AM0210ST - Modernize a Java Application using IBM Transformation Advisor and Cloud Pak for Applications

Contents

Modernize a Java EE application to IBM Cloud Pak for Applications on OpenShift using IBM Cloud Transformation Advisor

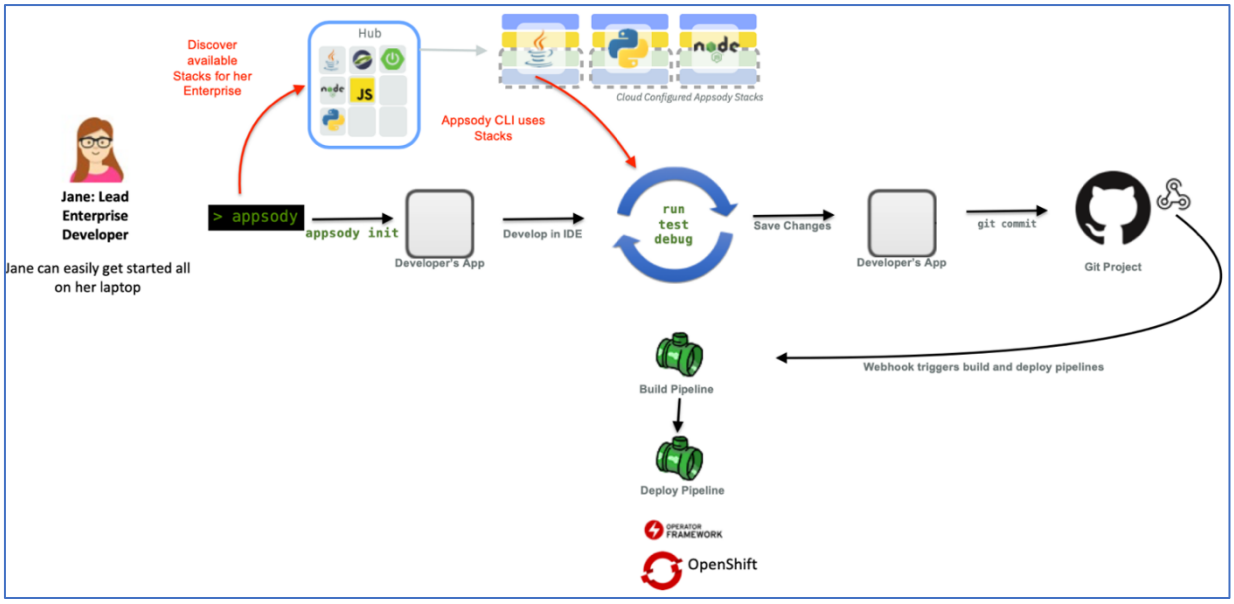
On the journey to cloud, enterprise customers are facing challenges moving their existing on-premises applications to the cloud quickly and cost-effectively. There are new cloud-native skills required for cloud deployments, including new cloud runtime configuration and administration, containerized development and testing, and Kubernetes orchestration.

The **IBM Cloud Pak for Applications** (**CP4Apps**) provides a complete and consistent experience and solution to modernize enterprise applications for cloud-native deployments. Customers can easily modernize their existing applications with IBM's integrated tools and develop new cloud-native applications faster for deployment on any cloud.

One of the features included in the Pak is **Accelerators for Teams,** a modern microservices-based framework that enables developers, architects and operation teams to work together, faster on end-to-end solutions for the team to build, deploy, and manage the lifecycle of Kubernetes-based applications.

IBM Cloud Transformation Advisor, also called **Transformation Advisor** (**TA**), bridges the gap between existing development activities and cloud-native development together with **Accelerators for Teams** in **CP4Apps**. **TA** will accelerate the development process by creating the configuration and deployment artifacts for an application, then facilitate the developer's use of the enhanced developer productivity tools in **CP4Apps**. For more information about **IBM Cloud Pak for Applications**, please visit Cloud Pak for Applications.

This lab brings the cloud-native development experience from the "Create and Deploy an Insurance Quote App to OpenShift with IBM Cloud Pak for Applications" lab, which develops a cloud native microservice application using the nodejs-express and java-spring-boot Application Stacks, to the modernization of existing application. For reference, the cloud-native development process using Application Stacks simplifies the process of building and deploying applications to Kubernetes:



Like the cloud-native experience, **Transformation Advisor** will simplify the modernization scenario, creating one development experience for both cloud-native and application modernization. This lowers the on-ramp to cloud development, test, deployment for Java EE developers.

1. Objective

The objective of this lab is to learn how to use **Transformation Advisor** together with the **Accelerators for Teams** in the **IBM Cloud Pak for Applications** to accelerate the development activities required during the modernization of a Java EE application currently deployed to IBM WebSphere Application Server, Oracle WebLogic, Apache Tomcat or Red Hat JBoss Application Server. The application will be deployed to the **Open Liberty** server in the **IBM Cloud Pak for Applications**, running on Red Hat OpenShift Container Platform (**RHOCP**).

The sample application used in this lab does not require any modification to the source code in its modernization. This enables the completion of the lab in a pre-determined amount of time. The modernization of applications that need modifications will be described, but not required for completion of the lab.

2. Prerequisites

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

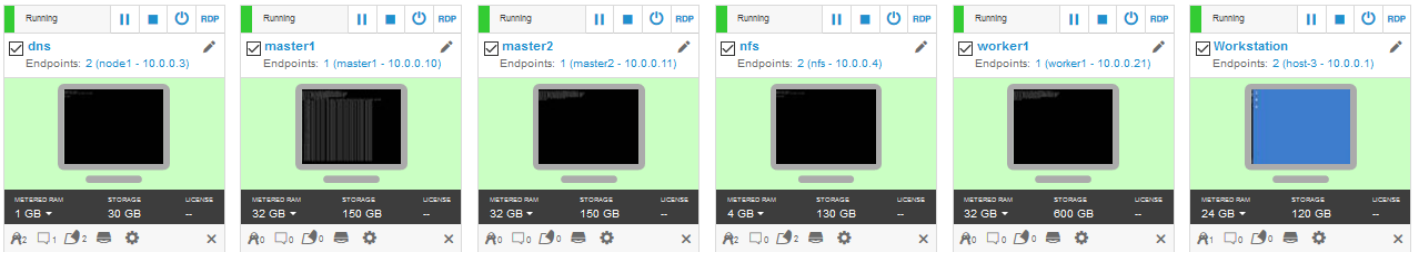
- Familiarity with basic Linux commands
- Have internet access
- Basic Java app development knowledge
- GitHub account - you can sign up for a free account at: https://github.com

The following symbols appear in this document at places additional guidance is available:

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step but is helpful or good to know.
	Trouble-shooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

3. What is Already Completed

Six Linux VMs have been provided for this lab.



- The Red Hat OpenShift Container Platform (**RHOCP**) v4.3, is installed in 4 VMs, the **master1** VM, the **master2** VM, the **dns** VM, and the **worker2** VM, with 2 master nodes and 3 compute nodes (the master nodes are serving as computer nodes as well).
- The **IBM Cloud Pak for Applications** (**CP4Apps**) v4.1 is installed in the **RHOCP**. For information on how to install IBM **Cloud Pak for Applications** on OpenShift, please visit: https://www.ibm.com/support/knowledgecenter/SSCSJL_4.x/welcome.html
- The Appsody client (v0.5.9) is installed on the **Workstation**.
- The **Workstation** VM is the one you will use to access and work with **RHOCP cluster** in this lab. The login credentials for the **Workstation** VM are:
  User ID: **ibmdemo**
  Password: **passw0rd**
  Note: Use the Password above in the **Workstation** VM Terminal for **sudo** in the Lab.
- The CLI commands used in this lab are listed in the **Commands.txt** file located at the **/home/ibmdemo/cp4a-labs/am0210st** directory of the **Workstation** VM for you to copy and paste these commands to the Terminal window during the lab.

4. Lab Tasks

During this lab, you will complete the following tasks:

1. Login to the Workstation VM and open Transformation Advisor
2. View the Transformation Advisor insights for a collection of applications and queue managers
3. Load a data collection for the **Mod Resorts** application currently deployed on-premise in WebSphere Application Server v9
4. View the migration bundle contents for the **Mod Resorts** application
5. Push the migration bundle into GitHub
6. Clone the GitHub repository on the local workstation and import the source for the application
7. Build and package the application in a Liberty container in Codewind using Application Stacks
8. Commit and push the application to the GitHub repository
9. Define the Tekton PipelineResources for the application's build and deployment
10. Deploy the application to OpenShift via the Tekton pipeline

Known Issues

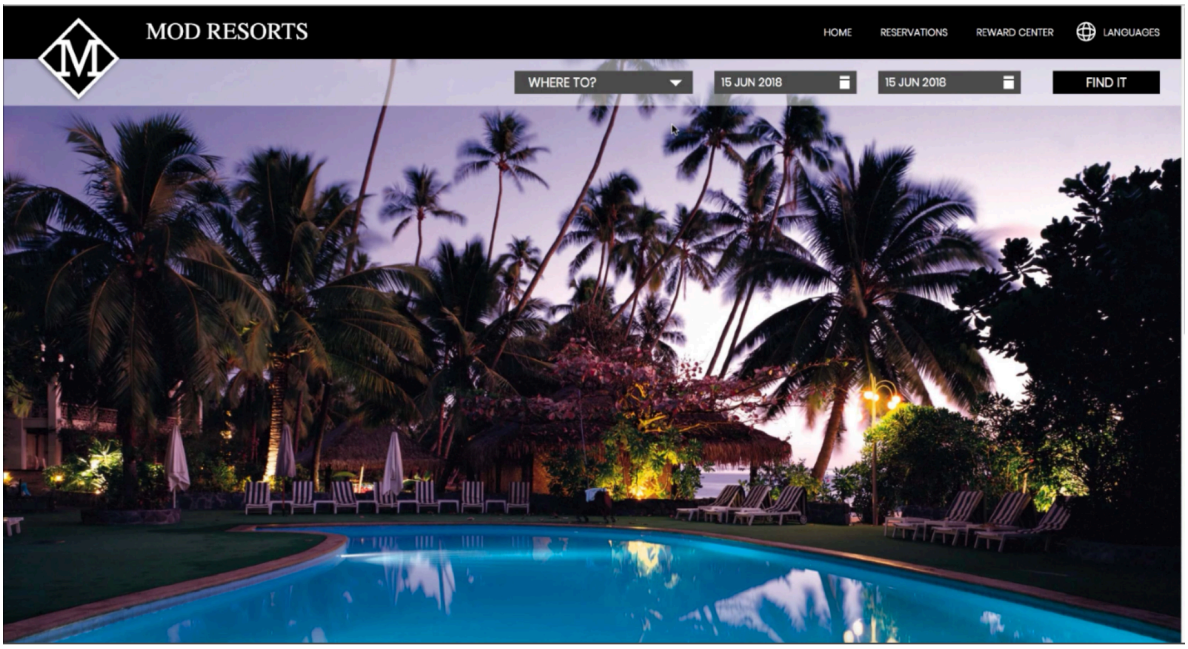
There are some known issues with the SkyTap environment and configuration used by this lab. The lab instructions will workaround these issues and flag them in the lab instructions with the **Important** icon:

- **Artifacts may not be correct if the user switches to / from and then back to Accelerator for Teams** in the migration plan within Transformation Advisor. This is a known bug and will be fixed in the next release of IBM Cloud Transformation Advisor. This lab will workaround this issue by exiting the migration plan view and re-entering the view.

- **GitHub webhooks cannot reach the Tekton pipelines** from outside of the SkyTap environment in the lab configuration of Skytap. This lab will workaround this issue by manually triggering the execution of the Tekton pipelines.

5. Business Scenario

As shown in the image below, your company has a web application called **Mod Resorts**, a WebSphere application showing the weather in various locations. Your company wants to move this application from on-premises to the cloud.



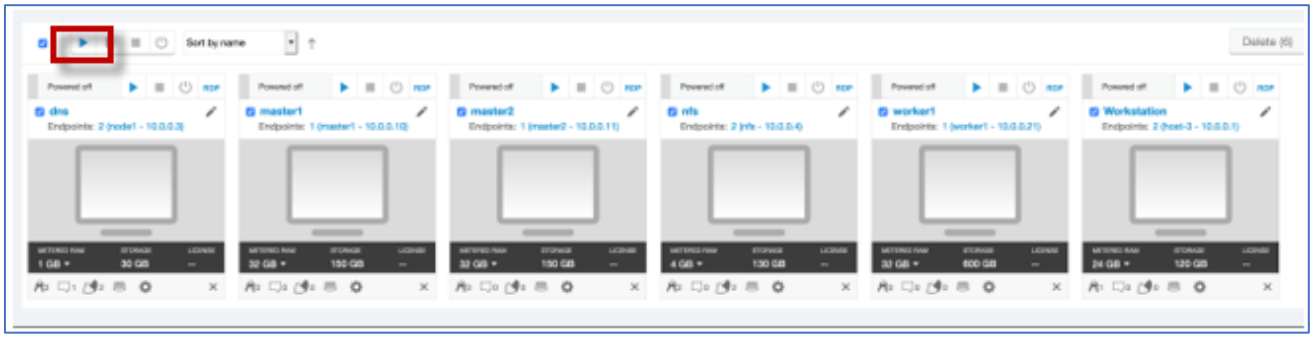
As a tech lead, you have already analyzed the application using the **Transformation Advisor** tool. Based on the analysis you know that you can move this application from the traditional WebSphere Server environment to a light-weighted Liberty server environment without any code change.

Now you are planning to use the **Transformation Advisor** migration plan to create the migration bundle, to containerize the application, and to deploy the Docker container to an OpenShift Kubernetes cluster environment.

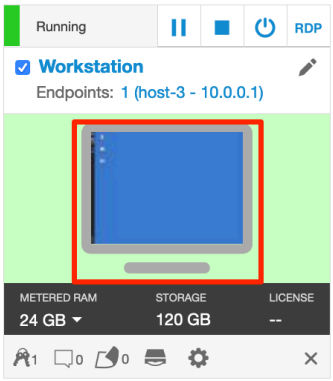
6. Execute Lab Tasks

6.1 Login to the Workstation VM and open Transformation Advisor

__1. If the VMs are **not** already started, start them by clicking the **play** button for the whole group.



__2. After the VMs are started, click the **Workstation** VM icon to access it.



The **Workstation** Linux Desktop is displayed. You will execute all the lab tasks on this VM.

__3. If requested to login to the OS, use credentials: **ibmdemo** / **passw0rd**

6.2 Launch Transformation Advisor

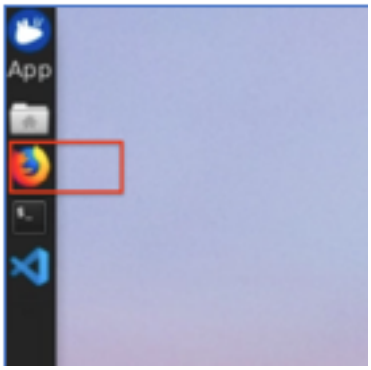
In this lab, you are going to use Transformation Advisor's capabilities designed to accelerate the development process by creating the configuration and deployment artifacts for an application.

Using these artifacts, and **Accelerators for Teams**, you will learn how to facilitate the use of the enhanced developer productivity tools in **IBM Cloud Pak for Applications** as the application is deployed to RedHat OpenShift, using the Tekton pipelines.

The Transformation Advisor used in the lab is installed as part of IBM Cloud Pak for Applications version 4.1. IBM Transformation Advisor is also available stand-alone (Beta), run locally in Docker on your workstation. Learn more about the stand-alone version of Transformation Advisor: **http://ibm.biz/cloudta**

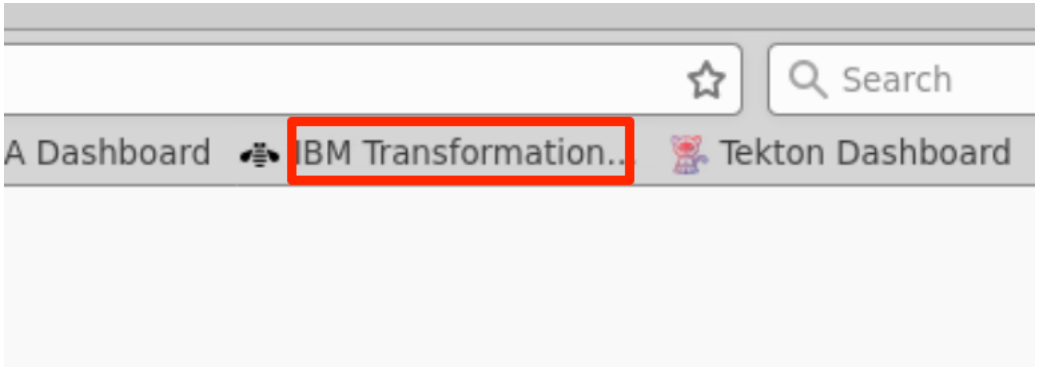
Launch the Transformation Advisor tool using the steps below.

