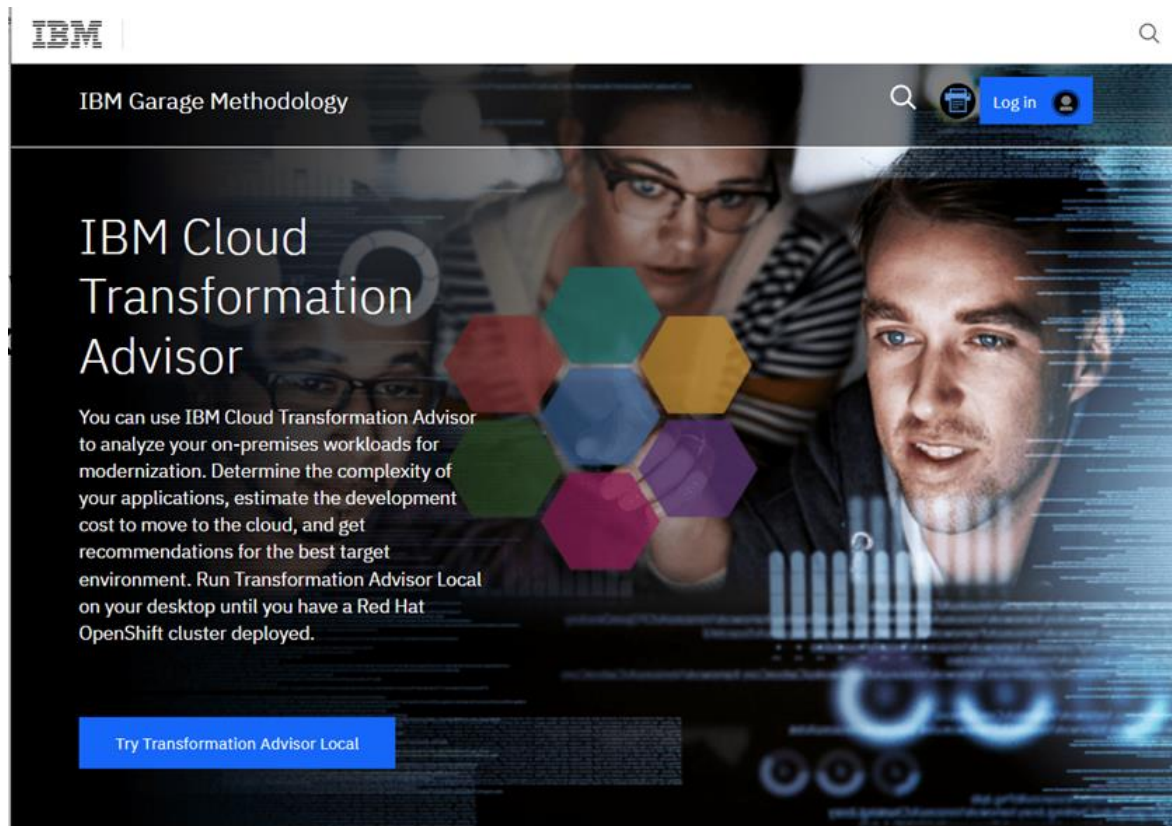


Modernize your applications with insight via IBM Cloud Transformation Advisor



Owner: Lars Besselmann, IBM

Current document version: 2.2

Used software: IBM Cloud Transformation Advisor 3.0

Last updated: February 2022

Duration: 60 mins

1. Contents

Modernize your applications with insight via IBM Cloud Transformation Advisor	1
1. Contents.....	2
2. Lab Introduction - IBM Cloud Transformation Advisor.....	3
1. Introduction to Transformation Advisor	3
2. Business Context	3
3. Learning Objectives.....	3
4. Lab requirements	4
5. What is Already Completed	4
6. Lab Tasks	4
7. The lab environment.....	4
3. Accessing and starting the environment	5
4. Getting Started with Transformation Advisor	6
1. Review the on-prem WebSphere apps	6
2. Access Transformation Advisor.....	8
5. Assess the applications	12
1. Use the Transformation Advisor collector.....	12
2. Evaluate On-Premises Java Applications.....	14
6. Migrate the application.....	22
1. Migrate the modresorts application to Liberty	22
2. Containerize the modresorts application	26
7. Troubleshooting.....	28
1. Transformation Advisor	28
2. Liberty startup fails	29
8. Cleanup	29
9. Summary	29
10. Appendix	30
1. Content of the file Lab_TAcommands.txt	30



Note: To ease the copy and paste, the commands used in the lab have been documented in the file https://larsbesselmannibm.github.io/labs/WSHE/lab_TAcommands.txt
If you want to copy it to your local system, use the following command to copy it to your desktop:

```
curl  
https://larsbesselmannibm.github.io/labs/WSA/lab_TAcommands.txt  
> /var/IBM/temp/lab_TAcommands.txt
```

2. Lab Introduction - IBM Cloud Transformation Advisor

1. Introduction to Transformation Advisor

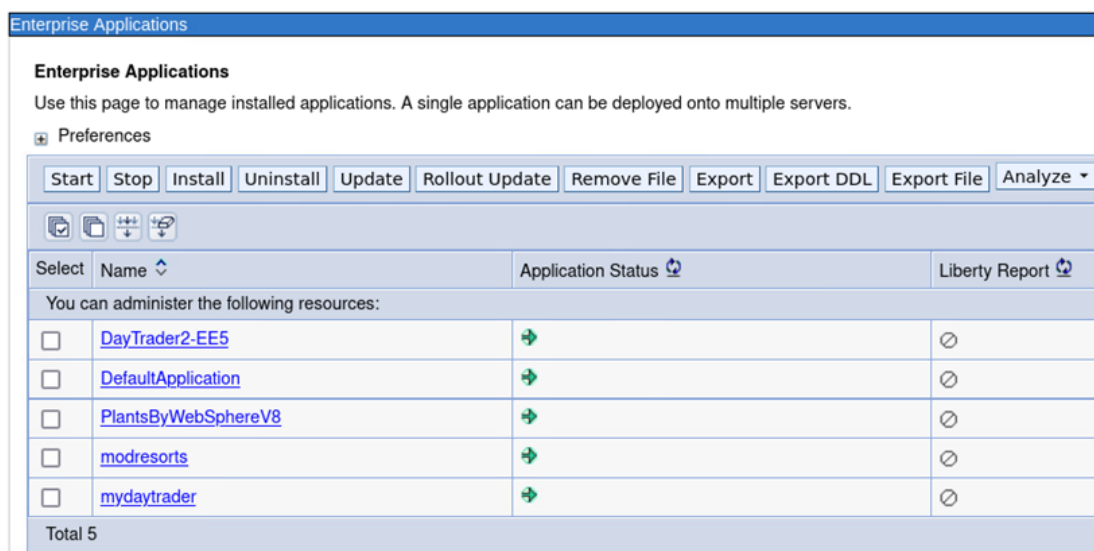
IBM Cloud Transformation Advisor helps you to evaluate on-premises Java applications and identify a migration candidate for moving to the cloud. When you complete this lab, you learn how to use this tool to quickly analyze on-premise Java applications without accessing their source code and to estimate the move to cloud efforts. The Transformation Advisor tool can

- identify the Java EE programming models in the app.
- determine the complexity of apps by listing a high-level inventory of the content and structure of each app.
- highlight Java EE programming model and WebSphere API differences between the WebSphere profile types
- learn any Java EE specification implementation differences that might affect the app

Additionally, the tool provides a recommendation for the right-fit IBM WebSphere Application Server edition and offers advice, best practices and potential solutions to assess the ease of moving apps to Liberty or newer versions of WebSphere traditional. It accelerates application migrating to cloud process, minimize errors and risks and reduce time to market.

2. Business Context

Your company has several web applications deployed to WebSphere Application Server (WAS) environment.



The screenshot shows the 'Enterprise Applications' management interface. It includes a header with the title and a description: 'Use this page to manage installed applications. A single application can be deployed onto multiple servers.' Below the header is a 'Preferences' section. A toolbar contains buttons for 'Start', 'Stop', 'Install', 'Uninstall', 'Update', 'Rollout Update', 'Remove File', 'Export', 'Export DDL', 'Export File', and 'Analyze'. Below the toolbar is a table with columns: 'Select', 'Name', 'Application Status', and 'Liberty Report'. The table lists five applications: 'DayTrader2-EE5', 'DefaultApplication', 'PlantsByWebSphereV8', 'modresorts', and 'mydaytrader'. Each application has a checkbox in the 'Select' column, a green arrow icon in the 'Application Status' column, and a circle with a slash icon in the 'Liberty Report' column. At the bottom of the table, it says 'Total 5'.

