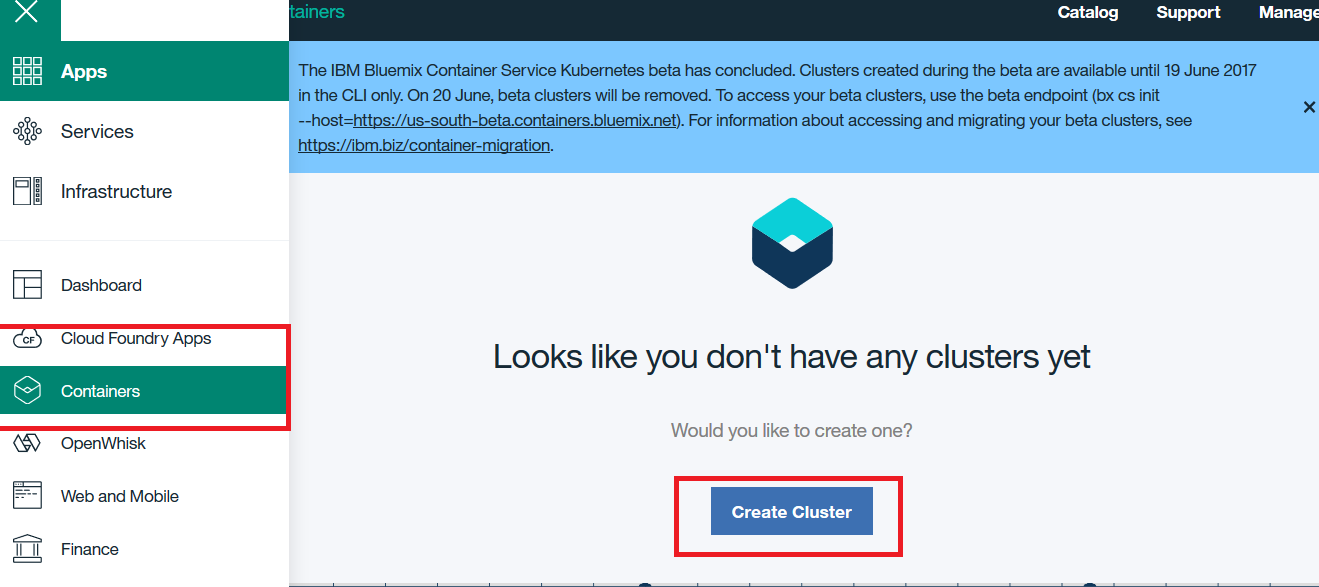
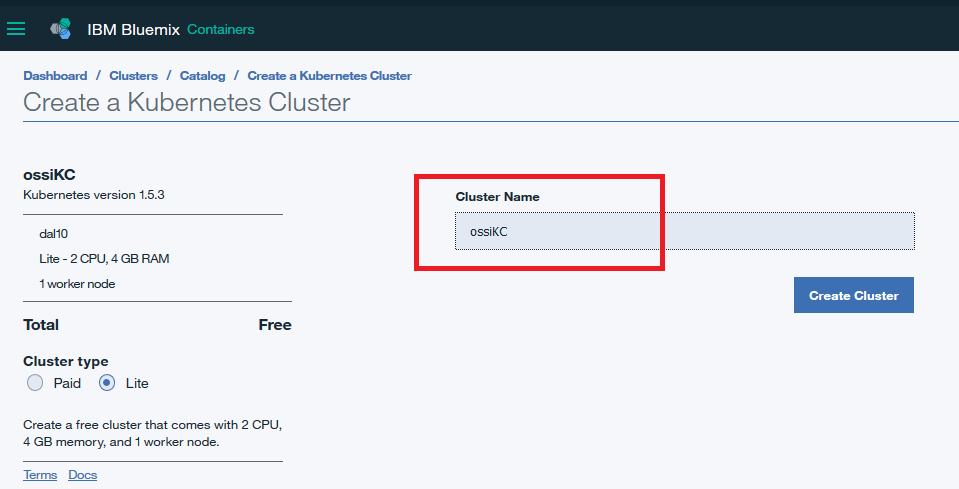
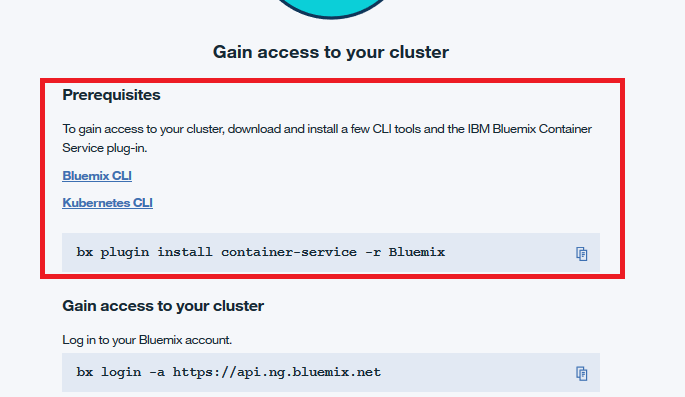
