InterConnect 2016
The Premier Cloud & Mobile Conference

Lab 6678A

Mobile App Development Using the IBM MobileFirst Platform Foundation Command Line Interface (MFP-CLI)

Carlos Santana
Senior Software Architect
csantana@us.ibm.com
@csantanapr

February 2016

February 21 – 25
MGM Grand & Mandalay Bay
Las Vegas, Nevada

Notices and Disclaimers

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained h erein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM InterConnect 2016 @Dev Hello World Labs

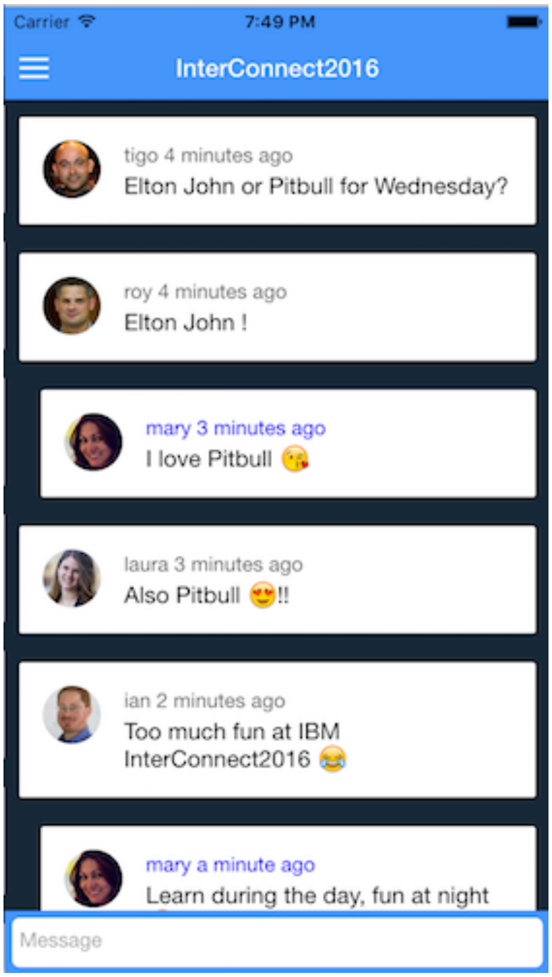
6678 - Mobile App Development Using the IBM MobileFirst Platform Foundation Command Line Interface (MFP-CLI)

Download

- Click ibm.biz/lab6678zip to download
- Click **Download ZIP** on github.com/csantanapr/IC2016-MFP-Hello-Lab

Your Goal!

- During this Lab you will build a Mobile Chat App that leverages the MFP Server and NodeJS using IBM Bluemix.



tigo 4 minutes ago
Elton John or Pitbull for Wednesday?

roy 4 minutes ago
Elton John !

mary 3 minutes ago
I love Pitbull 😘

laura 3 minutes ago
Also Pitbull 😍!!

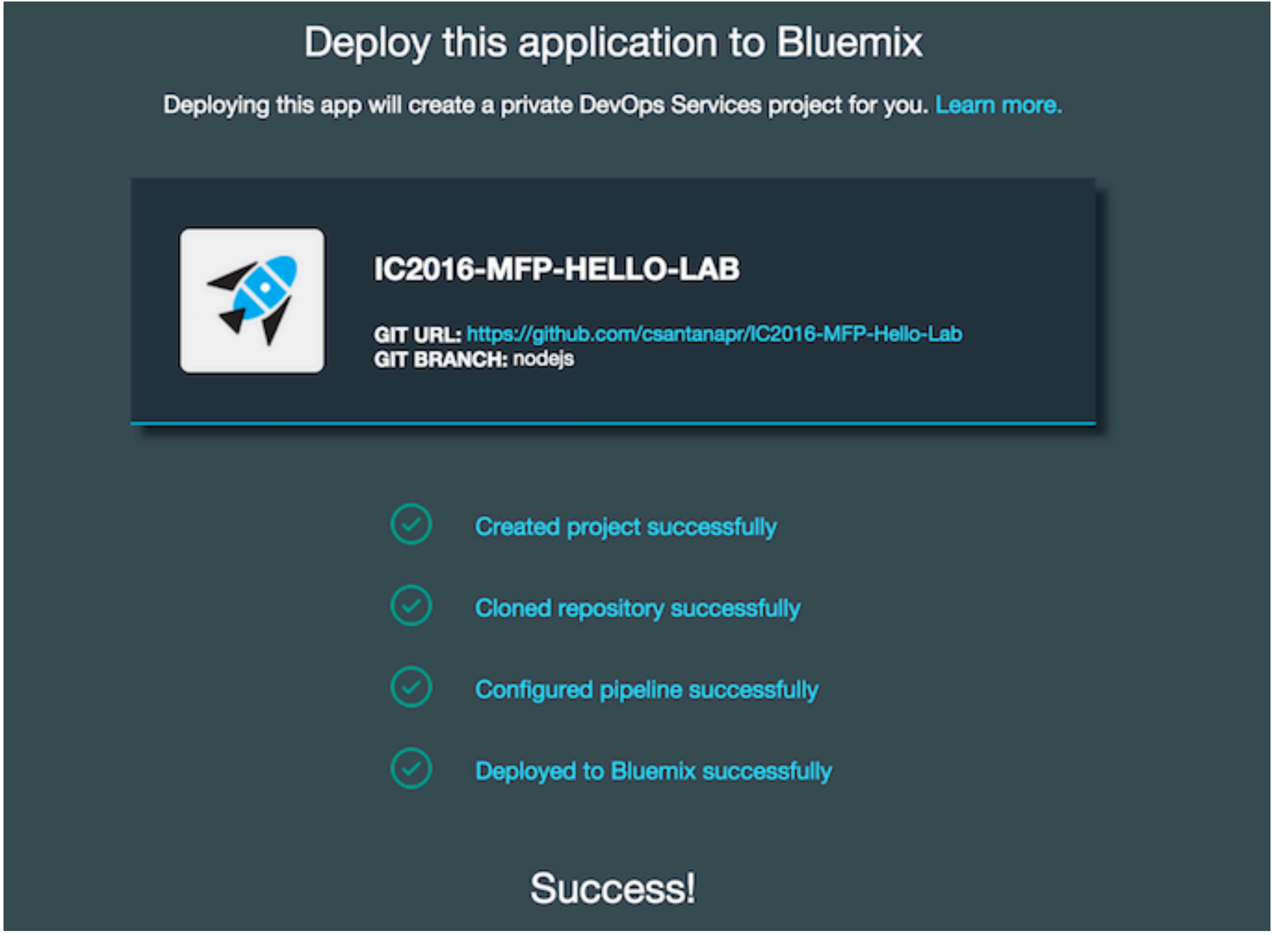
ian 2 minutes ago
Too much fun at IBM InterConnect2016 😂

