

Oracle® Fusion Middleware

WebLogic Server Administration

Lab 10: Side by Side Deployment (Production Redeployment)



14.1.2 & 15.1.1
DT246055-12
July 2026



Oracle Fusion Middleware: WebLogic Server Administering (14.1.2 & 15.1.1)

DT246055-12

Copyright © 2021, 2026, DigiTalk Systems and/or its affiliates.

Primary Author: DigiTalk Systems

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with Oracle WebLogic Server. The Company wishes to emphasize that this document is not affiliated with Oracle Corporation ("Oracle") in any way, and the content contained herein is based solely on publicly available product documentation provided by Oracle.

While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with Oracle's official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitalk.fmw@gmail.com.

We would like to make it clear that the code snippets and examples used in this document are sourced from publicly available Oracle demo applications, provided by Oracle Corporation for educational and learning purposes. These materials are subject to Oracle's copyright and licensing terms.

We would like to emphasize that our charges are solely for the efforts and resources invested in preparing the lab documents and explanations to enhance the learning experience. DigiTalk is not charging for the usage of Oracle's copyrighted material.

By using or accessing these documents, you acknowledge and agree that the Oracle code snippets and examples are the intellectual property of Oracle Corporation. We do not claim ownership of the Oracle code, and any trademarks or copyrights associated with Oracle remain the property of Oracle Corporation.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by Oracle.

The Company makes no claims or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitalk.fmw@gmail.com, and the Company will endeavour to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.

Contents

About Production Redeployment (Side by Side Deployment)

How Production Redeployment Works

Production Redeployment and Cluster

Deployment Requirements

Lab Verification – Production Redeployment (Side-by-Side Deployment)

Redeploying a New Version of an Application

About Production Redeployment (Side by Side Deployment)

The production redeployment strategy entails introducing a new iteration of an updated application alongside an existing older version of the same application. WebLogic Server automatically oversees client connections, ensuring that only fresh client requests are routed to the new version. Clients already linked to the application during redeployment persist in using the older version until they finish their tasks. Subsequently, WebLogic Server autonomously phases out the older application.

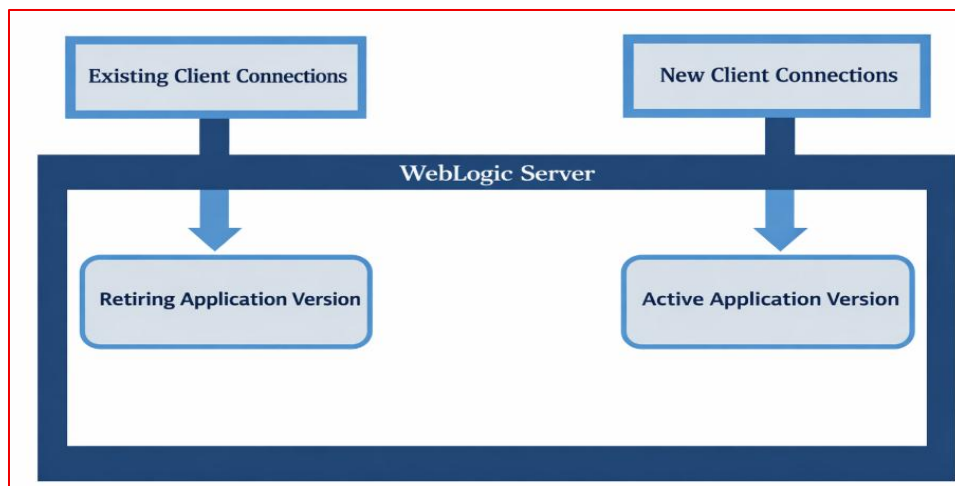
Production redeployment allows for the seamless updating and redeployment of an application in a production environment without interrupting the application or affecting client accessibility. **This method eliminates the need for scheduling application downtime, establishing redundant servers for hosting new application versions, manually overseeing client access to multiple application versions, and manually retiring older application iterations.**

How Production Redeployment Works

WebLogic Server executes production redeployment by introducing a new version of an application alongside an existing, operational version of the same application. During redeployment, one application version remains "active" while the other is in a "retiring" state. The active version handles all incoming client connection requests, while the retiring version manages only the client connections that were established during the redeployment process. WebLogic Server discontinues the retiring application version once all current clients have completed their tasks or when a predetermined timeout is reached.

