

Oracle WebLogic Server Multitenant: The World's First Cloud-Native Enterprise Java Platform



KEY BENEFITS

- Enable container-like DevOps and 12-factor application management and delivery
- Accelerate application delivery on premises and in the cloud
- Reduce Total Cost of Ownership of server-side Java infrastructure
- Simplify with Java Cloud Infrastructure

KEY FEATURES

- Pluggable partitions that serve as microcontainers for business services
- Shared platform for high-density deployments
- Isolation between microcontainers within the shared infrastructure
- End-to-end integration across the Web tier, middle tier, cache tier, and data tier

PREREQUISITES

- WebLogic Server Enterprise Edition
or
- WebLogic Suite

More and more organizations are looking to adopt cloud computing and microservice architectures as a means to improve general IT agility, and to reduce costs for the applications that are essential to run their businesses. WebLogic Server Multitenant provides container-like capabilities for Java applications to support 12-factor application strategies, while still utilizing application server facilities, such as clustering, transaction management, and security. WebLogic Server Multitenant also enables you to consolidate these microcontainers onto an efficient, shared platform. Using WebLogic Server Multitenant improves your time to market, simplifies movement of workloads to and from the cloud, enables up to 3X hardware consolidation, and reduces operating expenses by up to 25%.

WebLogic Server Multitenant is fully integrated with Oracle Traffic Director, and Oracle Coherence, and is a natural fit for use with Oracle Database 12c pluggable databases, making it the perfect solution as an end-to-end agile application platform.

Microcontainers for Business Services

Many companies have a large set of business-critical applications written as Java EE applications. These applications rely on the facilities in a Java Application Server, including configuration, management, security, database connectivity, transaction management, and integration technologies. While there is a desire to move to microservices, these services will continue to rely on these application server facilities.

With WebLogic Server Multitenant, you can migrate your Java applications to microcontainers and run them on a shared platform. You can optionally break your monolithic Java EE applications into smaller services, but continue to depend on proven and reliable application server facilities, such as clustering, transaction management, and security. There is no need to create or assemble these facilities from multiple sources, and then verify their quality and performance.

Light Weight Encapsulation

At the heart of the microservice architecture movement are a few key required aspects:

- Disposability – Fast creation/startup/shutdown of application processes
- Maximum portability between environments
- Parity between development and production

“We are excited about many of the innovations in Oracle WebLogic Server 12cR2. We believe that the new Oracle WebLogic Server Multitenant offering, with its pluggable partitions, will accelerate application delivery, and allow for implementing development best practices. We are convinced that the new features will help us to increase cross-team productivity, and – through support for maximum availability architectures – will substantially improve our quality of service.”

RALF ERNST
SENIOR IT ARCHITECT
BUNDESAGENTUR FÜR ARBEIT

Pluggable partitions in WebLogic Server Multitenant provide just enough encapsulation to deliver on these required characteristics. Pluggable partitions include application deployments and the system resources that those applications depend on, but they do not include pieces of the operating system nor the Java virtual machine and the application server. With optimized package contents, provisioning and starting a microcontainer is very fast. Memory footprint is minimized, and there is no need for the operating system, the JVM, and the process to start. Instead, only the application needs to start, which means startup times are very short, and incremental memory usage is low.

Maximum Portability

Despite the optimized contents of the container, pluggable partitions offer simple and utmost portability across environments. Tools are available for exporting pluggable partitions from one domain and for importing it into another domain. This enables movement from development to testing to production, and from on-premises to cloud environments. During this process, pluggable partitions can be imported without change for maximum parity across environments, or with modifications to tailor the application to the new environment.

Pluggable partitions can also be moved between clusters while they are running, without session loss. This Live Migration makes it possible to re-balance load across clusters and to handle planned maintenance without downtime. (Live Migration requires the WebLogic Server Continuous Availability option.)