Select	Name	Application Status	Liberty Report
<input type="checkbox"/>	DayTrader2-EE5	➔	⊗
<input type="checkbox"/>	DefaultApplication	➔	⊗
<input type="checkbox"/>	PlantsByWebSphereV8	➔	⊗
<input type="checkbox"/>	modresorts	➔	⊗
<input type="checkbox"/>	mydaytrader	➔	⊗
Total 5			

Your company wants to move these applications to a lightweight WebSphere Liberty server on cloud, but you are not sure how much effort the migration process might take. You decide to use the IBM Transformation Advisor to do a quick evaluation of these applications without their source code to identify a good candidate application to move to cloud based on the analysis result.

3. Learning Objectives

The objectives of this lab are to:

- learn how to collect Java application and configuration data using the Transformation Advisor Data Collector tool.
- learn how to use the Transformation Advisor to evaluate the move to cloud efforts and to identify the good candidate for migration.
- Learn how to use the migration assets created by Transformation Advisor

You will need an estimated **60 minutes** to complete this lab.

4. Lab requirements

The following prerequisites must be completed prior to beginning this lab:

- Familiarity with basic Linux commands and Docker
- Have internet access
- Have basic knowledge of WebSphere Liberty
- Have access to the Lab environment

5. What is Already Completed

A Lab environment with one Linux workstation VM has been provided for this lab.

- The VM contains all required software, so no need to download something from the internet.
- The login credentials for the workstation VM are: User ID: ibmdemo Password: passw0rd

6. Lab Tasks

In this lab, you access WebSphere Application Server to review the deployment of the JEE applications. Then you are going to the Transformation Advisor to identify a good candidate application for moving to cloud. To identify which Java EE programming models are on the server, you could run the Transformation Advisor Data Collector tool against the server. The Transformation Advisor creates an inventory of the content and structure of each application and learn about problems that might occur if you move the application to cloud. Finally, you review the analysis reports to determine the complexity of the move-to-cloud efforts and select the migration candidate app.

Here are the activities involved in this process:

- Log in to WebSphere Application Server to review the deployed JEE applications
- Run the Transformation Advisor Data Collector tool against the WebSphere Application Server to get application data
- Review the analysis reports that Transformation Advisor generates to identify the right candidate application for a rapid and cost-effective migration to cloud
- Use the migration bundle to migrate your application to Liberty
- Use the migration bundle to containerize your application

7. The lab environment

One (1) Linux VM has been provided for this lab. You execute all the lab tasks on this workstation VM.

There are several components installed in the VM:

- WebSphere Application Server Network Deployment v8.5.5
 - Binaries under /usr/IBM/WAS855ND
 - Profiles under /usr/IBM/WAS855ND/profiles
- WebSphere Liberty
 - Binaries under /usr/IBM/Liberty/wlp
- IBM Cloud Transformation Advisor 3.0
 - Program under /usr/IBM/TA/transformation-advisor-local-3.0.0

3. Accessing and starting the environment

The environment consists of one instance:

- A workstation which is a RHEL VM dedicated to one user. It is called **Student VM** in the lab instructions and contains everything what you need for the lab.

Access the Student VM

- Use the connection details that have been provided to you.
- If you are connected via VNC, use the URL <https://iccve.uk.ibm.com/cloudhur2>.

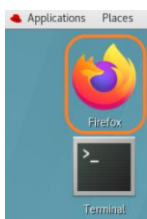
Login to the Student VM

- If you are connected via VNC, you should be automatically logged in as ibmdemo.
- Otherwise log in as user “ibmdemo” and enter “passw0rd” as the password:
Password: passw0rd (lowercase with a zero instead of the o)

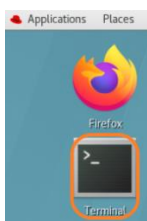
Preparation for the labs

As part of the lab, you will work with the Firefox browser as well as with Terminal window.

- On the *Student VM* and open a browser (ignore any comments to recover a session).
Enter the URL https://larsbesselmannibm.github.io/labs/WSHE/lab_TAcommands.txt



- Open a terminal window by clicking its icon from the Desktop toolbar.



Note: To ease the copy and paste, the commands used in the lab have been documented in the file

https://larsbesselmannibm.github.io/labs/WSHE/lab_TAcommands.txt

If you want to copy it to your local system, use the following command to copy it to your desktop:

```
curl  
https://larsbesselmannibm.github.io/labs/WSA/lab_TAcommands.txt  
> /var/IBM/temp/lab_TAcommands.txt
```

4. Getting Started with Transformation Advisor

1. Review the on-prem WebSphere apps

In this task, you take a look at the sample applications deployed to the local WebSphere Application Server (WAS) environment. You are going to identify one of them to be the god candidate to move the cloud later.

1. Start WebSphere Application Server

In the workstation VM, you have a local traditional WebSphere Application Server which hosts several sample applications.

To start the WAS server:

- a. In the terminal window, issue the command below to start the WAS server.

```
/usr/IBM/WAS855ND/profiles/StandaloneSrv1/bin/startServer.sh server1
```

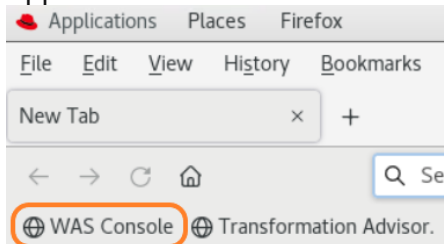
- b. Wait until the server has been started

```
[ibmdemo@RHEL7Guac ~]$ /usr/IBM/WAS855ND/profiles/StandaloneSrv1/bin/startServer.sh server1
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/StandaloneSrv1/logs/server1/startServer.log
ADMU0128I: Starting tool with the StandaloneSrv1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 1236
```

- c. Access the WAS Admin Console to view the application deployed by clicking the Firefox icon on the Desktop toolbar (or open a new tab in the existing Firefox window).



- d. From the web browser window and click the **WAS Console** bookmark to launch the WebSphere Application Server console.



- e. If you see the Warning: Potential Security Risk Ahead message, click Advanced>Accept the Risk and continue.
- f. In the WAS Admin Console login page, enter the User ID and Password as: **wasadmin/password** and click Login.
- g. On the WAS Console page, click Applications -> Application Types -> WebSphere enterprise applications to view the apps deployed.
- h. In the Enterprise Applications list, you can see all applications deployed.

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

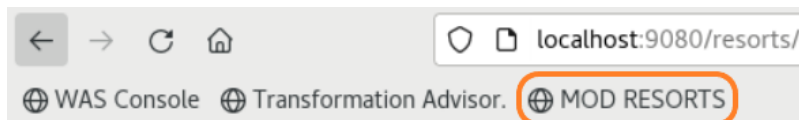
Preferences

Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL Export File Analyze ▾

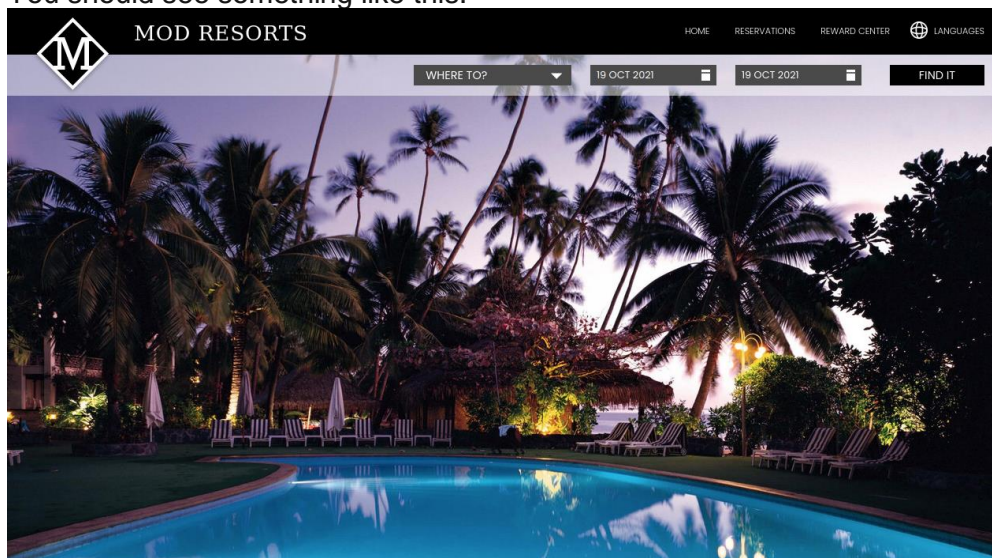
☐ ☐ ☐ ☐

Select	Name ▾	Application Status ↻	Liberty Report ↻
You can administer the following resources:			
<input type="checkbox"/>	DayTrader2-EE5	➔	⊗
<input type="checkbox"/>	DefaultApplication	➔	⊗
<input type="checkbox"/>	PlantsByWebSphereV8	➔	⊗
<input type="checkbox"/>	modresorts	➔	⊗
<input type="checkbox"/>	mydaytrader	➔	⊗
Total 5			

- i. Next, you use Transformation Advisor to analyze these applications to identify a good candidate to be moved to Liberty.
- j. But before doing the analysis, access the application modresorts via browser to see how it looks like (this is the application that we finally will migrate)
 - i. Open a new browser tab and insert the URL <http://localhost:9080/resorts>
Or click on the browser link



- ii. You should see something like this:



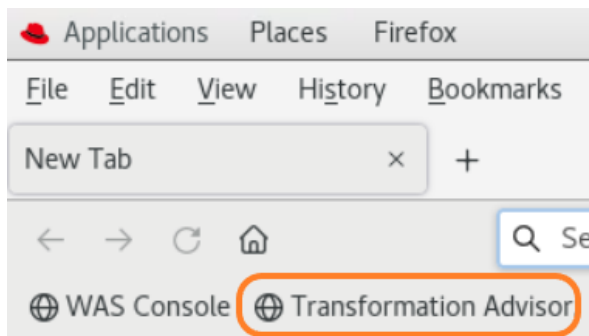
- iii. Feel free to navigate around to see how the application looks like.