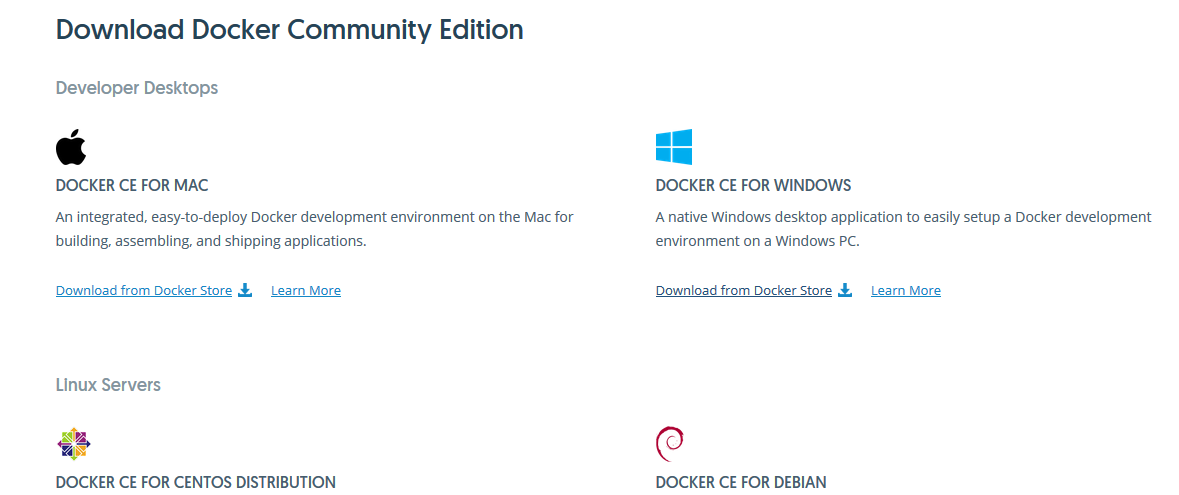
# Create & deploy java microservice in Bluemix Kubernetes Cluster