Upon redeploying a new application version, WebLogic Server designates the newly-deployed version as active and initiates the retirement process for the older version. Throughout the retirement phase, WebLogic Server automatically monitors the application's HTTP sessions and ongoing transactions. Each HTTP session is tracked until completion or timeout, and in-progress transactions are monitored until they conclude, roll back, or reach the transaction timeout period.

If needed, you can revert the production redeployment by making the older application version active. This could be necessary, for instance, if issues arise with the newer version, and you wish for WebLogic Server to transition clients back to the older version. To activate the older application version, simply redeploy it.



Production Redeployment and Cluster

Sample Lab Document

Deployment Requirements

Sample Lab Document

Lab Verification – Production Redeployment (Side-by-Side Deployment)

This lab demonstrates WebLogic Production Redeployment, where the old and new versions of an application run simultaneously during a transition period. An active HTTP session is required to observe this behavior.

The sample application creates an HTTP session automatically and displays the session details, including the Session ID, Session Creation Time, Last Access Time, and Current Time.

Verification Steps

1. Deploy Version 1 (V1) of the application.
2. Access the application using a web browser.
3. Verify that the application displays:
 - Application Version (V1)
 - Session ID
 - Session Creation Time
 - Last Access Time
 - Current Date and Time
4. Keep this browser window open. This browser maintains an active HTTP session with Version 1.
5. Deploy Version 2 (V2) of the application using Production Redeployment (Side-by-Side Deployment).
6. Open a new browser (or an Incognito/Private window) and access the same application URL.
7. Verify that the new browser displays Application Version V2, confirming that new client sessions are routed to the new application version.
8. Return to the original browser window (which is still using the active V1 session) and refresh the page.
9. Verify that:
 - The application continues to display Version V1.
 - The Session ID remains unchanged.
 - The Current Date and Time are updated on each refresh, confirming that the original session is still active and being served by Version 1.

Expected Result

- Existing HTTP sessions continue to use Version 1 until they expire or are retired.
- New HTTP sessions are served by Version 2.
- This demonstrates WebLogic's Production Redeployment (Side-by-Side Deployment) capability, allowing seamless application upgrades without interrupting active user sessions.

Redeploying a New Version of an Application

Demo Application:

We have two versions of the sample application HelloWorld

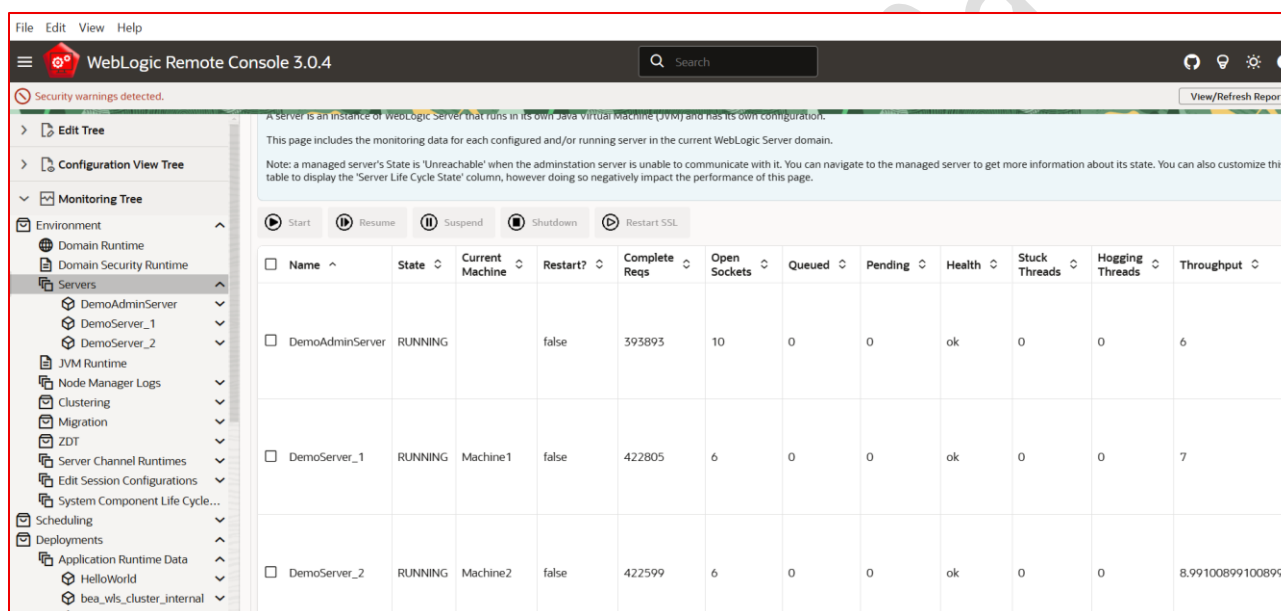
- Version 1: MANIFEST.MF contain “Weblogic-Application-Version: v1” entry
- Version 2: MANIFEST.MF contain “Weblogic-Application-Version: v2” entry

```
Manifest-Version: 1.0
Created-By: Maven WAR Plugin 3.4.0
Build-Jdk-Spec: 21
Weblogic-Application-Version: V1
```

```
Manifest-Version: 1.0
Created-By: Maven WAR Plugin 3.4.0
Build-Jdk-Spec: 21
Weblogic-Application-Version: V2
```

