12

Advanced Deployment

Objectives

After completing this lesson, you should be able to do the following:

- Configure an application for multiple development environments
- Create a deployment plan
- Stage a deployment plan
- Use production redeployment

Road Map

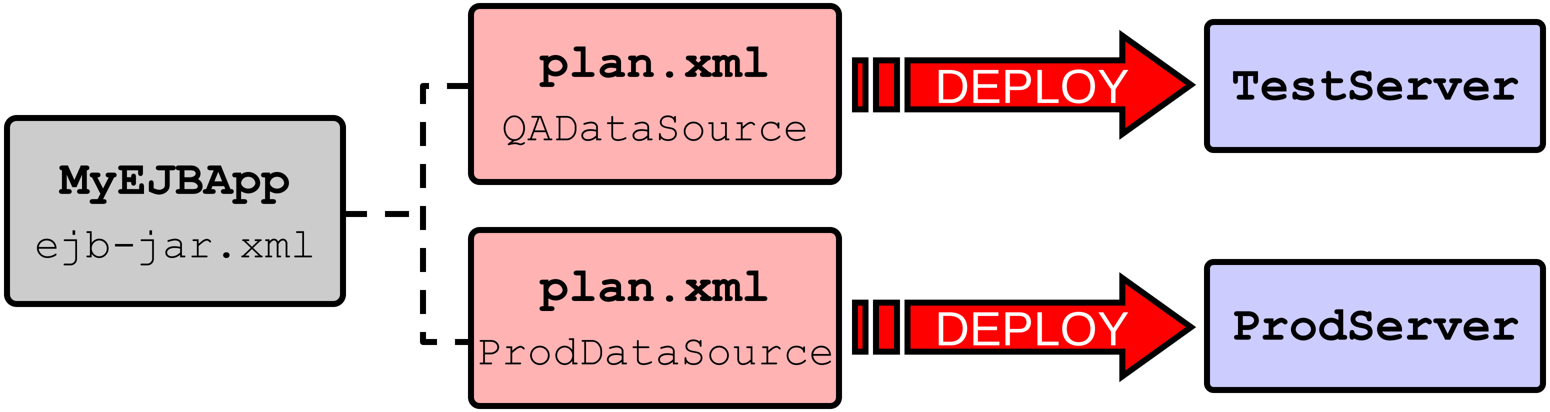
- Deployment plans
- Staged deployment
- Production redeployment

ORACLE

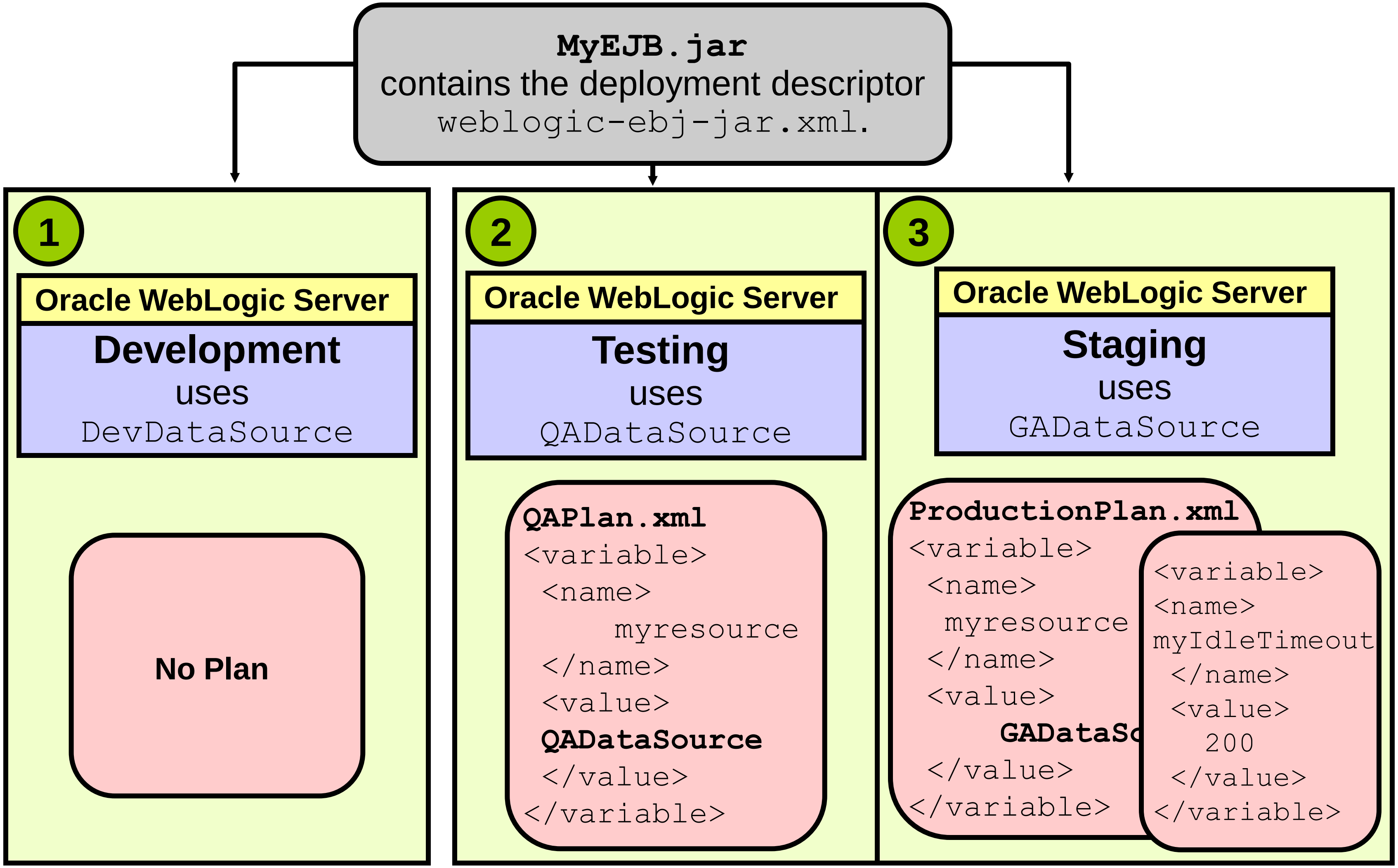
What Is a Deployment Plan?

