IBM Software
WebSphere

Lab: Using IBM Cloud Transformation Advisor

Contents

Contents

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

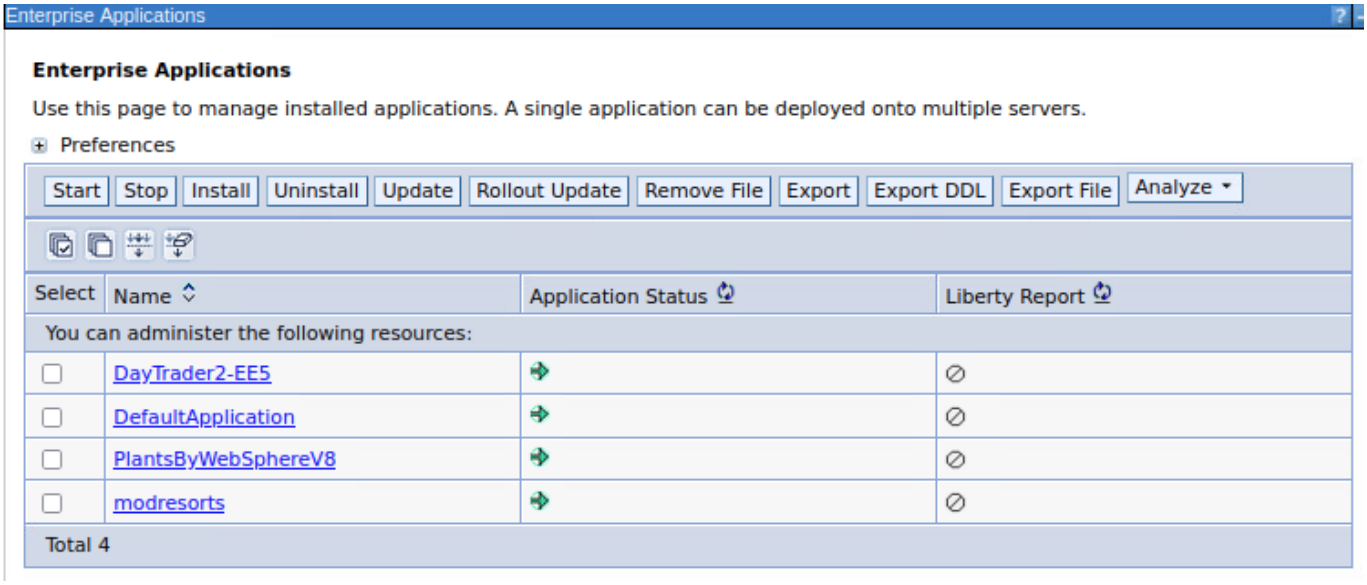
- identify the Java EE programming models in the app.
- determine the complexity of apps by listing a high-level inventory of the content and structure of each app.
- highlight Java EE programming model and WebSphere API differences between the WebSphere profile types
- learn any Java EE specification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

Lab - Using IBM Cloud Transformation Advisor

1.1 Business Scenario

As shown in the image below, your company has several web applications deployed to WebSphere Application Server (WAS) environment.



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.

1.2 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.

