Migrating WebSphere Application to IBM Kubernetes Service Using Transformation Advisor

Overview

This document describes how to use the Migration Bundle produced by IBM Transformation Advisor to migrate your applications to IBM Kubernetes Service. There are a number of steps in the process, and each need to be completed in turn.

1. Migrate the application to IBM WebSphere Liberty profile

- You configure the Liberty profile for your application and add the necessary application and third party binaries.

2. Move Liberty to IBM Cloud Kubernetes Service – Prerequisites

Phase 1: Migrating to IBM WebSphere Liberty Profile

In this Step you will migrate your application to Liberty using the migration helper artifacts from the Transformation Advisor Migration Bundle.

Prerequisites

1. To complete this stage you require the following software to be installed:

- WebSphere Application Server Liberty profile

2. To complete this stage you need to copy the Migration Bundle that you downloaded from Transformation Advisor to the machine where you have installed the WebSphere Application Server Liberty profile. Unzip the bundle.

You will use the following items from the Migration Bundle:

1. server.xml
2. All binaries

In the following

tasks:

- <MIGRATION_BUNDLE_HOME> refers to the location where you have unzipped the Transformation Advisor Migration Bundle.
- Liberty profile refers to WebSphere Application Server Liberty profile.
- <LIBERTY_HOME> refers to the location where you have installed Liberty profile
- <LIBERTY_HOME_MACHINE_IP> refers to the IP address of the machine where you have installed Liberty profile
- <APP_CONTEXT> refers to the context for your application

Tasks

1. Create a server in the Liberty profile to run your application

   ```
   cd <LIBERTY_HOME>/bin ./server create server1
   ```

2. Move to the location of your unzipped Migration Bundle and copy the application binary (ear/war) from the Migration Bundle to the dropins directory of the WebSphere Application Server Liberty profile

   ```
   cd <MIGRATION_BUNDLE_HOME>
   cp binary/application/* <LIBERTY_HOME>/usr/servers/server1/dropins
   ```

3. Create a lib directory for any additional binaries that are required and copy them into place

mkdir <LIBERTY_HOME>/usr/servers/server1/lib

cp binary/lib/* <LIBERTY_HOME>/usr/servers/server1/lib

NOTE: If you did not upload all the binary files before generating the Migration Bundle you should place them here before running the above commands

4. Update the server.xml if necessary

   NOTE: You may wish to change the port numbers given for the httpEndpoint
   NOTE: All passwords have been replaced with '???'. Enter the correct password now.

   NOTE: If there are any additional binaries listed in this file that you do not need, you should remove any reference to them.

5. Copy the server.xml into place

   cp server.xml <LIBERTY_HOME>/usr/servers/server1/server.xml

6. Start the Liberty profile server

   <LIBERTY_HOME>/bin/server start server1

7. Check the Liberty profile logs to confirm that your application has started correctly and to find the URL to access it at

   cd <LIBERTY_HOME>/usr/servers/server1/logs

   vi messages.log

   NOTE: If you define a <dataSource> in server.xml you may encounter an authentication issue similar to this: ***invalid username/password; logon denied****. If you see this issue you may need to enter your username and password in line. Do this by adding the attributes 'user' and 'password' to the property. For example like this:*

   <properties.oracle portNumber="1521" URL="jdbc:oracle:thin:@9.9.9.9:1522:orcl" ***user="system" password ="TransAdv01"*** *...../>*

8. In the logs there should be a line similar to this:

   *TCP Channel defaultHttpEndpoint has been started and is now listening for requests on host * (IPv6) port 9080*

9. Open your application in the browser by going to the following link

   http:// <LIBERTY_HOME_MACHINE_IP>:9080/<APP_CONTEXT>

Note: The Migration Bundle provides artifacts to assist you in the migration. Depending on the nature and complexity of your application additional configuration may be required to fully complete this Task.

