

Build 4 Micro Services, deploy them as Docker containers using IBM Cloud.

Microservices consists of an architecture where an application is divided in an n number of standalone modules. The failure of one module does not put the whole app down. Instead, only that specific service is down.

After a quick presentation of this architecture, you'll put in practice what you have learnt. In this workshop, we will build 3 microservices:

- 1- Login/Logout/Register
- 2- Pay a bill
- 3- Administrator billing system
- 4- Customer Support using IBM chatbot

One of the main advantages of this architecture is if a microservice goes down, it does not take down the whole application but rather only that specific micro service. Let's say we stop the deployment of Customer Support, it will be the only service not available. Customers visiting the website will still be able to login and pay their bills.

Also, we will build microservices in different languages for instance in our workshop both the login and bill pay services will be in NODEJS while the chatbot will be in PYTHON.

Learning objectives

In this tutorial, you'll learn how to build a web application using a Micro Services architecture, build docker images, as well as deploying docker images in IBM Cloud.

Prerequisites

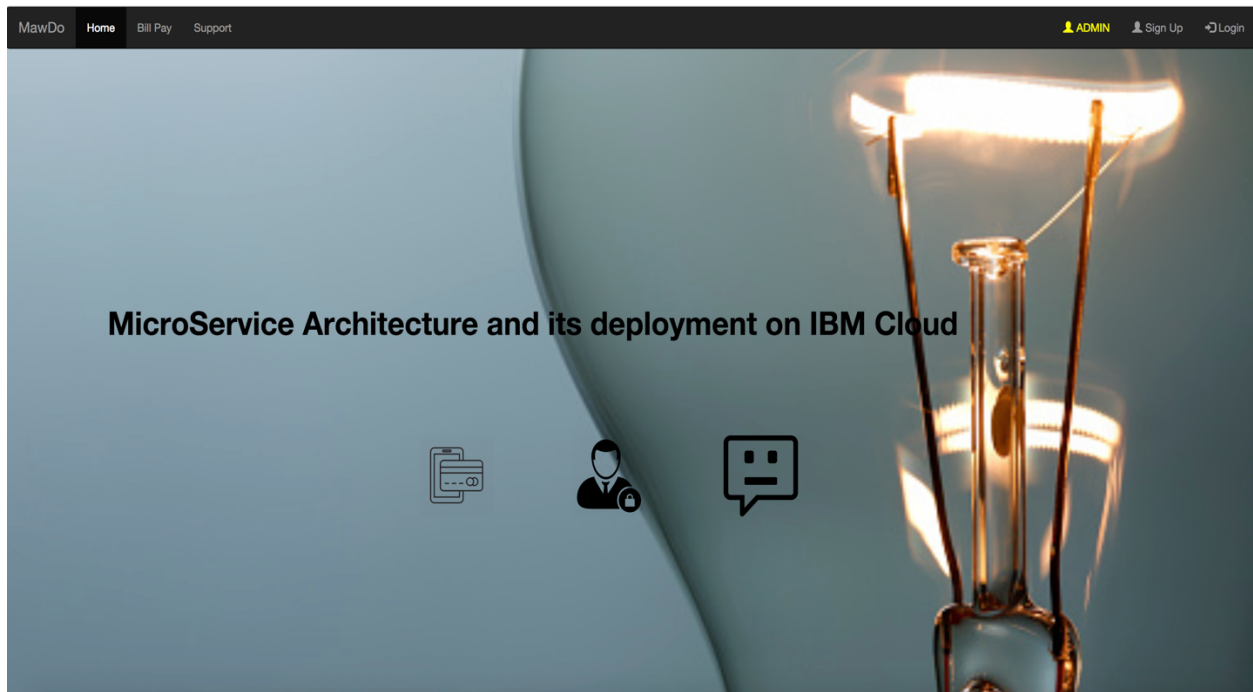
To use this tutorial, you'll need:

- A laptop
- An IBM Cloud account
- IBM Cloud CLI
- Docker & Docker account

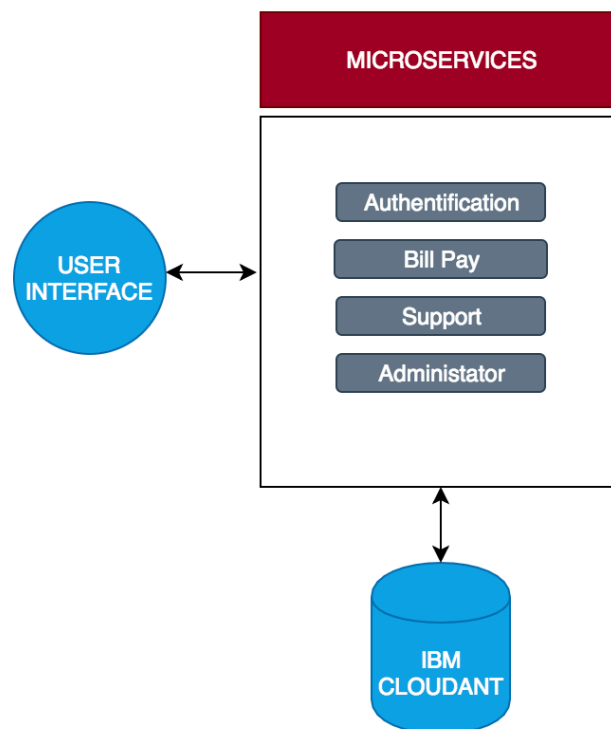
Estimated time

It should take you approximately 40 minutes to complete this tutorial.

Overview

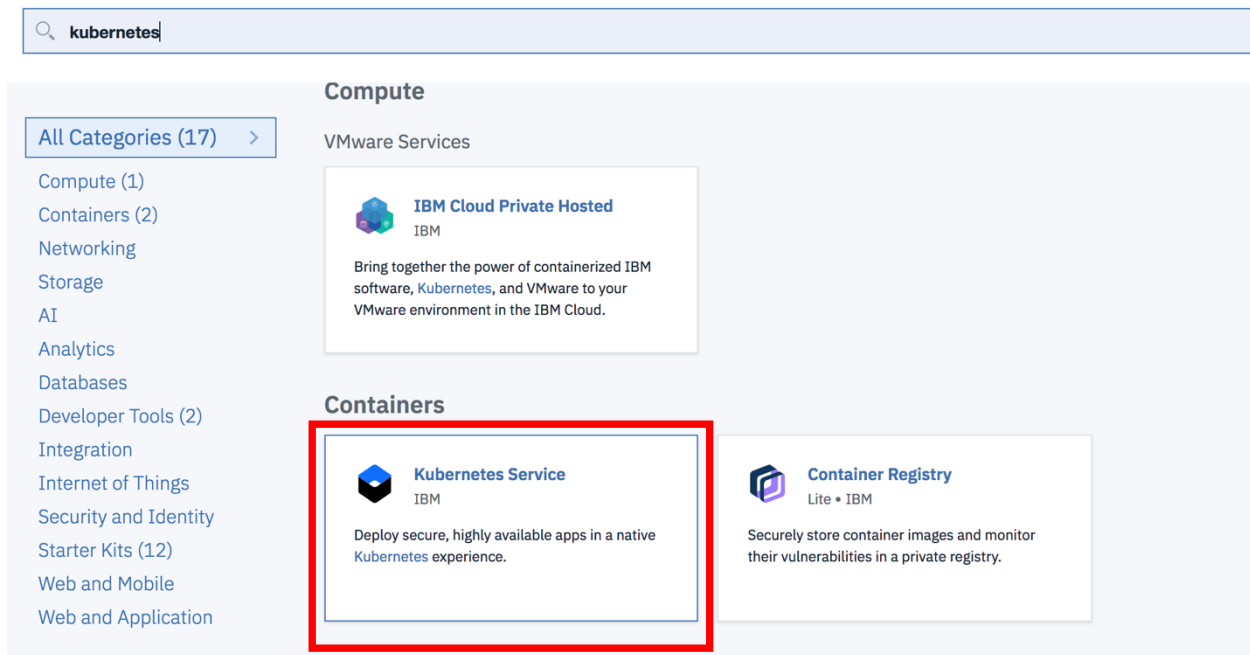


The following is the architecture of this web application.

