Oracle Enterprise Cloud Native Java
Modernizing Enterprise Java Applications using Weblogic Server
Applications and Helidon Microservices

Srinivas Pothukuchi – Master Principal Architect
Will Lyons – Senior Director Product Management

September 30, 2020
Safe Harbor

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

Statements in this presentation relating to Oracle’s future plans, expectations, beliefs, intentions and prospects are “forward-looking statements” and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle’s Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading “Risk Factors.” These filings are available on the SEC’s website or on Oracle’s website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.
Oracle WebLogic Server and Coherence Have Powered The Most Demanding Enterprise Applications

- Scale and Performance
- Robustness and Availability
- Operational Simplicity and Efficiency
- Proven, Secure, Integrated
The Enterprise Application World Is Changing

Cloud Native Container Deployments That Run Anywhere

Open Source DevOps Tool Chains

Microservices and “Serverless” Applications

“Polyglot” Applications That Use Multiple Languages

Copyright © 2020 Oracle and/or its affiliates.
How Do You Combine The Best Of Both Worlds?

- Modernize enterprise Java applications
- Adopt new operational models
- Run your applications anywhere
- Move forward at your own pace
Customer Question: How Can Oracle Help Us....

- Migrate traditional applications to containers, Kubernetes
- Evolve applications and adopt microservices
- Manage this changing environment
Oracle Enterprise Cloud Native Java

- Build and deploy containerized Java EE/Jakarta EE applications that run anywhere
- Build and deploy microservices alongside traditional applications
- Easily run and manage complex applications combining existing and new
- Select an approach that make sense for you

Oracle Cloud and Other Clouds

Migrate

Physical & VMs On Premise

Kubernetes on Premise
Oracle Enterprise Cloud Native Java

Supported on Kubernetes and the Cloud
Integration with Oracle Cloud, Database, FMW...
New Releases and Innovation
Current and Future Application Needs
Oracle Enterprise Cloud Native Java Topics

- Docker and Kubernetes certification
- WebLogic Kubernetes Toolkit
- WebLogic Server on OCI/OKE
- Coherence Operator
- Coherence Community Edition
- Helidon
- GraalVM
- Verrazzano

Supported on Kubernetes and the Cloud
Integration with Oracle Cloud, Database, FMW...
New Releases and Innovation
Current and Future Application Needs
WebLogic Server on Kubernetes

- Kubernetes certification
  - OKE on Oracle Cloud
  - Oracle Linux Cloud Native Environment
  - Oracle Private Cloud Appliance
  - Other Kubernetes (e.g. OpenShift)
- WebLogic Kubernetes ToolKit
  - Integrated tools available in open source...
  - Migration: WebLogic Deploy Tooling
  - Image Creation: WebLogic Image Tool
  - Management: Operator
  - Monitoring: Exporter for Prometheus
  - Logging: Exporter for Elastic Stack
WebLogic Kubernetes ToolKit Example - Migrate to Kubernetes

- Introspect Domain
- Create Image
- Publish Image
- Deploy Domain

WebLogic Domain
- Physical & VMs

WebLogic Image
- Properties
- Binaries

Repository
- Application Archive
- WebLogic Image

Domain Model

Available Today
WebLogic Kubernetes ToolKit Example – Automate Updates

Repository

WebLogic Image

Update Domain

New Properties
Patches
Config Changes

Update Image

Publish New Image

Available Today

Domain CR

WebLogic Operator

Kubernetes

Applicaton Updates

Copyrigh © 2020 Oracle and/or its affiliates.
WebLogic Server for Oracle Cloud Infrastructure

- Select listing to provision WebLogic domain on Oracle Cloud Infrastructure compute instances
- Metered usage – pay for what you use
  - Priced per OCPU/Hour
  - Consume Universal Credits
  - Create, destroy, start, stop, scale
- Bring Your Own License (BYOL)
- Develop and deploy in the cloud
- Migrate existing applications to cloud
- Integrate with other cloud services
Oracle WebLogic Server for OKE

- Provision WebLogic on OCI/OKE
- Built in CI/CD solution using Jenkins
- Configured to run WebLogic on specific nodes
- Public Load Balancer for WLS Cluster
- Private Load Balancer for Admin Consoles
- Scale K8s cluster through OCI Node Pool
- Scale WebLogic domain through WLS Operator
- Backups for every domain configuration change
- Domain image scan for vulnerabilities
- Leverages WebLogic Kubernetes ToolKit

WebLogic For OKE Components

- Admin host
- Bastion host
- Public LB
- Private LB
- File Storage
- Oracle Cloud Infrastructure Container Engine for Kubernetes
- OCI Control Plane
- OKE Cluster
- Non WebLogic Node Pool
- WebLogic Node Pool
- Oracle Cloud Infrastructure Registry (OCIR)
CERN is the world's premier research organization for nuclear physics.

