

Oracle Cloud Native Application Development and DevOps (CanDo)

George Moykin

Cloud Domain Sales Consultant

Oracle

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, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Oracle Cloud Platform Overview
- Cloud Native Application Development
- DevOps
- CanDo Workshop Hands-on

Oracle Cloud



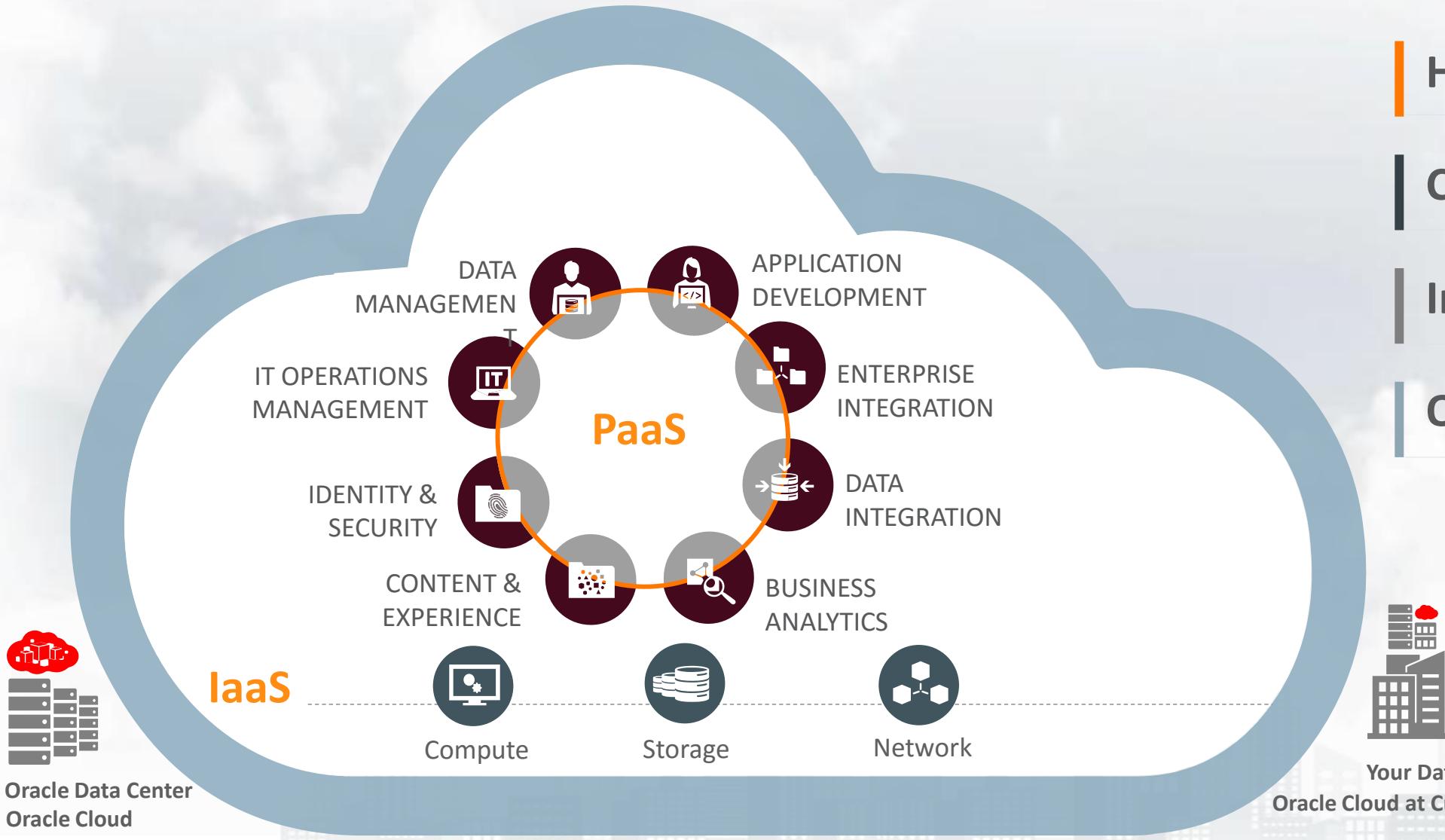
Data-as-a-Service

Software-as-a-Service

Platform-as-a-Service

Infrastructure-as-a-Service

Oracle Cloud Platform



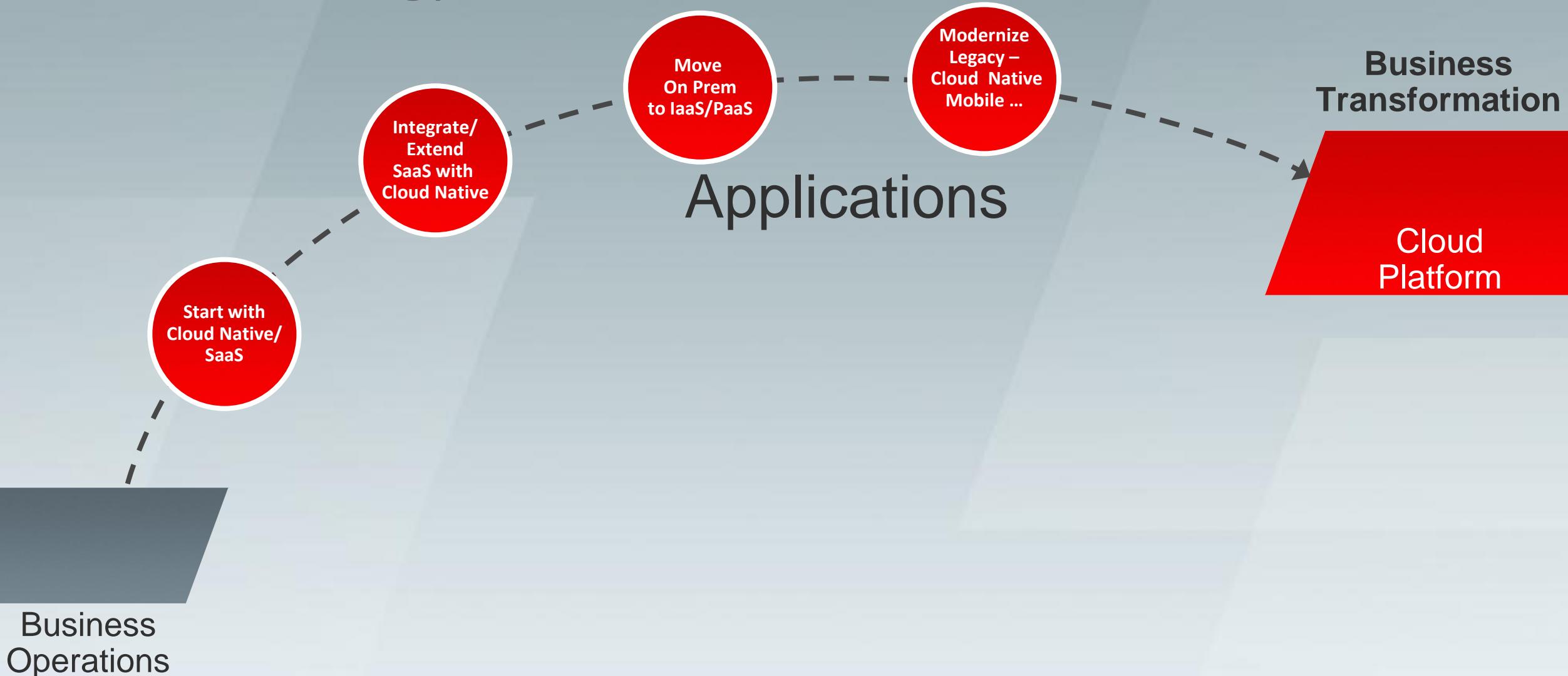
Hybrid Cloud

Comprehensive

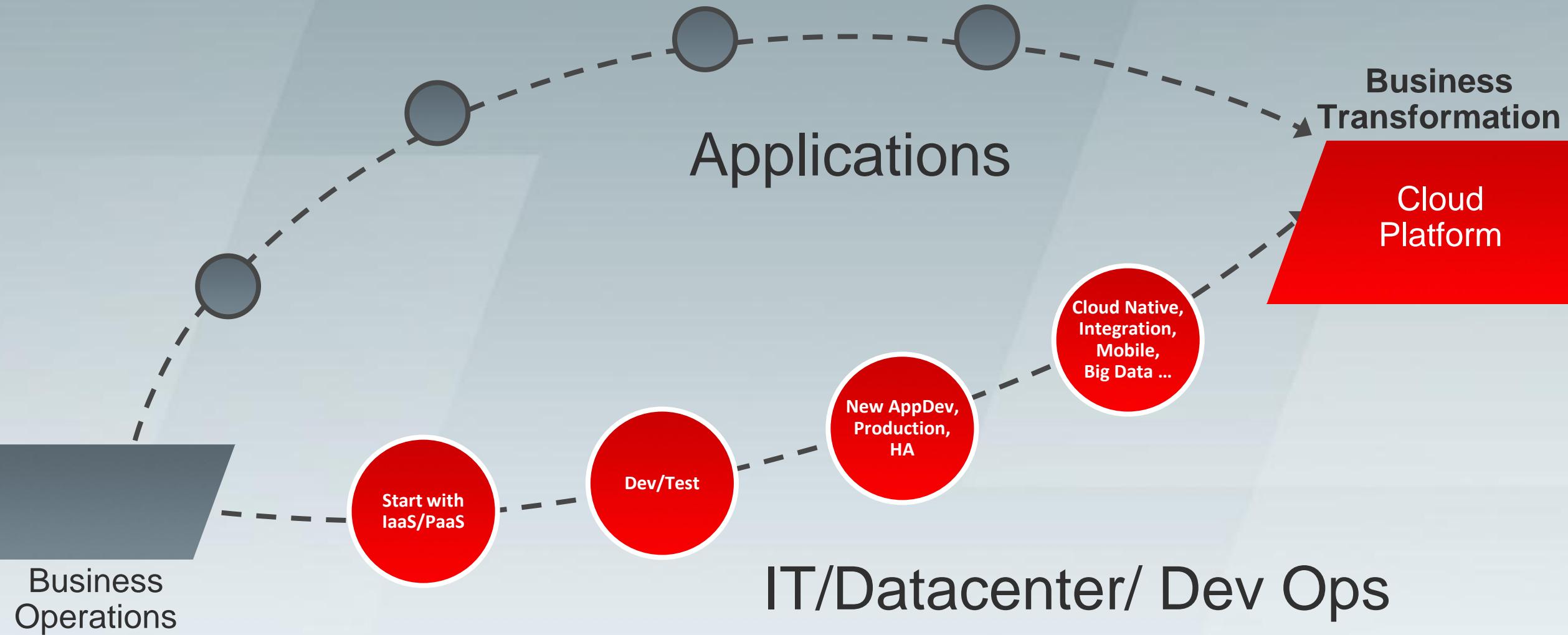
Integrated

Open

A Cloud Strategy Point of View



A Cloud Strategy Point of View



AppDev is About our Customer's Entire Portfolio

Innovation Software - Find the Next Business (PaaS/Cloud)

Differentiation Software - Run Current Business (Java/DB → PaaS)

Core Software - Keep the Lights On
(ERP → IaaS/SaaS)

Aged Software,
Limited DevOps

Mature, Stable Software,
First Generation DevOps

Modern,
Emerging Software,
Modern DevOps

Cloud Native Application Development



Oracle Cloud AppDev Strategy – Modernization First

Migrate/Extend to Cloud, Modernize to PaaS, Build New Cloud Native



On Premise

- Oracle
- Non Oracle Workloads



Rehost to Cloud

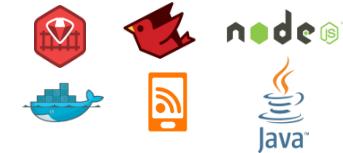
- Core (ERP/CRM) workloads
- Legacy workloads
- Non Oracle workloads



Java Cloud, SOA Cloud,
DB Cloud, Dev Cloud

Modernize to PaaS

- Differentiation workloads
- Java, SOA, and Database



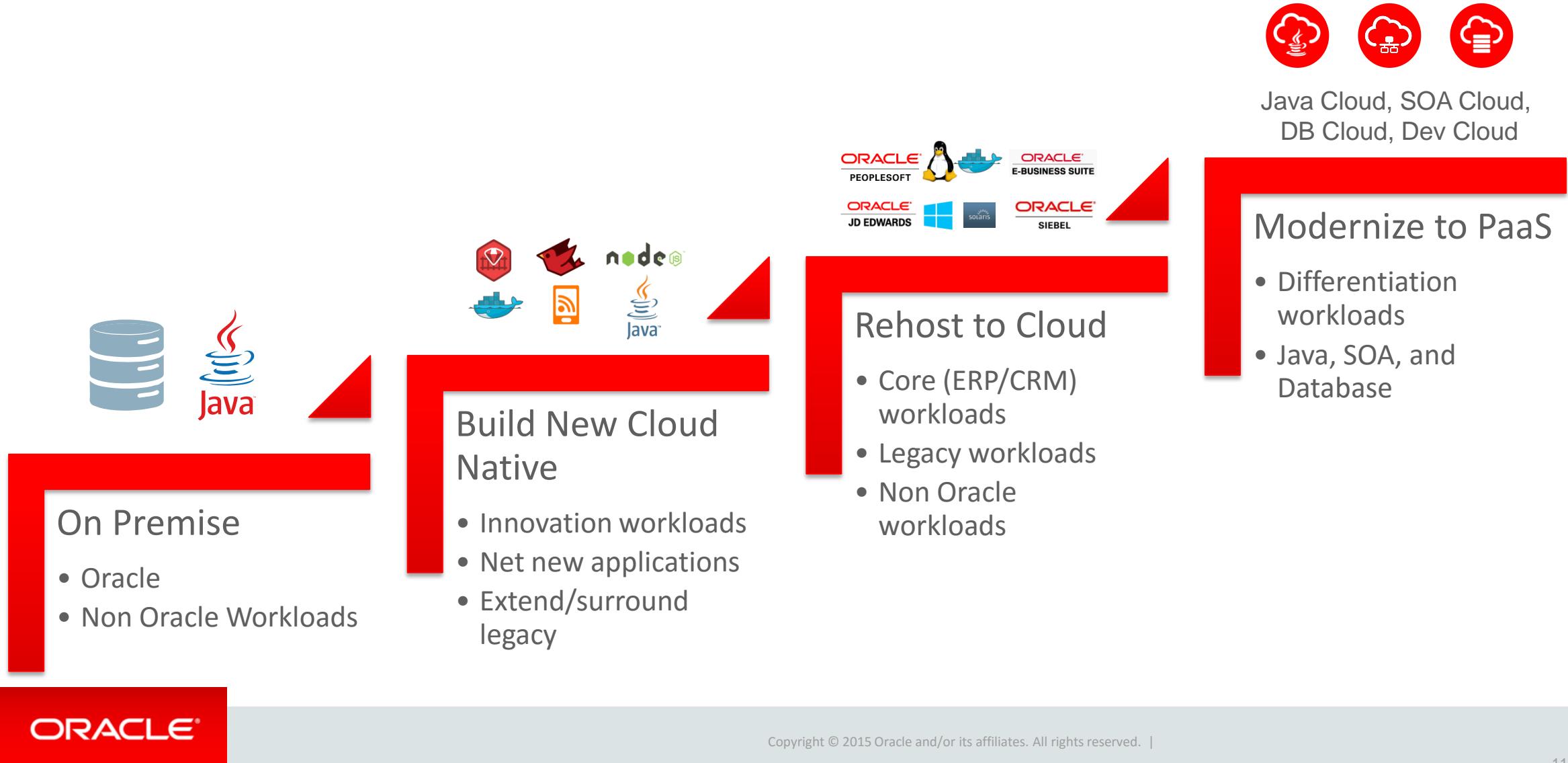
App Container, Dev Cloud, Mobile, API,
APM, App Builder, Integration

Build New Cloud Native

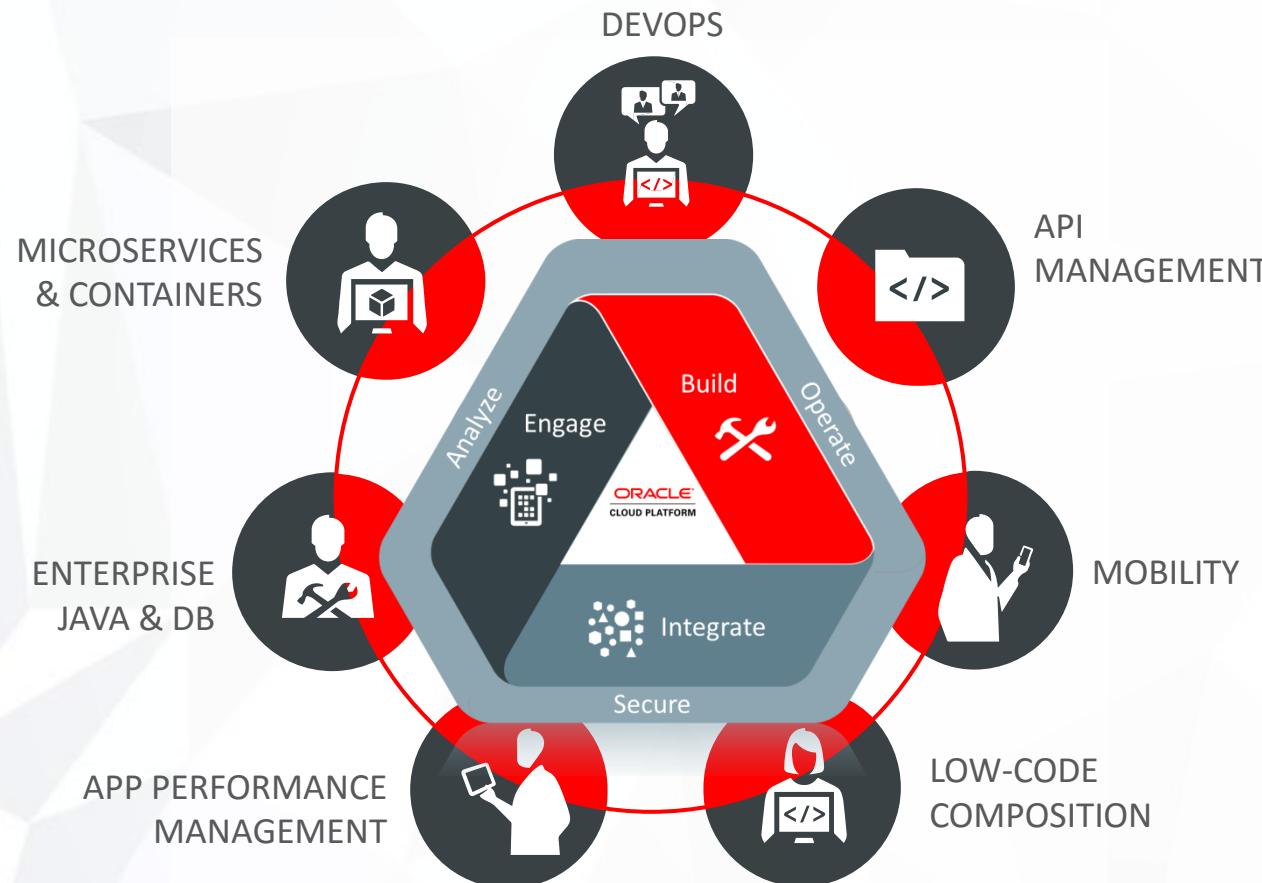
- Innovation workloads
- Net new applications
- Extend/surround legacy