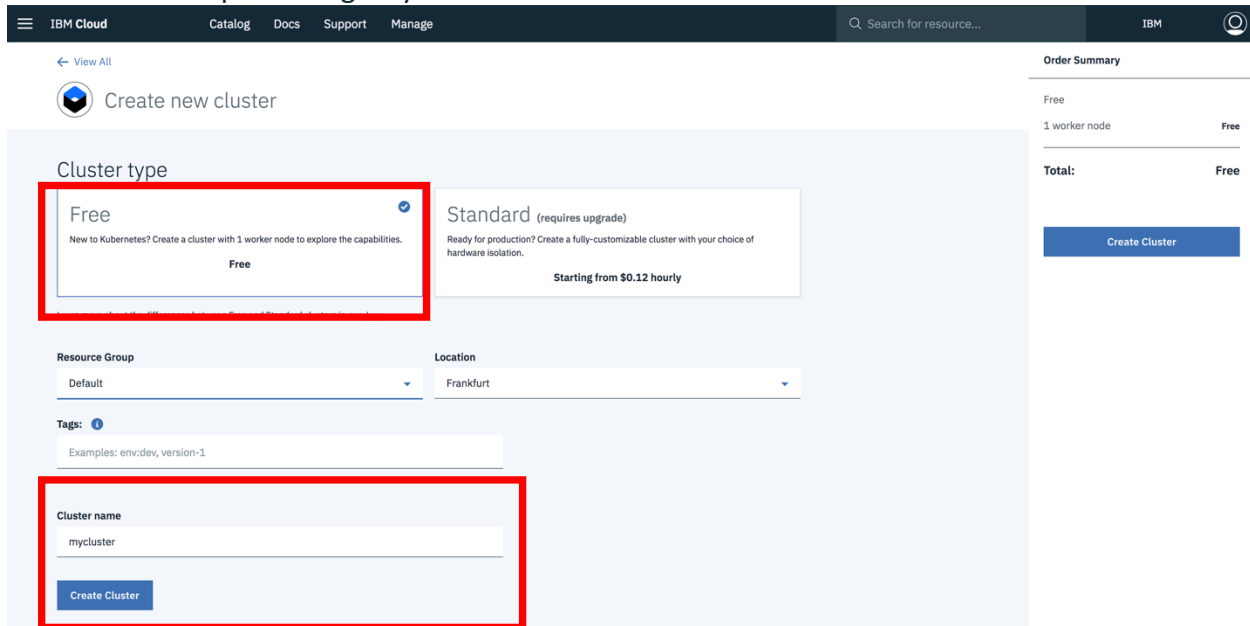


The first step is to set up our environment by creating a Kubernetes cluster. To do that, go to your IBM cloud console and look in the catalog for Kubernetes Service.

Catalog



Choose the free plan and give your cluster a name.



Start by cloning this repository to a folder in your local machine. You can do so by using the following command:

```
git clone https://github.com/yberrada/3MicroServicesDEMO.git
```

```
yberrada@youness-mbp:~/Desktop/WorkshopLaFactory/test4$ git clone https://github.com/yberrada/3MicroServices-DEMO-PYTHON-NODEJS-JAVA.git
Cloning into '3MicroServices-DEMO-PYTHON-NODEJS-JAVA'...
Username for 'https://github.com':
```

Enter your information as prompted by the system.