A Java EE deployment plan:

- Is an optional XML file associated with an application
- Resides outside an application archive
- Sets or overrides the values in the Java EE deployment descriptors
- Allows a single application to be easily customized to multiple deployment environments



Configuring an Application for Multiple Deployment Environments



MyEJB.jar
contains the deployment descriptor
weblogic-ebj-jar.xml.

1

Oracle WebLogic Server

Development
uses
DevDataSource

No Plan

2

Oracle WebLogic Server

Testing
uses
QADataSource

```
QAPlan.xml
<variable>
 <name>
     myresource
 </name>
 <value>
 QADataSource
 </value>
</variable>
```

3

Oracle WebLogic Server

Staging
uses
GADataSource

```
ProductionPlan.xml
<variable>
 <name>
  myresource
 </name>
 <value>
     GADataSo
 </value>
</variable>
```

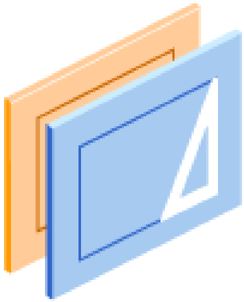
```
<variable>
<name>
myIdleTimeout
 </name>
 <value>
   200
 </value>
</variable>
```

Sample Deployment Plan

```
<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://www.bea.com/ns/weblogic/90"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.bea.com/ns/weblogic/90
     http://www.bea.com/ns/weblogic/90/weblogic-deployment-plan.xsd" global-
     variables="false">
  <application-name>myApp</application-name>
  <variable-definition> <variable>
      <name>WeblogicWebApp_ContextRoots_11945442520421</name>
      <value>/beasys</value>    </variable> </variable-definition>
  <module-override>
    <module-name>benefits_as_default.war</module-name>
    <module-type>war</module-type>
    <module-descriptor external="false">
      <root-element>weblogic-web-app</root-element>
      <uri>WEB-INF/weblogic.xml</uri><variable-assignment>
        <name>WeblogicWebApp_ContextRoots_11945442520421</name>
        <xpath>/weblogic-web-app/context-root</xpath>
     <operation>replace</operation>
      </variable-assignment> </module-descriptor>
    <module-descriptor external="false">
      <root-element>web-app</root-element>
      <uri>WEB-INF/web.xml</uri>    </module-descriptor> </module-override>
  <config-root xsi:nil="true"></config-root></deployment-plan>
```

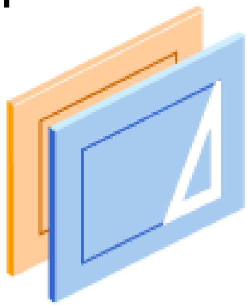
Creating a Deployment Plan

- Tools for creating a deployment plan:
  - Development tool—for example, JDeveloper or Eclipse
  - `weblogic.PlanGenerator`
  - Administration Console
- Goals for creating a deployment plan:
  - To expose the external resource requirements of the application as variables in the deployment plan
  - To expose additional configurable properties, such as tuning parameters as variables in the deployment plan



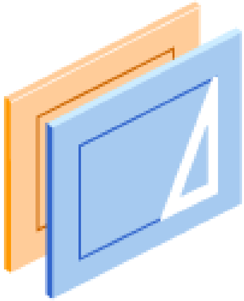
Creating a New Deployment Plan

- The WebLogic Server includes tools to accelerate deployment plan creation.
- The Administration Console:
  - Generates a skeleton `plan.xml` if a plan folder is detected with a newly deployed application
  - Updates the `plan.xml` when you use the console to modify the deployment descriptor settings
- The `weblogic.PlanGenerator` Java class can also generate a skeleton `plan.xml` for an existing application.