__1. From the **Workstation** desktop, open a Firefox window by clicking its icon on the Desktop Tool Bar.

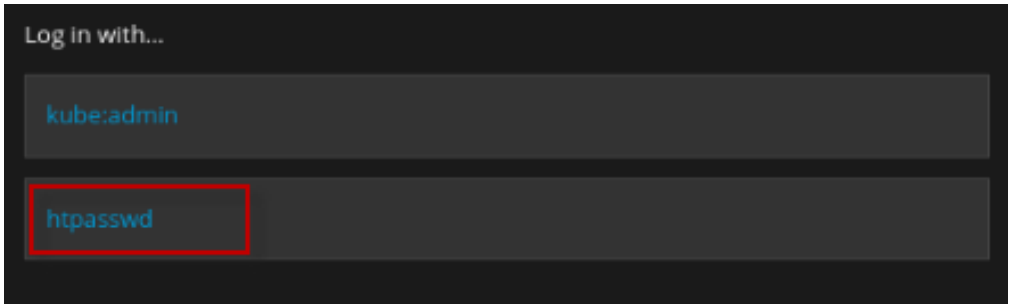


__2. Access the IBM Cloud Transformation Advisor that was installed as a part of IBM Cloud Pak for Applications in Red Hat OpenShift Container Platform in the **Firefox** browser window.

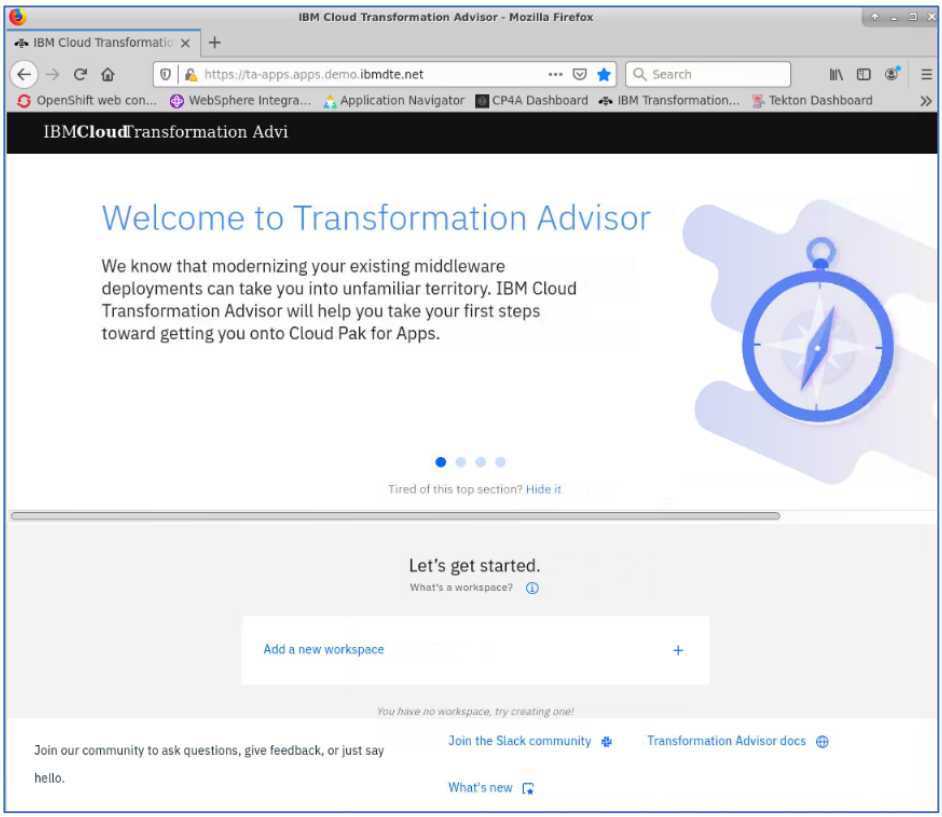
__a. Click on the **Transformation Advisor** bookmark
(or type the URL: **https://ta-apps.apps.demo.ibmdte.net/** in the address bar):



__b. If prompted to login to Red Hat OpenShift Container Platform, click **htpasswd** field. Then login with **ibmadmin** / **engageibm** as the username and password.

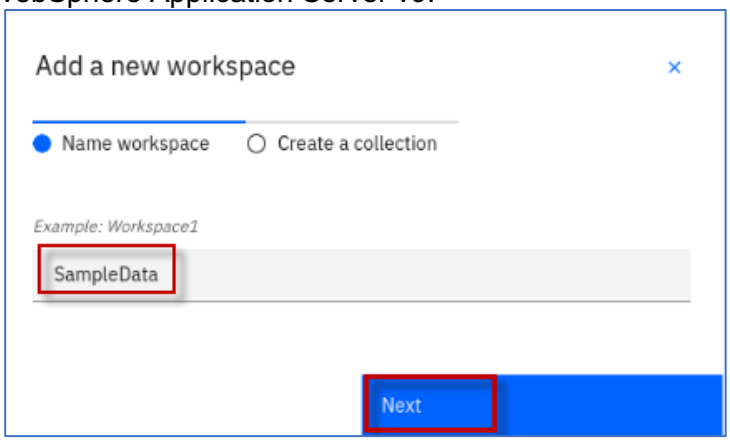


The Transformation Advisor **Welcome page** is displayed



6.3 Load a Data Collection

Transformation Advisor can evaluate Java based applications and help to package the good candidate application to move to cloud. Next, you will use Transformation Advisor to view the data analysis that shows this application is good candidate for Liberty in containers on RedHat OpenShift. To do this, you need first to load a data collection for the Mod Resorts application currently deployed on-premise in WebSphere Application Server v9.

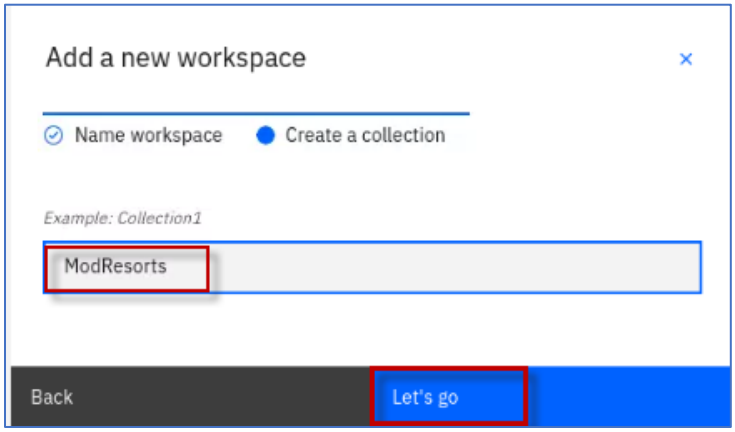


__1. From the Transformation Advisor Home page

__a. Click "**Add a new workspace",** enter the workspace name as **SampleData.** Then **c**lick **Next**.

A workspace is a designated area that will house the migration recommendations provided by **Transformation Advisor** from your application server environment. You can name and organize these however you want, whether it's by business application, location, or teams

__b. Enter the collection name as **ModResorts** and click **Let's go**.



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

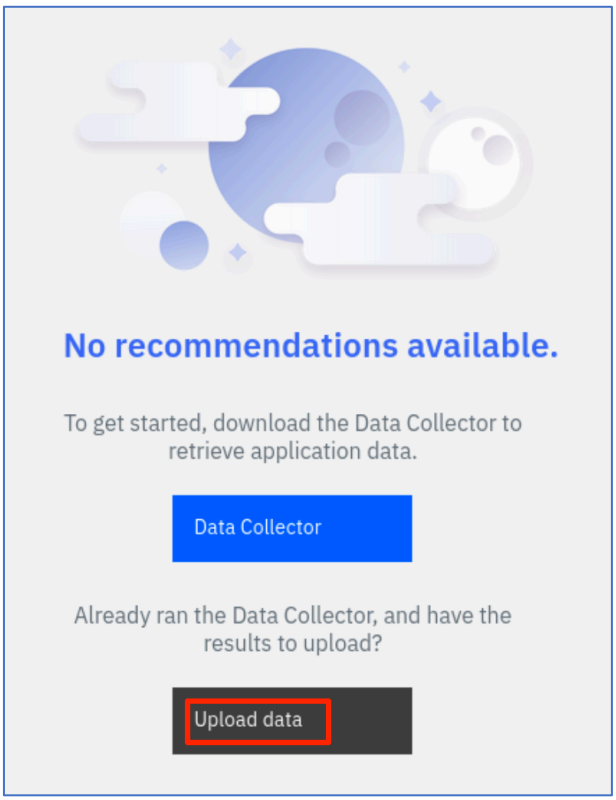
__3. Upload the Mod Resorts application data into Transformation Advisor

Once the Workspace and Collection are created, you will have options to either **download the Data Collector** utility or **upload existing data** file.

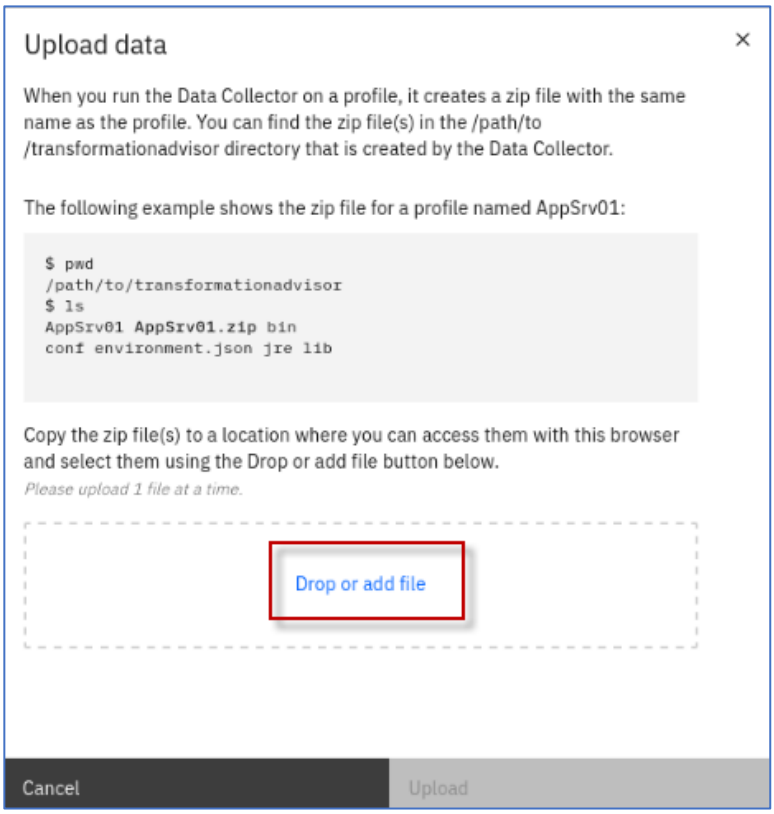
A video that illustrates the data collection process is available here: https://youtu.be/MhB-MI5aBc0.

In this lab, that data collection has been done for you. You will load the data collection **zip** file into **IBM Transformation Advisor** during this step.

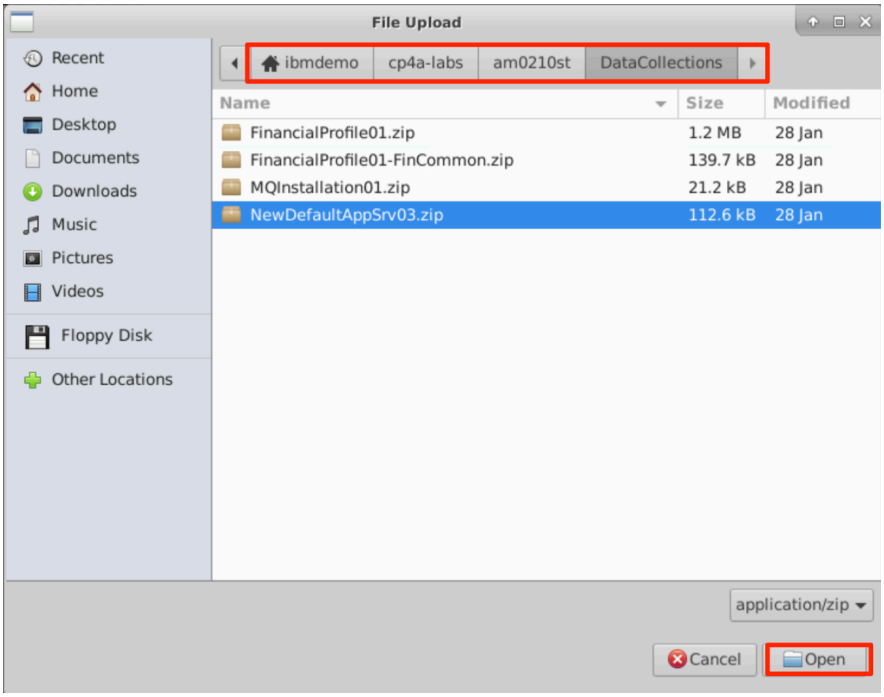
__a. Specify that you will **upload a data** collection zip file the upload page



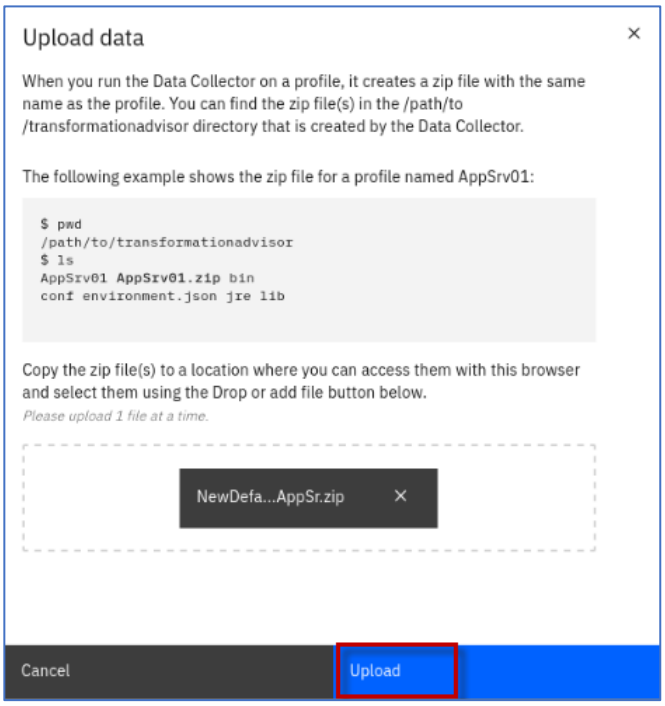
__b. Click **Drop or add file**



__c. Navigate to **/ibmdemo/cp4a-labs/am0210st/DataCollections** directory, select N**ewDefaultAppSrv03.zip.** Then click **Open**

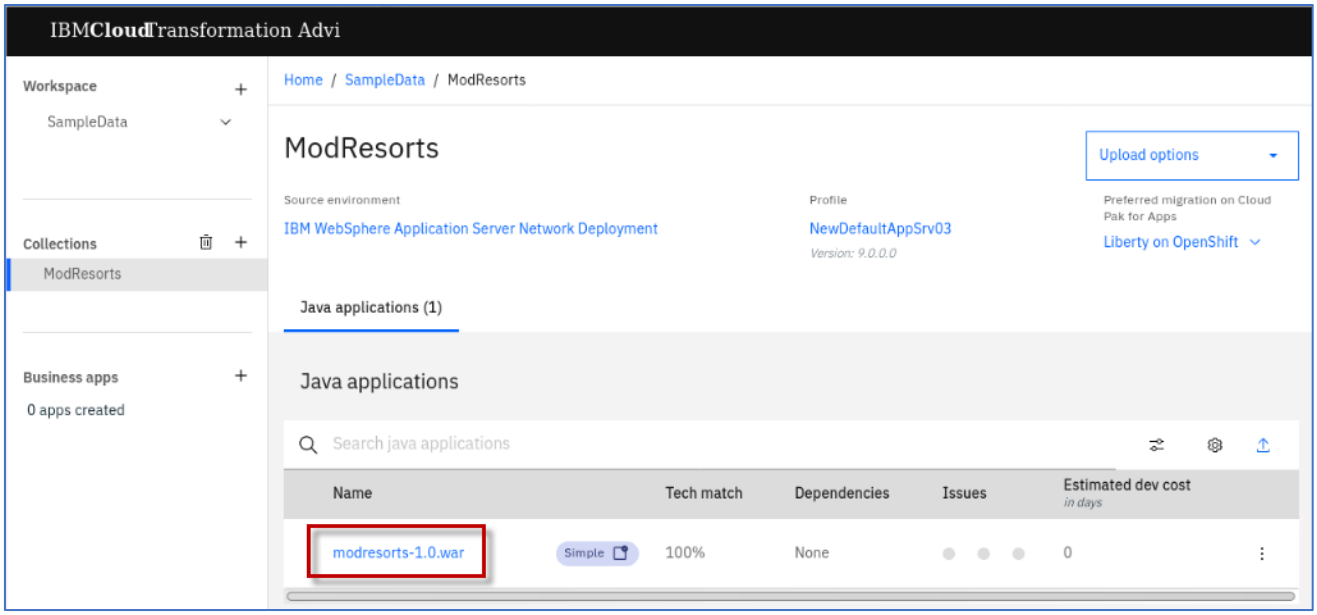


__d. Click **Upload**



Transformation Advisor will show its recommendations page with details about modernizing the application to Liberty in IBM Cloud Pak for Applications.