1. Create Kubernetes cluster
   1. Login to Bluemix <https://console.ng.bluemix.net/>
   2. Create Organization and Namespace.
   3. Navigate to Containers and create cluster (takes several minutes to complete the provisioning)



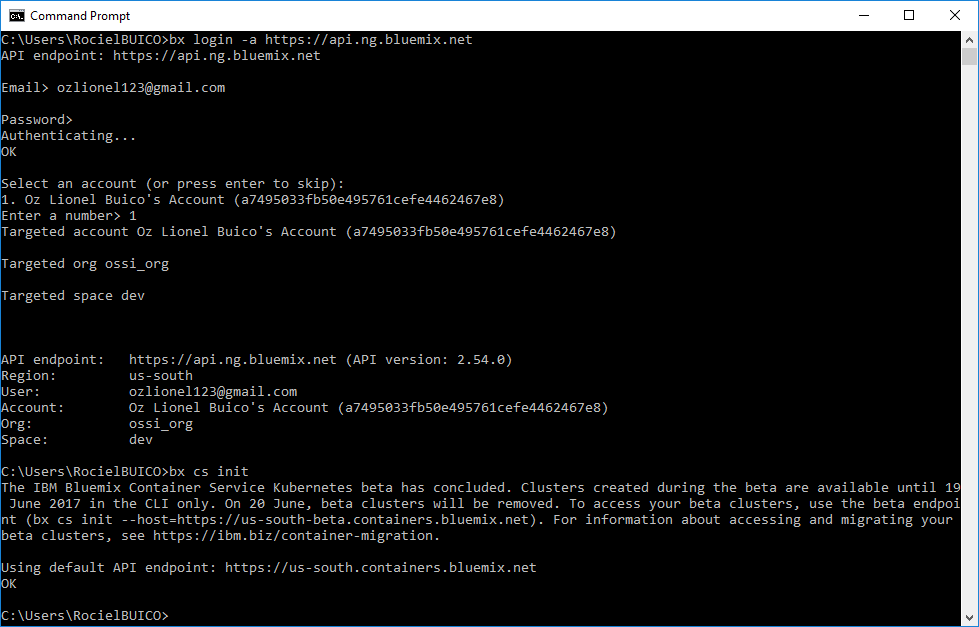


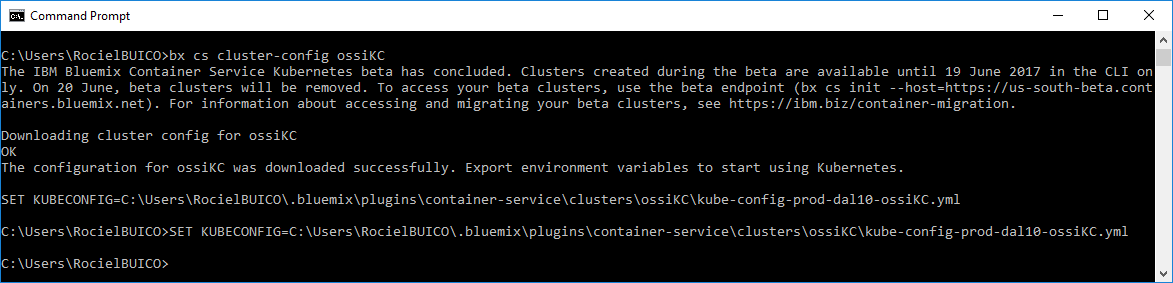
1. Install command line interfaces bluemix, docker and kubernetes
   1. Follow the links for Bluemix & Kubernetes CLI  
      
   2. Install Docker community edition - <https://www.docker.com/community-edition>



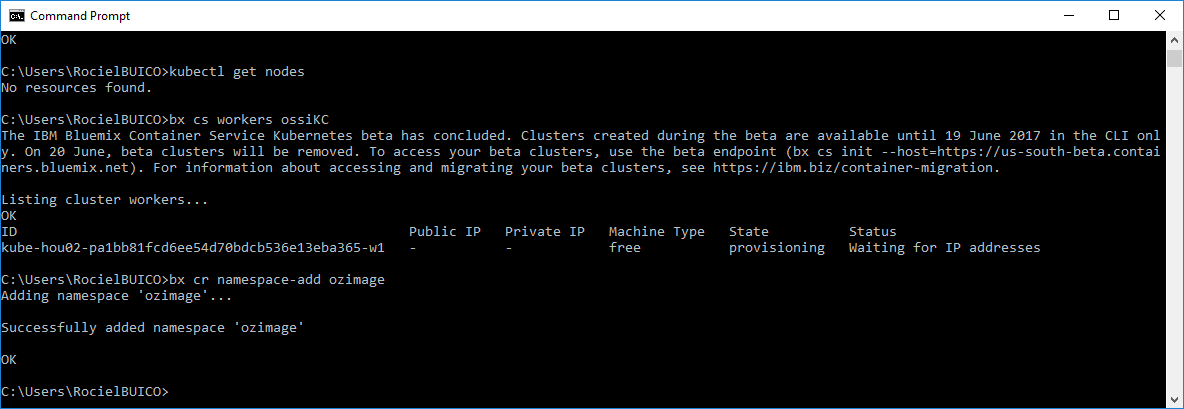
* 1. Once CLI is installed, gain access to your cluster via cmd prompt.