2. Access Transformation Advisor

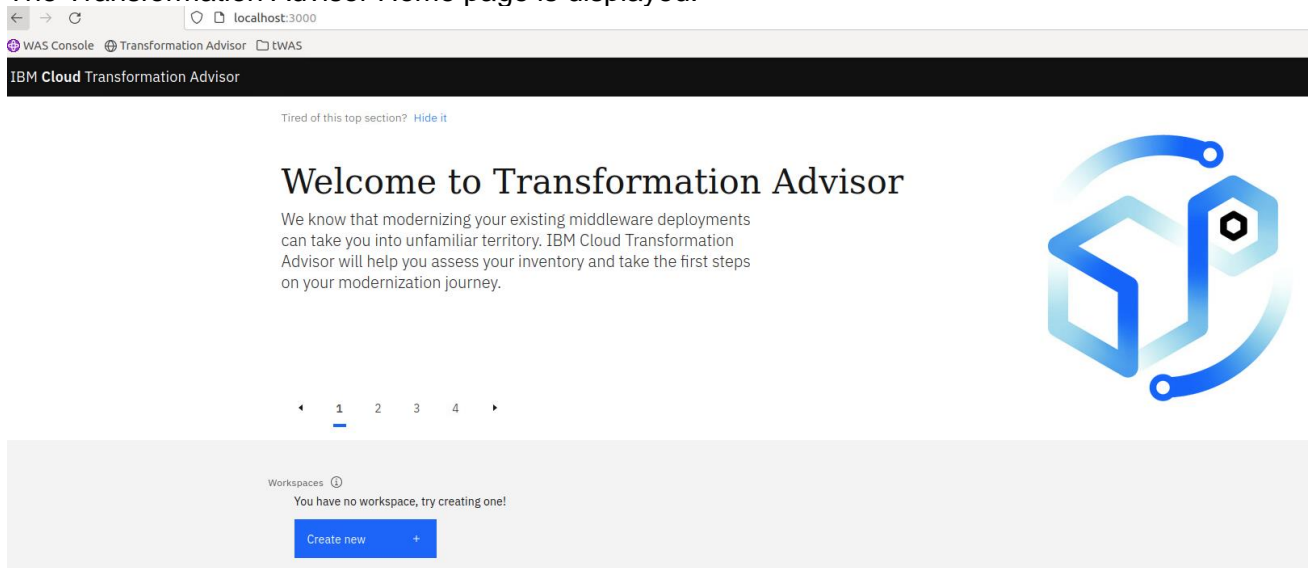
The Transformation Advisor can evaluate any Java based applications. In this lab, you are going to use it to evaluate whether the on-premises WebSphere application, Mod Resorts, is suitable to move to cloud and what the effort might be to get it there. You can use Transformation Advisor Data Collector utility to get the application data from the WebSphere Application Server running on the workstation VM. The utility can be downloaded from the Transformation Advisor web page.

The Transformation Advisor is installed as standalone version in the workstation VM.

1. In the web browser window, open a new tab, then click the Transformation Advisor bookmark.
(URL: <http://localhost:3000>)

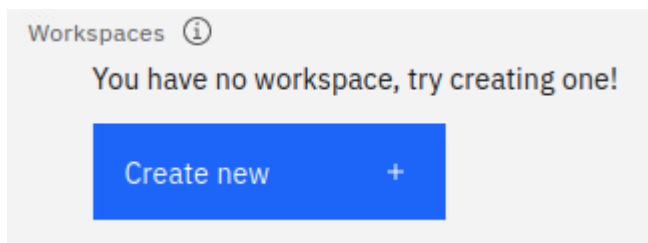


2. The Transformation Advisor Home page is displayed.



3. In the Transformation Advisor page, you first create a new workspace and then a collection.
Note: A workspace is a designated area that houses the migration recommendations provided by Transformation Advisor against your application server environment. You can name and organize these however you want, whether it's by business application, location or teams.
Each workspace can be divided into collections for more focused assessment and planning. Like workspaces, collections can be named and organized in whatever way you want.

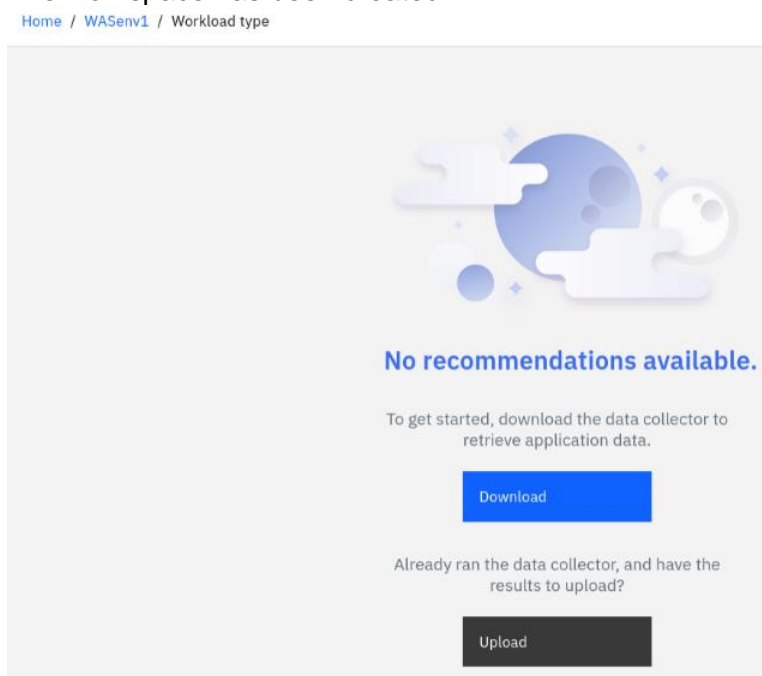
Click on the button **Create new** to create a new workspace



4. Enter as workspace name **WASenv1**, then click **Create**.

A screenshot of a 'Create a new workspace' dialog box. It has a title bar with a close button. Below the title, it says 'Workspace name'. There is a text input field containing 'WASenv1'. At the bottom, there are two buttons: 'Cancel' and 'Create'.

5. The workspace has been created.



Next step is to use the Transformation Advisor Collector to gather data about your applications. Click on the **Download** button.

6. Now you can select the operating system that fits to your application server environment.

Data collector

Getting results in your collection is not hard. To get started, download and run the Data Collector tool in the system your applications live.

Download

In order to download the appropriate data collector, please specify the source operating system.

Source Operating System

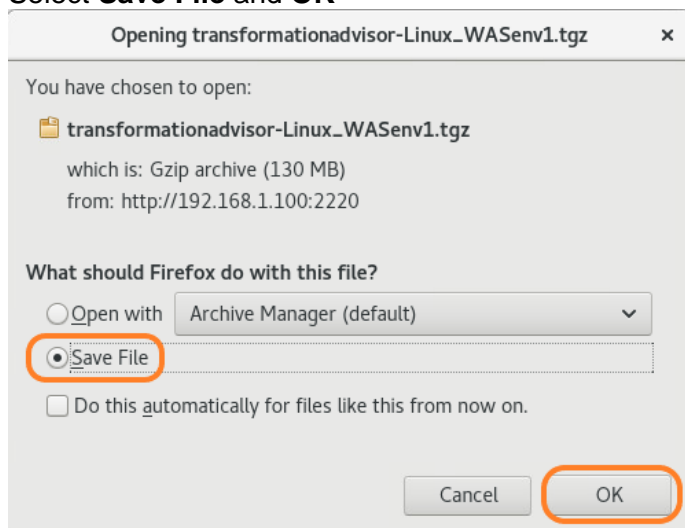
Linux ▼

Download for Linux



Leave the Operating System to **Linux** and click on the button **Download for Linux**.

7. Select **Save File** and **OK**



The file is saved in the Download folder `/home/ibmdemo/Downloads`, the name is: `transformationadvisor-Linux_WASenv1.tgz`

8. Scroll down to see the options for the different application server runtimes.

Run Tool

Select your domain and analysis type below and run the generated command from the data collector directory to begin scanning.