LAB Domain:

We have two managed servers running in a cluster, and we will deploy the application on cluster.



The screenshot shows the WebLogic Remote Console 3.0.4 interface. On the left, the 'Monitoring Tree' is expanded, showing the 'Servers' folder. The main area displays a table of server status for three servers: DemoAdminServer, DemoServer_1, and DemoServer_2. All servers are in a 'RUNNING' state. The table includes columns for Name, State, Current Machine, Restart?, Complete Reqs, Open Sockets, Queued, Pending, Health, Stuck Threads, Hogging Threads, and Throughput.

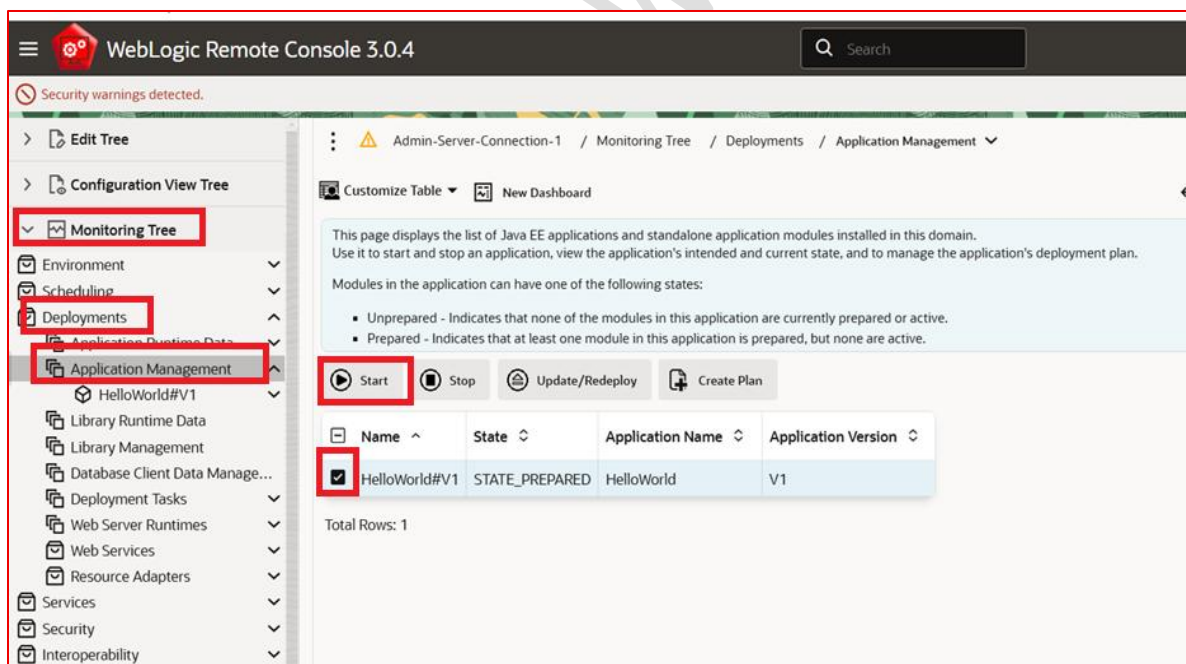
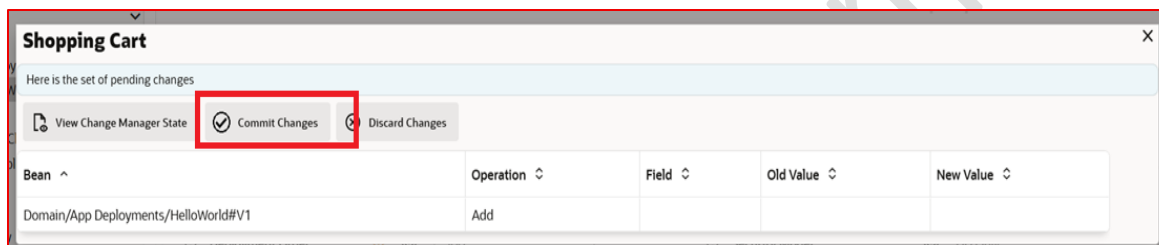
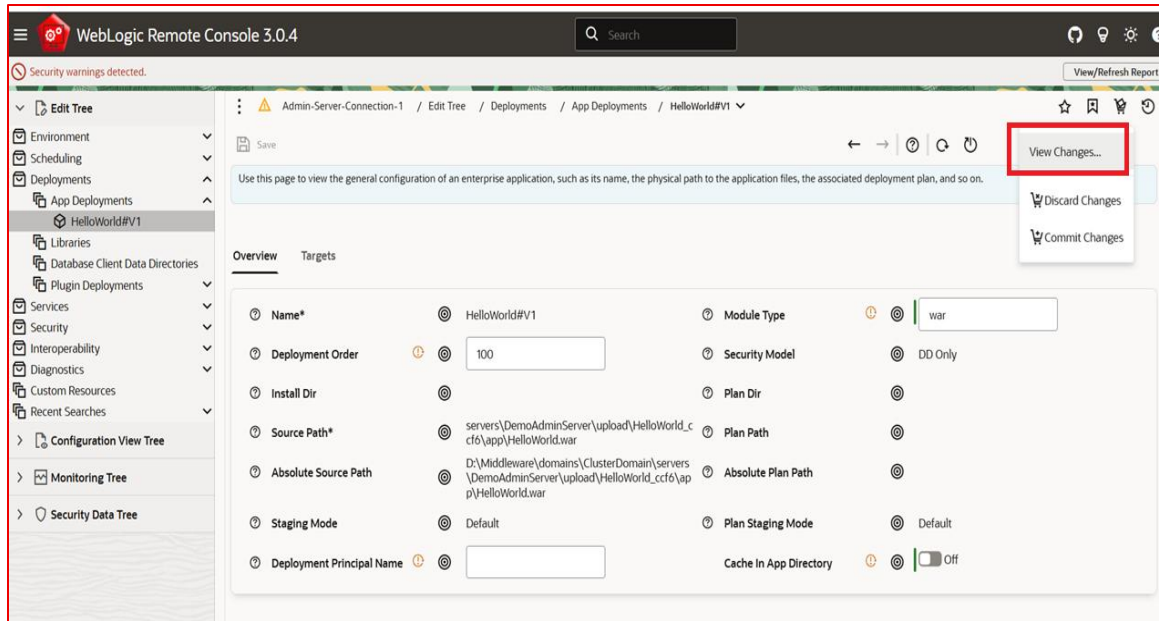
Name	State	Current Machine	Restart?	Complete Reqs	Open Sockets	Queued	Pending	Health	Stuck Threads	Hogging Threads	Throughput
DemoAdminServer	RUNNING		false	393893	10	0	0	ok	0	0	6
DemoServer_1	RUNNING	Machine1	false	422805	6	0	0	ok	0	0	7
DemoServer_2	RUNNING	Machine2	false	422599	6	0	0	ok	0	0	8.991008991008991