weblogic.PlanGenerator

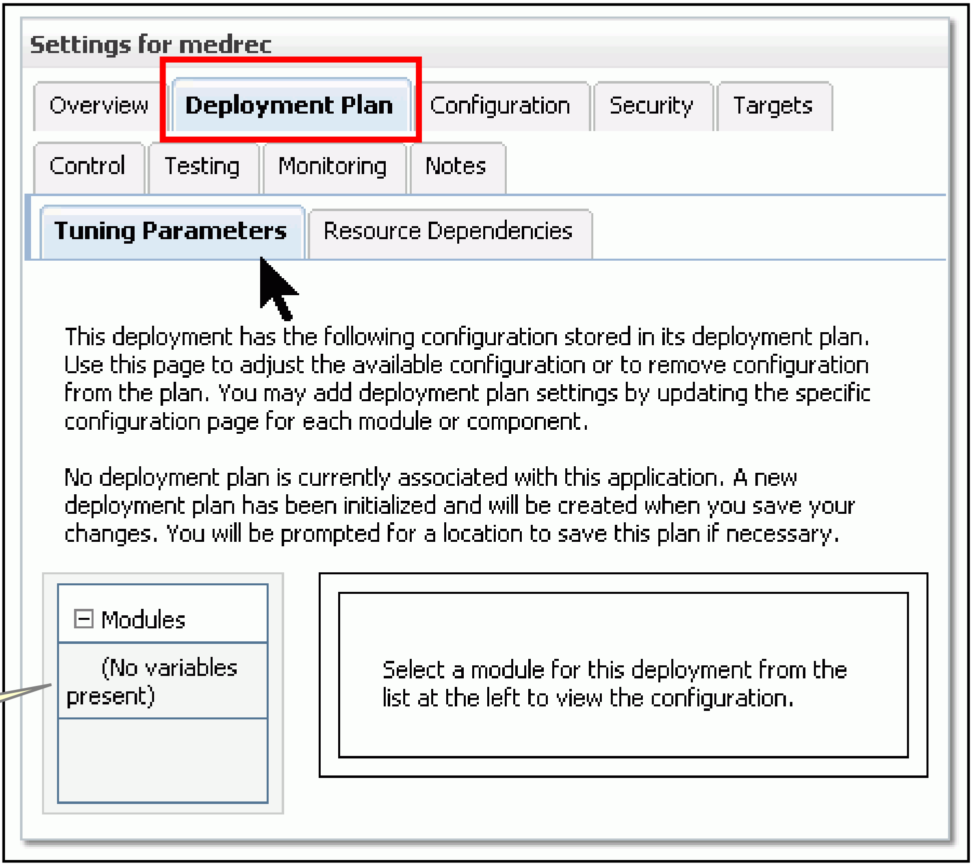
- Enables you to generate a basic Oracle WebLogic Server configuration for applications that have only Java EE deployment descriptors
- Enables you to:
  - Create an initial plan
  - Create a new plan based on an existing plan
  - Control components exported to a plan



Using the Administration Console to Generate a Deployment Plan

You can generate a deployment plan with the Administration Console using the following steps:

1. Prepare the deployment files.
2. Install the application archive.
3. Save the configuration changes to a deployment plan.



Settings for medrec

This deployment has the following configuration stored in its deployment plan. Use this page to adjust the available configuration or to remove configuration from the plan. You may add deployment plan settings by updating the specific configuration page for each module or component.

No deployment plan is currently associated with this application. A new deployment plan has been initialized and will be created when you save your changes. You will be prompted for a location to save this plan if necessary.

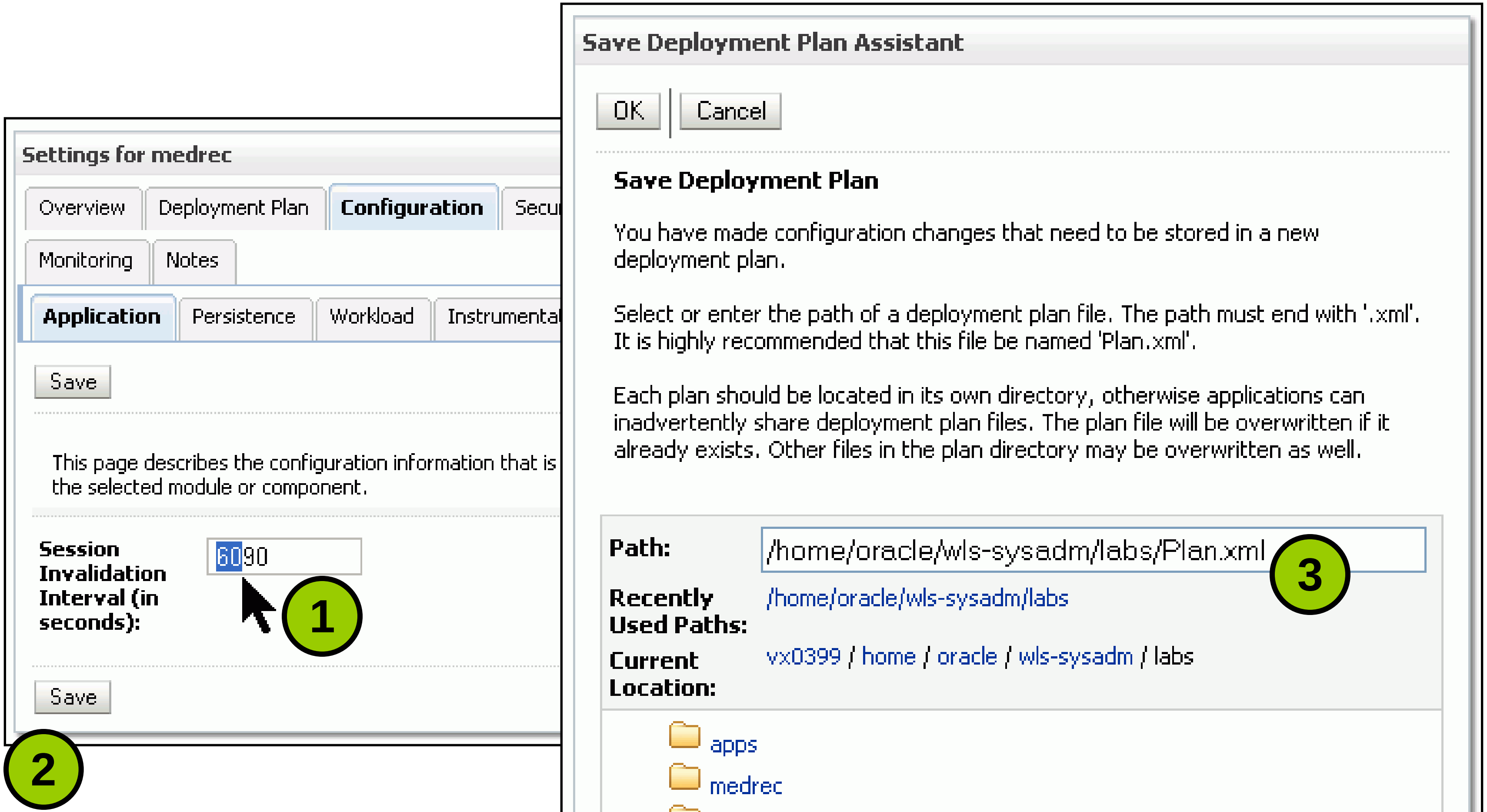
⊟ Modules

(No variables present)

Select a module for this deployment from the list at the left to view the configuration.