Oracle Cloud AppDev Strategy – Cloud Native First

Migrate/Extend to Cloud, Modernize to PaaS, Build New Cloud Native

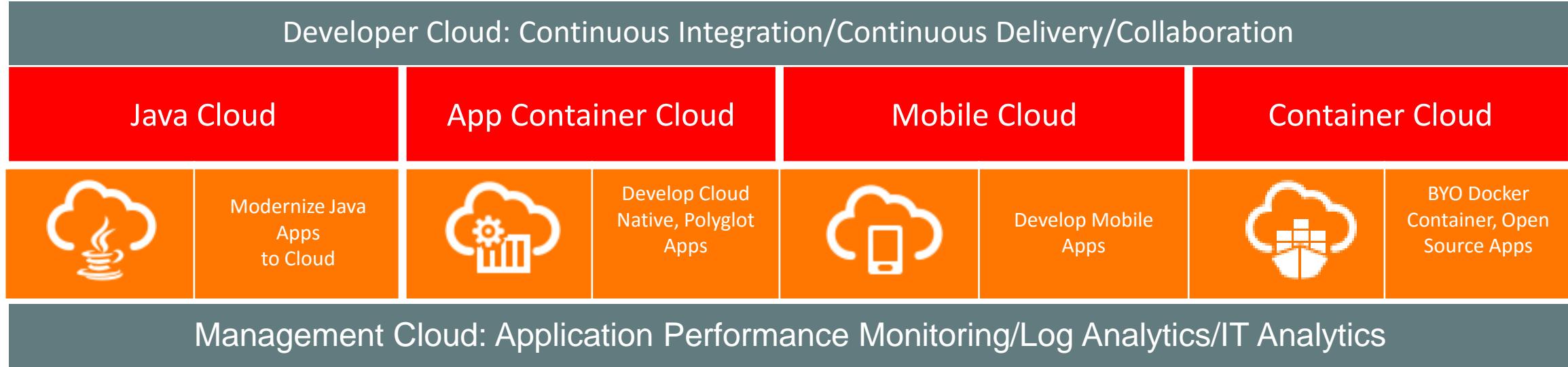


Oracle Cloud Platform: For Application Development



- **Comprehensive AppDev**
 - Cloud native, migrate, low code
- **Automated DevOps**
 - For continuous integration & delivery
- **API First**
 - Mobility & multi-channel delivery
- **Single Pane of Glass**
 - For monitoring & management

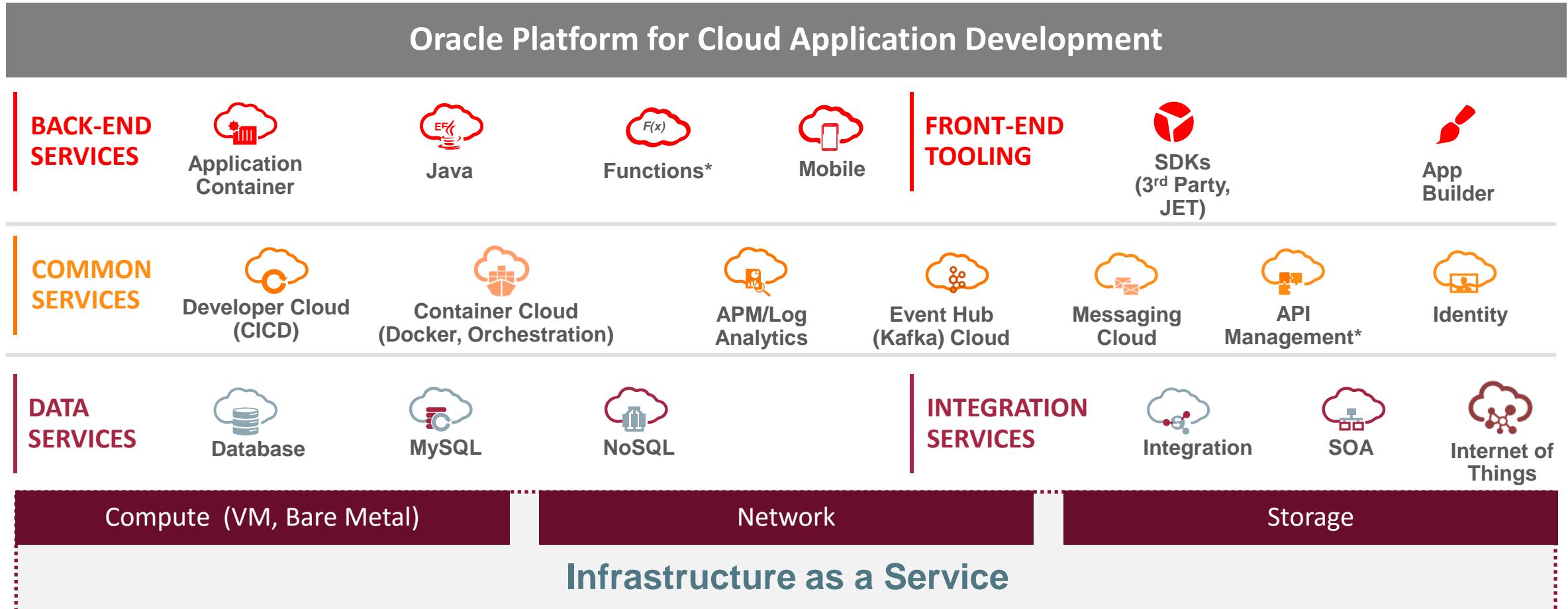
Oracle's Platform for Cloud Native Development



- For all application types
- Cloud native developer infrastructure, DevOps built-in
- Managed platform (patching, scaling, backups, ...)
 - Cloud management tools built-in
 - Cloud and on-Premise app management

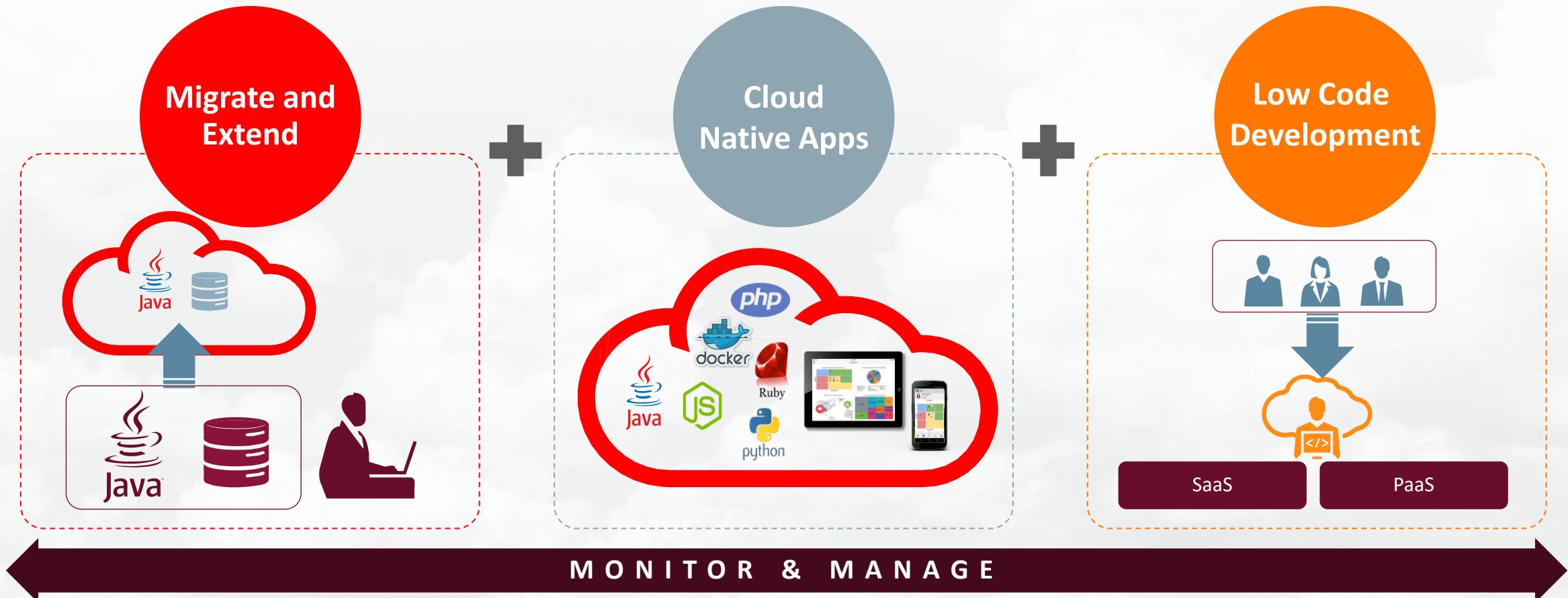
Oracle Cloud Platform: Application Development Services

Unique in Blending Traditional, Cloud Native and Low Code with End to End PaaS



Why is Oracle Different and Better at AppDev?

One Solution that Solves Migrate & Extend, Cloud Native and Low Code



Two Key Development Approaches

Build Modern, Cloud-Native Apps



App Container, App Builder, Mobile

More capabilities, faster
Speed to market

Increased rate of innovation
Easier experimentation

Modernize Existing Workloads



Java, Database, IaaS

Offloading operations/IT

Faster dev/test

Access to PaaS services

Pre-integration

Requirements for Cloud Native Development

**Scalable Elastic
Polyglot
Microservices**



**Automated
DevOps for Agility**



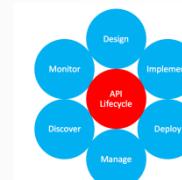
**Instrument for
APM and Diagnostics**



**Containers
For Simplicity &
Extensibility**



**API First
for Service
Consumption**



**Mobile First and
Modern Web UX**



What is Oracle's Cloud Native AppDev Solution?



Developer Cloud

Continuous
Integration/Continuous
Delivery



Application
Container Cloud

Polyglot Lightweight
Applications



API Platform Cloud

Publish/Manage Service
APIs



Container Cloud

Docker Container
Management &
Orchestration



Mobile Cloud

Mobile First, API,
Analytics, BOTS



Management
Cloud

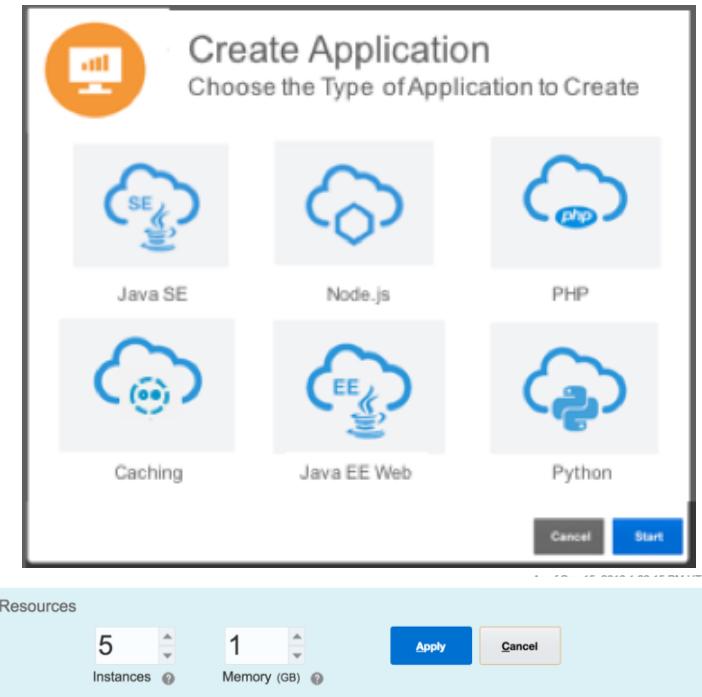
APM, Security and Log
Analytics

Application Container Cloud – Polyglot Cloud Native Apps

Java SE, Java EE, Node, PHP, Caching ...

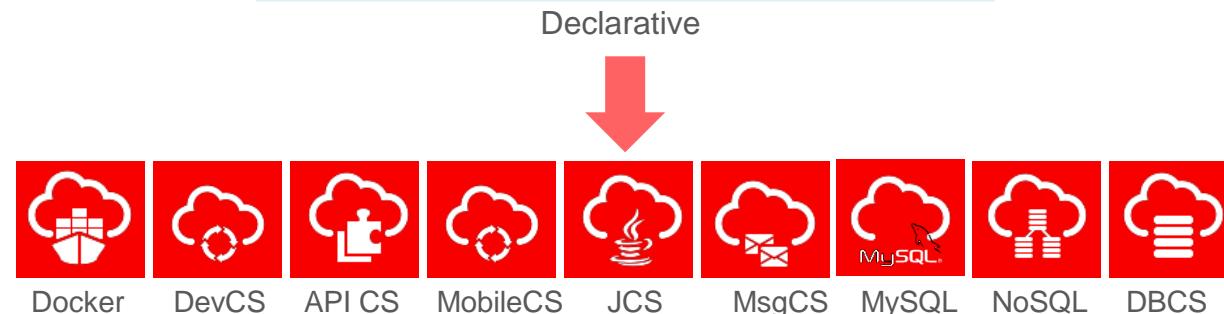
Cloud native, polyglot Applications

Integrated Cloud Native PaaS Services



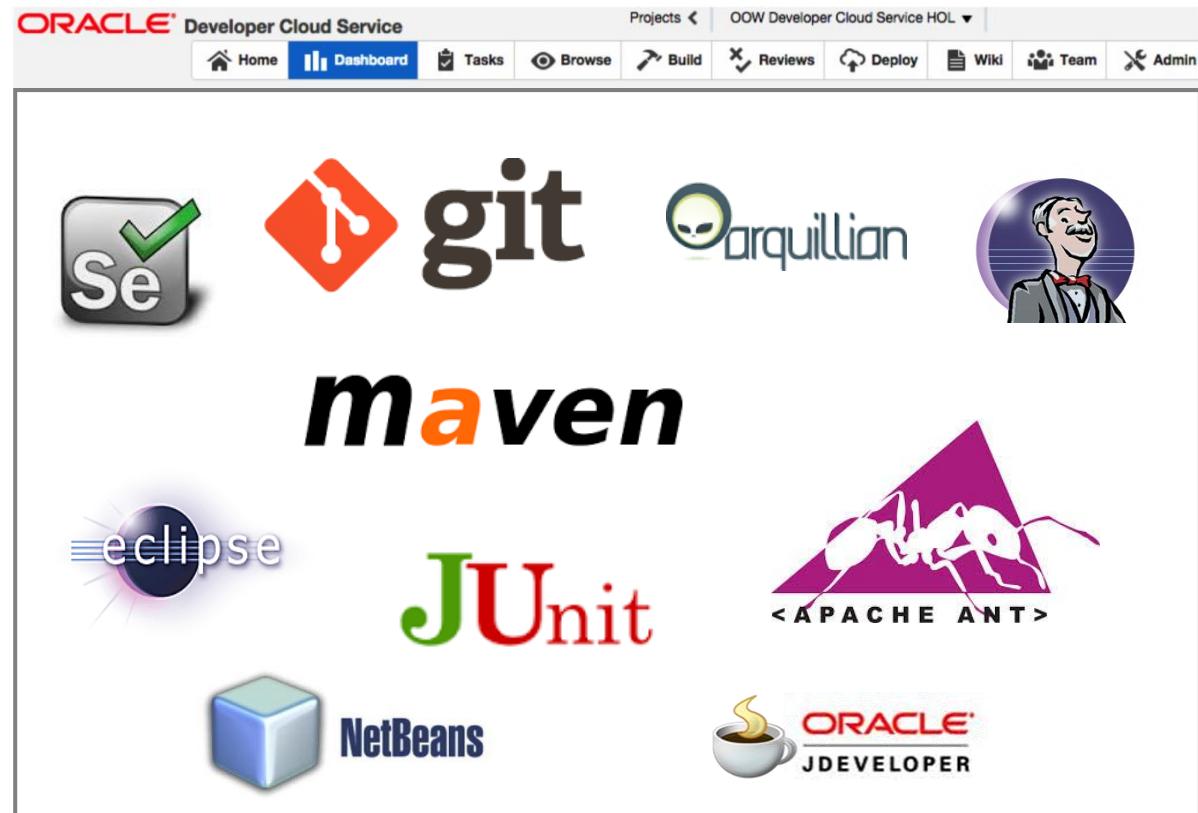
Built on Docker with Autoscaling/Elasticity

Integrates to Existing Oracle Workloads

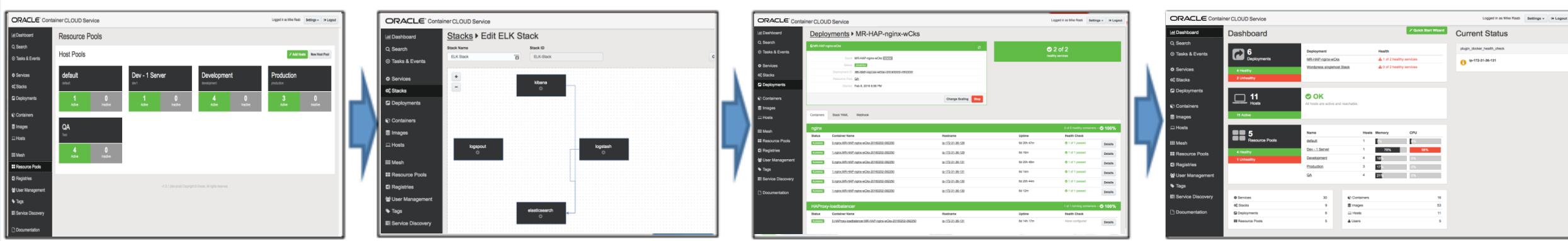


Developer Cloud Service – Continuous Integration/Delivery and Agile Development for Cloud Native Apps

- Standards Based
 - Hosted Git, GitHub, Maven, Hudson
- Built-in IDE Integration
 - Eclipse, NetBeans, JDeveloper
- Testing Designed in
 - Selenium, Arquillian
- Built in Collaboration
 - Scrum/Agile, Wikis, Issue Tracking
- Choice of Deployment Target
 - Java Cloud, Java SE or Node Cloud, SOA Cloud, Mobile Cloud



Container Cloud Service for Full Control Docker



Configuration Management

- Create Instances as Needed
- Define Resource Pools
- Add Private Registries

Application Deployment

- Edit Create New Services
- Compose Application Stacks
- Deploy Stacks with 1 Click

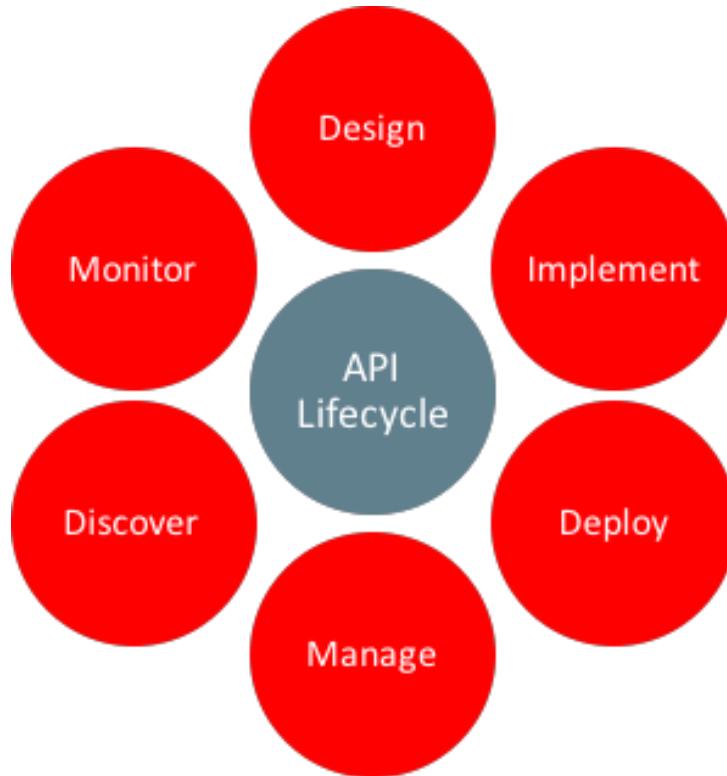
Container Orchestration

- Automated Deployment
- Multi-Host, Easy Scale Out
- Built in Service Discovery

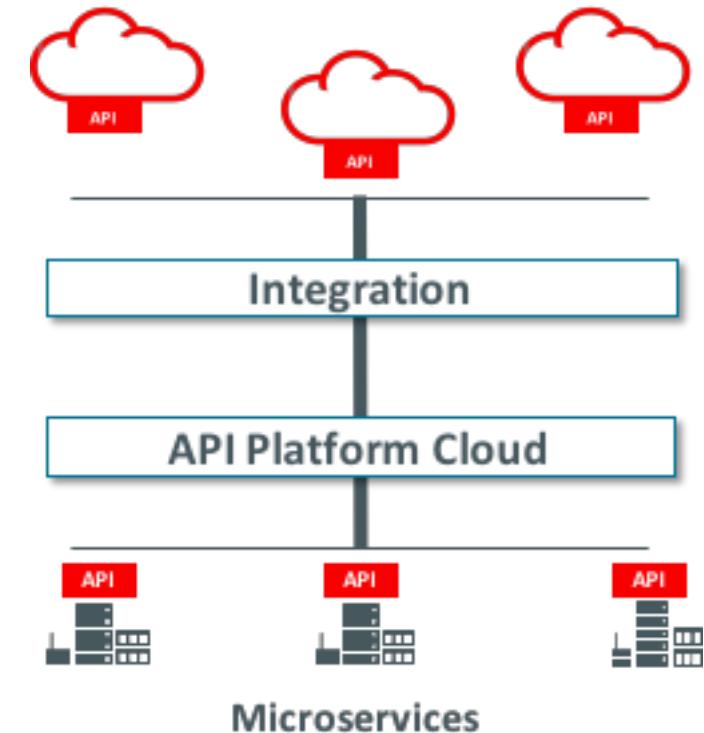
Operations Management

- Integrated Health Checks
- Unified Dashboard
- Monitoring and Auditing

API Platform Cloud – Design, Implement, Deploy, Secure and Manage Cloud Native Application APIs