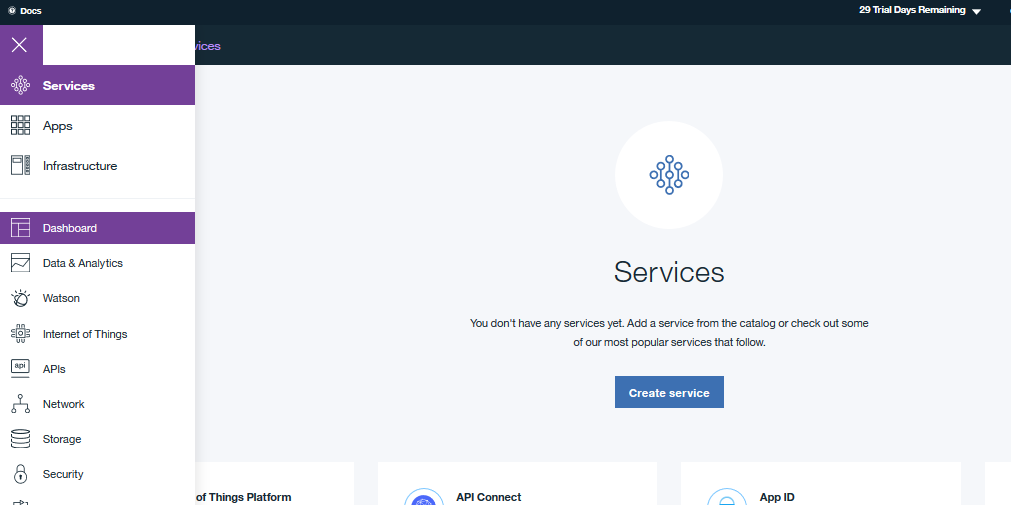




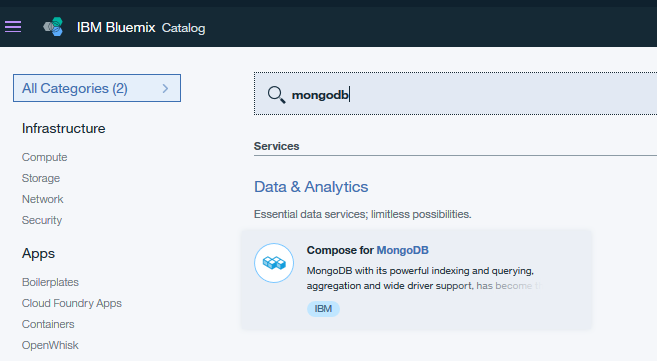
1. Create private registry to store container images
   1. Create name space (\>**bx cr namespage-add ozimage** )

