

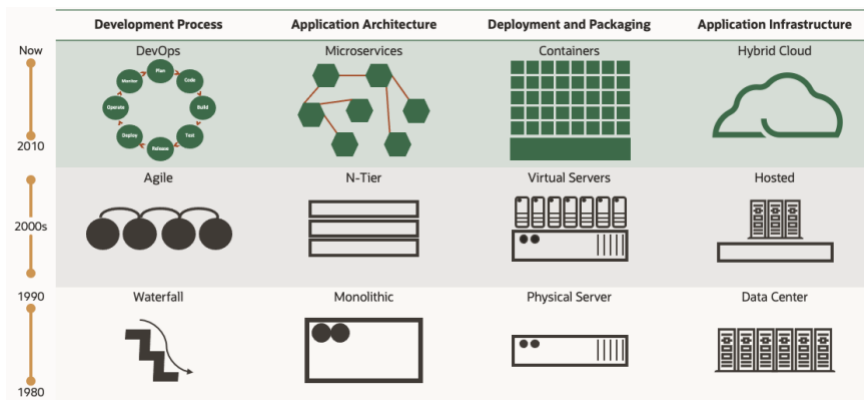
Oracle Cloud Native Environment

Oracle Cloud Native Environment is a selected set of open source software for the development and management of cloud native applications. Based on open standards, specifications, and APIs defined by the Open Container Initiative and Cloud Native Computing Foundation (CNCf), Oracle Cloud Native Environment has been tested for interoperability and provides a CNCf certified Kubernetes module for Oracle Linux. Oracle Cloud Native Environment delivers a simplified framework for installations, updates, upgrades, and configuration of key features for orchestrating microservices.

Evolution of application development and deployment

Application developers and development teams want an open, integrated platform to deploy container-based applications at scale across data centers, multicloud, and at the edge.

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 collaborating more effectively using DevOps methodologies to better align and speed up production deployments. This has allowed cloud native computing to evolve rapidly and enable organizations to build modern cloud applications that are resilient, manageable, and scalable.



Oracle Cloud Native Environment

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 Verrazzano](#)
- [Oracle VM VirtualBox](#)
- [Oracle Private Cloud Appliance](#)

Related support services

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

Related cloud services

- [Oracle Cloud Infrastructure Container Engine for Kubernetes](#)

Oracle Learning Library

- [Oracle Cloud Native Environment trainings](#)

Cloud Native Computing Foundation (CNCF)

CNCF is the open source, vendor-neutral hub of cloud native computing, hosting projects like Kubernetes and Prometheus to help make cloud native universal and sustainable. Oracle is a contributor to and platinum member of CNCF and the Linux Foundation.

Developing cloud native applications

Building and deploying cloud native applications can be challenging due to the rapidly evolving technology landscape. One option is to build your own cloud native environment from open source software, but this entails dealing with the complexity of picking the right software and getting everything to work together without vendor support. Another approach is to use a stack or distribution supported by an enterprise vendor but could potentially lock you in.

Oracle offers another alternative—one that can give you the best of both worlds by delivering open standards-based software promoted by the CNCF, the Open Container Initiative, and other best-of-breed cloud native projects, with enterprise-grade support and without lock-in.

Oracle Cloud Native Environment

Oracle Cloud Native Environment helps companies run applications using technologies such as Kubernetes, containers, and serverless functions for on-premises, Oracle Cloud Infrastructure, hybrid cloud, and multicloud environments. It provides an integrated set of components optimized to deploy, manage, and scale container-based applications without lock-in.

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 provides a production-ready environment for running containerized applications. It simplifies the configuration and setup of Kubernetes with support for backup and recovery. This solution 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 Container 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, the 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 an open source container runtime. 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 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.

Storage and Networking

Storage integration is provided through the use of plug-ins, referred to as the Container Storage Interface (CSI). The plug-ins adhere to a standard specification and include the Oracle Cloud Infrastructure Cloud Controller Manager module to set up dynamically provisioned persistent storage using Oracle Cloud Infrastructure and the Gluster Container Storage Interface module to set up dynamically provisioned persistent storage using Gluster Storage for Oracle Linux.