Before: All empty.

Modifying and Saving Data to Create a New Plan



Settings for medrec

Overview | Deployment Plan | **Configuration** | Secu

Monitoring | Notes

Application | Persistence | Workload | Instrumenta

Save

This page describes the configuration information that is the selected module or component.

Session Invalidation Interval (in seconds): 6090

Save

Save Deployment Plan Assistant

OK | Cancel

Save Deployment Plan

You have made configuration changes that need to be stored in a new deployment plan.

Select or enter the path of a deployment plan file. The path must end with '.xml'. It is highly recommended that this file be named 'Plan.xml'.

Each plan should be located in its own directory, otherwise applications can inadvertently share deployment plan files. The plan file will be overwritten if it already exists. Other files in the plan directory may be overwritten as well.

Path: /home/oracle/wls-sysadm/labs/Plan.xml

Recently Used Paths: /home/oracle/wls-sysadm/labs

Current Location: vx0399 / home / oracle / wls-sysadm / labs

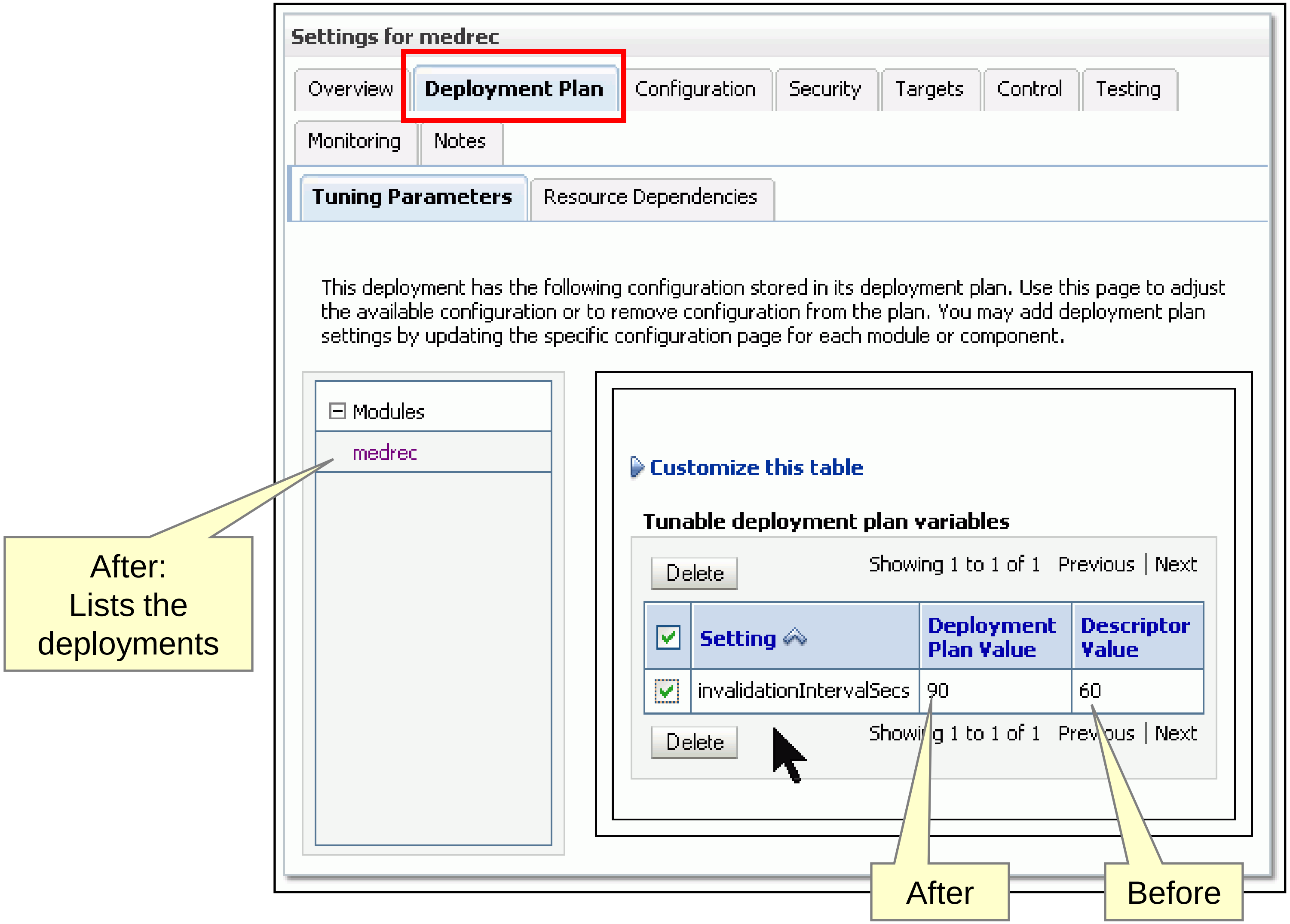
apps

medrec

sql

OK | Cancel

New Deployment Plan Shows Changed Values



This deployment has the following configuration stored in its deployment plan. Use this page to adjust the available configuration or to remove configuration from the plan. You may add deployment plan settings by updating the specific configuration page for each module or component.

