

# Deploy applications on a standalone server

## Goals

After completing this section, a system administrator should be able to do the following:

- Deploy a Java EE application to a server instance running as a standalone server.

## Implementations section of standalone.xml

The `<deployments>` section of the `standalone.xml` lists the applications deployed to the standalone server. The `<deployments>` element contains a single dependent element with the name `<deployment>`.



### Important

Remember that applications are deployed using the EAP administration tools, so you should not manually add these entries to the XML.

Here is an example of the `<deployments>` section:

```
<deployments>
  <deployment name="bookstore.war" runtime-name="bookstore.war"> <content
    sha1="e1e57cb8b89371794d6c7e80baeb8bf0e3da4fc"/>
  </deployment>
  <deployment name="example.war" runtime-name="example.war" enabled="false">
    <content sha1="0a07b224819ce516b231b1afba0eadc45b272298"/>
  </deployment>
  <deployment name="version.war" runtime-name="version">
    <fs-exploded path="deploying-filesystem/version.war" relative-to="labs"/>
  </deployment> </
</deployments>
```

In the example above, three applications are deployed: `bookstore.war`, `example.war`, and `version.war`. The first two applications are managed by EAP and the third one was deployed using the unmanaged method.

The `<content>` element represents a strong hash value from the implementation file and is used internally as a unique identifier for the implementation.

Managed deployments are stored in the `JBOSS_HOME/data/content` folder, which is the deployment cache folder. EAP creates a new folder whose name is defined by the first two characters of the SHA1 code. Inside this folder, another folder is created whose name is defined by the other characters of the SHA1 code.

## Chapter 3. Configuring scripts and deploying applications



### use

If the server is using a custom base directory, managed deployments are stored in the `BASE_DIR/data/content` folder.

For the implementations listed above, the following directory must be available:

```
[student@workstation labs]$ tree -d $JBoss_HOME/data/content/
standalone-instance/data/content/
├── 0a
│   └── 07b224819ce516b231b1afba0eadc45b272298
│       └── e1
│           └── e57cb8b89371794d6c7e80baeb8bf0e3da4fcf
└── 0a
```

Since the `version.war` application is unmanaged, you will not have the application file available in the content folder.



### Warning

Adding a `<deployment>` element to the `<deployments>` section of an EAP configuration should not be done manually. However, you can manually remove entries and restart the server to uninstall an application without any issues.

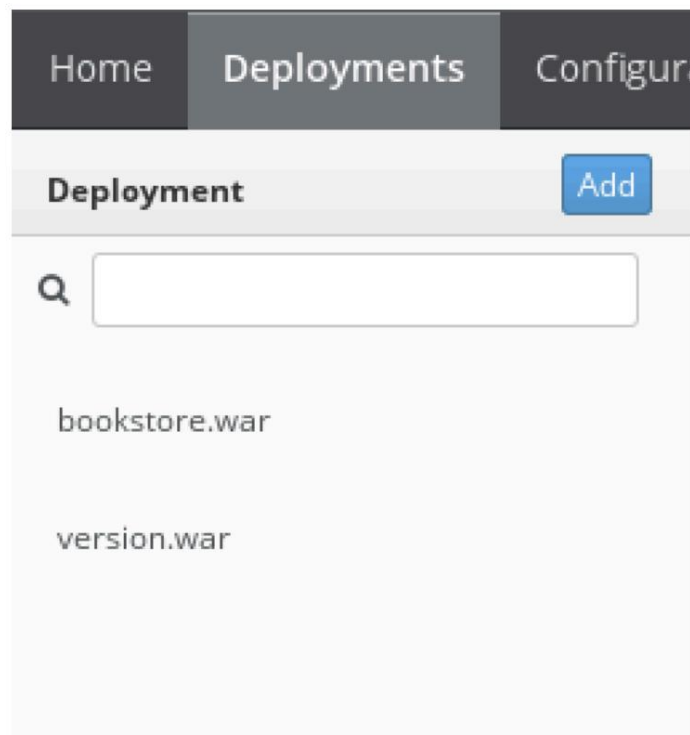
## Deploy apps with management tools

The deployment process is responsible for installing an application on JBoss EAP 7. This includes the copy of the package, as well as the configurations that need to be executed on the server.