Deploy V1 version from Console

Sample Lab Document

digitalk.fmw@gmail.com

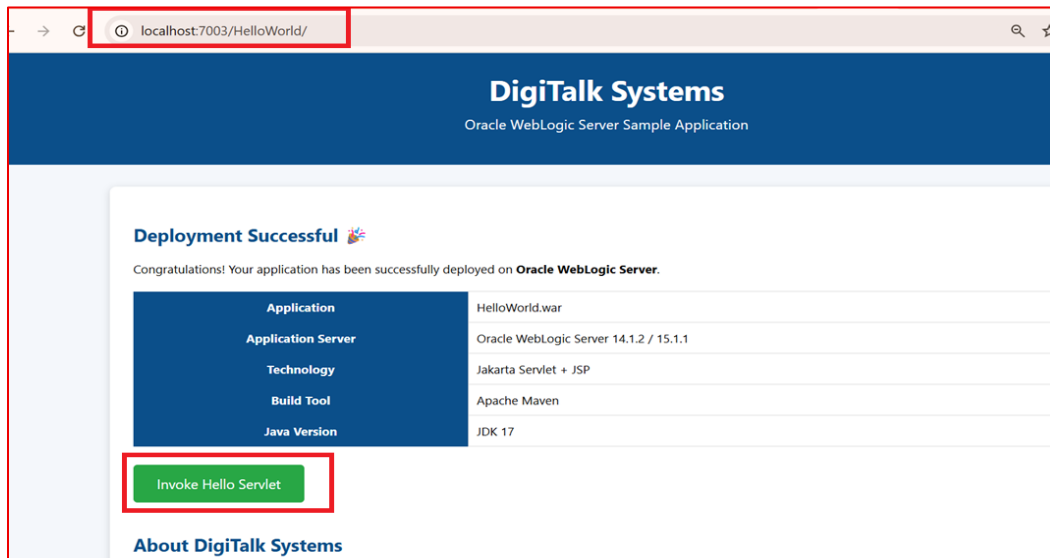


Sample Lab Document

digitalk.fmw@gmail.com

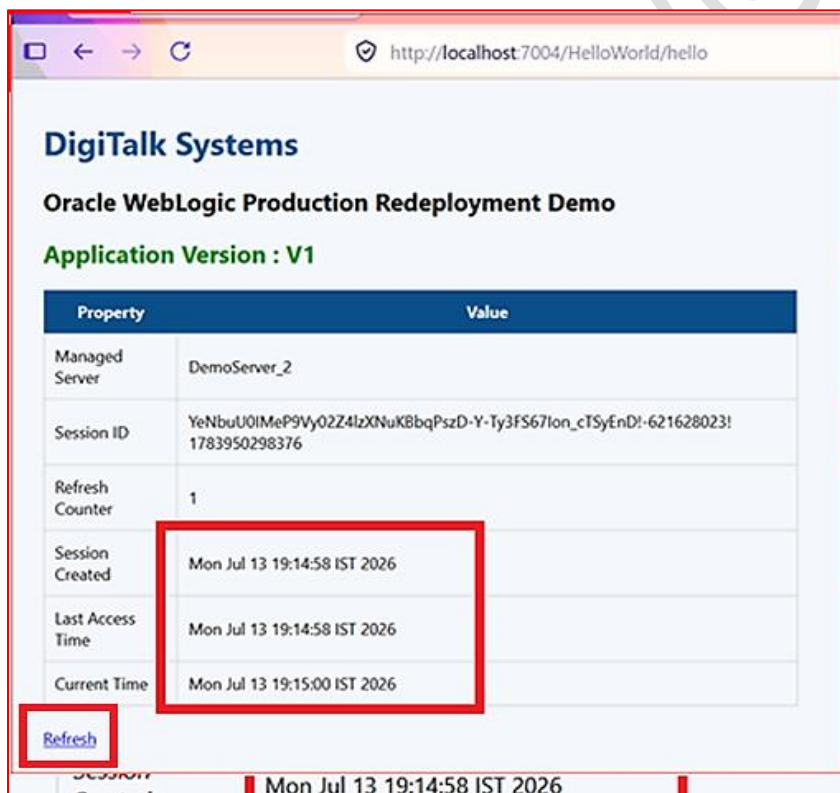
Access Application:

<http://localhost:7003/HelloWorld/>



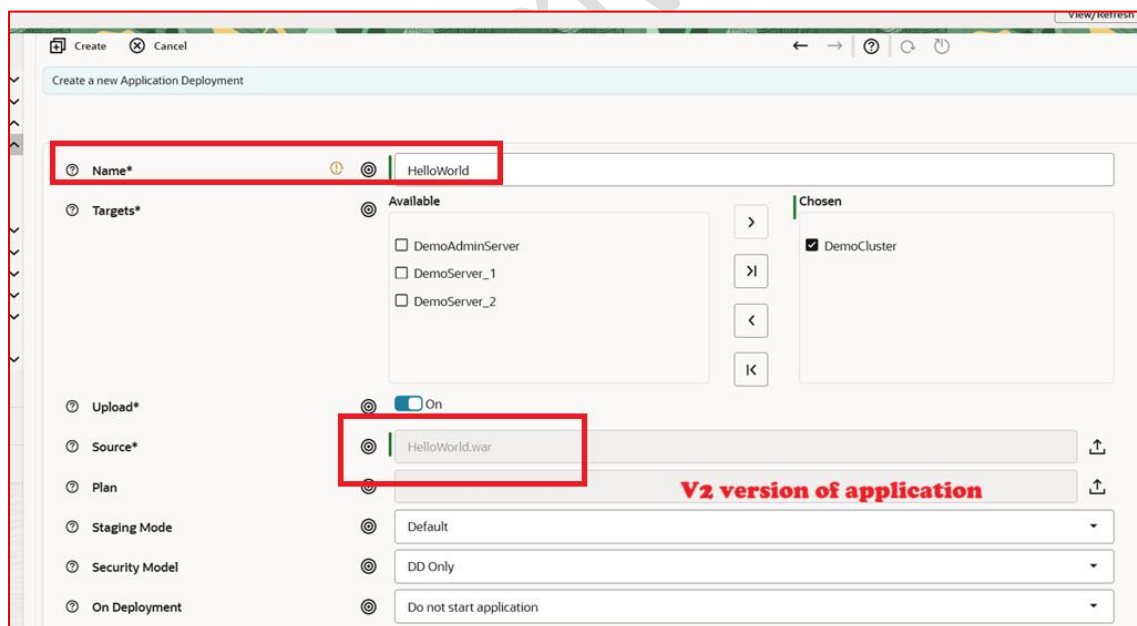
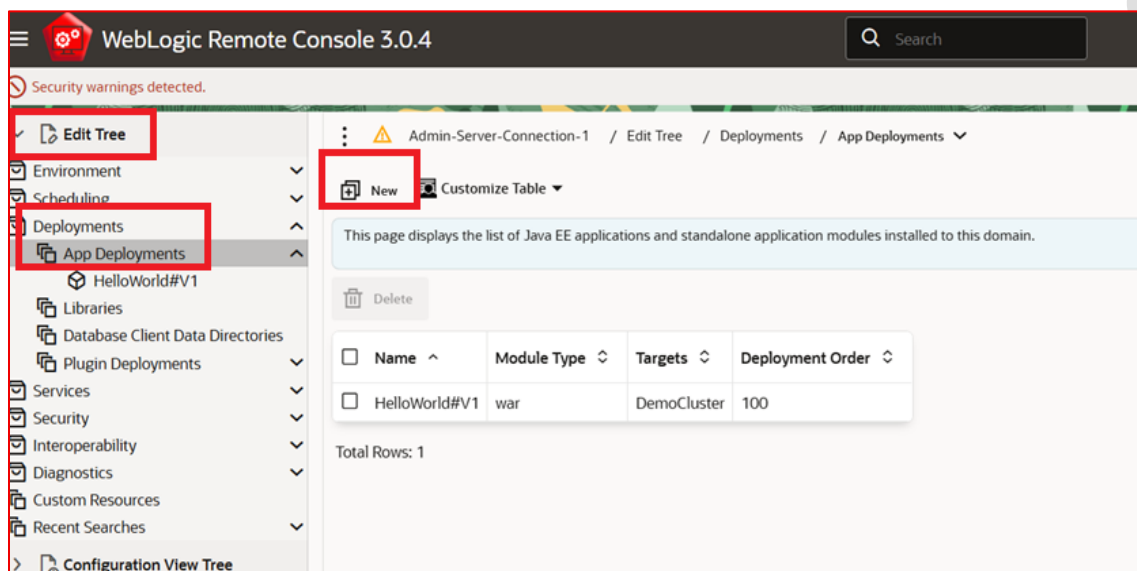
Click on "Invoke Hello Servlet"

Keep this browser session open. Once Version V2 deployed, we will click on Refresh button on page (Not browser) to ensure its still active, and the session date time will refresh to current date time.