Domain

IBM WebSphere ^



```

[RE_HOME_DIR] -p <PROFILE_NAME> [--scan-node --ignore-
-library --applications --applications-file] []
    
```

Run Tool

Select your domain and analysis type below and run the generated command from the data collector directory to begin scanning.

Domain

IBM WebSphere ▾

Analysis of

Apps & Configuration ^

Apps & Configuration ✓

Apps only

.ear and .war files

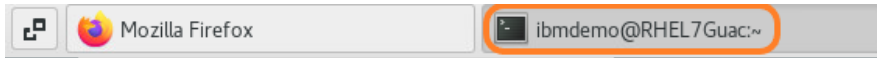
```
ERE_HOME_DIR> -p <PROFILE_NAME> [--scan-node --ignore-  
library --applications --applications-file] ([
```

5. Assess the applications

1. Use the Transformation Advisor collector

After downloading the zipped Data Collector utility, you need to unpack it and run the utility against the WAS server to collect all deployed applications and their configuration data from WAS server.

1. Switch to the command shell.



2. In the command shell, execute the following commands to extract the collector:

```
mkdir -p /var/IBM/temp/TA_collector/  
cd /var/IBM/temp/TA_collector/  
tar -zxvf /home/ibmdemo/Downloads/transformationadvisor-Linux_WASenv1.tgz
```

3. Switch to the collector directory

```
cd /var/IBM/temp/TA_collector/transformationadvisor-3.0.0
```

4. Set the Java SDK that fits to your environment

As the JDK shipped with the collector does not fit to our environment, we use the JDK provided with WAS. This is done by setting the JAVA_HOME environment

```
export JAVA_HOME=/usr/IBM/WAS855ND/java/
```

5. Run the collector with the help option to see the available options

```
bin/transformationadvisor --help
```

6. Run the collector to analyze the WAS applications
Execute the command

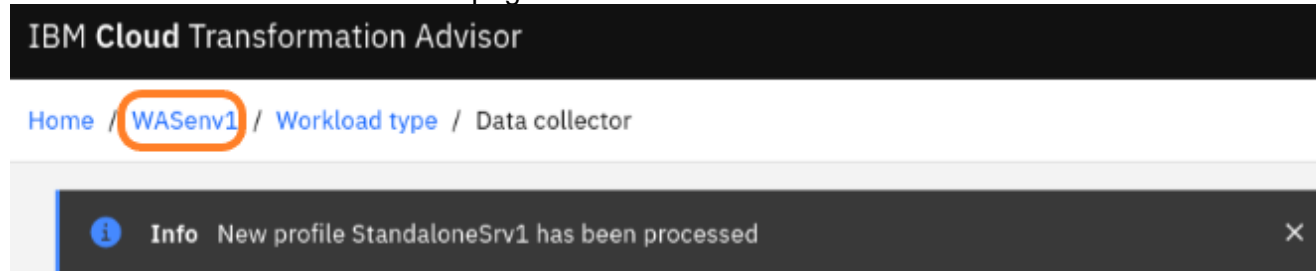
```
bin/transformationadvisor -w /usr/IBM/WAS855ND/ -p StandaloneSrv1
```

7. Accept the license agreement ("1. I have read and agreed to the license agreements").
A panel is shown, which will change over time to finally

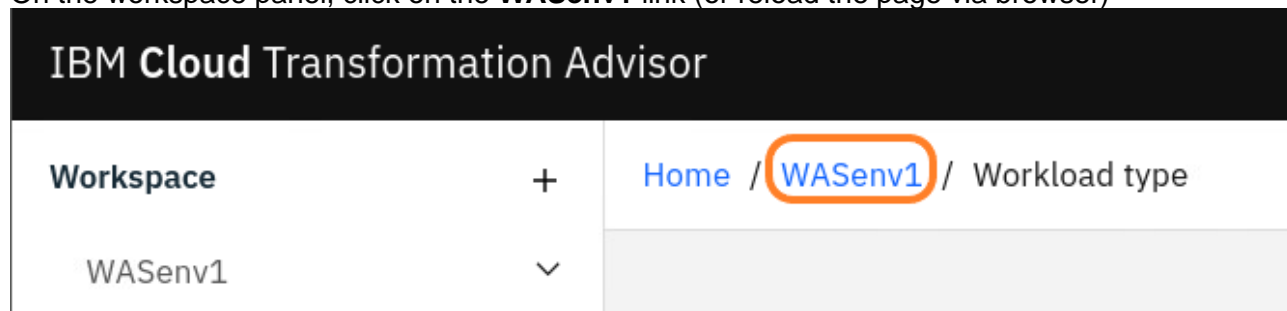
2. Evaluate On-Premises Java Applications

In this section, you are going to use the Transformation Advisor UI to view the application data analysis results.

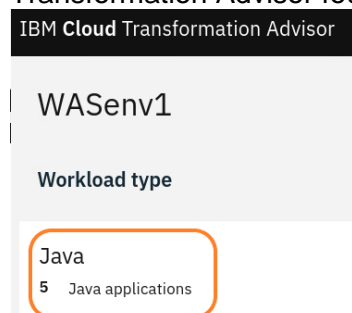
1. Go back to Transformation Advisor page in web browser and click the **WASenv1** link.



2. On the workspace panel, click on the **WASenv1** link (or reload the page via browser)



3. Transformation Advisor found Java applications, so click on the link to get the details



4. In the Recommendations page, you can see that the 5 WAS applications are listed.

IBM Cloud Transformation Advisor

Workspace: WASenv1

Workload type: Java

Home / WASenv1 / Java

Java

Options

Collections: Collections

Migration target: 1 x Migration targets

Total Applications: 5

Avg. cost per application: 1.6

Workspace estimated total costs: Common Code 3 days, Unique app code 5.5 days, Total cost 8.5 days

Java applications (5) Common code files (1)

Migration assessment: Java applications

Java application	Collection/Profile	Complexity	Issues	Common code files	Application cost in days
DayTrader2-EE5.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	4	1	8
DefaultApplication.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	1	0	0.5
modresorts.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Simple	2	0	0
mydaytrader.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	4	1	8
PlantsByWebSphereV8.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	1	0	0.5

5. In the upper area, you can see easily that Transformation Advisor identified some common code, which means the effort to migrate all applications in the workspace is lower than the effort to migrate each application on its own.

Collections: Collections

Migration target: 1 x Migration targets

Total Applications: 5

Avg. cost per application: 1.6

Workspace estimated total costs: Common Code 3 days, Unique app code 5.5 days, Total cost 8.5 days

6. The default migration target is set to Open Liberty. To see if WebSphere Liberty as target requires less effort, change the migration target to compare both Liberty runtimes.

Collections: Collections

Migration target: 2 x Migration targets

☒ Open Liberty

☒ WebSphere Liberty

☐ WebSphere traditional

7. As you can see immediately, the efforts for WebSphere Liberty are lower than for Open Liberty.

Collections: Collections































Migration target: 2 x Migration targets

Total Applications: 5

Workspace estimated total costs: Open Liberty 8.5 days, WebSphere Liberty 5.5 days

We will talk about the reasons later.


8. Click on “Application costs in days” to sort by cost.

Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ⓘ
modresorts.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Simple	   2	0	0
modresorts.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Simple	   2	0	0
PlantsByWebSphereV8.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 1   8	0	0.5
PlantsByWebSphereV8.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 1   8	0	0.5
DefaultApplication.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 1   4	0	0.5
DefaultApplication.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 1   4	0	0.5
▼ DayTrader2-EE5.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 1   7	1	5
▼ mydaytrader.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 1   7	1	5
▼ DayTrader2-EE5.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 4   8	1	8
▼ mydaytrader.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	 4   8	1	8

9. Let’s take a closer look at some of the applications.

- If you want to move the **modresorts.ear** application to Liberty, the complexity level is Simple, which indicates that the application code does not need to be changed. The application has no dependency, has two minor level issue and the estimated development effort is zero day because no code change is required.
- PlantsByWebSphereV8.ear** requires code changes, so the complexity level is Moderate. But as you can see, the estimated development effort is half a day, so also a possible candidate for Liberty.
- As the name indicates, **DayTrader2-EE5.ear** is an application that uses some older Java EE standards. The application **mydaytrader.ear** is a derivation of it, both share some common code. While the applications have for both runtimes a complexity level of Moderate, the estimated migration effort for WebSphere Liberty is less than for Open Liberty.


10. Open the twisty for the application DayTrader2 with target Open Liberty. The application uses the common file dt-ejb.jar which takes 3 days of the total costs of 8 days and is used by 2 applications.

Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ⓘ
 DayTrader2-EE5.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 4 ● 8	1	8

Common code files (1)

Source environment: IBM WebSphere Application Server Network Deployment
Common Code: 3 days ⓘ Unique app code: 5 days ⓘ Total cost: 8 days

Complexity	Name	Used By	Complexity
Unassessed 0	dt-ejb.jar	2 apps	Moderate
Simple 0			
Moderate 1			
Complex 0			



11. Sort by application name and click on the application **DayTrader2-EE5.ear** with migration target **Open Liberty**.

Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ⓘ
▼ DayTrader2-EE5.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 4 ● 8	1	8
▼ DayTrader2-EE5.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 1 ● 7	1	5

12. Scroll down and you can see that there are some technology issues in the common code around persistence.

Common Code Issues					
Technology issues	▲ 3 ● ●	3 days	^		
▲ Disable the persistence unit second-level cache		0.25 days	▼		
▲ OpenJPA and WebSphere JPA configuration properties must be migrated		2.5 days	▼		
▲ Do not use OpenJPA providers in the persistence.xml file		0.25 days	▼		

13. In the browser go back to the previous screen and click on the application **DayTrader2-EE5.ear** with migration target **WebSphere Liberty**.

Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ⓘ
▼ DayTrader2-EE5.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 4 ● 8	1	8
▼ DayTrader2-EE5.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 1 ● 7	1	5

14. Scroll down and you can see that there are no technology issues in the common code.

Common Code Issues					
External dependencies	● ● ● 1	0 days	▼		

This is because WebSphere Liberty supports next to Java EE 7 also Java EE 6 Web Profile, especially

JPA 2.0 which is used by the application to implement persistence.

15. Scroll down to the section called “Unique Code Issues” and open the twisty to see the details.

Unique Code Issues			
Technology issues	▲ 1 ● ●	5 days	^
▲ Java API for XML-based RPC (JAX-RPC) is unavailable		5 days	▼

As you can see, there are issues around JAX-RPC which take 5 days. You will see more details about it later in the analysis report.

Hint: In TA we limit the cost for JAX-RPC per application and workspace to 5 days.

16. Take a look at the **External dependencies** for the common and the unique code (you might have to open the twisty).

Common Code Issues			
External dependencies	● ● ● 1	0 days	
● Databases		0 days	
Unique Code Issues			
Technology issues	▲ 1 ● ●	5 days	
External dependencies	● ● ● 4	0 days	
● Remote EJB lookups		0 days	
● Java Message Service (JMS)		0 days	
● Remote web services		0 days	
● Databases		0 days	

As you can see, the application depends on one or more databases, messaging systems and accesses other systems via Remote EJB lookups and Remote web services. The dependencies give you some hints what to consider when moving into containers or public cloud for example. Feel free to open for each issue and dependency the related twisty to get more insight.

17. Scroll down to the bottom to see the additional reports.

Technology report	Inventory report	Analysis report
Details on which IBM platforms support the technologies used by the applications.	High-level inventory of the content and structure of each application, plus information about potential deployment problems and performance considerations.	Potential issues, their severity and possible solutions.

The three reports have been created by the collector and contain more technical details about:

- the issues (Analysis report),
- the application structure and deployment topology (Inventory report)
- the target runtime (Technology report).

Feel free to open the reports to get some idea about the content. (The reports open in a separate browser tab)

18. Even if we cannot deploy the application without any change to Liberty, let's take a look at the migration plan. Scroll to the top of the page and click on **View migration plan**.

DayTrader2-EE5.ear

[View migration plan](#)

Source environment	Migration Target	Complexity	Issues ⓘ	Common code files	Unique code cost	Total application cost
IBM WebSphere Application Server Network Deployment StandaloneSrv1 Version: 8.5.5.18	WebSphere Liberty	Moderate	▲ 1 ● 7	1	5	5 days

19. Transformation Advisor generates some deployment assets which help to deploy the application to Liberty, into containers and Kubernetes. As you have seen before, DayTrader has several external dependencies. Open the Liberty configuration file `server.xml` to see what has been gathered from the tWAS configuration. Click on the link for `server.xml` and open it with Firefox.

Migration Files

These files are generated by Transformation Advisor to assist in migrating this application:

[server.xml](#) ↓[jvm.options](#) ↓[pom.xml](#) ↓[Application CR](#) ↓[Dockerfile](#) ↓

- a. As you can see, the `server.xml` file contains the required features for JPA, JMS and SSL.

```
-<featureManager>
```

```
-<!--
```

```
    The following features are available in all editions of Liberty.
```

```
-->
```

```
    <feature>jpa-2.0</feature>
```

```
    <feature>jsp-2.2</feature>
```

```
    <feature>servlet-3.0</feature>
```

```
    <feature>transportSecurity-1.0</feature>
```

```
-<!--
```

```
    The following features are available in all editions of Liberty, except for Liberty Core.
```

```
-->
```

```
    <feature>jms-1.1</feature>
```

```
    <feature>wasJmsClient-1.1</feature>
```

```
    <feature>wasJmsServer-1.0</feature>
```

```
</featureManager>
```

- b. Scroll down and you can see the application configuration including bindings

```
-<enterpriseApplication location="daytrader2ee5-1.0.0.ear">
  <web-ext context-root="/daytrader" moduleName="web"/>
  <ejb-jar-bnd moduleName="dt-ejb">
    <message-driven name="DTBroker3MDB">
      <jca-adapter activation-spec-auth-alias="TradeAdminAuthData" activation-spec-binding-name="eis/TradeBrokerMDB"
        destination-binding-name="jms/TradeBrokerQueue"/>
    </message-driven>
    <message-driven name="DTStreamer3MDB">
      <jca-adapter activation-spec-auth-alias="TradeAdminAuthData" activation-spec-binding-name="eis/TradeStreamerMDB"
        destination-binding-name="jms/TradeStreamerTopic"/>
    </message-driven>
  </ejb-jar-bnd>
</enterpriseApplication>
```

- c. Further down, you can see the JDBC configuration

```
<jdbcDriver id="DB2_Universal_JDBC_Driver_Provider_Only_XA">
  <library>
    <file name="/var/IBM/db2drivers/db2drivers/db2jcc.jar"/>
    <file name="/var/IBM/db2drivers/db2drivers/db2jcc_license_cu.jar"/>
  </library>
</jdbcDriver>
<dataSource beginTranForResultSetScrollingAPIs="false" beginTranForVendorAPIs="false"
containerAuthDataRef="TradeDataSourceAuthData" id="NoTxTradeDataSource"
jdbcDriverRef="DB2_Universal_JDBC_Driver_Provider_Only_XA" jndiName="jdbc/NoTxTradeDataSource"
recoveryAuthDataRef="TradeDataSourceAuthData" transactional="false" type="javax.sql.XADataSource">
  <properties.db2.jcc databaseName="{NoTxTradeDataSource_databaseName_1}" driverType="4"
portNumber="{NoTxTradeDataSource_portNumber_1}" serverName="{NoTxTradeDataSource_serverName_1}"/>
  <connectionManager connectionTimeout="180" enableContainerAuthForDirectLookups="true" minPoolSize="10"/>
</dataSource>
<dataSource beginTranForResultSetScrollingAPIs="false" beginTranForVendorAPIs="false"
containerAuthDataRef="TradeDataSourceAuthData" id="TradeDataSource"
jdbcDriverRef="DB2_Universal_JDBC_Driver_Provider_Only_XA" jndiName="jdbc/TradeDataSource"
recoveryAuthDataRef="TradeDataSourceAuthData" statementCacheSize="60" type="javax.sql.XADataSource">
  <properties.db2.jcc databaseName="{TradeDataSource_databaseName_1}" driverType="4"
portNumber="{TradeDataSource_portNumber_1}" serverName="{TradeDataSource_serverName_1}"/>
  <connectionManager connectionTimeout="180" enableContainerAuthForDirectLookups="true" minPoolSize="10"/>
</dataSource>
```

- d. You can also find the JMS queue and topic configuration.

```
<jmsQueueConnectionFactory containerAuthDataRef="TradeAdminAuthData" id="TradeBrokerQCF"
jndiName="jms/TradeBrokerQCF">
  <properties.wasJms>
    <connectionManager connectionTimeout="180" enableContainerAuthForDirectLookups="true" maxPoolSize="10"
minPoolSize="1"/>
  </properties.wasJms>
</jmsQueueConnectionFactory>
<jmsTopicConnectionFactory containerAuthDataRef="TradeAdminAuthData" id="TradeStreamerTCF"
jndiName="jms/TradeStreamerTCF">
  <properties.wasJms>
    <connectionManager connectionTimeout="180" enableContainerAuthForDirectLookups="true" maxPoolSize="10"
minPoolSize="1"/>
  </properties.wasJms>
</jmsTopicConnectionFactory>
<messagingEngine>
  <queue id="TradeBrokerJSD"/>
  <topicSpace id="Trade.Topic.Space"/>
</messagingEngine>
<jmsQueue id="TradeBrokerQueue" jndiName="jms/TradeBrokerQueue">
  <properties.wasJms deliveryMode="NonPersistent" queueName="{TradeBrokerQueue_QueueName_1}"/>
</jmsQueue>
<jmsTopic id="TradeStreamerTopic" jndiName="jms/TradeStreamerTopic">
  <properties.wasJms deliveryMode="NonPersistent" topicName="TradeStreamerTopic" topicSpace="Trade.Topic.Space"/>
</jmsTopic>
<jmsActivationSpec authDataRef="TradeAdminAuthData" id="eis/TradeBrokerMDB">
  <properties.wasJms destinationRef="TradeBrokerQueue" destinationType="javax.jms.Queue"/>
</jmsActivationSpec>
<jmsActivationSpec authDataRef="TradeAdminAuthData" id="eis/TradeStreamerMDB">
  <properties.wasJms destinationRef="TradeStreamerTopic" destinationType="javax.jms.Topic"/>
</jmsActivationSpec>
```