Oracle Traffic Director can be a significant enabler in such a flexible environment. Oracle Traffic Director is a software load balancer with deep knowledge of the WebLogic Server runtime, and specific integration with WebLogic Server Multitenant. When a partition is created or imported into a domain, traffic routing for the partition is automatically added to the Traffic Director configuration. If the partition is moved, Traffic Director is notified of the move. The result is that the load

Highly Efficient Shared Platform for Your Java Applications

While encapsulation and portability are important for getting applications to market as quickly as possible, efficiency in the application platform is equally important. Typical application deployments include a dedicated set of virtual machines that host a dedicated WebLogic domain. Virtual machine and domain capacity is reserved at the maximum traffic level. This pattern of over-provisioned, dedicated domains means that capacity is trapped and underutilized, even with server virtualization. Add to that the fact there are many individual domains and JVMs that need to be managed and maintained.

WebLogic Server Multitenant offers a logical virtualization within the application server. It includes administrative constructs and infrastructure, including pluggable partitions, that enable you to share domains for many applications. By pushing virtualization higher in the technology stack, more of the stack can be shared. More sharing results in better efficiency. In internal benchmarks, we consolidated 10 applications into a single domain, and then ran those consolidated domains on one third the number of physical hosts. This consolidated environment resulted in a 66% savings in capital costs and a 25% savings in operational costs. These savings were realized without impact to response times or throughput rates for the applications running on the shared environment.

Isolation

Virtualization within the application server naturally implies some form of isolation between the applications running on the shared platform. For WebLogic Server Multitenant, Oracle built several forms of isolation that are critical for consolidating independent applications into a shared domain, including runtime isolation, security isolation, administrative isolation, and data isolation.

Runtime Isolation and Resource Management

Particularly within a shared Java virtual machine, it is important to enable sharing compute resources while protecting applications from other applications that may hog those resources. To enable this isolation, you can configure Resource Consumption Managers and Partition Work Managers.

Resource Consumption Managers are the result of a partnership between the WebLogic Server and Java Standard Edition development teams. These teams came together to design and implement the facility for managing retained heap, CPU time, and open file descriptors used by each pluggable partition. For each partition, you can configure policies that enable fair-share use of these low-level resources, and you can configure policies that define boundaries for use of these resources. These policies enable the JVM and WebLogic Server to manage the usage of heap, CPU time, and open file descriptors so that no single partition will hog these resources and negatively impact the other partitions running in the same JVM.

Similarly, Partition Work Managers enable you to define the fair share ratio for request processing within a WebLogic managed server. This ratio is used to prioritize request processing for all work done in the managed server. You can use the fair share ratio to set a generic higher priority for one partition over another, in which case, more requests for the higher-priority partition would be processed than for the lower priority partition, in a given time period. Partition Work Managers can also be used to limit the threads that any one partition can use at a time.

Used in combination, Resource Consumption Managers and Partition Work Managers provide the runtime isolation and protection needed for applications running in a shared environment.

Security Isolation

Security is important in all cases, and it is very important for applications running on a shared platform. In WebLogic Server Multitenant, each microcontainer can have a separate security realm. This enables you to define a separate authentication and authorization provider for the microcontainer, as well as a separate set of administrators and users.

Administrative Isolation

Each microcontainer can represent an application or business service that needs to be independently managed from all other services running on the shared platform. WebLogic Server Multitenant enables just such a set of administrative isolation features. Each microcontainer can be managed by its own set of administrators. It can also be started, stopped, configured, and updated independently. This level of administrative isolation enables independence for each service running on a shared platform.

