Lab 02_01: Deploying Applications

Performance Checklist

Lab Overview: In this exercise, you will deploy a web application using the Management Console.

Lab Resources/Configuration:

Lab Files Location:	*LABS*/Lab02_01
Application URL:	`http://192.168.0.`*`xx`*`:8080/example`

Success Criteria: After completing this exercise, you will see the **`example.war`** application running on your server.

Outcome: A web application deployed onto your Standalone server.

Lab Outline:

1. Verify the Deployment Scanner
2. Upload the WAR File
3. Enable the Deployment
4. Disable the Deployment
5. Remove the Deployment

Before you begin...

Make sure you have EAP running in Standalone mode.

□ 1. Verify the Deployment Scanner is Enabled

□ 1.1. If you are not logged in already, login to your Management Console.

□ 1.2. On the **Profile** page, expand the **Core** section of the navigation tree in the left panel and click on the **Deployment Scanners** link.

□ 1.3. Is the deployment scanner currently enabled on your Standalone server?

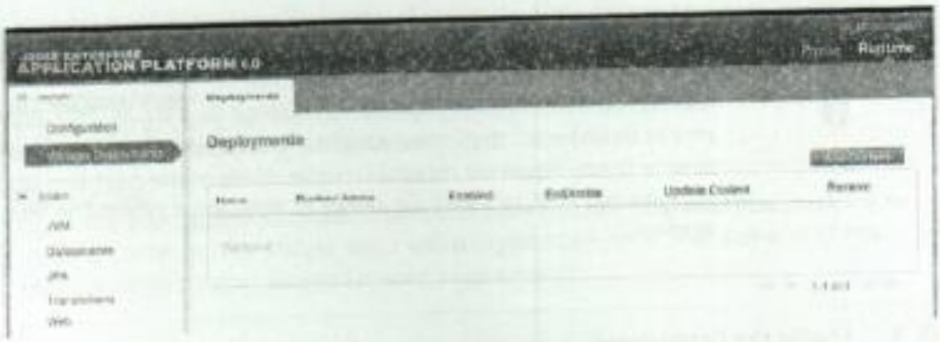
____

□ 1.4. How often does the Deployment Scanner scan for changes? _______________

□ 2. Upload the WAR File

□ 2.1. Now go to the **Runtime** page of the Management Console.

□ 2.2. In the **Server** section of the navigation tree, click on the **Manage Deployments** link. You do not have anything deployed yet, so the list of deployments will be empty.



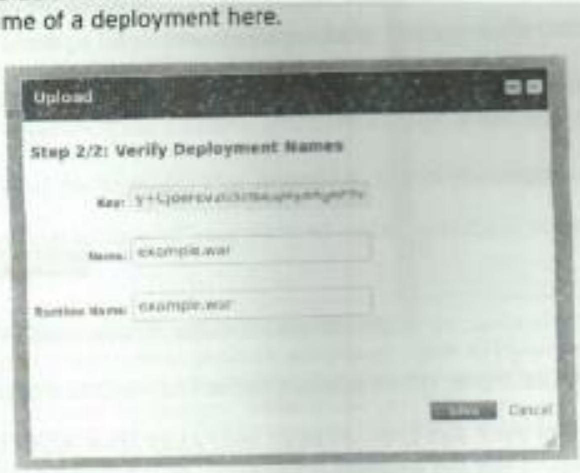
Deployments

Name	Runtime Name	Enabled	En/Disable	Update Content	Remove

□ 2.3. Click the **Add Content** button to add a new deployment.

□ 2.4. Click the **Browse...** button and select the file **`example.war`** in your ***LABS/*** **`Lab02_01`** folder. Click the **Next** button to go to step 2 of the wizard.

□ 2.5. In step 2, the default values are fine, but notice that you can change the name and runtime name of a deployment here.



