Container-native Applikationsentwicklung in der Cloud

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technology and Architecture
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.
Containers on Oracle Cloud Infrastructure

- Evolution of Development and Deployment
- Containers with Microservices and Orchestration
- Docker and Kubernetes
- Oracle Cloud Infrastructure Application Platform
- Demo
- Summary
## Evolution of Development and Deployment

<table>
<thead>
<tr>
<th>Development Process</th>
<th>Application Architecture</th>
<th>Deployment and Packaging</th>
<th>Application Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>~ 1980</td>
<td>Waterfall</td>
<td>Monolithic</td>
<td>Physical Server</td>
</tr>
<tr>
<td>~ 1990</td>
<td>Agile</td>
<td>N-Tier</td>
<td>Virtual Servers</td>
</tr>
<tr>
<td>~ 2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~ 2010</td>
<td>DevOps</td>
<td>Microservices</td>
<td>Containers</td>
</tr>
<tr>
<td>Now</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.
Entwicklungsmuster für Applikationen

Container Native

Serverless

Cloud Native PaaS

Enterprise Java/DB

Visual/Low Code
Docker and Kubernetes

- Popular, easy to use tooling targeting developer productivity
- De facto standard container runtime and image format
- Used for developer on-boarding and 1st Generation application management (Oracle ACCS, Compose, Swarm)

- Production grade container management targeting DevOps and operations, with widespread adoption
- Complex but powerful toolset supporting cloud scale applications
- Rich operations feature set, autoscaling, rolling upgrades, stateful apps and more.
Docker & Kubernetes leading the industry

Containers (Docker)

- 60% of enterprise companies (500+ hosts) use **Docker**
- 15% of all the hosts at these companies run Docker

Orchestration (Kubernetes)

- 40% of Docker users also use orchestrators
- 80% of these orchestration users prefer **Kubernetes**
The Containerization Journey

Focus: Applications: Automation: Community:
Developer adoption Dev/Test apps Simple orchestration Individual developers
DevOps deployment Production apps Advanced orchestration Teams & lines of business
Broad integration Business apps Self management Enterprises

Phase I Developer Focus
Phase II DevOps Focus
Phase III Business Focus
Microservices Architektur

Monolithische Architektur

- User Interface
- Business Layer
- Data Interface

Single Deployment Entity

Unabhängige Service-Kommunikation mit synchr. API’s oder asynchr. Events

Microservice Architektur

- Microservice UI
- Microservice
- Microservice
- Microservice

Single Deployment Entity
Helidon - Java Frameworks for writing microservices

• **Microframeworks**
  Simple, fun, intentionally small feature set. Examples are Spark, Javalin, Micronaut, etc.

• **MicroProfile**
  Friendly to Java EE developers but a bit heavier. Some of them are build on top of fully featured Java EE application servers. Examples are Thorntail (was Wildfly Swarm), OpenLiberty, Payara.

• **Full Stack**
  Fuller features set such as Spring Boot.
Helidon in two flavours: Microframeworks and MicroProfile

“It’s up to a developer to choose what to use in their applications.”

- **Helidon SE** — simple, functional, lightweight microframework developed in a modern reactive way. There is no injection “magic”. No special runtime required. JDK is used as runtime.

- **Helidon MP** — Eclipse Microprofile implementation providing development experience familiar to Java EE/Jakarta EE developers.
Helidon Architecture

- Helidon SE components are marked **green**. These are **Config**, **Security** and **RxServer**.

- The Java EE/Jakarta EE components we use are marked **grey**. These are **JSON-P**, **JAX-RS/Jersey** and **CDI**. These components are required for MicroProfile implementation.

- Helidon MP is a thin layer on top of Helidon SE components.