- Migrated their large WebLogic workloads comprising more than 250 clusters, 500 JVMs and 100 VMs, to Kubernetes (including production customers)
- Gained improvement in deployment agility with shorter downtimes and better manageability
- Achieved faster time-to-market
- Impressed by the results, CERN is currently planning a hybrid environment with some Kubernetes clusters running on OCI and some on premises

We were able to deploy the same WebLogic cluster in 2 minutes, what previously took us at least half a day.

Antonio Nappi, DevOps Engineer
Coherence on Kubernetes

• “Coherence-only” configurations
• Leverages REST Management, metrics
  Included in Coherence 12.2.1.4
  Simplified Coherence management
• Consistent approach with WLS Operator
  Helm chart for installing Operator
  Prometheus, Grafana, ELK
  Manage scaling, patching, versioning
Coherence Community Edition

- Open-source edition of Coherence
- Hosted on GitHub
- Artifacts in Maven Central
- Docker images to Docker Hub
- Subset of Coherence EE features
- Everything necessary to write microservices applications
- New option for new projects
- Part of a platform for cloud-native microservices applications
WebLogic and Coherence Supported on GraalVM

- GraalVM – universal virtual machine for running polyglot applications
  - JavaScript, Python, Ruby, R, Java, Scala...
- At runtime, translates Java bytecodes into machine code
- You can use GraalVM as your JDK – Java compliant
- Includes the same commands, flags, and options
- High performance – 5%-10% performance benefit
Coherence Support for GraalVM

Server-side polyglot language support
Certify running Coherence on Graal JVM OOTB, including docker image
Grid-side code in your favorite language – JavaScript, Python, R, Ruby …
Embedding Coherence into non-Java applications
Connect to or host Coherence in non-Java processes e.g. Node.js/Express
Coherence-Based Microservices at Union Pacific Railroad

Union Pacific Railroad is North America's premier railroad franchise, covering 23 states across the western two-thirds of the United States

✓ Replacing mainframe-based logistics system with microservices
✓ Coherence-based data management at massive scale:
  ✓ 5,600 microservices in production; 20,000 JVMs
  ✓ 1.3B API calls per day; producing 300M events per day
  ✓ 9TB data managed
  ✓ 10,000 TPS against Coherence API
✓ Multi-site architecture with data replication

Stateless systems are extremely data-hungry, and state management is one of the biggest challenges we faced. This is where the Oracle Coherence product played a huge role. Coherence is our state management platform for this logistics system today; that's where all our data is stored.

Arun Giri
Associate Vice President and Distinguished Technologist
Java Microservices with Helidon

- Java libraries for developing microservices
- Standards-based, open source, cloud native
- Helidon SE
  - Functional style, Reactive, Transparent
- Helidon MP (MicroProfile 3.2)
  - MicroProfile, Declarative, Dependency Injection
- GraalVM Native Image
- Integrates with other frameworks
- WebLogic, Coherence integration
- Support included with WebLogic
GraalVM Native Image

- Compiles your programs ahead-of-time into a native executable
  - Improves startup time
  - Reduces memory footprint
- Closed world assumption
Helidon Has Industry Leading Start-Up Performance
Turbocharged with GraalVM

Startup Time (secs)
Lower is better

Java SE JVM
- Helidon SE
- Quarkus
- DropWizard
- Helidon MP
- SpringBoot

GraalVM Native
- Helidon SE

26 Milliseconds

Available Today

Copyright © 2020 Oracle and/or its affiliates.
Microservice with NoSql data store

- Microservices and Data store deploy separately
- Scaling and Failover is separate and disjoint
- Monitoring is difficult

