

Deploy a Java App Using the IBM Cloud Command-Line Interface

Overview

In this lab, you will use the IBM Cloud (ibmcloud) command-line interface (CLI) to work with an IBM Cloud Foundry application. The ibmcloud CLI is a tool that you will use in a terminal or command window on your workstation.

Prerequisites

You need the following accounts and software:

- An [IBM Cloud account](#)
- An Internet Explorer, Firefox, or Chrome web browser
- Docker with a minimum version of 1.13.1
- If you are using Windows, you must use Windows 10 or later
- Maven
- Java version at least 1.7
- git

The IBM Cloud (ibmcloud) CLI will be installed as part of the lab exercise.

Section 1. Install the IBM Cloud CLI

1. In a web browser go to the [instruction page for installing the CLI](#).

If this method doesn't work for you, there is another method to install the CLI in Appendix A.

2. Copy the appropriate command for your operating system

Note: Windows users will have to run the command in PowerShell as an administrator. After installation, a system restart is required. **The ibmcloud CLI tool is not supported in a Cygwin bash shell on Windows. You may use Windows cmd, powershell, or git bash.**

Step 1: Run the install command

- For Mac and Linux, run the following command:

```
curl -sL https://ibm.biz/ibt-installer | bash
```

- For Windows 10, run the following command as an administrator:

```
Set-ExecutionPolicy Unrestricted; iex(New-Object Net.WebClient).DownloadStrin
```



Tip: Right-click the Windows PowerShell icon, and select **Run as administrator**.

Section 2. Using the IBM Cloud (ibmcloud) CLI

Now that your ibmcloud cli is installed, let's test it out.

1. In your terminal, type

```
$ ibmcloud --version
```

You should see the version number of 0.8.0 as seen below:

```
[Olivers-MacBook-Pro-2:CLI_lab odrodrig@us.ibm.com$ ibmcloud --version  
ibmcloud version 0.8.0+50d0d38c-2018-07-11T06:52:10+00:00
```

2. Next, use the “help” command to see what else can be done with the cli. Run the following command to see the CLI's capabilities:

```
$ ibmcloud help
```

```
[Olivers-MacBook-Pro-2:CLI_lab odrodrig@us.ibm.com$ ibmcloud help  
NAME:  
  ibmcloud - A command line tool to interact with IBM Cloud  
  
USAGE:  
  [environment variables] ibmcloud [global options] command [arguments...] [command options]  
  
VERSION:  
  0.8.0+50d0d38c-2018-07-11T06:52:10+00:00  
  
COMMANDS:  
  api          Set or view target API endpoint  
  login        Log user in
```

3. Now we can configure the CLI to target our IBM Cloud account. In your terminal, enter the following to connect to the API endpoint for the US South region:

```
$ ibmcloud api https://api.ng.bluemix.net
```

```
Olivers-MacBook-Pro-2:CLI_lab odrodrig@us.ibm.com$ ibmcloud api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK
API endpoint: https://api.ng.bluemix.net
```

4. Then, log in to your IBM Cloud account by using the following command and entering your credentials:

```
$ ibmcloud login
```

After logging in, you may see a list of accounts to target if you have been invited to other accounts. If so, select the number that corresponds to the account that you want to target.

```
Targeted resource group default

API endpoint: https://api.ng.bluemix.net
Region:      us-south
User:        odrodrig@us.ibm.com
Account:     OLIVER RODRIGUEZ's Account (532f5fed73d46ecfb52d9984bc526b7e)
Resource group: default
CF API endpoint:
Org:
Space:
```

5. Now we need to set the CF API endpoint. This endpoint is used to manage Cloud Foundry applications. Enter the following in your terminal:

```
$ ibmcloud target --cf
```

