

# Deploying an Application to a Java Cloud Service

---

## Overview

### Purpose

This tutorial covers preparing an application for deployment and deploying it to a Java Cloud Service.

### Time to Complete

Approximately 30 minutes

### Introduction

This tutorial shows how to check a sample Java web application with the Whitelist tool. This Java utility ensures that no features unsupported by the Java Cloud Service are being used by an application. The tutorial then shows how to edit one of the application's deployment descriptors to ensure that the web application uses the previously provisioned Database Cloud Service. Finally, the tutorial shows how to deploy the application to a Java Cloud Service by using the Java Cloud Services Control console.

### Scenario

This tutorial assumes that you are an Oracle customer. It also assumes that you have completed the Oracle by Example tutorial titled *Signing Up for a Java Cloud Service*.

### Prerequisites

Before starting this tutorial, you should have:

- An **Oracle.com** account
- Already completed the Oracle by Example tutorial titled *Signing Up for a Java Cloud Service*
- Downloaded the sample Java web application archive file **visitors.war**. Download the file here: **visitors.war**.

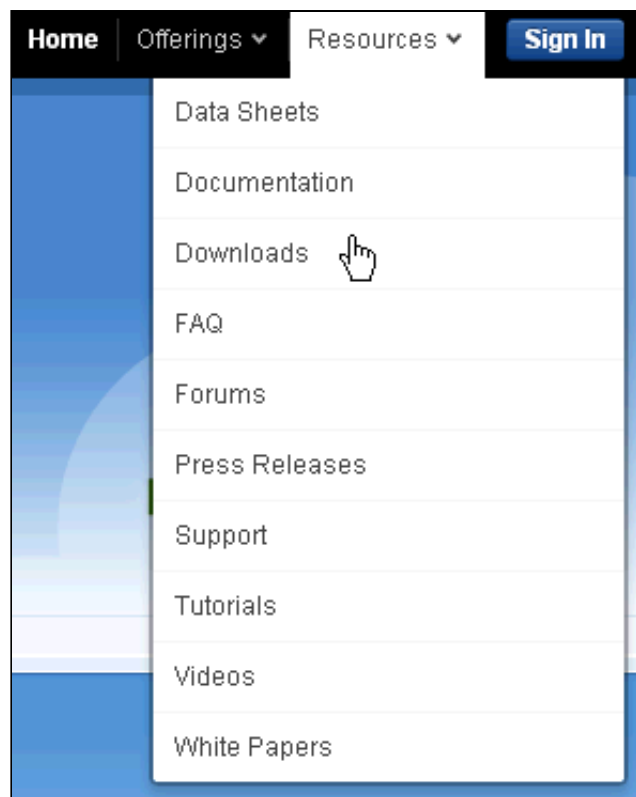
Note for Internet Explorer Users: IE may download **visitors.war** as **visitors.zip**. It is the same file, just rename it to **visitors.war**.

## Downloading and Unzipping the Java Cloud SDK

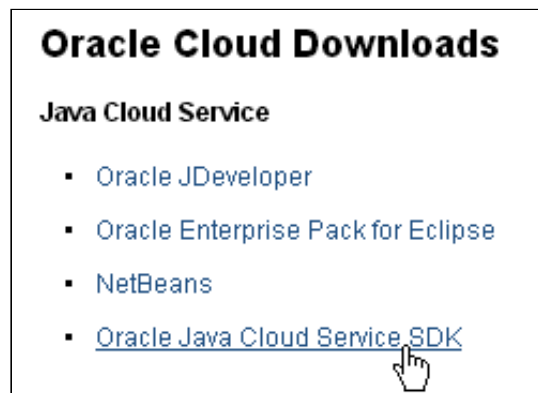
To download and unzip the Java Cloud SDK (which contains the Whitelist tool), perform the following steps:

1. You need to download the Java Cloud Software Development Kit (SDK). To download it, open a web browser to the Oracle Cloud home page:  
<https://cloud.oracle.com>.

