IBM

InterConnect
2017

March 19–23
MGM Grand &
Mandalay Bay
Las Vegas, NV

IBM

# Hands-on Lab
# Session 9012
# Working with Containers in IBM Bluemix

## Pam Geiger, Bluemix Enablement

**IBM**

IBM

## Bluemix Container Service

The IBM Bluemix Container Service is used to run Docker containers in a hosted cloud environment on IBM Bluemix. Containers are virtual software objects that include all of the elements that an app needs to run. A container has the benefits of resource isolation and allocation, but is more portable than a virtual machine. In this lab, you deploy an application in Bluemix as a cloud foundry app, and then deploy the same application in a Container.

# Log in to Bluemix and Deploy a cloud foundry app

1. Login to the virtual machine as `bmxuser` and password of `passw0rd`. Open a terminal Window by selecting **Activities** > ⊞ from the panel and select **XTerm**.

   For the purpose of the lab exercises, you use the US South Bluemix region.

2. Log in to Bluemix with the API endpoint for US south:

```
cf login -a api.ng.bluemix.net
```

- If prompted, enter the following information:
    - For Email, enter your IBMId for Bluemix.
    - For Password, enter the password for the IBMId. Your Bluemix organizations and spaces are retrieved.
    - Enter the number that represents one of your Bluemix organizations.
    - Enter the number that represents one of your existing Bluemix spaces. If you are using a Bluemix trial ID, containers can only be used in one space, so if you have already deployed containers in a space, you must use that space for all containers.

2. Create a directory that is named cf-pi-*[yourinitials]*. This is where you save all the files that are required to build the Docker image and to run your Node.js app. You can use any name for the directory. If you use the name from the example, replace *[yourinitials]* with your initials, such as cf-pi-pfg. Enter *passw0rd* if prompted for bmxuser password.

```
sudo mkdir cf-pi-[yourinitials]
```

3. Copy your Node.js app code and all related files into the directory. For this lab exercise, you will use the the Personal Insights app from Bluemix. To download the Personal Insights app code from Bluemix, follow these steps.

- Open the Bluemix user interface. For the US region, open a browser window to: **console.ng.bluemix.net**.
- Login with your Bluemix credentials.
- Click **Catalog** to open the Bluemix catalog.
- Click **Boilerplates** to navigate to the provided Boilerplates.
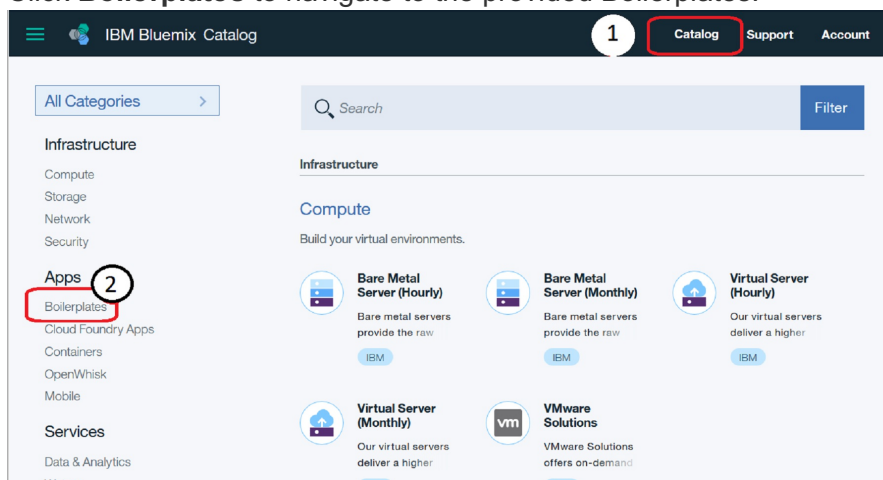


Boilerplates in Bluemix consist of a sample application and its associated runtime environment and