After running the command, you may be asked to select a space if you have more than one. Select the number that corresponds to the space that you would like to target.

```

Olivers-MacBook-Pro-2:CLI_lab odrodrig@us.ibm.com$ ibmcloud target --cf
Targeted Cloud Foundry (https://api.ng.bluemix.net)

Targeted org odrodrig@us.ibm.com

Select a space (or press enter to skip):
1. dev
2. Demo
Enter a number> 2
Targeted space Demo

API endpoint: https://api.ng.bluemix.net
Region: us-south
User: odrodrig@us.ibm.com
Account: OLIVER RODRIGUEZ's Account (532f5fed73d46ecfb52d9984bc526b7e)
Resource group: default
CF API endpoint: https://api.ng.bluemix.net (API version: 2.92.0)
Org: odrodrig@us.ibm.com
Space: Demo

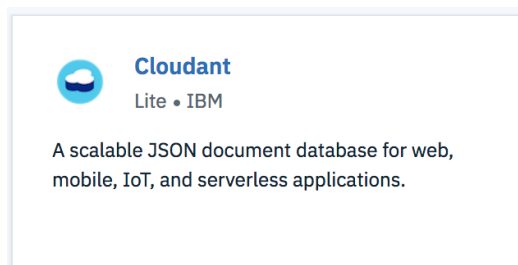
```

Section 4. Creating a service

The app that we will be deploying later is going to be using the Cloudant NoSQL database so first we need to create an instance of it from the IBM Cloud catalog.

Although you can create a service through the cli (see Appendix B), for this lab we will be using the gui to create the service.

1. From IBM Cloud, click on **Catalog** at the top of the page.
2. Once in the catalog, click on the **Databases** category on the left panel.
3. Find and click on the **Cloudant** service.



- Take note of the service name, ensure the region selected is **US South** and that the organization and space selected are correct, and then select **Create**.

Service name:

Cloudant-53

Choose a region/location to deploy in:

US South

Choose an organization:

odrodrig@us.ibm.com

Choose a space:

Demo

Pricing Plans

Monthly prices shown are for country or region: [United States](#)

PLAN	FEATURES	PRICING
✓ Lite	1 GB of data storage Provisioned throughput capacity fixed at: 20 Lookups/sec 10 Writes/sec 5 Queries/sec	Free
<div>The Lite plan provides access to the full functionality of Cloudant for development and evaluation. The plan has a set amount of provisioned throughput capacity as shown and includes a max of 1GB of encrypted data storage.</div> <div>Lite plan services are deleted after 30 days of inactivity.</div>		
Standard	20 GB of free data storage Throughput capacity scales in fixed ratio of: 100 Lookups/sec 50 Writes/sec 5 Queries/sec	\$1.00 USD/GB of data storage \$0.25 USD/Lookup per second \$0.50 USD/Write per second \$5.00 USD/Query per second

Create

Section 3. Getting the application code

In this lab, we will deploy a sample Java app. The app itself is very simple favorites organizer that allows you to upload documents. The documents are then stored in a Cloudant NoSQL database.

- First, go to [this repository](#)
- Click on the green “**Clone or Download**” button on the upper right of the page.
- Go to your terminal and type:

```
$ git clone https://github.com/odrodrig/java-cloudant.git
```

Feel free to take time now to explore the contents of the project

Section 4. Building and Running the app in the command-line

In this section, we will use the maven cli and the ibmcloud cli to build and deploy our application.

If you are interested in using Eclipse to build and deploy the application, the instructions can be found in **Appendix D**.

1. Navigate to the directory where you cloned the project.
2. Execute full Maven build to create the `target/JavaCloudantApp.war` file:

```
$ mvn clean install
```

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 2.887 s  
[INFO] Finished at: 2018-07-16T02:04:41-05:00  
[INFO] -----
```

3. Download and start a local Liberty server with the application:

```
$ mvn liberty:run-server
```

```
[INFO] [AUDIT   ] CWWKF0011I: The server defaultServer is ready to run a smarter planet.
```

4. Once the server is running, the application will be available under <http://localhost:9080/JavaCloudantApp>

You should be able to load the page but there will be a red Error underneath the Help link.

At this point the application only partially functions because it can't find the service credentials for the Cloudant database in the environment. When running in IBM Cloud, we bind the service instance to the application, this will make the service credentials available to the application through environment variables.

5. Update the application details in the `manifest.yml` file. IBM Cloud will reference the manifest when deploying the application to get the application name, memory allocation, services to bind, and more. Open the manifest.yml file and make the following edits:
 - Change the name of your application to something unique. Typically adding your initials to the beginning of the app name should be enough.

- Change the Cloudant service name under **services**. You can find the Cloudant service name by going to the IBM Cloud dashboard and finding the Cloudant instance.

Cloudant-53

US South

Your manifest.yml should look similar to the one below except for the different app name and service name. Be sure to save your changes.

```
1  ---
2  applications:
3  | - name: odrjava-cloudant
4  |   memory: 256M
5  |   path: target/JavaCloudantApp.war
6  |   services:
7  |     - Cloudant-53
```

Here's a little explanation of what is going on in each of the fields in this manifest:

- Name – Application name
- Memory – The memory allocation of the application. IBM Cloud Lite accounts only have 256MB to play with so you may need to stop other apps that are running to make space available.
- Path – This is where IBM Cloud will look for your application. In this case, we are pointing to our WAR file in the target directory.
- Services – Any services listed here will be bound to your application during deployment. It is important to note that the services must have been created beforehand in order for this to work.

6. Now time to deploy. Ensure you are still targeting the correct region, org, and space by running the following command:

```
$ ibmcloud target
```

```
Olivers-MacBook-Pro-2:javaLab odrodrig@us.ibm.com$ ibmcloud target

API endpoint:      https://api.ng.bluemix.net
Region:           us-south
User:             odrodrig@us.ibm.com
Account:          OLIVER RODRIGUEZ's Account (532f5fed73d46ecfb52d9984bc526b7e)
Resource group:   default
CF API endpoint:   https://api.ng.bluemix.net (API version: 2.92.0)
Org:              odrodrig@us.ibm.com
Space:            Demo
```

7. Run the following command to push the application to IBM Cloud:

```
$ ibmcloud app push
```

```
Olivers-MacBook-Pro-2:javaLab odrodrig@us.ibm.com$ ibmcloud app push
Invoking 'cf push'...

Pushing from manifest to org odrodrig@us.ibm.com / space Demo as odrodrig@us.ibm.com...
Using manifest file /Users/odrodrig@us.ibm.com/Documents/projects/bpDojo/javaLab/manifest.yml
Getting app info...
Creating app with these attributes...
+ name:      odrjava-cloudant
+ path:      /Users/odrodrig@us.ibm.com/Documents/projects/bpDojo/javaLab/target/JavaCloudantApp.war
+ memory:    256M
+ services:
+   Cloudant-53
+ routes:
+   odrjava-cloudant.mybluemix.net
```

Right away you will notice that your manifest.yml file has been recognized and the attributes you specified will be used for the deployment.

The deployment will take a minute or two but when it's done, you should see the following:

	state	since	cpu	memory	disk	details
#0	running	2018-07-16T07:32:39Z	175.9%	130.3M of 256M	204.7M of 1G	

8. Go to your IBM Cloud dashboard and look for your newly created app and click on it.
9. Find the **Visit App URL** link at the top of the page and click on it to go to the application.
10. Now you can play with the app check out the awesome sample.txt file and even upload your own files.

Section 5. Updating and redeploying the app

Now that we have an app deployed, let's go through the steps to make an update and push the change.

1. Open your application in a code editor and navigate to the following file:

```
../src/main/webapp/index.html
```

2. Find the appTitle on line 15 that starts with "Favorites Organizer ...".
3. Add a few exclamation points to the end of the title to show how excited we really are.