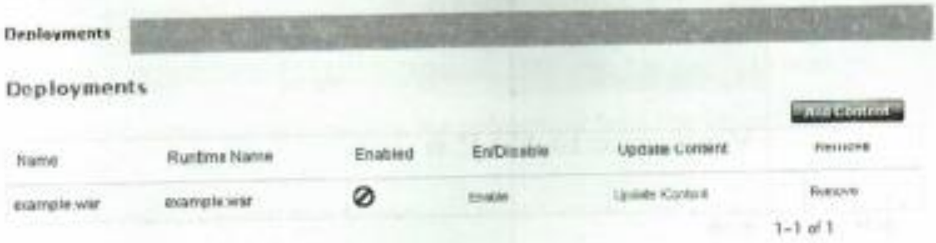
Step 2/2: Verify Deployment Names

Key: [illegible]

Name: example.war

Runtime Name: example.war

□ 2.6. Click the **Save** button to complete the wizard. You should now see **`example.war`** in the list of deployments.



Deployments

Name	Runtime Name	Enabled	En/Disable	Update Content	Remove
example.war	example.war	⊘	Enable	Update Content	Remove

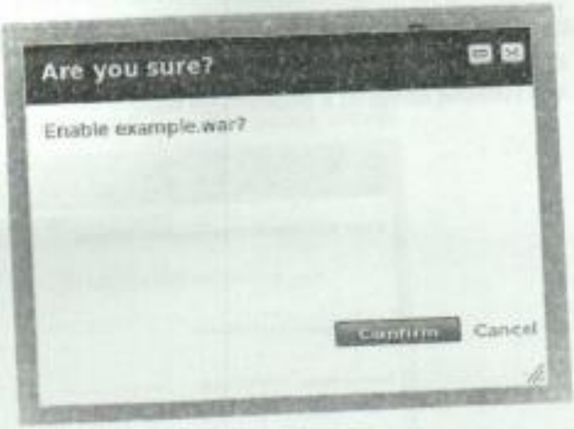
1-1 of 1

Insight

Notice the uploaded deployment is disabled by default. This allows you to define your deployments of an EAP server, then decide *when* to deploy them. When we discuss Domain mode in the next two Units, you will also see that this feature allows to determine *where* the files are deployed.

□ 3. Enable the Deployment

□ 3.1. Click the **Enable** link next to **`example.war`** in the list of deployments. On the confirmation screen, click the **Confirm** button.

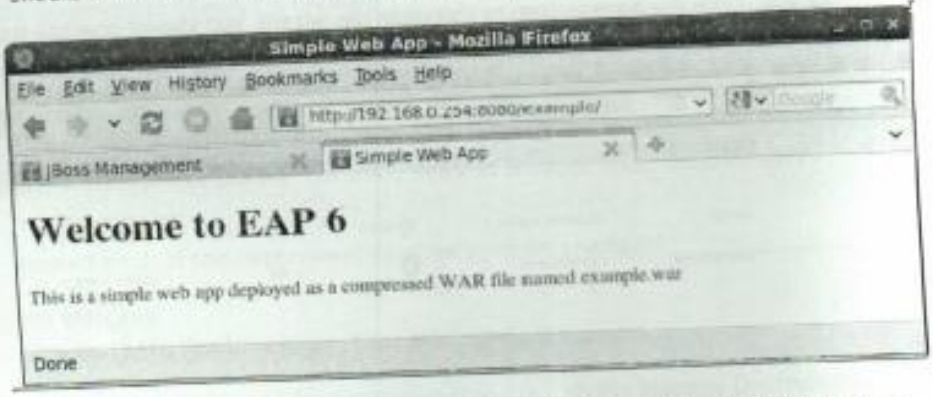


Are you sure?

Enable example.war?

□ 3.2. Notice a checkmark appears in the **Enabled** column next to **`example.war`**.

□ 3.3. Point your web browser to `http://192.168.0.xx:8080/example`. You should see the example application, which displays a simple web page.



Simple Web App - Mozilla Firefox

http://192.168.0.254:8080/example/

Welcome to EAP 6

This is a simple web app deployed as a compressed WAR file named example.war

Done

□ 3.4. Look in the terminal window of your running instance of EAP. You should see an output similar to:

```
INFO  [org.jboss.as.server.deployment] (MSC service thread 1-3)
 JBAS015876: Starting deployment of "example.war"
```

```
INFO  [org.jboss.as.osgi] (MSC service thread 1-4) JBAS011941: Register
 module: Module "deployment.example.war:main" from Service Module Loader
INFO  [org.jboss.web] (MSC service thread 1-5) JBAS018210: Registering web
 context:/example
INFO  [org.jboss.as.server] (HttpManagementService-threads - 7)
 JBAS018559: Deployed "example.war"
```

□ 3.5. Open the ***EAP_HOME*/standalone/configuration/standalone.xml** file to view its contents. You should see a **<deployments>** section at the end of this file that contains your **example.war** deployment:

```
<deployments>
    <deployment name="example.war" runtime-name="example.war"
 enabled="true">
        <content sha1="cbe0a3a1e144bf6d37dba6cba8ccbc46000fef39"/>
    </deployment>
</deployments>
```

□ 4. Disable the Deployment

□ 4.1. Go back to the **Manage Deployments** page of the Management Console.