predefined services. You can use a boilerplate to quickly get up and running. For this lab, you are using the Node.js Personality insights boilerplate. This means that when you deploy the boilerplate, you are deploying the IBM SDK for Node.js, a Node.js app that makes use of the Watson Personality Insights service, and an Instance of the Personality Insights service. - Click Personality Insights Node.js Web Starter. This boilerplate includes a Node.js app that uses the IBM Watson Personality Insights service to extract a spectrum of cognitive and social characteristics from text that a person generates through blogs, tweets, or forum posts.
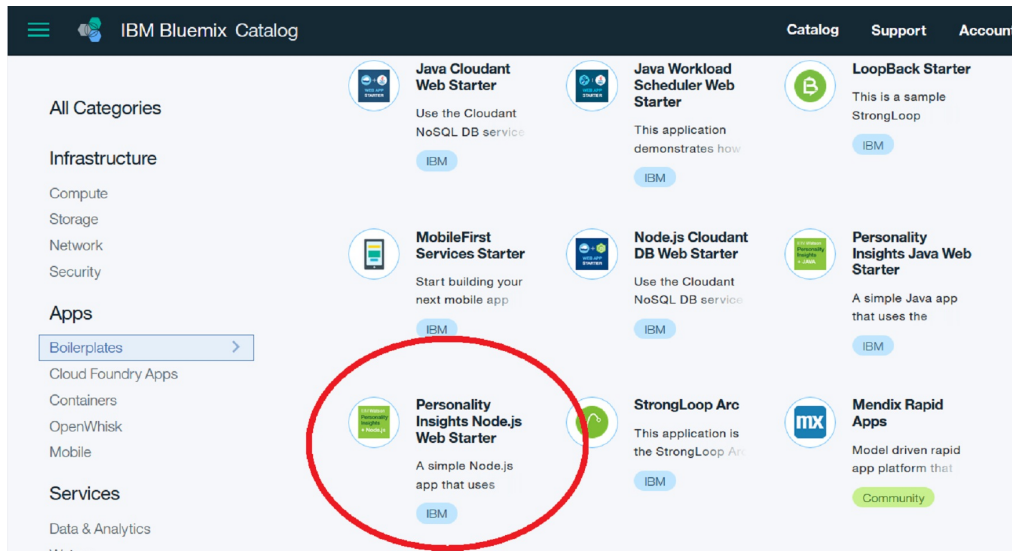


- Enter the app name cf-pi-*[yourinitials]* , replacing *[yourinitials]* with your own initials, such as *pfg* and click CREATE. The important thing is that the name must be unique in Bluemix.



This creates an app that is named cf-pi-*[yourinitials]* as well as an instance of the IBM Watson Personality Insights service that is named cf-pi-*[yourinitials]*-personality_insights.

As the app is deployed, instructions for deploying your app with the command line interface are displayed.

- Click DOWNLOAD STARTER CODE.

applications and service instances.

Before you begin, download and install the Cloud Foundry command line interface.

Download CF Command Line Interface ⬇

**Restriction:** The command line tool is not supported by Cygwin. Use the tool in a command line window other than the Cygwin command line window.

After you install the command line interface, you can get started:

① Download the code for your app to a new directory to set up your development environment.

DOWNLOAD STARTER CODE ⬇

- You will be prompted to open or Save, Click **Save File** and click OK.

- Copy the downloaded file from the Downloads directory to the directory that you created earlier that is named cf-pi-*[yourinitials]*. The downloaded file is named cf-pi-*[yourinitials]*.zip.

```
cd Downloads
cp cf-pi-[yourinitials.zip] ../cf-pi-[yourinitials]/
```

```
bmxuser@bluecompute:~$ cd Downloads/
bmxuser@bluecompute:~/Downloads$ ls
cf-pi-pfg.zip
bmxuser@bluecompute:~/Downloads$ sudo cp cf-pi-pfg.zip ../cf-pi-pfg/
```

Navigate to the directory that you created and extract the .zip file and save the app files to your directory, relacing *[yourinitials]* with your initials.

```
cd ../cf-pi-[yourinitials]
unzip cf-pi-[yourinitials]
```

The contents of the directory will look similar to the following example:

```
bmxuser@bluecompute:~/cf-pi-pfg$ ls
access.log      config          LICENSE       public
app.js          CONTRIBUTING.md manifest.yml  README.md
cf-pi-pfg.zip   i18n            package.json  views
```

Your Cloud Foundry app code is ready to be containerized.

# Create a Docker image with your CF app code
## About Images

**IBM provided images** IBM provides some public images as part of the Bluemix Container Service, such as the IBM Liberty and IBM Node images to test the features of IBM Bluemix Container Service. You can use these images as a parent image, modify the Dockerfile and build your own image with your own app code on it.
**Images from Docker Hub** Docker Hub is Dockers cloud-based image registry service. You can

copy images directly from Docker Hub into your private Bluemix registry or pull an image from Docker Hub, modify it locally, and then build it directly in your registry.

**Create your own image** If you have container images that you already use in your local Docker environment, you can push them to your private Bluemix registry to use them in IBM Bluemix Container Service. You can also create your own Dockerfile, build, test it locally, and then push it to your private images registry.

A Dockerfile is the file that Docker uses to build an image. The Dockerfile is essentially the build instructions to build the image. You create a Dockerfile that includes your Node.js app code and the necessary configurations for your container, and then build a Docker image from that Dockerfile. After you build the image you push it to your private Bluemix registry.

You can create the Dockerfile by using your preferred CLI editor or a text editor on your computer. The Dockerfile is provided for you here.

