Build and Run Micro Service using Docker and Kubernetes

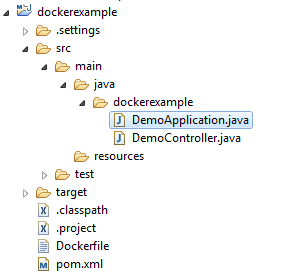
This tutorial mainly focus following steps

1. Build a micro service using spring boot + maven. It will produce a jar file as output with endpoints exposed as REST service
2. Build image using Docker.
3. Push the Docker image into public repository
4. Then deploy this image in Kubernetes using Minikube dashboard using single node cluster

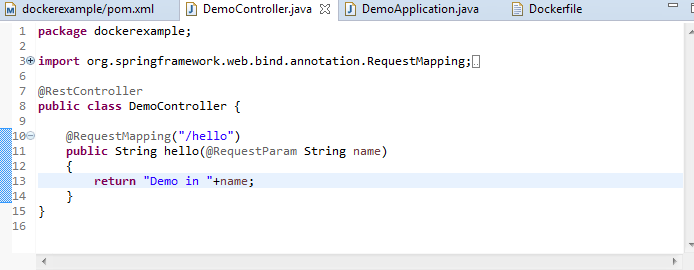
**Note**: I have used windows as operating system for all the steps mentioned above.

1. Build a micro service using Spring boot+maven

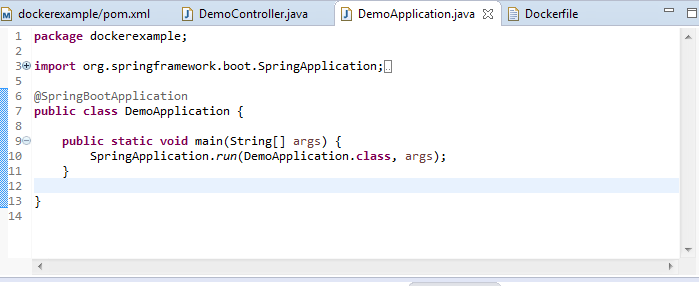
Project structure



Below is the sample controller with endpoint as /hello and parameter as “name”



Below is the spring boot main class.



Once you run this maven target clean package then it will provide output in the target folder as “dockerexample-0.0.1-SNAPSHOT.jar”( as I have used the default settings/artifacts ids)

1. Build image using Docker

Before you build the above micro service as Docker Image you need to install the Docker in your machine. As I have used window OS below are the list of steps for installing the Docker. It will be varying across other OS.

* 1. Docker for Windows

Docker Toolbox has been designed for older version of windows such as windows 7/8.1. you need to have the following configuration for installing Docker on Windows

|  |  |
| --- | --- |
| Windows OS | Windows 7, 8,8.1,10(Home Edition) |
| Memory | 2GB |
| Virtualization | This should be enabled |

You can download Docker Toolbox from <http://www.docker.com/products/docker-toolbox>

If you are using windows 10 other than Home edition you can download the Docker from the following link <https://docs.docker.com/docker-for-windows/>

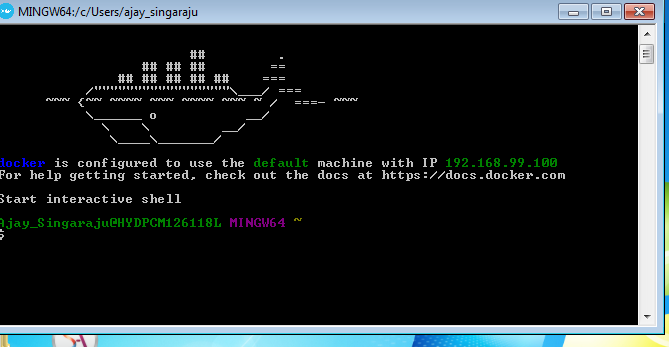
Following configuration is needed

|  |  |
| --- | --- |
| Windows OS | Windows 10 64 bit(other than Home edition) |
| Memory | 2GB |