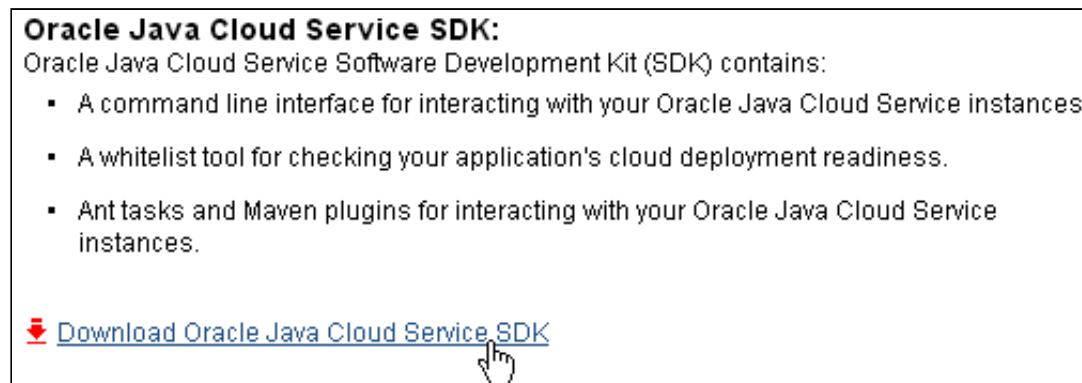
Once there, from the **Resources** drop-down list, select **Downloads**.