- Oracle Cloud services components are marked **red** and used in both Helidon SE and Helidon MP.
Docker

- Lightweight
  - No HW Specification
- Portable
- Layering for Optimization
  - UID’s for Each Layer
- Lives in a Registry
  - wcr.io
- Easy to Deploy
  - Consistent Environment
  - All Dependencies Included

Vendor Provided Cached Reused

Application

WebTier

Binaries

Libraries
Kubernetes

- Orchestrator
  - Manages Lifecycle
- Eventual Consistency
- Declarative vs. Imperative
  - Traceable, Reproducible Environment Changes
- Statelessness and Persistence
- Scales On Demand
  - Smaller containers makes for more efficient scaling operations
- Easy to Connect to Services
- Resiliency
  - Makes sure your App is always running
Kubernetes Konstrukte

• Wie die Kubernetes Konstrukte untereinander verbunden sind
  – Deployment
  – Container
  – Service
  – Ingress
  – Secret
  – Node
  – Namespace
Oracle & Open Source und Container-Community

• Oracle’s participation in open source community
  – Active Participation – Cloud Native Compute Foundation and Kubernetes
  – No forked code – straight from the source
  – Continue precedence of Java, MySQL, Linux

• Lead by example
  – Oracle software on Docker Store
  – Kubernetes engineering in CNCF

• Innovate in open source
  – Utilities like K8S installer, smith, railcar, crashcart

• Sponsor & contribute to key conferences
  – DockerCon, Kubecon, CoreOS Fest, others
Ziel ist eine ganzheitliche Container Native Suite
Build, Deploy, Operate – Offene, Standards-basierte, Cloud neutrale AppDev Plattform
Oracle Cloud Infrastructure Application Platform
Oracle Cloud Infrastructure Application Platform

Cloud Native Applications
Micro Services, Big Data, AI/ML

Container Services
Docker, Kubernetes, Serverless Functions

Cloud Infrastructure
25 GB Network, Bare Metal, VM, GPU, NVMe SSD, Block Storage, File Storage, Object Storage

Monitoring and Management, Streaming and Pipelines, Developer Tools, DevOps Tools and Services
Oracle Cloud Infrastructure mit direkter Virtualisierung

Oft verwendet

Oracle Cloud Infrastructure

Separates Network and Subscriber Environment

To or from other Tenants
Oracle Strategie für Container-Based-Infrastruktur

• Deliver a container based capabilities that are complete, integrated and open
  – Orchestration/Scheduling, CI/CD, Management/Operations, Analytics/Introspection
  – With application development platform for serverless and microservices

• Based on community driven open source technology
  – Investing in open source communities and foundations (Kubernetes, Docker, CNCF) via engineering resources, code contributions & sponsorship

• Differentiated on quality of service and operational excellence
  – Full, transparent management
  – Deployed to Oracle Cloud Infrastructure
  – Enterprise grade security, HA and governance
Oracle’s Commitment to Containers and Open Source

- Oracle’s participation in open source community
  - Active Participation – Cloud Native Compute Foundation and Kubernetes (see https://www.cncf.io/about/members/)
  - No forked code – straight from the source
  - Continue precedence of Java, MySQL, Linux
- Lead by example
  - Oracle software on Docker Store
  - Kubernetes engineering in CNCF
  - Java SE/EE open sourcing; transparent processes
- Innovate in open source
  - Utilities like K8S terraform install, smith, railcar, crashcart
- Sponsor & contribute to key conferences
  - DockerCon, Kubecon, CoreOS Fest, others
Oracle Cloud Infrastructure and Kubernetes
Roll Your Own, Pre-Built Installer, Managed Service, Tools for Cloud@Customer
DIY - Terraform Kubernetes Installer for OCI

• Open Source OCI Kubernetes installer, based on Terraform
  – Oracle developed for Kubernetes on OCI
  – Available on Github
  – https://github.com/oracle/terraform-kubernetes-installer

• Key Highlights
  – Highly available Kubernetes cluster configured in your OCI tenant and compartment
  – Creates VCN, subnets, LBs and instances for control plane
  – Specify number and shape of nodes for your cluster
  – Scale your cluster as needed

Kubernetes Challenges

- Managing, maintaining, upgrading Kubernetes Control Plane
  - API Server, etcd, scheduler etc.
- Managing, maintaining, upgrading Kubernetes Data Plane
  - In place upgrades, deploy parallel cluster etc.
- Figuring out container networking & storage
  - Overlays, persistent storage etc. - it should just work
- Managing Teams
  - How do I manage & control team access to my clusters?
- CI/CD Integration
  - How do I drive automated testing and conditional release into my application lifecycle?
Container Engine for Kubernetes - OKE