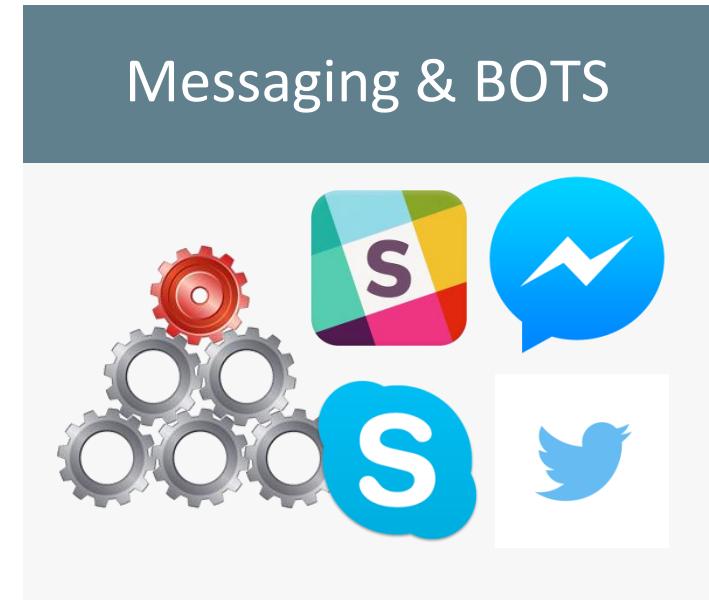
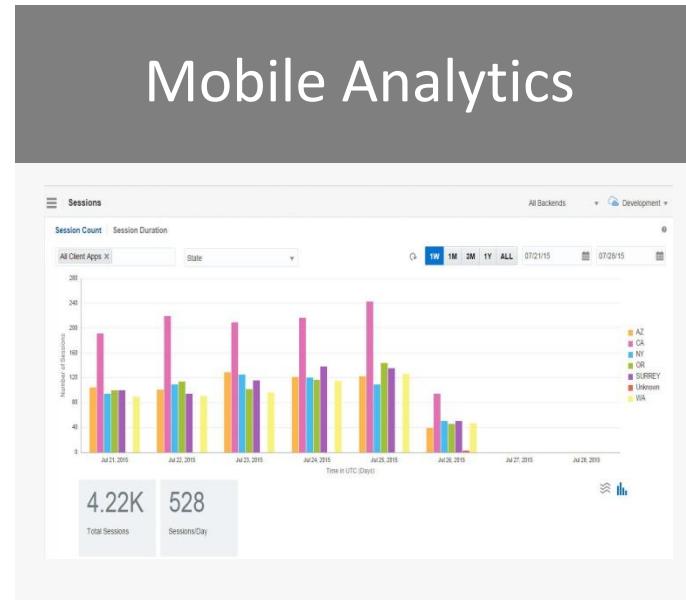


End to End Platform for Managing APIs



Bringing Unified Model Connecting Microservices to PaaS & SaaS Services

Mobile Cloud Service – Deliver Cloud Native Applications through Multiple Channels



Oracle Management Cloud Services: Monitor, Analyze and Diagnose Cloud Native Applications



Application Performance Monitoring

Improve End-User Experience and System Performance; Diagnose Performance Issues Faster



Log Analytics

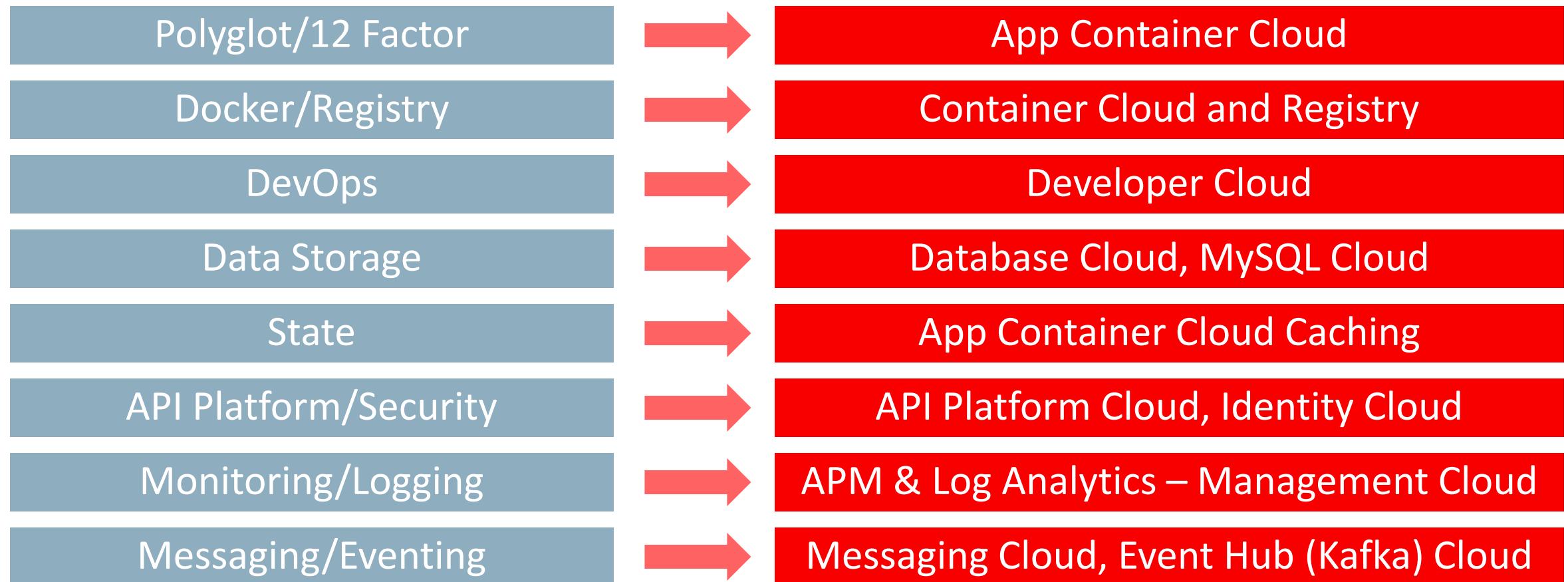
Extract Value from Logs by Collecting, Correlating, and Searching Any Kind of Log Data; Quickly Discover Anomalies



IT Analytics

Make Critical Decisions About Your IT Estate; Plan For Growth, Run What-If Analyses, Compare Resource Usage

End to End Cloud Native Capability Mapping



Summary: Why is Oracle Better at Cloud Native?

First Class Cloud Native Platform

- Polyglot microservices on Application Container Cloud built on Docker foundation – Full Lifecycle Support

Integrated DevOps Foundation

- Continuous integration and delivery pipeline with Developer Cloud Service

API Management for Application APIs

- API Platform Cloud Service – Secure, Version, Analyze APIs

Complete Visibility and Diagnostics

- Application Performance Monitoring and Log Analytics with Oracle Management Cloud

PaaS Ecosystem to Integrate and Extend Apps

- Integration Cloud Service, Java Cloud Service, Database Cloud Service, Mobile Cloud Service ...

Two Key Development Approaches

Build Modern, Cloud-Native Apps



App Container, App Builder, Mobile

More capabilities, faster
Speed to market
Increased rate of innovation
Easier experimentation

Modernize Existing Workloads



Java, Database, IaaS
Offloading operations/IT
Faster dev/test
Access to PaaS services
Pre-integration

Requirements for Modernizing Existing Workloads

Migrate Existing Workloads without Change



Automate Development with DevOps



Instrument for APM and Diagnostics



Automate Ongoing Maintenance/ Lifecycle



Extend with Microservices



Modernize with Mobile and Modern Web UX



What is Oracle's Java and Database Modernization Solution?

Onboarding Workloads to Automated DevOps, Maintenance, Elasticity, DR and More



AppToCloud
Migration

DB Backup/Restore
DB Clone
DB Multitenant
Data Pump



DevOps Enabled

Elasticity and
Autoscaling

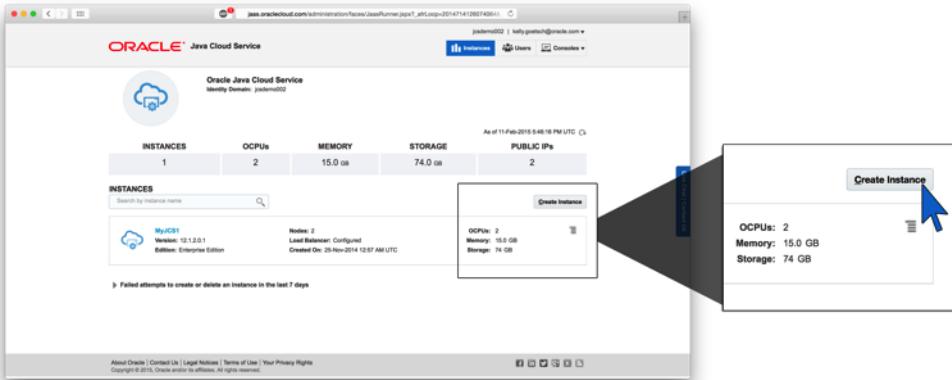
Automated Patching

Automated
Backup/Restore

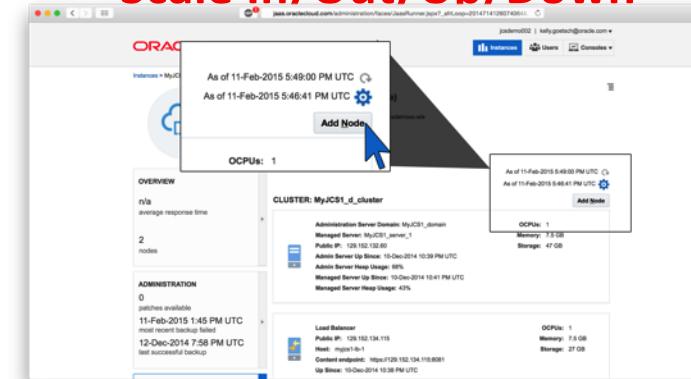
Automated Disaster
Recovery

Lifecycle Automation with Java Cloud and Database Cloud

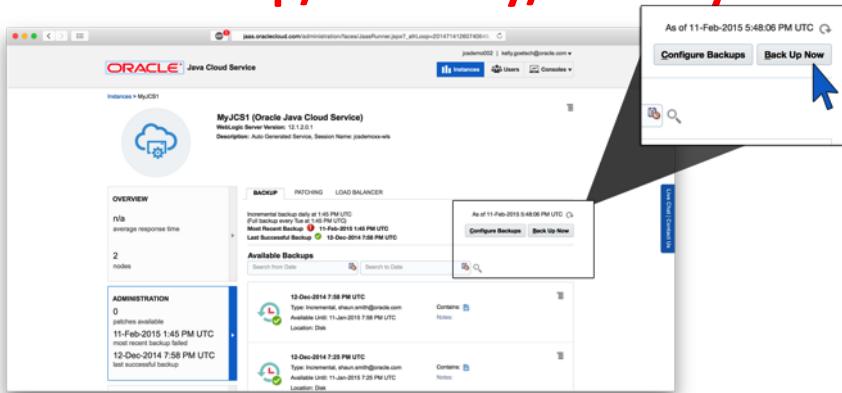
Instant Provisioning



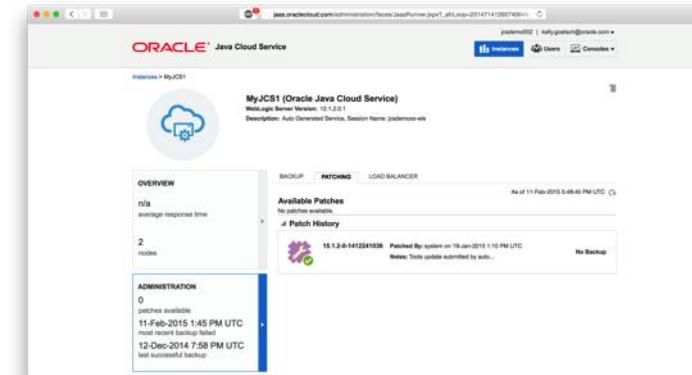
Automated Elasticity/Autoscaling Scale In/Out/Up/Down



Automated Backup/Recovery/Standby



Integrated, Automated Patching



Driven via API, CLI and Web UI

Seamlessly Integrated into Existing DevOps Pipelines

Automation Tools



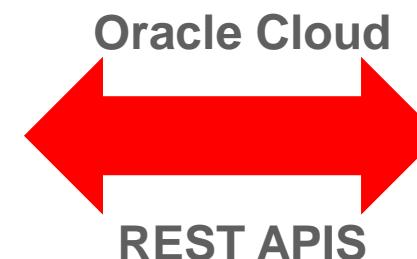
DevOps Pipelines



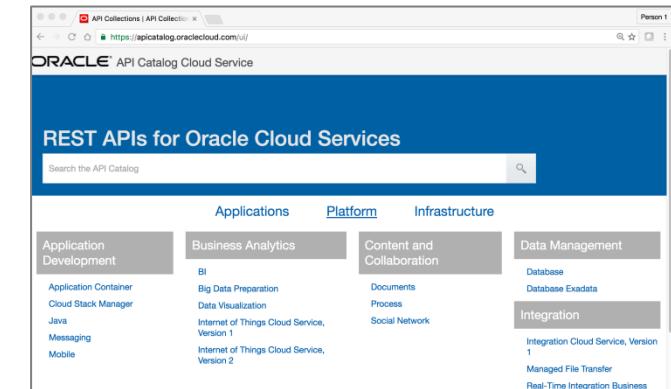
Command Line tools or curl

```
>psm setup  
>psm list services  
>psm push app ...
```

```
>curl -i -X GET -H  
"Authorization:joe  
@example.com:joePa  
ssword" \ -H ...
```



Service API Catalog



Service APIs

REST APIs for IaaS

- Docker, VMs, Storage, Network

REST API for PaaS

- AppDev, Integration, Mobile, BI, DB...

REST APIs for SaaS

- ERP, Service, Sales, Support ...

Enabling Modernization and Cloud Native Extensions

Strangle/Extend the Monolith by Taking Advantage of Cloud Native Services

Enterprise Java, FMW
Database Workloads on Cloud



Ecosystem to
Extend and
Modernize

Integration
(Integration Cloud)

Cloud Native
(Polyglot Microservices)

Mobile and IoT
(Mobile and IoT Cloud Service)

Low Code UX
(AppBuilder Cloud)

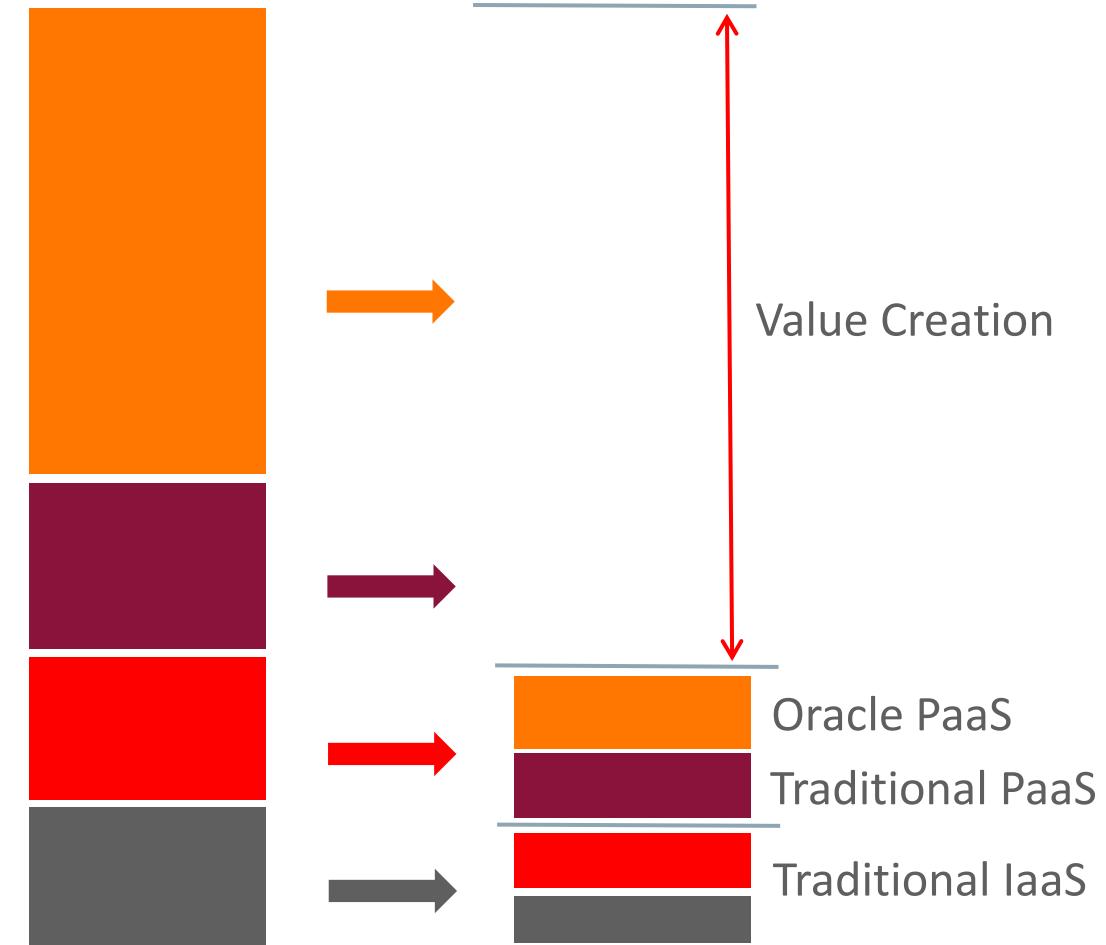
Value Creation with Cloud Automation for Enterprise Workloads

Ongoing Maintenance Cost: Backup, Patching, Hardware Upgrade, OS Upgrade, Firmware Upgrade, Software Upgrade, Test-Dev Synchronization, Cloning, Data Masking, Security Configuration Checks, Security Auditing, ...

Software Cost: License, Installation, Configuration, Security Setup, DR Setup, ...

