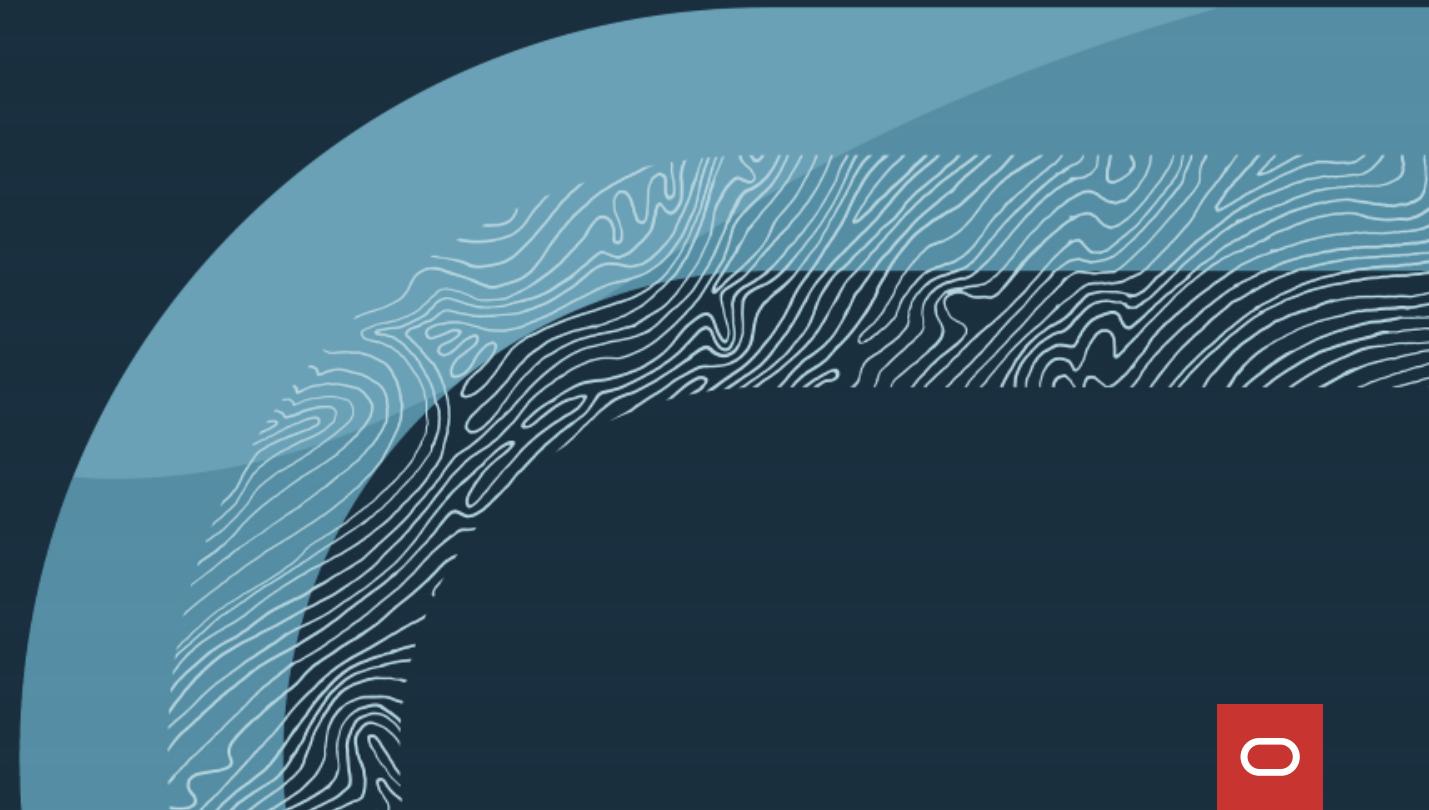


IT Operations



Multi-Cloud

# Demo DevOps with Functions



# Demo DevOps with Functions

<https://docs.oracle.com/en/solutions/build-cicd-pipelines-devops-function/>

Create Auth Token for the user

Launch Resource Manager (Terraform-as-a-Services)

Configure Options to the stack

In my case, the Auth Token contains:

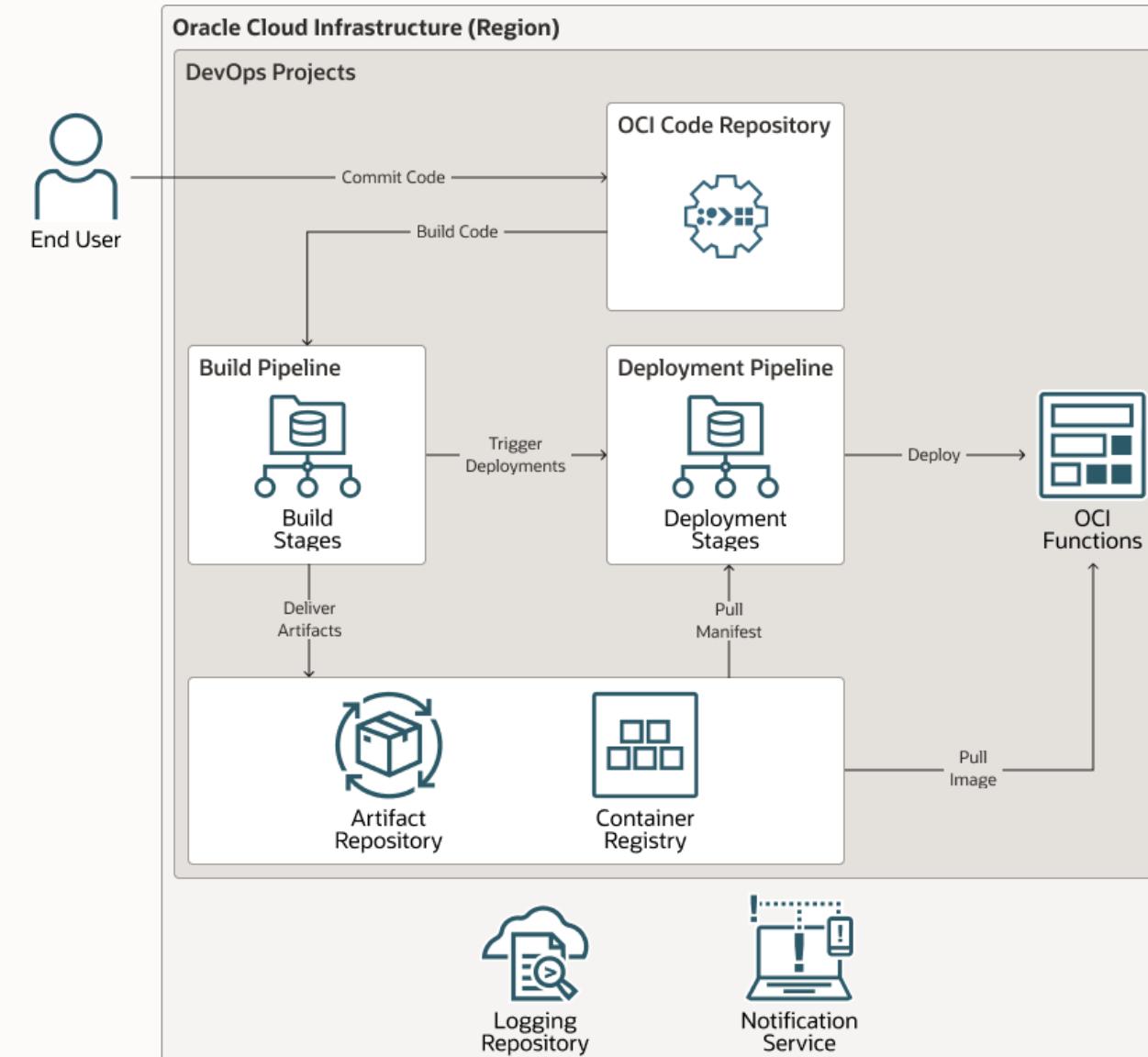
Usuario:

ladcsemrcn/oracleidentitycloudservice/francisco.m.moreno

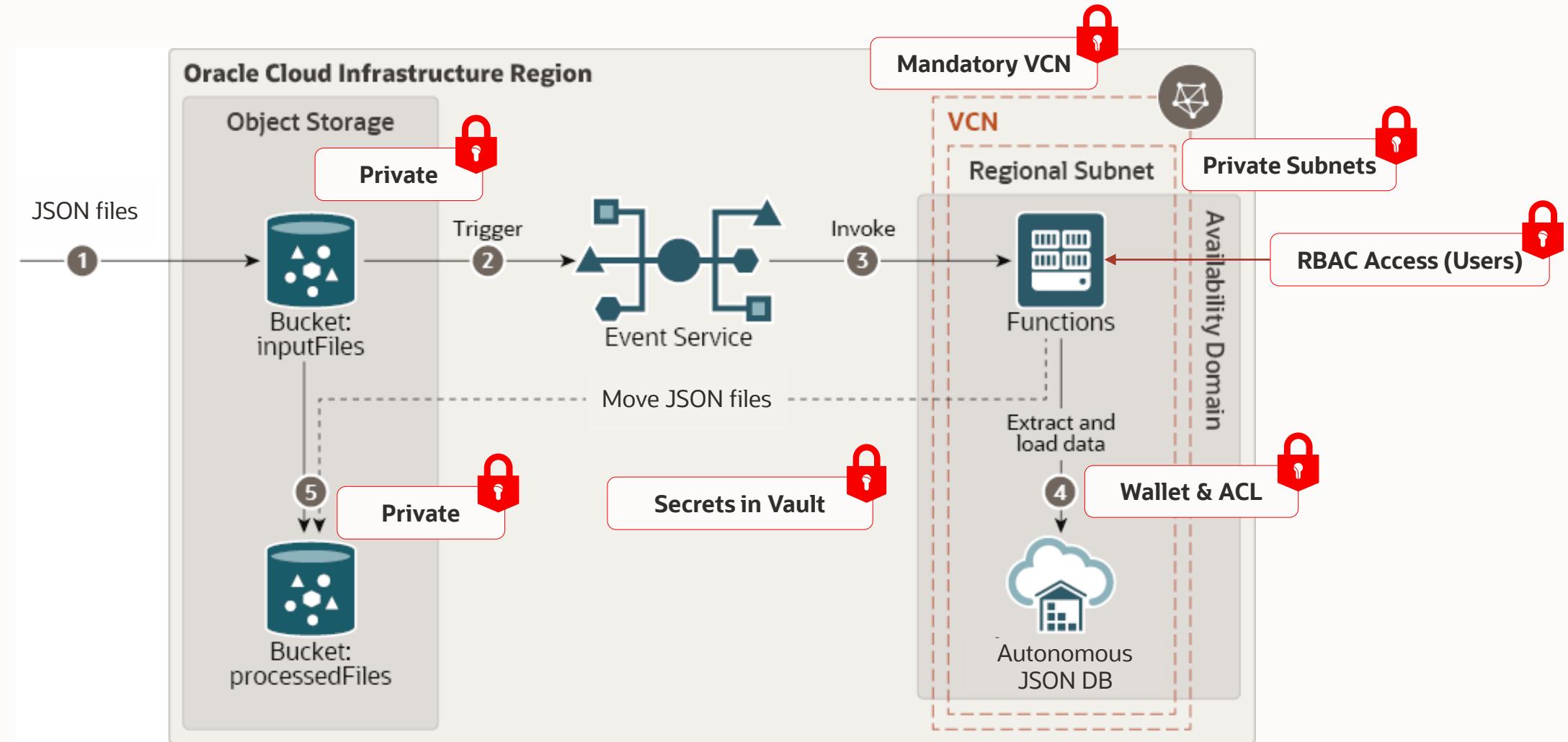
Password:

From OCI Cloud Console, select your username from Right Corner

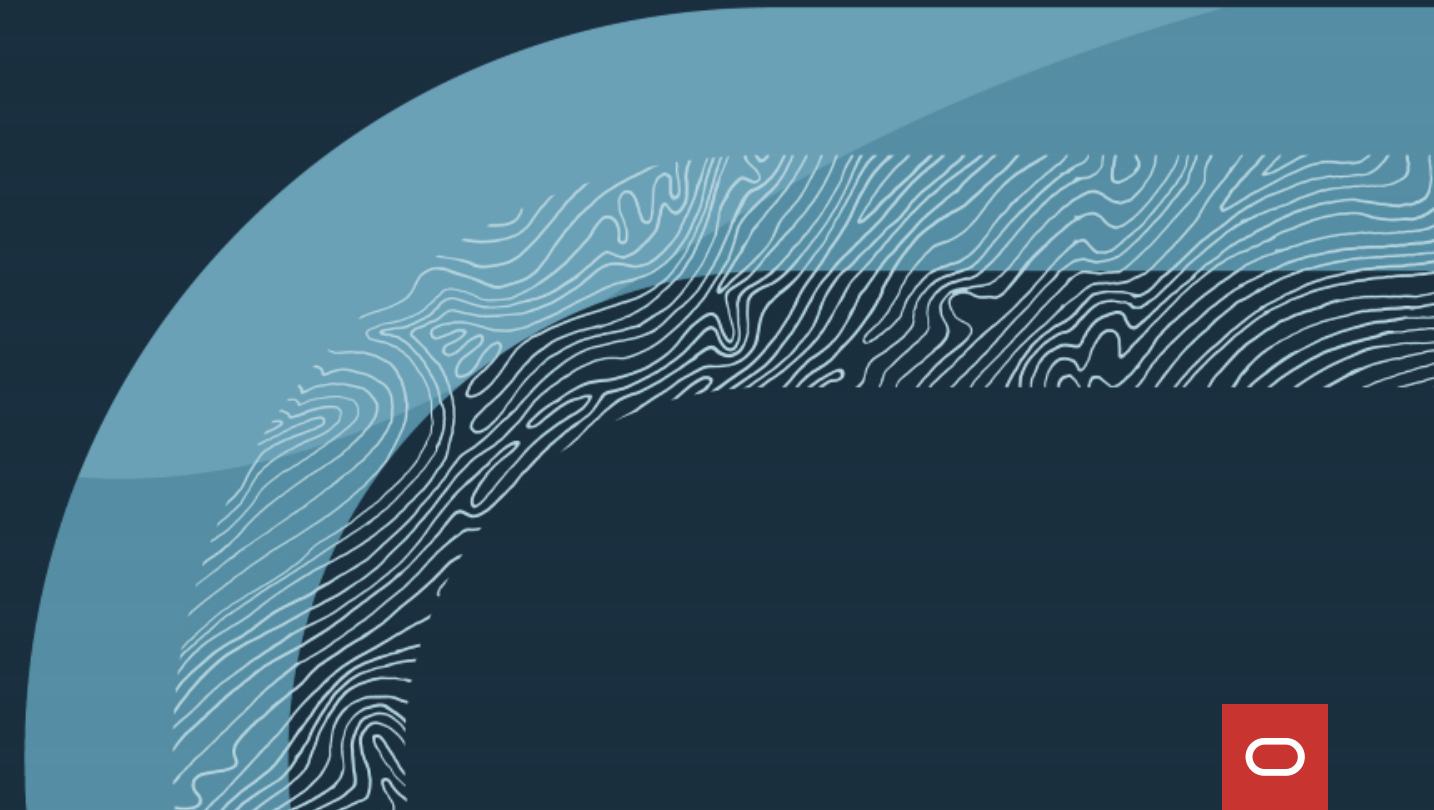
Then goes to *Auth Token*



# Securing the Books Catalog ETL Microservice



# References



# Tutorials Fn and Sample Code

<https://fnproject.io/tutorials/>

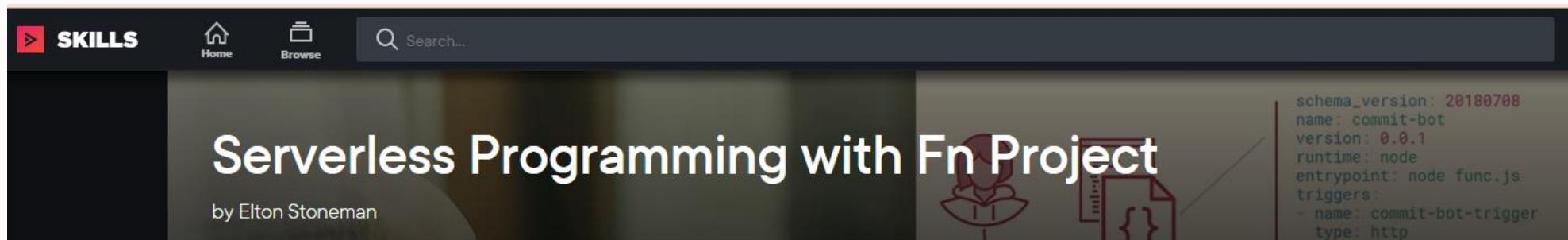
<https://app.pluralsight.com/library/courses/serverless-programming-fn-project>

