

Lab 12: Creating the MobileFirst Foundation Server instance using the MobileFirst tile in Bluemix

Lab Introduction

As you may recall, in our previous lab we deployed the mockServer into Bluemix, but we were unable to test the mobile app because the mockServer tried to connect to a local instance of MobileFirst server that didn't exist. In this lab, we will create the MobileFirst Server in Bluemix where the mockServer will be able to connect to it, and this will allow us to test the mobile app end-to-end.

Bluemix configuration

1. Using your Bluemix credentials, log into Bluemix using the URL below: <https://new-console.ng.bluemix.net/>
2. Once logged in, click on the **Catalog** option in the top right to show the list of services available in Bluemix. Once on the **Catalog** page click on the **Mobile** option on the left column to show all the mobile services available. Click on **Mobile Foundation** and you will land on the page shown below.

IBM Bluemix Console Catalog Docs 301 Jorge Gonzalez-Orozco jorgego@us.ibm.com | MFv8Tile

← View All

Mobile Foundation

The Mobile Foundation service is powered by the market-leading IBM MobileFirst Foundation Platform. This service radically simplifies building, managing, and updating your mobile app with a pre-integrated, comprehensive set of mobile application services. Build your app by using the front-end framework and tools of your choice. Easily add comprehensive security, app lifecycle management, and a back-end logic engine. Push and offline sync to your apps by using the cross-platform software developer kit (SDK).

Service name:

Mobile Foundation-wr

Features

- **Comprehensive Security**
Protect data on the device, APIs, and your brand from hackers and vulnerabilities. You don't need to be a security expert to get started right away with building secure apps. Employ best practices by using samples and templates, and quickly make and roll out changes with a configuration-based approach.
- **App Lifecycle Management**
Adopt a more streamlined and repeatable process for incremental app updates. For example, skip the app store approval by making changes to your app while it's live, with no recompile needed.

IBM

Note: You can leave the service name as it is.

3. Scroll down the page. For this lab we will use the **Developer** plan which is selected by default. Click on the **Create** button. Then on the next page note that we have an instance of the service created. In this page, if you click on the **Settings** tab you can change the Topology, Security and Server Configuration settings. **Do not change any settings for this lab.**
4. Click back on the **Overview** tab and click on **Start Basic Server**. The server deployment can take up to 10 minutes. After the process is complete, you will see MobileFirst Server dashboard as shown below

The screenshot displays the IBM Bluemix Mobile Foundation-PoTv8 interface. The top navigation bar includes the IBM Bluemix logo, a 'Mobile' dropdown, and links to 'Catalog' and 'Docs'. A box with the number '301' and a user profile for 'Jorge Gonzalez-Orozco' are also present. The main header shows 'Mobile Foundation-PoTv8' with a status of 'Service available'. Below this are tabs for 'Manage', 'Connections', 'Overview', and 'Settings'. The 'Overview' tab is active, showing a 'Status' of 'RUNNING', 'Server Size' of 'Small', 'Memory' of '1GB', and 'Server Route' of 'mobilefoundation-potv8-it-server.mybluemix.net'. There are buttons for 'Recreate', 'Change Size', 'Launch Console', and 'Add Analytics'.

5. Now click on the **Launch Console** button and login using the username and password provided under the button. With a few clicks, we have created and successfully deployed a MobileFirst Server now running in Bluemix.