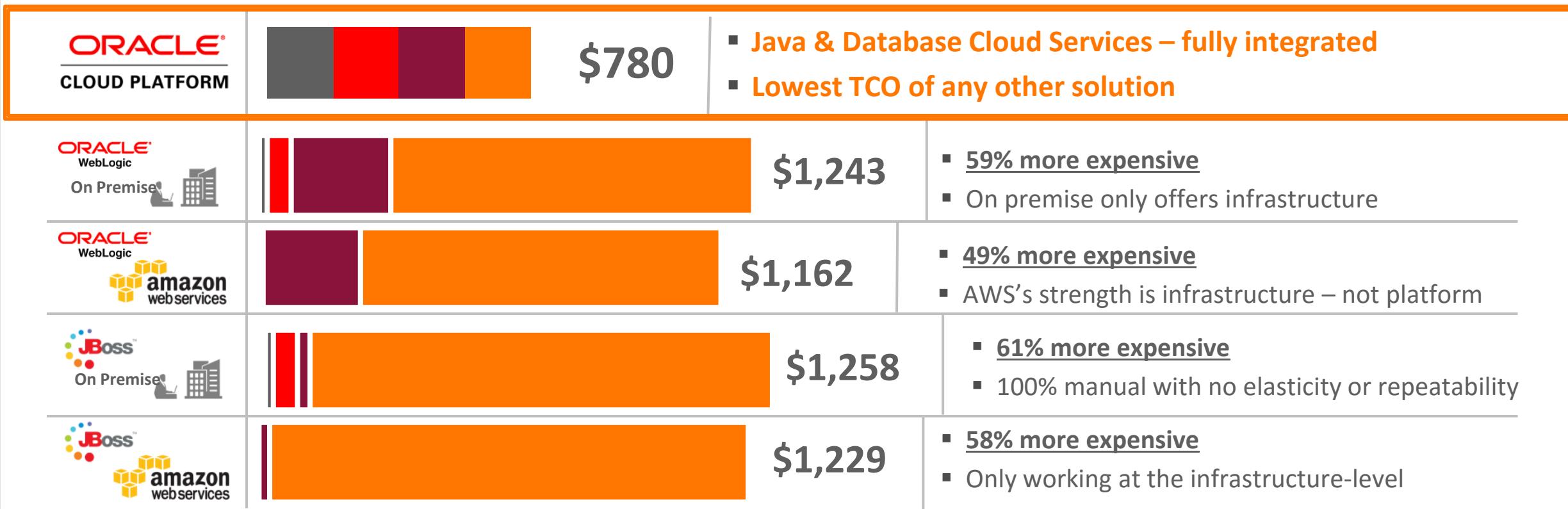
Hardware Cost: Servers, Storage, Network, ...

Facilities Cost: Data Center, ISP, CDN, DNS, ...



Java and Database Workload TCO Results with Oracle Cloud

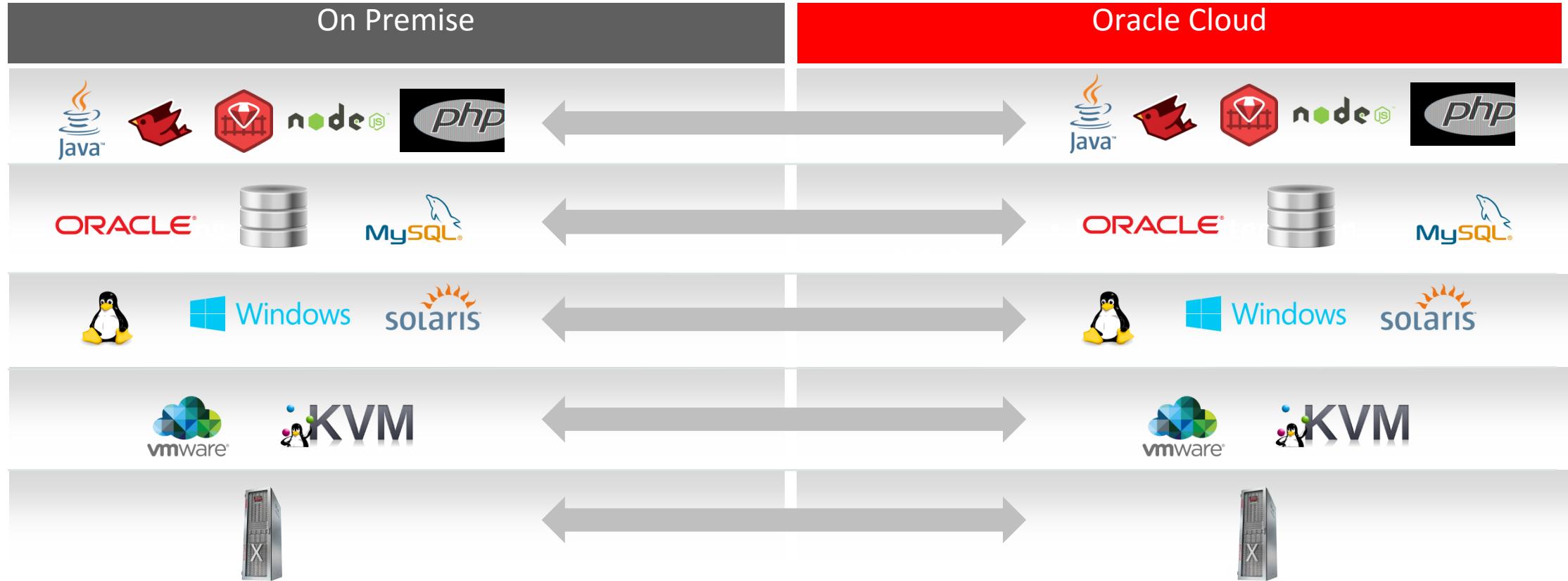
Cost/core/month over 4 years for 1,000 cores



■ Facilities ■ Hardware ■ Software ■ Maintenance

Part of a Full Solution for Modernizing Existing Workloads

Transparently Move **All** Application and Database Workloads to Cloud



Summary: Why is Oracle Better at Java and DB Modernization

Automated Tools to Migrate Workloads

- AppToCloud, Multitenant DB and WLS, Full support for OS/Virtualization - Windows, Solaris, Linux, VMWare, KVM ...

Complete Support for Oracle Workloads:
Java, Fusion Middleware and Database

- First class services - Java Cloud, SOA Cloud, Database Cloud with Full Enterprise Capabilities

Not Just Oracle Workloads – Open
Source, 3rd Party, Custom

- Over 120 Prepackaged OSS Images; 100's of Apps on Cloud Marketplace

Lowest Cost through Ongoing
Maintenance Automation

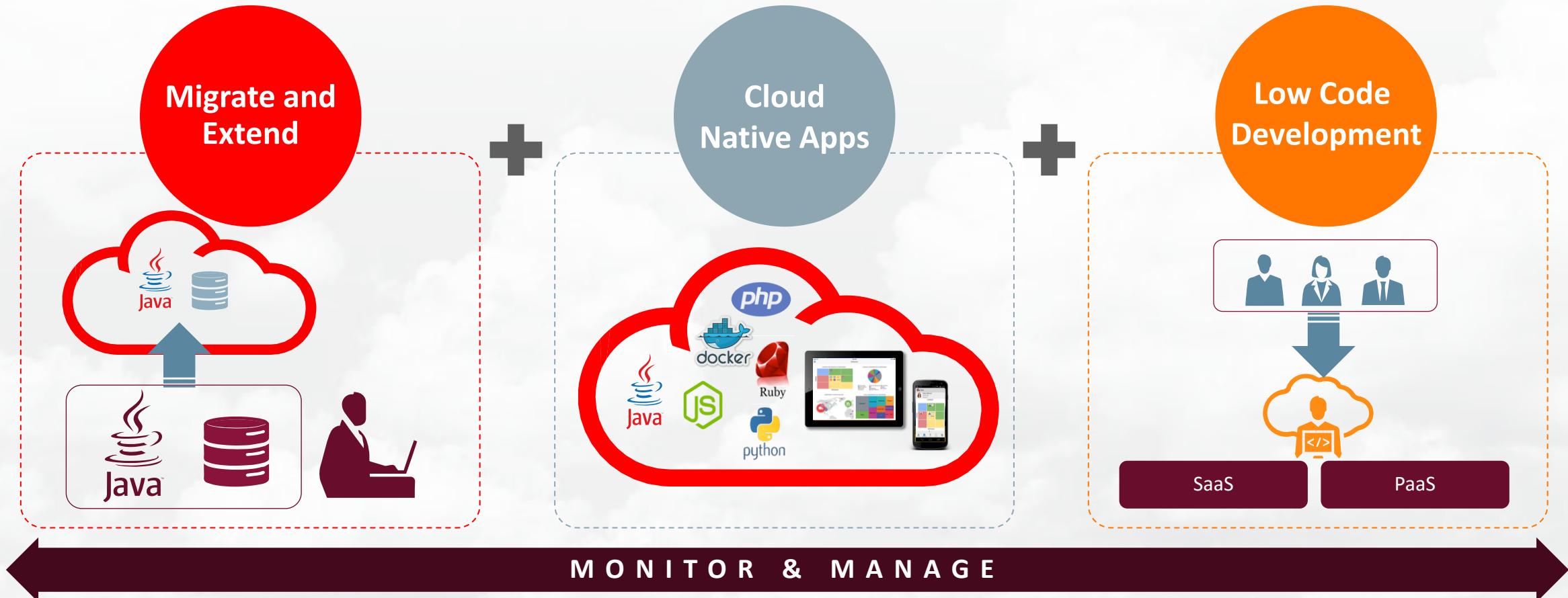
- Backup, Restore, Elasticity, DR, Patching ...

Extensive Ecosystem to Modernize

- Integration, Mobility, Internet of Things, Line of Business, Cloud Native, Big Data

Oracle Cloud Application Development

One Solution that Solves Migrate & Extend, Cloud Native and Low Code

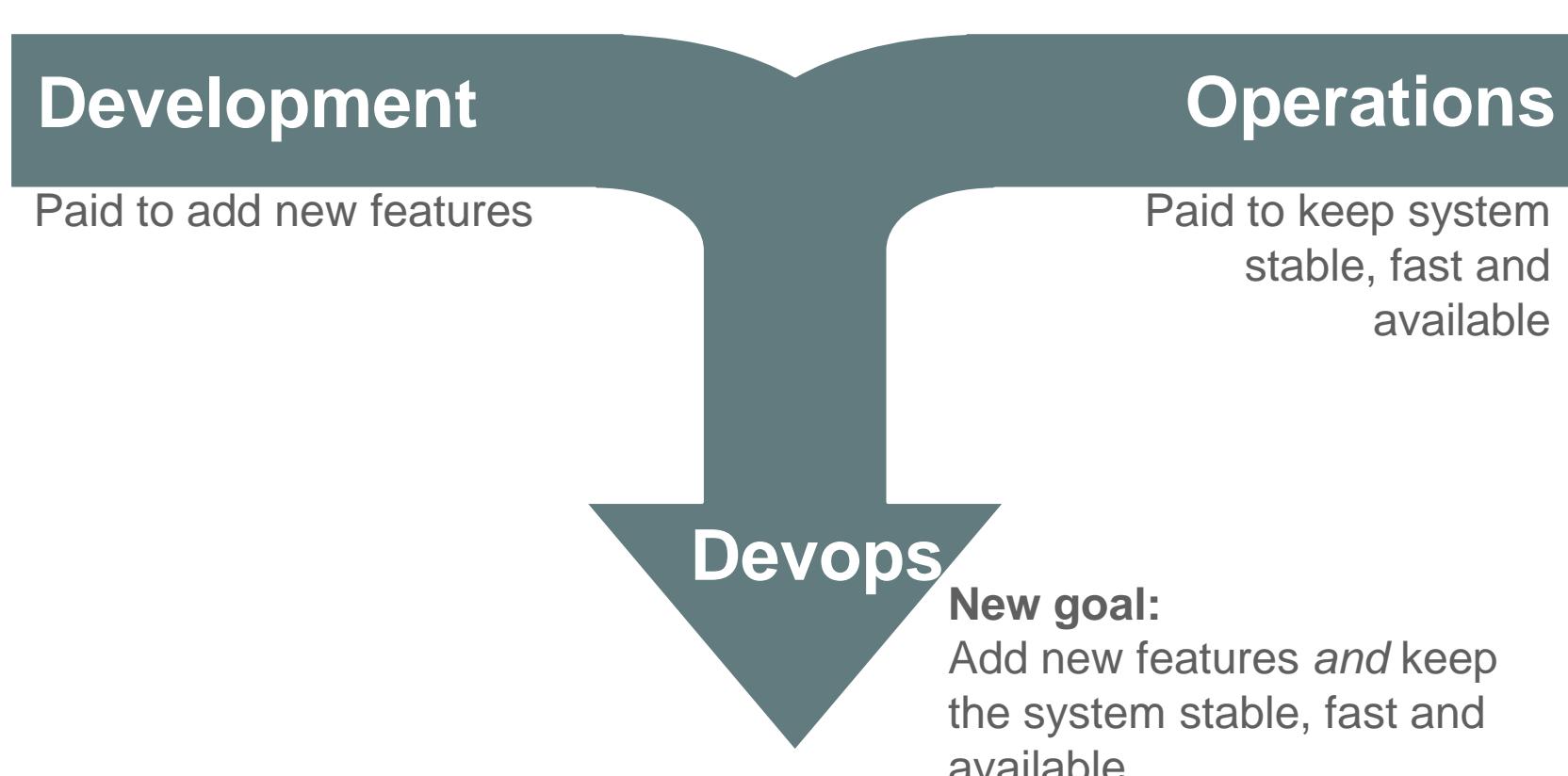


DevOps

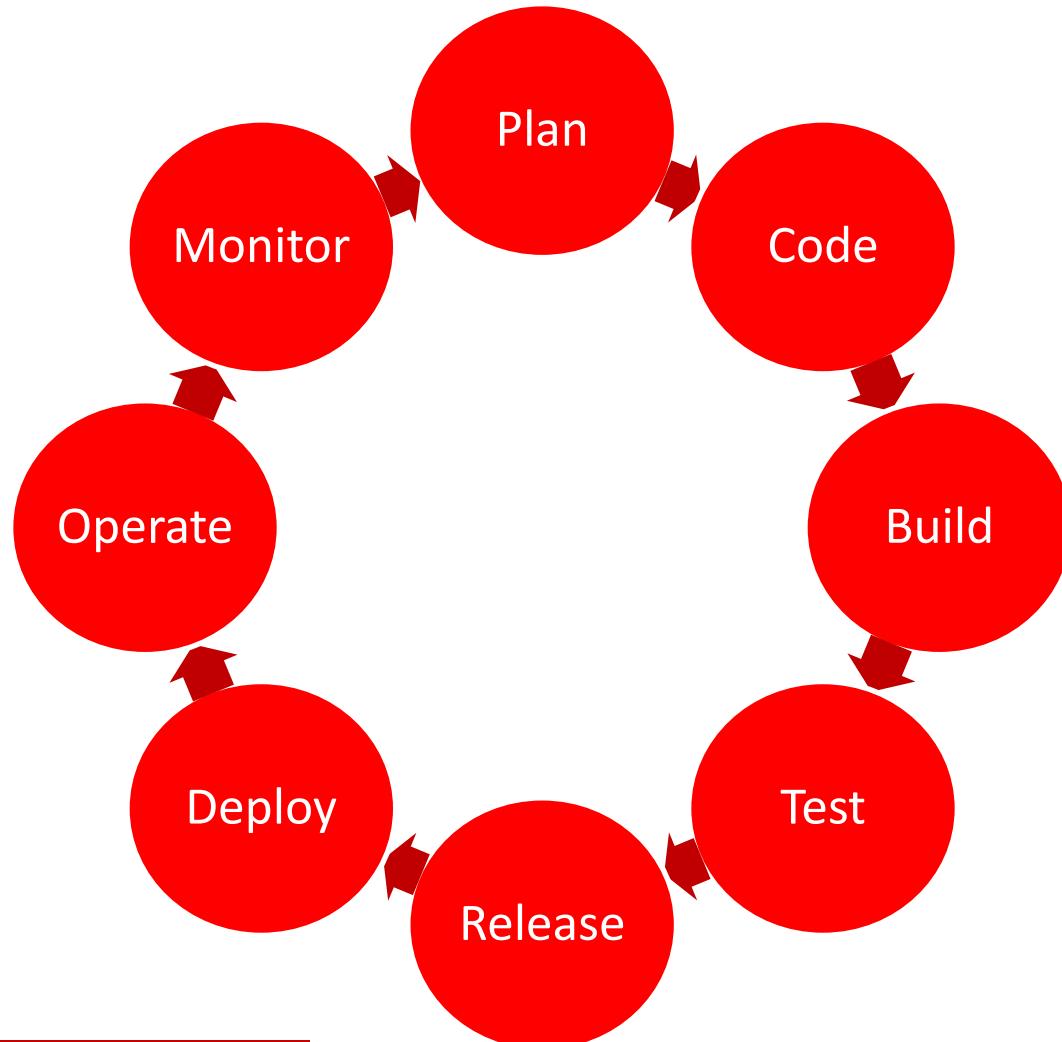


DevOps Principles

Cultural movement enabled by technology

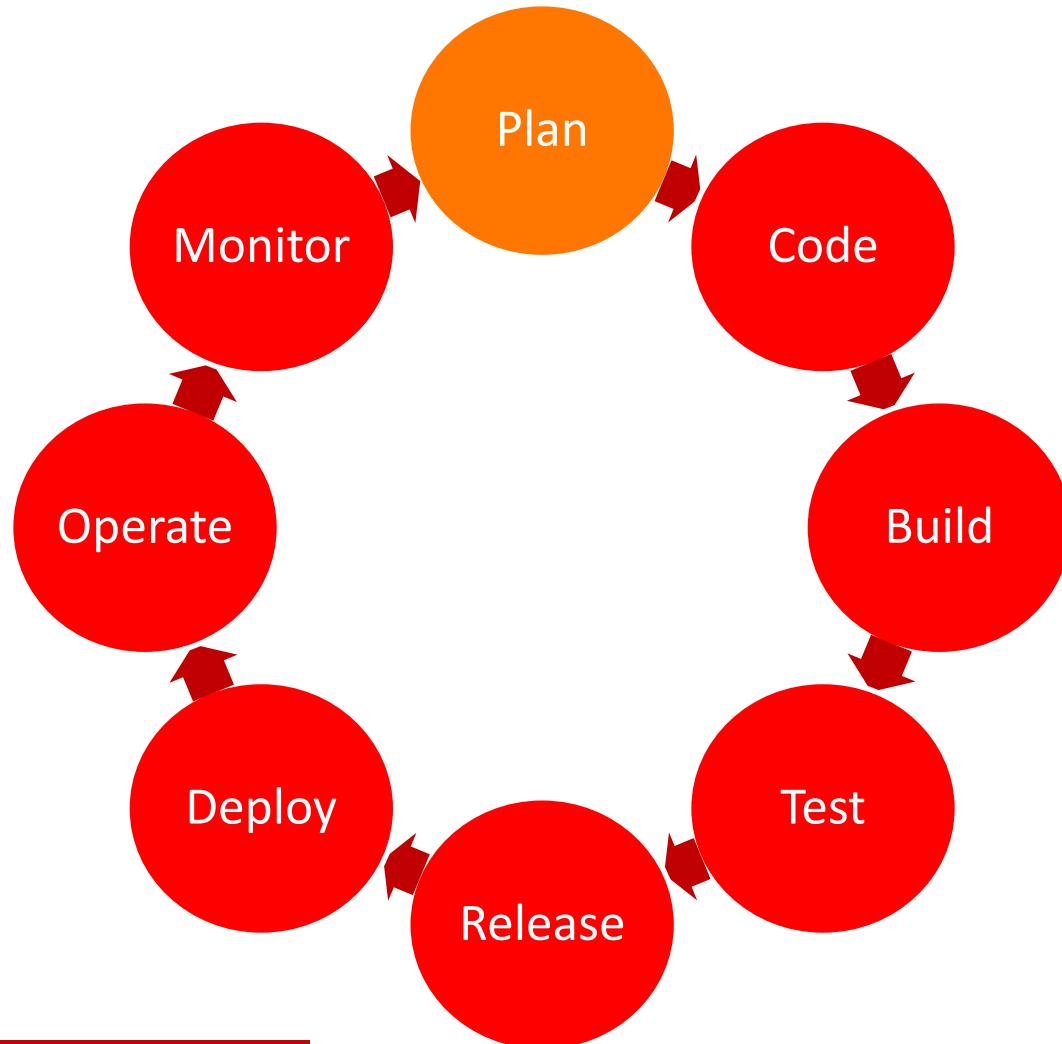


DevOps



- DevOps is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a **culture** and **environment** where **building, testing, and releasing software, can happen rapidly, frequently, and more reliably.**
— Wikipedia

Developer Cloud Service



- Track Issues
- Agile Project Management
- Wikis

Issue Tracking

The screenshot shows the Oracle Developer Cloud Service Agile interface. At the top, there's a navigation bar with links for Home, Code, Snippets, Merge Requests, Issues, Agile (which is selected), Build, Deploy, Wiki, and Administration. Below the navigation bar, there's a search bar labeled "Search Not Available". On the left, there's a "Team" dropdown and a green button labeled "+ New Sprint". In the center, there's a section for "Rel13Sprint" which contains 3 issues: "Defect 3" (Add UI test for value validation on Salaries), "Task 2" (Column Title on Employee Table not matching), and "Defect 1" (REST API for Employees only returns 20 rows). To the right of the sprint details, there are buttons for "Edit sprint...", "15", "0", and "15". Below the sprint details, there's a "New Issue" button. At the bottom, there's a "Backlog" section with 0 issues, a message stating "The backlog contains no issues.", and another "New Issue" button.