1.3 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

1.4 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

1.5 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

1.7 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

Note: To ease the copy and paste, the commands used in the lab have been documented in the file ttps://larsbesselmannibm.github.io/labs/WSHE/lab_TA_commands.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/WSHE/lab_TAcommands.txt >
~/Desktop/lab_WSHE_TAcommands.txt
```

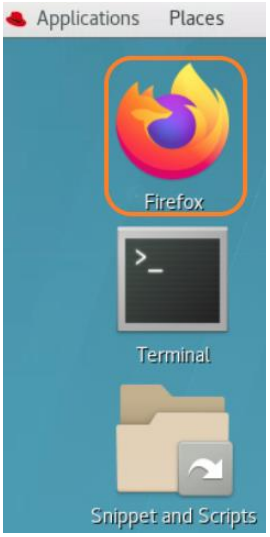
Lab 2 Getting Started with IBM Cloud Transformation Advisor

1. Access the environment
    a. Use the connection details That have been provided to you.
    b. If you are connected via VNC, use the URL https://iccve.uk.ibm.com/cloudhur2.

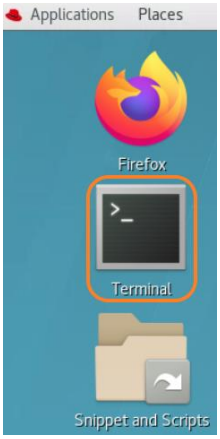
2. Login with **ibmdemo** ID.
    a. If you are connected via VNC, you should be automatically logged in as ibmdemo.

    Otherwise log in as user “ibmdemo” and enter “**passw0rd**” as the password:
    Password: **`passw0rd`** (lowercase with a zero instead of the o)

3. Open the file with the lab commands by click on Firefox and navigating to the URL
`https://larsbesselmannibm.github.io/labs/WSHE/lab_TAcommands.txt`



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



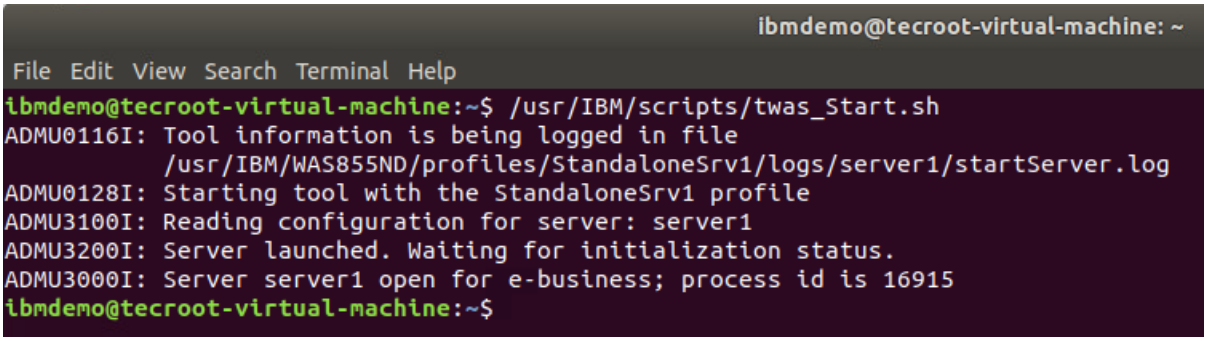
2.1.1 Review the on-prem WebSphere apps

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.

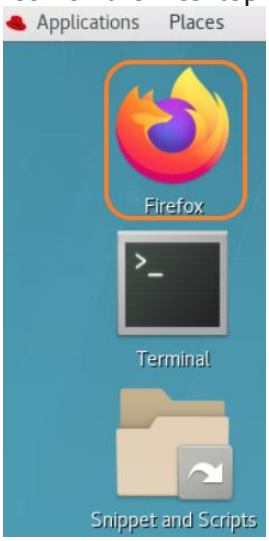
1. Start WebSphere Application Server
   In the workstation VM, you have a local traditional WebSphere Application Server which hosts several sample applications.

   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`
      Alternatively you can run the command **`/usr/IBM/scripts/twas_Start.sh`**

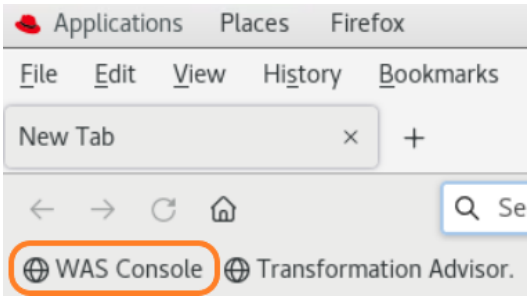
   b. Wait until the server has been started



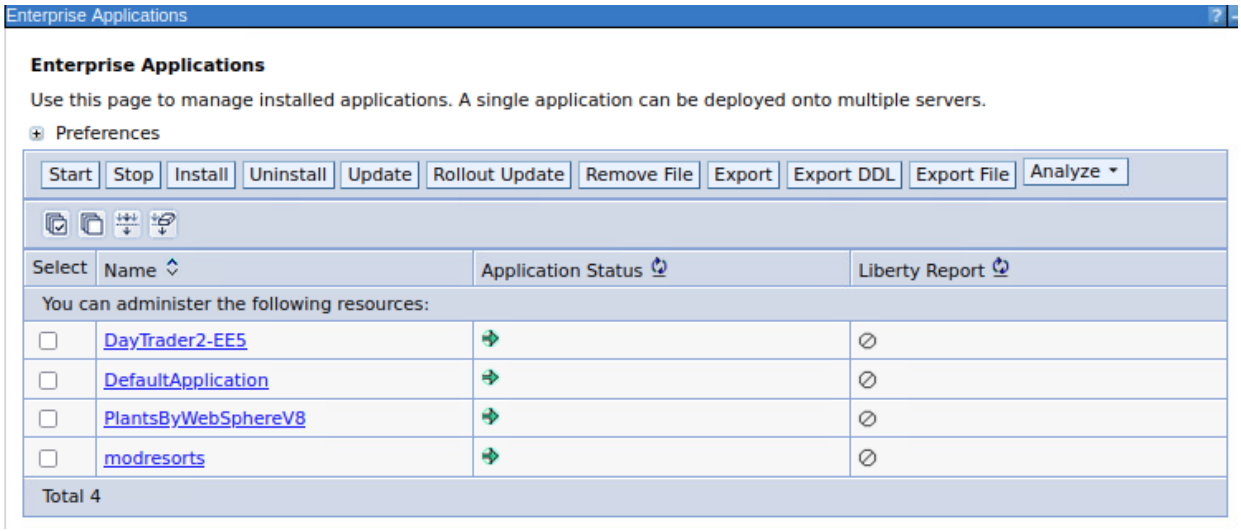
   c. Access the WAS Admin Console to view the application deployed by clicking the Firefox icon on the Desktop toolbar.



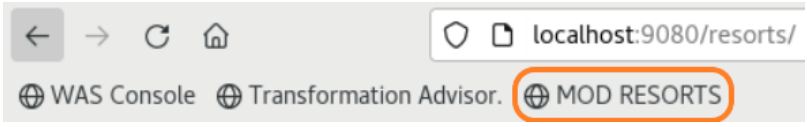
   d. From the web browser window and click the **WAS Console** bookmark to launch the WebSphere Application Server console.



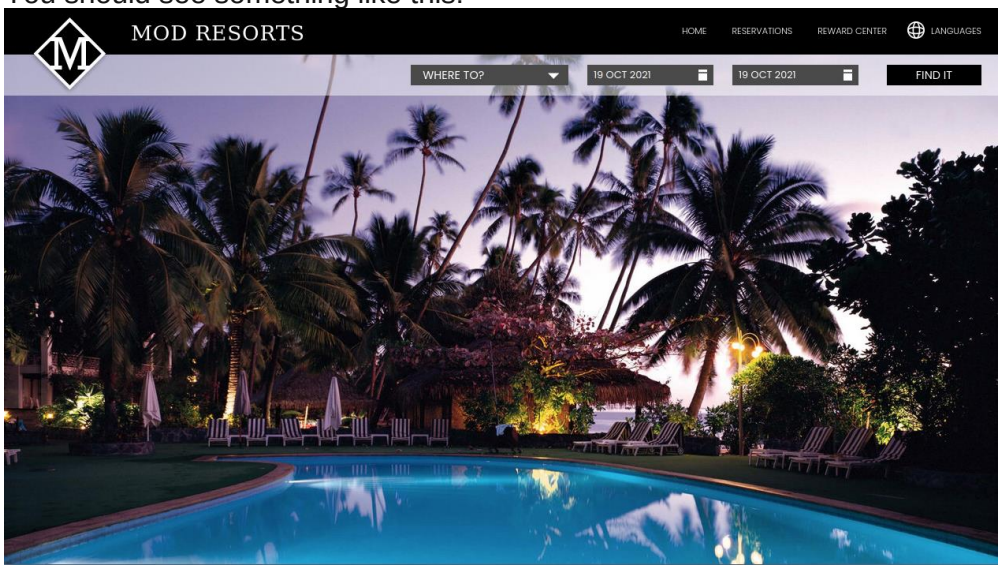
e. If you see the Warning: Potential Security Risk Ahead message, click Advanced>Accept the Risk and continue.
f. In the WAS Admin Console login page, enter the User ID and Password as: **wsadmin/password** and click Login.
g. On the WAS Console page, click Applications -> Application Types -> WebSphere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