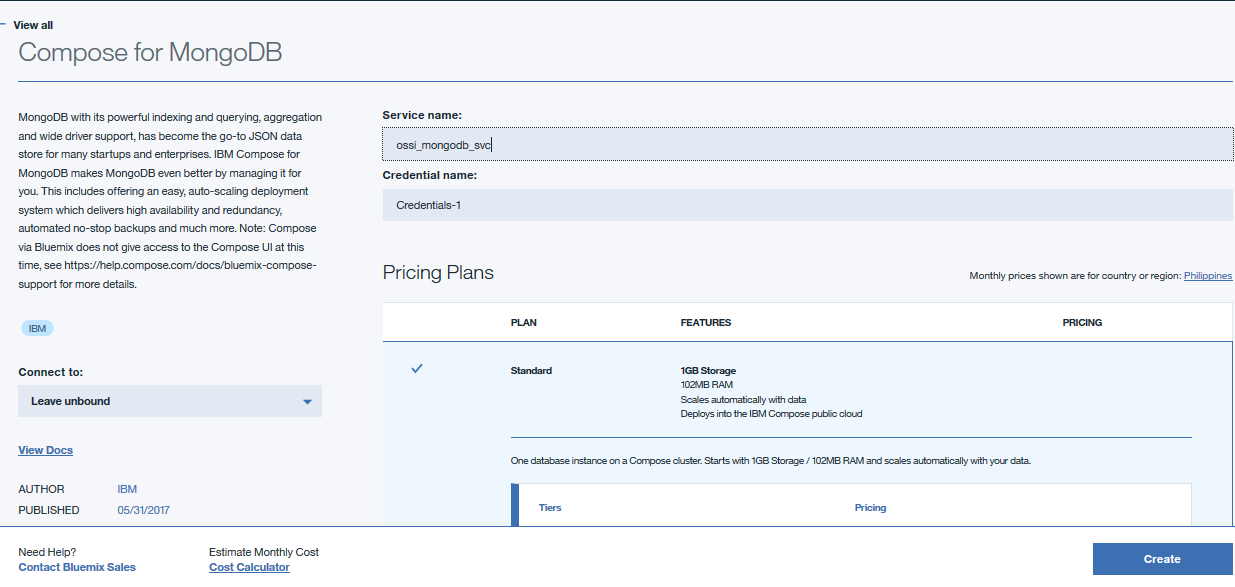


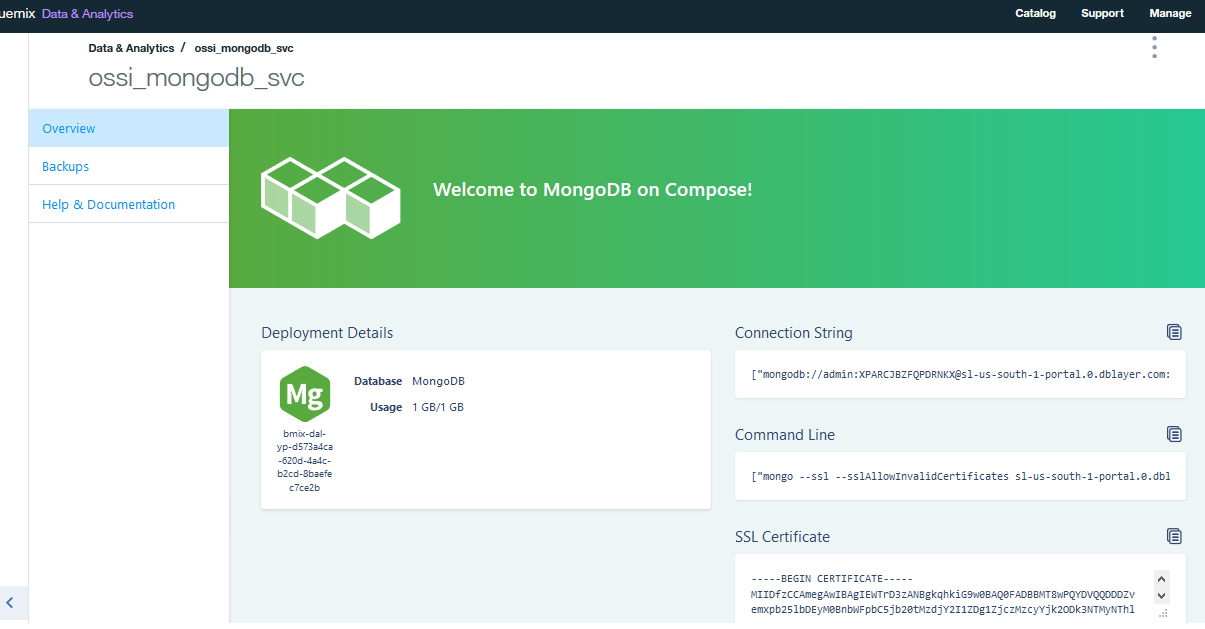
1. Create mongodb service in bluemix
   1. In Bluemix, navigate to Services > Dashboard and then click create service.

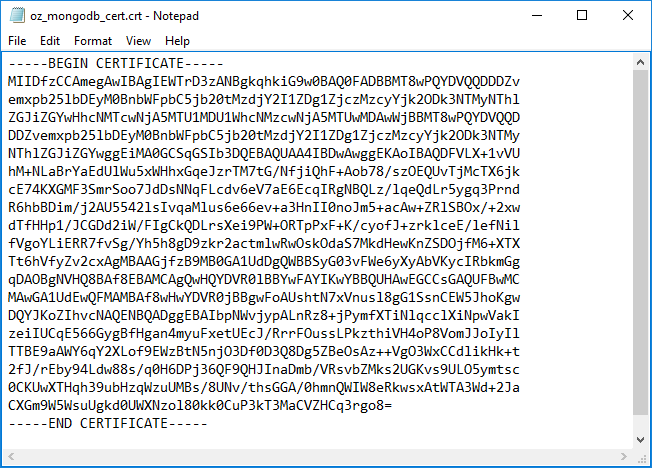


* 1. Find Compose for MongoDB service and select it.