- e. As nearly all settings use variables, the configuration is portable. You can find the default values at the bottom of the file (for passwords, there is no default value).

```
<variable defaultValue="" name="httpEndpoint_host_1"/>
<variable defaultValue="9080" name="httpEndpoint_port_1"/>
<variable defaultValue="9443" name="httpEndpoint_secure_port_1"/>
<variable defaultValue="DayTrader2-EE5_ear_StandaloneSrv1Node_NodeDefaultKeyStore_key.p12"
name="NodeDefaultKeyStore_location_1"/>
<variable defaultValue="TLS" name="NodeDefaultSSLSettings_sslProtocol_1"/>
<variable defaultValue="DayTrader2-EE5_ear_StandaloneSrv1Node_NodeDefaultTrustStore_trust.p12"
name="NodeDefaultTrustStore_location_1"/>
<variable defaultValue="tradedb" name="NoTxTradeDataSource_databaseName_1"/>
<variable defaultValue="50002" name="NoTxTradeDataSource_portNumber_1"/>
<variable defaultValue="ibmdemo-db2" name="NoTxTradeDataSource_serverName_1"/>
<variable defaultValue="TradeBrokerJSD" name="TradeBrokerQueue_QueueName_1"/>
<variable defaultValue="tradedb" name="TradeDataSource_databaseName_1"/>
<variable defaultValue="50002" name="TradeDataSource_portNumber_1"/>
<variable defaultValue="ibmdemo-db2" name="TradeDataSource_serverName_1"/>
<variable defaultValue="" name="NodeDefaultKeyStore_password_1"/>
<variable defaultValue="" name="NodeDefaultTrustStore_password_1"/>
<variable defaultValue="" name="TradeAdminAuthData_password_1"/>
<variable defaultValue="" name="TradeAdminAuthData_user_1"/>
<variable defaultValue="" name="TradeDataSourceAuthData_password_1"/>
<variable defaultValue="" name="TradeDataSourceAuthData_user_1"/>
```

To override the settings later, just use an environment variable for example.

- f. Finally close the browser tab for the server.xml.

20. Feel free to also inspect the other files like:

- jvm.options – contains the JVM settings extracted from tWAS
- Dockerfile – contains the instructions to containerize the application

21. Back on the page with the Migration plan, click on **Java**.

IBM Cloud Transformation Advisor

[Home](#) / ... / [Java](#) / [DayTrader2-EE5.ear](#) / Migration plan

22. Back on the Recommendations Page, add WebSphere traditional to the **Migration targets**.

This will allow you to see the efforts for a migration to tWAS v9 . Similar as for Liberty, Transformation Advisor also generates a migration plan for tWAS but in this case, you do not get a server.xml but the required jython scripts to configure the server and deploy the application.

Collections

Migration target ⓘ

3 × Migration targets

- ☒ Open Liberty
- ☒ WebSphere Liberty
- ☒ WebSphere traditional

Collections

Migration target ⓘ

3 × Migration targets

Total Applications

5

Workspace estimated total costs ⓘ

Open Liberty	8.5 days
WebSphere Liberty	5.5 days
WebSphere traditional	0 days

23. You can also export the reports to share the findings

[Home](#) / [WASenv1](#) / Java

Java

Options

Collections

Collections

Migration target ⓘ

3 × Migration targets

Total Applications

5

Workspace estimated total costs ⓘ

Open Liberty	8.5 days
WebSphere Liberty	5.5 days
WebSphere traditional	0 days

Java applications (5) Common code files (1)

Migration assessment

Java applications

Search

Export

Java application	Collection/Profile	Complexity	Issues	Common code files
DayTrader2-EE5.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 4 ● 8	1
DayTrader2-EE5.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Moderate	▲ 1 ● 7	1
DayTrader2-EE5.ear WebSphere traditional	RHEL7Guac / StandaloneSrv1	Simple	● 6	1

Export

- ☒ Summary of workspace
 - Assessment units: 5
 - Common code: 1
 - PDF and CSV
- ☒ Application details
 - PDF

Export

6. Migrate the application

1. Migrate the modresorts application to Liberty

The application modresorts.ear can be migrated to Liberty without any code change, so let's migrate it.

1. Click on the application modresorts.war for target Open Liberty to see the details

Migration assessment Java applications					
mod					
Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ①
modresorts.ear Open Liberty	RHEL7Guac / StandaloneSrv1	Simple	● ● ● 2	0	0
modresorts.ear WebSphere Liberty	RHEL7Guac / StandaloneSrv1	Simple	● ● ● 2	0	0
modresorts.ear WebSphere traditional	RHEL7Guac / StandaloneSrv1	Simple	● ● ● 1	0	0

2. As there are no issues and no dependencies, it should be easy to migrate the application to Liberty. Click on the button **View a migration plan**.

IBM Cloud Transformation Advisor

Home / WASenv1 / Java / modresorts.ear

modresorts.ear

View migration plan →

Source environment

IBM WebSphere Application Server Network Deployment
StandaloneSrv1
Version: 8.5.5.18

Migration Target

Open Liberty

Complexity

Simple

Issues ①

● ● ● 2

Common code files

0

Unique code cost

0

Total application cost

0 days

3. Transformation Advisor generates several assets which help to migrate to Liberty, into containers and Kubernetes.

IBM Cloud Transformation Advisor

Home / ... / Java / modresorts.ear / Migration plan

Migration plan

The files included in your migration bundle help you migrate to IBM Open Liberty, create an image, and package your application as a Kubernetes Operator for easy deployment.

Build type ①

Select the type of application you want to build to help Transformation Advisor determine what files to include in the bundle.

☒ Source code

☐ Binary

Migration Files

These files are generated by Transformation Advisor to assist in migrating this application:

server.xml ↓

pom.xml ↓

Application CR ↓

Dockerfile ↓

- The file server.xml contains the Liberty configuration extracted from tWAS.
- The file pom.xml helps with the integration into maven build
- The custom resources Application CR helps to deploy to Kubernetes
- Dockerfile helps to containerize the application

Select the Build type **Binary**.

Build type ⓘ

Select the type of application you want to build to help Transformation Advisor determine what files to include in the bundle.

☐ Source code

☒ Binary

4. Upload the modresorts-1.0.war file by clicking in the related field

Build type ⓘ

Select the type of application you want to build to help Transformation Advisor determine what files to include in the bundle.

☐ Source code

☒ Binary

Application dependencies

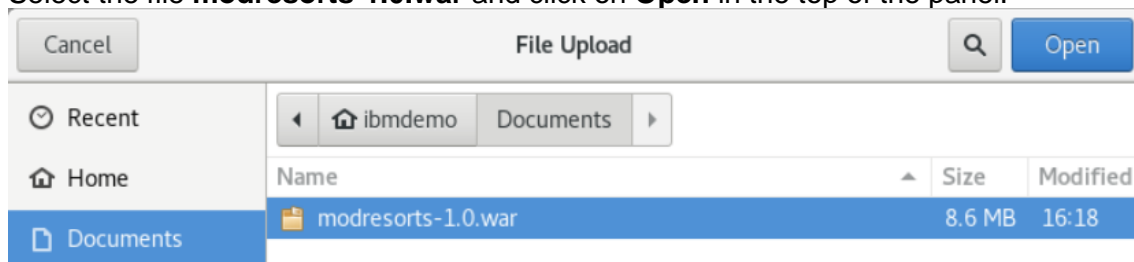
Transformation Advisor detected some application dependencies. Please provide the dependencies in addition to the application binary by manually uploading or supplying your Maven coordinates:

☒ Manual upload

☐ Maven repository

Detected dependencies	Uploaded files
Application binary	<div>Drag or add file</div>

then select in the pop-up panel "**Documents**" and navigate to the Documents folder. Select the file **modresorts-1.0.war** and click on **Open** in the top of the panel.



5. Back on the Migration plan page, click on **Download** to download the migration bundle.

IBM Cloud Transformation Advisor

Home / ... / Java / modresorts.ear / Migration plan

Migration plan

The files included in your migration bundle help you migrate to IBM Open Liberty, create an image, and package your application as a Kubernetes Operator for easy deployment.

Build type ⓘ
Select the type of application you want to build to help Transformation Advisor determine what files to include in the bundle.