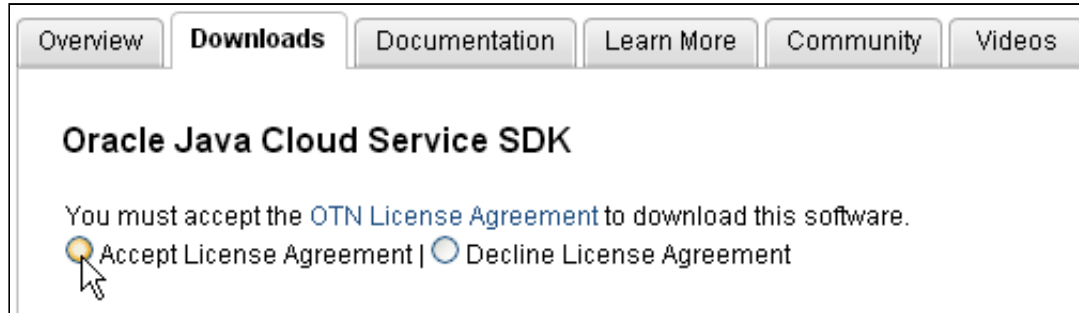


2. Under Oracle Cloud Downloads and Java Cloud Service, click the link for **Oracle Java Cloud Service SDK**.

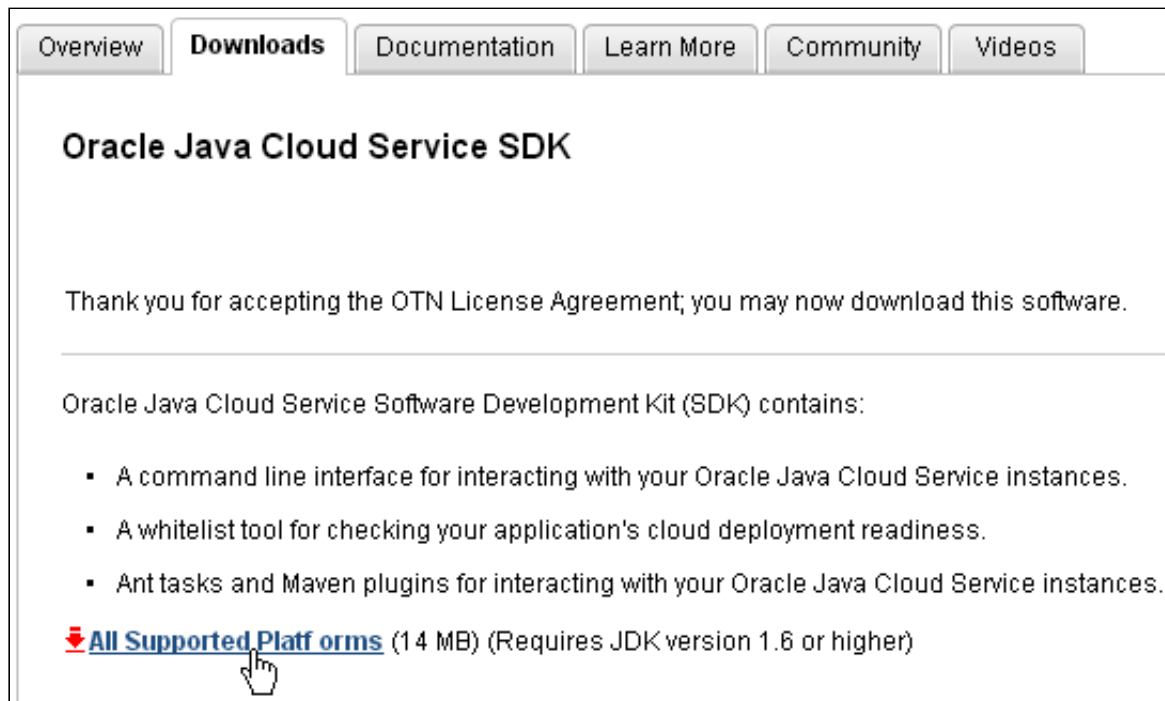


This jumps down to the Oracle Java Cloud Service SDK area. Click the link **Download Oracle Java Cloud Service SDK**.



**3. Next, select Accept License Agreement.**

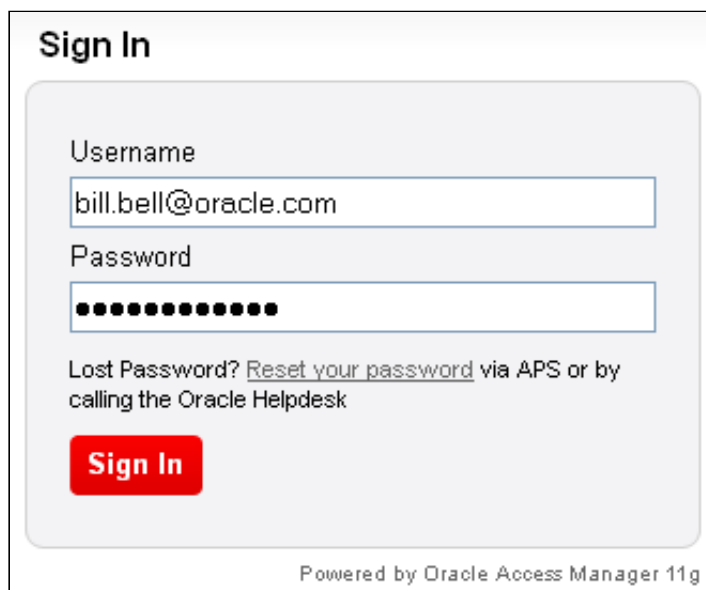
The screenshot shows the 'Downloads' tab of the Oracle Java Cloud Service SDK page. The title is 'Oracle Java Cloud Service SDK'. Below the title, it says 'You must accept the [OTN License Agreement](#) to download this software.' There are two radio buttons: 'Accept License Agreement' (which is selected) and 'Decline License Agreement'. A mouse cursor is pointing at the 'Accept License Agreement' radio button.

**4. Click the link to download the Oracle Java Cloud SDK to your local machine.**

The screenshot shows the 'Downloads' tab of the Oracle Java Cloud Service SDK page. The title is 'Oracle Java Cloud Service SDK'. Below the title, it says 'Thank you for accepting the OTN License Agreement; you may now download this software.' Below this, it says 'Oracle Java Cloud Service Software Development Kit (SDK) contains:' followed by a bulleted list:

- A command line interface for interacting with your Oracle Java Cloud Service instances.
- A whitelist tool for checking your application's cloud deployment readiness.
- Ant tasks and Maven plugins for interacting with your Oracle Java Cloud Service instances.

Below the list, there is a link: [All Supported Platforms](#) (14 MB) (Requires JDK version 1.6 or higher). A mouse cursor is pointing at the link.

**5. When the Sign In screen displays, log in with your Oracle.com Username and Password and click the Sign In button.**

The screenshot shows the 'Sign In' screen. It has a title 'Sign In'. Below the title, there are two input fields: 'Username' and 'Password'. The 'Username' field contains the text 'bill.bell@oracle.com'. The 'Password' field is masked with dots. Below the password field, there is a link: 'Lost Password? [Reset your password](#) via APS or by calling the Oracle Helpdesk'. At the bottom, there is a red button labeled 'Sign In'. At the very bottom, it says 'Powered by Oracle Access Manager 11g'.

6. Once the file has downloaded, unzip it on your local machine. The directory to which you unzip this file will be referred to as **<SDK\_HOME>** in the remainder of this tutorial.

Note: The **<SDK\_HOME>** directory chosen in this tutorial is **C:\oracle-javacloud-sdk**.