mary a minute ago
Learn during the day, fun at night

1. Bluemix NodeJS Server

1.1 Deploy the NodeJS Server for the Chat Service

- Login into https://blueminx.net and verify that you have at least 128MB free for Cloud Foundry Apps
- Click this button Deploy to Bluemix or visit http://ibm.biz/lab6678
- This will deploy the **nodejs** branch
- Select a unique hostname for your new App, like:"**mychat**"
- Your app will be assigned a https unique URL like:**https://mychat.mybluemix.net**



Deploy this application to Bluemix

Deploying this app will create a private DevOps Services project for you. Learn more.

IC2016-MFP-HELLO-LAB

GIT URL: https://github.com/csantanapr/IC2016-MFP-Hello-Lab
GIT BRANCH: nodejs

✓ Created project successfully

✓ Cloned repository successfully

✓ Configured pipeline successfully

✓ Deployed to Bluemix successfully

Success!

1.2 Save the url for your new Chat Service

- Click to see the chat app running on the browser and verify the URL like https://mychat.mybluemix.net

- Leave the browser window open since we are going to need the URL and use the website for testing our mobile App.

1.3 MFP-CLI Installation

- Verify that your Lab workstation has the MFP-CLI installed by running the following commands in a Terminal window

```
$ mfp help
$ mfp -v
```

- If you don't have the MFP-CLI installed you can download the MFP-CLI from the MFP Developer Center

2. MFP Server

2.1 Start MFP Server

- Change directory to MFP backend project

```
$ cd MFPBackend
```

- Use the MFP-CLI to start the MFP Server

```
$ mfp start
```

- You can use other MFP-CLI commands to manage the MFP Server

```
$ mfp status
$ mfp stop
$ mfp restart
```

2.2 Configure the MFP Adapter

- Edit `MFPBackend/adapters/ChatService/ChatService-impl.js`

- Set the variable chatUrl to the url of your new Chat Service deployed in Step 1

```
var chatUrl = 'https://mychat.mybluemix.net';
```