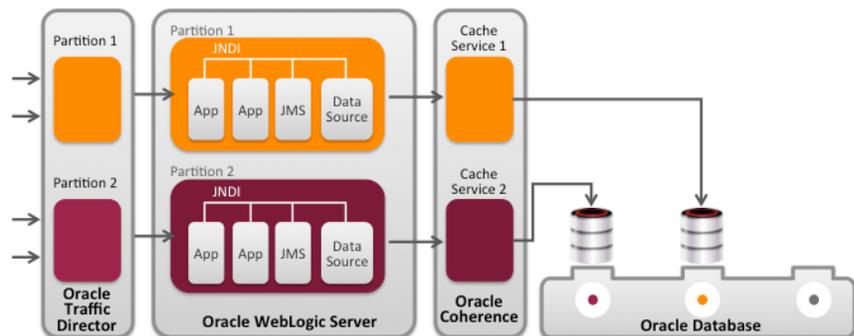
Data Isolation

Because each microcontainer has its own dedicated JNDI naming service, data source configuration, Java Message Service (JMS) configuration, and dedicated Oracle Coherence runtime cache service, data for the microcontainer is isolated from the data for all other microcontainers. A dedicated naming service allows for reuse of applications, but without naming conflicts across microcontainers. Data sources in a microcontainer provide a natural pointer to a separate data set – a hallmark of a microservice architecture. Because JMS messages are also stored and processed for an individual microcontainer, data in JMS messages is also naturally segregated. With dedicated cache services within a shared Coherence cluster, cached data is also segregated for each microcontainer.

End-to-End Integration

Microcontainers within WebLogic Server are only part of the story for WebLogic Server Multitenant. The solution provides end-to-end integration and ease of use with *Oracle Traffic Director*, *Oracle Coherence*, and *Oracle Database 12c* pluggable databases.

ORACLE WEBLOGIC SERVER MULTITENANT END-TO-END INTEGRATION



WebLogic Server Multitenant End-to-End Integration

Included in WebLogic Server Multitenant is the integration with Oracle Traffic Director, which is a high throughput, low latency HTTP(s), WebSocket, and TCP software load balancer. In a WebLogic Server Multitenant environment, Traffic Director can act as a microcontainer gateway. When you add a pluggable partition to a WebLogic Server configuration, Traffic Director configuration is automatically updated to include routing information to the services running in the microcontainer. When you move the microcontainer to a different environment, Traffic Director automatically routes traffic to the new location.

Oracle Coherence, the industry leading in-memory data grid solution, is also part of the end-to-end integration story. Coherence clusters can be part of the shared infrastructure, which means additional efficiency, ease of use, and cost savings. For services running in microcontainers, WebLogic Multitenant can automatically create dedicated or shared cache services within a shared Coherence cluster without additional manual configuration.

Because WebLogic Server Multitenant shares a similar multitenancy model, Oracle Database 12c pluggable databases are a natural fit for use in an end-to-end multitenant environment. Each microcontainer can have dedicated data source configurations that

connect to segregated data. Pluggable databases provide an efficient solution for hosting that segregated data in a shared data platform. Additionally, both pluggable databases and pluggable partitions offer exceptional ease of portability between environments, including site to site and on-premises to the cloud, for WebLogic Server, Coherence, and the Oracle Database.

Summary

Oracle WebLogic Server Multitenant provides maximum application portability, 3X consolidation with a shared application platform, and application isolation and independence, all in an end-to-end integrated solution.

Oracle WebLogic Server Multitenant enables you to encapsulate business services in microcontainers called pluggable partitions. These microcontainers enable you to quickly, easily, and efficiently move an application and its required resources from environment to environment, and from on-premises to the cloud.

By consolidating applications in microcontainers on a shared application platform, you can run three times the number of applications on the same hardware, which means a 66% savings in CAPEX costs. Use of a shared platform also means reduced administrative and operational costs, estimated at up to 25%.

WebLogic Server Multitenant provides runtime, security, administration, and data isolation for microcontainers. Coupled with the automated end-to-end integration with Oracle Traffic Director, Oracle Coherence, and Oracle Database, WebLogic Server Multitenant is a truly useful and usable consolidation and portability solution for your Java applications.

CONTACT US

For more information about [insert product name], visit oracle.com or call +1.800.ORACLE1 to speak to an Oracle representative.



CONNECT WITH US



Integrated Cloud Applications & Platform Services

Copyright © 2015, 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. 1015