i. Next, you use Transformation Advisor to analyze these applications to identify a good candidate to be moved to the cloud.
j. But before that access the application modresorts once via browser to know 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:



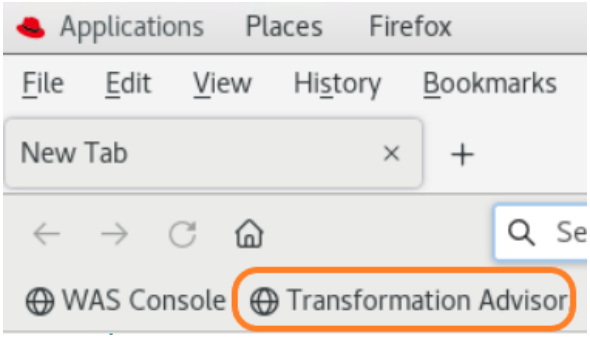
iii. Feel free to navigate around to see how the application looks like.

2.1.2 Access Transformation Advisor

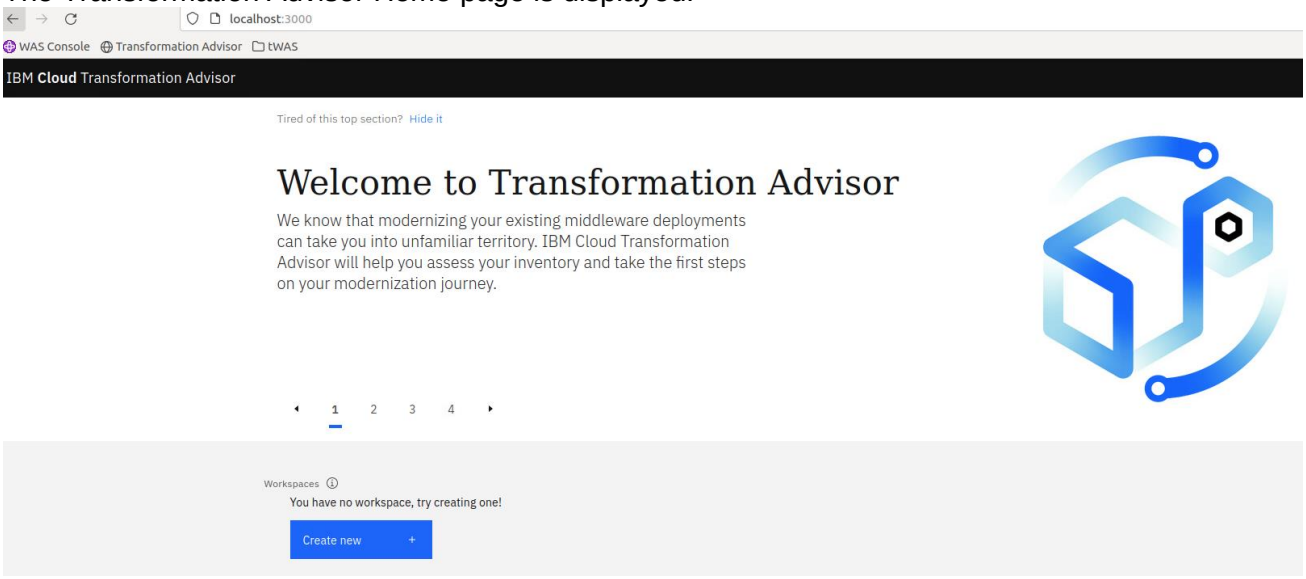
The Transformation Advisor can evaluate any Java based applications. In this lab, you are going to use it to evaluate whether the on-premises WebSphere application, Mod Resorts, is suitable to move to cloud and what the effort might be to get it there. You can use Transformation Advisor Data Collector 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)

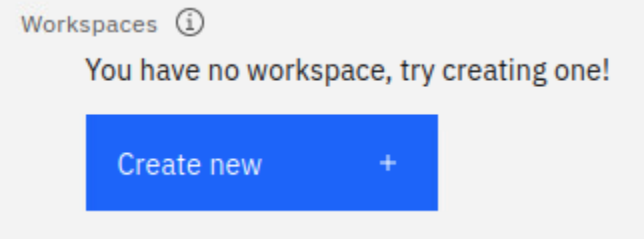


2. The Transformation Advisor Home page is displayed.

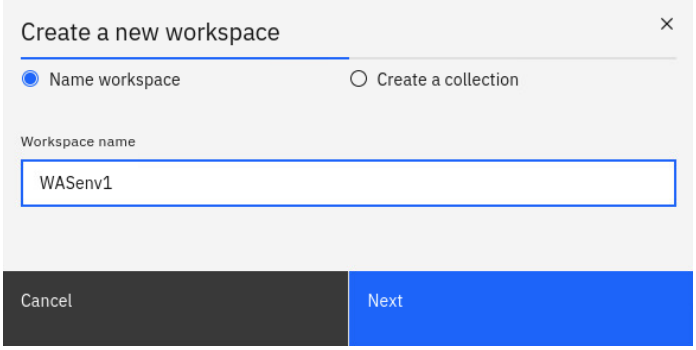


3. In the Transformation Advisor page, you first create a new workspace and then a collection.
   Note: A workspace is a designated area that houses the migration recommendations provided by Transformation Advisor against your application server environment. You can name and organize these however you 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.

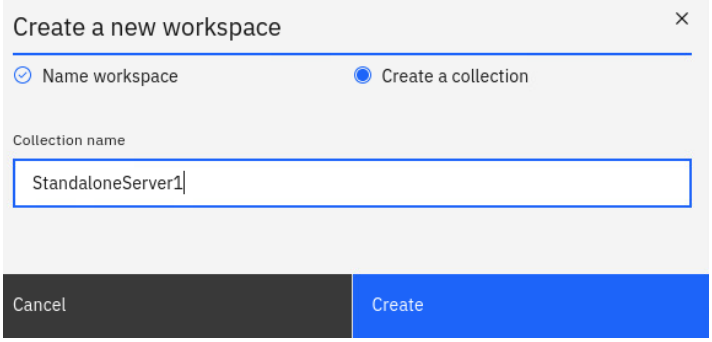
Click on the button **Create new** to create a new workspace



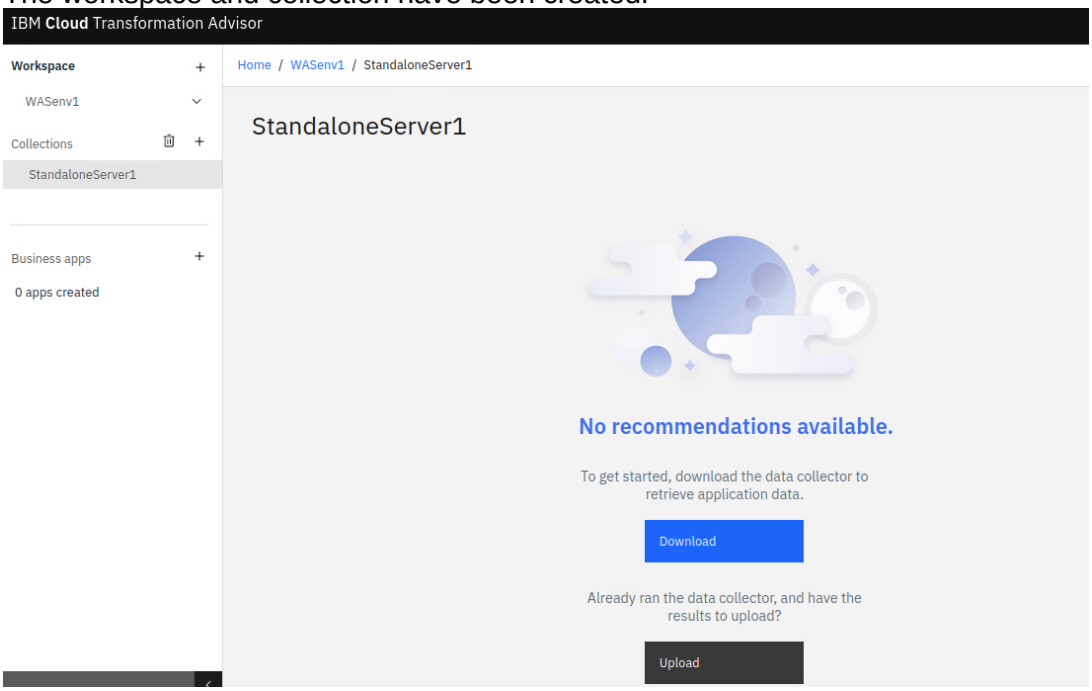
4. Enter as workspace name **WASenv1**, then click **Next**.



5. Enter the collection name as **StandaloneServer1** and click **Create**.

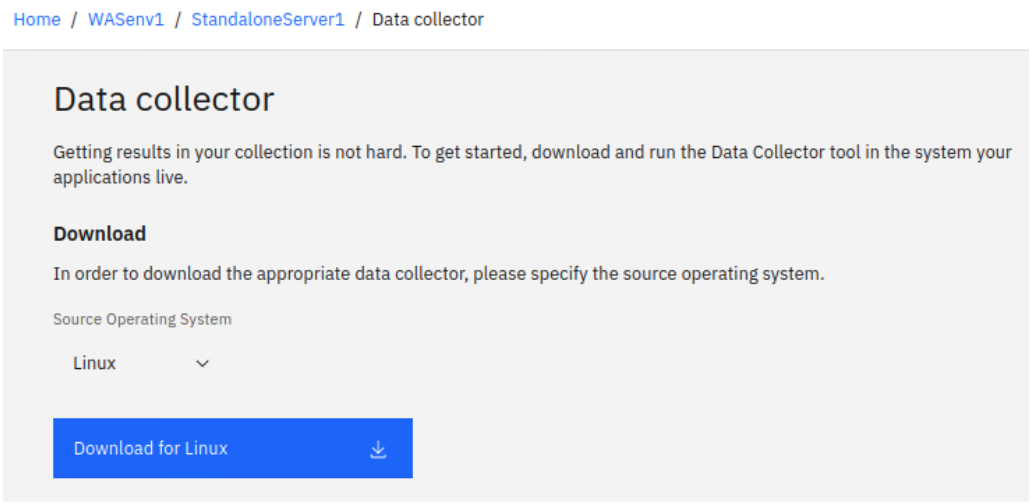


6. The workspace and collection have been created.



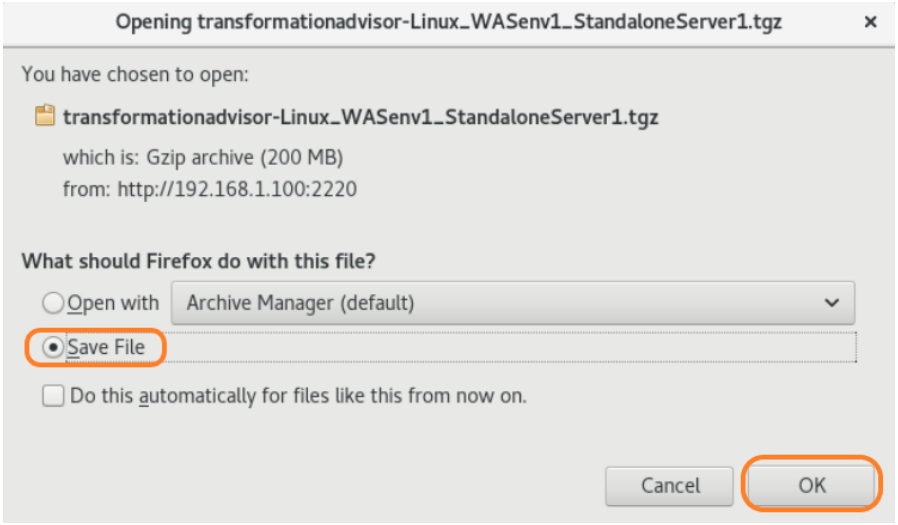
Next step is to use the Transformation Advisor Collector to gather data about your applications. Click on the **Download** button.

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



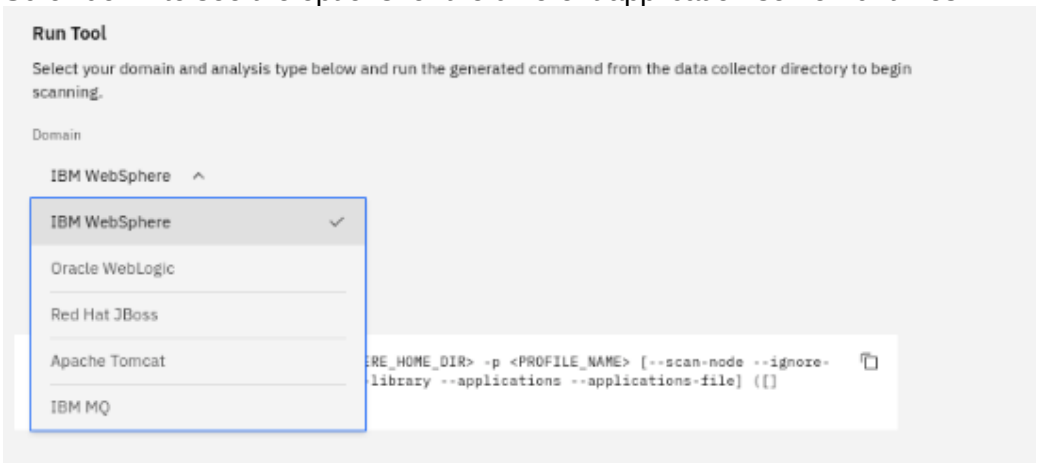
Leave the Operating System to **Linux** and click on the button **Download for Linux**.

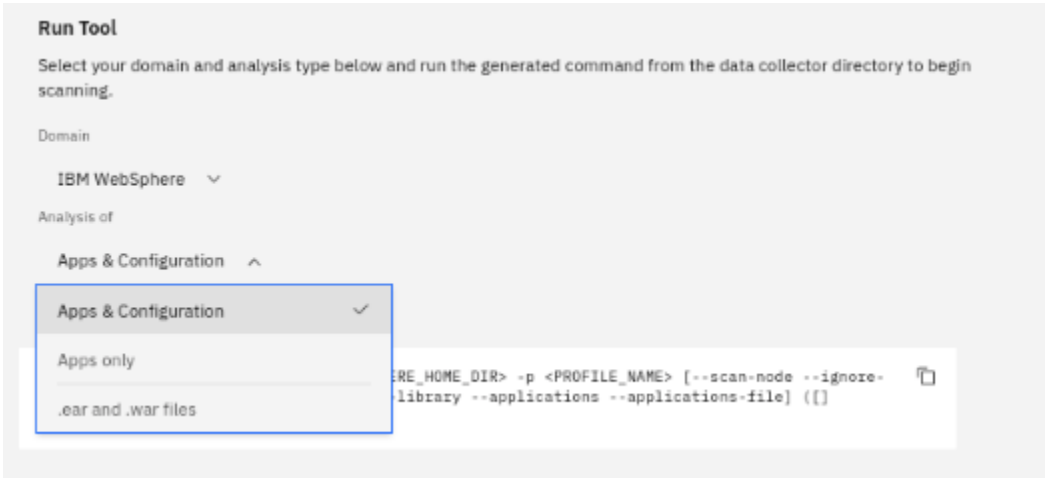
8. Select **Save File** and **OK**



The file is saved in the Download folder **`/home/ibmdemo/Downloads`**,
the name is: **`transformationadvisor-Linux_WASenv1_StandaloneServer1.tgz`**

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

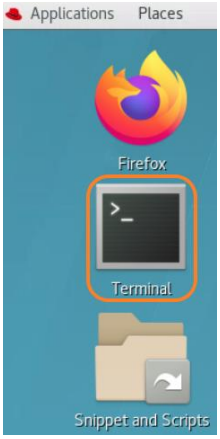




2.1.3 Use the Transformation Advisor collector

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:
   (You can also run instead the command **`/usr/IBM/scripts/unzipCollector.sh`**)