2.3 Deploy Adapter changes

- Change directory to the adapter directory

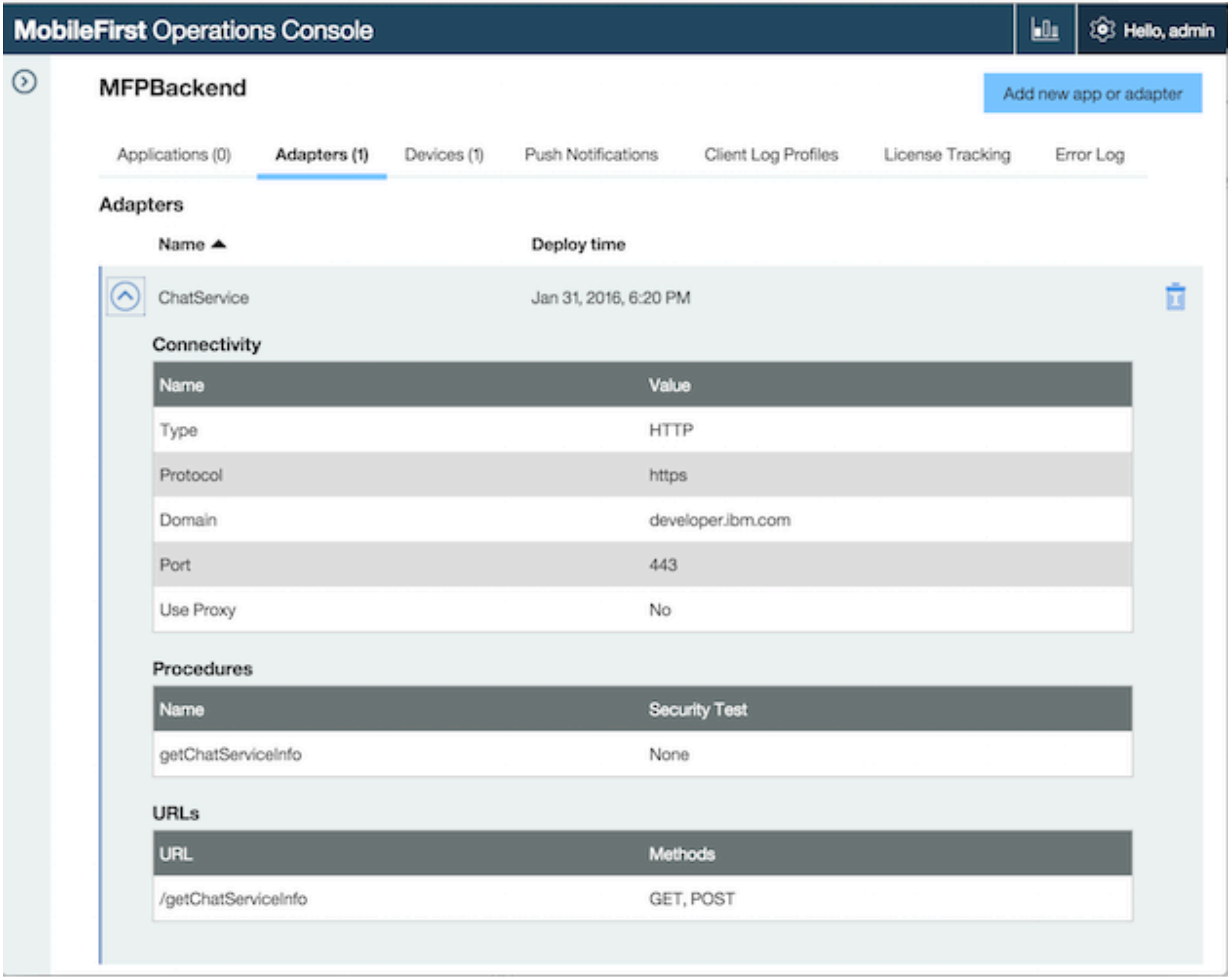
```
   $ cd adapters/ChatService/
```

- Use the MFP-CLI to push only the adapter changes

```
   $ mfp push
```

- Use the MFP-CLI to open the console and verify adapter is deployed

```
   $ mfp console
```



MFPBackend

Adapters

Name ▲	Deploy time
ChatService	Jan 31, 2016, 6:20 PM

Connectivity

Name	Value
Type	HTTP
Protocol	https
Domain	developer.ibm.com
Port	443
Use Proxy	No

Procedures

Name	Security Test
getChatServiceInfo	None

URLs

URL	Methods
/getChatServiceInfo	GET, POST

- Test the adapter and verify that returns the correct chat url

```
    $ mfp adapter call ChatService/getChatServiceInfo
```

- The command will return a response with the url of your nodejs app

```
Calling GET '/MFPBackend/adapters/ChatService/getChatServiceInfo?params=[]'

Response:     {
  "isSuccessful": true,
  "chatUrl": "https://chatservice.mybluemix.net"
}
```

3. MFP App

3.1 Setup App

- Change directory to the app directory

```
    $ cd ../../../MFPApp
```

- Use the MFP-CLI to add the Cordova Platform iOS

```
    $ mfp cordova platform add ios
```

3.2 Register App with MFP Server

- Use the MFP-CLI to push the app to the running MFP Server

```
    $ mfp push
```

- Use the MFP-CLI to verify that the app is registered on the correct server

```
    $ mfp server info local
```

