

# Setting Up the MobileFirst Server on IBM Containers using Scripts

## Overview

This tutorial demonstrates how to take a locally developed IBM MobileFirst 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 (Cloud Foundry CLI, Docker, and IBM Containers Extension (cf ic) Plug-in), set up your Bluemix environment, build a MobileFirst Foundation Server image and push it to the Bluemix repository. Finally, you run the image on IBM Containers as a single Container or a Container group 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 IBM MobileFirst Foundation on IBM Containers (../) tutorial.

Jump to:

- Register an account at Bluemix
- Set up your host machine
- Download the ibm-mfpf-container-8.0.0.0 zip
- Prerequisites
- Setting Up the MobileFirst and Analytics Servers on IBM Containers

## Register an account at Bluemix

If you do not yet have an account, visit the Bluemix website (<https://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: Docker, Cloud Foundry CLI and IBM Containers (cf ic) plugin.

## 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.

## Cloud Foundry Plugin and IBM Containers plugin

1. Install the Cloud Foundry CLI ([https://github.com/cloudfoundry/cli/releases?cm\\_mc\\_uid=85906649576514533887001&cm\\_mc\\_sid\\_50200000=1454307195](https://github.com/cloudfoundry/cli/releases?cm_mc_uid=85906649576514533887001&cm_mc_sid_50200000=1454307195)).
2. Install the IBM Containers Plugin (cf ic) ([https://www.ng.bluemix.net/docs/containers/container\\_cli\\_ov.html#container\\_cli\\_cfic](https://www.ng.bluemix.net/docs/containers/container_cli_ov.html#container_cli_cfic)).

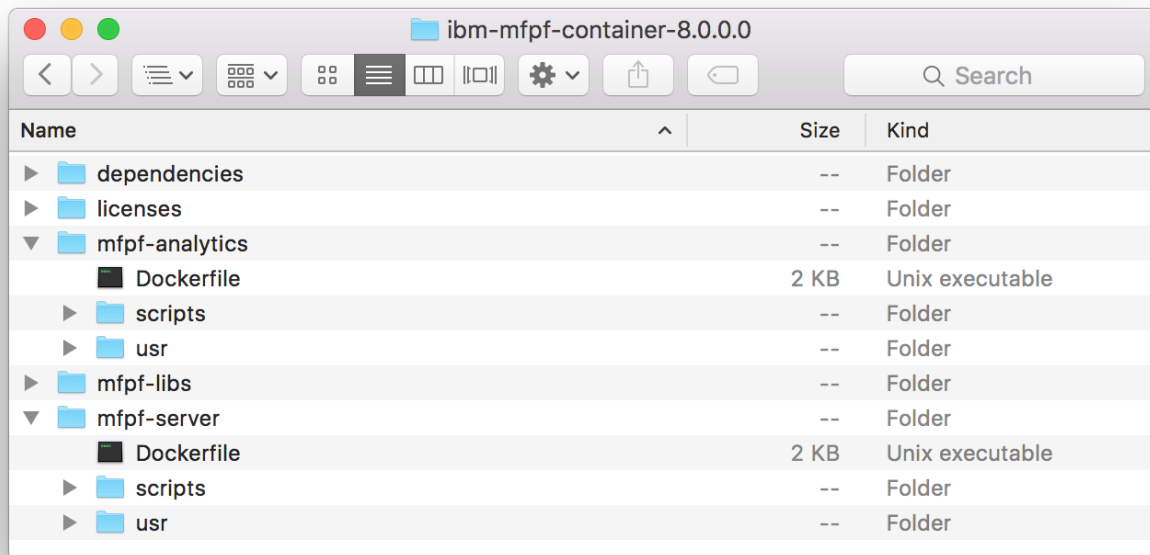
## Download the ibm-mfpf-container-8.0.0.0 zip

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

Follow the instructions and download the IBM MobileFirst Server for IBM Container v8.0 .zip file (search for: CNBL0EN) (<http://www-01.ibm.com/support/docview.wss?uid=swg2C7000005>)

## Structure of the mfp-cloud-container-8.0.0.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 Foundation Operational Analytics Container (**mfpf-analytics**) and files for configuring an IBM MobileFirst Foundation Server Container (**mfpf-server**).



## The mfpf-server and mfpf-analytics folders

- **Dockerfile**: Text document that contains all the commands in order to build an image.
- **scripts** folder: This folder contains the **args** folder, which contains a set of configuration files. It also contains scripts to run for logging into Bluemix, building a Mobilefirst Foundation Server/MobileFirst Foundation Operational Analytics image and for pushing and running the image on Bluemix. You can choose to run the scripts interactively or by pre-configuring the configuration files as will be further explained.
- **usr** folder:
  - **bin** folder: Contains the script file that gets executed on the start of the container. You can add your own custom code to be executed.
  - **config** folder: Contains the server configuration fragments (keystore, server properties, user registry) used by MobileFirst Server Foundation/MobileFirst Foundation Operational Analytics.
  - **env** folder: Contains the environment properties used for server initialization (server.env) and custom JVM options (jvm.options).
  - **jre-security** folder: You can update the JRE security related files (truststore, policy jars etc) by placing them in this folder.
  - **security** folder: The key store, trust store and the LTPA keys files (ltpa.keys) should be placed here.
  - **ssh** folder: The SSH public key file should be placed here to enable SSH access to the Container.
  - **wxs** folder (only for the MobileFirst Server Foundation): Contains the data cache / extreme scale client library when Data Cache is used as attribute store for the server.

## Prerequisites

1. Login to the IBM Bluemix environment.

This step is mandatory because you will be running IBM Containers commands during the following steps.

Run: `cf login`.

When prompted, enter the following information:

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

2. To run IBM Containers commands, you must first log in into the IBM Container Cloud Service.

Run: `cf ic login`.

3. Make sure that the `namespace` for container registry is set. The `namespace` is a unique name to identify your private repository on the Bluemix registry. The namespace is assigned once for an organization and cannot be changed.

Choose a namespace according to following rules:

- It can contain only lowercase letters, numbers, or underscores.
- 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: `cf ic namespace set <new_name>`.

To get the namespace that you have set, run the command: `cf ic namespace get`.

To learn more about IC commands, use the `ic help` command.

## Setting Up the MobileFirst and Analytics Servers on IBM Containers

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.

**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.

### MobileFirst Foundation Operational Analytics

If you intend to use analytics with your MobileFirst Server start here.

Using the configuration files

Running the scripts

### MobileFirst Foundation Server

Using the configuration files

Running the scripts

Launch the MobileFirst Console by loading the following URL: <http://MFCONTAINERHOST/mfpconsole> (it may take a few moments).

Add the remote server by following the instructions in the [Using MobileFirst CLI to Manage MobileFirst Artifacts](#) ([../using-the-mfpf-sdk/using-mobilefirst-cli-to-manage-mobilefirst-artifacts/#add-a-new-server-instance](#)) tutorial.

With the MobileFirst Server running on IBM Bluemix, You can now start your application development. Review the MobileFirst Foundation tutorials ([../all-tutorials](#)).