OKE
- Managed Kubernetes container service to deploy and run your own container based apps
- Tooling to create, scale, manage & control your own standard Kubernetes clusters instantly

Use Case
- Too complex, costly and time consuming to build & maintain Kubernetes environments
- Too hard to integrate Kubernetes with a registry and build process for container lifecycle management
- Too difficult to manage and control team access to production clusters

Benefit
- Enables developers to get started and deploy containers quickly. Gives DevOps teams visibility and control for Kubernetes management.
- Combines production grade container orchestration of open Kubernetes, with control, security, IAM, and high predictable performance of Oracle’s next generation cloud infrastructure
Oracle Cloud Infrastructure Registry - OCIR

**OCIR**
- A high availability Docker v2 container registry service
- Stores Docker Images in Private Repositories
- Runs as a fully managed service on Oracle Cloud Infrastructure

**Use Case**
- Without a registry it is hard for Development teams to maintain a consistent set of Docker images for their containerized applications
- Without a managed registry it is hard to enforce access rights and security policies for images
- It is too hard to find the right images and have them available in the region of deployment

**Benefit**
- Full integration with Container Engine for Kubernetes (OKE)
- Registries are private by default, but can be made public by an admin
- Co-located regionally with Container Engine for low latency Docker image deploys
- Leverages OCI for high performance, low latency and high availability
Working with OKE and OCIR on OCI

- OCI Registry
  - Encryption for Data in Transit (SSL) and at Rest

- OCI Container Engine for Kubernetes
  - Cluster Management
  - HA - 3 Masters/etcd across 3 ADs
  - OKE Dashboard in OCI Console

- Customer’s OCI Account/Tenancy
  - VM based Clusters and Nodes
  - Bare Metal Clusters and Nodes

Oracle Cloud Infrastructure

Oracle Managed

Customer Managed

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.
Detail: OCI Container Engine for Kubernetes and Registry
An Open, Fully-Managed Kubernetes Platform & Private Registry

- **Container Native**: Standard Upstream Kubernetes; Fully Managed Lifecycle; Integrated Registry
- **Developer Friendly**: Simple, Streamlined User Interface; REST API; Helm, and DNS Built-in
- **Enterprise Ready**: Oracle Cloud Infrastructure Performance; Highly Available; Secure with OCI Access Controls

Any CI/CD - ie Jenkins, Oracle Pipelines, etc.

Exposed Kubernetes Service

Build
Test
Test
Test
Push

OCI Registry

OCI Container Engine for Kubernetes

VCN

AD 1

AD 2

AD 3

K8S Cluster

Node Pool

Node Pool

Pods
# Oracle Container Engine & Registry

## Container Native
- **Standard Docker & Kubernetes**
  - Deploy standard & open upstream Docker and Kubernetes versions for compatibility across environments
- **Registry Integration**
  - Full Docker v2 compatible private registry to store and manage images
- **Container Engine**
  - Deploy and operate containers and clusters
- **Full integration to cloud networking and storage**
  - Leverage the enterprise class networking, load balancing and persistent storage of Oracle Cloud Infrastructure

## Developer Friendly
- **Streamlined Workflow**
  - Use your favorite CI to push containers to the registry, then Kubernetes to deploy to clusters and manage operations
- **Full REST API**
  - Automate the workflow, create and scale clusters through full REST API
- **Built In Cluster Add-Ons**
  - Kubernetes Dashboard, DNS & Helm
- **Open Standards**
  - Docker Based Runtime
  - Worker Node SSH Access
  - Standard Kubernetes

## Enterprise Ready
- **Simplified Cluster Operations**
  - Use the fully managed, highly available registry, master nodes and control plane
- **Full Bare Metal Performance and Highly Available IaaS**
  - Combine Kubernetes with bare metal shapes for raw performance
  - Deploy Kubernetes clusters across multiple Availability Domains for resilient applications
- **Team Based Access Controls**
  - Control team access and permissions to clusters
- **Autonomous Clusters**
  - Maintain cluster size and performance in face of node failures and load fluctuations
Oracle Container Native Microservices Architektur