□ 4.2. Click on the **Disable** link next to **example.war** to undeploy the application from your Standalone server. Click **Confirm** when the confirmation window appears.

□ 4.3. Notice in the list of deployments that **example.war** is no longer enabled.

□ 4.4. Look in the terminal window of your running instance of EAP. You should see an output similar to:

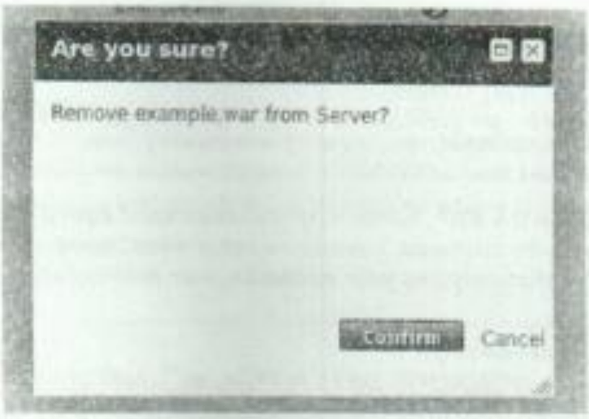
```
INFO  [org.jboss.as.osgi] (MSC service thread 1-2) JBAS011943: Unregister
 module: Module "deployment.example.war:main" from Service Module Loader
INFO  [org.jboss.as.server.deployment] (MSC service thread 1-7)
 JBAS015877: Stopped deployment example.war in 76ms
INFO  [org.jboss.as.server] (HttpManagementService-threads - 8)
 JBAS018558: Undeployed "example.war"
```

□ 4.5. Try reloading the URL `http://192.168.0.xx:8080/example` in your browser. You should get a 404 error.

□ 5. Remove the Deployment

□ 5.1. In the list of deployments, click the **Remove** button next to **example.war**.

□ 5.2. Click the **Confirm** button to remove the deployment from the Server:



Are you sure?

Remove example.war from Server?

□ 5.3. You have now removed the **example.war** from the Server, and it is no longer available for deployment. This lab demonstrated how to deploy an application onto a Standalone server using the Management Console. In the next lab, you will learn a manual technique for deploying applications onto a Standalone server.

Lab 02_02: Deploying Applications Manually

Performance Checklist

Manually Deploy an Application

Lab Overview:

In this exercise, you will deploy a web application manually using marker files.

Lab Resources/Configuration:

Lab Files Location:	*LABS*/Lab02_02
Application URL:	http://192.168.0.*xx*:8080/version

Success Criteria: After completing this exercise, you will see the **version.war** application running on your server.

Outcome: A web application deployed onto your Standalone server.

Lab Outline:

1. Deploy the Application
2. Create a **.dodeploy** Marker File
3. Redeploy an Application
4. Undeploy an Application
5. Using the **.skipdeploy** Marker File

□ 1. Deploy the Application

□ 1.1. Locate the folder ***LABS*/Lab02_02/version.war**, which represents a simple web application that displays the version of JBoss that it is deployed on. This is an example of an exploded application - the application is not in a single, compressed WAR file.

□ 1.2. Copy the folder **version.war** into your ***EAP_HOME*/standalone/deployments** folder.

□ 1.3. Look in the terminal window of your EAP instance and you should see the following output:

```
INFO  [org.jboss.as.deployment.scanner] (DeploymentScanner-threads -
 1) JBAS018559: Found version.war in deployment directory. To trigger
 deployment create a file called version.war.dodeploy
```

In the next step you will define this marker file.