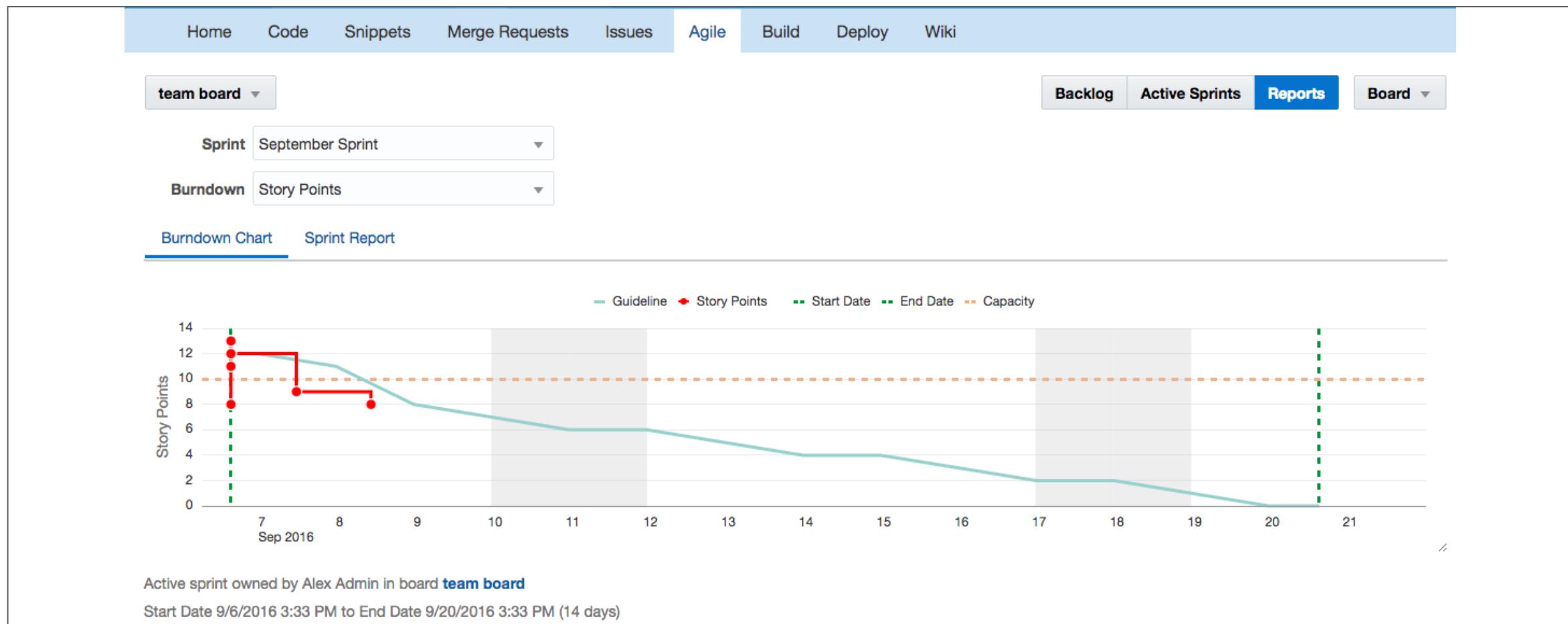
Issue Type	Description	Owner	Priority
Defect 3	Add UI test for value validation on Salaries	Shay Shmeltzer	5
Task 2	Column Title on Employee Table not matching	Shay Shmeltzer	2
Defect 1	REST API for Employees only returns 20 rows	Dana Singletary	8

Agile and Sprint Planning

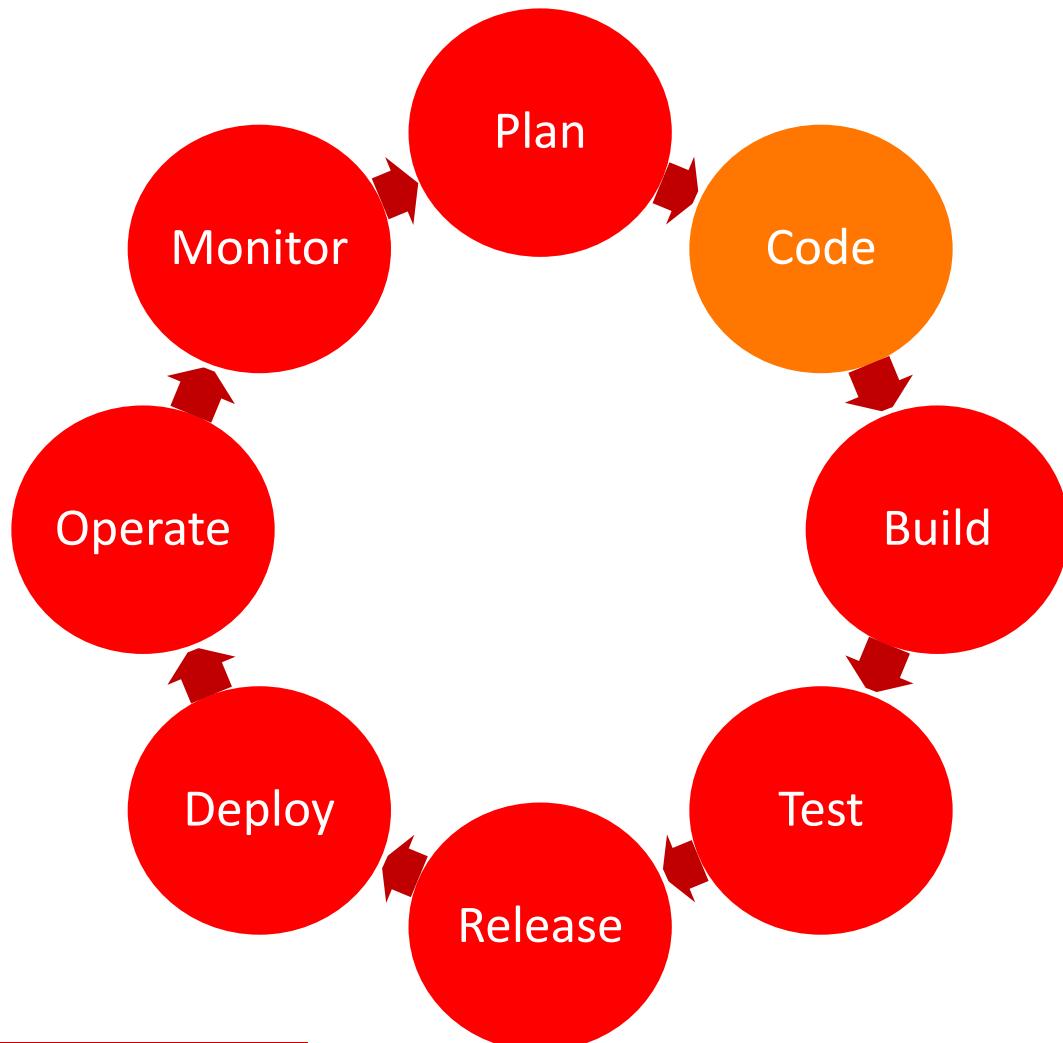
The screenshot shows the Oracle Developer Cloud Service interface for the EmployeeACCS project. The top navigation bar includes links for Home, Code, Snippets, Merge Requests, Issues, Agile, Build, Deploy, Wiki, and Administration. The Agile tab is selected. Below the navigation is a search bar and a 'Team' dropdown. A banner indicates a 'Complete Sprint' from Rel13Sprint (10/6/2016 - 10/20/2016). The main area displays a sprint backlog with three columns: To Do (13), In Progress (1), and Completed (0). The 'In Progress' column has one item assigned to user dana.singleterry@oracle.com, which is a defect related to the REST API returning only 20 rows. The 'Completed' column is currently empty. The 'To Do' column has one item assigned to user shay.shmeltzer@oracle.com, which is a task related to validating UI tests for salaries.

To Do (2)	In Progress (1)	Completed (0)
dana.singleterry@oracle.com Defect 1 REST API for Employees only returns 20 rows 8 UNCONFIRMED	shay.shmeltzer@oracle.com Task 2 Column Title on Employee Table not matching 2 ASSIGNED	

Agile Reports for Progress Tracking



Developer Cloud Service



- Git Repositories
- Coding in the Cloud
- Code Review
- IDE Integration



Manage Code Repositories

The screenshot shows the Oracle Developer Cloud Service interface for managing code repositories. The top navigation bar includes the Oracle logo, the service name, and a user dropdown. Below the header, the application name "summitADFApp" is displayed with a checkmark icon. A search bar labeled "Search Code" is on the right.

The main menu bar contains links for Home, Code (which is selected), Merge Requests, Issues, Build, Deploy, Wiki, and Administration. The "Code" section shows a repository named "summitadfapp.git" under the "master" branch. A tooltip indicates "Click to add description of this repository." The repository listing shows the following files and their details:

File	Description	Author	Date
.adf / META-INF	first check-in of project	shay.shmeltzer	April 21 2015 3:34 PM -0700
Model	first check-in of project	shay.shmeltzer	April 21 2015 3:34 PM -0700
src / META-INF	first check-in of project	shay.shmeltzer	April 21 2015 3:34 PM -0700
ViewController	updated welcome message to hello	shay.shmeltzer	April 21 2015 3:37 PM -0700
build.properties	just checking in	shay.shmeltzer	April 22 2015 4:12 PM -0700
build.xml	Adding build files	shay.shmeltzer	April 22 2015 4:04 PM -0700
SummitADF.jws	first check-in of project	shay.shmeltzer	April 21 2015 3:34 PM -0700

A context menu is open over the commit for "ViewController", showing options to "Create Branch" and "Create Tag". It also displays the repository URL: <https://shay.shmeltzer%40oracle.com@dev>.

Track Code Changes

The screenshot shows the summitADFAApp interface with the following details:

- Header:** summitADFAApp with a checkmark icon.
- Search Bar:** Search Code with a magnifying glass icon.
- Navigation:** Home, Code (selected), Merge Requests, Issues, Build, Deploy, Wiki, Administration.
- Code History:** Two commits are listed:
 - Commit 1:** Author: shay.shmeltzer, Date: April 21 2015 3:34 PM -0700, Message: first check-in of project. Diff: +1 -1 index.jspf ViewController/public_html.
 - Commit 2:** Author: shay.shmeltzer, Date: April 21 2015 3:37 PM -0700, Message: updated welcome message to hello. Diff: +1 -1 index.jspf ViewController/public_html.
- Diff View:** Shows the difference between the two commits. The changes are:

```
@@ -7,7 +7,7 @@
          id="pt1">
          <f:facet name="center">
            <af:panelTabbed id="pt2">
              <af:showDetailItem text="Welcome" id="sdi1"/>
              <af:showDetailItem text="Hello" id="sdi1"/>
              <af:showDetailItem text="Summit Management" id="sdi2" disclosed="true"
                stretchChildren="first">
```

Conduct Peer Code Review and Merge Requests

The screenshot shows a Git commit history for a project named "summitadapp". The commit "2b46258" by "shay.shmeltzer" was made on April 21 2015 at 3:37 PM -0700. The commit message is "updated welcome message to hello". Below the commit, there are two diff entries for "index.jspf" in the "ViewController/public_html" directory.

Diff 1: +1 -1 index.jspf ViewController/public_html

Diff 2: +1 -1 index.jspf ViewController/public_html

The second diff entry shows a change from line 10:

```
@@ -7,7 +7,7 @@
          id="pt1">
      <f:facet name="center">
          <af:panelTabbed id="pt2">
@@ -10 +10 @@
          <af:showDetailItem text="Welcome" id="sdi1"/>
          <af:showDetailItem text="Hello" id="sdi1"/>
@@ -11 +11 @@
          <af:showDetailItem text="Summit Management" id="sdi2" disclosed="true"
```

A modal dialog titled "Add Comment" is open over the diff view, containing the text "Need to use meaningful names for variable". The "OK" button is visible at the bottom of the dialog.

Code in the Browser*

The screenshot shows the Oracle Developer Cloud Service IDE interface. The left sidebar displays the project structure for 'PartsFinder' with files like .git, .settings, node-server (manifest.json, server.js, web.js), node_modules, .gitignore, .project, accsnode.zip, Gruntfile.js, package.json, partsfinderapi.zip, and README.md. The main editor window shows the 'server.js' file with code for adding topics and comments. A tooltip for the 'acceptNode' function is open, providing its definition and parameters. The bottom navigation bar includes 'Find in Files', 'Working Set', 'Search For', 'Search', and 'Options'.

```
function addTopic(tTitle, tText) {
    console.log("addTopic(" + tTitle + "," + tText + ")");
    var topicId = ++currentId;
    topicList.push({title: tTitle, id: topicId});
    topicDetail[topicId] = {title: tTitle, text: tText, comments: []};
    return topicId;
}
function addComment(topicId, text) {
    console.log("addComment(" + topicId + "," + text + ")");
    topicDetail[topicId].comments.push(text);
}

var id1 = addTopic("Topic 1", "Topic 1 content");
var id2 = addTopic("Topic 2", "Topic 2 content");
addComment(id1, "Good topic");
addComment(id2, "This is a comment");
addComment(id2, "This is another comment");

var server = http.createServer(function (request, response) {
    response.
    response._getResourcesString(key) : String
    response._ROW_STATUSES : String
    response.abbr : String
    response.abort() : undefined
    console.log(abort) : undefined
    console.log(abs(x)) : Number
    var request.accept : String
    request.acceptCharset : String
    request.acceptNode(n) : Integer
    request.accepts(types) : String
    request.acceptsCharsets(charset) : String
    handleRequest(request, response, requestOptions);
});
```

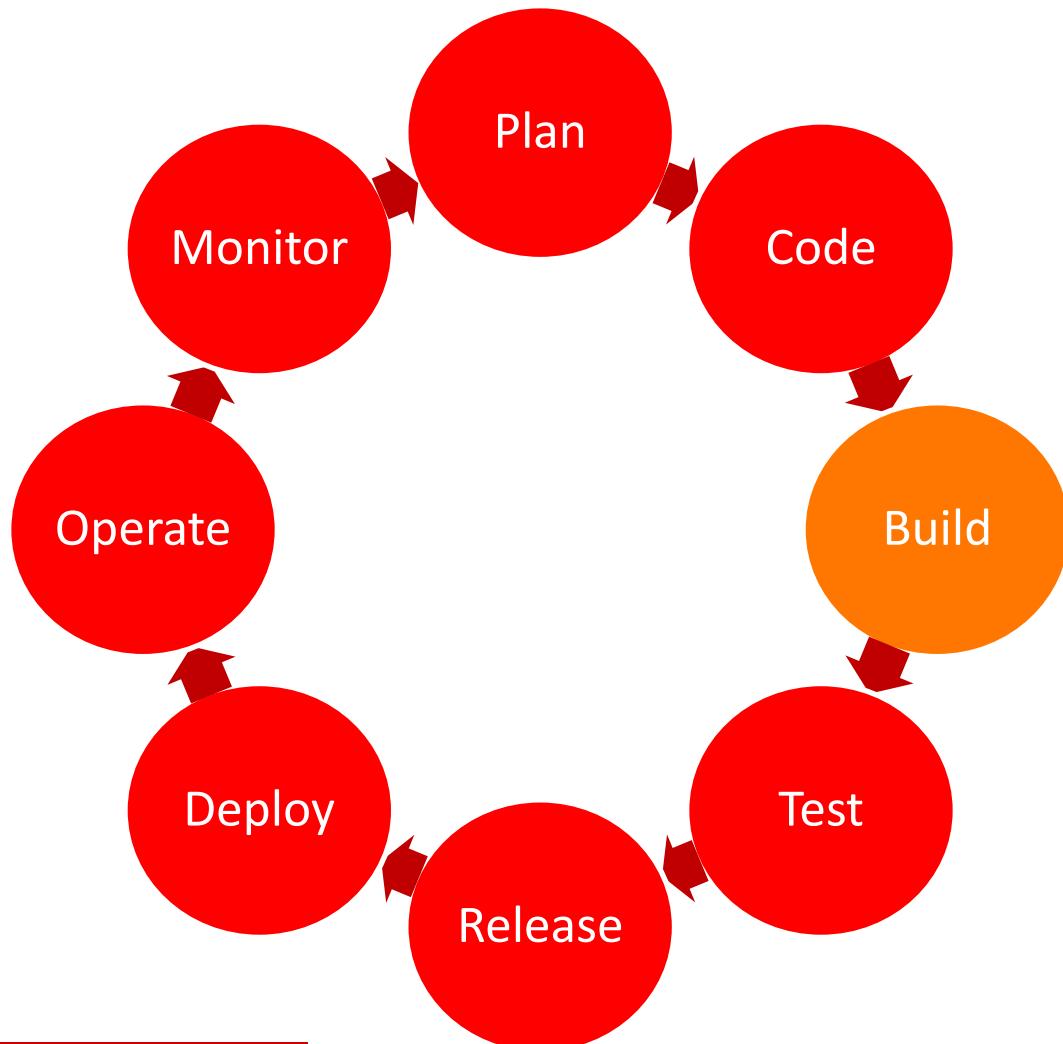
acceptNode
Test whether a specified node is visible in the logical view of a TreeWalker or Nodelerator. This function will be called by the implementation of TreeWalker and Nodelerator; it is not normally called directly from user code. (Though you could do so if you wanted to use the same filter to guide your own application logic.)

Parameters

n Node

***Coming soon**

Developer Cloud Service



- Support Popular Build Frameworks
- Orchestration and Dependencies
- Build Reports and Notifications



Build Dashboard

ORACLE® Developer Cloud Service shay.shmeltzer@oracle.com ?

EmployeeACCS ✓ Search Not Available

Home Code Snippets Merge Requests Issues Agile Build Deploy Wiki Administration

◀ Jobs Overview employees-app-build Build Now Configure Disable Delete

Description

Simple build job to compile the employee webapp, package the jar dependencies, and zip up the manifest.json descriptor.