Phase 2: Move Liberty to IBM Cloud Kubernetes Service – Prerequisites

In this step you will need to install the following mandatory CLIs as prerequisites:

1. You have completed Phase 1: Migrating To WebSphere Liberty

2. To complete this stage you require the following software to be installed:
   Docker

3. The machine where you complete this task requires access to the internet to download the Liberty base image.

4. To complete this stage you require the following CLIs to be installed:
   The Helm CLI
The IBM Cloud CLI
The IBM Kubernetes CLI

The IBM Cloud Container Registry plug-in (Only need to run the command below, the link is for reference if issues are encountered)
`ibmcloud plugin install container-registry -r Bluemix`

The IBM Cloud Container Service plug-in(Only need to run the command below, the link is for reference if issues are encountered)
`ibmcloud plugin install container-service -r Bluemix`

Confirm that all required plugins (container-registry and container-service) are installed
`ibmcloud plugin list`

5. Have at least one cluster available on IKS. If not, create one following this guide
   And the easier way to do it is via IKS UI instead of CLI.

In this section you will learn all the steps required from IKS to accomplish the migration to public cloud.
General guide: https://console.bluemix.net/docs/containers/cs_tutorials_apps.html#cs_apps_tutorial

1. Log into IKS from a terminal

```
ibmcloud login –sso
```

Note: for internal IBMer we need the --sso flag, for external user we do not need this flag). Afterwards, a one time passcode link will be prompted. Go to that link and copy the passcode back to the terminal to proceed.

otherwise use an API key generated by the account owner

```
ibmcloud login -a https://api.ng.bluemix.net --apikey xxxxxxxxxxx
```

2. Change to the right region in which your cluster resides by running

   To check all available regions, use this command:

```
ibmcloud ks regions
```

To set the correct region:

```
ibmcloud cs region-set <region-name>
```

Pick one region name, and run e.g.

```
ibmcloud cs region-set us-south
```

3. Set the context for the cluster in your CLI. Run

   `ibmcloud cs cluster-config <clus-ter_name_or_ID>` To check all available clusters under the region:

```
ibmcloud ks clusters
```

4. Log into the IBM Cloud Container Registry CLI.

```
ibmcloud cr login
```

5. Check the namespace

```
ibmcloud cr namespace-list
```

Note: Please refer to the official IKS set up link above if any issues were encountered

In this section you will perform the following steps in Transformation Advisor to complete the migration

1. Copy the Migration Bundle that you downloaded from Transformation Advisor to the machine where you have completed the install of the CLIs above

2. Use docker to build an image. Build at the same level as Dockerfile.

   cd <MIGRATION_BUNDLE_HOME>

   Example:

   docker build -t registry.ng.bluemix.net/<namespace>/<instance-name>:<tag> .

Note: ng here represents the region, so be sure you are in the correct region where your cluster resides, and also ensure you are in the correct namespace.

Command to find out the namespace of your cluster:

ibmcloud cr namespace-list

3. Upload the docker image to the IKS private registry

   Example:

docker push -t registry.ng.bluemix.net/<namespace>/<instance-name>:<tag>

To check if your image has been successfully uploaded to the registry:

ibmcloud cr images-list

4. Edit the values.yaml file under <Migration Bundle>/chart/<application-name>/ as follows:

   Edit the repository value to the new IKS private registry link

For example:

repository: "registry.ng.bluemix.net/[namespace]/[instance-name]"

Also, edit the “tag” field to match the tag used in your IKS private registry

5. helm install inside <Migration Bundle>/chart/<application-name>/

   Example: helm install . --name <application-name>

6. To check the Nodeports of this application. Run:

   helm status <application-name>

Note: The nodeports information is in the command output under the column "PORTS" similar to this - 9080:30845/TCP, and in this case 30845 is the nodeport.

7. Find out the public IP for this cluster by running

   ibmcloud cs workers <cluster_name>

Note: The public IP information is in the command output under the column "PUBLIC IP"

8. Access the migrated app on IKS by going to this address:
<Public IP>:<Newly Mapped NodePort>

In this example, to access the migrated web app, the access point would be <Public_IP>:30845