If you do not have a GitHub account, you can download the repository directly from this link:

<https://github.com/yberrada/3MicroServicesDEMO>

Once the page opens up download the repo by clicking on download zip:

yberrada / 3MicroServices-DEMO-PYTHON-NODEJS-JAVA Private

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

10 commits 1 branch 0 releases

Branch: master New pull request

Create new file Upload files Find file Clone or download

Clone with HTTPS Use SSH

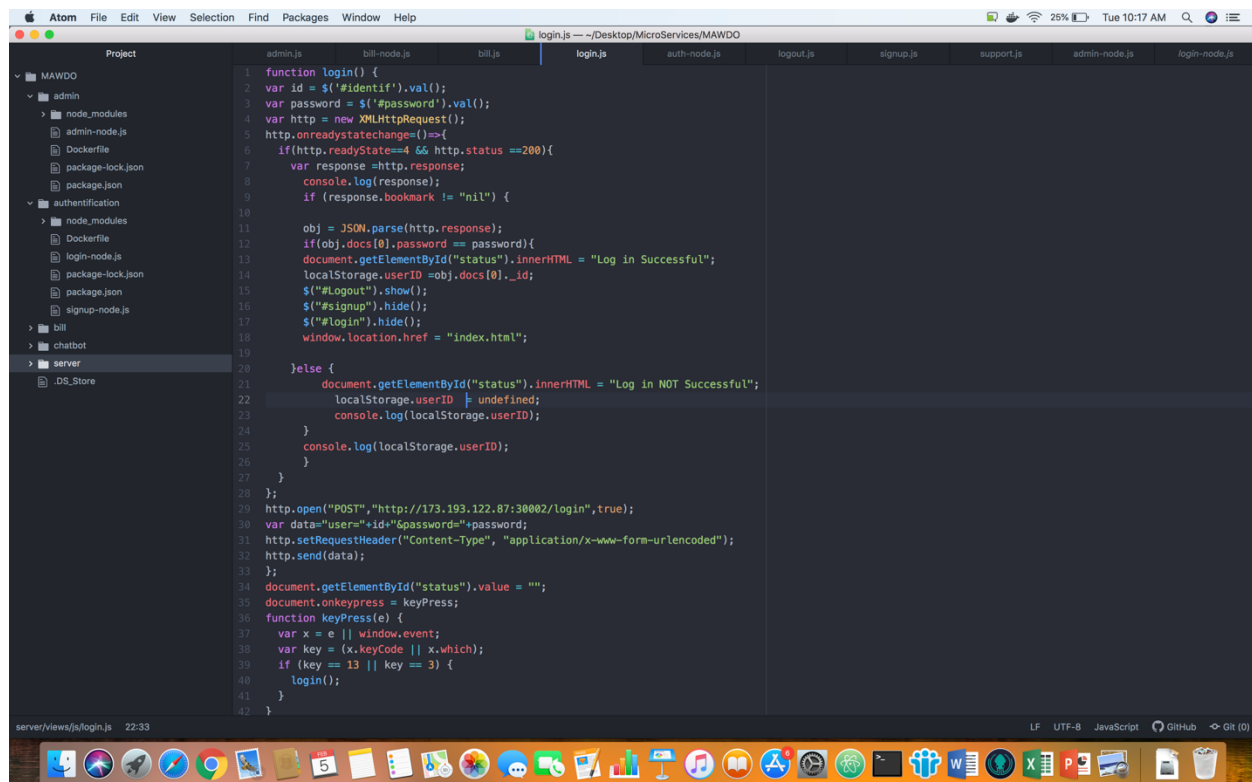
Use Git or checkout with SVN using the web URL.

https://github.com/yberrada/3MicroSer

Open in Desktop Download ZIP

File	Commit Message	Time
admin	App ready to be deployed on IBM CLOUD!! Create docker image, p	
authentication	App ready to be deployed on IBM CLOUD!! Create docker image, p	
bill	App done!! all three microservices are working	
chatbot	App ready to be deployed on IBM CLOUD!! Create docker image, p	
css	App Done . 3 microservices working .	12 days ago
images	2 Microservices working. Chatbot + Bill Pay	13 days ago
js	App Done . 3 microservices working .	12 days ago
server	Done	16 hours ago

Then, open folder on your favorite text editor.