Tunable deployment plan variables

☑	Setting	Deployment Plan Value	Descriptor Value
☑	invalidationIntervalSecs	90	60

After:
Lists the deployments

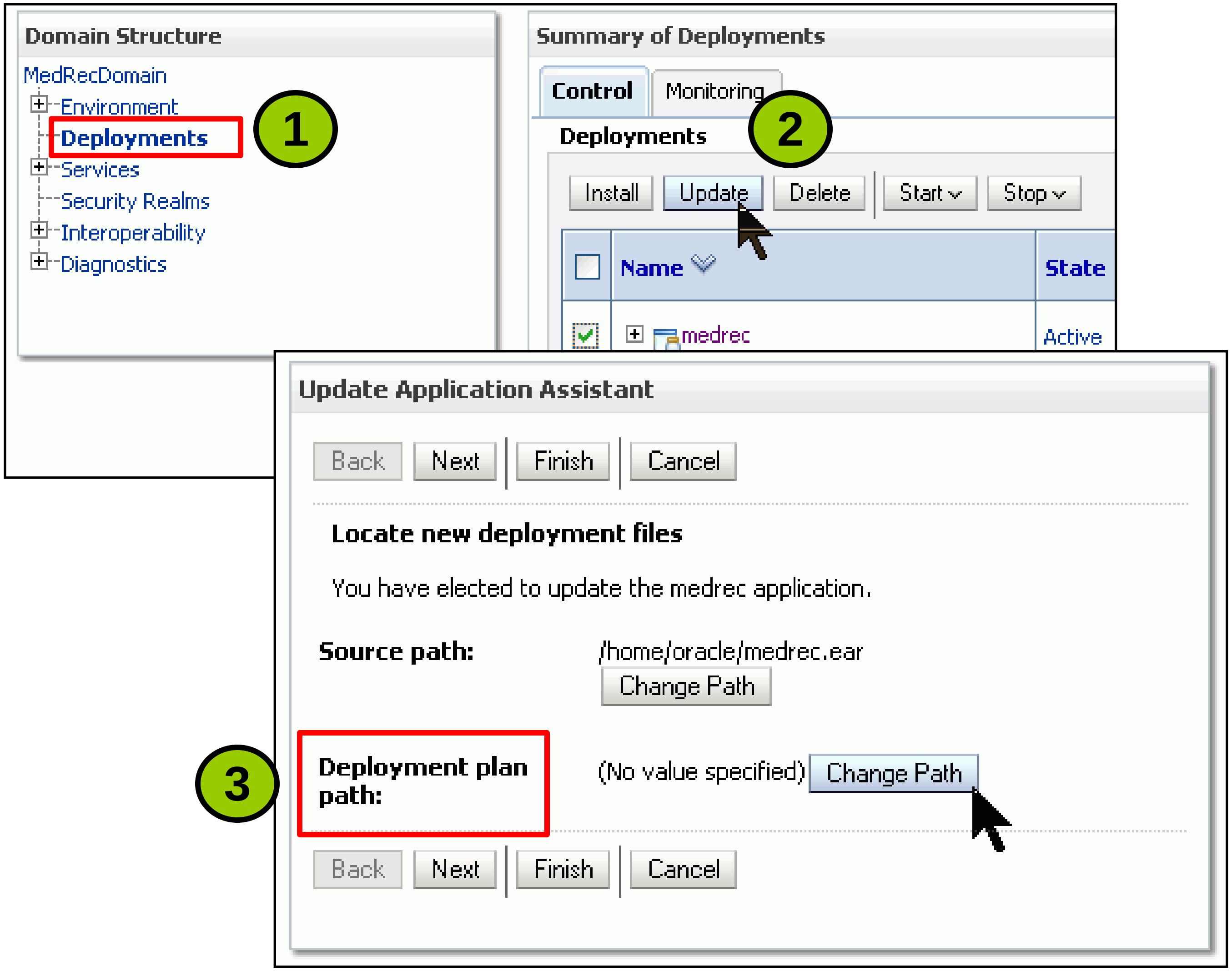
After

Before

Using an Existing Deployment Plan to Configure an Application

1. Prepare the application.
2. Place the existing deployment plan in the plan subdirectory of the application root.
3. Install the application.
   - The Administration Console validates the deployment plan configuration against the target servers and clusters that are selected during the installation.
4. Use the Administration Console or the `weblogic.Deployer` utility to identify both the application and the plan to use for deployment.

Using an Existing Deployment Plan



Locate new deployment files

You have elected to update the medrec application.

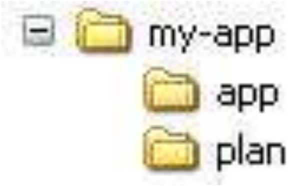
Source path: /home/oracle/medrec.ear

Deployment plan path: (No value specified)

Generic File-Loading Overrides

- Place application-specific files to be overridden into a new optional subdirectory (named `AppFileOverrides`) in the existing plan directory structure.
- The presence or absence of this new optional subdirectory controls whether file overrides are enabled for the deployment.
- If this subdirectory is present, an internal `ClassFinder` is added to the front of the application and module `ClassLoaders` for the deployment.
- The file override hierarchy rules follow the existing `ClassLoader` and resource-loading rules and behaviors for applications.