## Explore Fn

- [Create Apps with Fn](#) - Learn how to group your functions into an application/API and deploy them together.
- [Create a Function with a Docker Container](#) - This tutorial provides a simple example of how to define an Fn function using a custom built Docker container.
- [Create a Function with a Linux Command and HotWrap](#) - This tutorial provides an example of how to define an Fn function using Linux commands, HotWrap, and a custom Docker container.
- [Create a Function from a Docker image that contains a Node.js app with Oracle DB Support](#)
- [Fn Development with Multiple Contexts](#) - Shows how to setup and use multiple Fn configuration contexts for development.
- [Using Fn RuntimeContext with a Function](#) - See how to set Fn Application and Function variables and access them in your function using the Runtime Context.

## Test and Monitor Functions

- [Monitor Fn metrics with Grafana and Prometheus](#) - Learn how to view Fn server metrics with Prometheus and Grafana.
- [Troubleshoot and Log functions](#) - Resolve issues at both development and deployment time.



# References



## Technical Blogs

SCHOLL, Boris. Getting Started with Microservices, Taken from Dic, 2021. 3 Articles.

<https://blogs.oracle.com/developers/post/getting-started-with-microservices-part-1-advantages-and-considerations>

RESELMAN, Bob. Design principles for microservices, Taken from Jan, 2021.

[https://developers.redhat.com/articles/2022/01/11/5-design-principles-microservices#five\\_design\\_principles\\_for\\_microservices](https://developers.redhat.com/articles/2022/01/11/5-design-principles-microservices#five_design_principles_for_microservices)

Cloud Native Computing Foundation

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

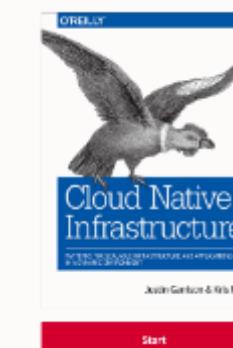


## Frameworks Ligeros

<https://helidon.io/>

<https://micronaut.io/>

<https://www.oracle.com/java/graalvm/>



# References



## Libro De Referencia

KHATRI, Anjali y otros. Mastering Service Mesh. Packt Publishing, May 2020.

<https://learning.oreilly.com/library/view/mastering-service-mesh/9781789615791/>

## Example DevOps Service

<https://docs.oracle.com/en/solutions/ci-cd-pipe-oci-devops/index.html>

## OCI Cloud Native

<https://www.oracle.com/cloud-native/>

## Ejemplo Cloud Native de los Demos

<https://github.com/oracle-quickstart/oci-cloudnative/>

## Oracle Kubernetes Engine

<https://www.oracle.com/cloud-native/container-engine-kubernetes/>

## Oracle Functions

<https://www.oracle.com/cloud-native/functions/>

## Fn Project

<https://fnproject.io/>

<https://app.pluralsight.com/library/courses/serverless-programming-fn-project>

<https://hackernoon.com/playing-with-the-fn-project-8c6939cf5cc>

<https://dev.to/kh40sz3r0/serverless-functions-with-fn-project-64e>

## Events Service (usando CNCF Events)

<https://www.oracle.com/cloud-native/events-service/>

## API Management

<https://www.oracle.com/cloud-native/api-management/>

## Verazzano

<https://www.oracle.com/java/verrazzano/>

<https://github.com/verrazzano/verrazzano>

## Mas Información

<https://oracle.com/application-development>



# References



## Service Mesh

<https://docs.oracle.com/en-us/iaas/Content/service-mesh/home.htm>  
<https://www.oracle.com/cloud/cloud-native/service-mesh/faq/>