```
14      <p class = "appTitle">
15      Favorites Organizer powered by Cloudant!!!!|
16    </p>
```

4. Save your changes
5. In your terminal run the following command to build the application:

```
$ mvn clean install
```

6. Deploy the application again by running the following command:

```
$ ibmcloud app push
```

7. After your app has successfully deployed, refresh the page in the browser and you should see you new more exciting title.



Favorites Organizer powered by Cloudant!!!!!

Summary and Clean up

In this lab, you learned how to use the IBM Cloud (ibmcloud) CLI tool to manage an application including deployment, binding application services, and updating the application.

1. **Note:** Check with your lab host before completing this step. If the workshop will be continuing with the local hosting of a CF application lab exercise, the application and service should not be deleted.

Otherwise, delete the application and service and confirm the deletion by running the following two commands:

Delete the application: `ibmcloud app delete bmx-doj0-webapp1 -r`

- `bmx-doj0-webapp1` is the application name to be deleted.
- `-r` instructs IBM Cloud to also delete the routes attached to the application.

Delete the service: `ibmcloud service delete myCloudantDB`

- `myCloudantDB` is the service instance to be deleted.

You can confirm that the application and service were deleted by checking the dashboard in the IBM Cloud web UI.

Appendix A. Optional install method

1. In a web browser, open the [download page](#).

Getting started with IBM Cloud CLI

Last Updated: 2018-06-21 | [Edit in GitHub](#)



Tip: It is recommended that you install the IBM Cloud CLI and all recommended dependencies by using the method that is described [here](#).

IBM Cloud CLI provides the command line interface to manage applications, containers, infrastructures, services, and other resources in IBM Cloud.

To get started with just the IBM Cloud CLI:

- ① Select the installer of your OS to download

Mac OS X 64 bit: [installer](#) / [sha1sums](#)

Windows 64 bit: [installer](#) / [sha1sums](#)

Linux X86 64 bit: [installer](#) / [sha1sums](#)

Linux LE 64-bit (ppc64le): [installer](#) / [sha1sums](#)

**32-bit releases and previous versions can be found [here](#)

2. Click on the download link specific to your workstation platform to download the package.
3. Follow the steps appropriate to your workstation operating system.

For MacOS and Windows, after downloading the package, run the installer.

For Linux, unpack the tarball and then use sudo (or root if necessary) to run the installer (replacing with the file name of your download).

```
$ tar -xvf IBM_Cloud_CLI_0.8.0_amd64.tar.gz
$ cd Bluemix_CLI
$ sudo ./install_bluemix_cli
```

Section A.2 Using the IBM Cloud (ibmcloud) CLI

1. Log in to IBM Cloud by issuing one of the following commands:

```
ibmcloud login -a https://api.ng.bluemix.net (Region: US South)
```

```
ibmcloud login -a https://api.eu-gb.bluemix.net (Region: United Kingdom)
```

```
ibmcloud login -a https://api.eu-de.bluemix.net (Region: Frankfurt)
```

```
ibmcloud login -a https://api.au-syd.bluemix.net (Region: Sydney)
```

2. Enter the email and password that you used to log in to the IBM Cloud web UI.
3. Select organization and space to use for the deployment interactively using (if you only have one organization and space in the account, they will automatically be selected):

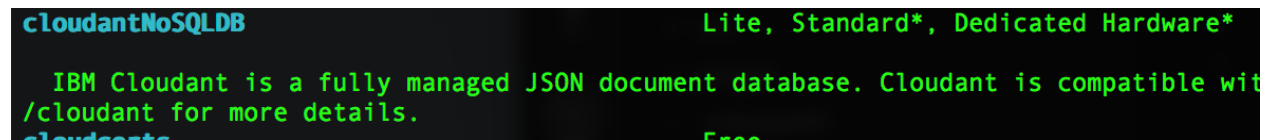
```
ibmcloud target -cf
```

Appendix B. Creating and binding a service through the CLI

1. You can view the available services by running the following command but **be warned this will take a while to load as the command collects all the catalog entries**:

```
ibmcloud service offerings
```

2. In the list of services, note the cloudantNoSQLDB service.