```
mkdir /usr/IBM/TA_collector/
cd /usr/IBM/TA_collector/
tar -zxvf /home/ibmdemo/Downloads/transformationadvisor-
Linux_WASenv1_StandaloneServer1.tgz
```

3. Switch to the collector directory

```
cd /usr/IBM/TA_collector/transformationadvisor-2.5.0
```

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

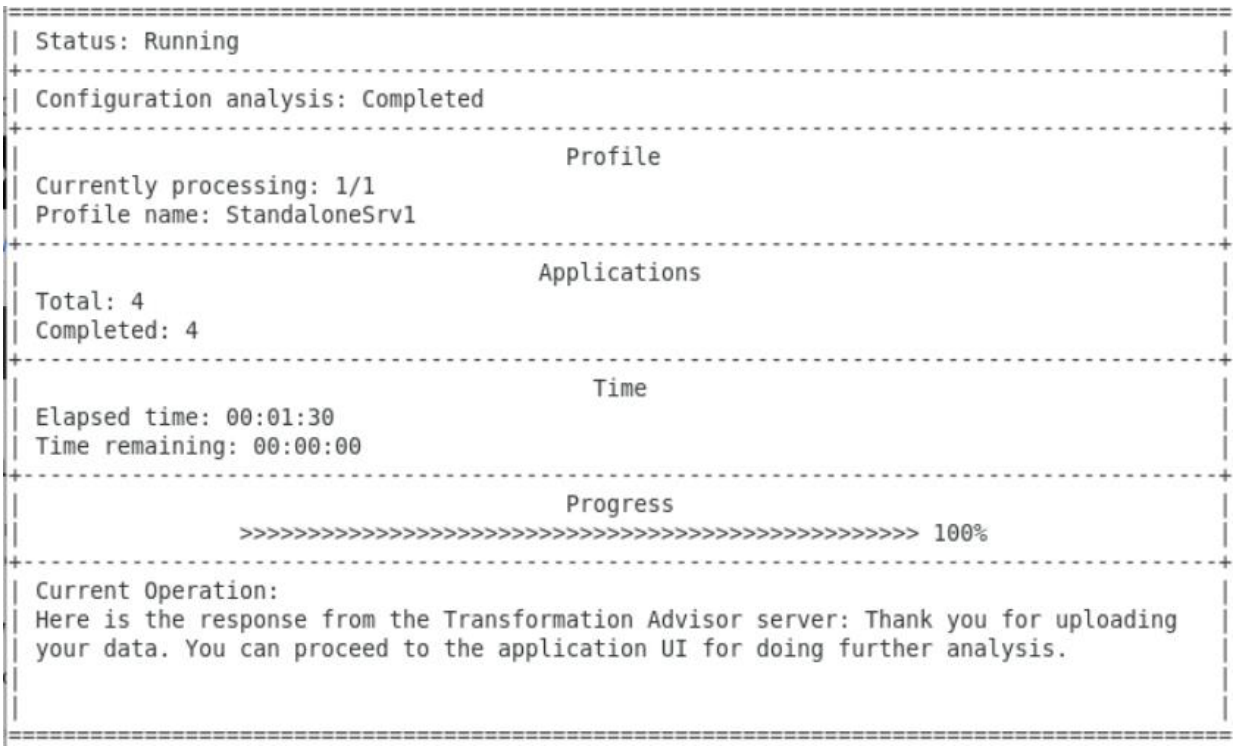
```
bin/transformationadvisor --help
```

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

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

You can also run the following script instead: **`/usr/IBM/scripts/runCollector.sh`**

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



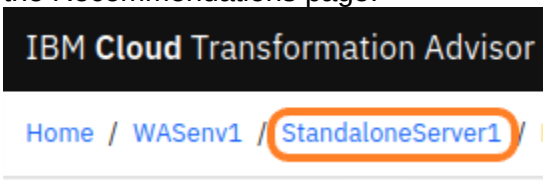
8. The collector creates a collection, a zip archive containing the reports, with the name of the profile, here: /usr/IBM/TA_collector/transformationadvisor-2.5.0/StandaloneSrv1.zip.

9. As the collector can connect to the Transformation Advisor server, it uploads the collection to the server. You could also upload the collection manually using the TA user interface.

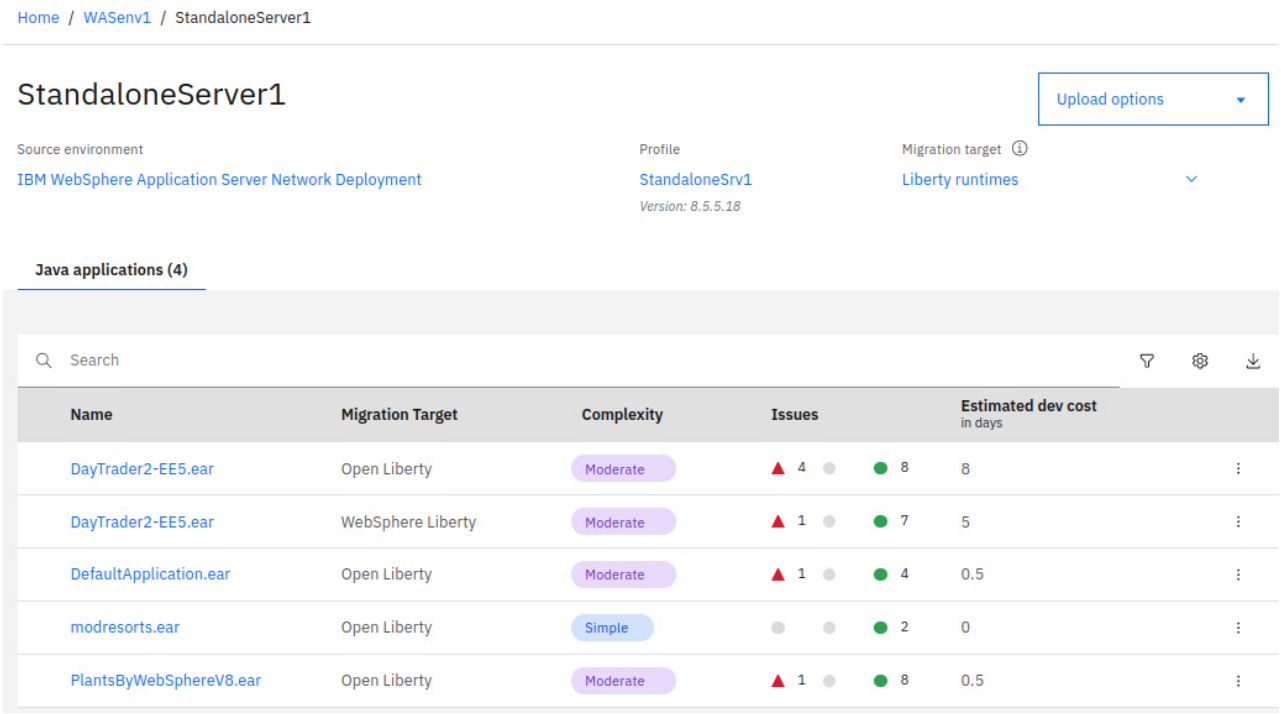
2.1.4 Evaluate On-Premises Java Applications

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, click the StandaloneServer1 link to go to the Recommendations page.



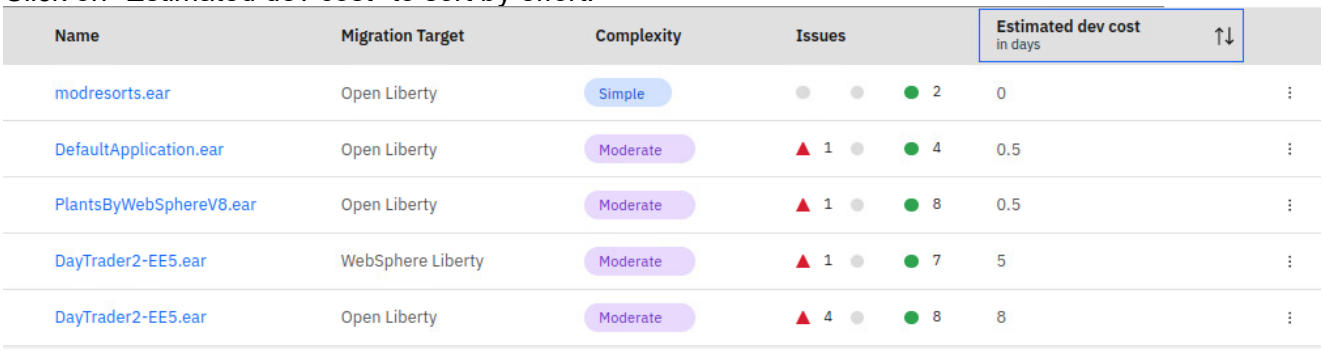
2. In the Recommendations page, you can see all applications deployed to the WAS server are listed.



3. On the Recommendations page, the identified migration source environment including edition and version as well as the profile name is shown in the Profile section. The Migration target is set to Liberty Runtimes, but there are more options available.

   The Recommendations page also shows the summary analysis results for all the scanned apps for the selected migration target. For each app, you can see these results:
   a. Name
   b. Migration Target
   c. Complexity
   d. Dependencies
   e. Issues