#### Testing an Application with the Whitelist Tool

There are a few Java packages, for example, **java.rmi**, that applications that are deployed to a Java Cloud Service are not allowed to use. The Whitelist tool is a Java utility that checks an application for disallowed packages and informs you whether the application can be deployed or not.

To test an application with the Whitelist tool, perform the following steps:

1. Open a command prompt and navigate to the **lib** directory under the **<SDK\_HOME>** directory.

```
C:\oracle-javacloud-sdk>cd lib
C:\oracle-javacloud-sdk\lib>
```

2. Run the following command to see help information on the Whitelist tool:  
**java -jar whitelist.jar -help**

```
C:\oracle-javacloud-sdk\lib>java -jar whitelist.jar -help
Mandatory argument(s):

Optional arguments(s):
log                - The path to log file where the scan result is to be written
                    Shortcut:l
grid               - The flag<true/false> that indicates if the warning,error
```

3. Now run the following command to check the sample web application archive file, **visitors.war**:  
**java -jar whitelist.jar -l whitelist.log <APP\_LOCATION>/visitors.war**

Note: In this tutorial, the **<APP\_LOCATION>** of the archive file **visitors.war** is the **applications** directory.

```
C:\oracle-javacloud-sdk\lib>java -jar whitelist.jar -l whitelist.log
                                     /applications/visitors.war
C:\oracle-javacloud-sdk\lib>
```

4. Notice this part of the command: **-l whitelist.log**. The file name given after the **-l** option ("l" stands for "log") will contain the output of the Whitelist utility.

Examine the log file in the current directory. This file will list any usage violations that need to be corrected. In general, you can ignore warnings. If there are no violations, you should see a message that the validation completed with 0 errors.

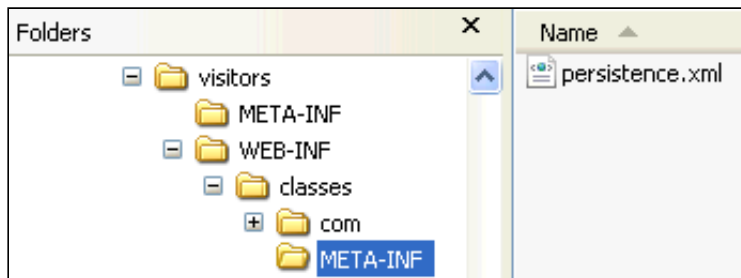
This message indicates that the application can be deployed to a Java Cloud Service.



## Ensuring That the Application Uses the Database Cloud Service

The sample web application, found in the **visitors.war** archive file, uses the database to store information about visitors to a fictitious aquarium. Before deploying the web application, you need to edit one of the its deployment descriptors to ensure that it uses the Database Cloud Service that was provisioned when you signed up for a Java Cloud Service. To edit the web application's persistence deployment descriptor, perform the following steps:

1. Make a copy of the **visitors.war** file and rename it **visitors.zip**. Unzip the **visitors.zip** file into a directory called **visitors**. In the new **visitors** directory, expand **WEB-INF**, then **classes**, and then **META-INF**, until you find the **persistence.xml** file.



2. Edit the **persistence.xml** file in a text editor. Look for this line:

```
<jta-data-source><Database Cloud Service data source Name></jta-data-source>
```

Change the generic value inside the **<jta-data-source>** tag to the name of the Database Cloud Service that was provisioned along with your Java Cloud Service. In this tutorial, the Database Service name is *database*.

With **database** inserted as the value of the **<jta-data-source>** tag, the XML becomes:

```
<jta-data-source>database</jta-data-source>
```

You also need to add a new **<property>** within the **<properties>** tag. Insert the following property:

```
<property name="eclipselink.target-database" value="Oracle10g" />
```

Save the file after editing it. It should look like this:



3. Zip the contents of the **visitors** directory into a new **visitors.zip** file. Include all subdirectories, but

Zip the contents of the **visitors** directory into a new **visitors.zip** file. Include all subdirectories, but not their full path. Also, do not include the **visitors** directory itself.

For example, if after you created the zip file you examined it in WinZip, the path of the **fish.jpg** file would be blank rather than **visitors**.