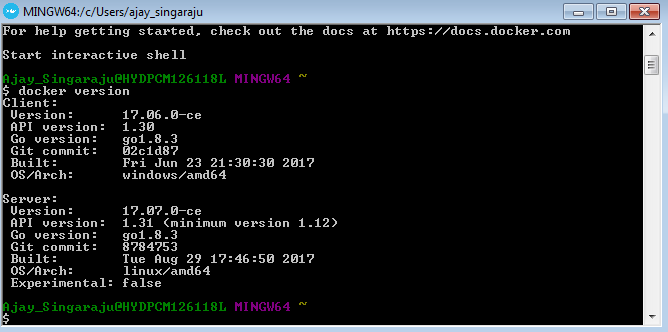
As I am using Windows 7, I have download the Docker Toolbox. Once you download and run the Docker Toolbox installer you will get the following option as shortcut.



Double-click on Docker QuickStart Terminal. If it shows as below then the Docker is successfully installed.



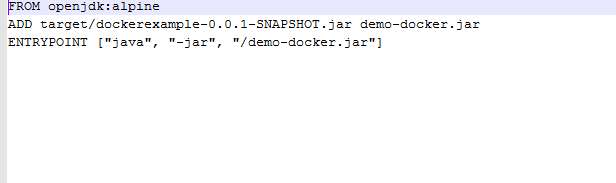
You can type the commands for example “docker version” to see the version of Docker installed.



* 1. Build our Micro Service Docker Image

As the Docker is installed properly, how we can build our micro service as the Docker Image.

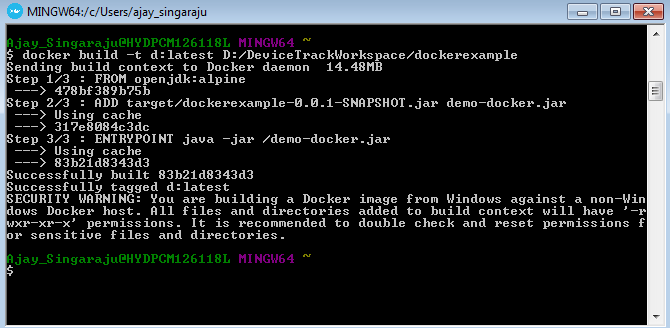
Need to write a “Dockerfile” in our code to build the Docker Image. Below is the snippet for our micro service.



FROM openjdk:alpine -> as we need a java environment to run our jar , we need to get the java container image.

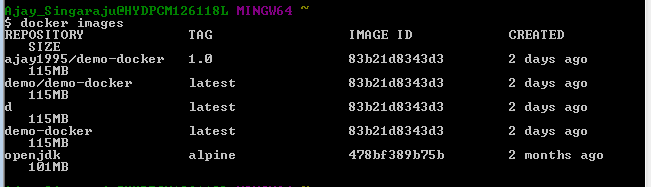
ADD -> adding our dockerexample-0.0.1-snapshot.jar to demo-docker.jar, build a new jar with the content from our maven project

ENTRYPOINT -> execute java –jar demo-docker.jar to run our micro service.



1. Push Docker Image into public repository

Once the images are built then you can type the following command to know the imageid