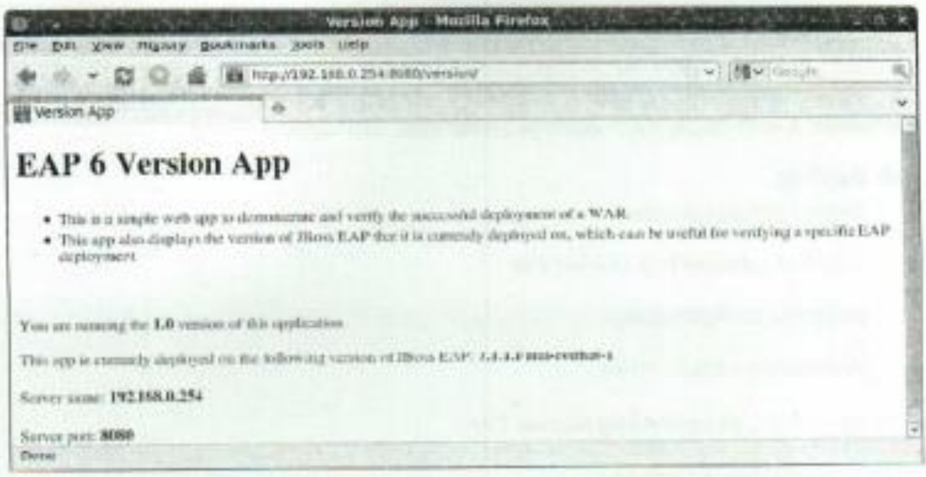
□ 2. Create a **.dodeploy** Marker File

□ 2.1. To cause EAP to deploy the exploded **version.war** application, you need to define a marker file that has the same name as your application. Using any technique you like, create a new, empty text file in the **standalone/deployments** folder named **version.war.dodeploy**.

□ 2.2. To verify your marker file was successful, check the terminal window of EAP. You should see something similar to:

```
INFO  [org.jboss.as.server.deployment] (MSC service thread 1-3)
 JBAS015876: Starting deployment of "version.war"
INFO  [org.jboss.as.osgi] (MSC service thread 1-2) JBAS011941: Register
 module: Module "deployment.version.war:main" from Service Module Loader
INFO  [org.jboss.web] (MSC service thread 1-7) JBAS018210: Registering web
 context: /version
INFO  [org.jboss.as.server] (DeploymentScanner threads - 2) JBAS018559:
 Deployed "version.war"
```

□ 2.3. Point your web browser to `http://192.168.0.xx:8080/version`. You should see a web page similar to the following:



Version App - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.254:8080/version/
Google
Version App

EAP 6 Version App

- This is a simple web app to demonstrate and verify the successful deployment of a WAR.
- This app also displays the version of JBoss EAP that it is currently deployed on, which can be useful for verifying a specific EAP deployment.

You are running the **1.0** version of this application

This app is currently deployed on the following version of JBoss EAP: [illegible]

Server name: **192.168.0.254**

Server port: **8080**

□ 2.4. Go to the **Runtime** page of the Management Console and click on the **Manage Deployments** link. You should see **`version.war`** in the list of deployed applications.

□ 3. Redeploy an Application

In this step, you will modify the **`index.jsp`** page of the **`version`** application and redeploy the changes.

□ 3.1. Using a text editor, open the file **`standalone/deployments/version.war/index.jsp`**.

□ 3.2. On line 56, change **`<b>1.0</b>`** to **`<b>2.0</b>`**.

□ 3.3. Save your changes to **`index.jsp`**.

□ 3.4. Notice in the terminal window of EAP that the **`version.war`** application was not automatically redeployed.

□ 3.5. Go to your web browser and refresh the page `http://192.168.0.xx:8080/version`. You should still version 1.0 of the application.

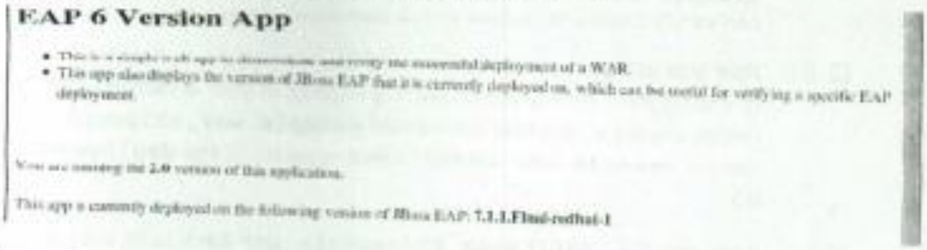
□ 3.6. To redeploy **version.war**, you need to change the timestamp on the file **standalone/deployments/version.war.deployed**. On Windows, open this file in a text editor, then save it. (You may need to modify the contents first, but the content of the file is ignored so make any change you like.) On RHEL, open a terminal window in the **deployments** folder and enter the command:

```
touch version.war.deployed
```