If you are zipping from the command line, first **cd** into the **visitors** directory. Then zip with the options **-r -p** (for recursion but not full path). Include all the files by using **\*.\***. The command is:

```
wzip -r -p visitors.zip *.*
```

Note: You may need to add the WinZip folder to the **PATH** to use the command-line version of WinZip:  
**set PATH="C:\Program Files\WinZip";%PATH%**



```
C:\visitors>wzip -r -p visitors.zip *.*
WinZip(R) Command Line Support Add-On Version 2.0 (Build 6663)
Copyright (c) WinZip International LLC 1991-2005 - All Rights Reserved

Adding fish.jpg
Adding index.jsp
Adding META-INF\
Adding META-INF\MANIFEST.MF
Adding visitor\
Adding visitor\fish.jpg
Adding visitor\index.jsp
Adding visitor\META-INF\

creating Zip file visitors.zip
```

After the new zip file has been created, rename it **visitors.war**. This is the web application archive file that you will deploy.

**IMPORTANT NOTE FOR MAC USERS:** If you are using a Mac, you cannot use the default Finder utility for zipping up the new archive file. You must instead download and use *WinZip Mac Edition*. If you use the default Mac utility, it places extra files in the archive which will cause a 403 error when you try to access the application after it is deployed.

### Deploying the Application to the Java Cloud Service and Testing It

To deploy the application to the Java Cloud Service and then test it, perform the following steps:

1. In a web browser, enter the URL for the Java Cloud Services Control console. You can find a link to this URL in the welcome email you received when signing up for a Java Cloud Service. It will be titled **Java Service Console**.

Sign in using the Service Administrator's credentials.

Enter the Service Administrator's username in the **Username** field. In this tutorial it is *bill.bell@oracle.com*.

Enter the Service Administrator's password in the **Password** field.

Enter the Identity Domain name in the **Identity Domain** field. The Identity Domain name can also be found in the welcome email. In this tutorial it is *trialaap*. (Yours will be different.)

Then click the **Sign In** button.

## Sign in to Oracle Cloud

\* Username

\* Password

\* Identity Domain  ?

[Forgot Password](#)

2. After signing in, you should see the Java Cloud Services Control console for your Java Cloud Service. To deploy an application, below the Applications heading, click the **Deploy New** button.

**Java Cloud Services Control**
Page Refreshed **Sep 12, 2012 12:32:40 PM PDT**

[Java Cloud Services](#) > [Java Cloud Service: java](#)

### Java Cloud Service: java

**Performance Summary**

Availability

Sep 12

**General**

Service Version 12.2

Application Disk Usage (MB) 0.00

**JDBC Usage**

Open JDBC Connections 0

JDBC Connection Creates (per minute) 2.88

**Applications**

View ▾
+ Deploy New
»

| Application Name | Status | Test Application | State  |
|------------------|--------|------------------|--------|
| welcome-app      | ↑      |                  | Active |

