

# Oracle Application Container Cloud Service Overview

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Objectives

After completing this lesson, you should be able to:

- Get an overview of Oracle Application Container Cloud
- List the unique features of Oracle Application Container Cloud
- Describe how to build, zip, and deploy applications to the cloud
- Get an overview of Java Cloud Service
- Describe the features of Java Cloud Service



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Oracle Application Container Cloud Service



An open, highly available, Docker container-based elastic polyglot cloud application platform

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

The Application Container Cloud Service is for running your applications in the cloud. It is a simplified experience that doesn't make you deal with virtual machines and configuring networking. It simply lets you deploy your application, scale it, and make it available to your users.

## Oracle Application Container Cloud

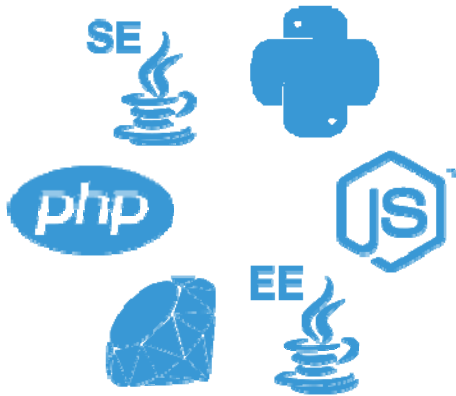


ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- Is a simple and easy-to-use deployment platform for Java SE and Node applications
- Is an open platform—use any application frameworks and libraries
- Runs applications in Docker containers for reliability and scalability

## Polyglot Platform



Deploying applications to a selection of popular language runtimes is supported.

- Latest release supports Java SE, Java EE Web Apps, Node.js, and PHP.

You can leverage unique Oracle Java SE features.

- Immediate access to platform upgrades, security, and platform optimizations
- Continued commercial support for Java SE versions no longer receiving public updates

Node access to Oracle Db is provided with open source database driver

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- Application Container Cloud supports multiple languages. So you have a choice of Java SE or Java EE web applications, Node.js, PHP, and others coming in the future. Oracle Application Container Cloud provides the Oracle JDK, which means that this is fully commercial Java with fully commercial support. And what's nice about the platform is that as new versions of Java come out with new security upgrades and patches, we make them available to you for upgrade immediately. So it's very easy to upgrade.
- On the node side, there's the Oracle database driver that the Oracle team has built. It's an open source project that's on GitHub. You can get it for yourself and use it for accessing Oracle Database from Node.js. But for a simplified deployment experience, Oracle actually includes the latest version of that driver in the environment for Node. So your Node applications can easily access the Oracle Database, which is available in the Oracle Database Cloud Service.

## Open Platform



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

This is a completely open platform; so you build your application and use whatever frameworks you like. Whether they are open source or commercial, that's really not a problem.

## Container-Based Application Platform as a Service



Applications run on Oracle Linux in Docker containers.

### Stateless Applications

- Ephemeral disk
- Permanent storage through database or storage service

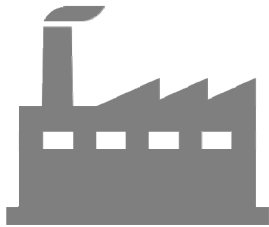
ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Docker provides a nice runtime environment per application. It seals them off from other applications so that they are independent; they don't interfere with one another.

Disks inside the Docker containers are ephemeral. So the file systems are available for your application to use for, say, temporary caching of files and of data, as part of a conversation with a client. But these disks are ephemeral, which means that when the container is shut down, which could happen because you deploy a new version of your application, or you restart them yourself, the disks are lost. So any storage requirements that you have for permanently stored data or durable data need to be put into permanent storage outside of the container, that is, in Oracle Database Cloud Service, for example, or Oracle Storage Cloud Service, or any other external service that has data storage that is durable. But data inside the containers itself is not durable and is lost when the container is restarted.

# Build



# Zip



# Deploy!

**ORACLE**

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Build

- Use your favorite or corporate standard build system to produce binaries and deployable resources.

## Zip

- Zip up all binaries, scripts, HTML files, images, and so on, that make up your application. The structure of the zip is entirely up to the user—we have no opinion on structure.

## Deploy

- Deploy the application archive (zip) to the platform and tell us how to start the application. This could be “java -jar”, “java-classpath ... <main>”, “node myapp.js”, or “sh bootmyapp.sh.”