Directory Structure for Easier Production Deployment

Directory Tree	Description
 my-app	Application root
app	Application deployment files (archive or exploded)
plan	`plan.xml`

This allows the deployment configuration files to be located in a well-known location.

Performing a Sanity Check in Production Without Disruption to the Clients

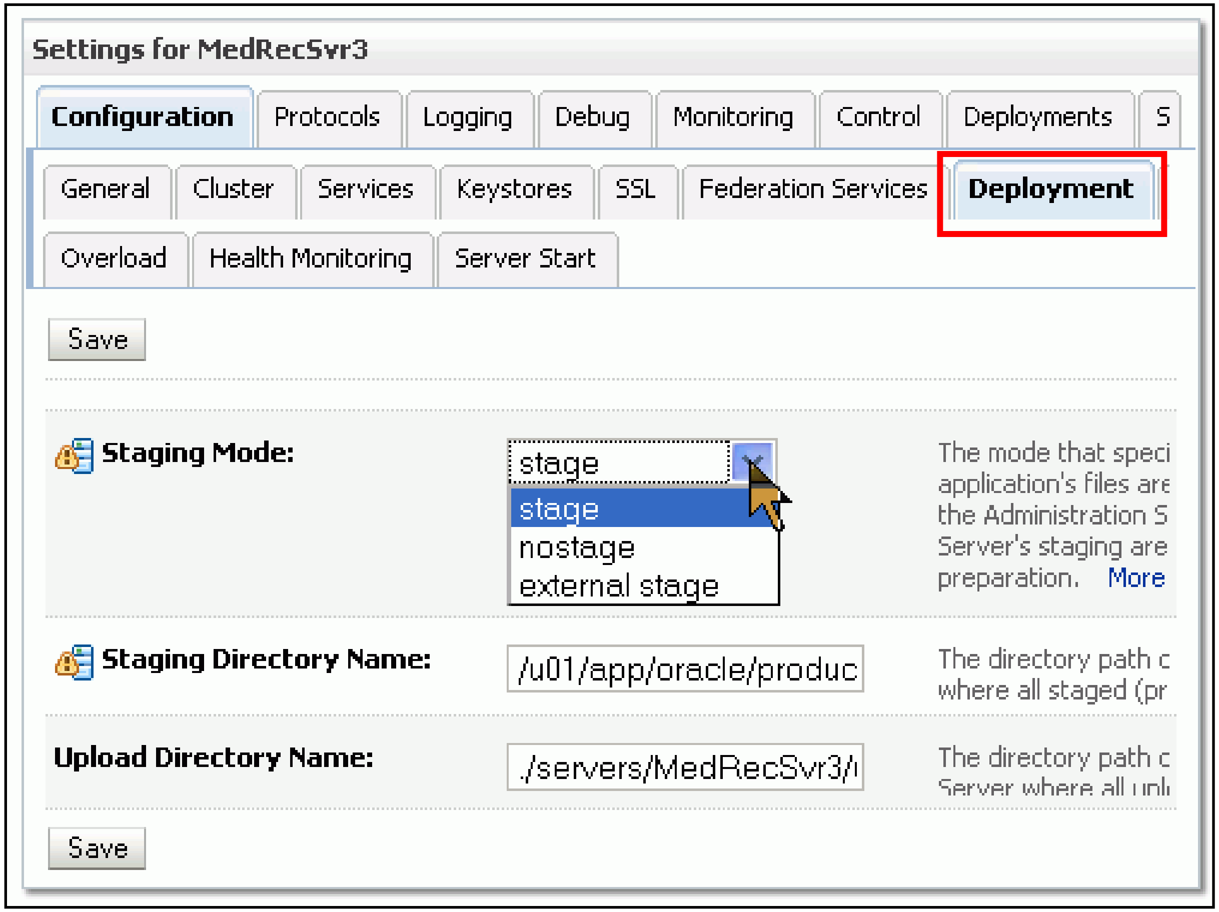
- Using Administration mode, administrators can deploy an application into a production environment without exposing the application to external clients.
- Access to the application is restricted to a configured administration channel.
- You can perform a final (“sanity”) check on the application directly in the production environment without disruption to the clients.

Road Map

- Deployment plans
- Staged deployment
- Production redeployment

Staged Deployment

You can configure deployment per server or for each application.



Settings for MedRecSvr3

Staging Mode: stage

The mode that speci application's files are the Administration S Server's staging are preparation. More