There are three ways to deploy an application using a standalone server:

- The management console
- The CLI
- The file system implementer

**Deployment using the Admin Console**  
The Admin Console has a dedicated section for managing the deployments available on the standalone server. This can be accessed in the admin console by clicking the Deployments menu at the top of the page.

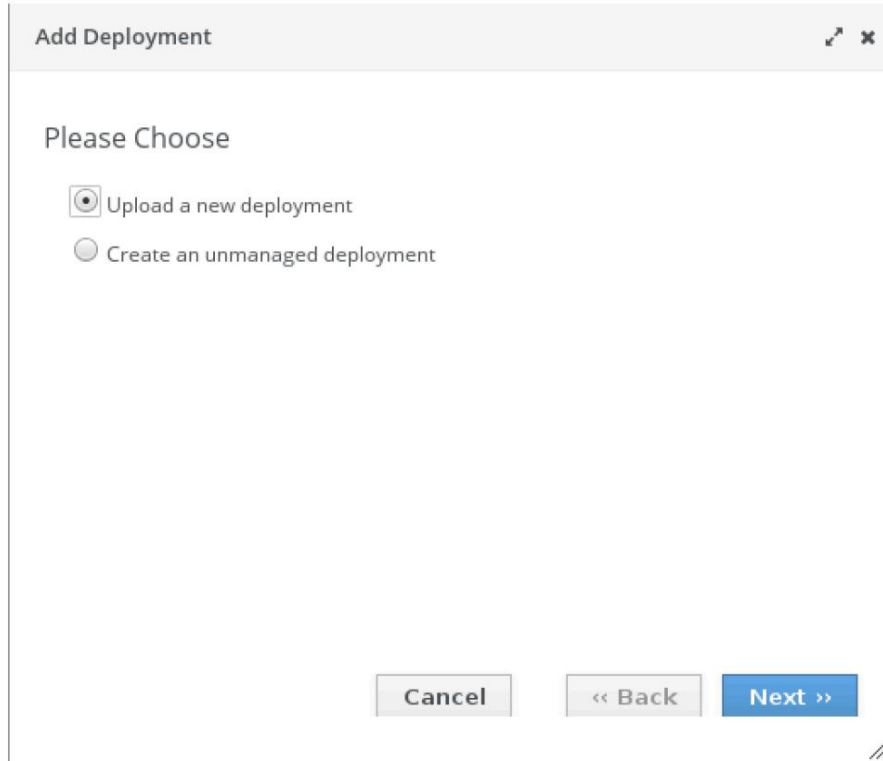


*Figure 3.1: Deployments view*

A new deployment can be created by clicking Add. A wizard will begin to deploy a new application. In the first step of the wizard, the deployment type is requested. Two options are available:

- **Load a new implementation** – Using this option should load an application, which will be available in the cache. A reference to the application will be created in `standalone.xml`.
- **Create an unmanaged deployment** – When using this option, you must specify a path where the application file is available. It is called unmanaged because EAP does not send the application file to the deployment cache, and any user who has permission to access the file can delete it, thus undeploying the application and making it unavailable on the server.