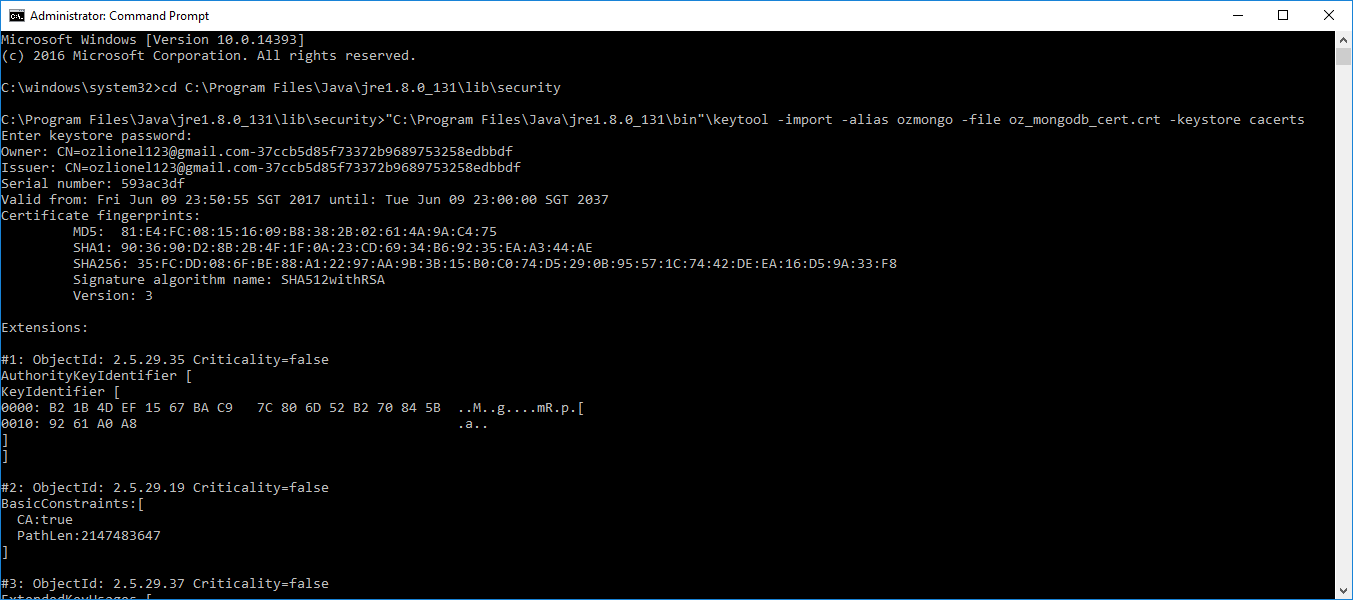
* 1. Provide name of your mongodb service.  
     
  2. Copy the SSL Certificate and save it to a file, you need to install the certificate to gain access of the remote mongodb using mongo client (ex mongobooster)



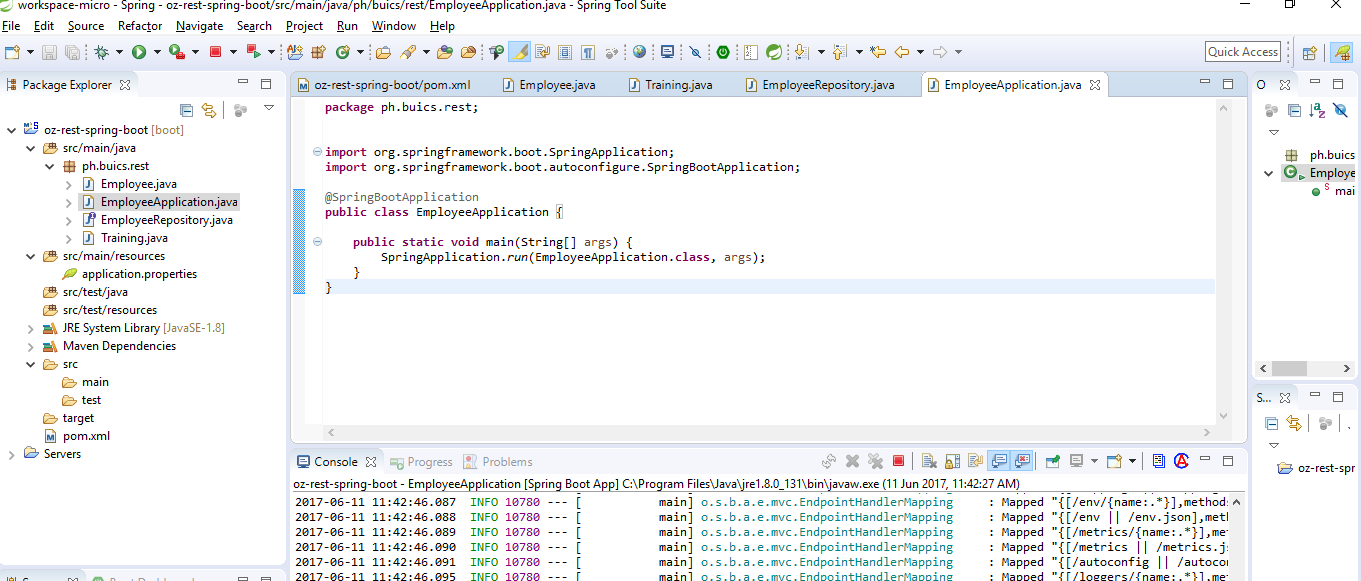


* 1. Import certificate into the java keystore cacerts using keytool.  
     keytool -import -alias ozmongo -file oz\_mongodb\_cert.crt -keystore cacerts