Remote server configuration

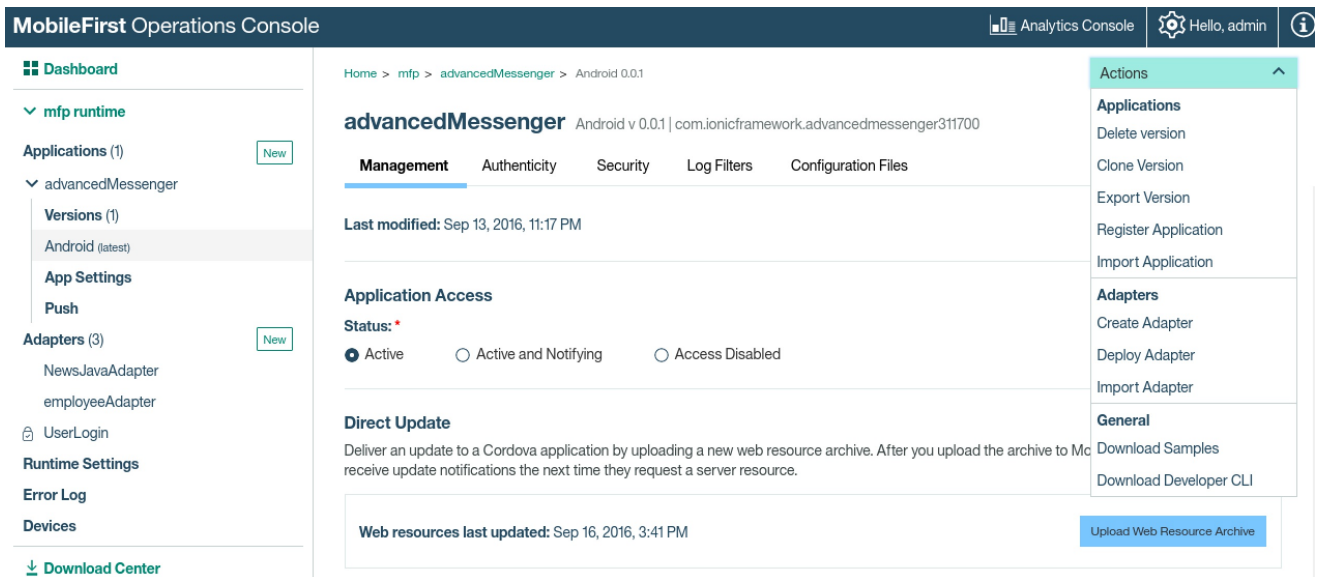
We now need to register this new remote server in our local development environment so that we can deploy mobile application components.

1. In your terminal window, type in the `mfpdev server add` command in the mobile app folder `~/dev/workspaces/am/advancedMessenger`. This will invoke the interactive mode of MFP CLI and allow you to key in your server parameters. **As you fill in these parameters, make sure you use the same (Bluemix) host name and admin password for the MobileFirst Server you created on Bluemix.**

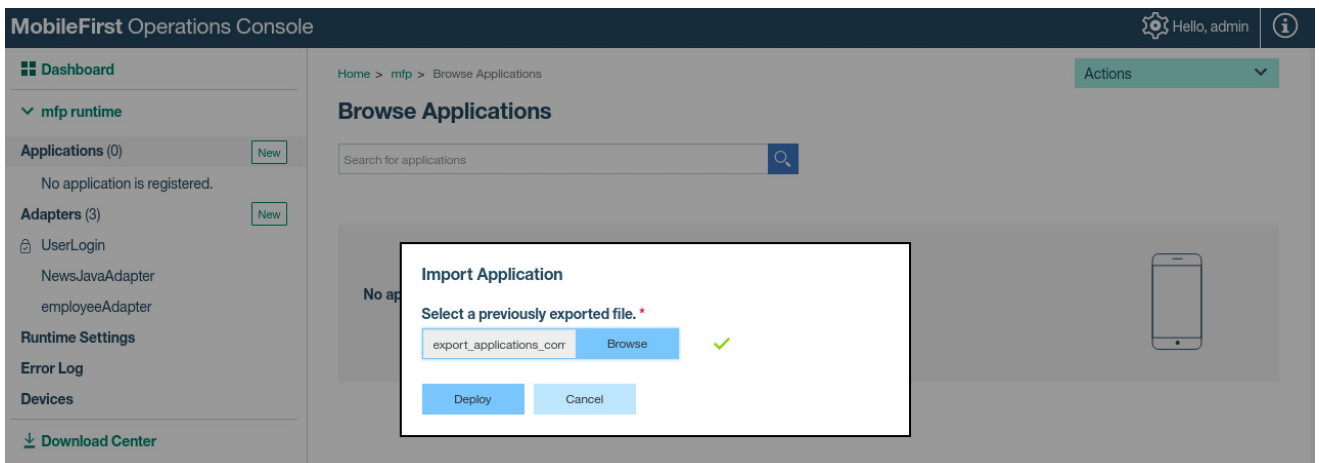
```
ibm@mfp-seminar-31:~/dev/workspaces/am/advancedMessengers
? Enter the name of the new server profile: bluemix
? Enter the fully qualified URL of this server: http://mfp-seminar-31.ibmcloud.com
? Enter the MobileFirst Server administrator login ID: admin
? Enter the MobileFirst Server administrator password: *
? Save the administrator password for this server?: Yes
? Enter the context root of the MobileFirst administrative console: /
? Enter the MobileFirst Server connection timeout in seconds: 30
? Make this server the default?: No
```

2. After this you should see the message **Server profile 'bluemix' added successfully**.
3. We will now deploy our three adapters to the new server in Bluemix. From a Terminal command window, run the following command from inside all three adapter folders:

You will now have all three adapters deployed.