   f. Estimated development cost in days

4. Click on “Estimated dev cost” to sort by effort.



Name	Migration Target	Complexity	Issues	Estimated dev cost in days
modresorts.ear	Open Liberty	Simple	2	0
DefaultApplication.ear	Open Liberty	Moderate	1 4	0.5
PlantsByWebSphereV8.ear	Open Liberty	Moderate	1 8	0.5
DayTrader2-EE5.ear	WebSphere Liberty	Moderate	1 7	5
DayTrader2-EE5.ear	Open Liberty	Moderate	4 8	8

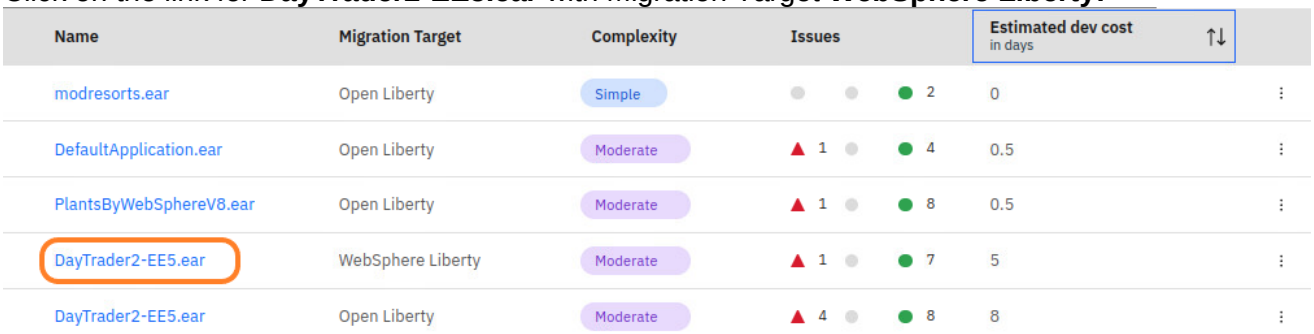
5. Let’s take a closer look at some applications.

   a. If you want to move the **modresorts.ear** application to Open 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.

   b. **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.

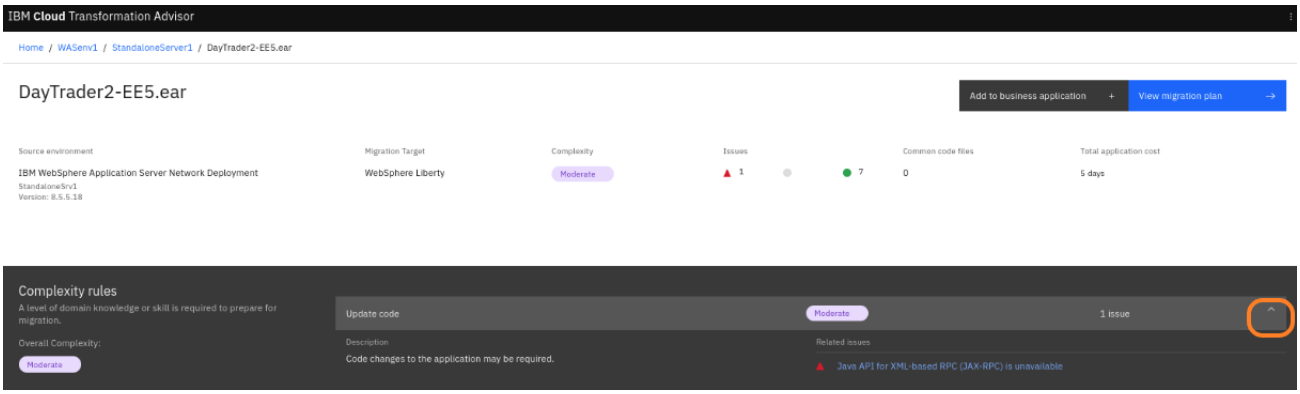
   c. As the name indicates, **DayTrader2-EE5.ear** is an older application that uses some older Java EE standards. Other than for the other applications, the application is listed with the migration target WebSphere Liberty as well as with the migration target Open Liberty. This indicates that there are different migration efforts for the two Liberty runtimes. While both runtimes have a complexity level of Moderate, the estimated migration effort for WebSphere Liberty is less than for Open Liberty.

6. Click on the link for **DayTrader2-EE5.ear** with Migration Target **WebSphere Liberty:**



Name	Migration Target	Complexity	Issues	Estimated dev cost in days
modresorts.ear	Open Liberty	Simple	2	0
DefaultApplication.ear	Open Liberty	Moderate	1 4	0.5
PlantsByWebSphereV8.ear	Open Liberty	Moderate	1 8	0.5
DayTrader2-EE5.ear	WebSphere Liberty	Moderate	1 7	5
DayTrader2-EE5.ear	Open Liberty	Moderate	4 8	8

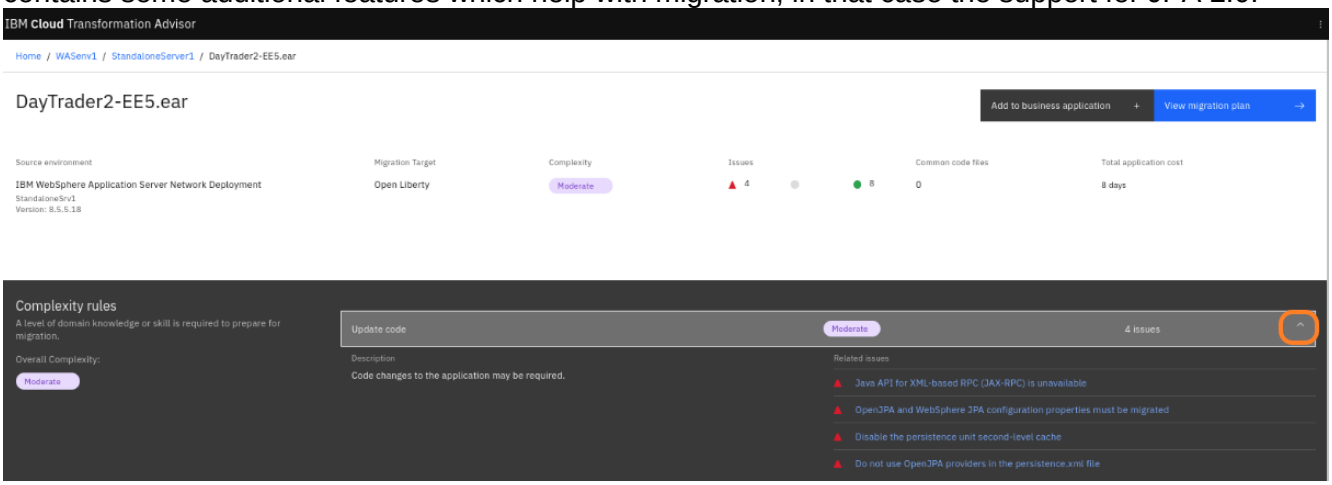
7. The Application Details Page opens and displays more information about the efforts.
   As you can see, the main effort is around JAX-RPC (you might have to open the twisty).



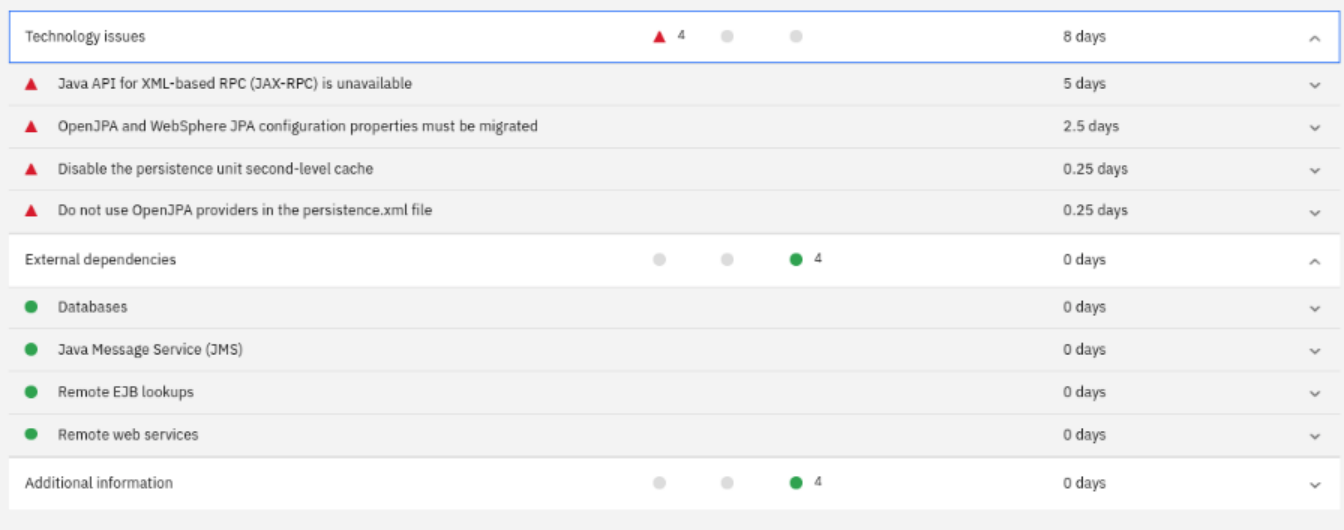