☐ Source code
☒ Binary

Application dependencies
Transformation Advisor detected some application dependencies. Please provide the dependencies in addition to the application binary by manually uploading or supplying your Maven coordinates:

☒ Manual upload
☐ Maven repository

Detected dependencies	Uploaded files
Application binary	modresorts-1.0.war

Migration Files
These files are generated by Transformation Advisor to assist in migrating this application:

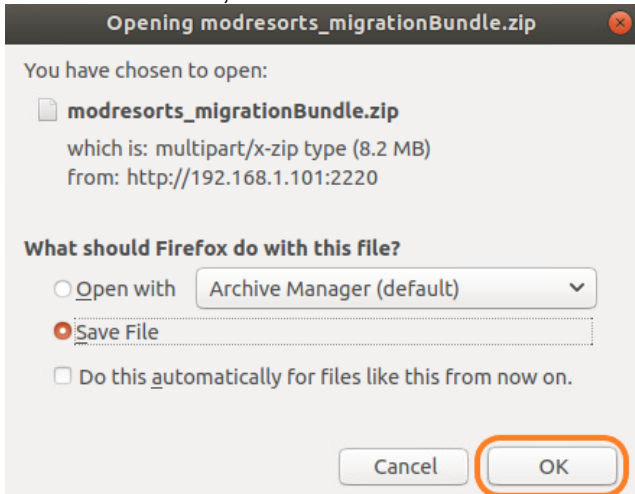
[server.xml](#) ⬇

Send your bundle to Git to begin building and deploying your applications. You can also download the bundle below.

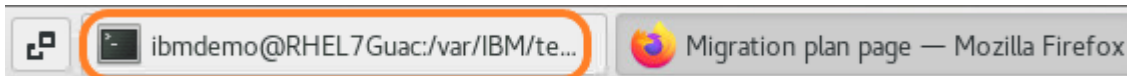
[Send to Git](#)

[Download](#) ⬇ [Send](#) 📧

6. Select **Save File**, then click **OK**.



7. Switch to the command shell.



8. Stop the running WAS Traditional instance
To avoid port conflicts between tWAS and Liberty, let's stop the tWAS instance.
In the terminal window, issue the command below to start the WAS server.

```
/usr/IBM/WAS855ND/profiles/StandaloneSrv1/bin/stopServer.sh server1
```

9. In the command shell, execute the following commands to
- install WebSphere Liberty,
 - extract the migration bundle,
 - create a Liberty server instance,
 - copy the migration assets to Liberty

```

mkdir /var/IBM/temp/modLiberty
cd /var/IBM/temp/modLiberty

# Install Liberty
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.3.jar -acceptLicense .

# Extract the migration bundle
unzip /home/ibmdemo/Downloads/modresorts_migrationBundle.zip

# Create a Liberty instance called modServer
wlp/bin/server create modServer

# Copy the Liberty configuration created by TA to the Liberty instance
cp src/main/liberty/config/server.xml wlp/usr/servers/modServer/
# Copy the application war file to the Liberty instance
cp target/modresorts-1.0.war wlp/usr/servers/modServer/apps/

```

10. In the command shell, execute the following commands to start the Liberty server.

```

# Start the Liberty instance called modServer
wlp/bin/server run modServer

```

You can ignore any errors that the keystore does not exist.

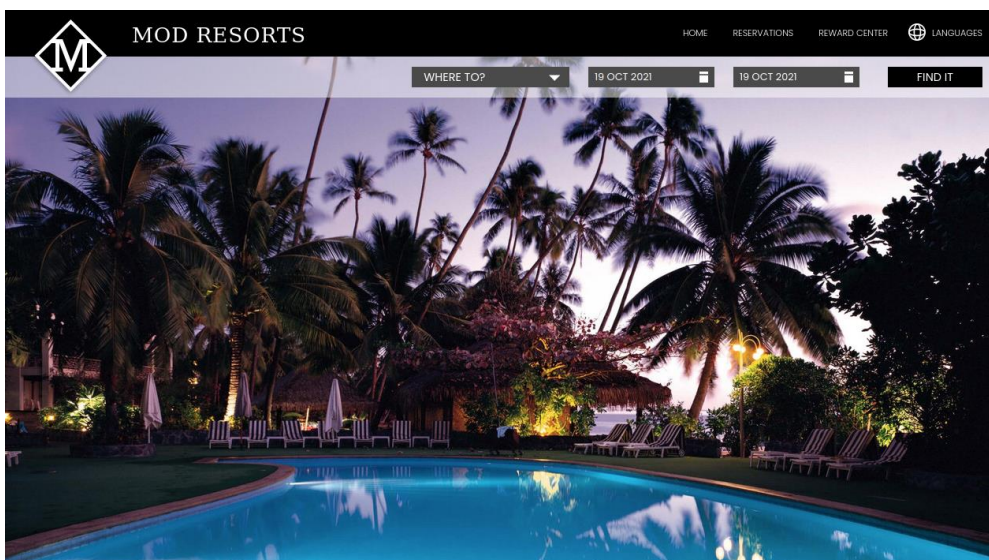
11. Wait until you see the message that server and application have been started.

```

[AUDIT ] CWWKT0016I: Web application available (default host): http://rhel7guac:9080/resorts/
[AUDIT ] CWWKZ0001I: Application modresorts-1.0 started in 1.959 seconds.
[AUDIT ] CWWKF0012I: The server installed the following features: [cdi-1.2, distributedMap-1.0, jndi-1.0, json-1.0, mpConfig-1.2, mpMetrics-1.1, servlet-3.1, ssl-1.0, transportSecurity-1.0].
[AUDIT ] CWWKF0011I: The modServer server is ready to run a smarter planet. The modServer server started in 8.901 seconds.

```

12. Access the modresorts application on Liberty via browser using the URL **localhost:9080/resorts**
You should see something like this:



13. Switch back to the terminal window and press Ctrl-C to stop the Liberty server.

2. Containerize the modresorts application

Now let's create a Liberty container with modresorts.

1. You will use the Dockerfile that has been created by Transformation Advisor.

```
cd /var/IBM/temp/modLiberty
```

2. Open the Dockerfile for example via cat.

```
cat Dockerfile
```

The Dockerfile has instructions to

- use adoptopenjdk as base image
- copy the migration assets from the migration bundle into the image
- use the Open Liberty kernel image
- enhance the Open Liberty image with additional features required by the application
- apply available interim fixes and optimize caching

```
[ibmdemo@RHEL7Guac modLiberty]$ cat Dockerfile
# Generated by IBM TransformationAdvisor
# Thu Nov 04 09:41:44 UTC 2021
```

```
FROM adoptopenjdk/openjdk8-openj9 AS build-stage
```

```
RUN apt-get update && \
    apt-get install -y maven unzip
```

```
COPY . /project
WORKDIR /project
```

```
#RUN mvn -X initialize process-resources verify => to get dependencies from maven
#RUN mvn clean package
#RUN mvn --version
RUN mvn --version
```

```
RUN mkdir -p /config/apps && \
    mkdir -p /sharedlibs && \
    cp ./src/main/liberty/config/server.xml /config && \
    cp ./target/*.war /config/apps/ && \
    if [ ! -z "$(ls ./src/main/liberty/lib)" ]; then \
        cp ./src/main/liberty/lib/* /sharedlibs; \
    fi
```

```
FROM icr.io/appcafe/open-liberty:kernel-slim-java8-openj9-ubi
```

```
ARG TLS=true
```

```
RUN mkdir -p /opt/ol/wlp/usr/shared/config/lib/global
COPY --chown=1001:0 --from=build-stage /config/ /config/
COPY --chown=1001:0 --from=build-stage /sharedlibs/ /opt/ol/wlp/usr/shared/config/lib/global
```

```
# This script will add the requested XML snippets to enable Liberty features and grow image to be fit-for-purpose using featureUtility.
# Only available in 'kernel-slim'. The 'full' tag already includes all features for convenience.
RUN features.sh
```

```
# Add interim fixes (optional)
# COPY --chown=1001:0 interim-fixes /opt/ol/fixes/
```

```
# This script will add the requested server configurations, apply any interim fixes and populate caches to optimize runtime
RUN configure.sh
```

```
# Upgrade to production license if URL to JAR provided
ARG LICENSE_JAR_URL
RUN \
    if [ $LICENSE_JAR_URL ]; then \
        wget $LICENSE_JAR_URL -O /tmp/license.jar \
        && java -jar /tmp/license.jar -acceptlicense /opt/ibm \
        && rm /tmp/license.jar; \
    fi
```

3. Use the above Dockerfile generated by Transformation Advisor to build the Liberty container with modresorts.

Be aware that there is a dot at the end of the command to tell docker build to use the Dockerfile from the current directory.


```
docker build -t modresorts .
```

Finally you should see something like:

```
Successfully built 5dbala556971
Successfully tagged modresorts:latest
```