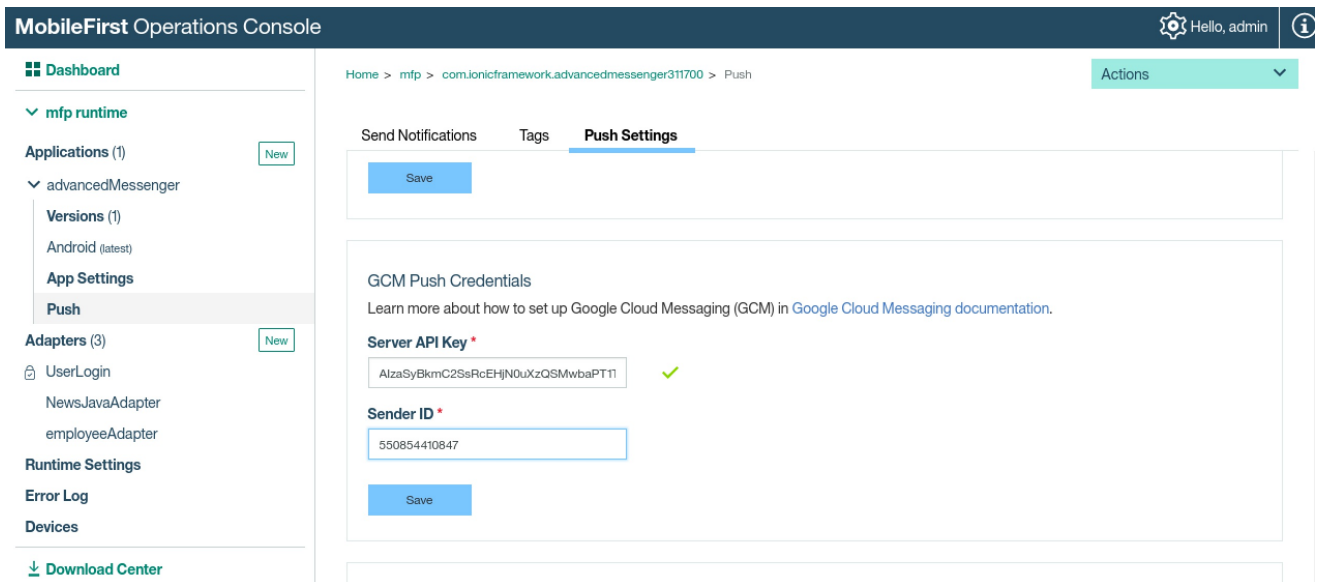


- Now open your remote MobileFirst Operations Console for your MobileFirst Server on Bluemix and select **Import Application** from the **Actions** menu of the top left as shown, browse to the file you download in the previous step and click OK.

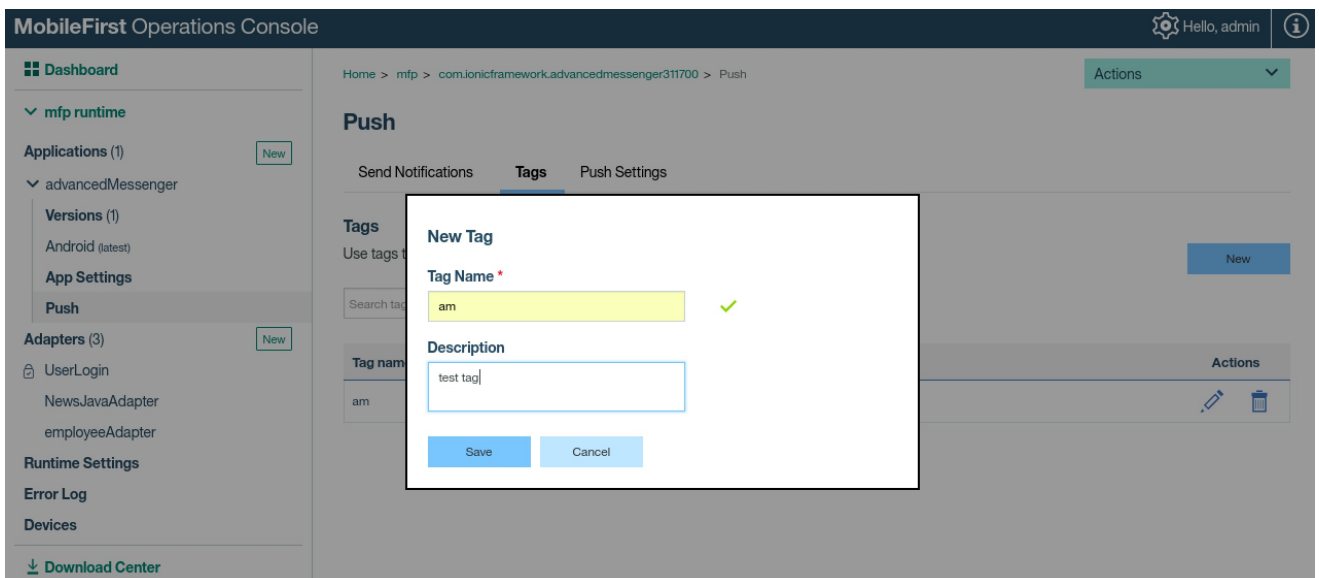


Click on the **Deploy** button.