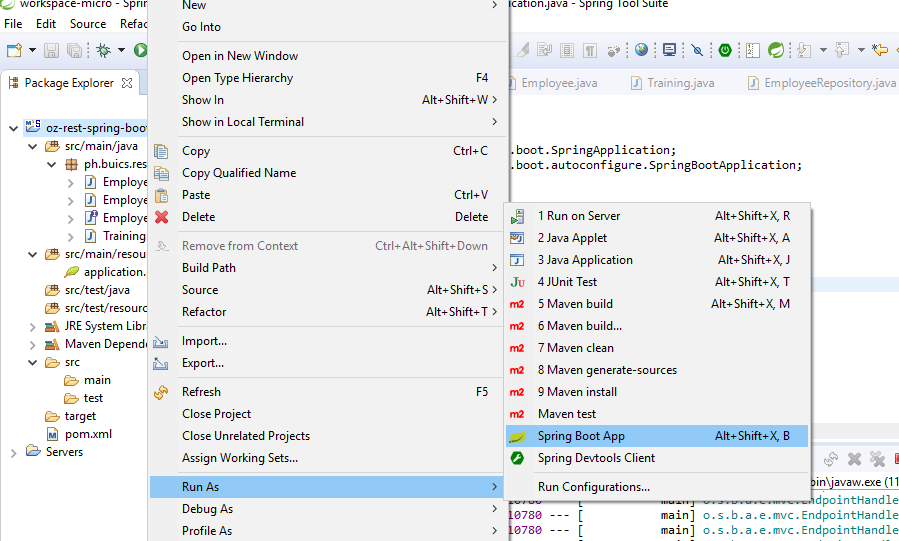
https://docs.oracle.com/javase/tutorial/security/toolfilex/rstep1.html



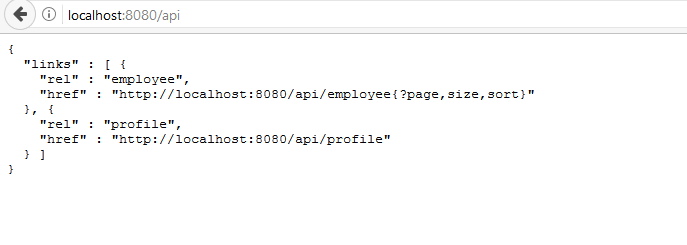
1. Create rest api with spring-boot and use spring-data api to access mongodb database
   1. Clone repo – git clone https://github.com/buics/employee-rest.git
   2. Import the employee-rest project into eclipse / sts.



* 1. Run project as Spring Boot App.



* 1. Using your favorite browser, browse the rest api.



1. Create docker image & push to repository
   1. Create a file name Dockerfile and paste the following commands. The **cacerts** file is the java keystore file where your mongodb certificate was imported (steps 4.e).