□ 3.7. In your EAP terminal window, you should the **version.war** app is redeployed:

```
INFO [org.jboss.as.osgi] (MSC service thread 1-1) JBAS011943: Unregister
 module: Module "deployment.version.war:main" from Service Module Loader
INFO [org.jboss.as.server.deployment] (MSC service thread 1-1) JBAS015877:
 Stopped deployment version.war in 36ms
INFO [org.jboss.as.server.deployment] (MSC service thread 1-5) JBAS015876:
 Starting deployment of "version.war"
INFO [org.jboss.as.osgi] (MSC service thread 1-5) JBAS011941: Register
 module: Module "deployment.version.war:main" from Service Module Loader
INFO [org.jboss.web] (MSC service thread 1-4) JBAS018210: Registering web
 context: /version
INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS018562:
 Redeployed "version.war"
```

□ 3.8. Refresh the **http://192.168.0.xx:8080/version** page in your web browser, and you should now see version 2.0 of this application:



EAP 6 Version App

- This is a simple web app to demonstrate and verify the successful deployment of a WAR.
- This app also displays the version of JBoss EAP that it is currently deployed on, which can be useful for verifying a specific EAP deployment.

You are running the 2.0 version of this application.

This app is currently deployed on the following version of JBoss EAP: **7.1.1.Final-redhat-1**

□ 4. Undeploy an Application

□ 4.1. In your **standalone/deployments** folder, delete the file **version.war.deployed**.

□ 4.2. Within a few seconds, the Deployment Scanner will undeploy the **version.war** app and create a new marker file named **version.war.undeployed** in the **deployments** folder.

□ 4.3. In your EAP terminal window, you should see that the **version.war** application was undeployed:

```
INFO [org.jboss.as.osgi] (MSC service thread 1-3) JBAS011943: Unregister
 module: Module "deployment.version.war:main" from Service Module Loader
INFO [org.jboss.as.server.deployment] (MSC service thread 1-6) JBAS015877:
 Stopped deployment version.war in 40ms
INFO [org.jboss.as.server] (DeploymentScanner-threads - 2) JBAS018558:
 Undeployed "version.war"
```

□ 4.4. You can also verify the application is undeployed by pointing your web browser to `http://192.168.0.xx:8080/version` - you should get a 404 error.

□ 4.5. Refresh the **Deployments** page of the Management Console. The **`version.war`** application should no longer appear on the list.

□ 5. Using the .skipdeploy Marker File

In this step, you will use a marker file to denote that an application should not auto-deploy even though it is a compressed WAR file in the **`deployments`** folder.

□ 5.1. On the **Manage Deployments** page of the Management Console, remove the **`example.war`** application (if it appears on the list of deployments).

□ 5.2. Copy the file ***`LABS`*`/Lab02_01/example.war`** into the ***`EAP_HOME`*`/standalone/deployments`** folder.

□ 5.3. Look in the terminal window of your EAP server. You should see that the **`example.war`** application deployed. Why did it deploy automatically, but the **`version.war`** application did not?

□ 5.4. Refresh the **Deployments** page. The **`example.war`** application should appear in the list of deployments.

□ 5.5. Go back to your **`standalone/deployments`** folder and delete the file **`example.war`**, along with the marker file. This undeploys **`example.war`**, as you can verify from the output in the terminal window of your EAP server.