Note: The **modresorts-1.0.war** application used in this lab does not require any modification to the source code in its modernization. This enables the completion of the lab in a pre-determined amount of time. The modernization of applications that need modifications will be described, but not required for completion of the lab.



The Transformation Advisor provides all migration recommendations for all applications deployed to the WAS server based on the specified source and target environments.

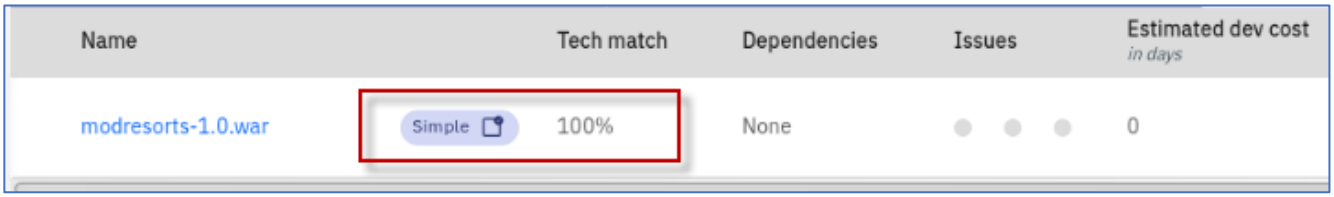
On the **Recommendations** page, the identified migration source environment is shown in the Profile section, and the target environment is shown in the Preferred migration section. The data collector tool detects that the source environment is your WebSphere Application Server ND **NewDefaultAppSrv03** Profile. The target environment is **Liberty on OpenShift**, which is the default target environment.

The Recommendations page also shows the summary analysis results for all the apps in the **NewDefaultAppSrv03** environment to be moved to a Liberty on Private Cloud environment. For each app, you can see these results:

- Complexity level
- Technology match
- Dependencies
- Issues
- Estimated development cost in days

For example, if you want to move the **modresorts-1_0_war** application to Liberty on Private Cloud, the complexity level is **Simple** and the Tech match is **100%**.

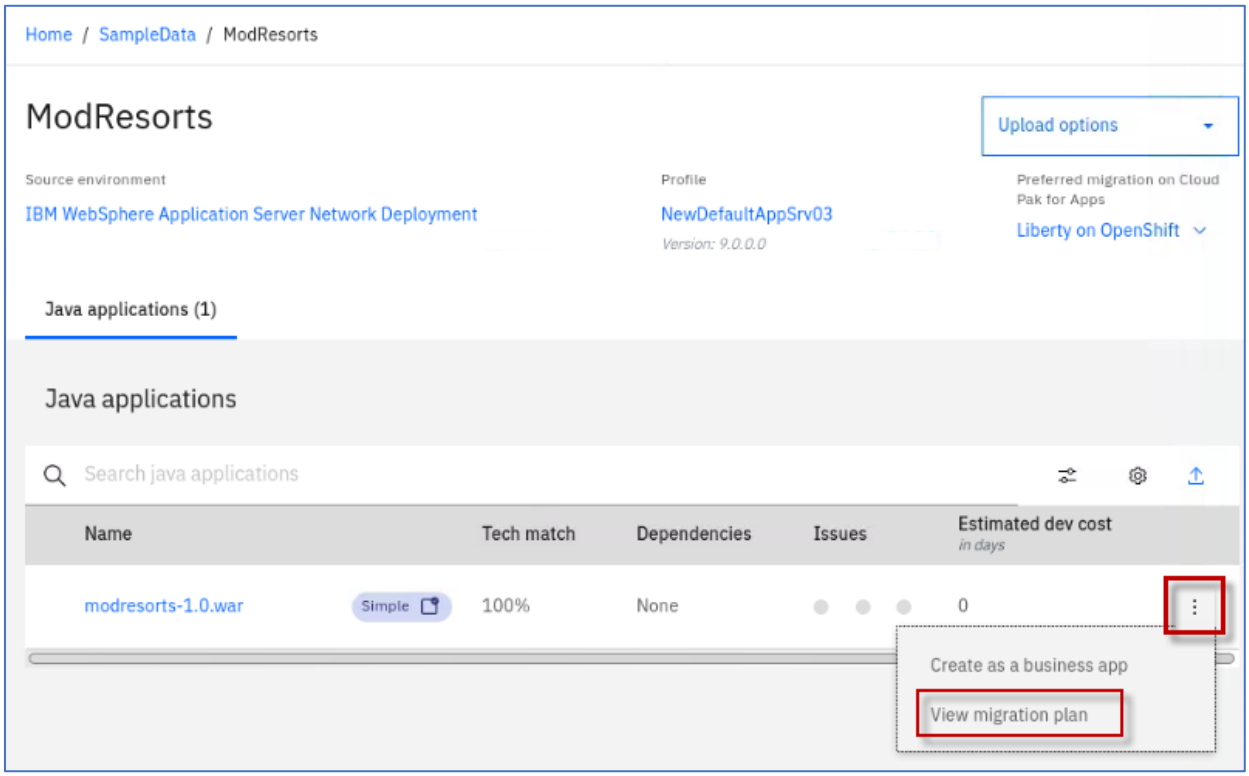
A Simple complexity score indicates that the application code does not need to be changed before it can be moved to cloud. The application has no dependency, has 1 minor level issue and the estimated development effort is 0 day.



6.4 View the Migration Bundle

In this section, you view the migration bundle contents for the Mod Resorts application. The migration bundle is created by the Transformation Advisor for a quick deployment to cloud while minimizing the migration error.

__1. Click on the menu with the "**hamburger icon**" next to the **Mod Resorts** application analysis to display the menu choices and choose **View migration plan**:



Transformation Advisor will display details about the migration bundle that it generated to accelerate the modernization of this application into Liberty on OpenShift. The migration bundle includes diverse artifacts, depending on the needs of the application.

The user can choose to create a migration bundle for either of the options below:

- a **binary project** of an application (uploading a WAR/EAR file and its dependent libraries), or
- a **source project** of an application, so that the application source files can be modernized and maintained.

The user can also choose to create a migration bundle that will either:

- allow the project to integrate with the **Accelerator for Teams**, making the modernization developer experience align with the developer experience for cloud-native applications
- deploy into **RHOCP** via an Operator

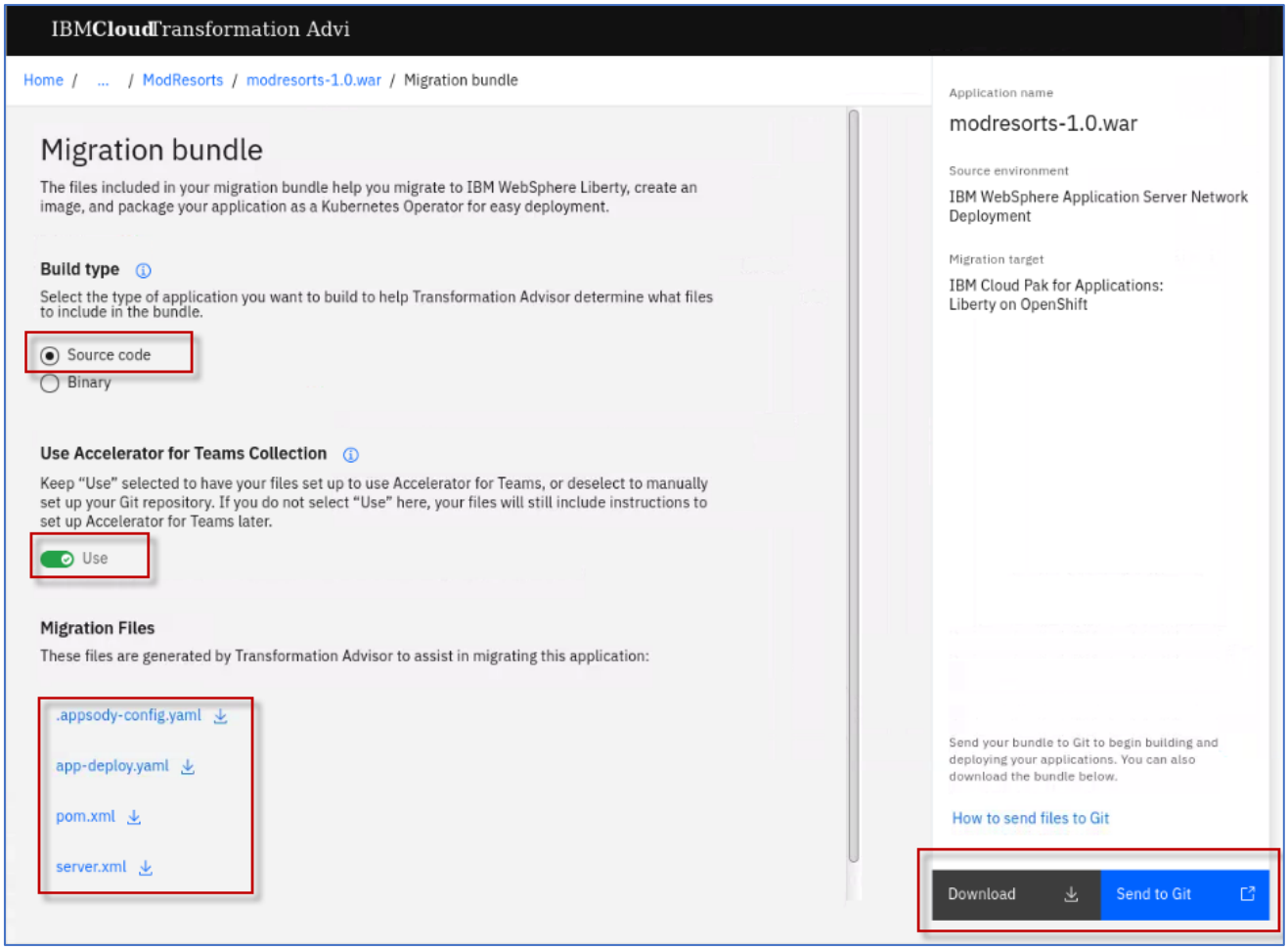
The **default selection is for a source project** that will integrate with the **Accelerator for Teams** (derived from the open source project Kabanero.io).

__2. View the migration bundle settings and functions.

The artifacts change, depending on the user's modernization choices for the application. The default selection for an application that uses only EE7 features and above will be to deploy as a source application project, and to use the Accelerator for Teams. The project will be a source project, with the directory structure set up with a starter application and the Liberty configuration for the application.

To accelerate the application modernization, the artifacts produced by **Transformation Advisor** include the configuration for the Liberty server, the build file for the application and the Accelerator for Teams configuration files (which can used with the Application Stacks).

The user will be able to choose to download the artifacts as a migration bundle or push the bundle contents into a GitHub repository.

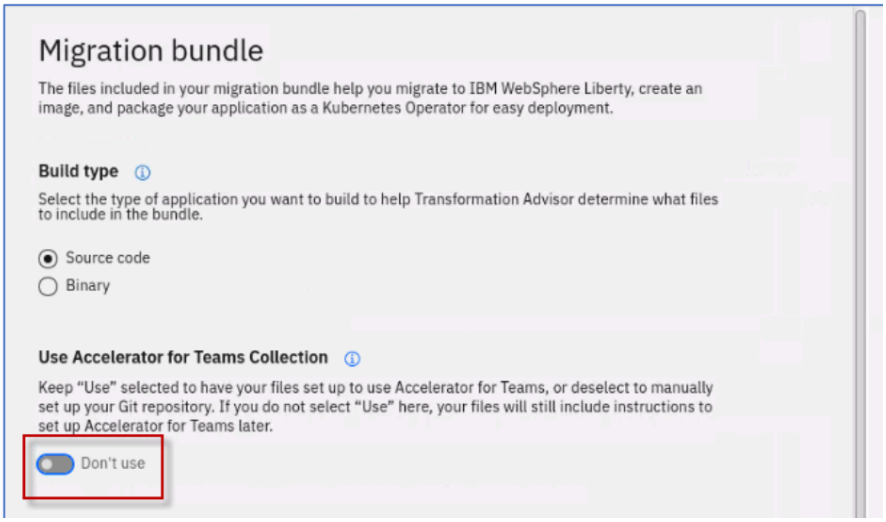


__a. Move the slider for **Use Accelerator for Teams Collection** to **Don't use**

This change means the **Mod Resorts** migration bundle will not use the **Accelerator for Teams**, but will build an image with Docker and deploy via an Operator into **RHOCP**.

The project will still be a **source** project, with the directory structure set up with a starter application and the Liberty configuration for the application.

When not using the **Accelerator for Teams**, the artifacts produced by **Transformation Advisor** include the configuration for the Liberty server, the build file for the application, the Dockerfile to produce the deployed application image, and the Operator to deploy the applications into **RHOCP**.

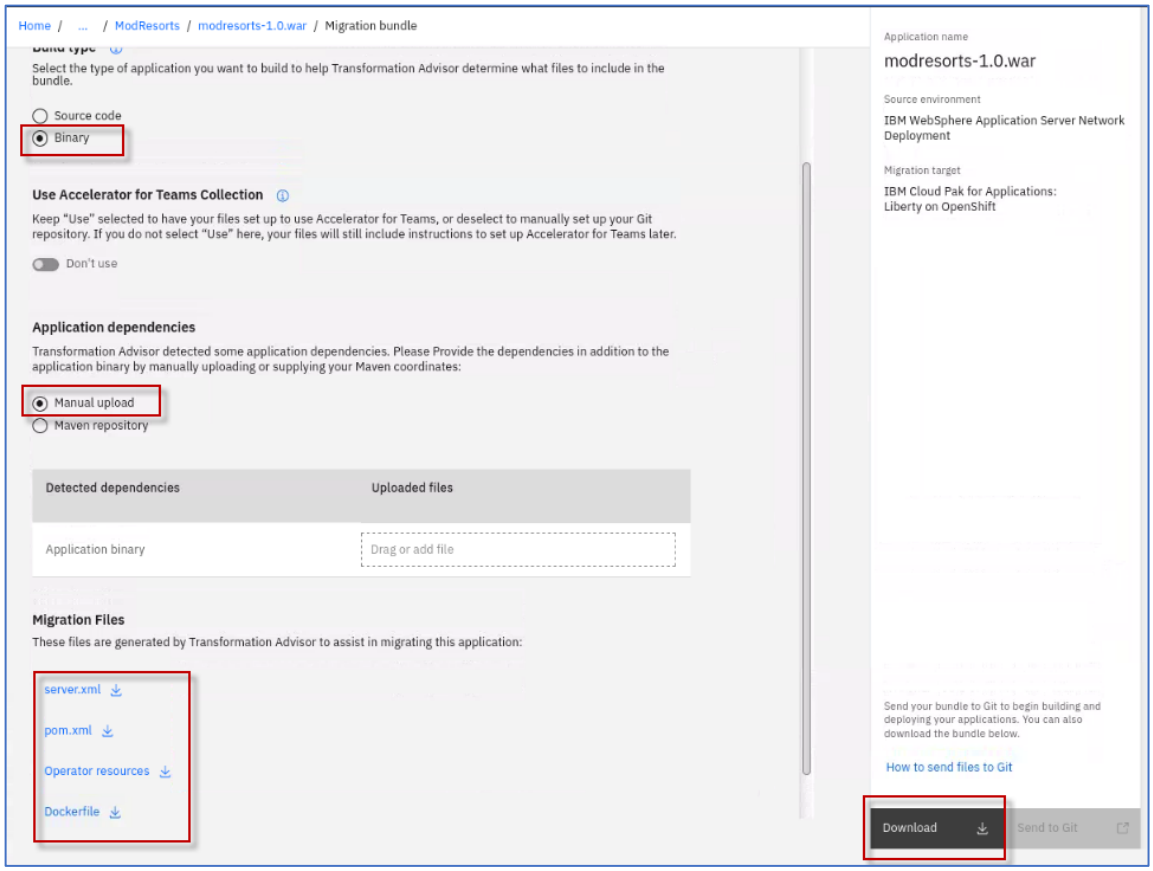


__b. Click **Binary** to change the Mod Resorts migration bundle to build from a binary application project.

Changing the selection for any application to deploy as a binary project will not only change the artifacts created, but the application cannot use **Accelerator for Teams**, as there is not yet an **CP4Apps** Application Stack that will build and deploy a binary project from **Transformation Advisor**.

The user can choose to upload the binary application and its dependencies for the project or provide **Maven** coordinates to identify the location of the binary application and its dependencies.

