MobileFirst Platform {dev}

# Run IBM MobileFirst Platform Foundation on IBM Containers

Relevant to:

Native iOS Native Android Native Windows Phone 8 Native Windows 8
Hybrid

## 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.

**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 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, Docke, and IBM Containers Extension (ICE) CLI.

### MobileFirst Platform Foundation CLI

Follow the Using CLI to create, build, and manage MobileFirst project artifacts tutorial to install the MobileFirst Command Line Interface.

## Docker

Go to the Docker Documentation > 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 Kitematic software.

## 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
    - Mac OS X
    - Windows
2. Install the Cloud Foundry CLI from the Cloud Foundry CLI GitHub repository.

### Installing ICE

- Install the IBM Containers Extension by running:

```
$ sudo pip install https://static-ice.ng.bluemix.net/icecli-3.0.zip
```

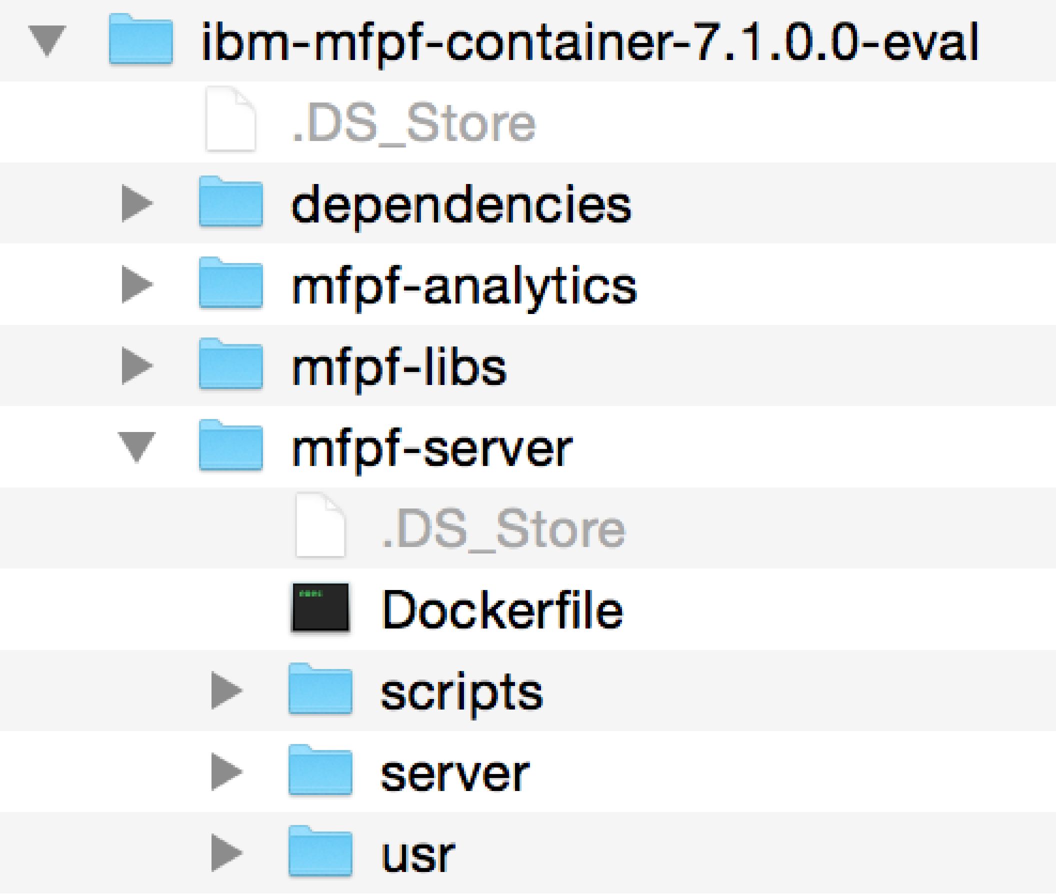
# Run IBM MobileFirst Platform Foundation on IBM Containers

To run IBM MobileFirst Platform Foundation on IBM Containers, you must first create an image that will later be pushed to Bluemix.

If you have not downloaded the IBM MobileFirst Platform Foundation Evaluation on Containers v7.1 ZIP file yet, click the button below, accept the license, and download it.