Console Changes Git Logs Audit

Permalinks

Last | Successful | Completed | Stable

Notifications On Off CC Me

Build History New Build 63% ↗

Status	Build	Time	Duration	Console
⌚	#3	Just now	N/A	↗
✓	#2	16 minutes ago	58 s 780 ms	↗

Artifacts of Last Successful Build

- employees-app
- (all files in zip)

Build Trend

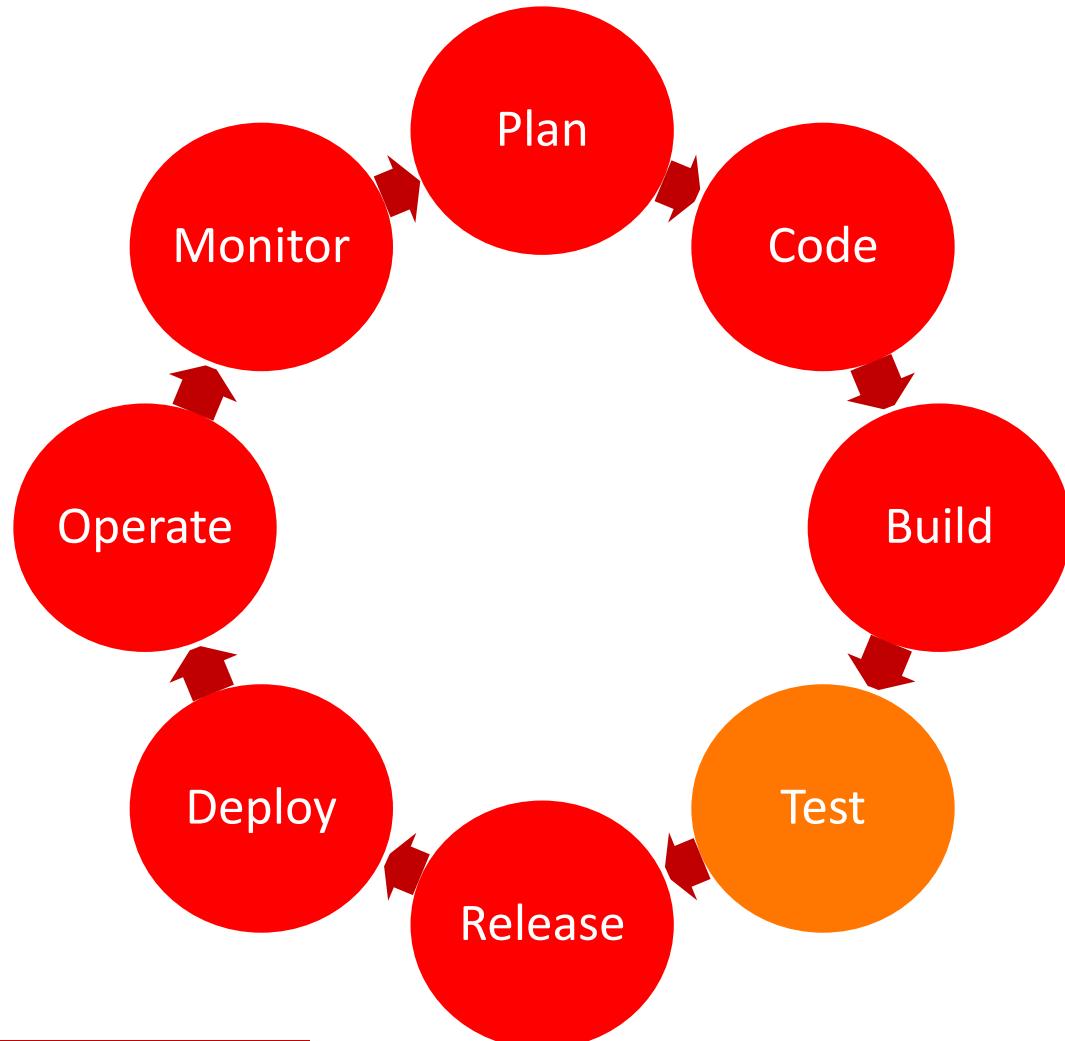
Duration in seconds

Success

Multiple Build Steps Orchestrations

The screenshot shows the Oracle Developer Cloud Service interface for configuring a build job named "employees-app-build". The "Build Steps" tab is selected. A dropdown menu titled "Add Build Step" is open, listing various options: Execute shell, Invoke Ant, Invoke Maven 2 (Legacy), Invoke Maven 3, Invoke Gradle, Invoke NodeJS, and Copy Artifacts. The "Invoke Maven 3" step is currently configured with "Maven 3 (Bundled)" selected, goals set to "clean install package", and a POM file of "employees-app/pom.xml".

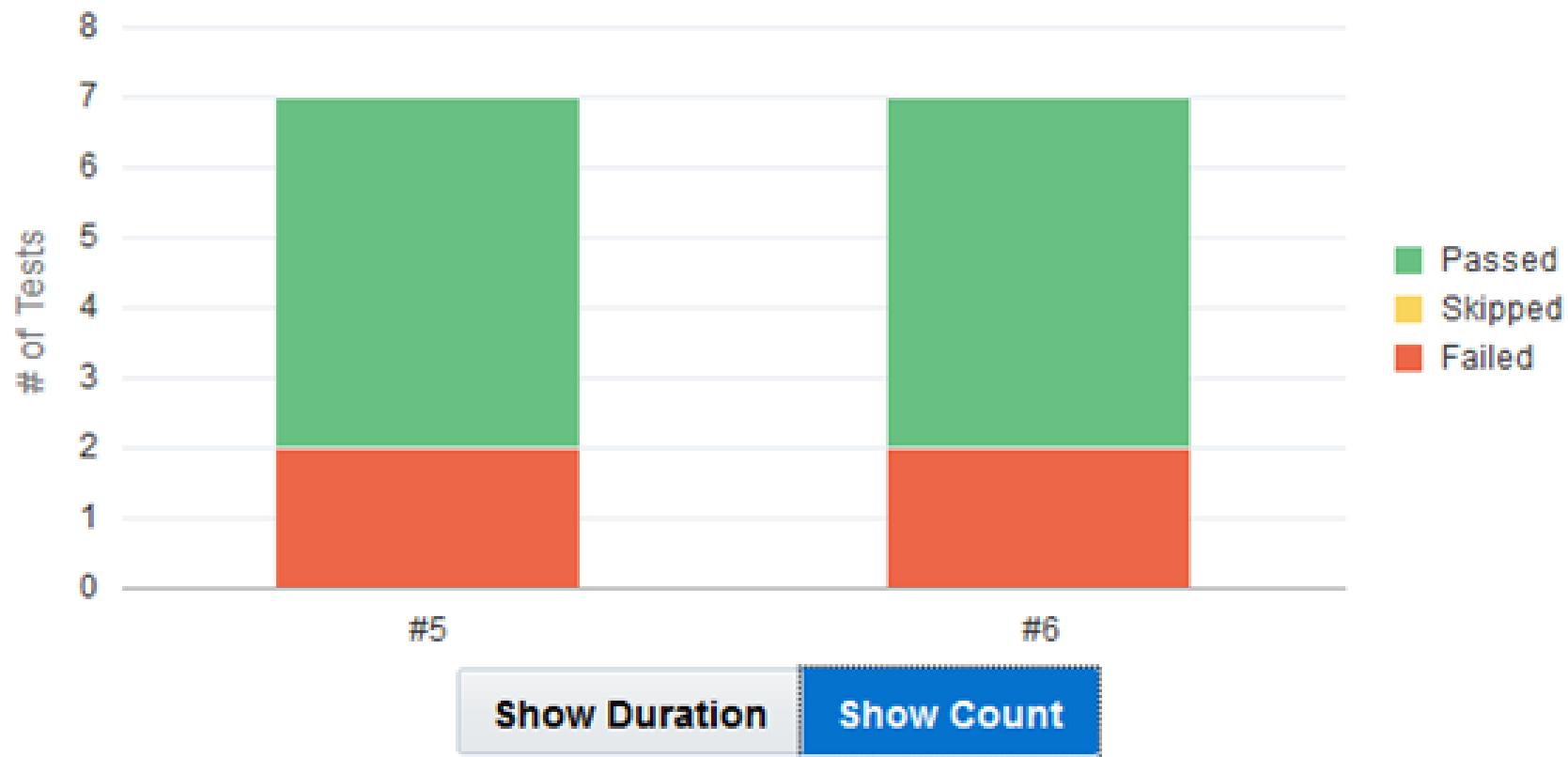
Developer Cloud Service



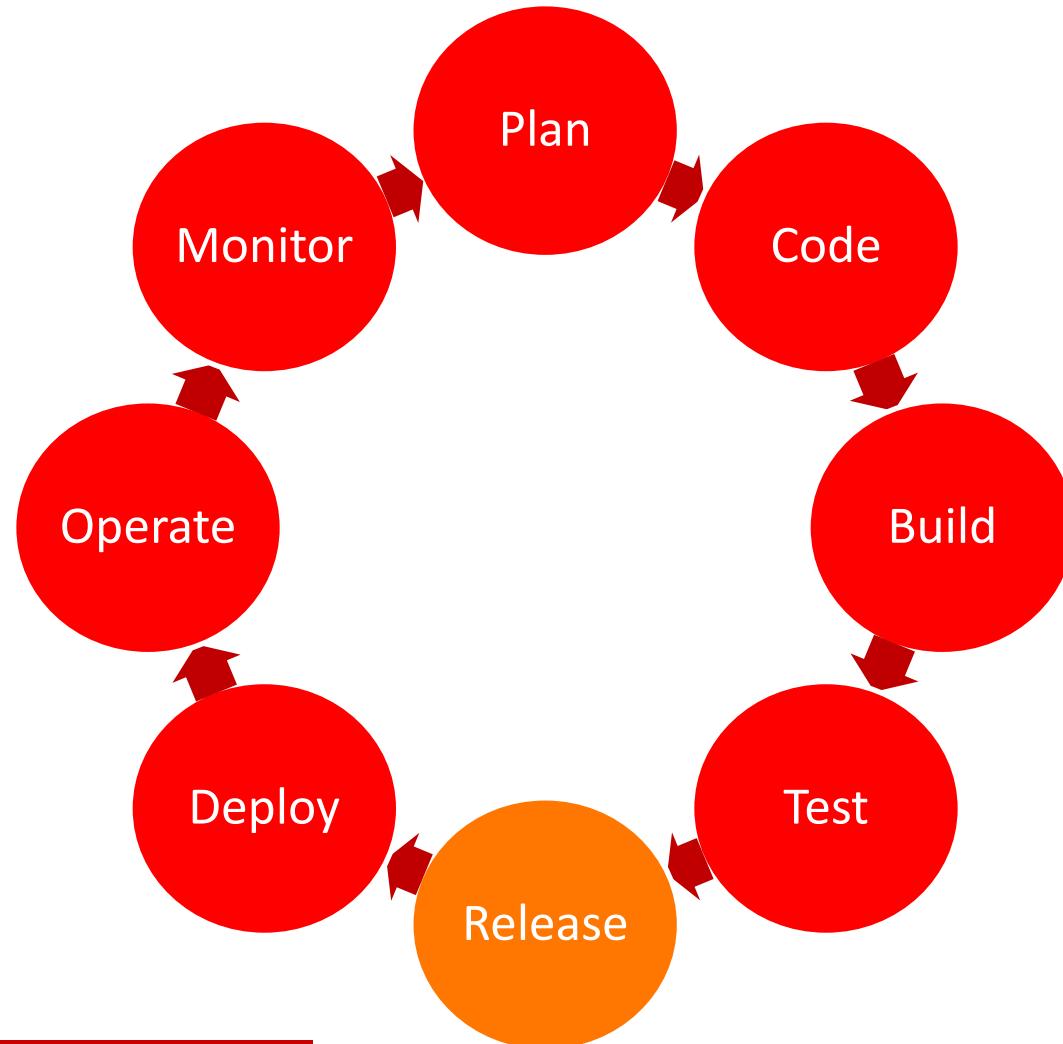
- JUnit
- Selenium
- FindBugs
- QA Deployments



Test Results Dashboard



Developer Cloud Service



- Create packages
- Direct deployment

Deployment Configuration

ORACLE® Developer Cloud Service [shay.shmeltzer@oracle.com](#) [?](#)

EmployeeACCS [▼](#) [Search Not Available](#)

Home Code Snippets Merge Requests Issues Agile Build Deploy Wiki Administration

Edit Configuration [Save](#) [Cancel](#)

Application Name EmployeePortal

Deployment Target em2 / paas124 / greg.stachnick@oracle.com

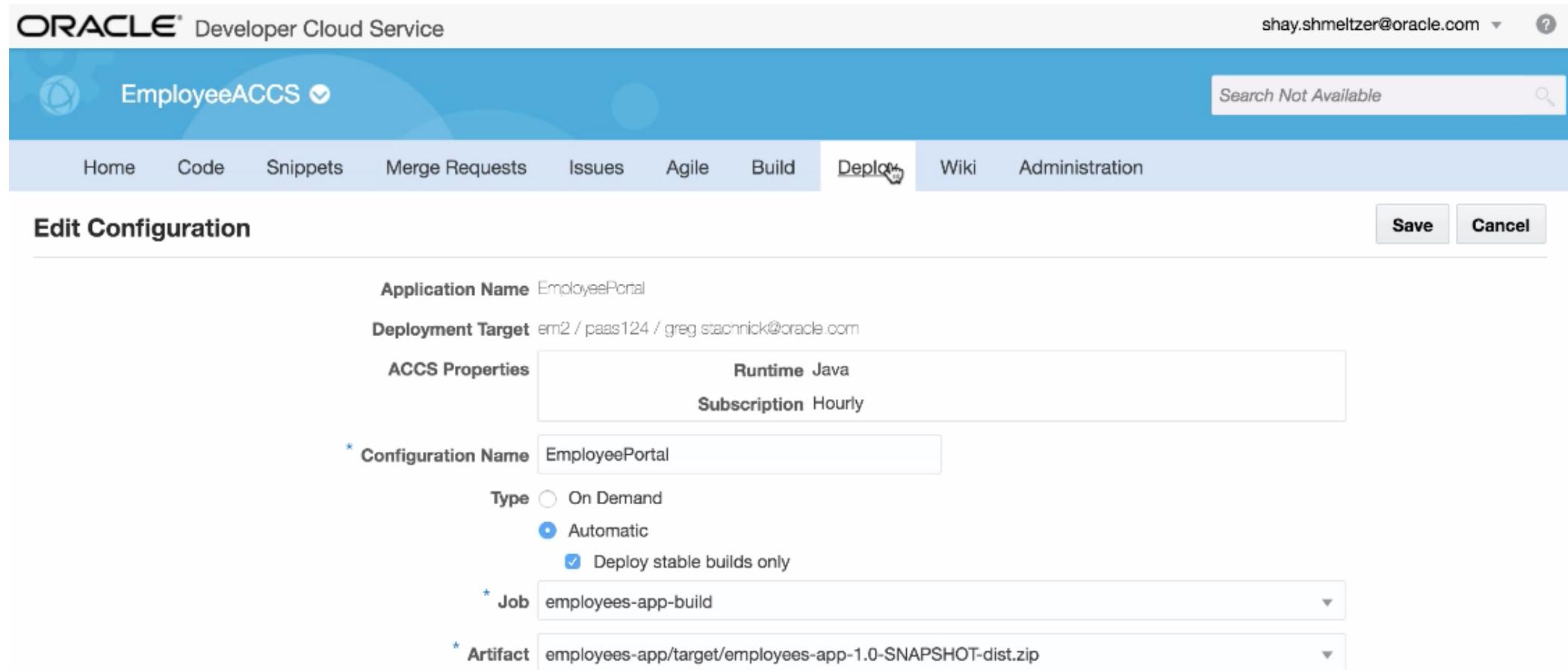
ACCS Properties	Runtime Java
	Subscription Hourly

*** Configuration Name** EmployeePortal

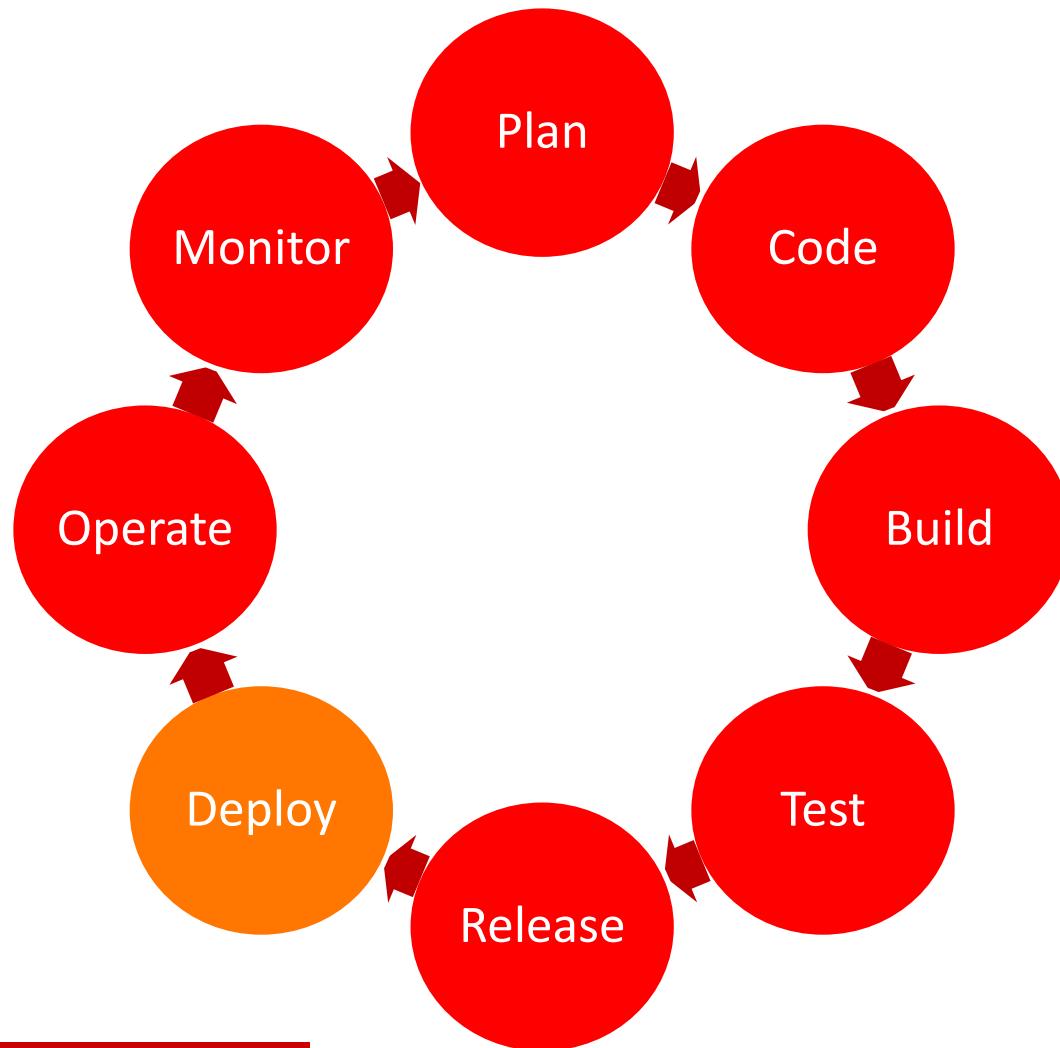
Type On Demand Automatic
 Deploy stable builds only