4. Run Liberty in a container using the command
If there are errors regarding the keystore, you can ignore them.

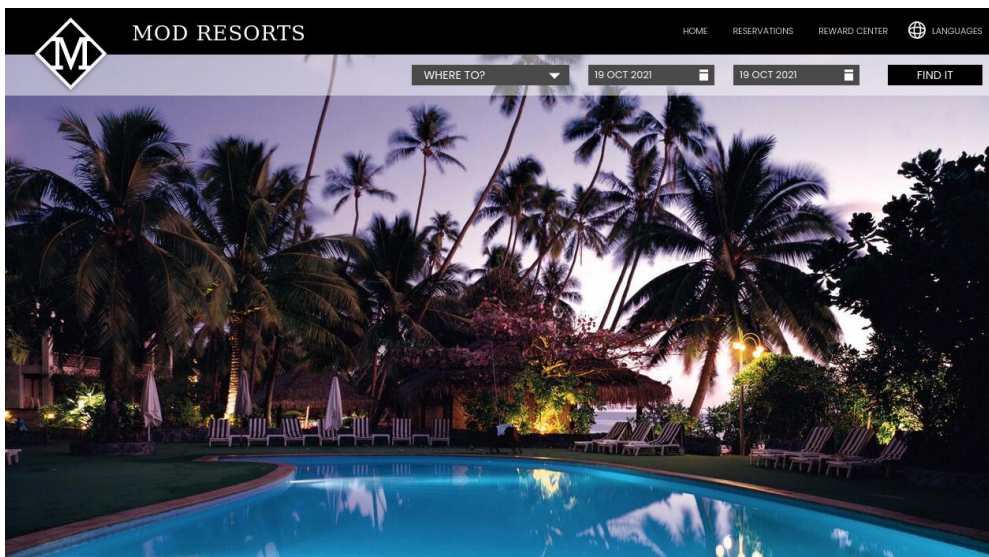
```
docker run -name -p 9080:9080 modresorts:latest
```

5. Wait until you see the message that server and application have been started.

```
[AUDIT ] CWWKT0016I: Web application available (default host): http://fb47510d30ed:9080/resorts/
[AUDIT ] CWWKZ0001I: Application modresorts-1.0 started in 0.376 seconds.
[AUDIT ] CWWKF0012I: The server installed the following features: [cdi-1.2, distributedMap-1.0, jndi-1.0, json-1.0, mpConfig-1.2, mpMetrics-1.1, servlet-3.1, ssl-1.0, transportSecurity-1.0].
[AUDIT ] CWWKF0011I: The defaultServer server is ready to run a smarter planet. The defaultServer server started in 1.552 seconds.
```

The container has been started and mapped from the internal port 9080 to the external port 9080.

6. Access the application from your browser with this link: localhost:9080/resorts/.
You should see something like this:



7. After testing, switch back to the terminal window and press Ctrl-C to stop the container.
8. Close any open browser window, file explorer or command shell in the VM.

Congratulations! You have successfully used the IBM Cloud Transformation Advisor to evaluate an existing WebSphere application and migrate it to Liberty to run standalone or in a container.

===== END OF LAB =====

7. Troubleshooting

1. Transformation Advisor

Access to UI fails:

If the Transformation Advisor GUI cannot be accessed via browser (URL: <http://localhost:3000>), make sure that TA is started. To verify this, open a command shell and run the command

docker ps

The output should be like this:

```
[ibmdemo@RHEL7Guac modLiberty]$ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
0e43f71793e2	icr.io/appcafe/transformation-advisor-ui:3.0.0		"/start.sh &"	3 days ago	Up 5 hours	0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
8d198b7ea734	icr.io/appcafe/transformation-advisor-server:3.0.0		"/opt/ibm/helpers/ru..."	3 days ago	Up 5 hours	9443/tcp, 0.0.0.0:2220->9080/tcp, :::2220->9080/tcp
79e100fald99	icr.io/appcafe/transformation-advisor-db:3.0.0		"/usr/local/bin/tini..."	3 days ago	Up 5 hours	4369/tcp, 5984/tcp, 9100/tcp
1c85a8951a4c	icr.io/appcafe/transformation-advisor-neo4j:3.0.0		"/sbin/tini -g -- /d..."	3 days ago	Up 5 hours	7473-7474/tcp, 0.0.0.0:7687->7687/tcp, :::7687->7687/tcp

```
ibm-transformationAdvisor-Graph
```

If the images are not started, switch to the TA directory and run the launcher with these commands:

```
cd /usr/IBM/TA/transformation-advisor-local-3.0.0
./launchTransformationAdvisor.sh
```

Choose option 5 to start the TA.

```
[ibmdemo@RHEL7Guac modLiberty]$ cd /usr/IBM/TA/transformation-advisor-local-3.0.0/
[ibmdemo@RHEL7Guac transformation-advisor-local-3.0.0]$ ./launchTransformationAdvisor.sh

Prerequisites
-----
Docker installed.
Docker Compose installed.

Status
-----
Transformation Advisor 3.0.0 is available for us at the following URL> http://192.168.1.100:3000

Select the operation.....

1) Install Transformation Advisor
2) Uninstall Transformation Advisor (keep database data)
3) Uninstall Transformation Advisor (remove database data)
4) Stop Transformation Advisor
5) Start Transformation Advisor
6) Check for latest Transformation Advisor
7) Working in an Air Gapped Environment
8) Quit
```

2. Liberty startup fails

If Liberty cannot be started, make sure that the WAS Traditional instance has been stopped.

In the terminal window, issue the command below to stop the WAS server.

```
/usr/IBM/WAS855ND/profiles/StandaloneSrv1/bin/stopServer.sh server1
```

Investigate into the Liberty logs which you can find in

```
/var/IBM/temp/modLiberty/wlp/usr/servers/modServer/logs
```

8. Cleanup

Remove TA collection from the download directory

```
rm ~/Downloads/*  
rm -rf /var/IBM/temp/*  
docker rm modresorts  
docker rmi modresorts:latest
```

Remove collection from TA GUI

9. Summary

Congratulations! You have completed the Transformation Advisor lab.

The insights and assets provided by IBM Cloud Transformation Advisor, enable teams to modernize their applications from traditional application servers like WebSphere Traditional to Liberty on VMs, containers or Kubernetes. The details about mandatory migration steps and the related efforts as well as dependencies help to identify the best applications to start with. Information like the common code analysis provides insight about positive side-effects when migrating one application.

IBM Cloud Transformation Advisor is part of IBM WebSphere Hybrid Edition.

To learn more about IBM WebSphere Hybrid Edition, visit
<https://www.ibm.com/cloud/websphere-hybrid-edition>.

The Total Economic Impact™ Of IBM WebSphere Hybrid Edition: <https://ibm.biz/WSHE-TEI>

10. Appendix

1. Content of the file Lab_TAcommands.txt

```
The latest version of the TAcommands file is available at
https://larsbesselmannibm.github.io/labs/WSHE/lab_TAcommands.txt
If you want to copy it to your local system, use
curl https://larsbesselmannibm.github.io/labs/WSHE/lab_TAcommands.txt >
/var/IBM/temp/lab_TAcommands.txt

# Start tWAS
/usr/IBM/WAS855ND/profiles/StandaloneSrv1/bin/startServer.sh server1

# Create TA workspace with name WASenv1

# Extract collector
mkdir -p /var/IBM/temp/TA_collector/
cd /var/IBM/temp/TA_collector/
tar -zxvf /home/ibmdemo/Downloads/transformationadvisor-Linux_WASenv1.tgz

# Run Collector
cd /var/IBM/temp/TA_collector/transformationadvisor-3.0.0
export JAVA_HOME=/usr/IBM/WAS855ND/java/
bin/transformationadvisor --help

bin/transformationadvisor -w /usr/IBM/WAS855ND/ -p StandaloneSrv1

# Stop tWAS
/usr/IBM/WAS855ND/profiles/StandaloneSrv1/bin/stopServer.sh server1

# Build Liberty Instance with modresorts
mkdir /var/IBM/temp/modLiberty
cd /var/IBM/temp/modLiberty
# Install liberty via archive
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar --acceptLicense .

# Extract migration bundle
unzip /home/ibmdemo/Downloads/modresorts_migrationBundle.zip
# Create Liberty instance
wlp/bin/server create modServer

# Copy application war and Liberty configuration from migration bundle
cp src/main/liberty/config/server.xml wlp/usr/servers/modServer/
cp target/modresorts-1.0.war wlp/usr/servers/modServer/apps/

# Start Liberty instance
wlp/bin/server run modServer

# Access the modresorts application via URL localhost:9080/resorts

# Create a Liberty container
cd /var/IBM/temp/modLiberty
docker build -t modresorts .
docker run --name modresorts -p 9080:9080 modresorts:latest

# Clean up
# Remove TA collection from the download directory
rm ~/Downloads/*
rm -rf /var/IBM/temp/*
docker rm modresorts
docker rmi modresorts:latest
```