## Deploy: Application Archive (Zip)

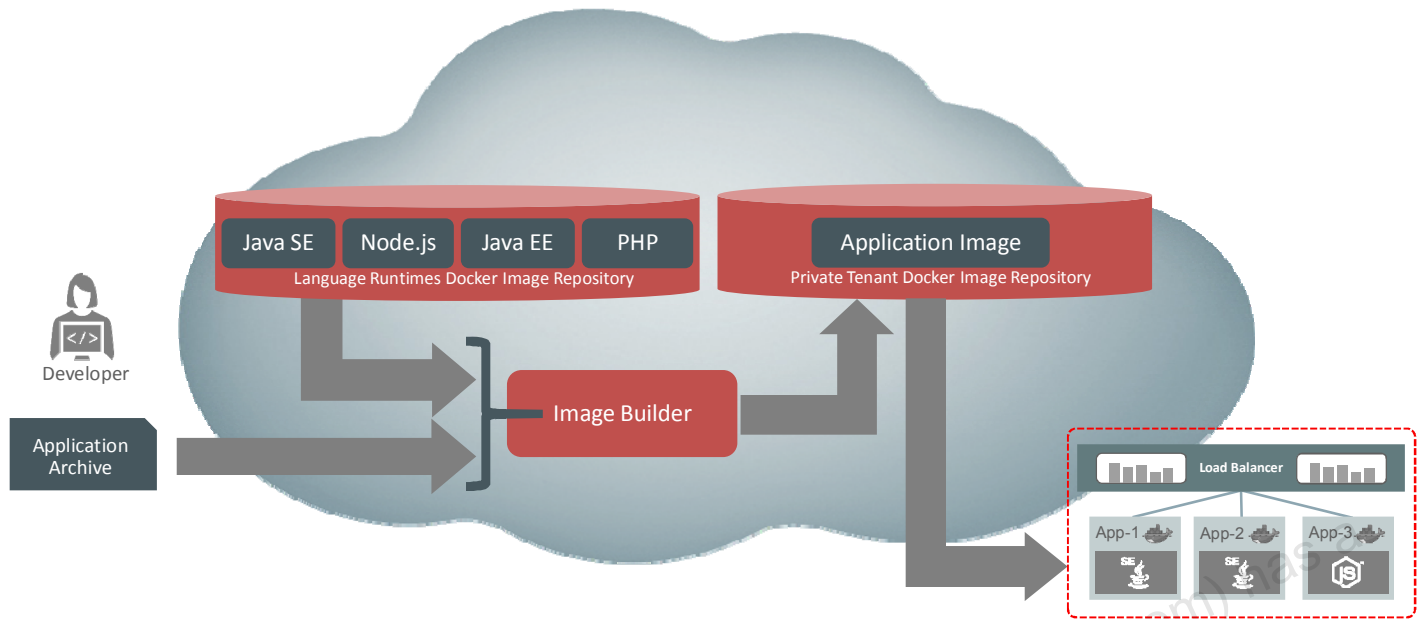
- All application binaries
- All required libraries
- Binaries of any container/embedded container
- Images files
- HTML files