The Container Network Interface (CNI) project, currently incubating under CNCF, seeks to simplify networking for container workloads by defining a common framework for dynamically configuring networking resources. The CNI plug-in, included with Oracle Cloud Native Environment, simplifies container-to-container networking.

Great value

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

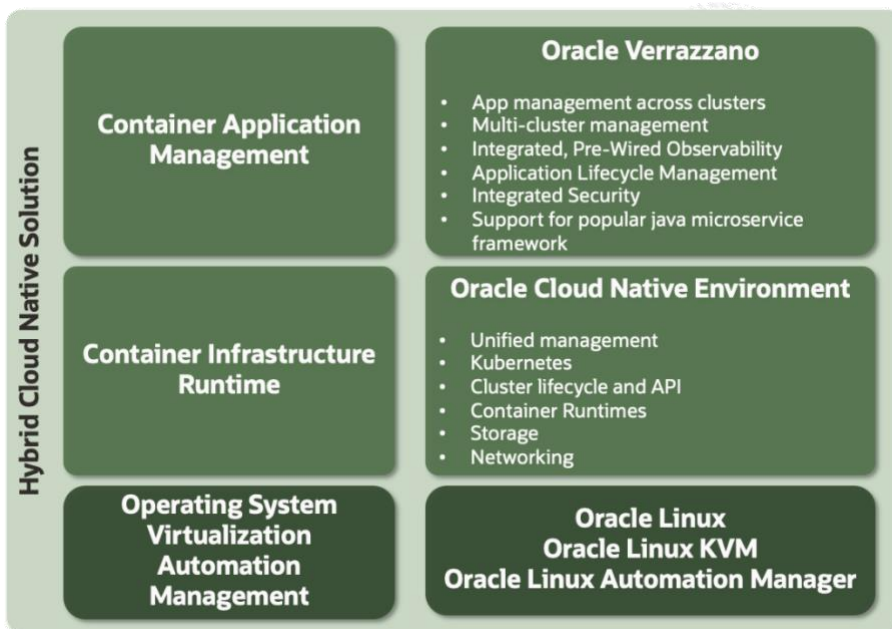
Extended value with Oracle Verrazzano Enterprise Container Platform

Most enterprises have invested in custom applications that are critical to their business. As the industry continues to move to the cloud, these enterprises are looking for flexible solutions that can enable them to adopt cloud native technologies to help improve productivity and accelerate innovation. By modernizing existing applications, these enterprises can get the benefits of cloud native capabilities and run their applications where they choose.

Oracle Verrazzano Enterprise Container Platform is a container deployment and management platform that provides a simplified application modernization, management, and observability stack. It runs on top of Kubernetes, which makes it ideal to deploy with Oracle Cloud Native Environment, allowing container applications to be deployed on the Kubernetes clusters, on-premises or in the cloud.

Oracle Verrazzano Enterprise Container Platform is certified and supported on Oracle Linux and Oracle Cloud Native Environment. It is available via subscription from Oracle and enables enterprises to:

- Deploy, manage, and secure container-based applications on a Kubernetes cluster.
- Unify application lifecycle management across microservices and traditional server applications.
- Standardize management across Kubernetes clusters on-premises, on Oracle Cloud Infrastructure, and on other public clouds.



Additional details are available on the [Oracle Verrazzano Enterprise Container Platform datasheet](#).

Getting started with Oracle Cloud Native Environment

Oracle Linux is a prerequisite to getting started with Oracle Cloud Native Environment. The first step is to install 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 and configure the required Oracle Cloud Native Environment packages into Oracle Linux. You can download container images via [Oracle Container Registry](#), [GitHub Container Registry](#) or [Docker Hub](#) and deploy to Oracle Cloud Native Environment.

[Free training videos](#) are also available to help you quickly adopt 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](#), the most popular open source virtualization software. You can [download VirtualBox](#) to run Oracle Linux and Oracle Cloud Native Environment 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](#)  [facebook.com/oraclelinux](#)  [twitter.com/oraclelinux](#)

Copyright © 2022, 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.

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