1.Change directories to the cf-ic- directory that you just created. Enter the following command to download the Dockerfile:

```
git clone https://github.com/bluemix-enablement/ic179012.git
```

A directory is created that is callled ic179012

2. Copy the Dockerfile to the directory that you created earelierby entering the following command:

```
cp ic179012/Dockerfile .
```

**IBM**

3. Enter the following command to view the Dockerfile:

```
cat Dockerfile
```

```
#Use the IBM Node image as a base image
FROM registry.ng.bluemix.net/ibmnode:latest
#Expose the port for your Personal Insights app, and set
#it as an environment variable as expected by cf apps
ENV PORT=3000
EXPOSE 3000
ENV NODE_ENV production
#Copy all app files from the current directory into the app
#directory in your container. Set the app directory
#as the working directory
ADD . /app
WORKDIR /app
#Install any necessary requirements from package.json
RUN npm install
#Sleep before the app starts. This command ensures that the
#IBM Containers networking is finished before the app starts.
CMD (sleep 60; npm start)
#Start the Personal Insight app.
CMD ["node", "/app/app.js"]
~
~
```

4. Get your namespace.

```
cf ic namespace get
```

```
bmxuser@bluecompute:~/cf-pi-pfg$ cf ic namespace get
pgeiger
```

You need to use this to tell the build command where to push the image that you create in the next step. In this example, the namespace is pgeiger. Replace the namespace in the build command with your namespace information.

If this is the first time that a user logs into your organization, there will be no namespace and you will need to set the namespace. A namespace is a unique name to identify your private Bluemix images registry. When you create a container, you must specify an image's location by including the namespace with the image name. Important: The namespace is assigned one time for an organization and cannot be changed after it is created. If you need to set the namespace, enter the following command, replacing *[yournamespace]* with a namespace of your choice.

```
cf ic namespace set [yournamespace]
```

5. Initialize the Bluemix Containers CLI

```
cf ic init
```

The result is similar to the following:

```
OK
The client certificates were retrieved.

Checking local Docker configuration...
OK

Authenticating with the IBM Containers registry host registry.ng.bluemix.net...
OK
You are authenticated with the IBM Containers registry.
Your organization's private Bluemix registry: registry.ng.bluemix.net/pgeiger

You can choose from two ways to use the Docker CLI with IBM Containers:


Option 1: This option allows you to use 'cf ic' for managing containers on IBM C
ontainers while still using the Docker CLI directly to manage your local Docker
host.
        Use this Cloud Foundry IBM Containers plug-in without affecting the loca
l Docker environment:



        Example Usage:
        cf ic ps
        cf ic images

Option 2: Use the Docker CLI directly. In this shell, override the local Docker
environment by setting these variables to connect to IBM Containers. Copy and pa
ste the following commands:
        Note: Only some Docker commands are supported with this option. Run cf i
c help to see which commands are supported.
        export DOCKER_HOST=tcp://containers-api.ng.bluemix.net:8443
        export DOCKER_CERT_PATH=/home/bmxuser/.ice/certs/containers-api.ng.bluem
ix.net/0c528ff4-1547-450e-910c-f8cef8b94337
        export DOCKER_TLS_VERIFY=1

        Example Usage:
        docker ps
        docker images

bmxuser@bluecompute:~/cf-pi-pfg$
```

6. Build the Docker image with your app code and push it to your private Bluemix registry.

```
cf ic build -t registry.ng.bluemix.net/[yournamespace]/cf-pi-[yourinitials] .
```

**Understanding this command's components**

- build: The build command.
- registry.ng.bluemix.net/*[yournamespace]*/cf-pi-*[yourinitials]*: Your private Bluemix registry path, which includes your unique namespace and the name of the image. For this example, the same name is used for the image as the CF node app cf-pi-[yourinitials], but you can choose any name for the image in your private registry.
- . The location of the Dockerfile. If you are running the build command from the directory that includes the Dockerfile, enter a period (.). Otherwise, use a relative path to the Dockerfile.

You will see the steps of building the image and the image is created in your registry. You can run the cf ic images command to verify that the image was created.

```
cf ic images
```

```
registry.ng.bluemix.net/pgeiger/microservices-refapp-zuul
        latest              430ff00f2319        4 months ago        313.3 MB
registry.ng.bluemix.net/pgeiger/mysql
        latest              994b6beb6b20        4 months ago        164.4 MB
registry.ng.bluemix.net/pgeiger/cf-pi-pfg
        latest              0b55cf5a03af        6 minutes ago       257.7 MB
```

# Deploy a container from your image

Before you create a container with IBM Bluemix Container Service, decide on the type of container that you need. IBM Bluemix Container Service allows you to create a single container, or a container group. The approach that you choose depends on the requirements and dependencies of the app that runs in your container.