Everything you'd need to run your application on a machine

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Application Deployment

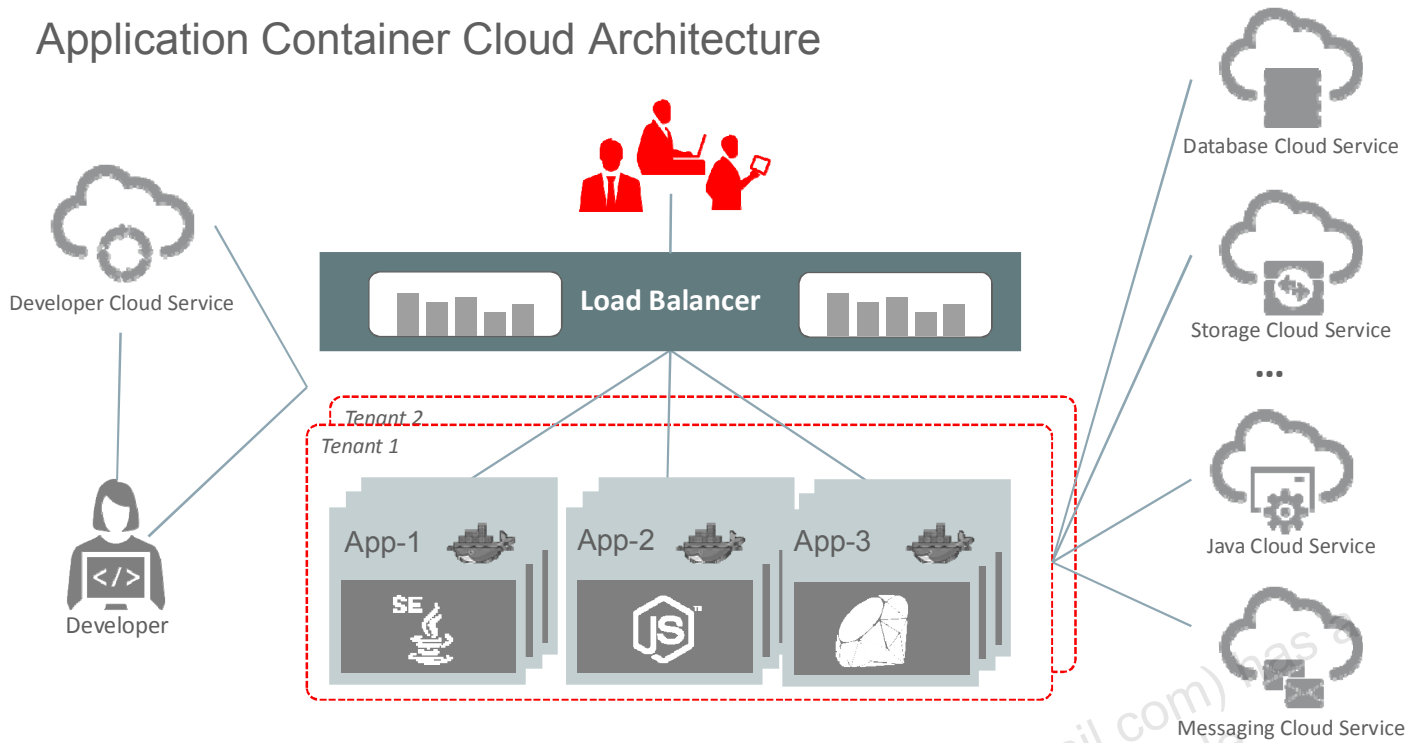


ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- After you deploy your application, what we do is we take the application—the zip file with all your binaries—and we take an off-the-shelf image for our runtimes with Java SE, Node, and so on.
- We get the right version of the runtime that you're looking for.
- We build a new Docker image with your application binaries and the base image, and publish it to your internal repository.
- We now have a repository with, say, your Application 1 and Application 2 images.
- When we want to run the application, we simply take the image and spin up some containers.
- As we split scale out, adding more containers, we use the same image to create containers.
- So it's a very straightforward process, and not dissimilar to, again, what you would be doing yourself on premise.

## Application Container Cloud Architecture



ORACLE

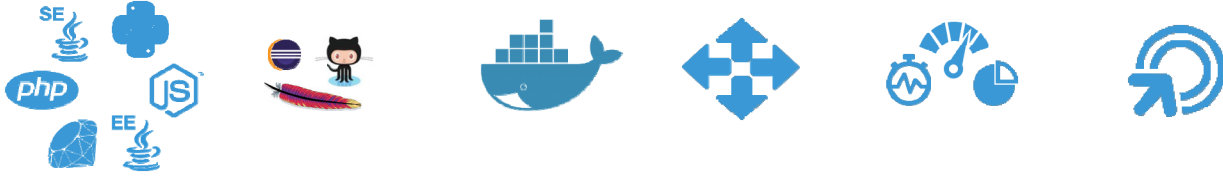
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

In front of your application, there's a shared load balancer. This load balancer distributes traffic to all your applications. In the diagram in the slide, you see that there is App 1, 2, and 3, and different technologies. And they're all scaled out with multiple instances. Incoming traffic is routed to each of the appropriate applications. Additionally, traffic is spread across the instances of that application. So this is fully automatic. There is nothing for you to do. Traffic goes through automatically as you scale, and so on. Everything is handled.

One of the key values of using cloud, is the ecosystem. And in the Oracle Public Cloud, there are several enterprise cloud services such as Database Cloud Service, Storage Cloud Service, Messaging Cloud Service, and Java Cloud Service for Java EE enterprise applications. All these are available services for you to consume. So if you're building cloud-native applications, what you're typically doing is building smaller applications than you would have in the past, where you bundled in everything and the kitchen sink into one big, monolithic application.