Once the user has uploaded the binaries or provided the **Maven** coordinates, the user can choose to download the artifacts as a migration bundle.

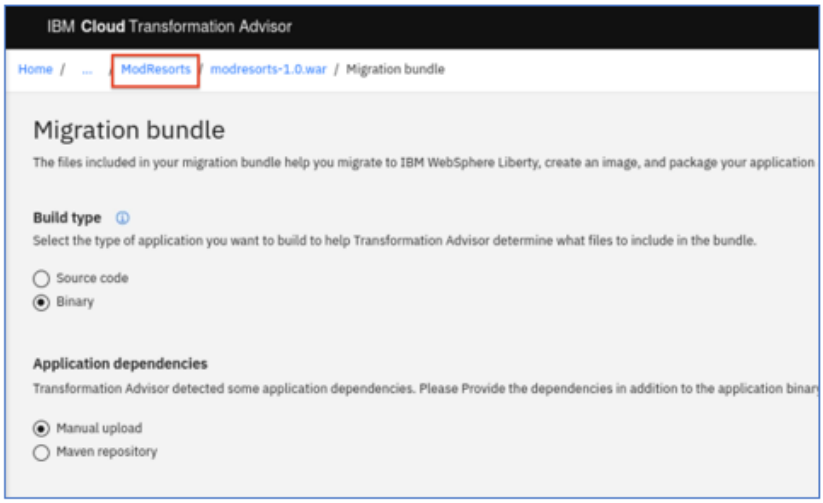

Home / ... / ModResorts / modresorts-1.0.war / Migration bundle
Select the type of application you want to build to help Transformation Advisor determine what files to include in the bundle.
Source code
Binary
Use Accelerator for Teams Collection
Keep "Use" selected to have your files set up to use Accelerator for Teams, or deselect to manually set up your Git repository. If you do not select "Use" here, your files will still include instructions to set up Accelerator for Teams later.
Don't use
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
Drag or add file
Migration Files
These files are generated by Transformation Advisor to assist in migrating this application:
server.xml
pom.xml
Operator resources
Dockerfile
Application name
modresorts-1.0.war
Source environment
IBM WebSphere Application Server Network Deployment
Migration target
IBM Cloud Pak for Applications: Liberty on OpenShift
Send your bundle to Git to begin building and deploying your applications. You can also download the bundle below.
How to send files to Git
Download
Send to Git

6.5 Push the migration bundle into GitHub

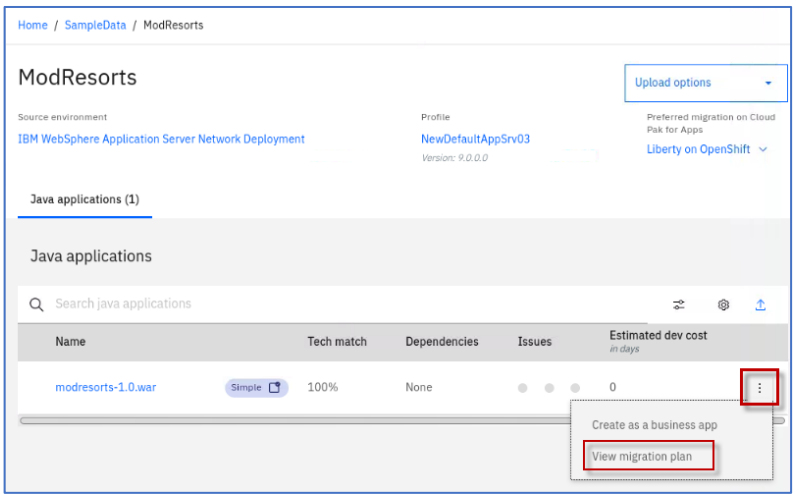
This step requires that you have a public GitHub account. An Enterprise account (i.e. https://github.ibm.com) will not work with SkyTap.

If you do not have a GitHub account, create a free account at: https://github.com

__1. Go back to the **ModResorts** collection by clicking on the collection name in the breadcrumb:

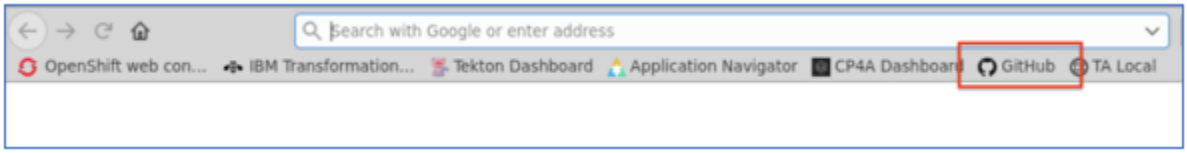


__2. Click on the menu with the "**hamburger icon**" next to the **Mod Resorts** application analysis and choose **View migration plan**:

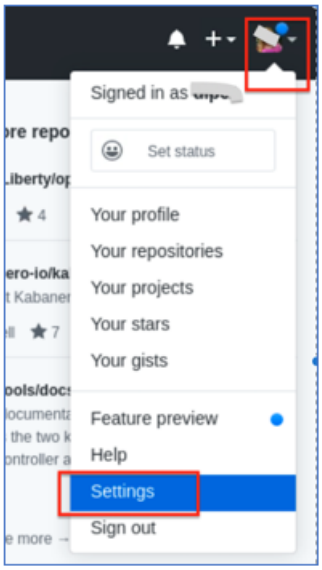


__3. Create a GitHub personal access token.

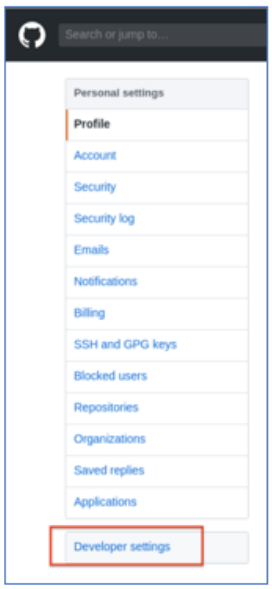
__a. In a **new tab** in the browser, click the **GitHub** bookmark to launch it.



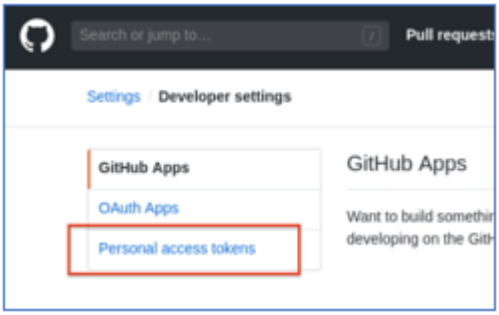
__b. Log in to your existing GitHub account using your userid and password. Once you have logged into your GitHub account, click on your user image in the top right corner and choose **Settings**:



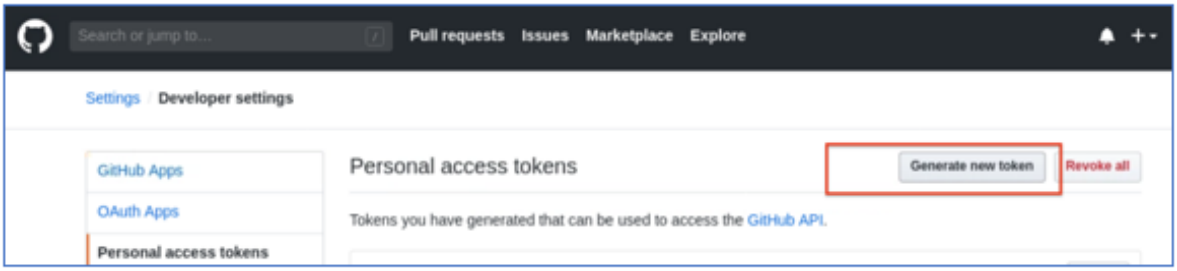
__c. On the bottom left of the next screen, click on **Developer settings**



__d. On the next screen, click on **Personal access tokens**

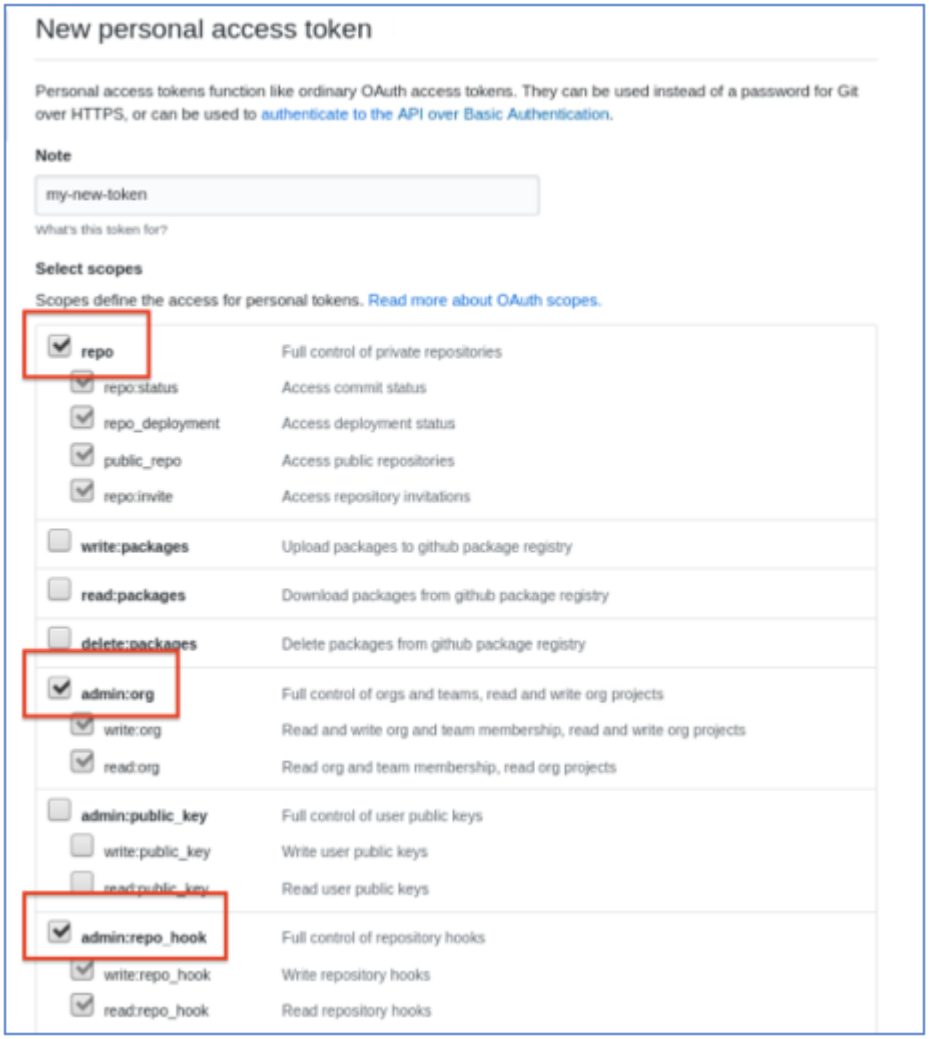


__e. On the next screen, click on **Generate new token**.



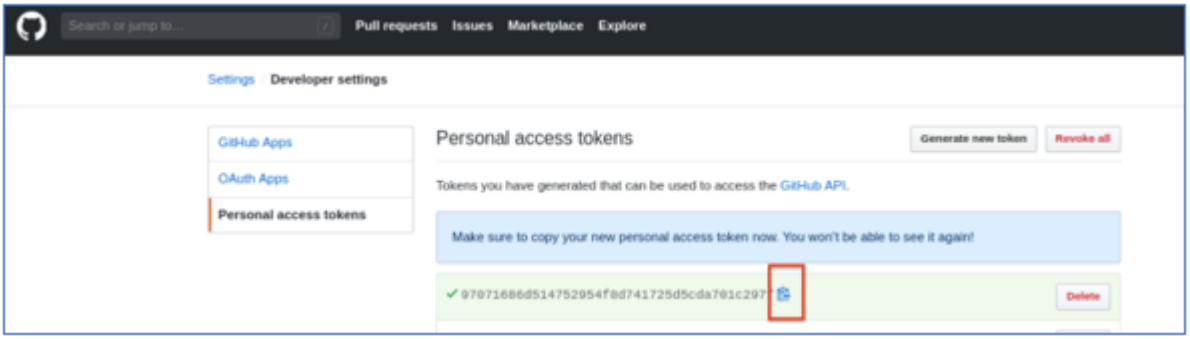
__f. Enter the token Name as: **my-new-token** with these three scopes

- **repo**
- **admin:org**
- **admin:repo_hook**



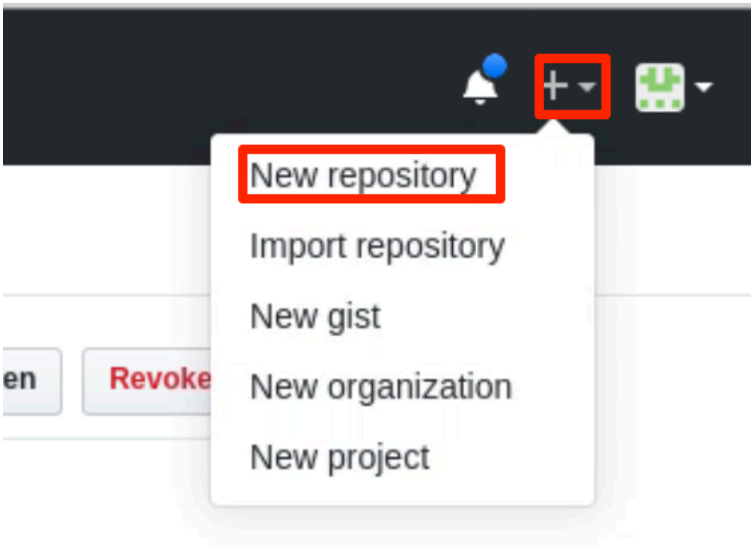
__g. At the bottom of the screen, click **Generate token**.

__h. **Be sure to save your access token**. It is valid for all repositories for your account. You can save it in a text file on the VM or you can leave the browser window open to be able to copy it.

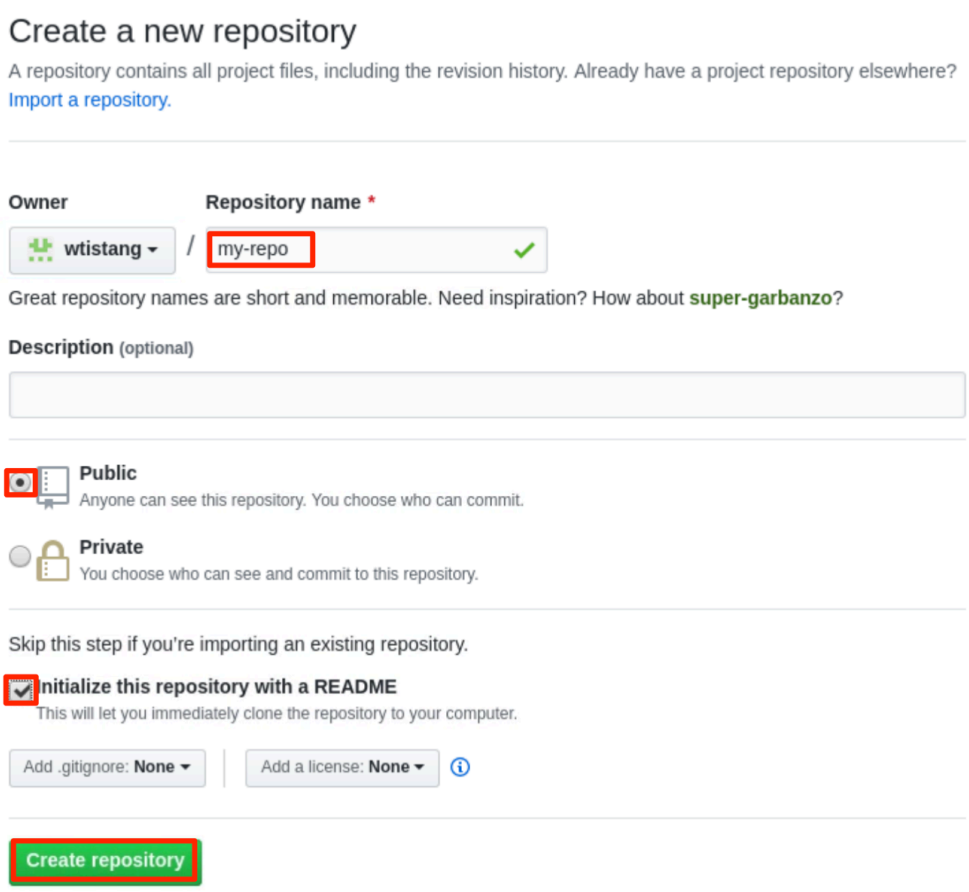


__4. Create a public **GitHub repo** for the application.

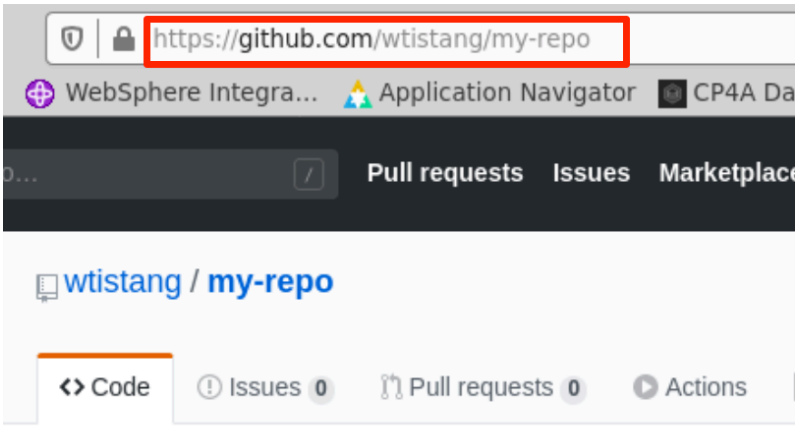
__a. At the left side of your login screen, click **Add** and select **New repository** to create a new repository:



__b. Specify the **Repository name** as "**my-repo**", and create the new repository as a **Public** repository with a **README**:

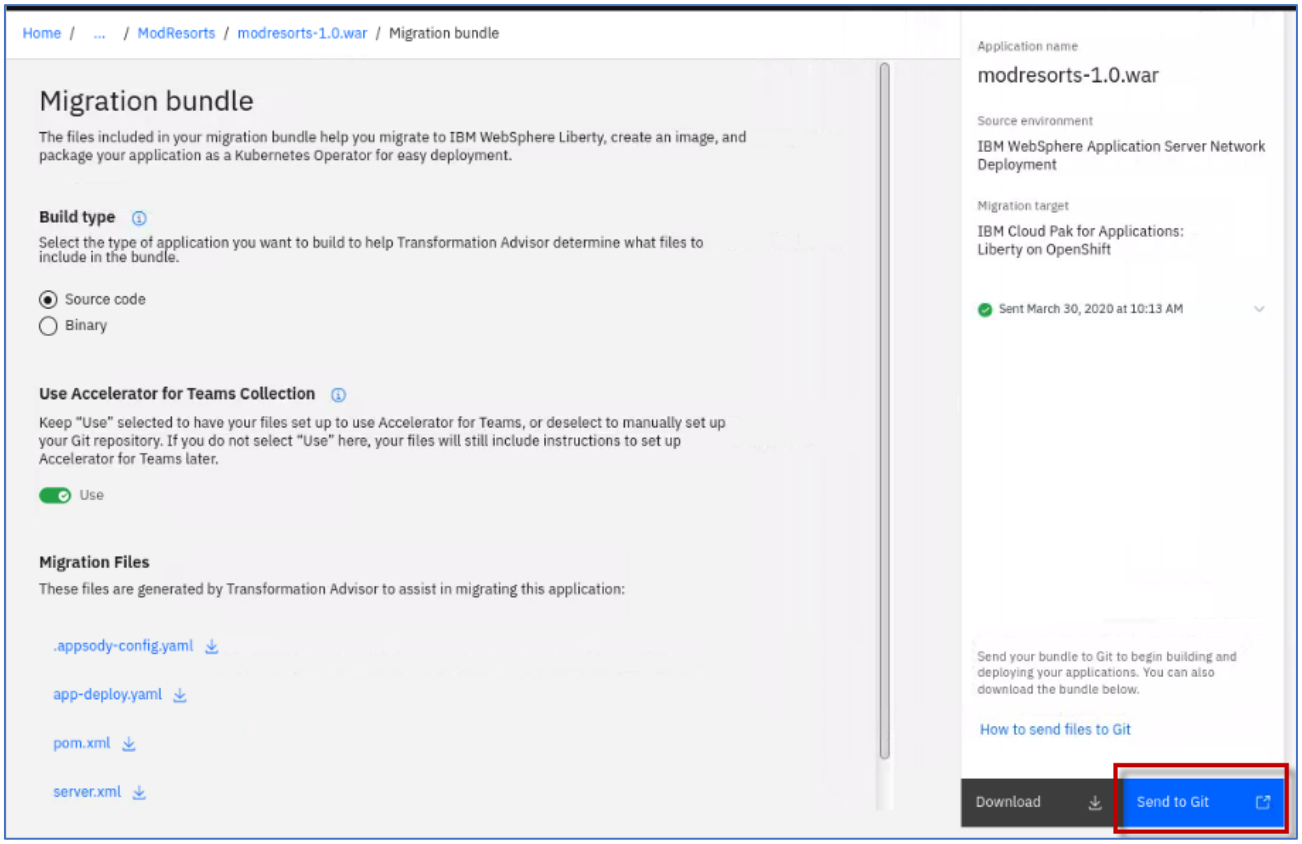


The URL for your new repository is in the address bar:

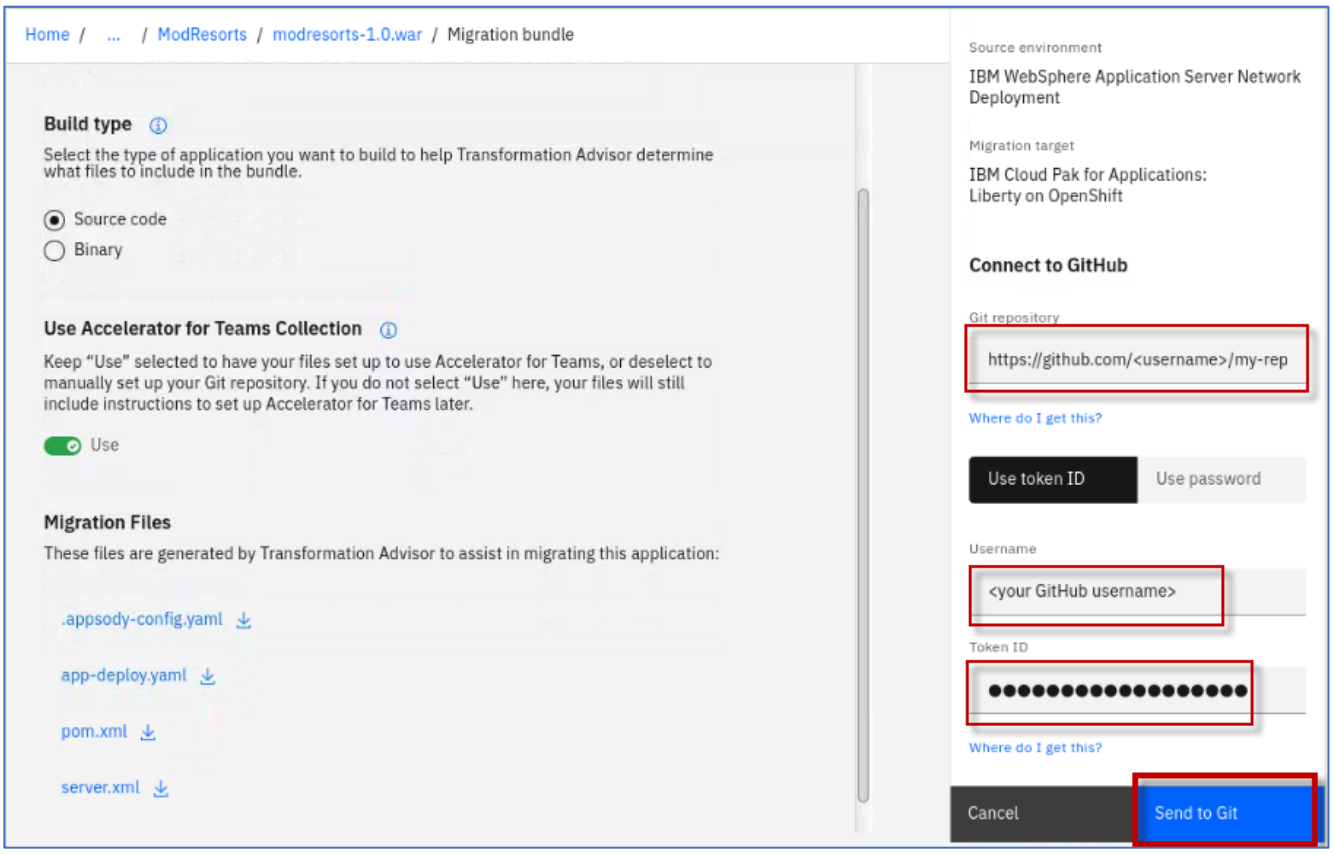


Now you have the three values required to push your migration bundle into GitHub: your **repository URL**, your **GitHub ID** and your **GitHub personal access token**.

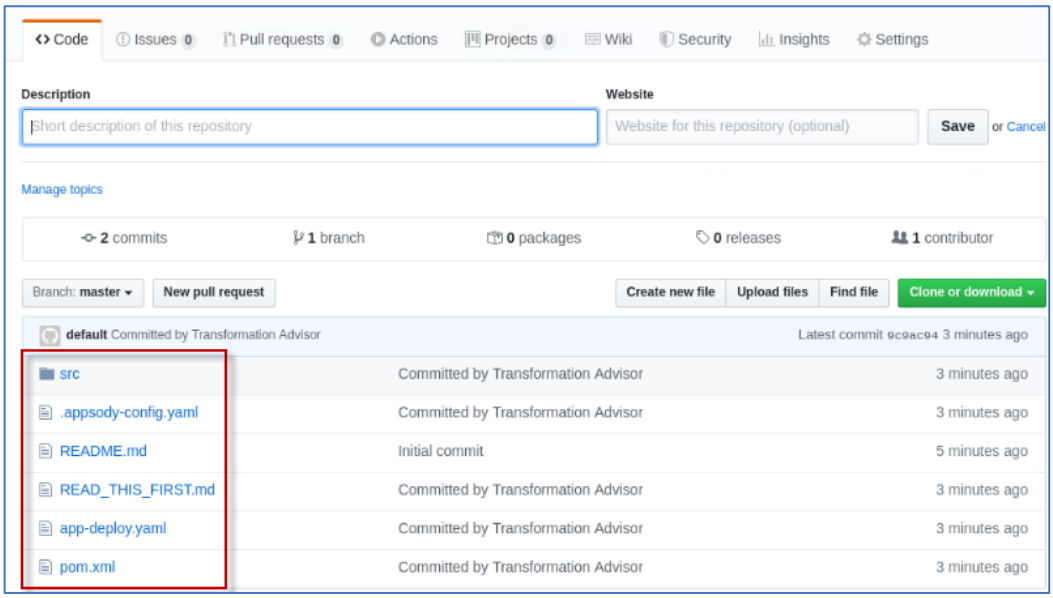
__5. Go back to **Transformation Advisor** page, click on **Send to Git** to prepare to send the migration bundle for **Mod Resorts** application to Git.



__6. Specify the GitHub repository URL, your GitHub ID and your GitHub personal access token: Then Scroll down and click **Send to Git.**



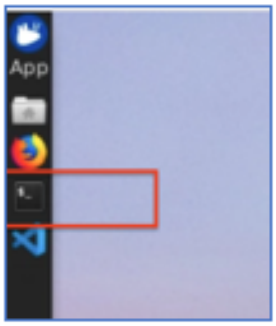
__7. Once the application artifacts are sent to **GitHub**, you can now view the contents of the migration bundle in **GitHub** by refreshing the home page for the repository URL:



6.6 Clone the GitHub Repository

In this section, you clone your GitHub repository on the local workstation and import the source for the application.

__1. Open a Terminal window by clicking its icon on the Desktop tool bar:



__2. From the Terminal window, clone the GitHub repository using the following commands:

cd /home/ibmdemo
git clone https://github.com/<your GitHub username>/my-repo

Where <**your GitHub username**> is your GitHub user id.

__3. Change to the **my-repo** directory that you just cloned with command:

cd my-repo

__4. From the Terminal window, extract the source code (saved in the **/home/ibmdemo/cp4a-labs/am0210st/modresorts_src.zip** file) into the local repository source directories using the following command:

unzip /home/ibmdemo/cp4a-labs/am0210st/modresorts_src.zip

Note: answer **y** when prompted to replace the **index.html** file.

6.7 Build and Package the ApplicationBuild and package the application in a Liberty container in Codewind using

The source code development activities might not be executed by the same person as the previous steps (see the diagram in the Objective section). The solutions architect might use Transformation Advisor's insights to make decisions about the modernization of the application estate and send projects to GitHub for the developer.

The developer will begin working on the modernization of the application, including any needed source code changes, building and testing using the Accelerator for Teams.

This lab uses **Visual Studio Code** (**VS Code**) with the **Codewind extension** installed; there are no code changes required for the **Mod Resorts** application modernization.

Applications which require code changes, due to required Java EE and/or Java SE differences in the modern development environment, would benefit from the combination of the Eclipse Codewind and WebSphere Application Migration Toolkit (**WAMT**) plugins in the Eclipse Integrated Development Environment (IDE).
See more about the WebSphere Application Migration Toolkit (WAMT) plugin here: http://ibm.biz/WAMT4Eclipse.

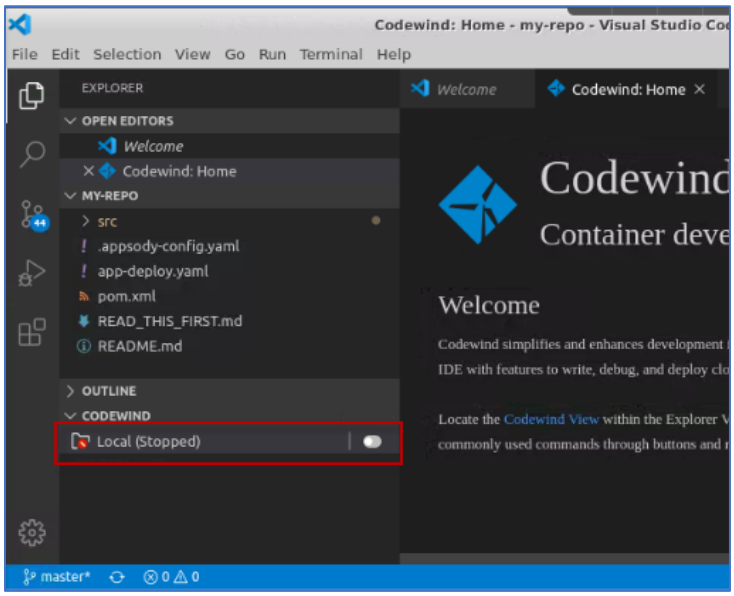
__1. Go to the Terminal window and launch VS Code using the following commands:

cd /home/ibmdemo/my-repo

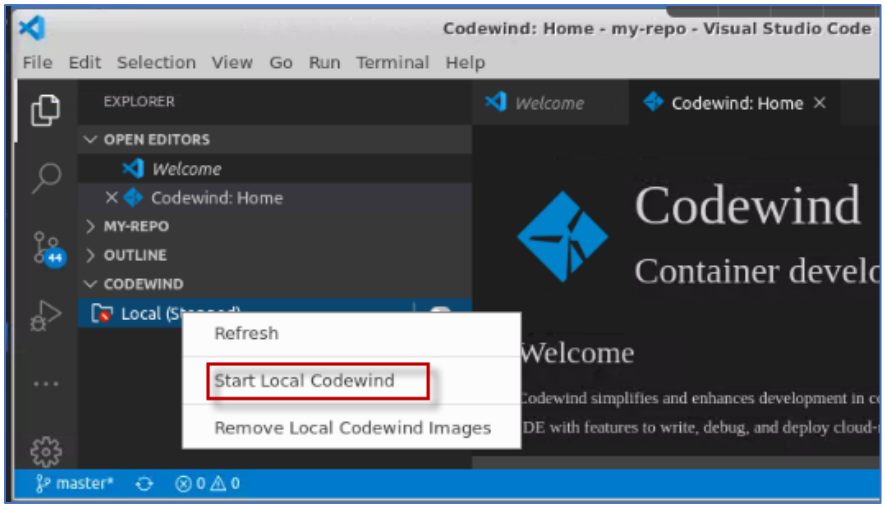
code .

The VS Code IDE is displayed.