- Order Service
- Shopping Cart Service
- Catalog Service
- Foo1 Service
- Foo2 Service

- API Registry
- Event Management
- Open Service Broker

- Managed Kubernetes

- Automatic service registration and discovery
- Inter-service communication security
- Automatic diagnostics data feeds to ops tools
- Dynamic load balancing and health checks
- Resiliency with circuit breakers, timeouts and retries
- Dynamic request routing for canary releases, A/B tests, gradual rollouts
Container Native Application Platform Responsibilities

<table>
<thead>
<tr>
<th>Container CI/CD</th>
<th>Container Registry “Releases”</th>
<th>Container Engine – Managed Kubernetes “Clusters”</th>
<th>Customer’s OCI Account/Tenancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>wercker</td>
<td>Encryption for Data in Transit (SSL) and at Rest</td>
<td>Cluster Management</td>
<td>VM Based Clusters and Nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HA Masters and etcd</td>
<td>Bare Metal Clusters and Nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K8S Dashboard</td>
<td></td>
</tr>
</tbody>
</table>

Oracle Cloud Infrastructure

Oracle Managed

Customer Managed
Oracle Container Native Serverless - Functions as Service
Eine funktionsbasierte Plattform

- Polyglotte Funktionsplattform mit Java Optimierungen
- Open Source
- Container Native mit lokaler Entwicklerunterstützung – On Premise & Cloud
- Cloud- und Scheduler-Neutral
Die bekannten Mechanismen mit DevOps benutzen
Build, deploy and publish services and APIs with your preferred tool chain

CI-CD/DevOps Pipelines
- Jenkins
- wercker
- Atlassian
- Hudson
- Developer Cloud

Command Line tools or curl
- `psm setup`
- `psm list services`
- `psm push app` ...
- `curl -i -X GET -H "Authorization:joe@example.com:joePassword" -H ...`

Automation Tools
- puppet
- chef
- ansible
- Spinnaker
- Hashicorp

Oracle Container Native

Service API Catalog

REST API’s

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.
Bestenfalls mit einer führenden CI/CD & Container-Lifecycle-Lösung

Container Pipelines (Wercker) - easy to assemble and automated builds to registries and production-grade clusters

Git → Build → Push to Registry → Registry → Deploy to Orchestration Scheduler

Test → Push → Deploy

OSS CLI

Pipeline/Build Console

Workflow Automation

Chatbot Integration with Slack (Walterbot)

Oracle, Pivotal, Amazon, Google ...
Oracle Cloud-Native-Application-Entwicklungs-Plattform
Integrierte Suite mit Container-Native-Fähigkeiten

Container Pipelines
A market leading solution for application lifecycle management with a Docker centric product view

Container Engine
Fully managed container service based on Kubernetes running on Oracle Cloud Infrastructure Bare Metal

Container Microservices
A collection of services, frameworks and libraries for the modern cloud developer; based on Cloud Native Compute Foundation – Istio/Envoy

Container Functions
Open source, cloud neutral, community driven functions as a Service for any language, best of class for Java

Container Diagnostics
Unparalleled real-time observability and diagnostics for large scale distributed Java systems

Build --- Deploy --- Operate

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.
Zusammenfassung – Oracle Container Native

- **Open source, Cloud-neutral und von der Community getrieben**
  - Nicht proprietär, Cloud-neutral, unterschiedliche Service-Qualität

- **Leistungsfähige und evolutionäre Weiterführung von IaaS**
  - Oracle-Cloud-Infrastruktur von der Basis konzipiert und aufgebaut für Performance, Sicherheit & Hochverfügbarkeit

- **Optimiert für Java-Workloads**
  - Tiefgreifende Diagnosemerkmale enthalten und eingebautes Performance-Tuning

- **Container-Native-Entwickler-Umgebung**
  - Nahtlos vom lokalen Laptop zu CI/CD, zum vollständig integriertem Site-Reliability-Engineering (SRE) in der Cloud
Danke!

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technology and Architecture