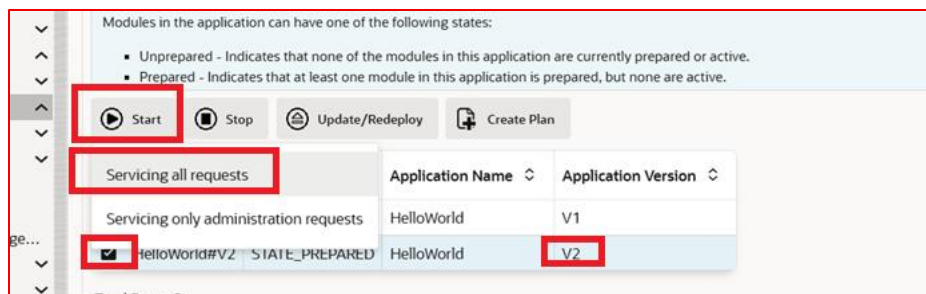
Update application Version to 2.0

Unlike WebLogic Server 12c and 14.1.1, where a new application version is deployed using the Update (Production Redeployment) option, WebLogic Server 14.1.2 and later (using the WebLogic Remote Console) deploys a new application version by selecting Install (New Deployment) while specifying the same application name and a different Weblogic-Application-Version in the application manifest. WebLogic recognizes the deployment as a new version of the existing application and performs Production Redeployment (Side-by-Side Deployment), allowing existing sessions to continue on the old version while new sessions are routed to the new version.



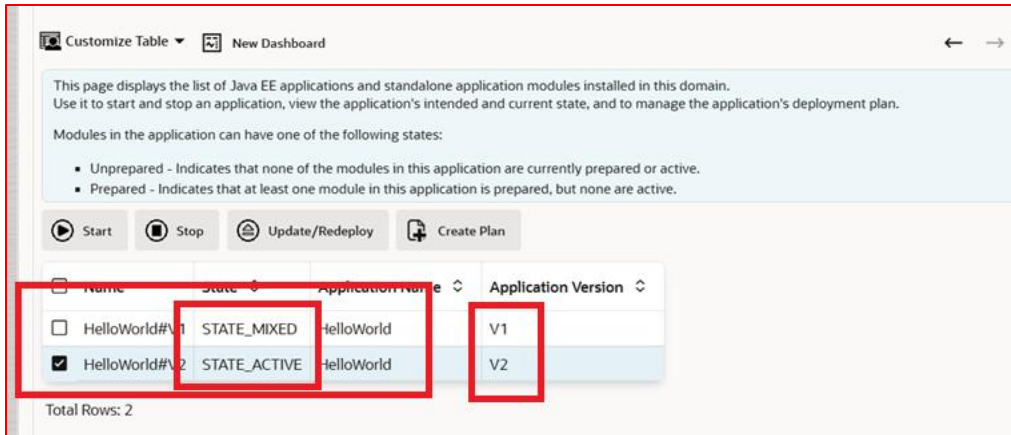
Sample Lab Document

digitalk.fmw@gmail.com



STATE_MIXED states doesn't mean application is retired, it has still connected sessions processing while all the new request are now going to V2.0

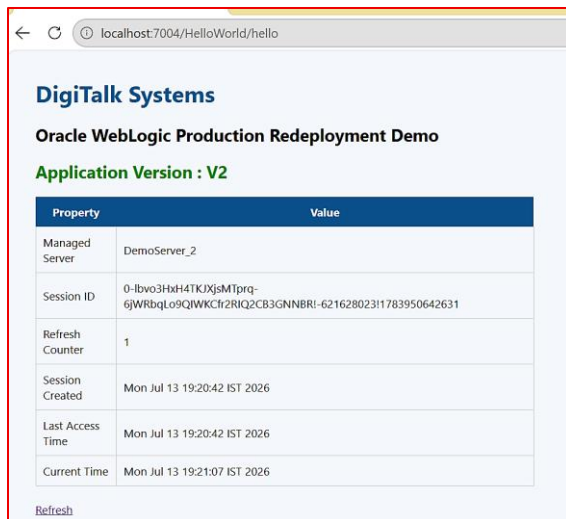
Sample Lab Document



Access application:

<http://localhost:7003/HelloWorld/>

Sample Lab Document



Property	Value
Managed Server	DemoServer_2
Session ID	0-1bvo3HxdH4TKjXjsMTprq-6jWRbqLo9QWKCfr2RIQ2CB3GNNBRI-62162802311783950642631
Refresh Counter	1
Session Created	Mon Jul 13 19:20:42 IST 2026
Last Access Time	Mon Jul 13 19:20:42 IST 2026
Current Time	Mon Jul 13 19:21:07 IST 2026

[Refresh](#)

Test Application Again

To see whether the old version is still running for old users, return to the old web browser session which was kept open initially. As long as the HTTP Session is valid, the old user will continue to use "Version 1."

Sample Lab Document

In case of any typo, mistake or suggestions please write on digitalk.fmw@gmail.com