Review the license and download the ibm-mfpf-container-7.1.0.0-eval.zip

## Structure of the ibm-mfpf-container-7.1.0.0-eval.zip archive



The extracted ZIP file contains the files for building an image (`dependencies` and `mfpf-libs`), the files for building and deploying an IBM MobileFirst Platform Foundation Operational Analytics Container (`mfpf-analytics`), and files for configuring an IBM MobileFirst Platform Server Container (`mfpf-server`). This tutorial does not cover the analytics part.

**The mfpf-server folder**

- `Dockerfile`: text document that contains all the commands in order to build an image.
- `usr` folder:
  - `config` folder: Contains server key store configuration, User registry configuration, MobileFirst Platform Foundation Server properties (includes runtime configuration – analytics, attribute store etc).
  - `env` folder: Contains server environment configuration (ports, application root names etc).
  - `projects` folder: The location of your MobileFirst Platform project runtime (`.war` file).
  - `security` folder: The key store, trust store and the LTPA keys files (ltpa.keys) should be placed here.
  - `ssh` folder: Contains the id_rsa.pub file – the ssh public key file to enable ssh on the container.
  - `wxs` folder: Contains the data cache / extreme scale client library when Data Cache is used as attribute store for the server.

  > This tutorial refers only to the `projects` folder.

- `server` folder: Contains elements that are required for the IBM MobileFirst Platform Foundation Operational Server deployment.

- `scripts` folder: This folder contains the `args` folder, which contains a set of configuration files. It also contains scripts to run for logging in to Bluemix, building a Mobilefirst Platform Foundation Server image, deploying your project runtime, and for pushing and running the image on Bluemix. You can choose to run the scripts interactively or by pre-configuring the configuration files. See Step 3: Using the configuration files.

## Step 1: Create an IBM MobileFirst Platform Foundation project

Create a new MobileFirst project or use an existing one. You can find tutorials on how to create a new project, and their associated sample projects, in the Getting Started with Foundation page.

## Step 2: Prerequisites

1. **`ice login`**: To run ICE commands, you must first log in into the IBM Container Cloud Service.
   This step is mandatory because you will be running ICE commands during the following step.
   Run:

   ```
   $ ice login
   ```

   When prompted, enter the following information:

   - Email
   - Password
   - Organization, if you have more than one
   - Space, if you have more than one

2. Make sure that the namespace for container registry is set.
   - To set a namespace, run the command: `ice your-namespace set`.
   - 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.

initenv.properties

- **BLUEMIX_API_URL –** Bluemix API endpoint. The default is *https://api.ng.bluemix.net*.
- **BLUEMIX_REGISTRY –** The IBM Containers registry domain. The default is *registry.ng.bluemix.net*.
- **BLUEMIX_CCS_HOST –** The IBM Container Cloud Service Host. The default is *https://containers-api.ng.bluemix.net/v3/containers*.
- **BLUEMIX_USER –** Your Bluemix username (email).
- **BLUEMIX_PASSWORD –** Your Bluemix password.
- **BLUEMIX_ORG –** Your Bluemix organization name.
- **BLUEMIX_SPACE –** Your Bluemix space (as explained previously).

prepareserverdbs.properties

- **DB_TYPE –** Bluemix DB service type (sqldb, cloudantNoSQLDB).
- **DB_SRV_NAME –** Your Bluemix DB service instance name.
- **DB_SRV_PLAN –** Bluemix database service plan.
  For SQL DB, the accepted values are sqldb_small, sqldb_free, sqldb_premium.
  For Cloudant DB, the accepted value is *Shared*.
- **APP_NAME –** Your Bluemix DB application name.
  **Note:** Choose a unique name.
- **RUNTIME_NAME –** The MobileFirst project runtime name. Required for configuring runtime databases only, as explained in the next step.
- **SCHEMA_NAME –** Your database schema name. The default names are:
  - For the admin database: WLDAMIN
  - For the MobileFirst project runtime database: the value of RUNTIME_NAME

prepareserver.properties

- **SERVER_IMAGE_TAG –** A tag for the image. Should be of the form: *registry-url/repository-namespace/your-tag*, where the repository namespace is a unique name to identify your private repository on the Bluemix registry. The namespace is assigned once and for all for an organization and cannot be changed.