*** Job** employees-app-build

*** Artifact** employees-app/target/employees-app-1.0-SNAPSHOT-dist.zip

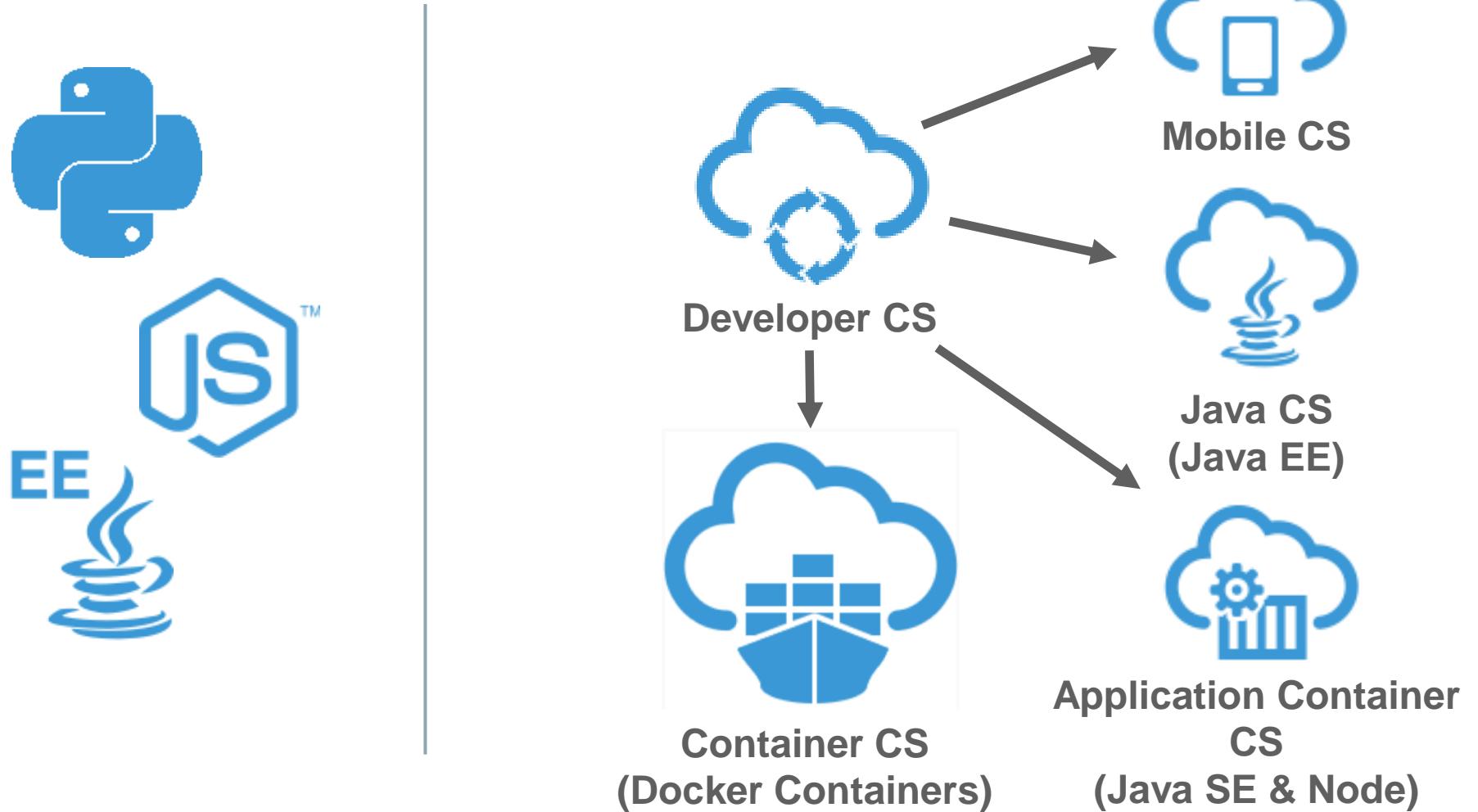


Oracle Cloud Services



- Easily provision deployment platforms
- Java Cloud Service
 - Java EE in the Cloud
- Application Container Cloud
 - Java SE, NodeJS, PHP, Ruby*, Python*
- Mobile Cloud Service
 - Mobile Services
- Container Cloud Service
 - Docker Containers

Polyglot Platform for Running any Workload



Easy Configuration – Fast Provisioning

The screenshot shows the Oracle Cloud My Services interface. At the top, there's a navigation bar with the Oracle logo, 'CLOUD My Services', 'Dashboard', 'Users', and 'Notifications'. Below the navigation bar, on the left, is a sidebar titled 'Oracle Application Container' with a green icon. It has sections for 'Applications' and 'Cloud Functions'. A message says 'You don't have any applications. Use Create Application to start.' Below this is a link 'Application Create and Delete API'. At the bottom of the sidebar, a note states '*Node.js is a trademark of Joyent, Inc. and is used with permission.'

The main content area is a 'Create Application' dialog. It has two tabs: 'Application' (selected) and 'Instance'. The 'Application' tab contains fields for 'Name' (set to 'CorporateDirectory'), 'Subscription' (set to 'Monthly'), 'Application Archive' (set to 'Upload application archive' with file 'corporatedirectory-dist.zip' chosen), and a 'Notes' text area. The 'Instance' tab contains fields for 'Instances' (set to 1), 'Memory (GB)' (set to 1), and 'Version' (set to 'Java SE 8'). At the bottom right of the dialog are 'Create' and 'Cancel' buttons.

Cloud Native Completely Automated with DevOps

Plugging into Existing DevOps Pipelines Seamlessly

Automation Tools



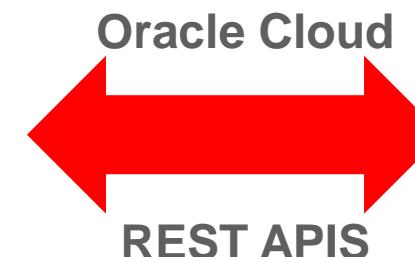
DevOps Pipelines



Command Line tools or curl

```
>psm setup  
>psm list services  
>psm push app ...
```

```
>curl -i -X GET -H  
"Authorization:joe  
@example.com:joePa  
ssword" \ -H ...
```



Service API Catalog

A screenshot of a web browser displaying the Oracle API Catalog Cloud Service. The page title is "ORACLE API Catalog Cloud Service". The main header says "REST APIs for Oracle Cloud Services". Below it are tabs for "Applications", "Platform", and "Infrastructure". Under "Applications", there are sections for "Application Development", "Business Analytics", "Content and Collaboration", and "Data Management". Each section lists specific services like Application Container, Big Data Preparation, etc.

Service APIs

REST APIs for IaaS

- Docker, VMs, Storage, Network

REST API for PaaS

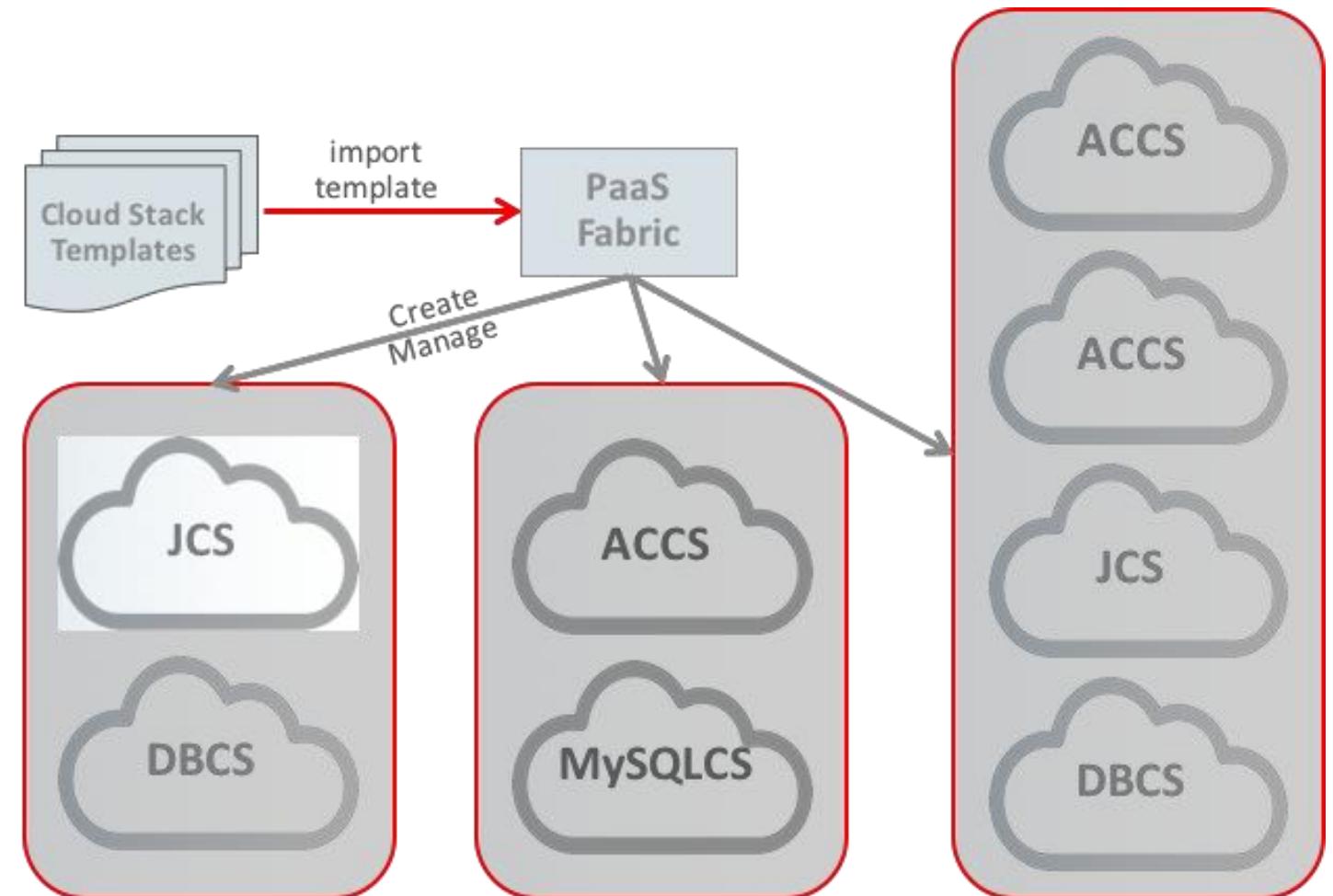
- AppDev, Integration, Mobile, BI, DB...

REST APIs for SaaS

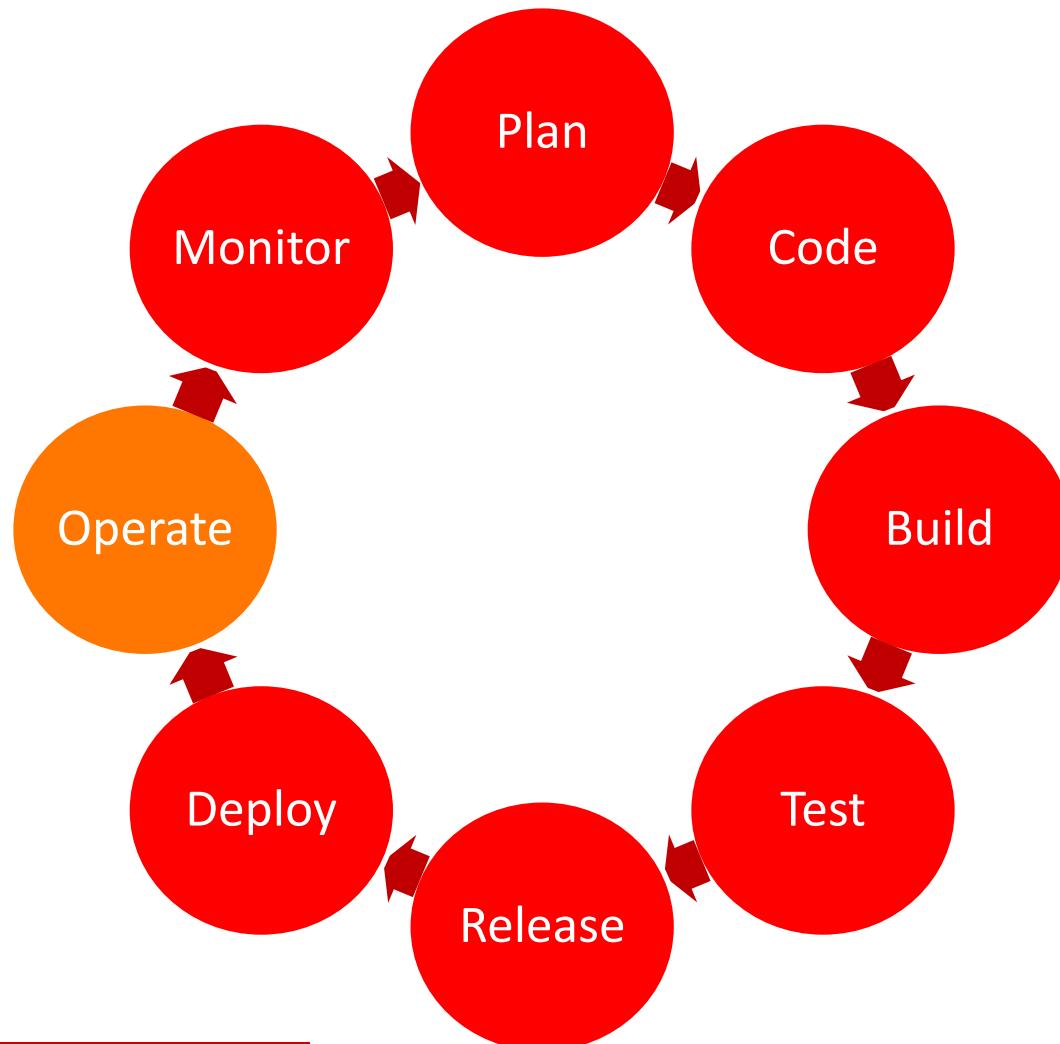
- ERP, Service, Sales, Support ...

Cloud Stack Manager: Manage Many PaaS Services as One

- Manage group of cloud services as a single atomic unit called a Cloud Stack
- Template driven approach to maximize reuse and sharing
- Eliminate manual tasks to create a cloud environment
- Automatically manages dependencies
- True Infrastructure-as-Code for your DevOps needs



Oracle Cloud Services



- Automatic Scaling
- Easy Patching/Upgrade
- Live Reporting

Scale Out & Scale Up

The screenshot shows the Oracle Cloud My Services interface. At the top, there's a navigation bar with the Oracle logo, 'CLOUD My Services', and links for 'Dashboard', 'Notifications', and 'Users'. Below the navigation is a sidebar with a green circular icon containing a cloud and a gear, labeled 'Applications / CorporateDirectory'. The main content area has a blue header bar with 'Overview' and 'Resources' tabs. The 'Overview' tab is selected, showing '1 Instances'. The 'Resources' tab shows 'Instances' and 'Memory (GB)' settings, with '2' instances and '1' GB memory currently selected, with 'Apply' and 'Cancel' buttons below. The 'Instances' section lists one instance named 'web.1' with '1GB' memory. On the left, a sidebar also lists 'Deployments' (0), 'Service Bindings' (0), and 'Environment Variables' (0). The bottom right corner shows the timestamp 'As of Feb 1, 2016 7:54:03 PM UTC'.

As of Feb 1, 2016 7:54:03 PM UTC

Overview

1 Instances

Resources

Instances Memory (GB)

2 1

Apply Cancel

Instances

Name:	Memory:
web.1	1GB

Deployments

0 Service Bindings

0 Environment Variables

Automatic Scaling Settings

The screenshot shows the Oracle Java Cloud Service interface for managing automatic scaling rules. The main navigation bar at the top includes a cloud icon, the service name "Oracle Java Cloud Service / SJCSC321", and a "Auto Scaling" link. Below the navigation, there are sections for "Rules" and "Alert History". A modal dialog box titled "New Rule" is open in the center, containing the configuration for a new scaling rule.

New Rule

Scaling operation to be performed on this service.

Perform **Scale Out** to Maximum Cluster Size of **5**

whenever **Average** of CPU Utilization is \geq **80 %**

for at least **1** consecutive period(s) of **5 minutes**

on **All** VM instances

and wait for **30** minutes of cool down period.

Create **Cancel**

One Click Upgrades

As of Jul 4, 2016 4:19:31 PM UTC 

Overview

2 Instances

Deployments

0 Service Bindings

0 User-defined Variables

Administration

2 Updates Available

Updates 

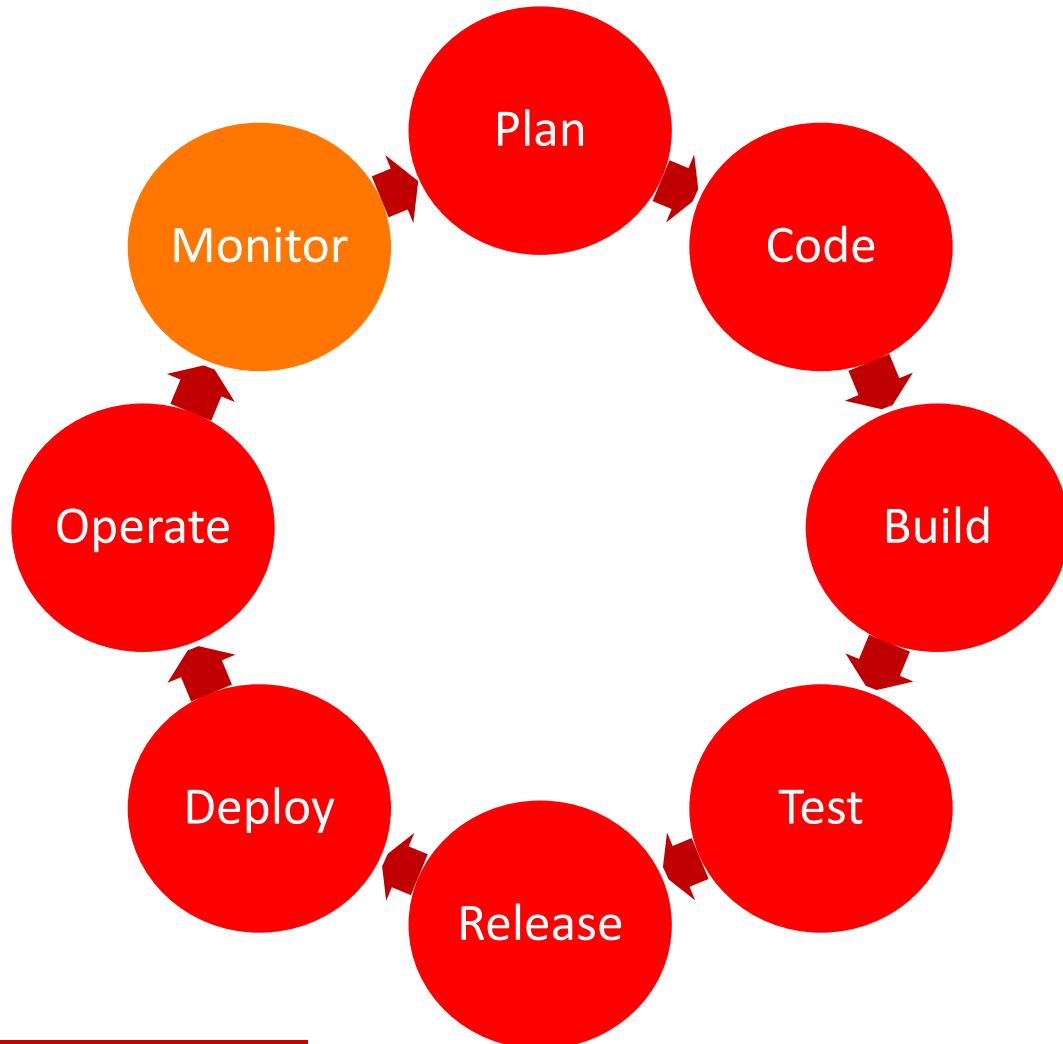
Current Version: Java SE 8u71

Available Updates

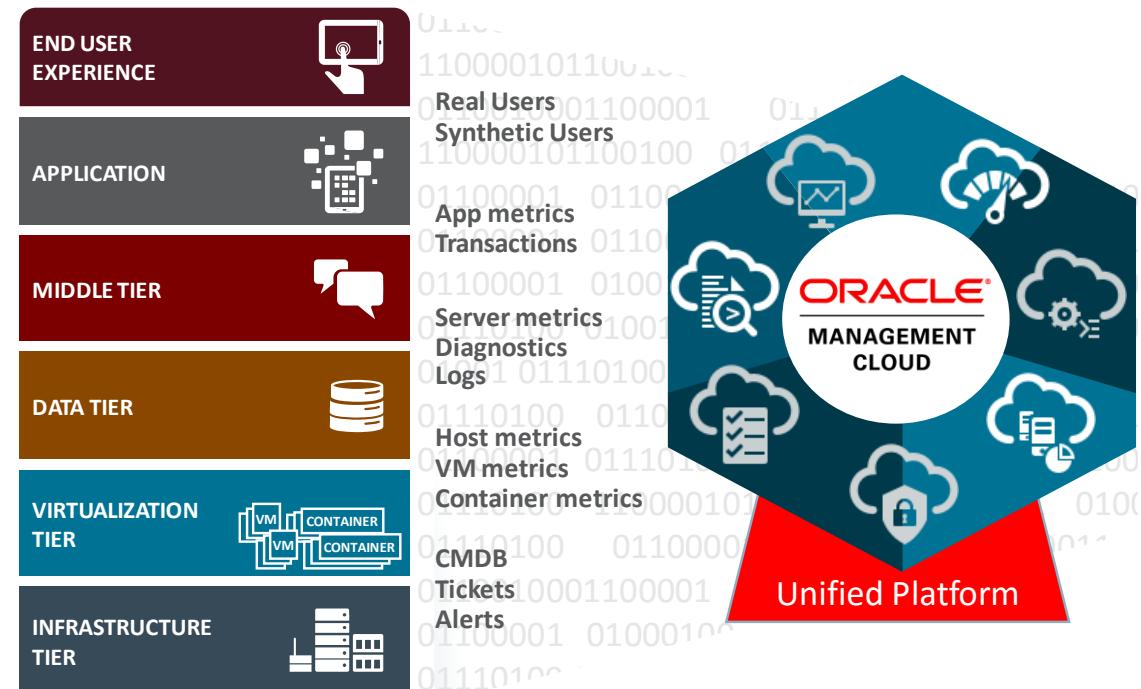
	Runtime: Java SE 8u91 Release Notes	Release Date: Jul 4, 2015 12:00:00 AM UTC	Update
	An Application Container Cloud update is available. Your application will be automatically upgraded when you make a change that requires a restart, such as updates to the deployment environment or scaling up, or you can upgrade now by clicking the Restart button.		