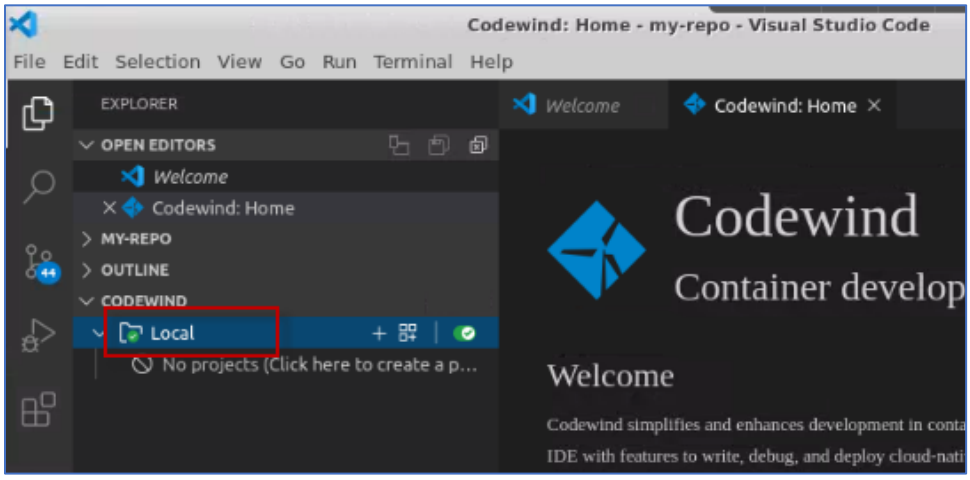
__2. If the local **Codewind** extension in VS Code is not started, you will need to start it now:



__a. Right click on **Local (Stopped)** and select **Start Local Codewind**:

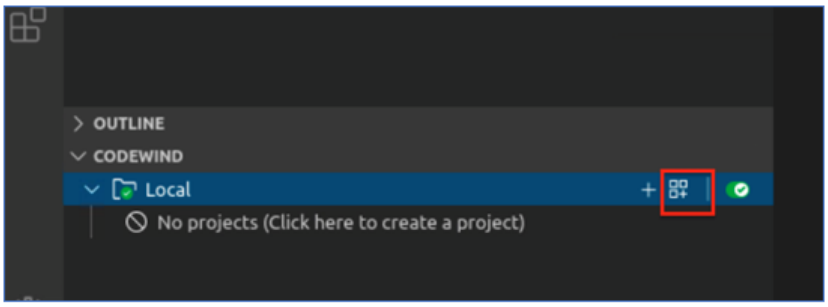


The Codewind extension will then be started:

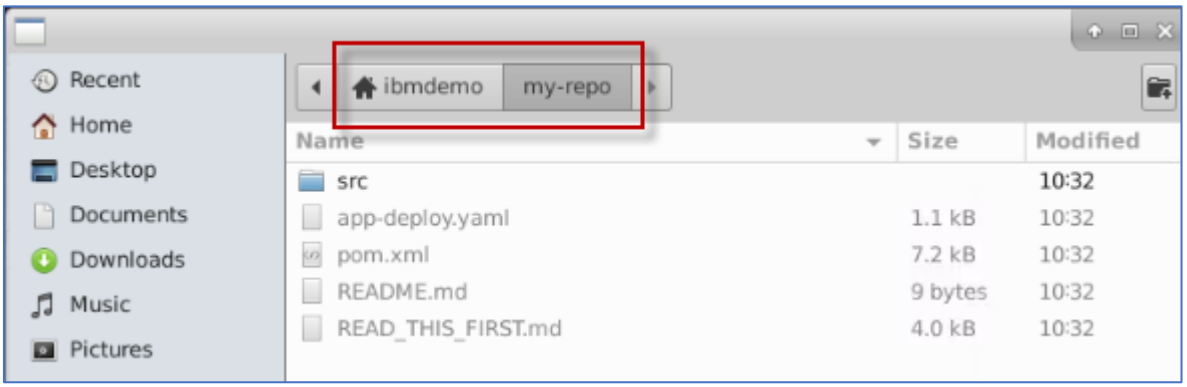


__3. Add the ModResorts project to Codewind.

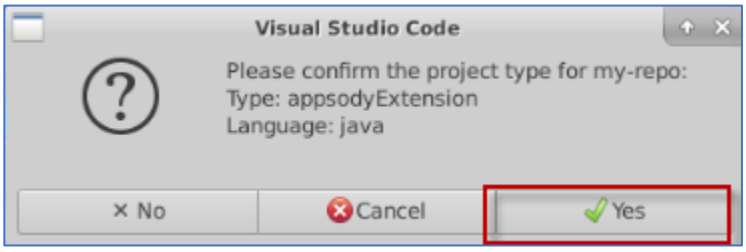
__a. Click on the icon to **Add Existing Project**



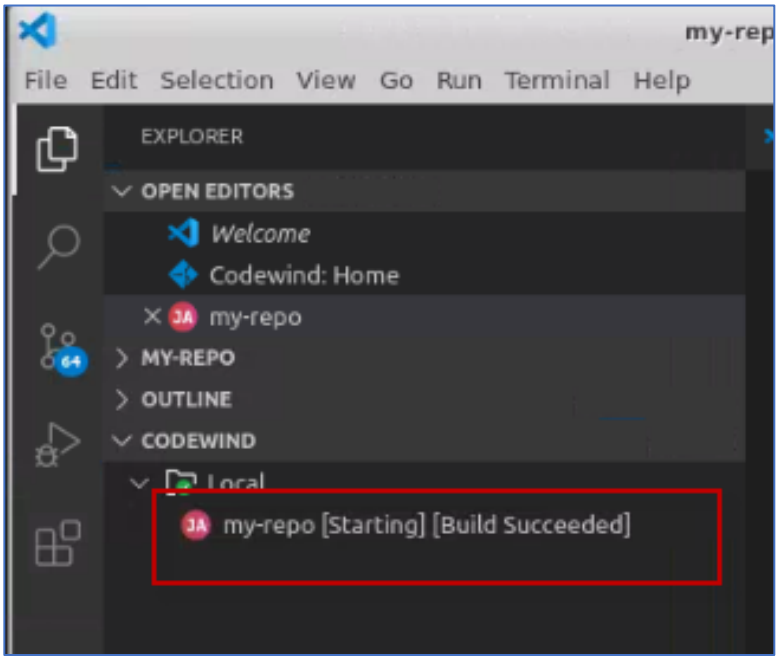
__b. Select the **/home/ibmdemo/my-repo** directory and click **Add to Local Codewind**:



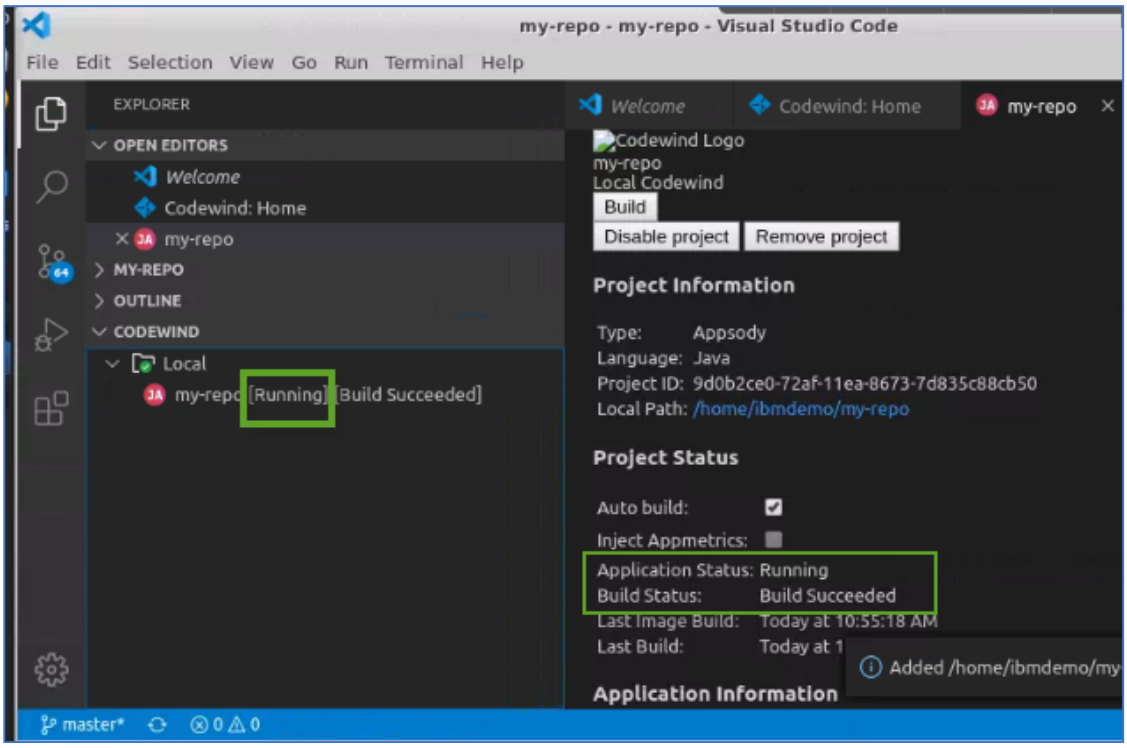
__c. Click **Yes** when **Codewind** detects that the project is for the Java Application Stack:



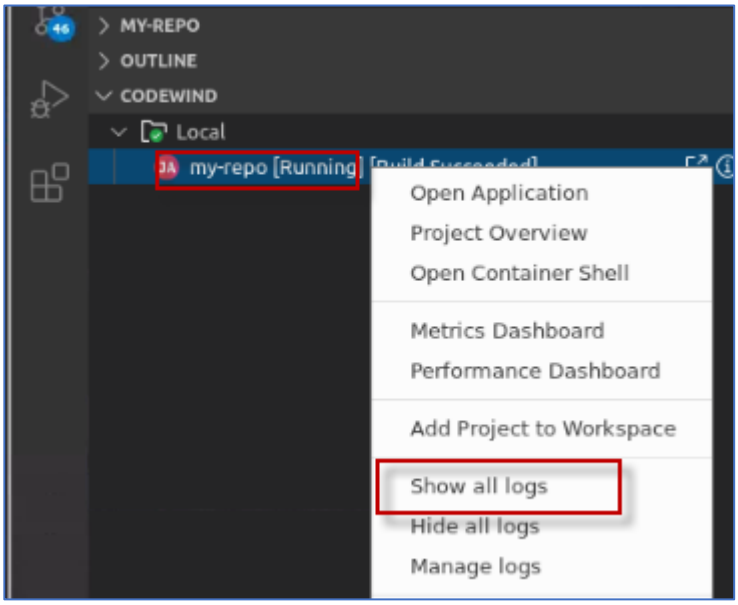
Codewind will detect that the application project is already in the VS Code workspace. Therefore, **Codewind builds** and **starts** the application automatically:



The application state should automatically change to “**Running**” once the app is built and started.

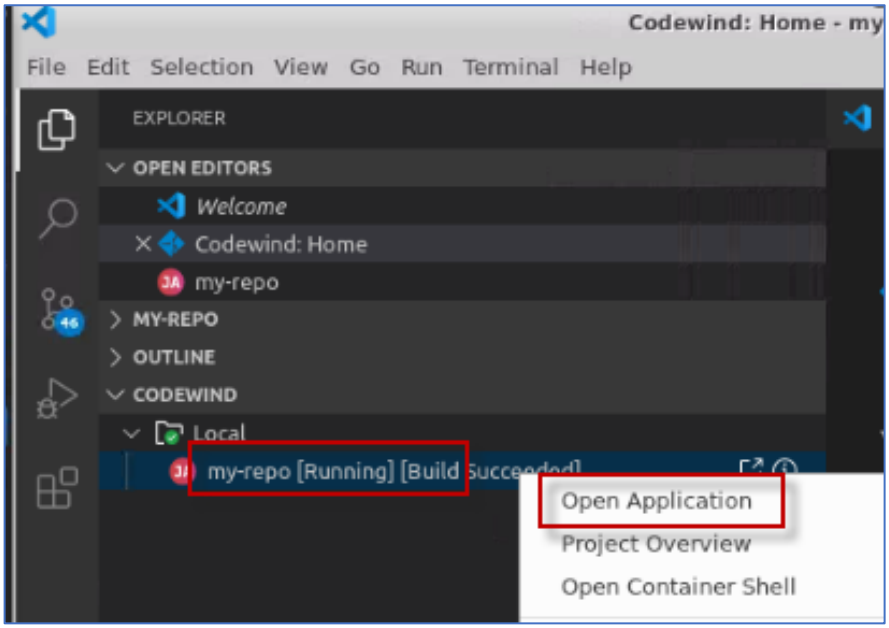


__d. If it seems that the application is not starting, you can view the logs by right-clicking the project name and selecting **Show all logs**:

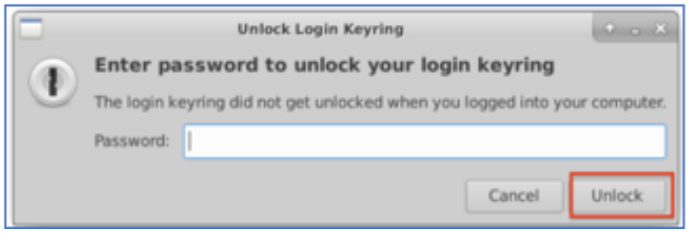


__4. Open the application when it is running.

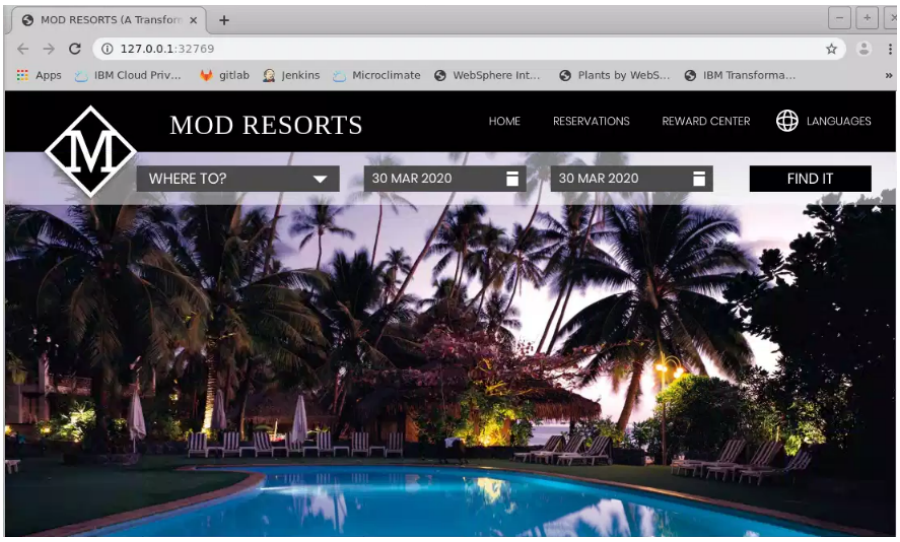
__a. When the status of your application changes from **Starting** to **Running**, right-click on the application. Then select **Open Application** from the menu.



__b. Clicking **Open Application** will open the Google Chrome browser. If prompted for a Keyring password, enter the **ibmdemo** password (**passw0rd**) and click **Unlock**:



__c. Opening the application will load the application (running in a container) in the browser.



__d. When you are satisfied that the application is ready to deploy outside of the local development and test environment, Close the browser window.

__e. Close the VS Code window by clicking **File>Exit**.

__f. From the Terminal window, commit the inclusion of the application source code and any changes made and push the code (using the Github personal access token) into the Git repository using the following commands:

cd /home/ibmdemo/my-repo
git add .
git commit -m "added ModResorts source code"
git push

Note: If prompted for a userid, specify your public GitHub user id. If prompted for a password by git push, you must specify your Git personal access token in place of the password if your Git account uses two-phase authentication.

6.8 Define the Tekton PipelineResources for the application's

Normally, Transformation Advisor and the Accelerator for Teams will set up webhooks from the user's GitHub repository into the Tekton pipelines and trigger the Tekton pipelines when there is a commit to the repository.

Because these webhooks will not work into SkyTap, this lab will set up and trigger the Tekton pipelines manually.

In this task you define two **PipelineResources** to be used by the Tekton pipeline for the application build and deployment.

One is the Tekton PipelineResource identifying the **image location**. The tag for that image must be changed every time the application is updated.

The second is a Tekton PipelineResource identifying the **repository** from which the pipeline will pull its data during a build.

The two **PipelineResources** are defined in **modresorts-pipeline-res.yaml** file located in the **/home/ibmdemo/cp4a-labs/am0210st** directory. You need to update the **[GIT_URL]** value and the **[IMAGE-TAG]** value in the YAML file**.**

__1. Modify the **modresorts-pipeline-res.yaml** file.

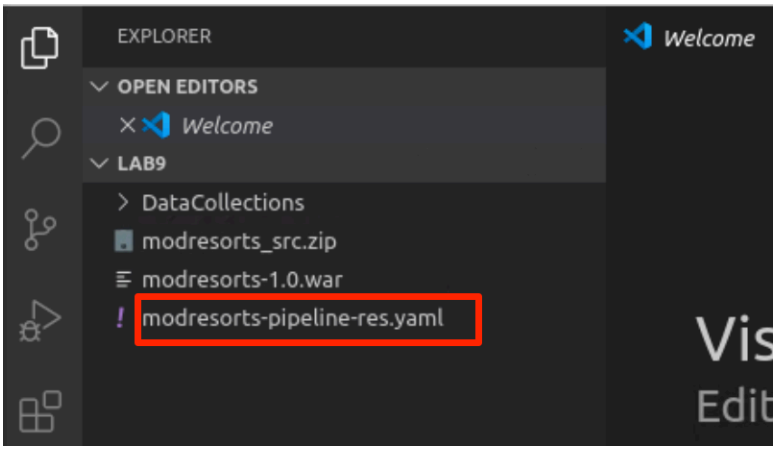
__a. From the Terminal window, navigate the **/home/ibmdemo/cp4a-labs/am0110st** directory with command:

cd /home/ibmdemo/cp4a-labs/am0210st

__b. Launch VS Code:

code .

