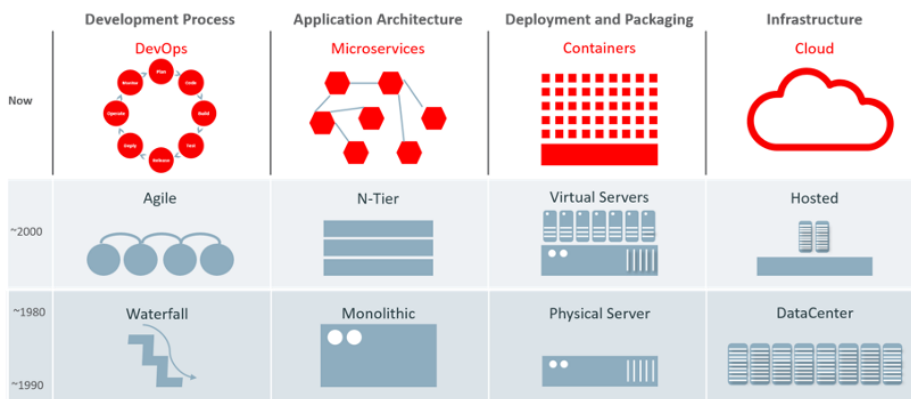


# Oracle Cloud Native Environment

Enterprises are looking for technologies that can help them achieve faster time to market and keep ahead of the competition. Moving to cloud native, microservices-based applications can provide the agility and increased productivity needed. With Oracle Cloud Native Environment, customers can develop these applications using a curated set of open source components that are tested and supported.

## Evolution of application development and deployment

The decomposition of applications into microservices, running on container infrastructure has occurred over the last several years. At the same time, developers and IT operations are more effectively collaborating, using DevOps methodologies, to improve alignment. This has allowed cloud native computing to evolve rapidly and has proven to enable organizations to build modern cloud applications that are resilient, manageable, and scalable.



## Oracle Cloud Native Environment

Oracle Cloud Native Environment is a fully integrated suite for the development and management of cloud native applications.

Oracle Cloud Native Environment is part of [Oracle's extensive portfolio](#) of standards-based open software technologies for cloud native application development, deployment, and lifecycle management in the cloud and on-premises.

### Related products

- [Oracle Linux](#)
- [Oracle Verrazano Enterprise Container Platform](#)
- [Oracle VM VirtualBox](#)
- [Oracle Private Cloud Appliance](#)

### Related support services

- [Oracle Linux Premier Support](#)
- [Oracle Premier Support for Systems](#)

### Related cloud services

- [Oracle Container Engine for Kubernetes](#)

## Cloud Native Computing Foundation

Critical to this approach is the existence of an industry organization, the Cloud Native Computing Foundation or CNCF. The CNCF promulgates guidelines and defines certifications for cloud-native microservices software. Oracle is a platinum member of CNCF as well as a platinum member of the Linux Foundation.

## Developing cloud native applications

Building and deploying cloud native applications can be challenging due to the changing technology landscape. One option is to build your own cloud native environment from open source software. This requires dealing with the complexity of picking the right software and getting it all to work together, without any vendor support. Another approach is to use a stack or distribution from a software vendor. This option offers support but that could mean lock-in with that vendor.

Oracle offers a better alternative—one that can give you the best of both worlds by delivering software that supports the open standards promoted by the Cloud Native Computing Foundation (CNCF), the Open Container Initiative, and other cloud native projects, with enterprise-grade support and without lock-in.

## Oracle Cloud Native Environment

With Oracle Cloud Native Environment, Oracle provides the components required by customers to develop microservices-based applications that can be deployed in environments that support open standards and specifications.

Oracle closely tracks CNCF projects as well as the Open Container Initiative and contributes to and abides by the standards defined by both. For example, Oracle delivers the tools to create and provision Open Container Initiative compliant containers. Oracle Cloud Native Environment incorporates the Kubernetes project and is released under the CNCF Kubernetes Certified Conformance program.