Rather, here you are building smaller applications that consume other services to achieve their goals. For example, if you need messaging, you get it from somewhere else. If you want storage, you go somewhere else. If there's an interesting cloud service available that you want to consume and incorporate into your application, some REST-based services, you plug it in and consume. This platform, therefore, is a rich ecosystem that your applications can leverage. That's the key feature – access to these services. You can build your applications either yourself locally and deploy them, or you can deploy them with Developer Cloud Service.

## Application Container Cloud Service Advantages



- Integrated **enterprise** ecosystem and services from IaaS to PaaS and SaaS
- Java SE Advanced – Completely **unique** and unavailable on any other cloud platform
- Developer Cloud Service – **Included** and *integrated*

**ORACLE**

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Introducing Java Cloud Service

### Your platform for running business applications in the cloud

Key Oracle Cloud component

- Self-service application platform with advanced cloud tools
- Saved time and cost with simplified provisioning
- Reduced down time: automated patching, backup, and recovery
- Increased data and processing capacity on-demand to scale for new business needs
- Optional enabling of Oracle Coherence for caching and data grid functions and Oracle Traffic Director for load balancing
- Pre-configured for Database and Developer Cloud Services for complete cloud application management



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Java Cloud Service: Three Options



**JAVA CLOUD SERVICE**

Full-Featured Service



**JAVA CLOUD SERVICE  
SAAS EXTENSION**

- Easy to enrich Oracle SaaS apps
- Tailor-made WebLogic Server for rapid extension deployment
- Ready marketplace with pre-built extensions, automated deployment



**JAVA CLOUD SERVICE  
VIRTUAL IMAGE**

- Simple, hosted WebLogic instance
- Oracle controlled, updated

**ORACLE**

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

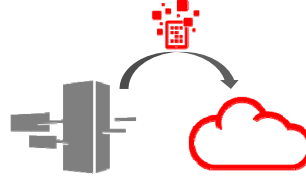
## Java Cloud Service Main Use Cases



Dev/Test in the Cloud



New App Development



Migrate Apps to Cloud



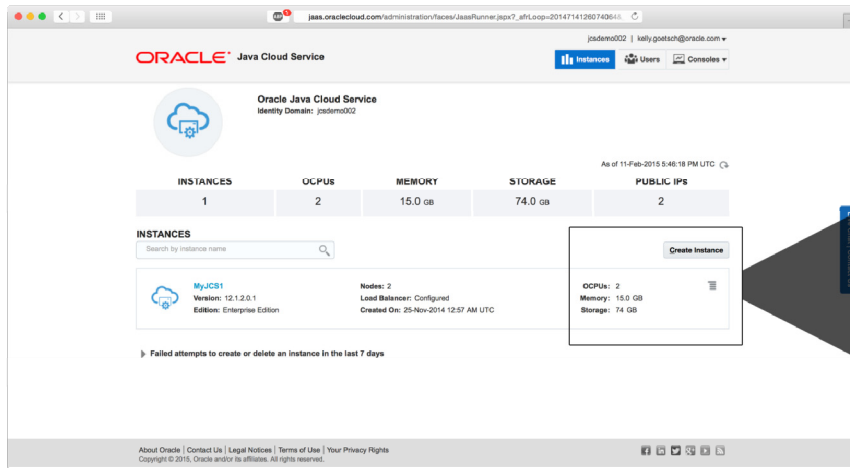
Strategic Outsourcing  
Recapture

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Java Cloud Service Feature: Provisioning

- Can pick shape/size – no complexity
- Can select from popular versions: 11g or 12c
- Can meet evolving technical and budgetary needs with popular Edition choices: Standard, Enterprise, and Suite

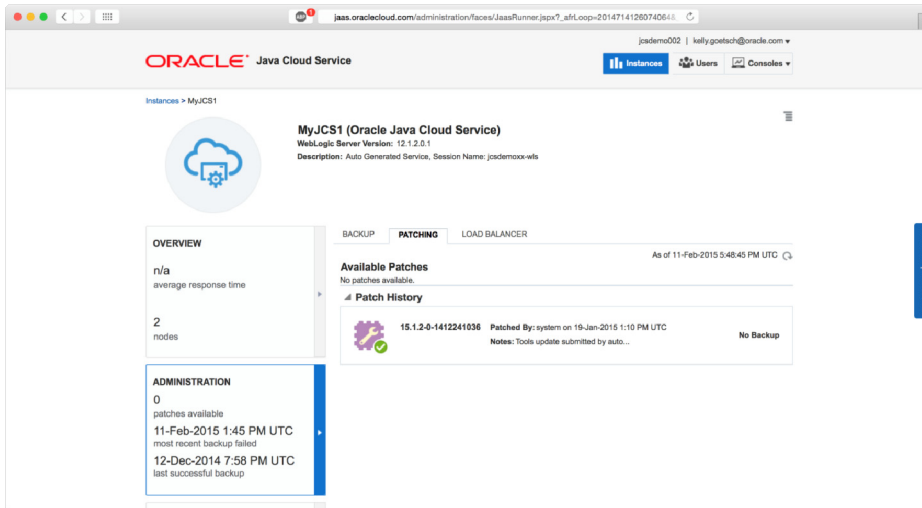


ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.