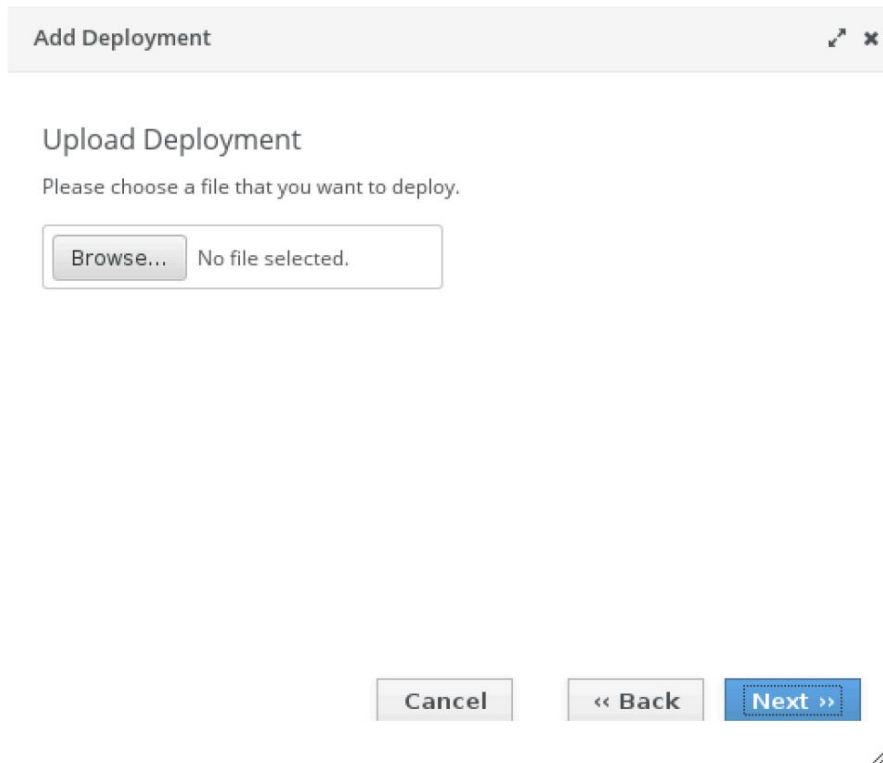
## Chapter 3. Configuring scripts and deploying applications



The image shows a dialog box titled "Add Deployment" with a close button (X) and a help icon (question mark). Below the title bar, the text "Please Choose" is displayed. There are two radio button options: "Upload a new deployment" (which is selected) and "Create an unmanaged deployment". At the bottom of the dialog, there are three buttons: "Cancel", "<< Back", and "Next >>".

**Figure 3.2: Options available for implementation**

When using the Upload a new implementation option, the second step will prompt you to upload the file. You can select the desired file by clicking Browse.



The image shows a dialog box titled "Add Deployment" with a close button (X) and a help icon (question mark). Below the title bar, the text "Upload Deployment" is displayed, followed by the instruction "Please choose a file that you want to deploy." Below this, there is a "Browse..." button and the text "No file selected." At the bottom of the dialog, there are three buttons: "Cancel", "<< Back", and "Next >>".

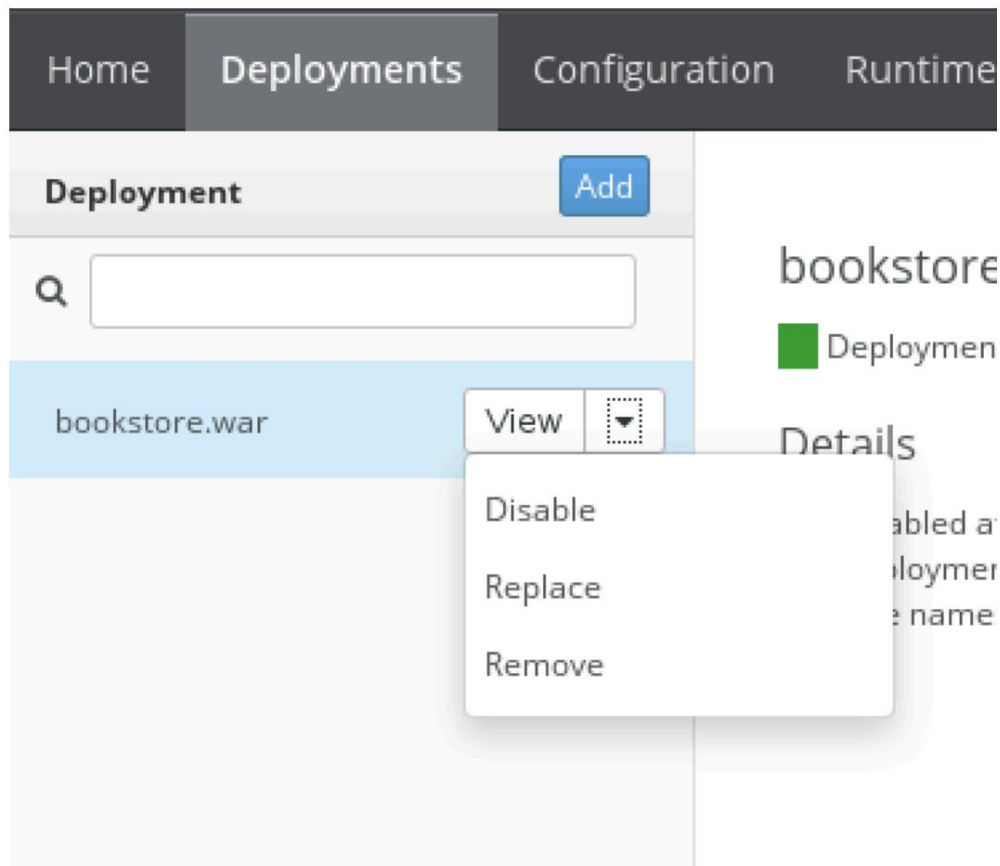
*Figure 3.3: Select the implementation file*