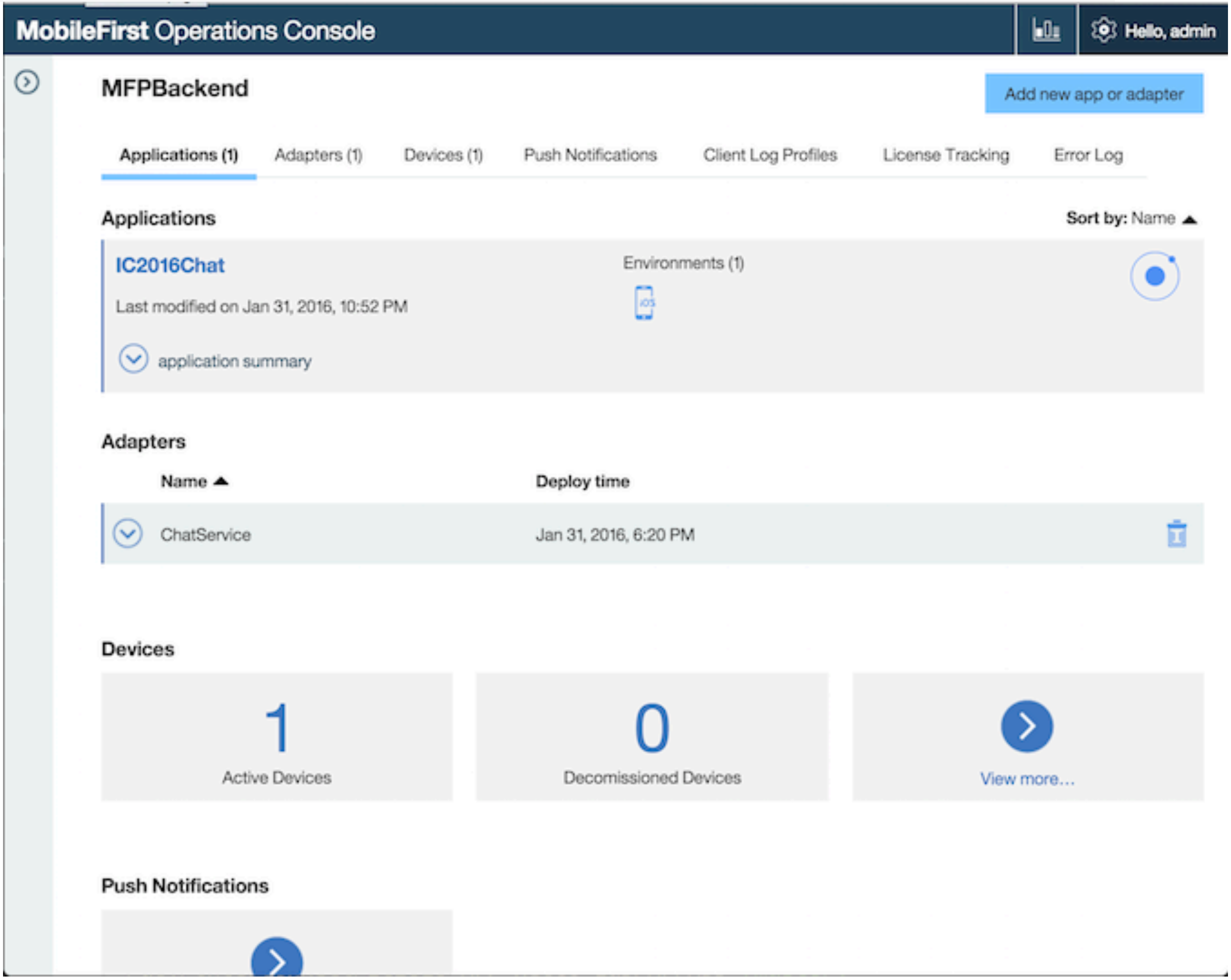
- The command will return a list of adapters and apps

```
Connecting to server 'local'
Profile name: local
URL: http://localhost:10080
MFP version: 7.1.0.00-20160125-0742
Is default: true
------------------------------------------------------------
Runtime: MFPBackend
Bound apps
  IC2016Chat, v1.0.0 - Hybrid (iphone)
```

```
Adapters
  ChatService
------------------------------------------------------------
```

- You can also open the Console and verify that the IC2016Caht app is registered

```
    $ mfp console
```



MobileFirst Operations Console

Hello, admin

MFPBackend

Add new app or adapter

Applications (1) Adapters (1) Devices (1) Push Notifications Client Log Profiles License Tracking Error Log

Applications

Sort by: Name ▲

IC2016Chat

Last modified on Jan 31, 2016, 10:52 PM

Environments (1)

application summary

Adapters

Name ▲	Deploy time
ChatService	Jan 31, 2016, 6:20 PM

Devices

1 Active Devices

0 Decomissioned Devices

View more...

Push Notifications

3.3 Running the App

- Preview the App on the Browser

```
    $ mfp cordova preview --type browser
```

- Run the App on the Simulator using the MFP-CLI

```
    $ mfp cordova emulate
```

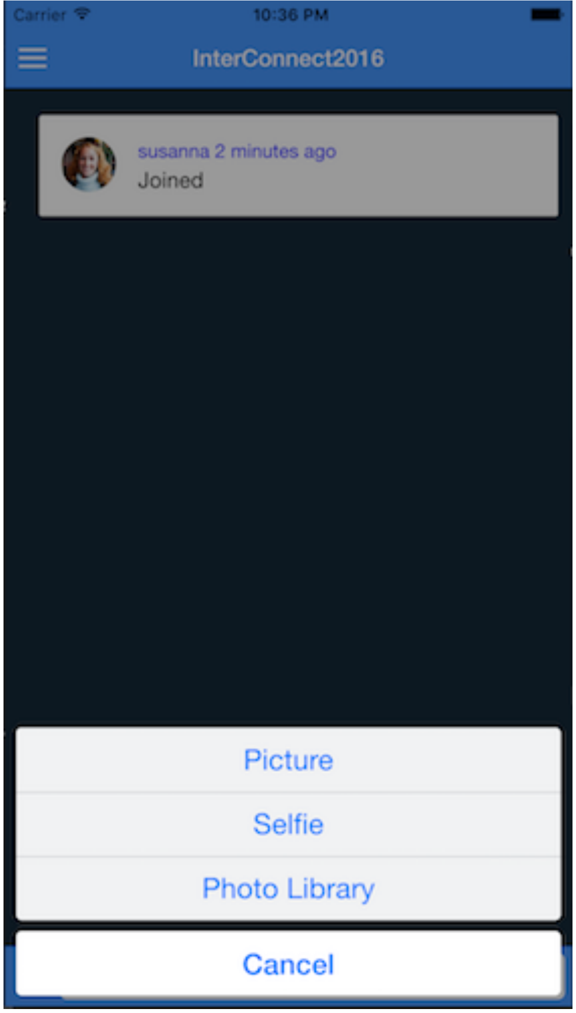
- When prompted select an iPhone Simulator from the list
- To run a specific Simulator you can pass `--target` argument

```
    $ mfp cordova emulate --target iPhone-6s-Plus
```