## Java Cloud Service Feature: Patching

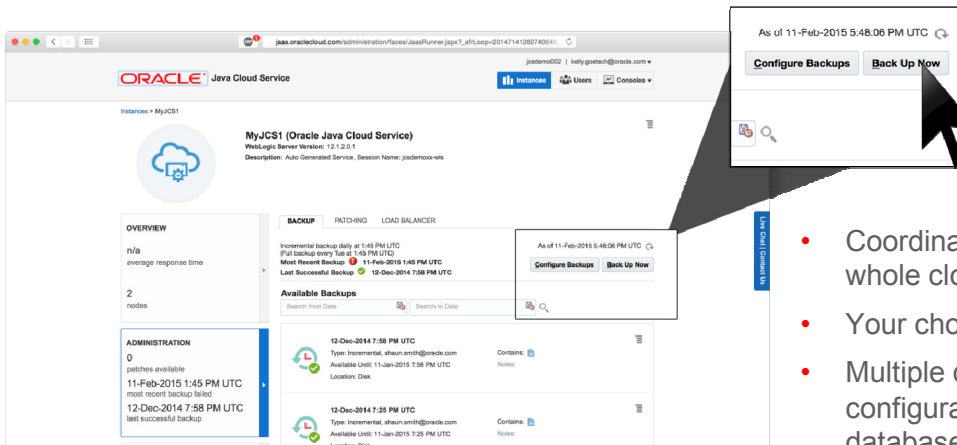


- Patching made simple (we handle the details)
- Controlled patch timing: on demand or scheduled
- Unified patching of JDK, WLS, and JRF/ADF included
- Rolling patching supported
- Patchset Updates (PSUs), Patchsets (PS), and Upgrades supported
- Full backup created before patching (don't mess with backups!)

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Java Cloud Service Feature: Backup / Restore



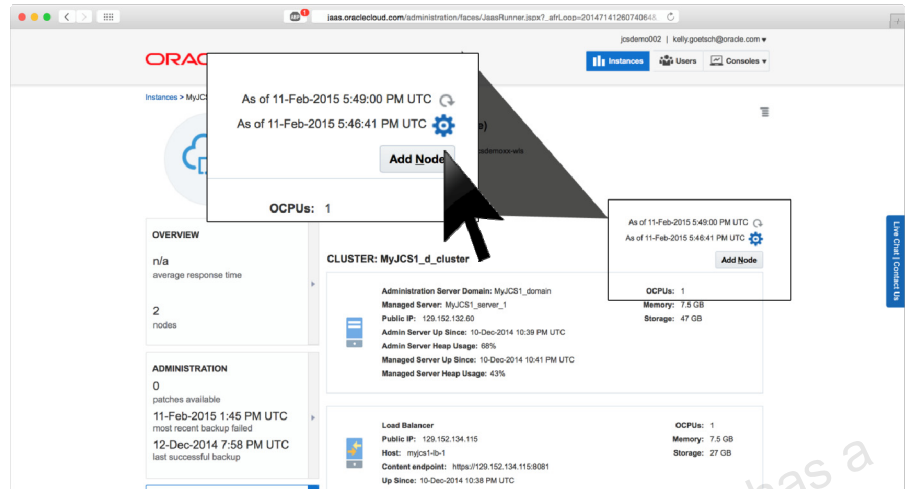
- Coordinated backups with database and whole cloud stack – holistic backups
- Your choice – scheduled or on demand
- Multiple depths supported: configuration/apps, logs, binaries, and database
- Configurable: 7 day backup on local disk, older backups pushed to storage service

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Java Cloud Service Feature: Scaling