In the next step of the wizard, three options must be defined:

- **Name** – Identifies the implementation and must be a unique value across all implementations.
- **Runtime Name**: Defines the context of the application. The context is the name of the application in the runtime environment. If a deployment has a runtime name defined as myapp, it will be available at `http://server:port/myapp`.
- **Enabled** – Defines whether a deployment should start immediately. If I don't know check, it is possible to enable it later.

Enabling and disabling a deployment using the management console An application can be configured to start or not start at EAP 7 boot. Set the deployment to enabled if it should start during EAP 7 startup.

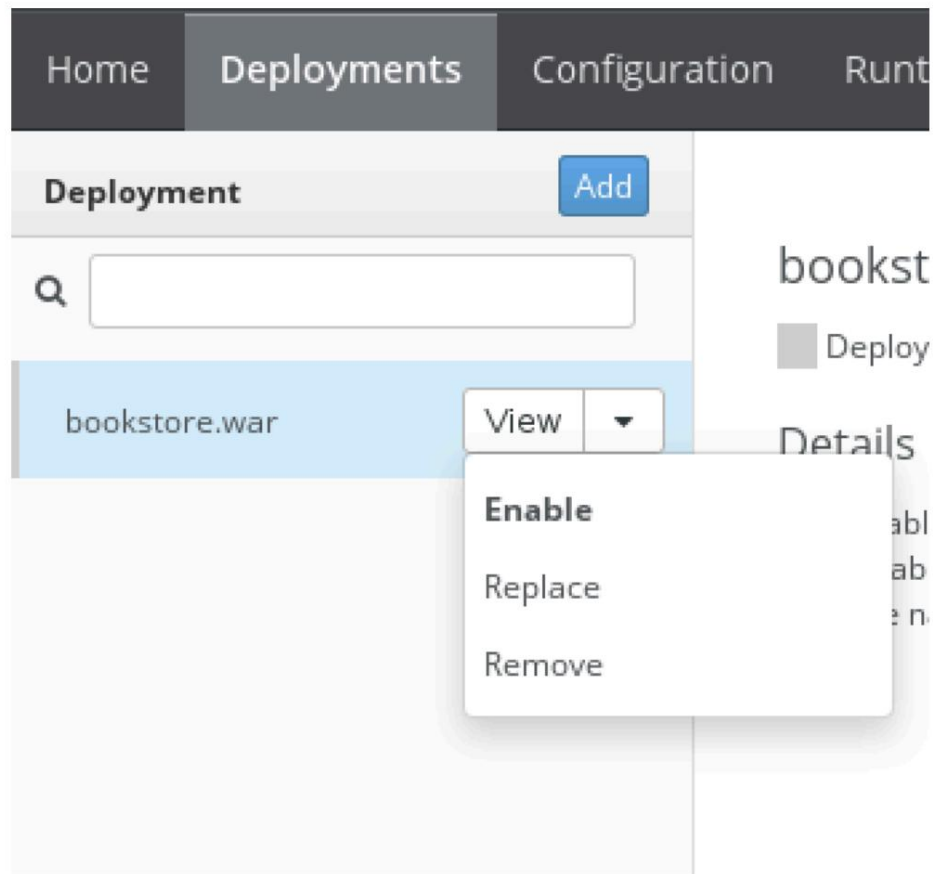
The management console enables and disables a deployment using the menu Implementations. To disable an app, click on it. A combo box should appear. Clicking the down arrow will bring up the Disable option:

*Figure 3.4: Disable a deployed application*

A confirmation screen should appear. Click Confirm and the deployment will be disabled.

## Chapter 3. Configuring scripts and deploying applications

To enable a deployment, click on the app and a combo box will appear again. Click the down arrow to see the Enable option.



*Figure 3.5: Enabling an application*

Undeploy an app using the admin console The admin console undeploy an app using the menu Implementations. To undeploy an app, click it. A combo box should appear. Clicking the down arrow will bring up the Delete option:

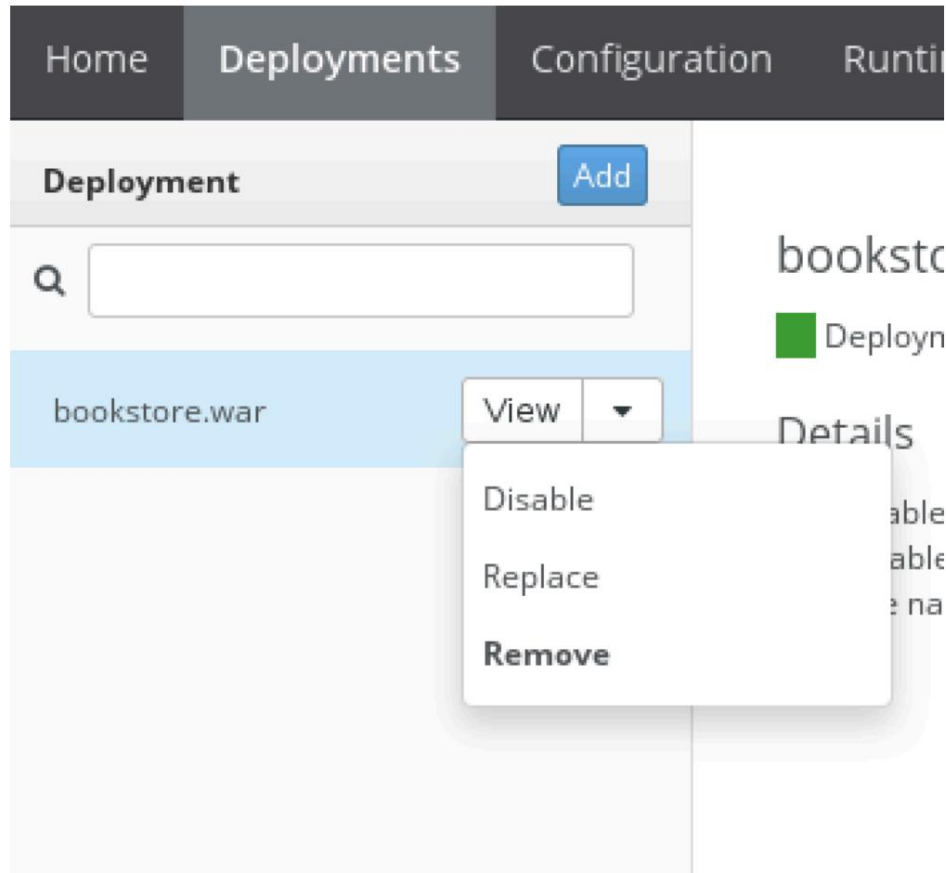


Figure 3.6: Undeploying a deployed application

A confirmation screen appears. Click Confirm and the application will be undeployed.



## use

Undeploying is not the same as disabling a deployment. Undeploying will uninstall the EAP application by removing it from the server, while disabling it will not start it, but the deployment will be available and startable without a new upload.