3.4 Test the App

- Visit the Chat Service on Bluemix (i.e. https://mychat.bluemix.net)
- Now test the chat service between the Browser and the iPhone Simulator
- If the Keyboard doesn't show up in the Simulator, then press Command (⌘) + K
- You can share the Bluemix URL to a friend located in any part of the world and have a conversation.
- This demo App is not configured with Login Authentication, this is something you can easily add using MobileFirst Platform check MFP Developer Center for more information on security related topics.

4. Extra Credit (Camera access)

- If you finish the Lab in less than 15 minutes you can go ahead and continue
- Enable the App to send photos using the camera or photo library
- 

susanna 2 minutes ago
Joined

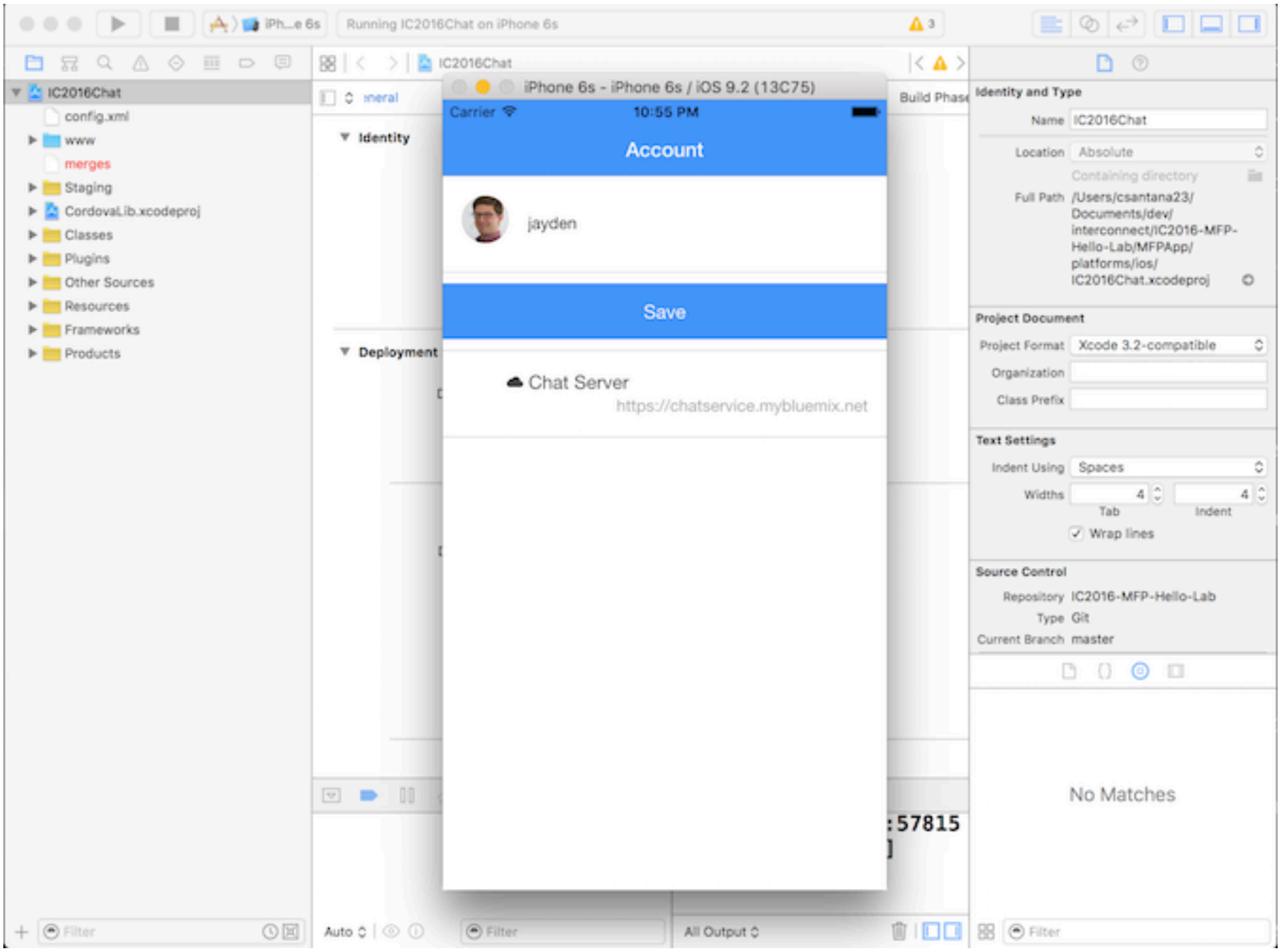
4.1 Open XCode project

- Open `MFPApp/platforms/ios/IC2016Chat.xcodeproj`

- You can also open from Terminal

```
  $ open platforms/ios/*.xcodeproj
```

- Click Play Button in XCode to run



iPhone 6s - iPhone 6s / iOS 9.2 (13C75)