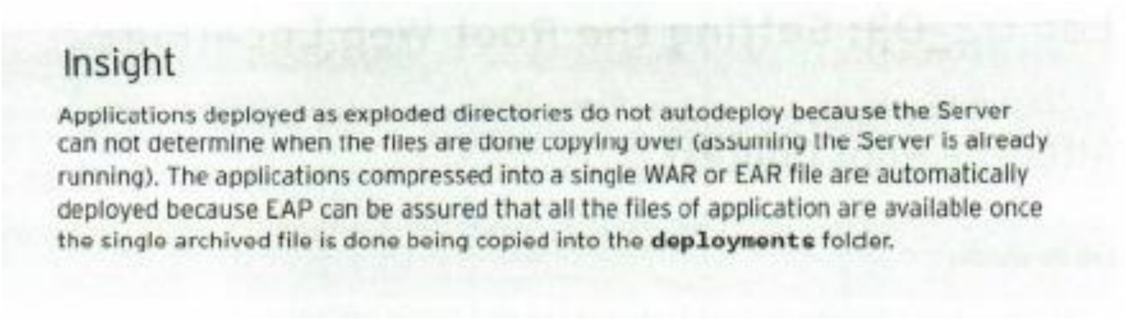
□ 5.6. Now you are going to deploy **`example.war`** again, but this time you are going to configure it to not automatically deploy. In your **`standalone/deployments`** folder, create a new text file named **`example.war.skipdeploy`**. (If a marker file named **`example.war.undeployed`** appears in the **`deployments`** folder, delete it.)

□ 5.7. Copy the file ***`LABS`*`/Lab02_01/example.war`** back in to your **`standalone/deployments`** folder.

□ 5.8. Check the output in the EAP terminal window. The **`example.war`** application should not have been deployed. In fact, no output should have appeared in the terminal window when you copied **`example.war`** into the **`deployments`** folder. What you have just done is push out a WAR file to the **`deployments`** folder, but the app is not deployed until you are ready to deploy it, even though the Deployment Scanner is enabled.

□ 5.9. To deploy the **`example.war`** application, rename the **`example.war.skipdeploy`** file to **`example.war.dodeploy`**.

□ 5.10. Verify the **`example.war`** app is now deployed by pointing your browser to `http://192.168.0.xx:8080/example`. The application should also appear in the list on the **Deployments** page of the Management Console.



Insight

Applications deployed as exploded directories do not autodeploy because the Server can not determine when the files are done copying over (assuming the Server is already running). The applications compressed into a single WAR or EAR file are automatically deployed because EAP can be assured that all the files of application are available once the single archived file is done being copied into the **deployments** folder.

Lab 02_03: Setting the Root Web Location

Performance Checklist

Alter the Root Context

Lab Overview:

In this exercise, you will alter the root context of server.

Lab Resources/Configuration:

Application URL:	`http://192.168.0.`*`xx`*`:8080/`

Success Criteria: After completing this exercise, you will see the **`example.war`** application running on your server at the root context.

Outcome: A web application deployed onto your Standalone server in the root context.

Lab Outline:

1. Turn Off The Welcome File
2. Edit the Root Context
3. Test the Root Context

□ 1. Turn Off The Welcome File

□ 1.1. Open the ***EAP_HOME*/standalone/configuration/standalone.xml** file, and locate the **`<virtual-server>`** tag for the **`default-host`**.

□ 1.2. Change the value of **`enable-welcome-root`** from **`true`** to **`false`**.

Insight

Why can't I do this from the Management Console? In the initial release of EAP 6, this setting is not accessible from the Management Console, though it is accessible from the CLI, which we will introduce later on.

□ 1.3. Save your settings, then restart your server, and navigate to `http://192.168.0.`*`XX`*`:8080/`. You should see an error that the webpage is not available, the dreaded 404.

□ 2. Edit the Root Context

□ 2.1. Open the standalone.xml file again, and locate the same **`virtual-server`** tag as before.

□ 2.2. Add the following bold code:

```
<virtual-server name="default-host" enable-welcome-root="false" default-web-
module="example">
```

Important

The name "default-web-module" can be confusing. This does not refer to the WAR file, it refers to the deployed context of the application. In our case, "example" is the context, because we accessed this applcation via `http://192.168.0.XX/example`.

□ 2.3. Save the file, and restart your server.

Insight

Why can't I do this from the Management Console? In the initial release of EAP 6, there is a bug which adds extraneous quotes around the context name when attempting to edit it via the GUI. To see this for yourself, in the Management Console, go to the **Servlet/HTTP** section under **Web**, and click on the **Virtual Servers** tab. Edit the **Default Module**, and you'll see the bug.

□ 3. Test the Root Context

Simply load the page at `http://192.168.0.XX:8080/`. You should see the example.war application.

Important

If you try to load the context explicitly now, it will fail with a 404!

At this point, it is suggested that you roll back your changes.