### Deployment using the CLI The CLI

provides the deploy command to start a new deployment. This command has arguments and the main ones are listed:

- **file\_path**: path to the application to be deployed.
- **--url**: URL at which the content of the implementation is available to load at the repository of the implementation content.
- **--name** – The unique name of the deployment. If no name is provided, at that case, the file name is used.
- **--runtime-name**: optional, defines the runtime name for the deployment.

## Chapter 3. Configuring scripts and deploying applications

- **--force:** If the implementation with the specified name already exists, so  
By default, the implementation is aborted and the corresponding message is printed. **--force (-f)** forces replacement of the existing implementation with the one specified in the command arguments.
- **--disabled :** Indicates that the implementation should be added to the repository in a state disabled.

To deploy an app located at `/home/student/myapp.war`, use the following command:

```
[standalone@localhost:19990 /] deploy /home/student/myapp.war --name=myapp.war
```

**Undeploy Using the CLI Tool** The CLI provides the `undeploy` command to undeploy a deployment. This command has several arguments and the most common are:

- **name** – The name of the application to be undeployed.
- **--keep-content** – Disable the deployment but do not remove its content from the repository.

```
[standalone@localhost:19990 /] undeploy myapp.war
```

## Deploy applications using the file system deployer

The filesystem implementer is a subsystem of EAP that will parse the JEE complaint application in a given directory. Now it is possible to execute operations in this type of implementation, using the administration tools, whereas in EAP 6, it was only possible to manage directly from the file system. For example, a deployment using the file system deployer can be disabled using the management console.

The standalone server supports deploying a new application manually. To start a manual deployment, it is only required to copy the application to the `JBOSS_HOME/deployments` folder.

The deployment process is managed using the marker files. A bookmark file is an empty file that uses the same name as the application, adding the bookmark file target to the end of its name. The marker file must be created in the deployments folder. The following marker file extensions can be used:

- **.dodeploy** – Created by the user. Triggers a new app deployment.
- **.deployed** – Created by the deployment scanner. Indicates that the application.
- **.isdeploying** – Created by the deployment scanner. Indicates that the application is being implementing.
- **.failed** – Created by the deployment scanner. Indicates that the application has failed.



## Demo: Deploy applications using the filesystem deployer

- **.isundeploying** – Created by the deployment scanner. Indicates that you are canceling the application implementation.
- **.undeployed** – Created by the deployment scanner. Indicates that the application implementation.
- **.skipdeploy**: Created by the user. Indicates that the application should not be deployed.
- **.pending** – Created by the deployment scanner. Indicates that you have noticed the need to implement content, but have not yet instructed the server to implement it.

To deploy an application named `myapp.war`, create the marker file `myapp.war.dodeploy`.

## Demo: Deploy applications using the filesystem deployer

1. Open a terminal window from the workstation virtual machine (Applications > Favorites > Terminal) and run the following command:

```
[student@workstation ~]$ demo deploying-filesystem setup
```

The previous command:

- It will verify that EAP is installed.
- Verify that you ran the Create a Standalone Server guided exercise.
- It will verify that EAP is not running.
- You will download the files required for this demonstration.

2. Run the following commands to start an EAP server using `/home/student/JB248/labs/standalone-instance` as the base directory:

```
[student@workstation standalone]$ cd /opt/jboss-eap-7.0/bin [student@workstation bin]$ ./standalone.sh \ -Djboss.server.base.dir=/home/student/JB248/labs/standalone-instance/
```

Remember that this base directory has set port-offset to 10000. It means that the management port is 19990 and the server port is 18080.

3. Ubique el directorio `/home/student/JB248/labs/deploying-filesystem/version.war`, which represents a simple web application displaying the version of JBoss it is deployed to. This is an example of an exploited application; the application is not in a single compressed WAR file.

It is important to understand that the application is not a static HTML page, and the expanded file is not a standard Java EE web application. This is a benefit provided by EAP 7.

4. Open a new terminal window and copy the `version.war` directory to your `/home/student/JB248/labs/standalone-instance/deployments` directory.