__c. Double-click the **modresorts-pipeline-res.yaml** file to open it for editing.

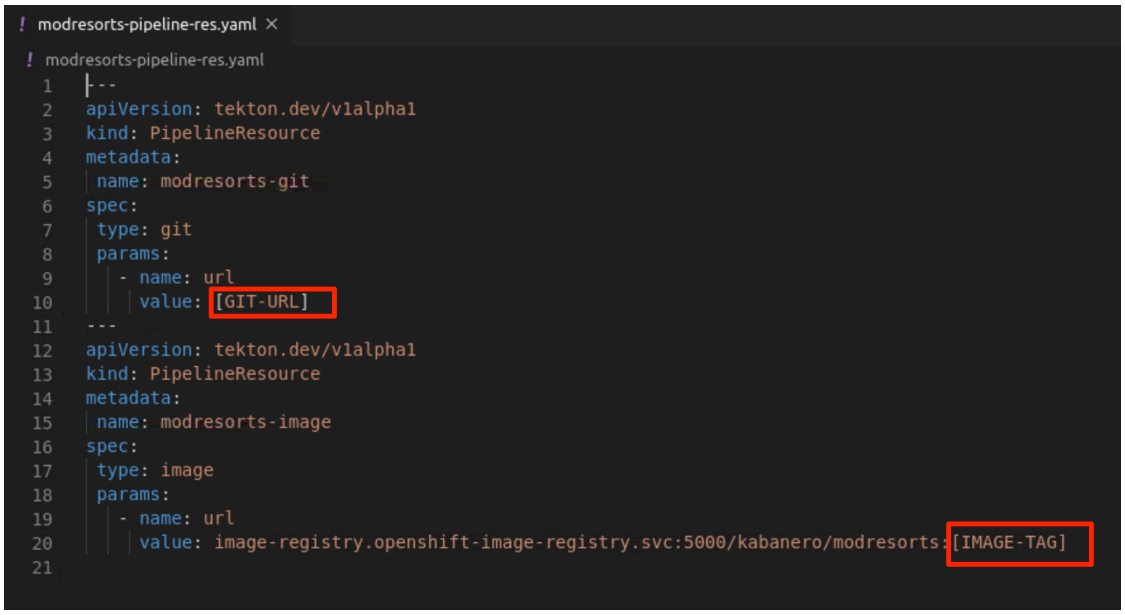


__d. Change the **[GIT_URL]** value to the URL of you GitHub repository and change the **[IMAGE-TAG]** value to some arbitrary tag:

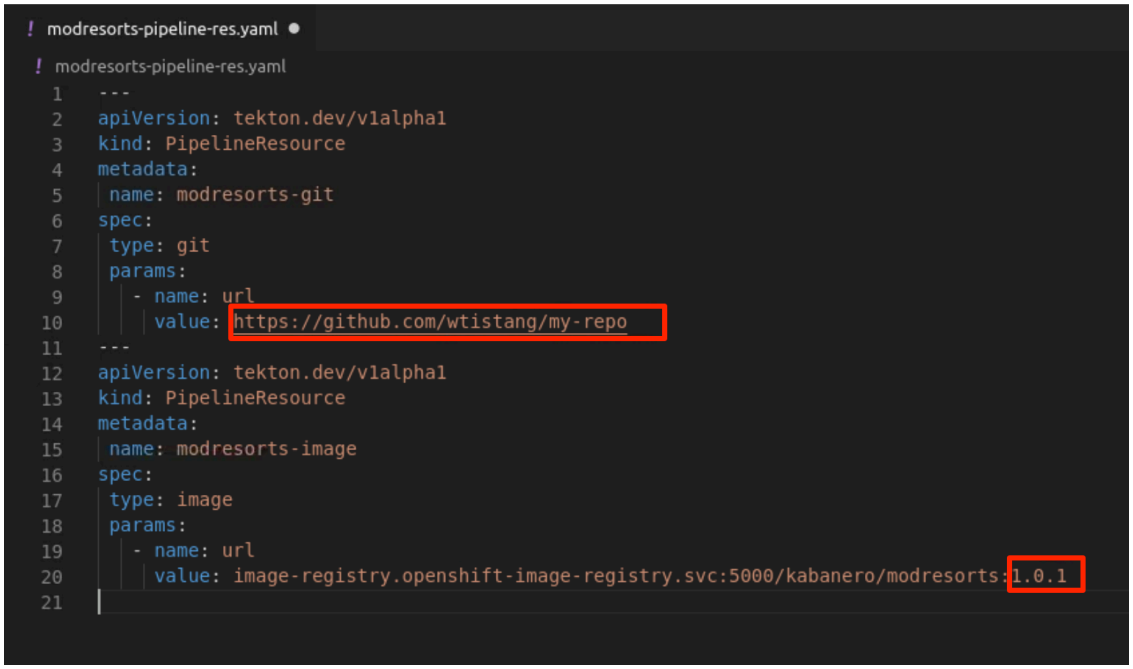
Example:

[GIT_URL] > **https://github.com/user01/my-repo**

[IMAGE-TAG] > **1.0.1**



You changes will look like this:



__e. Click **File>Save** All to save the changes and click **File>Exit** to close the Editor window.

__2. Deploy the PipelineResources to **RHOCP**.

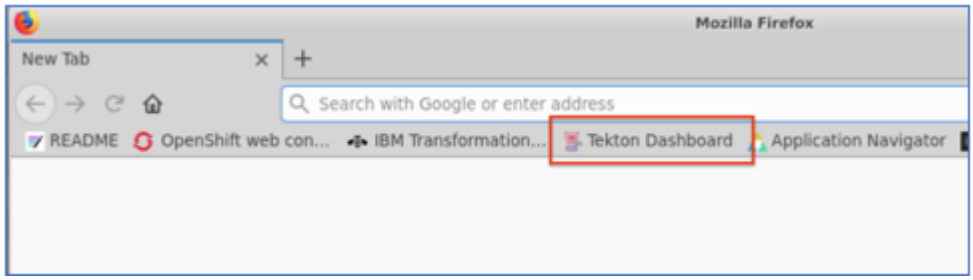
__a. Login to OpenShift and deploy the **modresorts-pipeline-res.yaml** in the **kabanero** namespace with commands:

oc login -u ibmadmin -p engageibm https://api.demo.ibmdte.net:6443
oc apply -f modresorts-pipeline-res.yaml -n kabanero

Note: Answer 'y' if asked if you want to use insecure connections

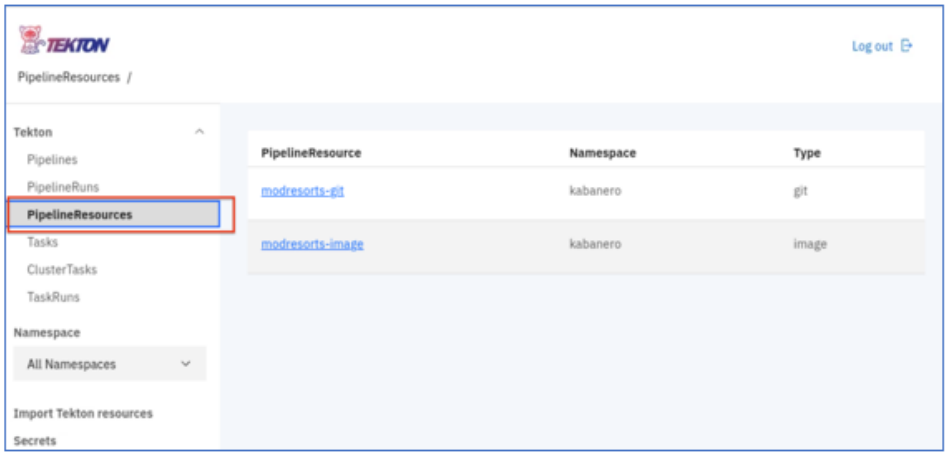
__3. View the Tekton PipelineResources in the Tekton Dashboard

__a. To open the **Tekton Dashboard**, click on the **Tekton Dashboard** bookmark of the **Firefox** browser window (or type the URL **https://tekton-dashboard-tekton-pipelines.apps.demo.ibmdte.net/** in the address bar):



__b. If prompted to login to the **RHOCP** login page, click **htpasswd** field. Then log in with **ibmadmin**/**engageibm** as the username and password as you did before.

__c. Click on **Pipeline Resources** to view the resources that you created for the Mod Resorts application.



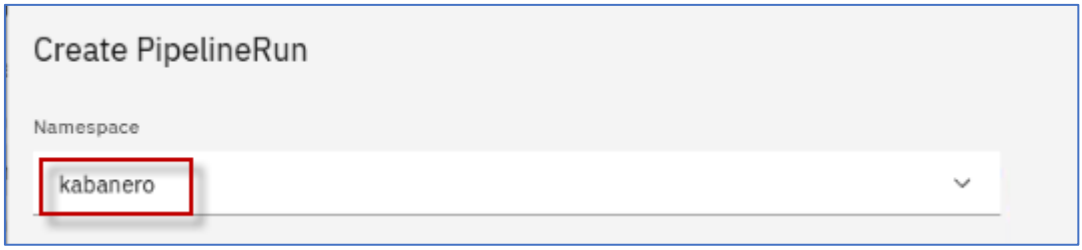
6.9 Deploy the application to OpenShift via the Tekton pipeline

Because the Git commit will not trigger the pipeline automatically via a Git webhook due to the network constrain of the lab environment, this lab will deploy the application by creating a pipeline run manually.

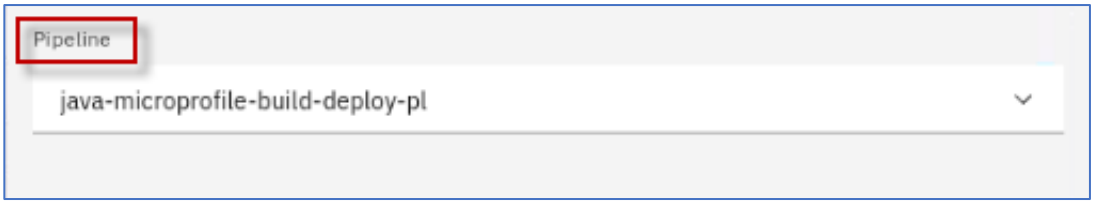
__1. In the Tekton UI, Click on **Pipeline Runs** and then click the **Create** button

__2. Fill in the fields for the **PipelineRun** as illustrated below:

- Namespace: **kabanero**

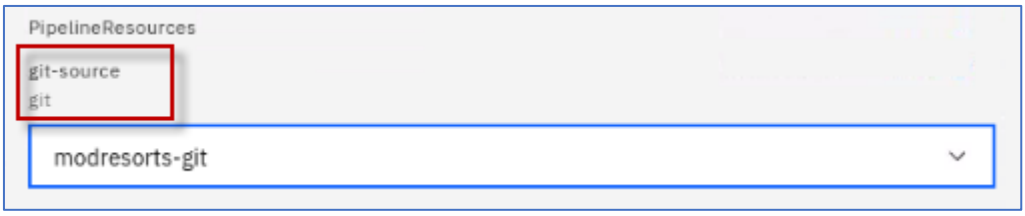


- Pipeline: **java-microprofile-build-deploy-pl**

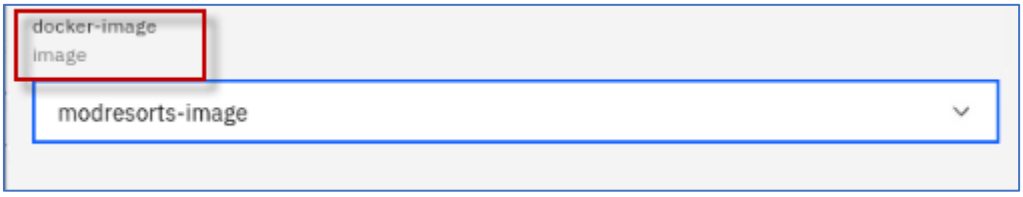


- Select the **modresorts-git** Tekton pipeline resource to build the **Mod Resorts** application.

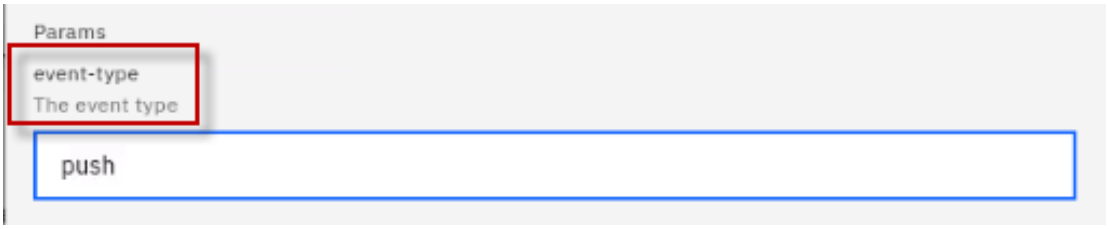
  This pulls from your git repo that you specified in the PipelineResource yaml file.



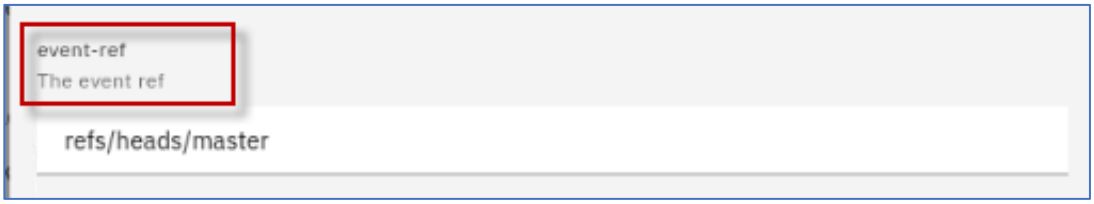
- Select **modresorts-image** for the Docker-image. This is the resource you defined in the Tekton PipelineResource yaml file



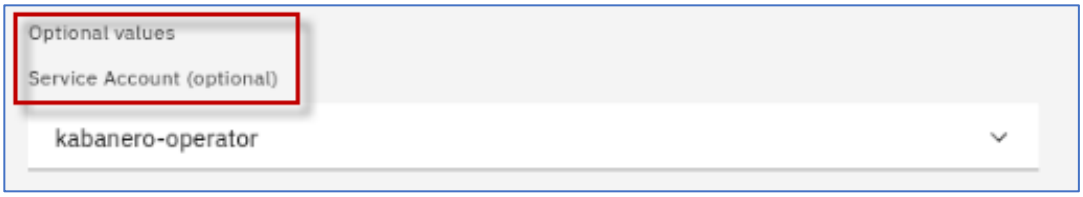
- Enter the **event-type** as **push**



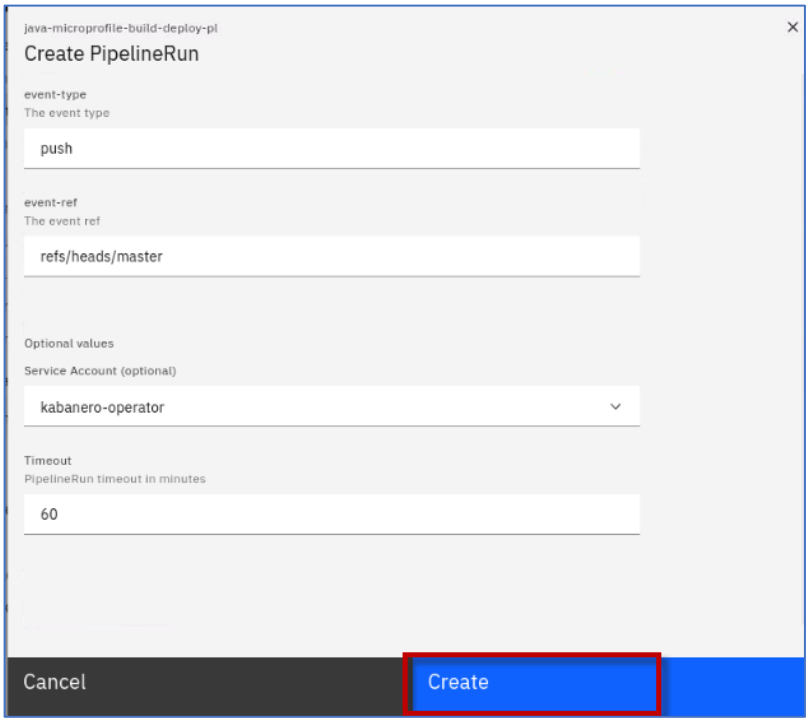
- Enter the **event-ref** as **refs/heads/master**