## Container orchestration and management

Oracle Cloud Native Environment simplifies the configuration and setup of Kubernetes with support for backup and recovery. This solution is developed by Oracle for Oracle Linux and is installed and managed by our open source platform components to provide a comprehensive container and orchestration environment for the delivery of microservices and next generation application development.

## Container runtimes

Oracle Cloud Native Environment leverages the native Kubernetes container runtime interface, CRI-O. CRI-O allows you to run containers directly from Kubernetes without any unnecessary code or tooling. As long as the container is compliant with the Open Containers Initiative specification, CRI-O can run it, cutting out extraneous tooling and allowing containers to do what they do best: fuel your next-generation cloud native applications.

## Key benefits

- Accelerates time-to-value and delivers agility through modularity and developer productivity
- Modernizes applications and lowers costs by fully exploiting the economic advantages of cloud and open source
- Avoids vendor lock-in with standards-based software
- Certified Kubernetes to meet CNCF software conformance requirements for interoperability
- Accelerates creating and provisioning Open Container Initiative (OCI)-compliant containers
- Easily scales up and down Kubernetes clusters to manage peak workloads
- Deploys highly available Kubernetes clusters
- Can be deployed anywhere with multiple cloud and on-premises options
- Backed by enterprise-class worldwide support
- Reduces DevOps costs with support included at no extra charge with Oracle Linux Premier Support

Alongside CRI-O, Oracle Cloud Native Environment provides two runtime engines, runC and Kata Containers. runC is a container runtime based on the Linux Foundation's Runtime Specification (runtime-spec) and is developed by the Open Container Initiative.

Kata Containers is based on the upstream Kata Containers OpenStack Foundation project. Kata Containers delivers the framework for creating lightweight virtual machines that can easily plug into a container ecosystem. Kata Containers offers additional levels of security, while maintaining the same development and deployment speed of traditional containers. In fact, zero development changes are required.

### **Service mesh**

Istio is an open source project that is included with Oracle Cloud Native Environment and which coordinates communication between services, providing service discovery, load balancing, security, recovery, telemetry, and policy enforcement capabilities. Istio uses a sidecar service mesh model powered by the Envoy proxy. The Istio sidecar service mesh frees developers from having to program these types of capabilities into application code and makes development and enhancement of applications in a microservice architecture much more efficient and rapid.

### **Observability and diagnostics**

Prometheus is a powerful, flexible, instrumentation solution for monitoring container environments. It provides time-series dimensional data, powerful query tools, and alerting features to improve visibility across the environment. Oracle Cloud Native Environment includes an embedded version of Prometheus as part of the Istio service mesh, which incorporates an automatically configured instance of Grafana to provide both monitoring dashboards and alerting.

### **Great value**

Oracle Cloud Native Environment is free to download and use. Support for Oracle Cloud Native Environment is included with [Oracle Linux Premier support](#) at no additional cost.

### **Getting started**

To get started with Oracle Cloud Native Environment, you need have Oracle Linux which can be downloaded, used, and distributed free of charge and updates and errata are freely available from the [Oracle Linux yum server](#). Then follow the [documentation](#) to install required packages into Oracle Linux to install and configure Oracle Cloud Native Environment. You can download container images via [Oracle Container Registry](#), [GitHub Container Registry](#) or [Docker Hub](#) and deploy to Oracle Cloud Native Environment.

A great way to set up an Oracle Cloud Native Environment demo or [development environment](#) is to use [Oracle VM VirtualBox](#) which is the most popular cross-platform virtualization software. You can [download VirtualBox](#) to run Oracle Linux and the cloud native software on your desktop and easily deploy to the cloud.

---

## Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com/linux**. Outside North America, find your local office at: **oracle.com/contact**.

 [blogs.oracle.com/linux](https://blogs.oracle.com/linux)  [facebook.com/oraclelinux](https://facebook.com/oraclelinux)  [twitter.com/oraclelinux](https://twitter.com/oraclelinux)

---

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0821