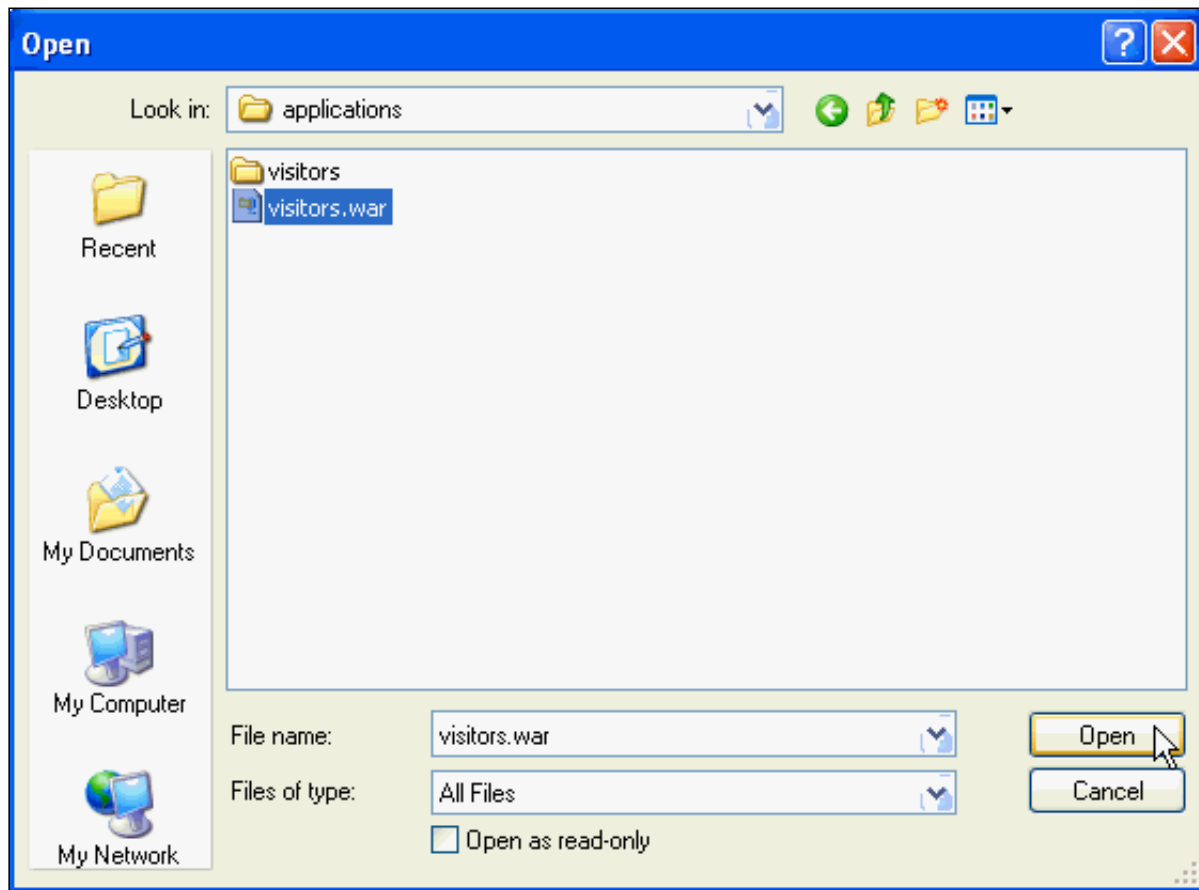
3. On the Deploy Application screen, enter **visitors** in the **Application Name** field, and click the **Choose File** button to browse the file system for the application archive file.

Provide a name for the application you wish to deploy, then navigate to the application archive.

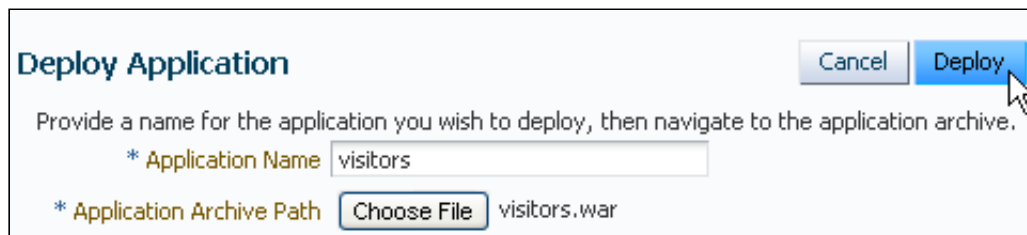
\* Application Name

\* Application Archive Path Choose File No file chosen

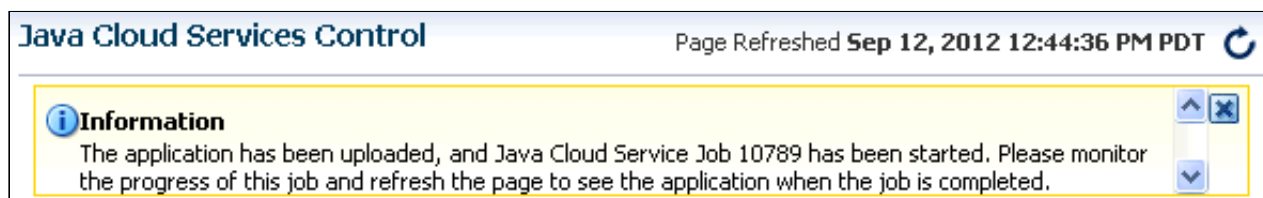
4. Browse to the location of the new **visitors.war** file that you just created. This is the one that has the updated deployment descriptor. Select this file and click **Open**.



5. Back on the Deploy Application screen, click the **Deploy** button.



6. After the file is uploaded, you are returned to the Java Cloud Service screen. Notice the message that the application is uploaded and will be deployed.



7. After a while, refresh the page. You can refresh the page by clicking the curved arrow at the top-right of the screen next to the date and time. You will eventually see an entry for the **visitors** application in the Applications table. You can tell that the deployment was successful by the green arrow in the Status column and the indication of "Active" in the State column.