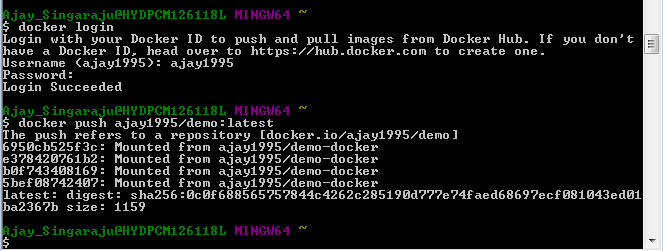


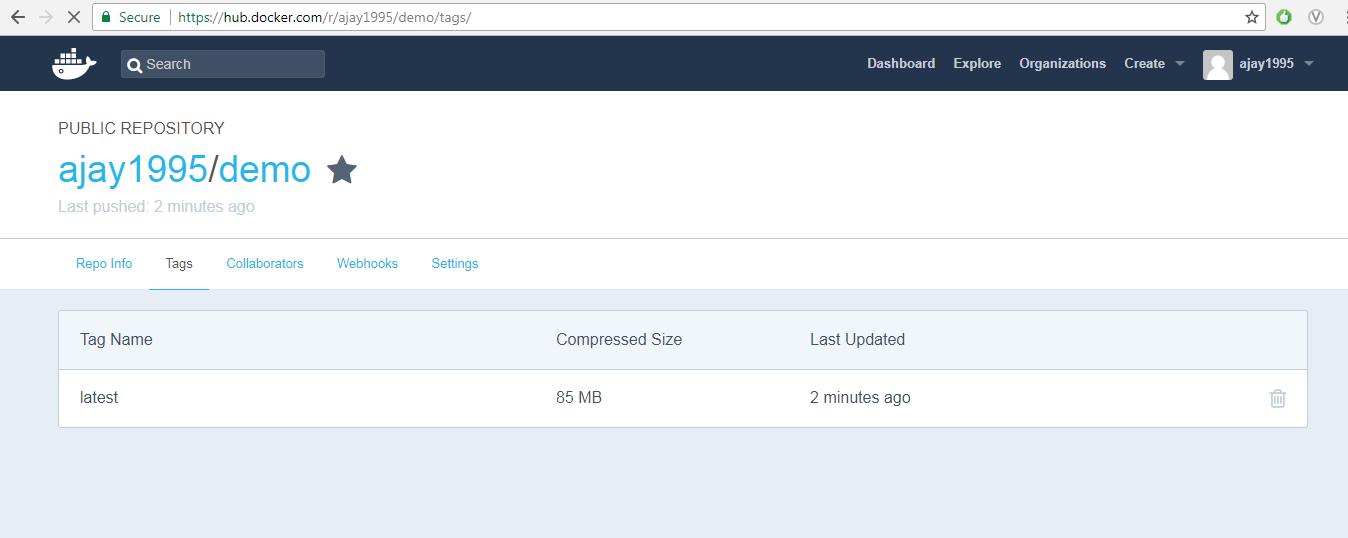
Tag the image using the following command



Push the command to the public repository, for this you need to create the login in the site <https://www.docker.com/products/docker-hub>

And type following command to login and push the image





1. Deploy Docker Image in Kubernetes

The earlier images which we built can be run only in the Docker Container, so it will be accessible only with the Docker Host and port. If we want to expose this service to the outside then we need to deploy it in some server for which we need Kubernetes.

Download minikube.exe and kubectl.exe and set it to path variables.

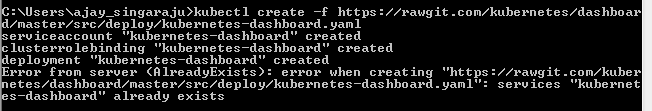
Start the minikube



IP at which Kubernetes is accessible.