- Now we need to add the Push credentials, as they are not copied in the export file. On your remote MobileFirst Operations Console, select **Push** under **Applications/advancedMessenger** and type in the **Server API Key** and **Sender ID** as shown below. Note that these values are the same used in the Push Lab. If you don't have them don't worry, as setting up Push is not required to complete this lab.



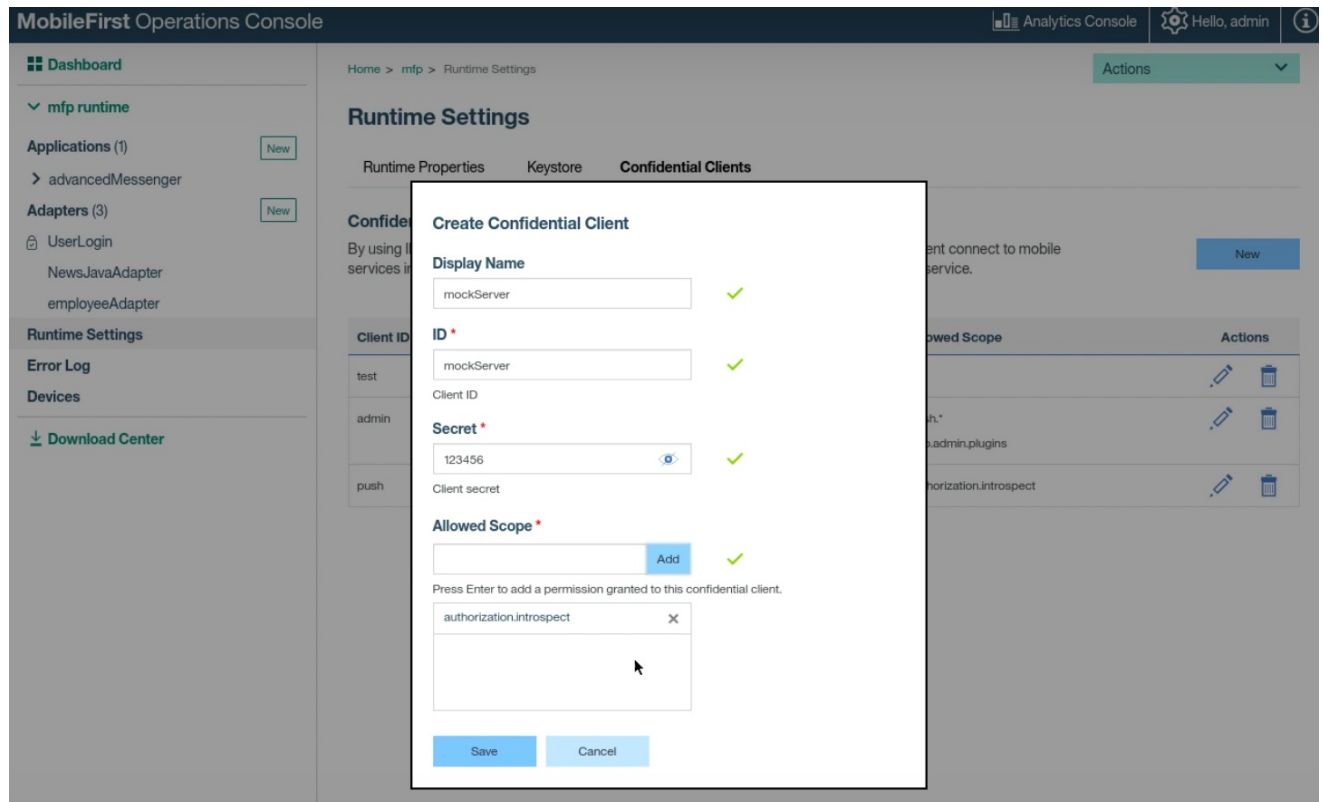
7. One more setup step for Push is to add the tags. Click on the **Tags** tab, click on **New** and add the tag name **am** and description **test tag** as shown below