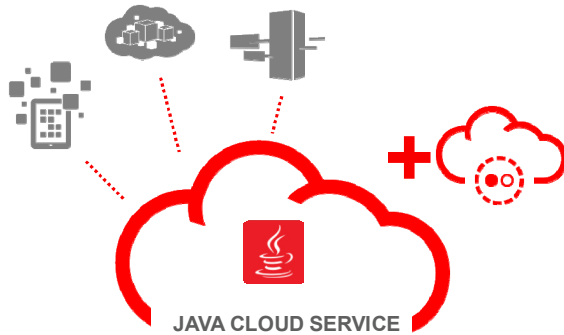
- Fully-automated, on-demand – Do it yourself without IT!
- Each managed server on separate virtual machine
- Zero down time during scaling – Keep customers happy.
- Scalable data capacity and processing on demand
- Rules to trigger scaling based on current workload



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## Oracle Coherence Option: Data Caching and Scaling



- Scaling applications' caching/data grid capacity in-memory to support growth
- Offload and protection of shared cloud services and databases
- Delivery of data to cloud apps in real time
- Transparency and high availability in the cloud's data grid tier

ORACLE

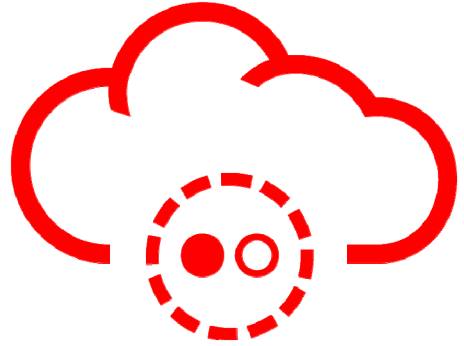
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Kevin Gómez González (kev.gomez97@hotmail.com) has a non-transferable license to use this Student Guide.

## Oracle Coherence Option: Your Cloud Data Grid

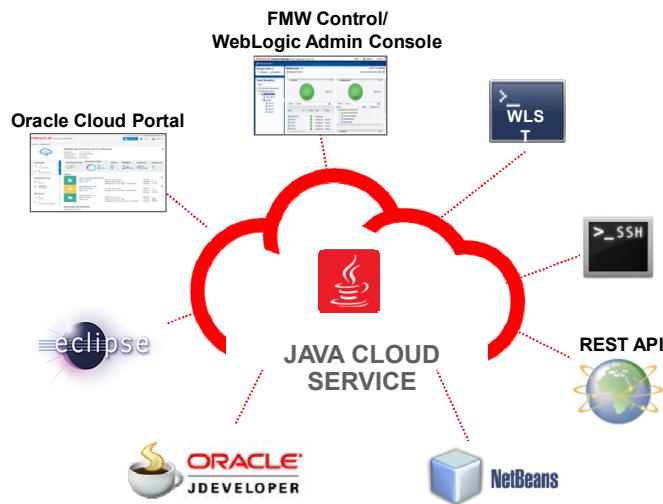
### Scalable, fault-tolerant cloud infrastructure

- Reliable in-memory, key-value store
- Dynamically scalable
- Scaling of processing with data
- Entries that can be
  - Reliably processed in-place
  - Queried
  - Aggregated
- Integration with Database and Developer Cloud Services

**ORACLE**

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

## How You Interact with Java Cloud Service



- New Cloud Portal
- WebLogic Admin Console
- Fusion Middleware Control
- Traffic Director Admin Console
- Public REST APIs
- Command Line Interface
- SSH or VNC to VM
- Standard IDEs

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- Oracle Cloud Portal is one of the main ways to administer Java Cloud Service.
- Both Fusion Middleware Control and WebLogic Administration Console run on the administration server instance of the WebLogic domain. They allow fine-grained control of the WebLogic environment.
- For those who prefer the command line, or using scripts, WebLogic Scripting Tool, or WLST, provides another option to manage Java Cloud services and the WebLogic domain. Administrators can use SSH consoles, using SSH tunnels to get into the otherwise hidden WebLogic managed servers and other components.
- There's a fully featured REST API. REST is a lightweight web services protocol based on HTTP commands. The full catalog of REST APIs is documented in the Java Cloud Service documentation.

## Demos

In this demo, you will learn to:

- Request and activate Oracle Cloud Trial account
- Deploy an application to the Application Container Cloud Service



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

The `MAVEN_OPTS` environment variable is used to specify JVM properties that can be used to supply extra options to Maven, such as the value `-Xms256m -Xmx512m`. Optionally, add this environment variable.

## Summary

In this lesson, you should have learned how to:

- Provide an overview of Oracle Application Container Cloud
- List the unique features of Oracle Application Container Cloud
- Describe how to build, zip, and deploy applications to the cloud
- Provide an overview of Java Cloud Service
- Describe the features of Java Cloud Service



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.