8. In the browser go back to the previous screen and click on the link for **DayTrader2-EE5.ear** with Migration Target **Open Liberty**

Name	Migration Target	Complexity	Issues	Estimated dev cost in days
modresorts.ear	Open Liberty	Simple	2	0
DefaultApplication.ear	Open Liberty	Moderate	1 4	0.5
PlantsByWebSphereV8.ear	Open Liberty	Moderate	1 8	0.5
DayTrader2-EE5.ear	WebSphere Liberty	Moderate	1 7	5
DayTrader2-EE5.ear	Open Liberty	Moderate	4 8	8

9. The Application Details Page opens and displays more information about the efforts.
   As you can see, there are additional issues next to the one around JAX-RPC (again you might have to open the twisty).
   Reason for the difference between Open Liberty and WebSphere Liberty is, that WebSphere Liberty contains some additional features which help with migration, in that case the support for JPA 2.0.



10. Scroll down a bit to see additional information gathered by the collector.
    Click on **Technology issues**, to see the migration issues with estimated development efforts in more detail, then click on **External dependencies**.



As you can see under external dependencies, the application depends on databases, messaging systems and accesses other systems via Remote EJB lookups and Remote web services. The dependencies help you to identify side effects when moving into containers or into a public cloud for example. Feel free to open for each issue and dependency the related twisty to get more insight.

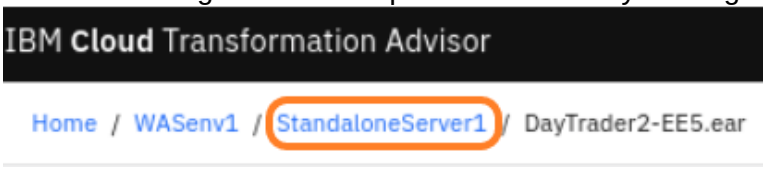
11. Scroll down to the bottom to see the additional reports.

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

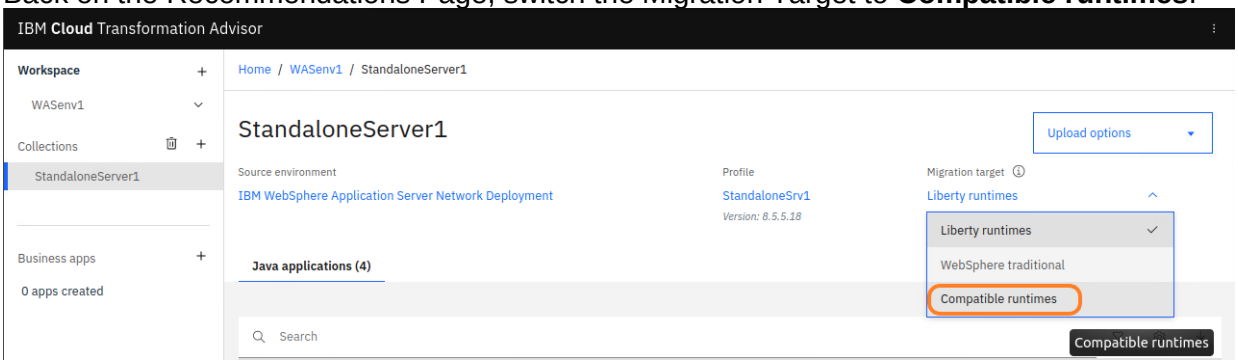
a. the issues (Analysis report),
b. the application structure and deployment topology (Inventory report)
c. the target runtime (Technology report).

Feel free to open the reports to get some idea about the content.

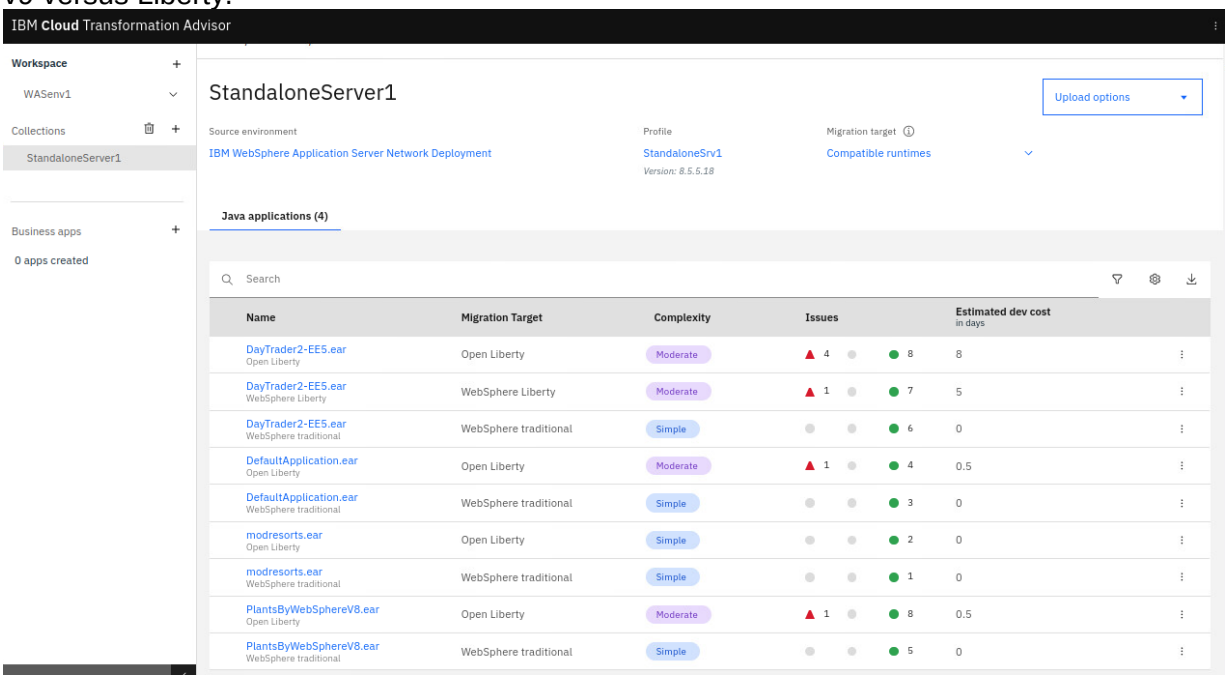