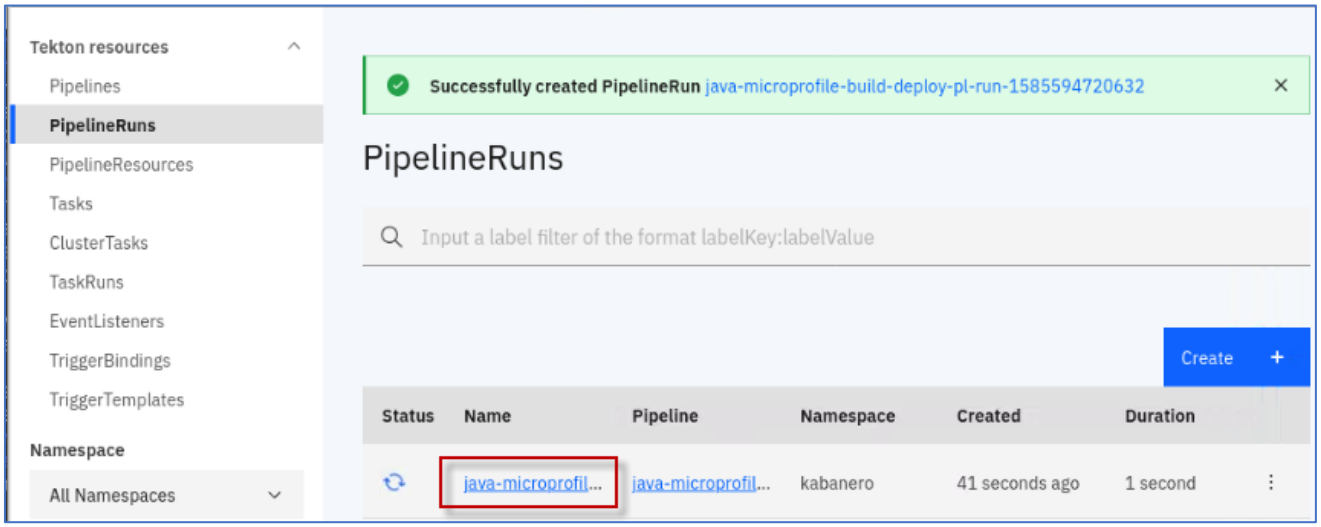
- Service Account as **kabanero-operator**



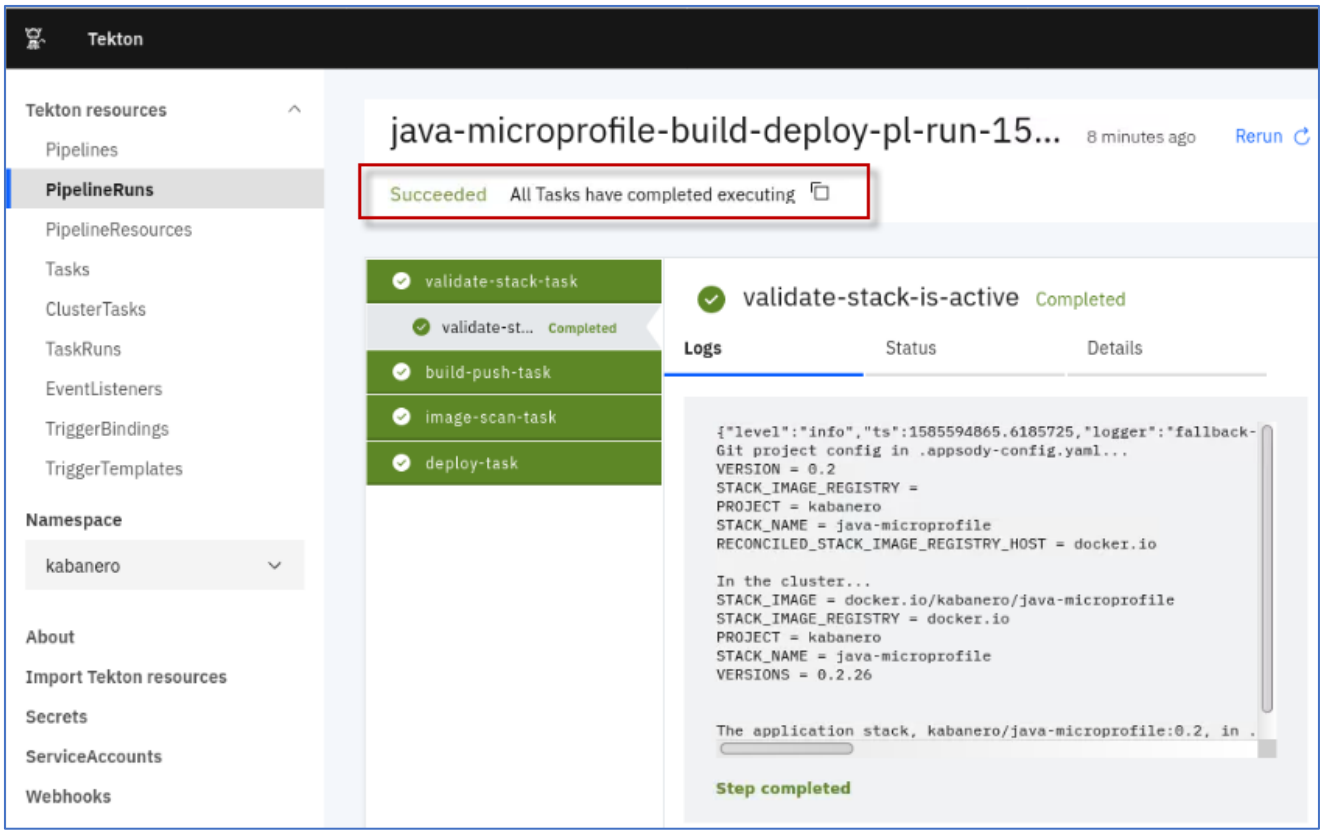
__2. Click **Create** to run the Pipeline



__3. Click on the name of the new pipeline to view the live status of the run:

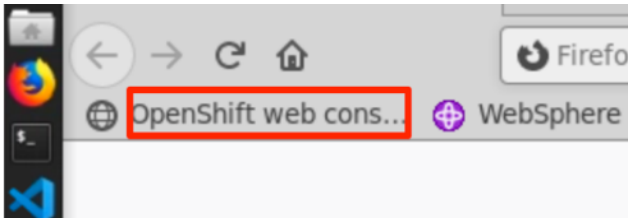


__4. Wait for the **PipelineRun** to complete. The Tekton pipeline will take **several minutes** to complete. As the Pipeline run, you can view its logs to see the details.



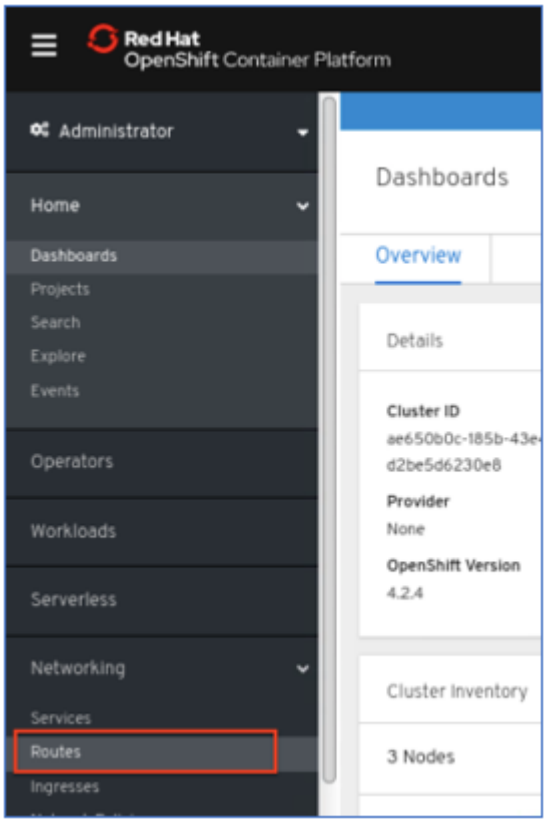
__5. When the run has completed successfully, view the running application.

__a. Click on the **OpenShift web console** bookmark

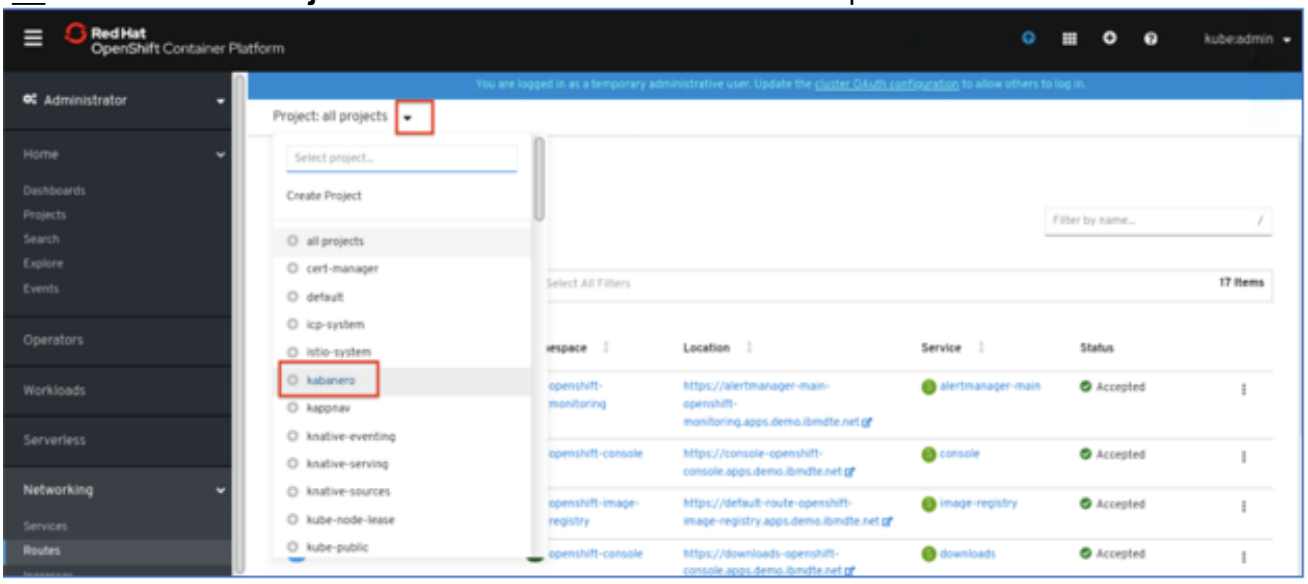


__b. If prompted to login to Red Hat OpenShift Container Platform, click **htpasswd** field. Then login with **ibmadmin**/**engageibm** as the username and password.

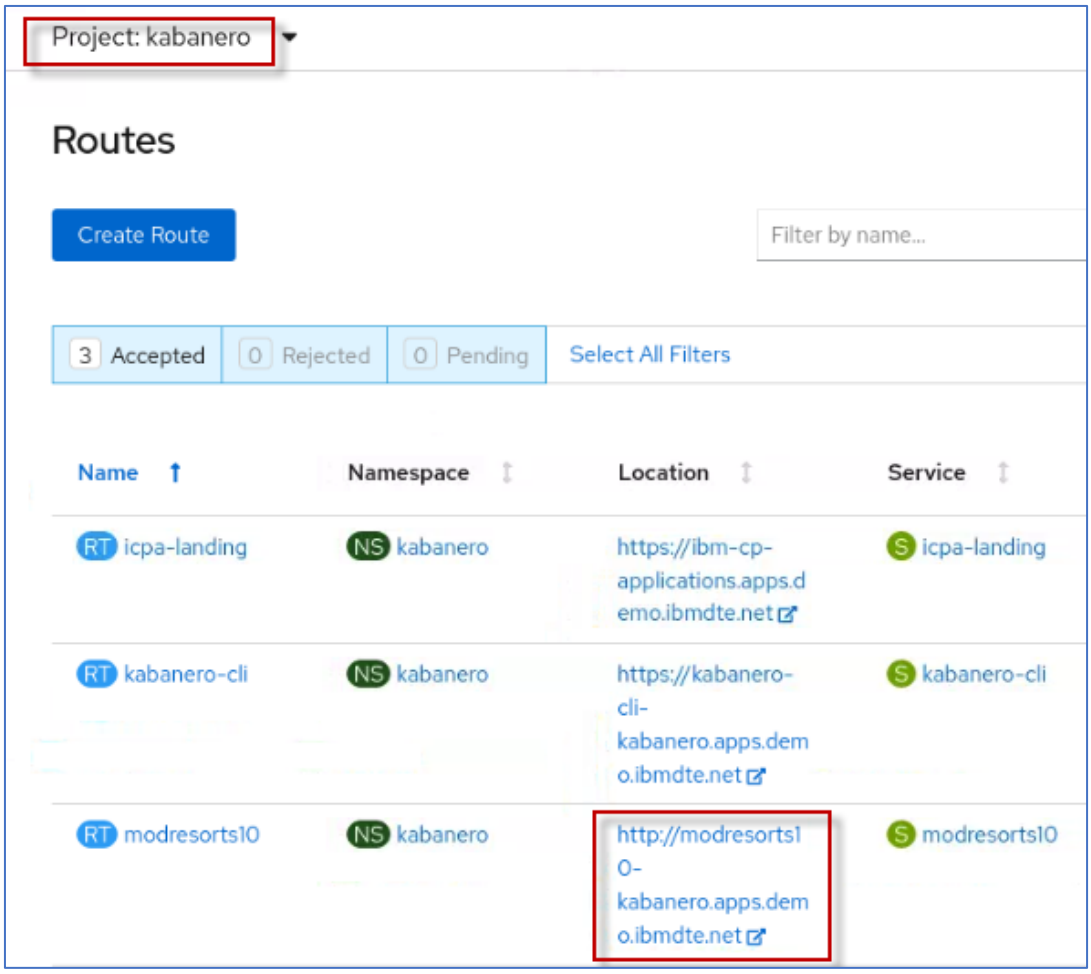
__c. Expand the **Networking** menu and select **Routes.**



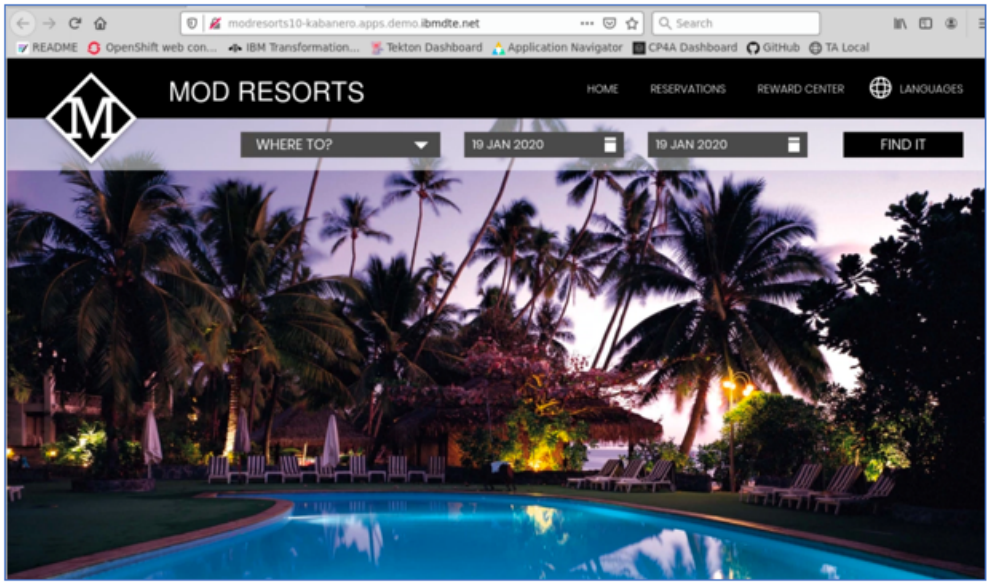
__d. Click on the **Project** filter and select the **kabanero** namespace.



__e. Click on your **Mod Resorts** application URL



__f. View your application running in OpenShift.



7. Summary

In this lab, you have learned how to modernize a Java application that is running in an on-premise application server environment using Transformation Advisor's integration with **Accelerator for Teams** in **IBM Cloud Pak for Applications**, then deploy it to OpenShift with Application Stacks and Integrated Dev/Ops Toolchain and Pipelines.

As a part of IBM Application Modernization solutions in **IBM Cloud Pak for Applications**, **Transformation Advisor** and the Accelerator for Teams (with Application Stacks and Integrated Dev/Ops Toolchain and Pipelines) helps users effectively to develop, package, deploy and manage modernized applications on a Kubernetes cluster.

To learn more about IBM Application Modernization solutions, please visit Cloud Pak for Applications.

Congratulations! You have successfully completed the lab "AM0210ST - Modernize a Java Application using IBM Transformation Advisor and Cloud Pak for App".