## Nuevas VCN para Kubernetes

<https://blogs.oracle.com/cloud-infrastructure/post/announcing-vcn-native-pod-networking-for-kubernetes-in-oci> (15-Julio-22)

## Enable Service Mesh on OKE

<https://docs.oracle.com/en/solutions/oci-service-mesh-oke/index.html#GUID-12216A44-2E62-48D8-8C89-864335BE9EF6>

## Example DevOps Service

[https://learning.oreilly.com/videos/11-steps-to/9780135945346/9780135945346-esak\\_01\\_10\\_01\\_00/](https://learning.oreilly.com/videos/11-steps-to/9780135945346/9780135945346-esak_01_10_01_00/)

<https://www.bmc.com/blogs/serverless-faas/#:~:text=This%20offers%20users%20more%20flexibility,not%20limited%20to%20creating%20functions.>

<https://kubesphere.io/blogs/serverless-vs-faas/>

## Mas Información

<https://oracle.com/application-development>

## Verazzano

<https://www.oracle.com/java/verrazzano/>  
<https://github.com/verrazzano/verrazzano>

## Hands-on Labs

<https://go.oracle.com/hols>

<https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/home>

## Architecture Center - Reference Architecture

<https://docs.oracle.com/solutions/?lang=en>



# Por donde iniciar?

Contenedores y Functions	<b>Documentation &amp; tutorials</b> <ul style="list-style-type: none"><li><a href="#">Container Engine for Kubernetes</a></li><li><a href="#">Container Registry</a></li><li><a href="#">Functions</a></li></ul>	<b>Solutions Playbooks and Reference Architectures</b> <ul style="list-style-type: none"><li><a href="#">Set up a Kubernetes cluster for deploying containerized applications</a></li><li><a href="#">Deploy Cloud Native Apps with MySQL</a></li><li><a href="#">OCI for AWS Professionals</a></li><li><a href="#">OCI for Azure Professionals</a></li></ul>	<b>Labs</b> <ul style="list-style-type: none"><li><a href="#">Build Cloud Native Java Apps with Micronaut and GraalVM OCI Certification</a></li><li><a href="#">Automate the Deployment of Java Apps to OCI/DevOps Pipeline Developer Community</a></li><li>//</li></ul>
API y Streaming	<b>Training:</b> <ul style="list-style-type: none"><li><a href="#">Coursera Course: API Gateway</a></li></ul>	<ul style="list-style-type: none"><li><a href="#">API For Developers: Learn how to design, develop and manage APIs</a></li></ul>	
AI y Data Services	<b>AI Workshops</b> <ul style="list-style-type: none"><li><a href="#">Digital Assistant</a></li><li><a href="#">Language</a></li><li><a href="#">Speech</a></li></ul>	<b>Blogs</b> <ul style="list-style-type: none"><li><a href="#">Develop XR With Oracle Ep 3: Computer Vision AI, ML, and Metaverse</a></li><li><a href="#">ML and AI blog</a></li></ul>	
DevOps y Application Development	<ul style="list-style-type: none"><li><a href="#">DevOps service example workflows and terraform automation</a></li></ul>	<ul style="list-style-type: none"><li><a href="#">OCI DevOps Documentation</a></li></ul>	<ul style="list-style-type: none"><li><a href="#">DevOps CI/CD Reference Architecture</a></li><li><a href="#">Deploy a Jenkins CI/CD pipeline</a></li><li><a href="#">OCI DevOps Professional Certification</a></li><li><a href="#">DB Operations with DevOps</a></li></ul>



# Functions: Additional Features

- [Micronaut Functions and API example](#). Has Java and Java Native support. Has Gradle support.
- [Managing Functions Lifecycle with Visual Builder Studio \(CI-CD\)](#)
- [LiveLab: Develop/Deploy a Modern Application with Node.js Functions & MySQL Database Service](#)
- Functions with Oracle NoSQL DB
- [Blog: Node/ADB functions](#) using a custom Dockerfile e.g., install Oracle Client for Node
- Service Connector Hub-triggered functions e.g., [Send OCI Logs to Datadog](#)
- Function chaining with Integration Cloud

