

Run IBM MobileFirst Platform Foundation on IBM Containers

Overview

This tutorial demonstrates how to take a locally developed IBM MobileFirst Platform Foundation project and run it on Bluemix. To achieve this result, you go through the following steps: set up your host computer with the required tools (MobileFirst CLI, Docker, and IBM Containers Extension (ICE) CLI), set up your Bluemix environment, build a MobileFirst Platform Foundation Server image, deploy your project runtime and push it to the Bluemix repository. Finally, you run the image on an IBM Container and update it with the MobileFirst project application and adapter.

Note: Windows OS is currently not supported.

Note: The MobileFirst Server Configuration Tools cannot be used for deployments to IBM Containers.

Prerequisite: Make sure to read the Introduction to IBM MobileFirst Platform Foundation on IBM Containers (../) tutorial.

Topics

- Register an account at Bluemix
- Set up your host machine
- Run IBM MobileFirst Platform Foundation on IBM Containers

Register an account at Bluemix

If you do not yet have an account, visit the Bluemix website (<http://www.bluemix.net>) and click **Get Started Free** or **Sign Up**. You'll need to fill up a registration form before you can move on to the next step.

The Bluemix Dashboard

After signing in to Bluemix, you are presented with the Bluemix Dashboard, which provides an overview of the active Bluemix **space**. By default, this work area receives the name "dev". You can create multiple work areas/spaces if needed.

Set up your host machine

To manage containers and images, you need to install the following tools: IBM MobileFirst Platform Foundation CLI, Docker, and IBM Containers Extension (ICE) CLI.

MobileFirst Platform Foundation CLI

Follow the Using CLI to create, build, and manage MobileFirst project artifacts (../advanced-client-side-development/using-cli-to-create-build-and-manage-mobilefirst-project-artifacts/) tutorial to install the MobileFirst Command Line Interface.

Docker

Go to the Docker Documentation (<https://docs.docker.com/>) > on the left menu, select **Install > Docker Engine**, select your OS type and follow the instructions to install the Docker Toolbox.

Note: IBM does not support Docker's Kitematic software.

In OS X there are two options to run Docker commands:

- From the OS X Terminal
- From the Docker Quickstart Terminal

If you choose to work from the Docker Quickstart Terminal no further setup is needed. You must work only from it.

If you choose to work from the OS X Terminal, do the following:

- Run the command: `docker-machine env default`
- Set the result as environment variables, for example:

```
$ docker-machine env default
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.101:2376"
export DOCKER_CERT_PATH="/Users/mary/.docker/machine/machines/default"
export DOCKER_MACHINE_NAME="default"
```

For further information consult the Docker documentation.

IBM Containers Extension (ICE)

Prerequisites: Before you install the ICE CLI tool, you must first install Python, Python Setuptools, Python Pip, and Cloud Foundry CLI.

Installing Python, Python Pip, and Python Setuptools

1. Install Python, Python Pip, and Python Setuptools:
 - Linux (<http://docs.python-guide.org/en/latest/starting/install/linux/>)
 - Mac OS X (<http://docs.python-guide.org/en/latest/starting/install/osx/>)
 - Windows (<http://docs.python-guide.org/en/latest/starting/install/win/>)
2. Install the Cloud Foundry CLI from the Cloud Foundry CLI GitHub repository (<https://github.com/cloudfoundry/cli/releases>).

Installing ICE

- Install the IBM Containers Extension by running:

- It can be 4 - 30 characters. If you plan to manage containers from the command line, you might prefer to have a short namespace that can be typed quickly.
- It must be unique in the Bluemix registry.

To set a namespace, run the command:

```
$ ice namespace set <new_name>
```

To get the namespace that you have set, run the command:

```
$ ice namespace get
```

To learn more about ICE commands, use the `ice help` command.

Step 3: Using the configuration files

Note: If you choose to run the scripts interactively, you can skip the configuration but it is strongly suggested to at least read and understand the arguments you will need to provide.

The `args` folder contains a set of configuration files which contain the arguments that are required to run the scripts. Fill in the arguments' values in the following files:

```
initenv.properties
prepareserverdbs.properties
prepareserver.properties
startserver.properties
```

Step 4: Running the scripts

As explained above you can choose to run the scripts interactively or by using the configuration files:

- Using the configuration files - run the scripts and pass the respective configuration file as an argument
- Interactively - run the scripts without any arguments

The following demonstrate the first option.

1. `installcontainercli.sh` - Adding Container Extension to the MobileFirst CLI

In order to use the Container Extension you must first add it to the MobileFirst CLI.

Run:

```
$ ./installcontainercli.sh
```

Note: You may need to use `sudo`

2. `initenv.sh` - Logging in to Bluemix

Run the `initenv.sh` script in order to create an environment for building and running IBM MobileFirst Platform Foundation on the IBM Containers:

```
$ ./initenv.sh args/initenv.properties
```

3. `prepareserverdbs.sh` - Prepare the MobileFirst Server database

The `prepareserverdbs.sh` script is used to configure your MobileFirst project database. You will need to run it separately, once for the admin database and once for every MobileFirst project runtime database.

- For the admin database make sure to comment out the `RUNTIME_NAME` argument and run:

```
$ ./prepareserverdbs.sh args/prepareserverdbs.properties
```

- For each MobileFirst project runtime database - first uncomment the project `RUNTIME_NAME` argument, change it value to match the specific project war file and run:

```
$ ./prepareserverdbs.sh args/prepareserverdbs.properties
```

Note: If you are getting an error: "Application not configured correctly" - try to run the script (with the same properties) again.

4. `prepareserver.sh` - Prepare a Mobilefirst Platform Foundation Server image

Uncomment the `PROJECT_LOC` argument and run the `prepareserver.sh` script in order to build a MobileFirst Platform Foundation Server image, deploy your project runtime and push it to your Bluemix repository:

```
$ ./prepareserver.sh args/prepareserver.properties
```

To view all available images in your Bluemix repository run:

```
$ ice images
```

The list contains the image name, date of creation and ID.

5. `startserver.sh` - Running the image on an IBM Container

The `startserver.sh` script is used to run the Mobilefirst Server image on an IBM Container. It also Binds your image to the public IP you configured in the `SERVER_IP` property.

- Run:

```
$ ./startserver.sh args/startserver.properties
```

- Launch the MobileFirst Console by loading the following URL: <http://:9080/mfpconsole> (it may take a few moments).
- Upload the `.wlap` and `.adapter` files.

- MobileFirst Operations Console

Analytics Console

Hello, admin

Runtimes

InvokingAdapterProcedures

InvokingAdapterProcedures

Add new app or adapter

Applications (1)

Adapters (1)

Devices (0)

Push Notifications

Client Log Profiles

License Tracking

Error Log

Applications

Sort by: Name

HybridInvoking

Environments (2)

Last modified on Jun 22, 2015, 3:47 PM

application summary

Adapters

Name

Deploy time

RSSReader

Jun 22, 2015, 3:46 PM

Devices