## Single container

A single container in IBM Bluemix Container Service is similar to a container that you create in a local Docker environment. Single containers are a good way to start with IBM Bluemix Container Service and to learn about how containers work in the IBM cloud and the features that IBM Bluemix Container Service provides. You can also use single containers to run simple app tests or during the development process of an app. Deployment speed considerations for single containers:

The size of the image has a significant impact. The smaller the image, the quicker the deployment. After the first few times an image is deployed, deployment speeds improve. Initially, the image must be downloaded to the registry on the host. Subsequent deployments are faster. The networking setup might take a few minutes. A single container deploys faster than a container group because of the routing setup for groups. Deployments with linked containers might not be as quick as other deployments because of the connections that must be made.

## Container group

A container group consists of multiple single containers that are all created from the same container image and configured in the same way. The number of instances that your app requires depends on the expected workload and the time in which a request can be processed by the app. For example, multi-tasking or network connectivity affects how many instances might be required for your app.

See the Bluemix Container Service documentation for more details on how to determine how many instances your app will require.

For this lab exercise, you deploy your app in a container group.

1. Create a container group from the image that you created in the previous step by running the following command, replacing *[yourinitials]* with your own initials, and *[namespace]* with the name of your namespace.

```
cf ic group create --name cf-pi-[yourinitials] -m 64 -n pi-app-[yourinitials]
-d mybluemix.net -p 3000 -e CCS_BIND_SRV=cf-pi-[yourinitials]-personality_insights
--desired 3 --auto --anti registry.ng.bluemix.net/[namespace]/cf-pi-
[yourinitials]:latest
```

**About this command**

- group create: The group create command.

- --name cf-pi-*[yourinitials]* : The name of the container as you want it to display from the Bluemix GUI.

- -m 64: Each instance in the group is assigned 64 MB of memory.

- -n pi-app-*[yourinitials]* : The host name in the route URL.
  **Tip:** If you are using the starter code, you might want to use a different name for the host name than what you used in the previous lesson to deploy the Cloud Foundry app. If the same host is used and the CF app is still running, load balancing occurs between the container group and app. In this case, you will just delete the Cloud Foundry app after your container is running.

- -d mybluemix.net The domain name in the route URL. Unless you have a custom domain, use mybluemix.net.

- -p 3000: The port number to expose. Connections to the route URL are forwarded to port 3000 on the container.

- -e CCS_BIND_SRV=cf-pi-*[yourinitials]*-personality_insights: Bind the IBM Watson Personality Insights service instance to the container, so you can analyze cognitive and social characteristics from the text that you enter in the Personal Insights app. The service instance was created as part of the CF app deployment. If you are unsure what the name of your service instance is, run the following command.

```
cf services
```

- --desired 3: The number of single container instances to create in the container group. All of the instances are the same.

- --auto: Enable automatic recovery for the container group. If one of the instances in the group becomes unavailable, another instance is automatically created that replaces the unresponsive one.

- --anti: Enable anti-affinity for the container group. Each instance of the group is placed on a separate physical compute node, which reduces the odds of all containers in a group crashing due to a hardware failure.

- registry.ng.bluemix.net/*[namespace]*/cf-pi-*[yourinitials]*:latest: The destination repository path, which includes your unique namespace and the name of the destination image. For this example, the same name is used for the image as the CF node app cf-pi-*[yourinitials]*.

```
bmxuser@bluecompute:~/cf-pi-pfg$ cf ic group create --name cf-pi-pfg -m 64 -n pi
-app-pfg -d mybluemix.net -p 3000 -e CCS_BIND_SRV=cf-pi-pfg-personality_insights
 --desired 3 --auto --anti registry.ng.bluemix.net/pgeiger/cf-pi-pfg:latest
OK
The container group creation was requested.
The container group "cf-pi-pfg" (id: 241a4ca4-d549-4427-9c27-1d4a64d201c4) was c
reated.
Minimum container instances: 1
Maximum container instances: 3
Desired container instances: 3
```
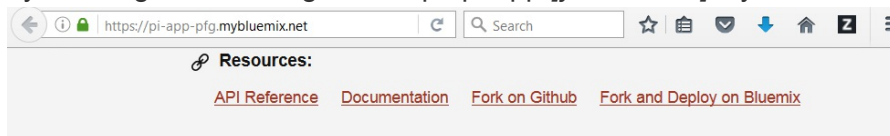
2.  Remove the CF app. Now that your CF app code is deployed in a container, you can remove the CF app to maximize the use of your quota.

```
cf delete cf-pi-[yourinitials]
Confirm the deletion by entering y.
```

3.  Wait a few minutes for the container to deploy and for the networking to be set up. When the container is deployed, you can view your Personal Insights app in a browser by entering the following URL http://pi-app-[yourinitials].mybluemix.net/.



Use the text in the sample for analysis or enter your own to see how the Personality Insights service works.

This completes the lab.