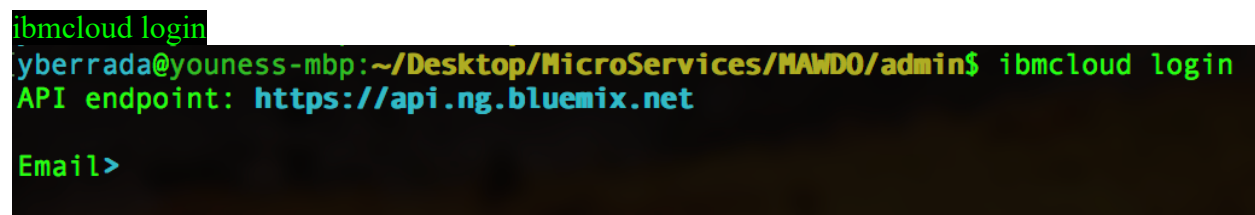
Read the files and understand how they work.

Now you're about to get into the real deal. Next steps will include building docker images, pushing them to IBM cloud registry, and deploying them on IBM cloud.

Install IBM cloud CLI by following the steps on this link:
<https://console.bluemix.net/docs/cli/index.html#overview>

Install Docker by following the instructions found in the link below:
 Windows: <https://docs.docker.com/docker-for-windows/install/>
 macOS: <https://docs.docker.com/docker-for-mac/>

Then login in IBM cloud:



And run the following commands: `ibmcloud cr login`
 /* to be able to push docker images to cloud registry */

```
yberrada@youness-mbp:~/Desktop/MicroServices/MAWDO/admin$ ibmcloud cr login
Logging in to 'registry.ng.bluemix.net'...
Logged in to 'registry.ng.bluemix.net'.

IBM Cloud Container Registry now supports IBM Cloud Identity and Access Management (IAM) access policies. If you started using Container Registry before 4 October 2018, policies are not enforced until you manually enable them. When IAM policy enforcement is enabled, users that order clusters must have the Administrator role on both Kubernetes Service and Container Registry. For more information, see https://console.bluemix.net/docs/services/Registry/registry\_users.html.

OK
```

`ibmcloud cr namespace-add la_factory` /* create namespace name la_factory */\

```
yberrada@youness-mbp:~/Desktop/MicroServices/MAWDO/admin$ ibmcloud cr namespace-add demox
Adding namespace 'demox'...

Successfully added namespace 'demox'

OK
```

You're ready to start building a docker image for each microservice:

From inside the main folder of application, build all the docker images by running the script buildImages: This script automates the step where we build docker images for each microservice.

For macOS: `./buildImages`

For Windows: `buildImages.bat`

```
yberrada@youness-mbp:~/Desktop/WorkshopLaFactory/demo/3MicroServicesDEMO$ ./buildImages.sh

Build Image for admin Microservice
docker build --tag registry.ng.bluemix.net/lafactory/admin-image .
/Users/yberrada/Desktop/WorkshopLaFactory/demo/3MicroServicesDEMO/admin
Sending build context to Docker daemon 10.9MB
Step 1/6 : FROM node:8
----> 1f6c34f7921c
Step 2/6 : COPY package*.json ./
----> Using cache
----> bffa037a8b8c
Step 3/6 : RUN npm install
----> Using cache
----> 0d9c982b5c73
Step 4/6 : COPY . .
----> Using cache
----> a39fca845cb4
Step 5/6 : EXPOSE 3000
----> Using cache
----> e89fd61938fd
Step 6/6 : CMD [ "npm", "start" ]
```

Next, push the image to your IBM Cloud Registry using another script: the following command:

For macOS: `./pushImages`

For Windows: `pushImages.bat`

```
yberrada@youness-mbp: ~/Desktop/WorkshopLaFactory/demo/3MicroServicesDEMO$ ./pushImages.sh

Push Image : admin Microservice
docker push registry.ng.bluemix.net/lafactory/admin-image
The push refers to repository [registry.ng.bluemix.net/la_factory/admin-image]
35570789aec7: Layer already exists
0f517bffd930: Layer already exists
e432beafbea0: Layer already exists
f4427e82e722: Layer already exists
3596679ac8a9: Layer already exists
64f6c1748ef2: Layer already exists
56a89222b908: Layer already exists
a89464ad2a8f: Layer already exists
76dfa41f0a1d: Layer already exists
c240c542ed55: Layer already exists
badfbcebf7f8: Layer already exists
latest: digest: sha256:77346f3d4ca5d16003634ebc8a1ed812410e7ddec4cfbf6529305163dc5c172e size: 2638
Push Image : authentication Microservice
docker push registry.ng.bluemix.net/lafactory/auth-image
The push refers to repository [registry.ng.bluemix.net/la_factory/auth-image]
bfcd075dc3be: Layer already exists
d2facfbfb624: Layer already exists
44659c1f5d43: Layer already exists
f4427e82e722: Layer already exists
3596679ac8a9: Layer already exists
64f6c1748ef2: Layer already exists
56a89222b908: Layer already exists
a89464ad2a8f: Layer already exists
76dfa41f0a1d: Layer already exists
c240c542ed55: Layer already exists
badfbcebf7f8: Layer already exists
latest: digest: sha256:6d9835bb326df6f20c5c45d6d9cc30c0e4e8dfd9f154b73fc282db87b3a78666 size: 2639
Push Image : Bill Microservice
docker push registry.ng.bluemix.net/lafactory/bill-image
```

Now, you have all the docker images in your IBM container registry. Next, we will deploy these images in the cluster.

We'll start by configuring our CLI:

Run the following command

```
ibmcloud cs cluster-config <cluster_name_or_ID>
```

and copy its output and run it in the command line. Refer to the below picture for more details.

```
yberrada@youness-mbp: ~/Desktop/MicroServices/NAWD0/authentication$ ibmcloud cs cluster-config LaFactory_Workshop
OK
The configuration for LaFactory_Workshop was downloaded successfully.

Export environment variables to start using Kubernetes.

export KUBECONFIG=/Users/yberrada/.bluemix/plugins/container-service/clusters/LaFactory_Workshop/kube-config-hou02-LaFactory_Workshop.yml

yberrada@youness-mbp: ~/Desktop/MicroServices/NAWD0/authentication$ export KUBECONFIG=/Users/yberrada/.bluemix/plugins/container-service/clusters/
LaFactory_Workshop/kube-config-hou02-LaFactory_Workshop.yml
```

Time to run those images and create Kubernetes deployments and services:

```
kubectl create -f configuration.yaml
```

```
yberrada@youness-mbp:~/Desktop/WorkshopLaFactory/demo/3MicroServicesDEMO$ kubectl create -f configuration.yaml
deployment.apps "admin" created
service "admin" created
deployment.apps "auth" created
service "auth" created
deployment.apps "bill" created
service "bill" created
deployment.apps "chat" created
service "chat" created
deployment.apps "server" created
service "server" created
```

You successfully deployed the app on the cloud!! Congratulations!!

Run this:

`ibmcloud cs workers <yourclustersname>` Make sure you take note of your clusters public IP.

```
yberrada@youness-mbp:~/Desktop/MicroServices/MAWD0/admin$ ibmcloud cs workers LaFactory_Workshop
OK
ID                                     Public IP                               Machine Type  State  Status  Zone  Version
kube-hou02-pa7e9be54b6ea14992baa93ad84677e18d-w1 173.193.122.87                         free         normal Ready  hou02  1.10.12_1540*
```

You can access the application using the public IP of the cluster and node port of server that you set to 30100. <http://PublicIp:30100/>

In my case its <http://173.193.122.87:30100/>, feel free to access this link to test the app.