Carrier 10:55 PM

Account

jayden

Save

Chat Server
https://chatservice.mybluemix.net

Identity and Type
Name IC2016Chat
Location Absolute
Containing directory
Full Path /Users/csantana23/Documents/dev/interconnect/IC2016-MFP-Hello-Lab/MFPApp/platforms/ios/IC2016Chat.xcodeproj

Project Document
Project Format Xcode 3.2-compatible
Organization
Class Prefix

Text Settings
Indent Using Spaces
Widths 4 Tab 4 Indent
☑ Wrap lines

Source Control
Repository IC2016-MFP-Hello-Lab
Type Git
Current Branch master

4.2 Add the Cordova Camera Plugin

- Change directory to MFPApp if not already

```
$ cd MFPApp
```

- Use the MFP-CLI to add the camera plugin

```
$ mfp cordova plugin add camera
```

4.3 Add option to share photo

- Edit `MFPApp/www/templates/chat.html`

- Add a camera icon to call Camera Plugin as the first child of the `<form id="footer-input">`

```
<i class="icon ion-camera" ng-click="takePicture()" ng-show="isWebView"></i>
```

4.4 Prepare App Changes

- Use the MFP-CLI to prepare your changes

```
$ mfp cordova prepare
```

- Click Play Button in XCode to run Simulator
- Touch the Camera button
- Select Photo Library (Real device supports Selfie and Picture buttons)
- Select a Photo to share
- Check the other clients and see the picture you just share

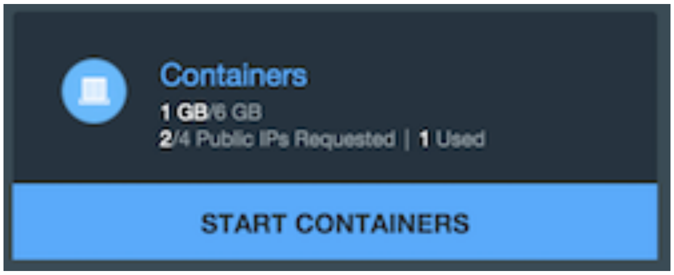
5 Homework (MFP Docker on Bluemix)

- If you want to be able to run the App on your iPhone and take it for a spin, you will need to deploy the MFP Server on a public IP Address that your iPhone can reach.
- The easiest way to run MFP Server on the Cloud is using Bluemix Containers.

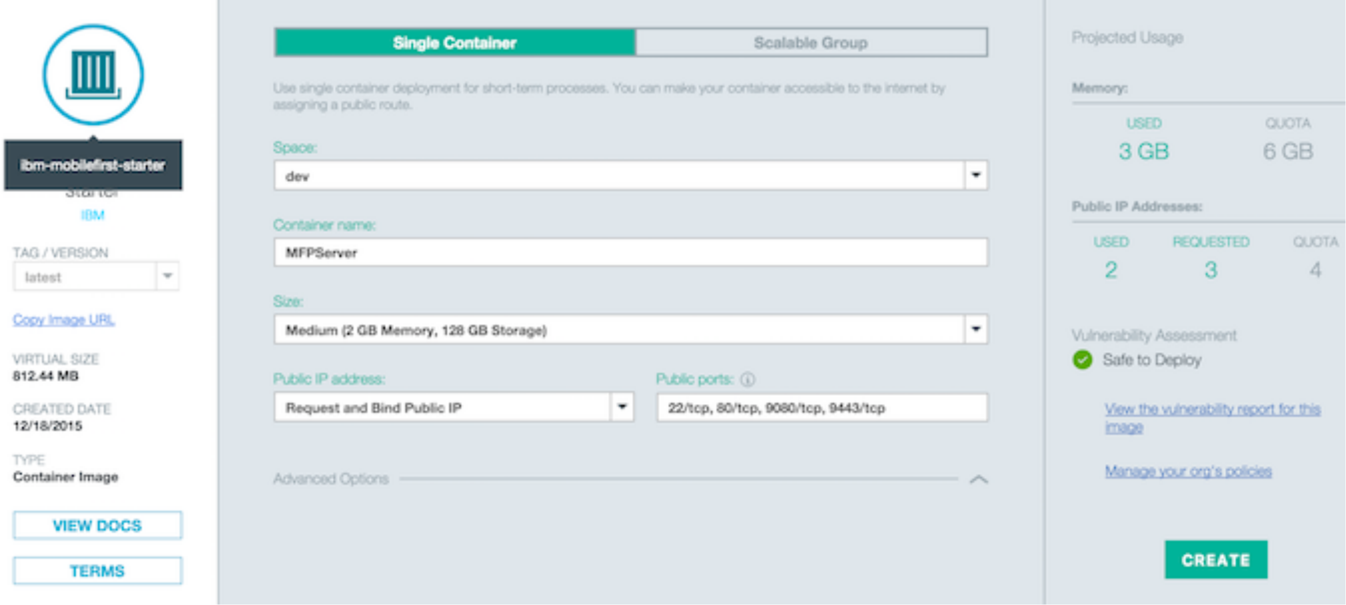
5.1 Run MFP Server Docker Image

- Login into Bluemix and make sure you have at least 1GB of memory and 1 Public IP Address for Containers.