12. In the browser go back to the previous screen by clicking on the link for the collection.



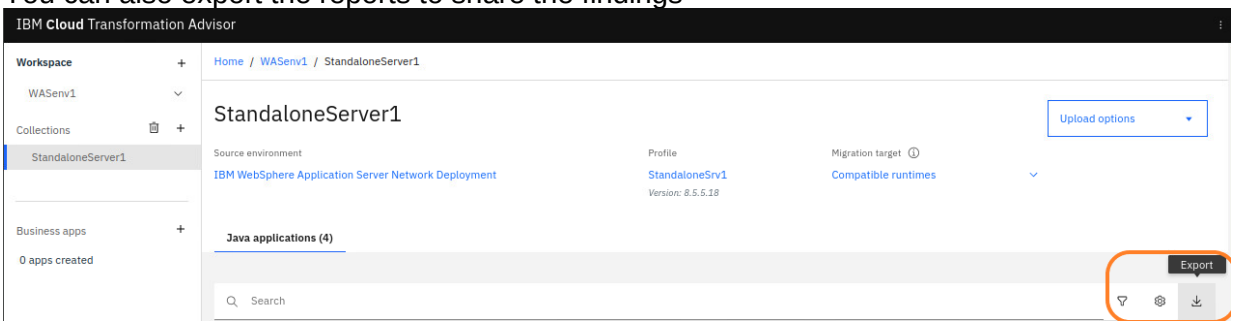
13. Back on the Recommendations Page, switch the Migration Target to **Compatible runtimes**.



14. Now you can easily compare for all applications the estimated migration efforts for traditional WAS v9 versus Liberty.



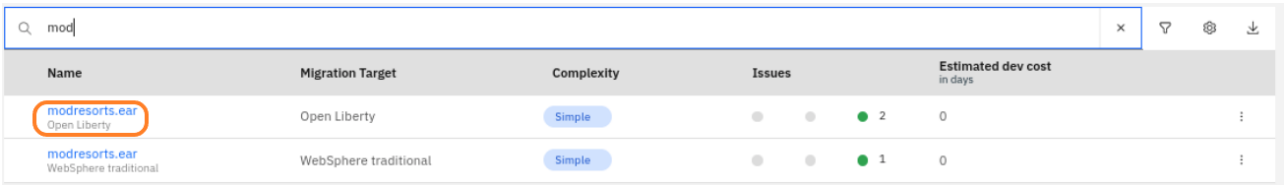
15. You can also export the reports to share the findings



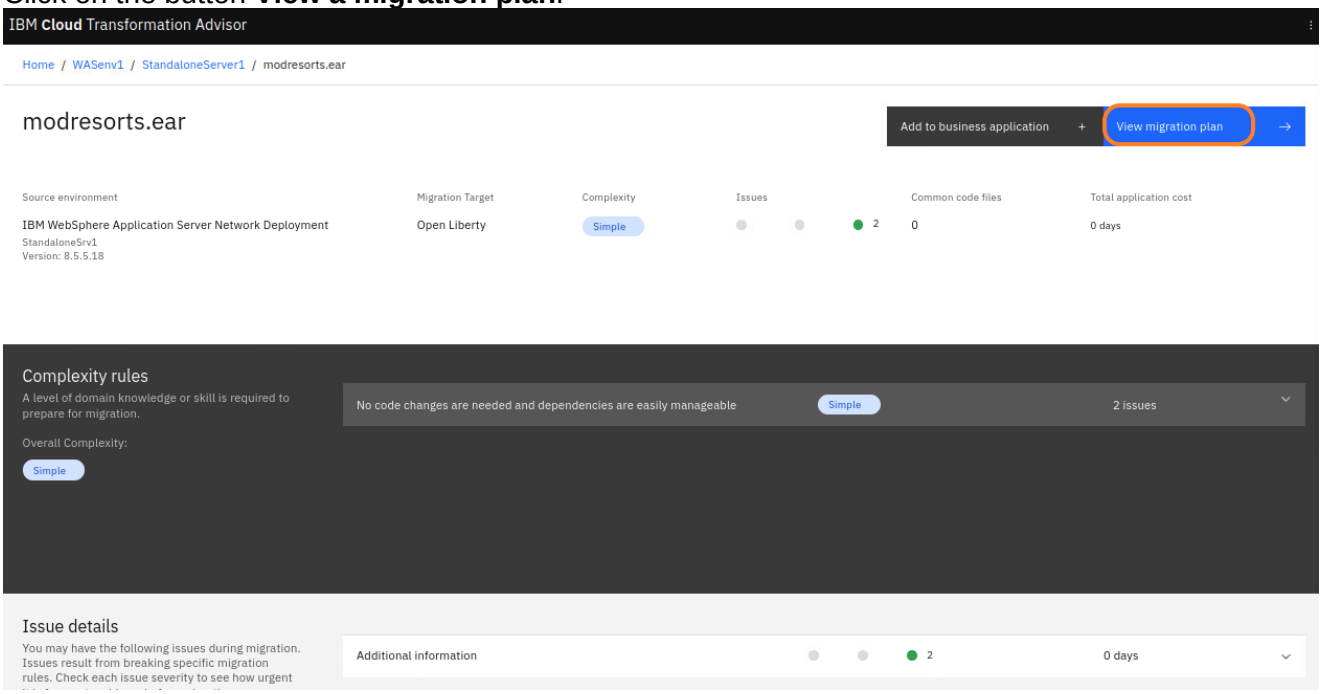
2.1.5 Migrate the modresorts application

As you can see, for modresorts.ear the estimated development efforts for WAS traditional and Liberty are both zero, so let's try to migrate the application to Liberty.

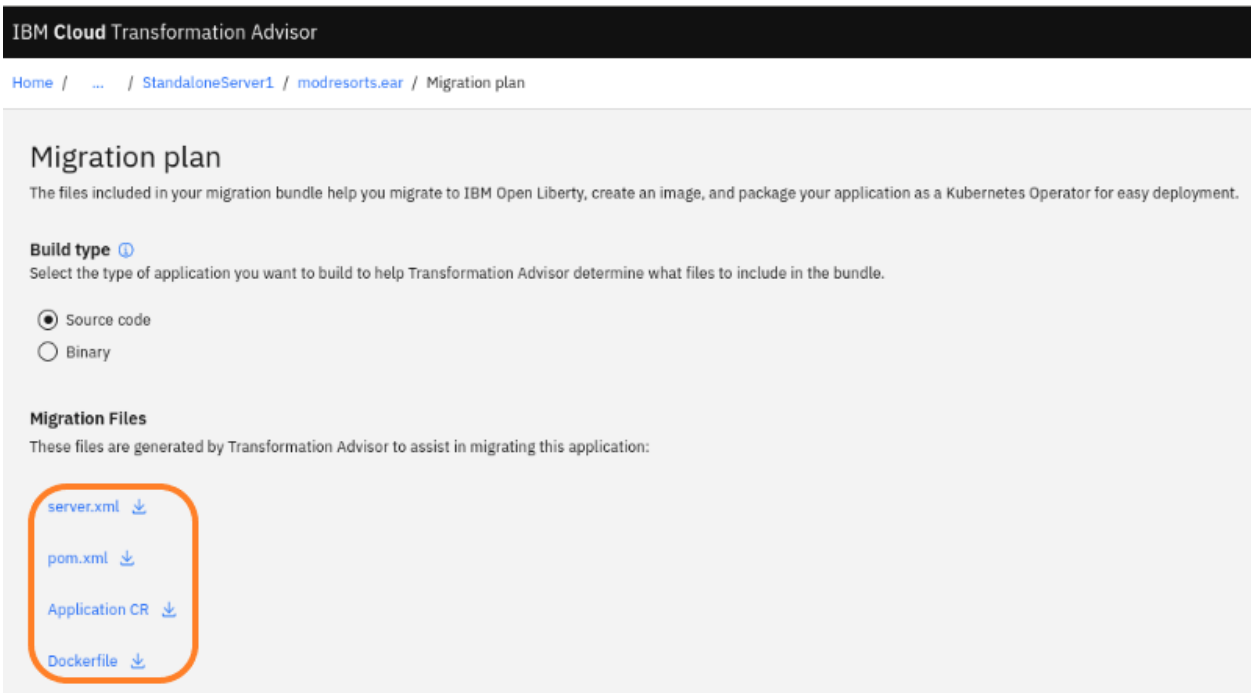
1. Click on the application modresorts.war for target Open Liberty to see the details



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**.

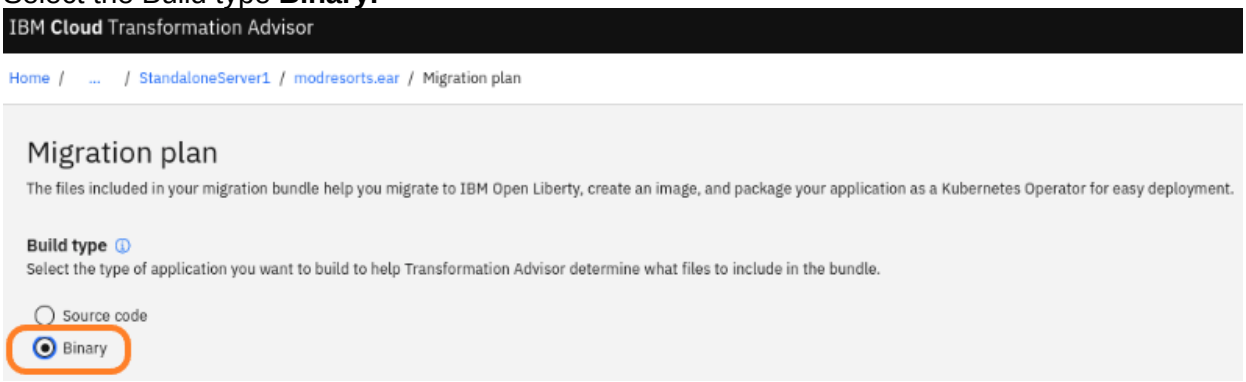


3. Transformation Advisor generates several assets which help to migrate to Liberty, into containers and Kubernetes.

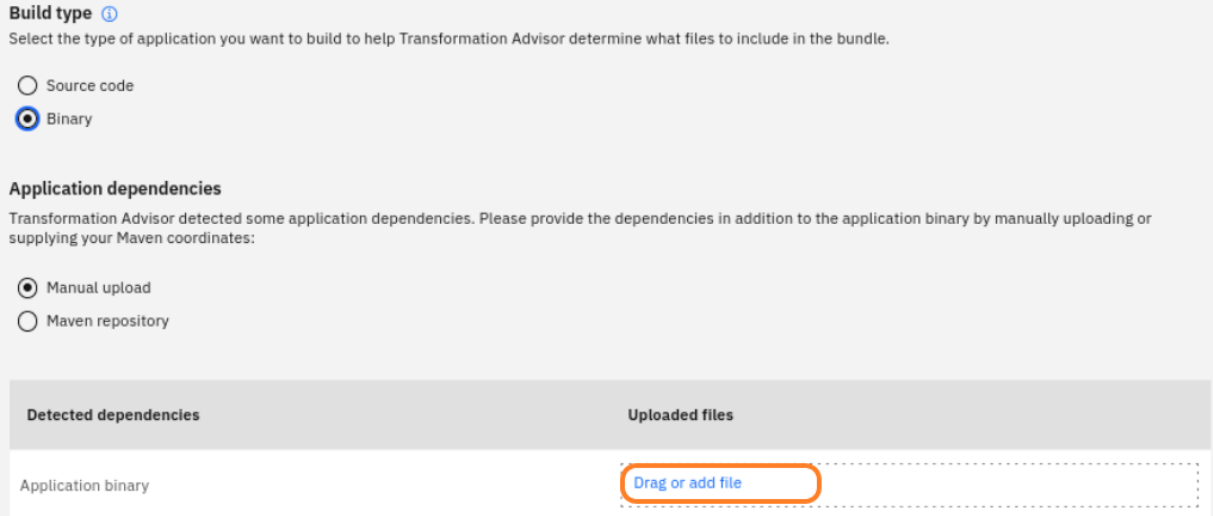


a. The file server.xml contains the Liberty configuration extracted from tWAS.
b. The file pom.xml helps with the integration into maven build
c. The custom resources Application CR helps to deploy to Kubernetes
d. Dockerfile helps to containerize the application

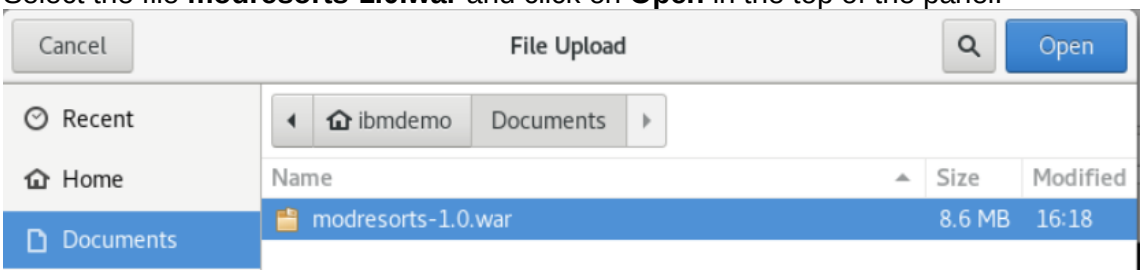
Select the Build type **Binary.**



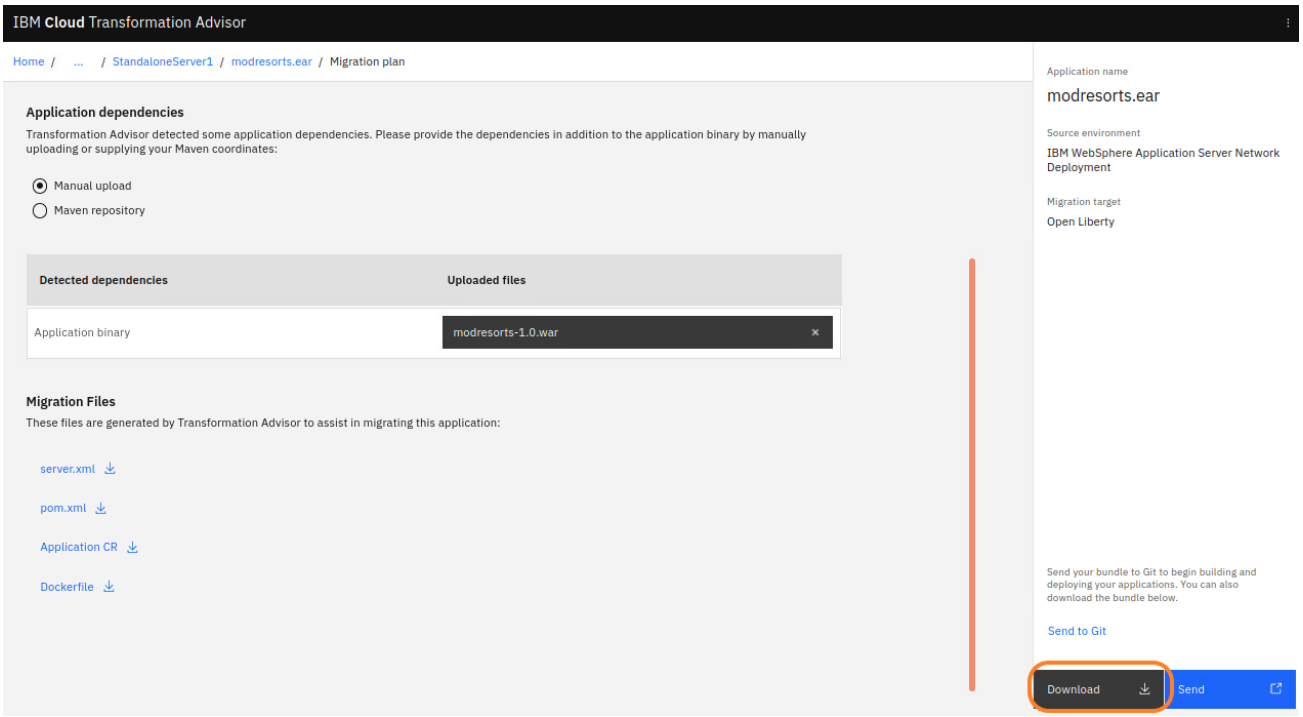
4. Upload the modresorts-1.0.war file by clicking in the related field



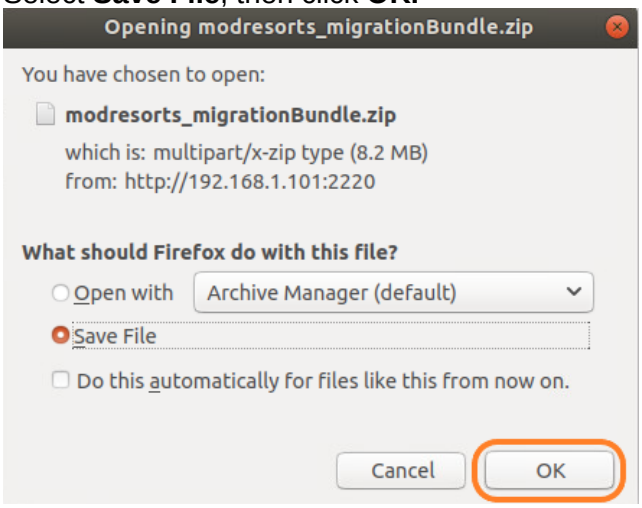
then select in the pop-up panel "**Documents**" and navigate to the Documents folder. Select the file **modresorts-1.0.war** and click on **Open** in the top of the panel.



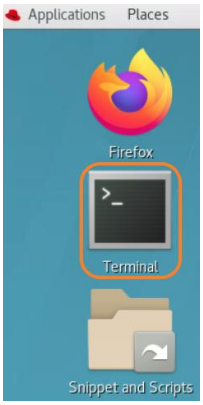
5. Back on the Migration plan page, click on **Download** to download the migration bundle.



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
```