```
cloudantNoSQLDB                               Lite, Standard*, Dedicated Hardware*
IBM Cloudant is a fully managed JSON document database. Cloudant is compatible with
MongoDB. For more details, see https://cloudant.com/docs/cloudant-compatibility-mongodb/
cloudantNoSQLDB                               Free
```

3. Create the service by running this command:

```
ibmcloud service create cloudantNoSQLDB Lite myCloudantDB
```

where:

- **CloudantNoSQLDB** is the name of the service from the `ibmcloud service offerings` command.
- **Lite** is the name of the service plan that you want to use for the service
- **myCloudantDB** is the name of the service instance that you want to use. You can use any name here or just keep it as `myCloudantDB`. This name is required to be unique within your IBM Cloud Foundry space.

4. You can bind a service instance to an application by using the following command. Substitute the application name and service instance names that you used previously:

```
ibmcloud service bind bmx-doj0-webapp1 myCloudantDB
```

where:

- `bmxdodo-webapp1` is the application name used for the ibmcloud app push.
- `myCloudantDB` is the service instance name used when the service was created.

If you open the application dashboard in the web UI, you will see that the service is now showing in the connections list for the application.

Appendix C. SSH into an app container

There are times when a developer really wants to see what is going on in the actual environment of the application. IBM Cloud Foundry uses the Diego architecture of Cloud Foundry so you can ssh to the container running your application if needed. Try this with your application now:

```
ibmcloud cf ssh bmxdodo-webapp1
```

The application artifacts are all in the app subdirectory of this shell. From the ssh to the container hosting the application you can use common tools like `top` or modify file data of the application. However, this filesystem is ephemeral and will be lost if the application is restaged. Typing `exit` will close the session as usual.

This IBM Cloud CLI command also shows another feature. You can run standard Cloud Foundry CLI (`cf`) commands in the context of your IBM Cloud login by placing the command after `ibmcloud`.

Appendix D. Deploy an application through Eclipse

IBM® Eclipse Tools for IBM Cloud® provides plug-ins that can be installed into an existing Eclipse environment to assist in integrating the developer's integrated development environment (IDE) with IBM Cloud.

1. Download and install IBM Eclipse Tools for IBM Cloud by going to <https://developer.ibm.com/wasdev/downloads/#asset/tools-IBM-Eclipse-Tools-for-IBM-Cloud>.
2. Import this sample into Eclipse using `File` -> `Import` -> `Maven` -> `Existing Maven Projects` option.
3. Create a Liberty server definition:
 - In the `Servers` view right-click -> `New` -> `Server`
 - Select `IBM` -> `WebSphere Application Server Liberty`

- Choose `Install from an archive or a repository`
- Enter a destination path (/Users/username/liberty)
- Choose `WAS Liberty with Java EE 7 Web Profile`
- Continue the wizard with default options to Finish

4. Run your application locally on Liberty:

- Right click on the `JavaCloudantApp` sample and select `Run As` -> `Run on Server` option
- Find and select the localhost Liberty server and press `Finish`
- In a few seconds, your application should be running at <http://localhost:9080/JavaHelloWorldApp/>

5. Create an IBM Cloud server definition:

- In the `Servers` view, right-click -> `New` -> `Server`
- Select `IBM` -> `IBM Bluemix` and follow the steps in the wizard.
- Enter your credentials and click `Next`
- Select your `org` and `space` and click `Finish`

6. Run your application on IBM Cloud:

- Right click on the `JavaCloudantApp` sample and select `Run As` -> `Run on Server` option
- Find and select the `IBM Bluemix` and press `Finish`
- A wizard will guide you with the deployment options.
- Select your Cloudant service on the Services step
- In a few minutes, your application should be running at the URL you chose.

Now you have your code running locally and on the cloud!

© Copyright IBM Corporation 2018

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

This document is current as of the initial date of publication and may be changed by IBM at any time.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.