8. Now we need to add again the confidential client settings in the same way we did in the local server in a previous lab. *As a recap, for the external resource server (the mock server) to be able to request a token for the authorization.introspect scope, it needs to be registered as a confidential client via the MobileFirst Operations Console.*

In the remote MobileFirst Operations Console, select **Runtime Settings** from the left menu. Select the **Confidential Clients** tab and click on the **New** button. Type **mockServer** for both **DisplayName**

and **ID**. For **Secret** type the password **123456**. For **Allowed Scope** type **authorization.introspect** and click on **Add**. Your screen should look like the picture below:



9. Click **Save** to save the configuration.

Updating the mockServer in Bluemix to use the MobileFirst remote server

Now we will update the mockServer in Bluemix to leverage our brand new MobileFirst server running in Bluemix.

1. Open the file `~/dev/workspaces/am/mockServer/server.js` in your text editor and update the localhost and port:

Before:

```
authServerUrl: 'http://localhost:9080/mfp/api',
```

After:

You can get the remote URL from the remote Console in your browser. Change the port to 80.

```
authServerUrl: 'http://<<console host name>>:80/mfp/api'
```

2. Since we will not use analytics on the remote server, remove the following code section:

```
analytics: {  
  onpremise: {  
    url: 'http://localhost:9080/analytics-service/rest/',  
    username: 'admin',  
    password: 'admin'  
  }  
}
```

3. Save and close the **server.js** file.
4. We will now push these mockServer changes to Bluemix. Open a command window and from the

```
~/dev/workspaces/am/mockServer
```

 run the following command.

Note that you will be prompted for the local Operating System password, which is QQqq1234 on the VMWare image. Also, we assume that you have the Bluemix and Cloud Foundry Command Line Interface tools already installed in your local environment as covered in the previous lab. If this is not the case, then refer to Lab 11 for instructions.

```
sudo cf push mockAdvancedMessenger
```

Updating the mobile app to use the

MobileFirst remote server

Let's run this command to make the mobile app use the remote server. From a terminal window, run the following command from the mobile app folder `~/dev/workspaces/am/advancedMessenger` :

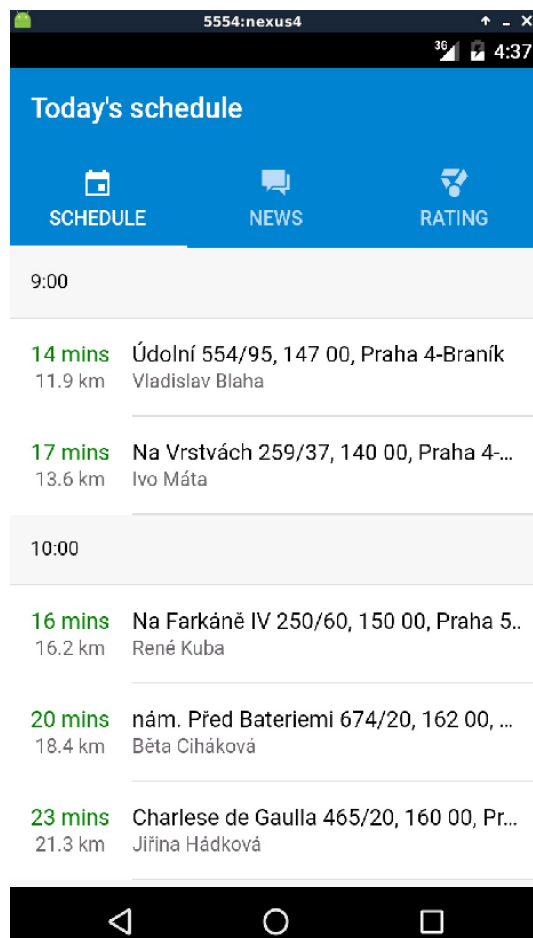
```
mfpdev app config server bluemix  
cordova prepare
```

Testing the mobile app

Now we are ready to test the mobile app. The mobile app now points to a remote MobileFirst Server and to a remote NodeJS mockServer, both running in Bluemix. From a terminal window, run the following command from the mobile app folder `~/dev/workspaces/am/advancedMessenger` in order to build and deploy the app to the Android emulator.

```
cordova prepare  
cordova run android
```

After a minute or so, the mobile app is deployed in the Android emulator. If you want to debug your app by using the Chrome browser inspector, type in this URL **chrome://inspect/#devices** and click on **inspect**. Your mobile app should be able to connect to the remote server and show schedule, news and rating data as shown below:



< LAB COMPLETE >