   Alternatively you can run the command `/usr/IBM/scripts/twas_Stop.sh`

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
   - start the Liberty server.

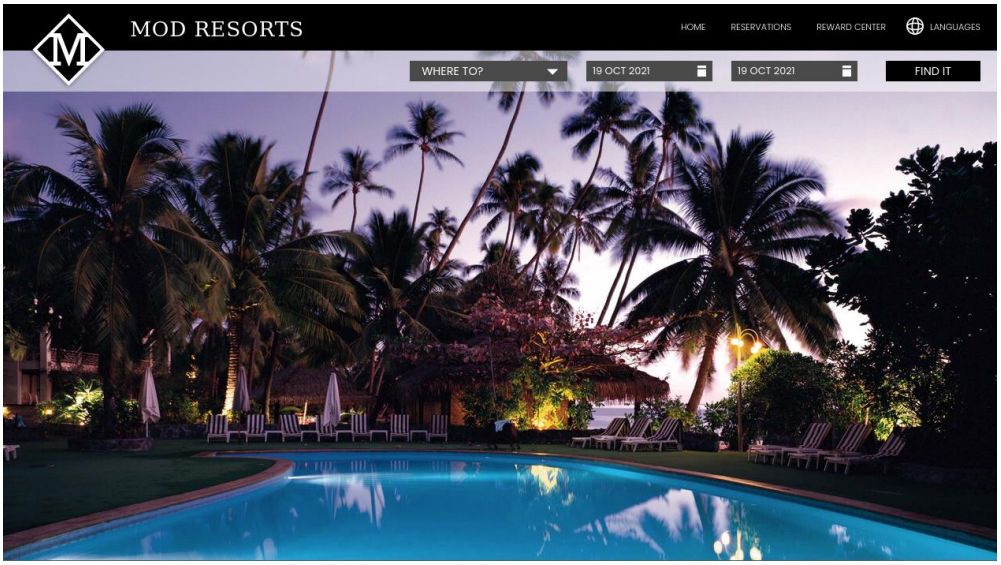
   You can also run instead the command **`/usr/IBM/scripts/buildLiberty.sh`**
   During Liberty installation, you have to accept the license agreement (press 'x', 'x', '1') and accept the default for the directory by pressing **Enter**.

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

# Install Liberty – accept the license agreement and the defaults provided
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.3.jar

# 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/dropins/

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

   You can ignore any errors that the keystore does not exist.
10. 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, mpM
etrics-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.
```

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



12. Switch back to the terminal window and press Ctrl-C to stop the Liberty server.
13. Now let's create a Liberty container with modresorts.
    You will use the Dockerfile that has been created by Transformation Advisor.

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

14. if you are interested in the Dockerfile, open it for ex, 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/*.*ar /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
```

15. 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 5dba1a556971
    Successfully tagged modresorts:latest
    ```

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

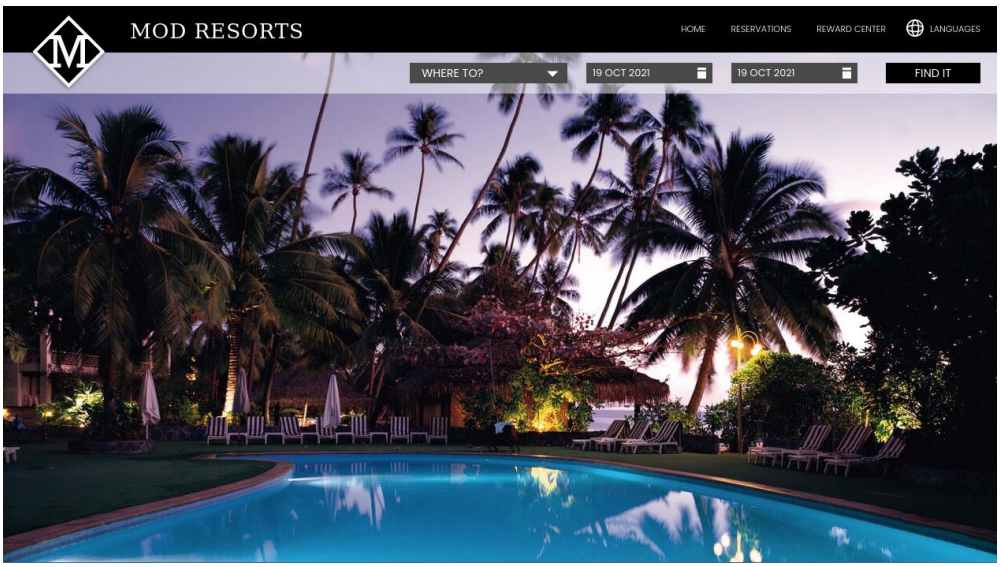
    ```
    docker run –name -p 9080:9080 modresorts:latest
    ```

17. 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, mpM
etrics-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.

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



19. After testing, switch back to the terminal window and press Ctrl-C to stop the container.
20. 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 =====

2.2 Troubleshooting

2.2.1 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`

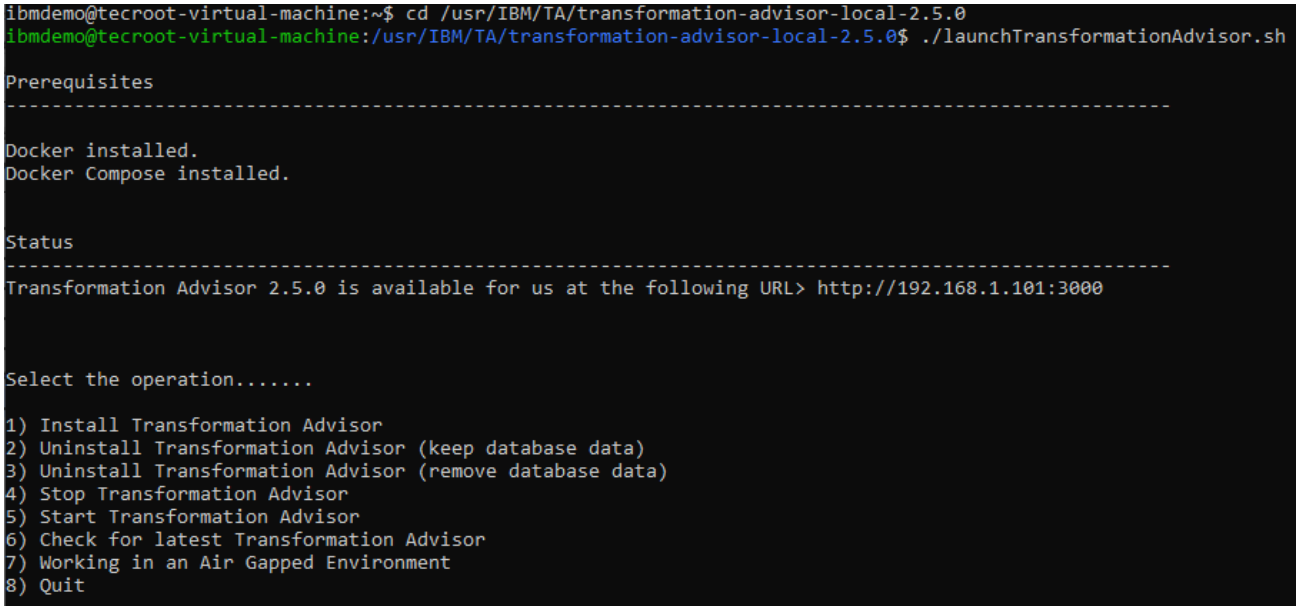
The output should be like this:

```
ibmdemo@tecroot-virtual-machine:~$ docker ps | grep trans

b471dac9c4fe   icr.io/appcafe/transformation-advisor-ui:2.5.0       "./start.sh &"           21
hours ago   Up 21 hours   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
ibm-transformationAdvisor-UI

f65a4c2de9c0   icr.io/appcafe/transformation-advisor-server:2.5.0   "/opt/ibm/helpers/ru…"   21
hours ago   Up 21 hours   9443/tcp, 0.0.0.0:2220->9080/tcp, :::2220->9080/tcp
ibm-transformationAdvisor-Server

dbd7b6d069ff   icr.io/appcafe/transformation-advisor-db:2.5.0       "/usr/local/bin/tini…"   21
hours ago   Up 21 hours   4369/tcp, 5984/tcp, 9100/tcp
ibm-transformationAdvisor-couchDB
```

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-2.5.0

./launchTransformationAdvisor.sh
```

Choose option 5 to start the TA.



2.2.2 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
Alternatively you can run the command **/usr/IBM/scripts/twas_Stop.sh**

Investigate into the Liberty logs which you can find in

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

2.3 Cleanup

Remove TA collection from the download directory

```
$ rm  ~/Downloads/*

$ rm -rf /var/IBM/temp/*

$ docker rm modresorts

$ docker rmi modresorts:latest

$ rm -rf /usr/IBM/TA_collector

Remove collection from TA GUI
```