- Click Containers on your Dashboard



- Select the ibm-mobilefirst-starter docker image



ibm-mobilefirst-starter

IBM

TAG / VERSION: latest

Copy Image URL

VIRTUAL SIZE: 812.44 MB

CREATED DATE: 12/18/2015

TYPE: Container Image

Use single container deployment for short-term processes. You can make your container accessible to the internet by assigning a public route.

Space: dev

Container name: MFPServer

Size: Medium (2 GB Memory, 128 GB Storage)

Public IP address: Request and Bind Public IP

Public ports: 22/tcp, 80/tcp, 9080/tcp, 9443/tcp

Projected Usage

Memory: USED 3 GB, QUOTA 6 GB

Public IP Addresses: USED 2, REQUESTED 3, QUOTA 4

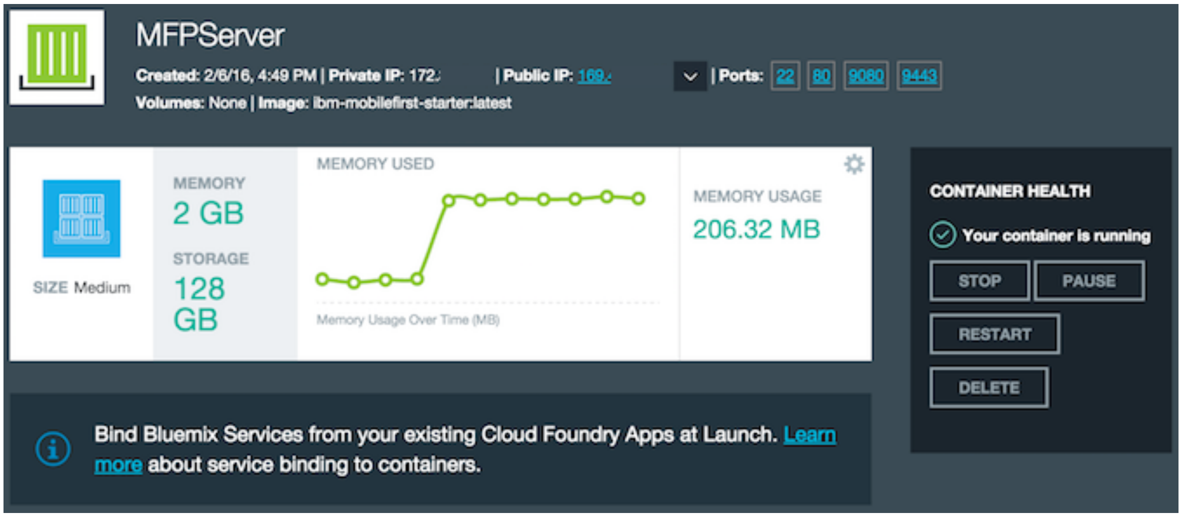
Vulnerability Assessment: Safe to Deploy

View the vulnerability report for this image

Manage your org's policies

- Enter the following minimum information:
  i. Container name: *MFPServer*
  ii. Size: *Medium 2GB*
  iii. Public IP Address: *Request and Bind IP*
- Click *Create*

- Wait for a public IP address to be assigned, and container to be running



MFPServer

Created: 2/6/16, 4:49 PM | Private IP: 172. | Public IP: 169. | Ports: 22 80 9080 9443

Volumes: None | Image: ibm-mobilefirst-starter:latest

SIZE Medium

MEMORY 2 GB

STORAGE 128 GB

MEMORY USAGE 206.32 MB

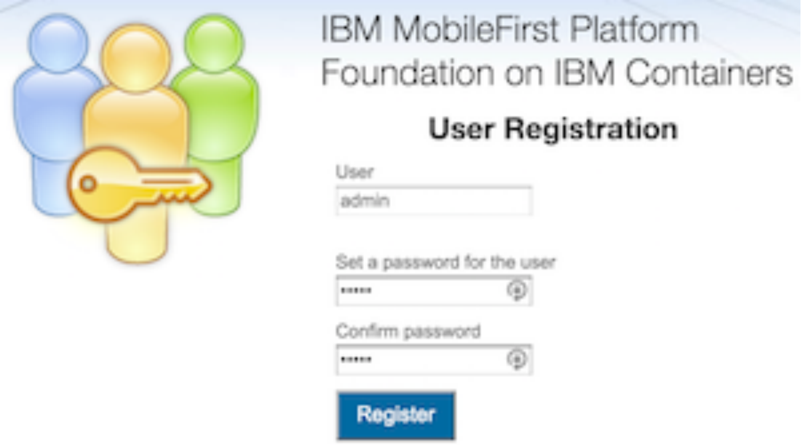
CONTAINER HEALTH

Your container is running

Bind Bluemix Services from your existing Cloud Foundry Apps at Launch. Learn more about service binding to containers.