Java Cloud Services Control Page Refreshed Sep 12, 2012 12:46:19 PM PDT

Java Cloud Services > Java Cloud Service: java

## Java Cloud Service: java

**Performance Summary**

Availability

Sep 12

**General**

Service Version 12.2

**JDBC Usage**

Open JDBC Connections 0

**Applications**

View + Deploy New

| Application Name | Status | Test Application | State  |
|------------------|--------|------------------|--------|
| visitors         | ↑      |                  | Active |
| welcome-app      | ↑      |                  | Active |

8. To access the deployed application, click the icon under the Test Application column in the Applications table.

**Applications**

View + Deploy New

| Application Name | Status | Test Application | State  |
|------------------|--------|------------------|--------|
| visitors         | ↑      |                  | Active |
| welcome-app      | ↑      |                  | Active |

9. In the Application URLs pop-up window, click the URL for the application. The application will come up in another browser window or tab.
10. Go to the new browser window or tab for the application. To use this application you must log in. Log in using your Service Administrator's credentials and Identity Domain name.

**Note:** If you are not asked to log in, it is because you are already logged in to the Java Cloud Services Control console and your web browser uses one session ID for all open windows and tabs. If you are already logged in, proceed to the next step.

In this tutorial:

The **Username** that is used to log in is: *bill.bell@oracle.com*. (Use your Service Administrator username instead.)

The **Identity Domain** is: *trialaaop*. (Yours will be different.)

## Sign in to Oracle Cloud

\*

 Username

\*

 Password

\*

 Identity Domain  [?](#)

[Forgot Password](#)

[Sign in](#)

11. On the first page of the sample application, click the **Start** link.

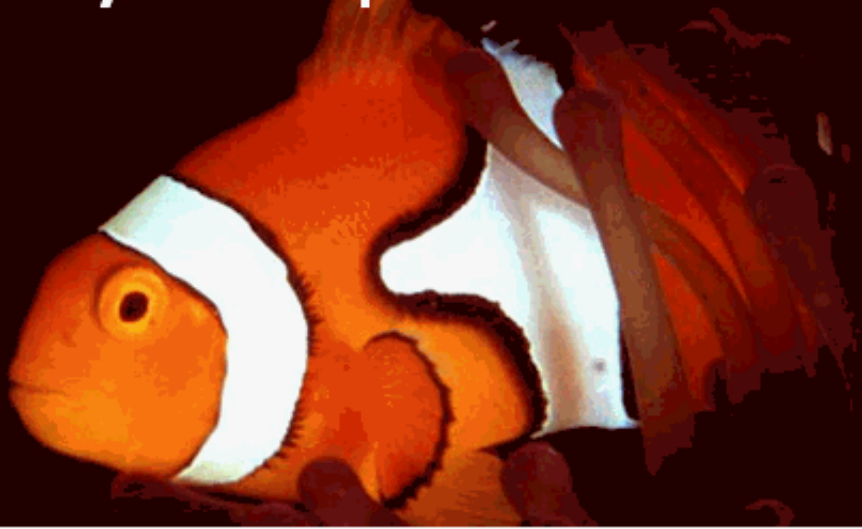
## Visitors

This sample application shows simple database operations using JPA.

[Start](#)

12. Here's the screen that appears after clicking **Start** on the application's home page. This sample application allows you to add a name and comment about a fictitious aquarium. You can also update or delete visitor comments. The information entered is stored in the Database Cloud Service.

# Welcome to Plymouth Aquarium



Please sign our visitor log

**Name:**

**Comment:**

Update Visitor:

**Id:**

**Name:**

**Comment:**

Delete Visitor:

**Id:**

## Recent Visitors

| ID                                     | Name | Visit Date | Comment |
|--|------|------------|---------|
| Number of Visitors 0                   |      |            |         |
| <input type="button" value="Refresh"/> |      |            |         |

### Summary

In this tutorial, you have learned how to:

- o Use the Whitelist Java utility to check a sample application before deploying it
- o Modify the **persistence.xml** deployment descriptor of a sample application so that it uses a Database Cloud Service

- Deploy an application to a Java Cloud Service by using the Java Cloud Services Control console

## Resources

- Oracle Java Cloud Service
- Oracle Cloud Tools
- Using the Oracle Java Cloud Service
- To learn more about the Oracle Cloud or the Java Cloud Service, refer to additional OBEs in the OLL website.

## Credits

- **Lead Curriculum Developer:** Bill Bell
- **Other Contributors:** Reza Shafii