  To set the namespace, run:

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

- **PROJECT_LOC –** A path to the root directory of your MobileFirst project. Multiple project locations can be delimited by a comma.

startserver.properties

- **SERVER_IMAGE_TAG –** Same as in *prepareserver.sh*.
- **SERVER_CONTAINER_NAME –** A name for your Bluemix Container.
- **SERVER_IP –** An IP address that the Bluemix Container should be bound to. To assign an IP address, run:

  ```
  $ ice ip request
  ```

  IP addresses can be reused in multiple containers in a space. If you’ve already assigned one, you can run:

  ```
  $ ice ip list
  ```

## 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.

**Launch the Docker Toolbox terminal to run the scripts:**

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:

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

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 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.

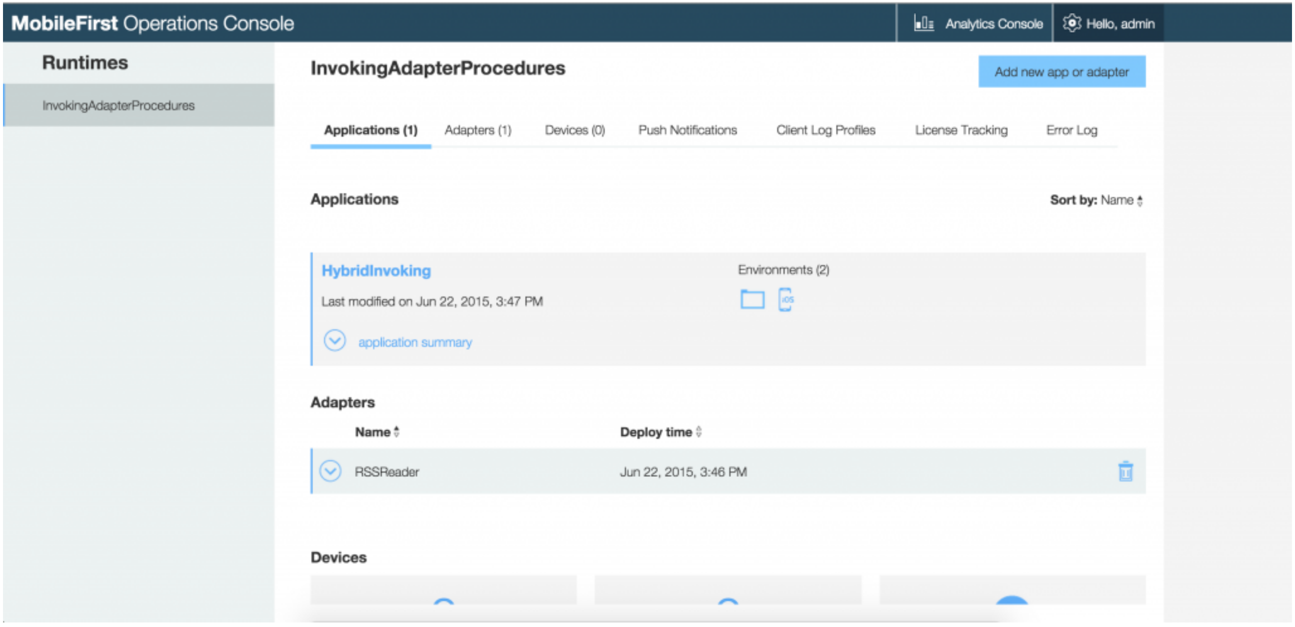
5. **startserver.sh – Running the image on an IBM Container**
   The `startserver.sh` script is used to run the Mobilefirst Server image on an IBM Container. It also Binds your image to the public IP you configured in the `SERVER_IP` property.

   - Run:

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

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

- Update the application’s `worklight.plist` (for iOS) and/or `wlclient.properties` (for Android, Windows Universal, Windows Phone) with the protocol, host and port values of the IBM Container.
- You can now run your application to verify that it successfully connects to the MobileFirst Server, running in Bluemix.


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
Last modified on Jun 22, 2015, 3:47 PM
Environments (2)
application summary
Adapters
Name
Deploy time
RSSReader
Jun 22, 2015, 3:46 PM
Devices