▶ **Update and rollback history**

Oracle Management Cloud



- Intelligent management for heterogeneous environments
- Unified cross-stack operational data



Rapid Troubleshooting Across The Stack

ORACLE Management Cloud
Application Performance Monitoring

Server Request

/RideShare/checkout

Type: AppServer
Deployment: RideShareEar
Servlet: oemwlssrv.oracleleads.com:9001

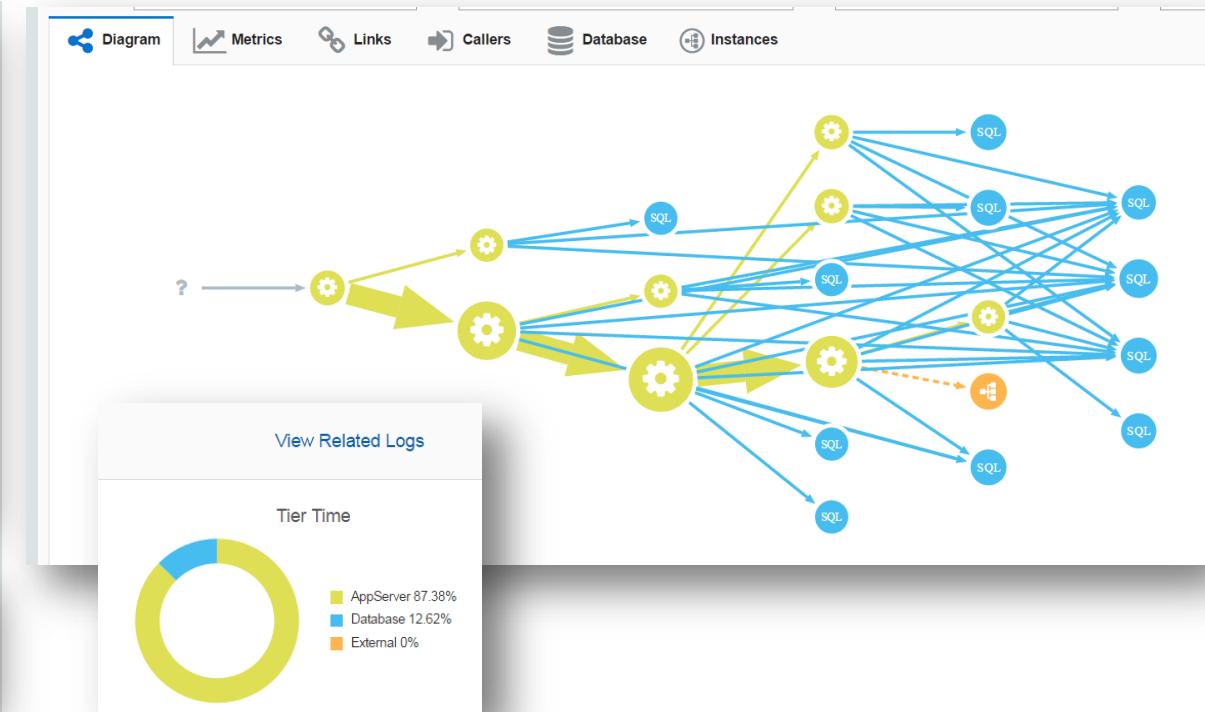
REQUEST RESPONSE TIME
463.81 ms avg
↑<1% prior
Max 923 ms
Min 188 ms

TIER AVERAGE RESPONSE

Category	Percentage
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.



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

Why Choose DevOps with Oracle?

1

FAST

Provisioning end-to-end Continuous Integration and Continuous Deployment environments in minutes

2

COMPREHENSIVE

Pluggable across customer stages and technology choices, and includes both development and deployment technologies.



Developer Automation

Developer Cloud Service: Collaborative Agile Development & Continuous Delivery

Oracle Advantage

- Complete – agile management, collaborative development, and CI/CD
- Standards Based
- IDE Integration
- Full development lifecycle

Less than

1 min

TO PROVISION ENVIRONMENTS

Supported Technologies





Cloud Deployment

Java Cloud Service, Application Container Cloud Service , Container Cloud Service

Oracle Advantage

- Offers a ready-built platform for deploying your applications to. You deploy your application - Oracle worries about provisioning and managing
- Same code powers both private and public versions; very easy portability
- Near instantaneous environments for Dev Test

Up to
59%
**LOWER COST
THAN ON-PREM**

Supported Technologies





Cloud Management

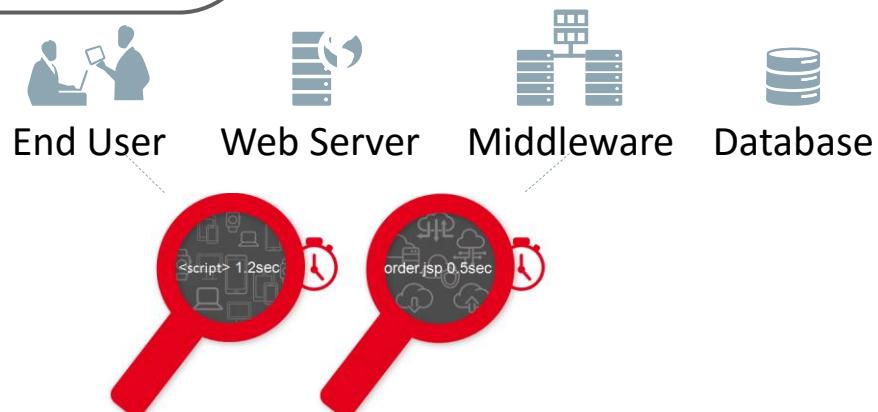
Application Performance Management

Oracle Advantage

- Triage Cross-Tier Issues from End-User to Code
- Unify Monitoring Visibility Across Dev and Ops
- Ensure Good User Experience

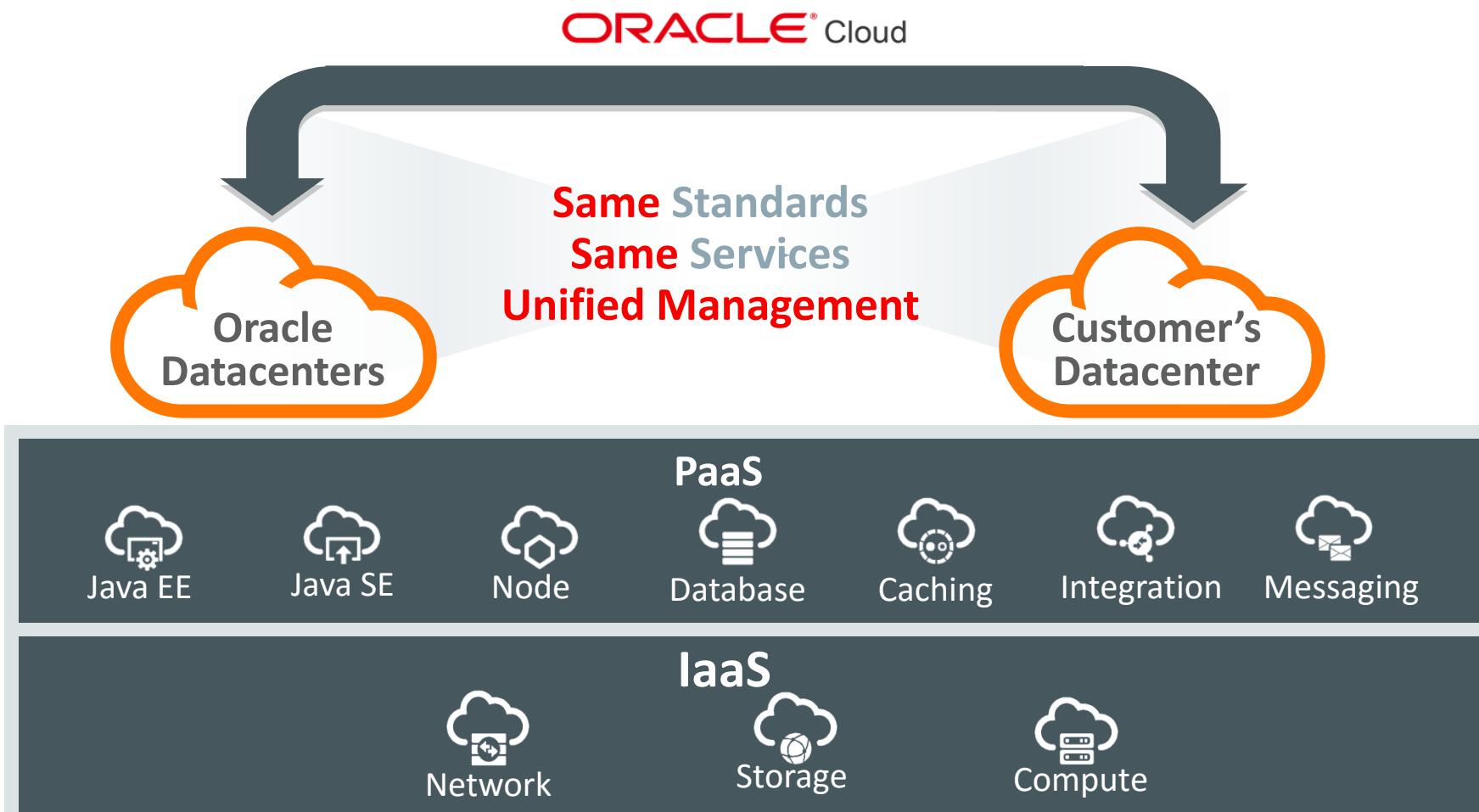
**QUICKLY
DIAGNOSE
Hundreds of
lines of code**

***For faster
trouble
shooting***

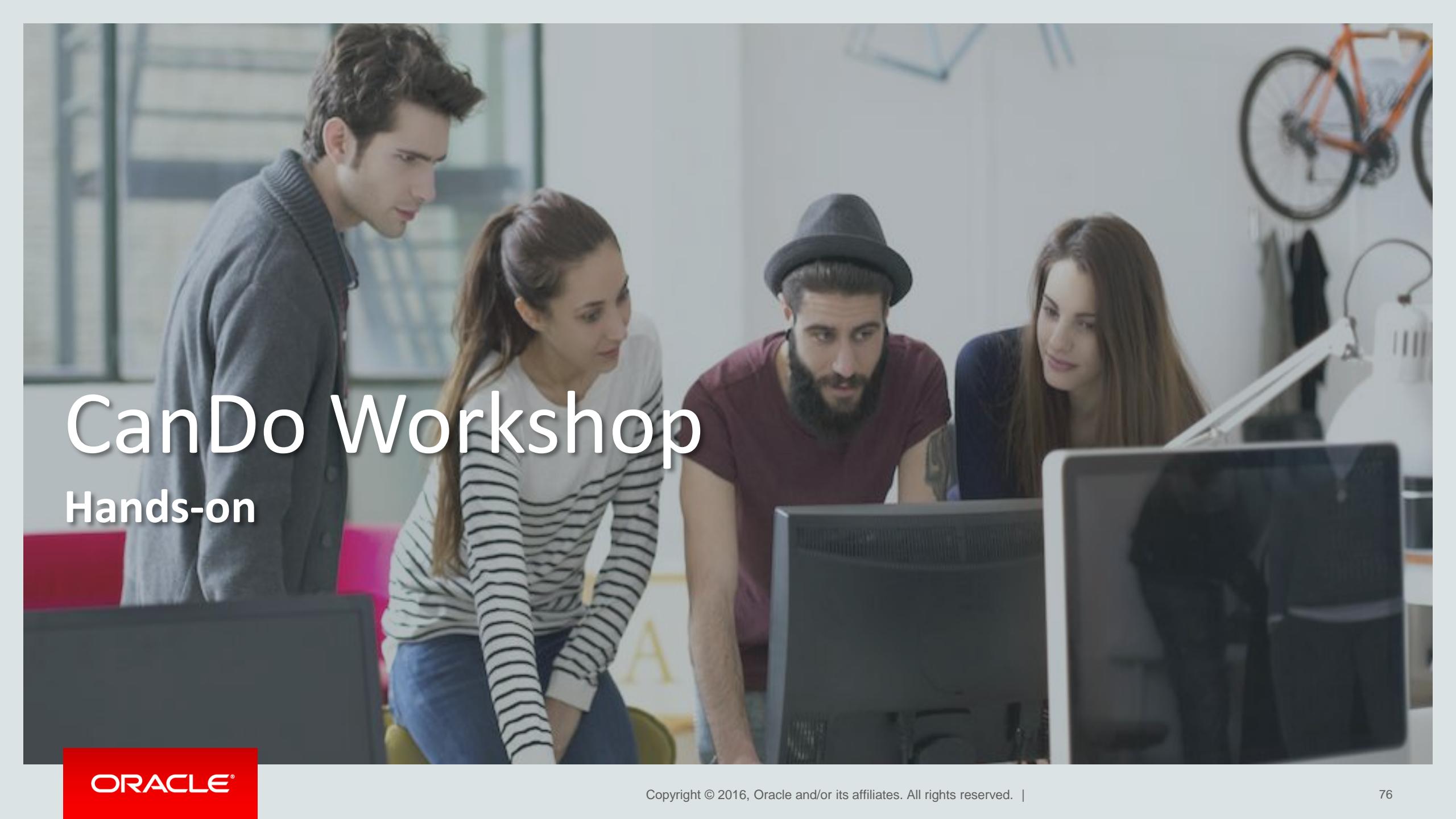


Oracle Cloud Machine

Complete deployment choice



- Oracle Cloud operated and **delivered as a service** behind your firewall
- **Same PaaS and IaaS** software, same updates as Oracle Cloud
- Same cost-effective **subscription** pricing model as Oracle Cloud
- Conforms to regulatory, privacy, legal, and **business requirements**

A photograph showing four people in a modern office environment. A man in a grey sweater is leaning over a desk, looking at a computer screen. A woman in a white and black striped shirt is seated next to him. In front of them, another man with a beard and a fedora hat is also looking at the screen. To his right, a woman with long brown hair is partially visible, also looking at the screen. They are all focused on the computer monitor. The background shows a window with horizontal blinds and a bicycle hanging on the wall.

CanDo Workshop

Hands-on

Hands-on preparation

- You must have active Oracle Cloud account for Oracle PaaS services
 - Sign-In to Oracle Cloud with your account details and make sure that on the Dashboard you have at least the following services: Java, Database, Developer, Application Container
 - If your Cloud account is new -> Please set up **Replication Policy** on Storage Service!
 - see [Selecting a Replication Policy for Oracle Storage Cloud Service](#)
- You must have Oracle VirtualBox installed on your laptop
 - Import the given Oracle Virtual Machine (OVM) into VirtualBox
 - On the OVM Linux Desktop
 - if you are not in an Oracle VPN Network then click on the icon Oracle Proxy OFF
 - click on the icon **Update Demos** to get the latest content
- You must have access to GitHub (<https://github.com>):
 - Access the CanDo Workshop Git repository on: <https://github.com/gmoykin/oracle-cando-workshop>
 - Read the **README.md** file located in the main folder **/oracle-cando-workshop**

CI/CD and Microservices with DevCS, ACCS, Eclipse and Git

Create and Deploy the SpringBoot Microservice with DevCS & ACCS

- On **oracle-cando-workshop** Git repository follow the instruction in the **springboot-sample/README.md** file
- When configuring the ACCS deployment (*New Deployment Configuration*) provide your ACCS details

CI/CD and Microservices with DevCS, ACCS, Eclipse and Git

Using Agile Methodology in DevCS

- Open your DevCS *springboot-sample* project
- On **oracle-cando-workshop** Git repository follow the instruction in the **agile/README.md** file
- You can play and improvise with Agile instruction to match your *springboot-sample* project

CI/CD and Microservices with DevCS, ACCS, Eclipse and Git

CI/CD using Eclipse with DevCS & ACCS

- On **oracle-cando-workshop** Git repository follow the instruction in the **oepe/README.md** file
- You can play and improvise with Agile instruction to match your *springboot-sample* project

CI/CD and Microservices with DevCS, ACCS, Eclipse and Git

CI/CD using Git client with DevCS & ACCS

- Prerequisites:
 - You have created the **springboot-sample** project in DevCS
 - You have configured a **Build** for the **springboot-sample** app based on *SCM polling schedule*
 - You have configured a **Deploy** configuration with Automatic deployment of **springboot-sample** app to ACCS
 - You are familiar with using Agile Methodology in DevCS
- Continuous build integration using Git with DevCS & ACCS
 - On **oracle-cando-workshop** Git repository follow the instruction in the **gitclient/README.md** file
 - You can play and improvise with Agile instruction to match your projects

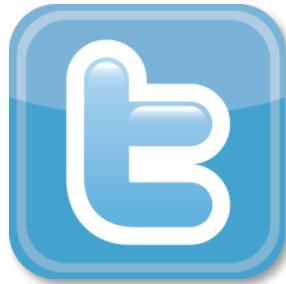
Deploy Java EE application to Oracle Java Cloud Service

- On **oracle-cando-workshop** Git repository open the main **README.md** file and follow the instruction of the topic **Deploy Java EE application to Oracle Java Cloud Service**

Migrate WebLogic applications to Java Cloud Service

- On [oracle-cando-workshop](#) Git repository open the main **README.md** file and follow the instruction of the topic **Migrate WebLogic 10.3.6 (on premise) Application to Java Cloud Service with App2Cloud tool**

Follow us on Social Media



<https://cloud.oracle.com>

<https://cloud.oracle.com/delevoper>

<http://oracle.com/cloud>

www.youtube.com/user/oracle/

Integrated Cloud Applications & Platform Services