Staging Directory Name: /u01/app/oracle/produc

The directory path c where all staged (pr

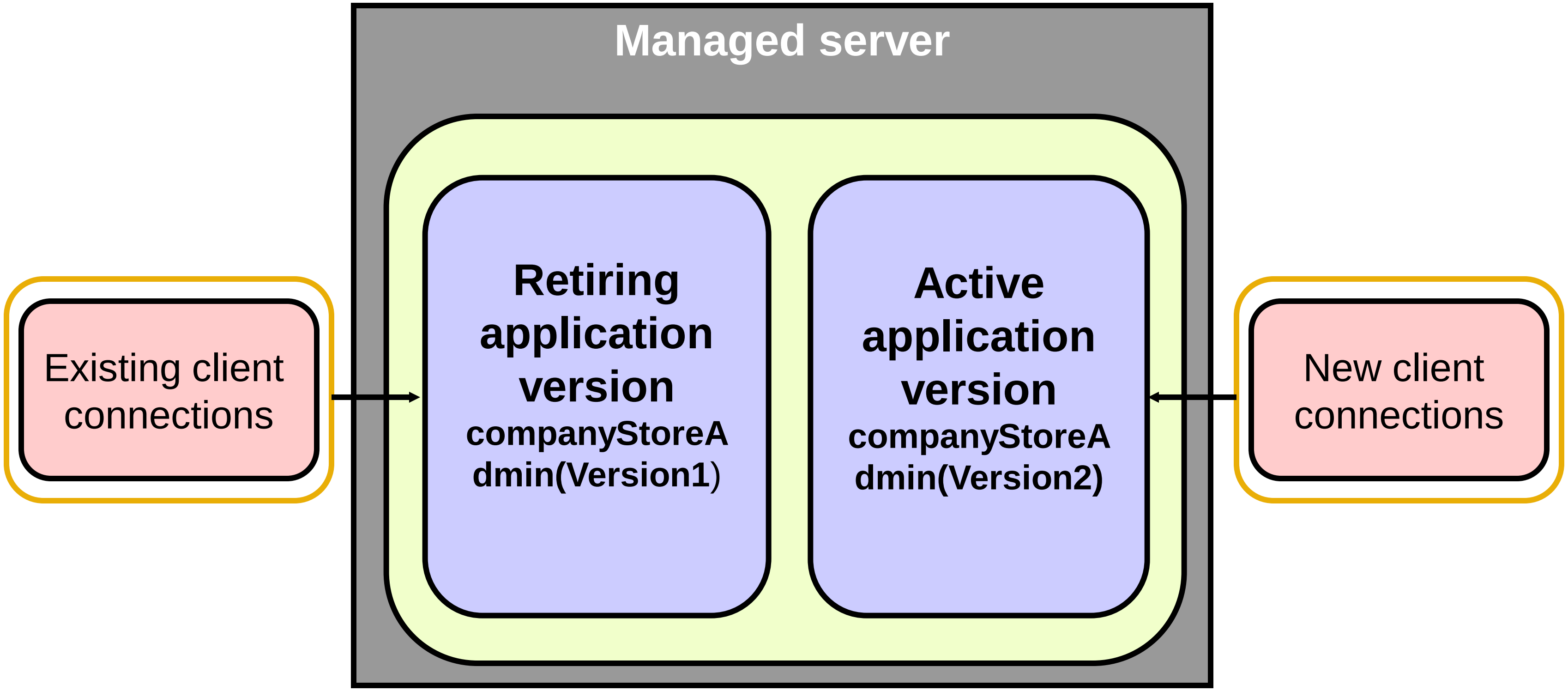
Upload Directory Name: ./servers/MedRecSvr3/

The directory path c Server where all unl

Road Map

- Deployment plans
- Staged deployment
- Production redeployment

Application Availability


Managed server
Existing client connections
Retiring application version companyStoreAdmin(Version1)
Active application version companyStoreAdmin(Version2)
New client connections

Production Redeployment and Application Versioning

- You can redeploy a revised version of a production application alongside the older version:
  - Without affecting the existing clients to the application
  - Without interrupting the availability of the application to the new client request
- Oracle WebLogic Server automatically manages client connections so that:
  - Existing clients continue to use the older application
  - New client requests are directed to the newer application
- The older version is retired and then undeployed after all current clients complete their work.

WebLogic Production Redeployment

- Production redeployment:
  - Enables two versions of a single Web application or module to run simultaneously
  - Requires you to include unique version information either:
    - Within the application's `META-INF/MANIFEST.MF` file
    - As part of the deployment process
- When a new version is redeployed, WLS automatically:
  - Routes existing clients to the prior (retired) version
  - Routes new clients to the new version
  - Undeploys the prior version when all existing clients finish their work or their conversations time out

Production Redeployment

- To support the production redeployment strategy, Oracle WebLogic Server now recognizes a unique version string entry in the Enterprise `MANIFEST` file.
- When a redeployment operation is requested, Oracle WebLogic Server checks the version string to determine whether to deploy a new version of the application.
- Production redeployment is performed automatically if:
  - An application supports production redeployment
  - Its deployment configuration is updated with changes to resource bindings
- This occurs even if no version string is specified in the application’s manifest file.

Advantages of Production Redeployment

Saves the trouble of:

- Scheduling application down time
- Setting up redundant servers to host new application versions
- Managing client access to multiple application versions manually
- Retiring older versions of an application manually

Requirements and Restrictions for Production Redeployment

- Production redeployment strategy is supported for:
  - Stand-alone WAR modules and EARs whose clients access the application via HTTP
  - Enterprise applications that are accessed by inbound JMS messages from a global JMS destination or from inbound JCA requests
  - All types of Web services
- Production redeployment is *not* supported for:
  - Stand-alone EJB or RAR modules
  - Applications that use Java Transaction Service (JTS) drivers
  - Applications that obtain JDBC data sources via the DriverManager API
  - Applications that include EJB 1.1 container-managed persistence (CMP) EJBs