Run the following command to configure kubernetes dashboard

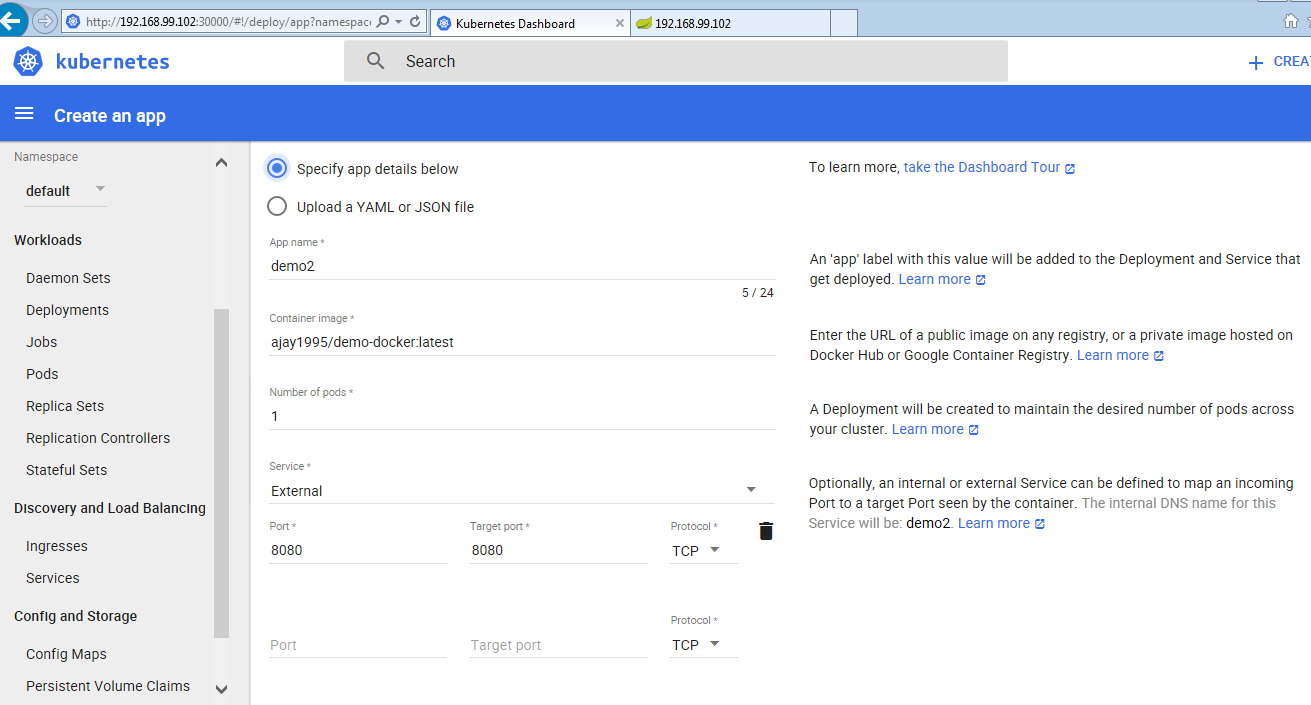


Run the minikube dashboard to open the browser and do the deployments

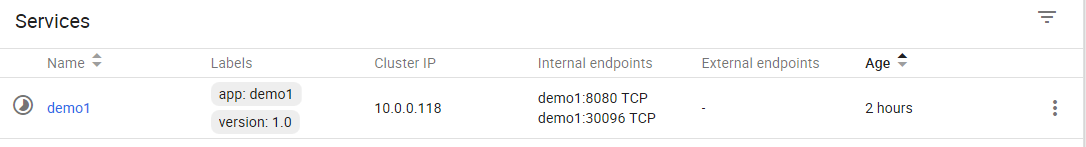


Usually it opens in the browser with <http://192.168.99.100> , but as I have Docker Instance running it has opened <http://192.168.99.102>

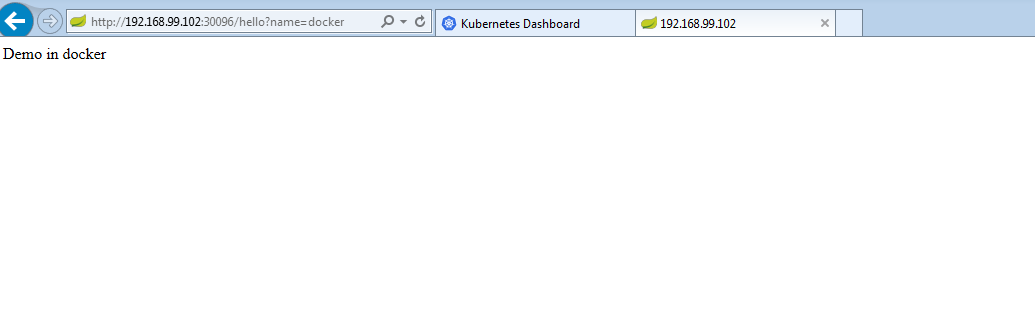
Either you can have the YAML file or you can directly enter the details in the UI screen as shown below and click on deploy



Once that is successfully deployed it will appear as below under services and port details will be visible.



Access with the link and port as given in the internal endpoints as below, it will provide the desired output



Once done you can stop the kubernetes using the below command