## Chapter 3. Configuring scripts and deploying applications

```
[student@workstation bin]$ cp -r \ /home/student/
JB248/labs/deploying-filesystem/version.war \ /home/student/JB248/labs/
standalone-instance/deployments
```

### 5. Look at the terminal window of the EAP instance; you should see the following output:

```
12:54:37,969 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner threads - 1) WFLYDS0004:
Found version.war in deployment directory. To trigger deployment create a file called version.war.dodeploy
```

A marker file is an empty file read by the deployment scanner subsystem to deploy new applications. It must have the same name as the deployed application, with a predefined extension. The next step is the definition of this marker file.

### 6. For EAP to deploy the exploited version.war application, you must create the marker file. Create a new empty text file in the /home/student/JB248/labs/standalone-instance/deployments folder with the name version.war.dodeploy.

```
[student@workstation bin]$ cd \ /home/
student/JB248/labs/standalone-instance/deployments [student@workstation
deployments]$ touch version.war.dodeploy
```

### 7. To verify that the marker file has been processed correctly by the deployment scanner subsystem, review the EAP terminal window. You should see something similar to the following:

```
12:55:48,100 INFO [org.jboss.as.server.deployment] (MSC service thread 1-2)
WFLYSRV0027: Starting deployment of "version.war" (runtime-name: "version.war")
...OUTPUT OMITTED...
12:55:51,237 INFO [org.jboss.as.server] (DeploymentScanner-threads - 2)
WFLYSRV0010: Deployed "version.war" (runtime-name : "version.war")
```

Check the files in /home/student/JB248/labs/standalone-instance/deployments/ and verify that the file name has changed. During deployment, the file is named version.war.deploying. When the file is finalized, the file name will be changed to version.war.deployed.

### 8. Abra el archivo /home/student/JB248/labs/standalone-instance/configuration/standalone.xml and verify that no entries have been made in the deployment-related file. You should see that the deployment tag is not available. Also, manual deployment does not store files in the managed cache. This can be verified by checking that the /home/student/JB248/labs/standalone-instance/data/content directory is empty.

### 9. Go to <http://localhost:18080/version>. You should see the sample app.

EAP 7 can deploy multiple applications and each will be available at <http://localhost:8080/<appname-without-the-war-extension>> by default. Because EAP started with a port offset from

## Demo: Deploy applications using the filesystem deployer

10000, estará disponible en `http://localhost:18080/<appname-without-the-war-extension>`.

10. Using the text editor, open the file `/home/student/JB248/labs/standalone-instance/deployments/version.war/index.xhtml`. It will modify the main page of the application version.
11. On line 25, change `<b>1.0</b>` to `<b>2.0</b>` and save the file. Observe in the EAP terminal window application `version.war` was not redeployed automatically. However, the page will update automatically when accessed through a web browser.
12. Go to your web browser and refresh the page `http://localhost:18080/version`. You should see version 2.0 of the app.
13. To redeploy the entire `version.war` application, you must change the timestamp in the `/home/student/JB248/labs/standalone-instance/standalone/deployments/version.war.deployed` file:

```
[student@workstation deployments]$ touch version.war.deployed
```

14. In the EAP terminal window, you should see that the application `version.war` has been reverted to implement.

```
15:10:38,010 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 62) WFLYUT0022: Unregistered web context: /version
... OUTPUT OMITTED...
15:10:39,288 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) WFLYSRV0013: Redeployed "version.war"
```

15. To undeploy the `version` application, delete the file `version.war.deployed`:

```
[student@workstation deployments]$ rm version.war.deployed
```

16. Within a few seconds, the deployment scanner will undeploy the application `version.war` and create a new marker file named `version.war.undeployed` in the `deployments` folder.
17. In the EAP terminal window, you should see that the `version.war` application has been uninstalled.

```
15:11:54,462 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 67) WFLYUT0022: Unregistered web context: /version
...OUTPUT OMITTED...
15:11:54,556 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) WFLYSRV0009: Undeployed "version.war" (runtime-name: "version.war")
```

This concludes the demo.