- Click the public IP address to open a new Browser window

- Enter a password to register the user *admin* for example *admin* and click Register



IBM MobileFirst Platform Foundation on IBM Containers

User Registration

User
admin

Set a password for the user
••••••

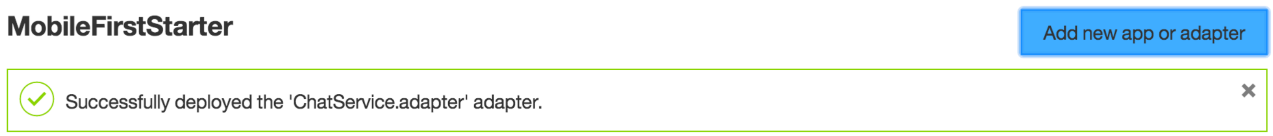
Confirm password
••••••

- Save and bookmark the base URL for the MFP Server (i.e. http://169.x.x.x:9080)
- Click *Open Console* to open the MFP Server Console

5.2 Upload Adapter

- In the MFP Server Console Click *Add new Adapter* button at the top right

- Browse to the location of ChatService.adapter for example *Downloads/IC2016-MFP-Hello-Lab-master/MFPBackend/bin/ChatService.adapter*



MobileFirstStarter

Successfully deployed the 'ChatService.adapter' adapter.

5.3 Add Remote MFP Server Profile

- Use the MFP-CLI to add a new server profile with the name *bluemix* located using the public IP address on bluemix, and the *admin* credentials you enter in Step 5.1

```
    $ mfp server add bluemix -l admin -p admin -u http://169.x.x.x:9080
```

Notice to replace 169.x.x.x with the correct public IP address of your docker container

- Use the MFP-CLI to verify the list of server profiles

```
    $ mfp server info
```

- The *bluemix* profile will show up

```
 Name     URL                       Description
------------------------------------------------------------
local    http://localhost:10080    Local Dev Server [Default]
bluemix  http://169.x.x.x:9080     Remote Server
------------------------------------------------------------
```

5.4 Register App with Remote MFP Server

- Change directory to MFPApp if not already

```
    $ cd MFPApp
```

- Use the MFP-CLI to register the App with the docker container on *bluemix*

```
    $ mfp push bluemix
```

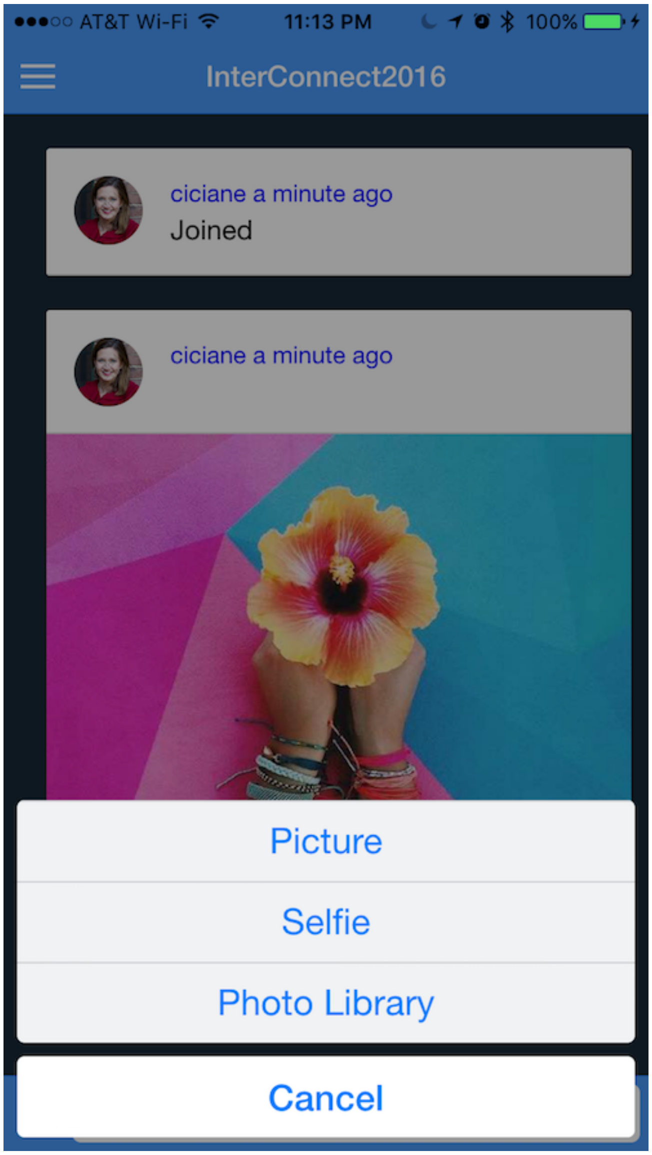
Notice that the name of the runtime on the remote MFP Server is *MobileFirstStarter*, This is different from the local runtime *MFPBackend* on the local MFP Server

5.5 Run the App on your iPhone

- Plug your iPhone with a USB cable
- Open XCode

```
$ open platforms/ios/*.xcodeproj
```

- In XCode change the target from Simulator to the iPhone device
- Click the Play button



ciciane a minute ago
Joined

ciciane a minute ago