Helidon with Coherence Cache and GraalVM

- Microservices and Coherence share GraalVM runtime
- Scaling and HA/Failover is combined and simple
- Metrics/Monitoring/Trace is end-to-end from http to cache objects
Coherence and Helidon MP Microservices

- Bootstrap Coherence via CDI within Helidon MP apps
  - Ensures that the REST and gRPC services can access their data as soon as they are up
  - Helidon controls the main method, Coherence is just a library
- CDI support
  - Inject Coherence resources e.g. NamedMap into Helidon services
  - Inject CDI-managed objects into Coherence (event interceptors, cache stores, etc.)
  - Use CDI observers to handle Coherence server- and client-side events
- Metrics
  - Coherence metrics available via standard Helidon MP /metrics endpoint
- Configuration
  - Configure Coherence using MP Config
  - Use Coherence as a mutable, observable MP Config Source
- Tracing
  - Coherence tracing spans automatically included into Helidon traces

Copyright © 2020 Oracle and/or its affiliates.
# WebLogic, Coherence, Helidon Versions

<table>
<thead>
<tr>
<th>WebLogic Versions</th>
<th>Standards</th>
<th>WebLogic 10.3.6</th>
<th>Java EE 5, Java SE 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebLogic 12.1.3</td>
<td>Java EE 6, Java SE 7/8</td>
<td>Coherence 3.71</td>
<td>Java SE 7</td>
</tr>
<tr>
<td>WebLogic 12.2.1.4</td>
<td>Java EE 7, Java SE 8</td>
<td>Coherence 12.1.3</td>
<td>Java SE 7/8</td>
</tr>
<tr>
<td>WebLogic 14.1.1</td>
<td>Java EE 8/Jakarta EE 8 Java SE 8/11</td>
<td>Coherence 12.2.1.4</td>
<td>Java SE 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coherence 14.1.1</td>
<td>Java SE 8/11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coherence CE 20.0.6</td>
<td>Java SE 8+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coherence Versions</th>
<th>Standards</th>
<th>Coherence 3.71</th>
<th>Java SE 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coherence 12.2.1.4</td>
<td>Java SE 8</td>
<td>Coherence 12.2.1.4</td>
<td>Java SE 8</td>
</tr>
<tr>
<td>Coherence 14.1.1</td>
<td>Java SE 8/11</td>
<td>Coherence 14.1.1</td>
<td>Java SE 8/11</td>
</tr>
<tr>
<td>Coherence CE 20.0.6</td>
<td>Java SE 8+</td>
<td>Coherence CE 20.0.6</td>
<td>Java SE 8+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Helidon Versions</th>
<th>Standards</th>
<th>Helidon 1.4</th>
<th>MicroProfile 3.2, Java SE 8/11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helidon 2.0</td>
<td>MicroProfile 3.2, Java SE 11</td>
<td>Helidon 2.0</td>
<td>MicroProfile 3.2, Java SE 11</td>
</tr>
</tbody>
</table>

[https://www.oracle.com/java/weblogic/](https://www.oracle.com/java/weblogic/)
[https://blogs.oracle.com/weblogicserver/](https://blogs.oracle.com/weblogicserver/)

[https://www.oracle.com/middleware/coherence/](https://www.oracle.com/middleware/coherence/)
[https://blogs.oracle.com/oraclecoherence/](https://blogs.oracle.com/oraclecoherence/)
[https://github.com/oracle/coherence](https://github.com/oracle/coherence)
[https://coherence.community/](https://coherence.community/)

[helidon.io](https://helidon.io)
[https://helidon.io/docs](https://helidon.io/docs)
[https://github.com/oracle/helidon](https://github.com/oracle/helidon)
[https://medium.com/helidon](https://medium.com/helidon)
Oracle Enterprise Cloud Native Java

- Build and deploy containerized Java EE/Jakarta EE applications that run anywhere
- Build and deploy microservices alongside traditional applications
- Easily run and manage complex applications combining existing and new
- Select an approach that make sense for you
The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle’s products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle’s future plans, expectations, beliefs, intentions and prospects are “forward-looking statements” and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle’s Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading “Risk Factors.” These filings are available on the SEC’s website or on Oracle’s website at http://www.oracle.com/investor. All information in this presentation is current as of September 2020 and Oracle undertakes no duty to update any statement in light of new information or future events.
Thank You