

Migrate Java Apps to Oracle Java Cloud Service (JCS) and Application Container Cloud Service (ACCS)

Guidelines

Migrate WebLogic Server (WLS) to Java Cloud Service (JCS)

Flexible options according to your needs

Options	No WLS Multi Tenancy
1. App2Cloud Tool	<ol style="list-style-type: none">1. Capture on-Premise WLS 11g (10.3.6) or 12c domain w/o Partitions with App2Cloud tool2. Create new JCS by selecting App2Cloud option3. JCS 12c is always created4. Modify resources if needed
3. Generic	<ol style="list-style-type: none">1. Create JCS 11g or 12c instance2. Create resources3. Deploy apps

Options	With WLS Multi Tenancy
1. MT Partitions	<ol style="list-style-type: none">1. Export partition from WLS 12c (12.2.x) domain on-Premise2. Create JCS 12c instance with Domain Partitions3. Import the partition into Java Cloud Service Instance4. Modify resources if needed
2. Domain to Partition Conversion Tool (DPCT)	<ol style="list-style-type: none">1. Export WLS 11g (10.3.6) or 12c (12.1.1, 12.1.2, 12.1.3) domain to WLS 12c (12.2.x) partition by using DPCT2. Create JCS 12c instance Domain Partitions3. Import the Partition to JCS4. Modify resources if needed
3. Generic	<ol style="list-style-type: none">1. Create JCS 12c instance with Domain Partitions2. Create resources3. Deploy apps

Migrate JEE Apps to JCS: when the App already exists

Options depends on your existing Java App

App exists in WAR/JAR/EAR

- App is WebLogic (WLS) based
 - WSL 11g (10.3.6) or 12c (i.e. built with JEE7)
 - Directly deploy app on JCS via WLS Admin or FMW console
 - Older WLS 10/9/8 built with JEE < JEE7
 - App re-build with JEE7 is required. Then deploy app on JCS via WLS Admin or FMW console
- App is JEE standard-based
 - App is built with JEE7
 - Directly deploy app on JCS via WLS Admin or FMW console
 - App is built with JEE < JEE7
 - App re-build with JEE7 is required. Then deploy app on JCS via WLS Admin or FMW console
- App has custom JEE server libs (i.e. JBoss, WebSphere,...)
 - App migration on WLS platform 11g or 12c with JEE7 is required:
 - Re-compile app via JDeveloper or OEPE by choosing WebLogic runtime (either 11g or 12c with JEE7)
 - Code modifications (on classes/interfaces/libs) might be needed
 - Re-build app
 - Deploy app on JCS via WLS Admin or FMW console

Migrate JEE Apps to JCS: when the App doesn't exist

Flexible options according to your needs

- Develop & Build in IDE (Eclipse, JDeveloper)
 - Develop app in IDE
 - Deploy either:
 - directly the app on JCS via remote connection from IDE to the WLS (on JCS)
 - or the built WAR/JAR/EAR on JCS via WLS Admin or FMW console
- Develop in IDE, Build in DevCS:
 - Develop app in IDE
 - Push code to DevCS via IDE's Oracle Cloud Tools plugin
 - Build in DevCS
 - Deploy either:
 - from DevCS on JCS via Deploy
 - or the built artifact WAR/JAR/EAR on JCS via WLS Admin or FMW console
- Develop in Git & Build in IDE
 - Import the code from Git to IDE
 - Build in IDE
 - Deploy either:
 - directly the app on JCS via remote connection from IDE to the WLS (on JCS)
 - or the built WAR/JAR/EAR on JCS via WLS Admin or FMW console
- Develop in Git, Build in DevCS:
 - Import the code from Git to DevCS
 - Build in DevCS
 - Deploy either:
 - from DevCS on JCS via Deploy
 - or the built artifact WAR/JAR/EAR on JCS via WLS Admin or FMW console

Migrate Java Apps to Application Container Cloud Service (ACCS)

App is already built (i.e. packaged in ZIP with JAR & manifest.json)

- Create an app on ACCS via Cloud Tooling by importing the built ZIP

App is not already built or doesn't exist

- Develop & build in IDE (Eclipse, JDeveloper):
 - Develop & build the app in IDE
 - Create an app on ACCS via Cloud Tooling by importing the built ZIP
- Develop in IDE, build in DevCS
 - Develop app in IDE
 - Push code to DevCS via IDE's Oracle Cloud Tools plugin
 - Build in DevCS
 - Deploy either:
 - from DevCS on ACCS via Deploy
 - or create an app on ACCS via Cloud Tooling by importing the ZIP artifact
- Develop in Git, build in IDE:
 - Import the code from Git to IDE
 - Build the app in IDE
 - Create an app on ACCS via Cloud Tooling by importing the built ZIP
- Develop in Git, build in DevCS
 - Import the code from Git to DevCS
 - Build in DevCS
 - Deploy either:
 - from DevCS on ACCS via Deploy
 - or create an app on ACCS via Cloud Tooling by importing the ZIP artifact