Redeploying a New Application Version

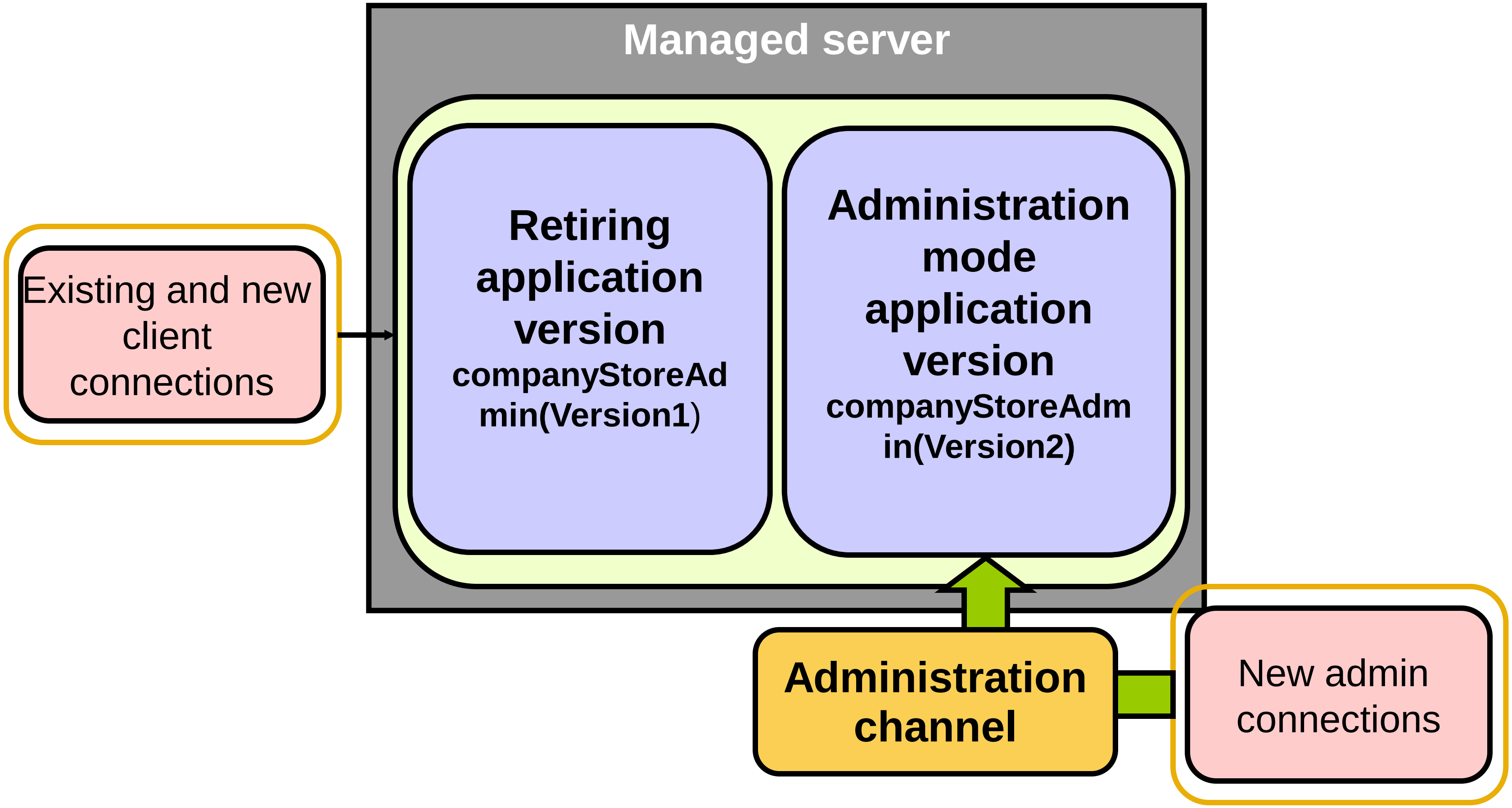
1. Verify that only one version of the application is currently deployed.
2. Verify the `MANIFEST.MF` files to ensure that both applications have different versions.
3. Copy the new version into a suitable directory.
4. Redeploy the new application version and specify the updated deployment files.
5. Verify that both versions are deployed and that new requests are being sent to the new version.

Redeploying Versus Distributing

Distributing is an alternative to deploying an application.

Distributing a new version of the application makes it available for testing before being released for general consumption.	**Redeploying** a new version of an application places the application immediately into use and makes it available to new client requests.

Distributing a New Version of the Production Application


Managed server
Retiring application version companyStoreAdmin(Version1)
Administration mode application version companyStoreAdmin(Version2)
Existing and new client connections
Administration channel
New admin connections

Distributing a New Application Version

1. Use the `weblogic.Deployer -distribute` command.
2. After the application is distributed, start the application in Administration mode.
3. Test the application.
4. When ready, start the application (without using `-adminmode`).
5. Optionally, set a retirement timeout for the older version of the application.

Production Redeployment

- Create `MANIFEST.mf` with at least the following contents:

```
Manifest-Version: 1.0
Weblogic-Application-Version: Version1.0Beta
```

- The value for WLS versioning has any text you choose, up to 215 characters, to indicate the version.

Quiz

Which of the following is NOT true about the deployment plans in Oracle WebLogic Server?

1. Overrides values in application descriptors
2. Can be created by Oracle WebLogic Server during deployment
3. Is packaged within an application archive
4. Is an XML file
5. Can be created with `weblogic.PlanGenerator`

Quiz

When an application is in the ____________ state, it is distributed to the servers, but is not yet available to clients:

1. Activated
2. Staged
3. Targeted
4. Prepared
5. Loaded

Quiz

Name four techniques or tools that can be used to deploy new applications to Oracle WebLogic Server.

1. Administration Console
2. WLST
3. `weblogic.PlanGenerator`
4. `weblogic.Deployer`
5. JMS
6. Autodeployment

Summary

In this lesson, you should have learned how to:

- Create deployment plans
- Stage deployments
- Perform production redeployment
- Configure an application for multiple development environments

Practice 12 Overview: Deploying Production Applications

This practice covers the following topics:

- Version-enabling an application
- Deploying a versioned application
- Monitoring the status of a versioned application
- Deleting the expired version of an application