FROM java:8

VOLUME /tmp

ADD target/ oz-rest-spring-boot\*.jar oz-rest-spring-boot.jar

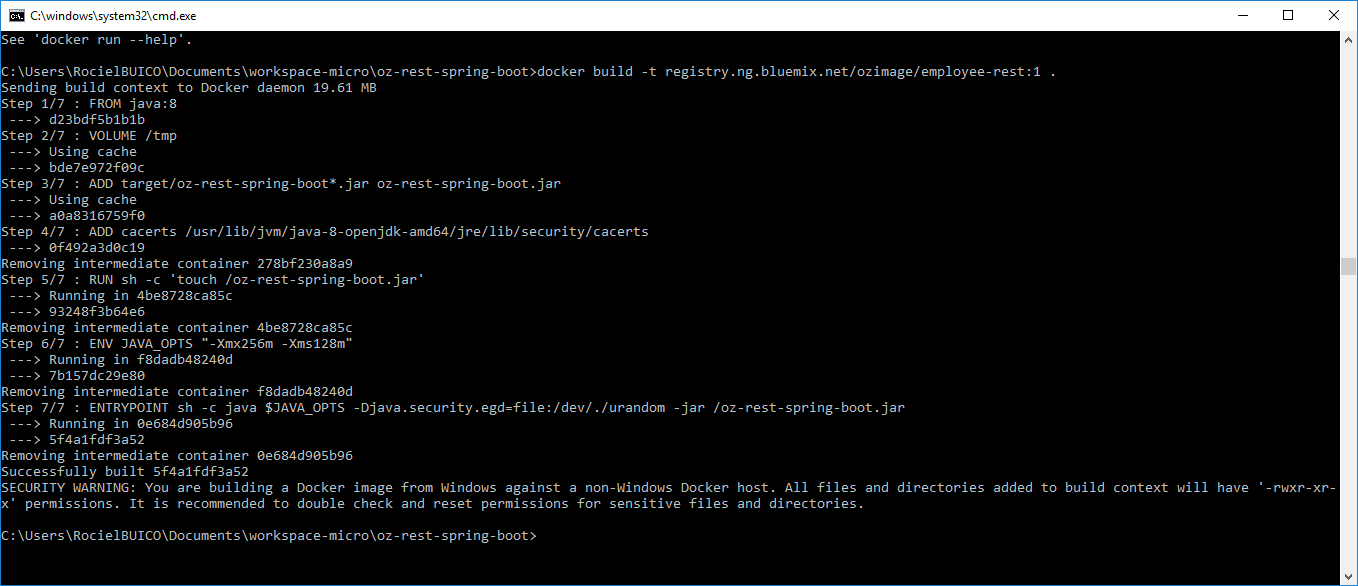
ADD cacerts /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/security/cacerts

RUN sh -c 'touch / oz-rest-spring-boot.jar'

ENV JAVA\_OPTS="-Xmx256m -Xms128m"

ENTRYPOINT [ "sh", "-c", "java $JAVA\_OPTS -Djava.security.egd=file:/dev/./urandom -jar / oz-rest-spring-boot.jar"]

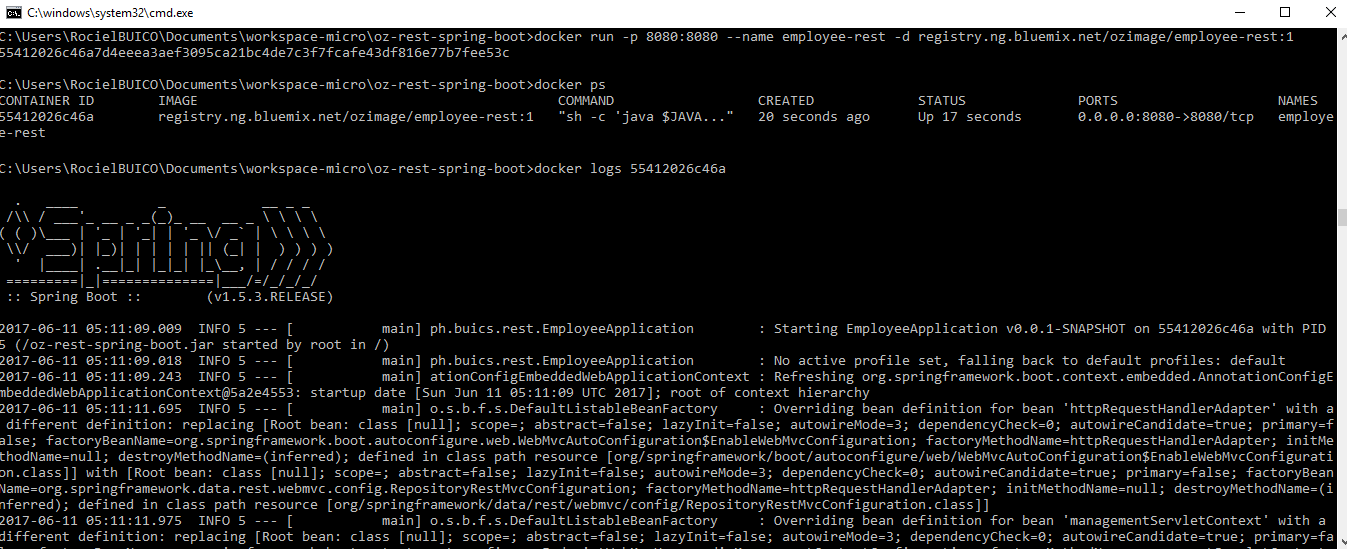
* 1. Build docker image using the bluemix\_registry/namespace\_steps\_3a/project\_name:tag



1. Run docker image
   1. Run docker container using the newly create image. This is to test the application locally.

>docker run -p 8080:8080 --name employee-rest -d registry.ng.bluemix.net/ozimage/employee-rest:1

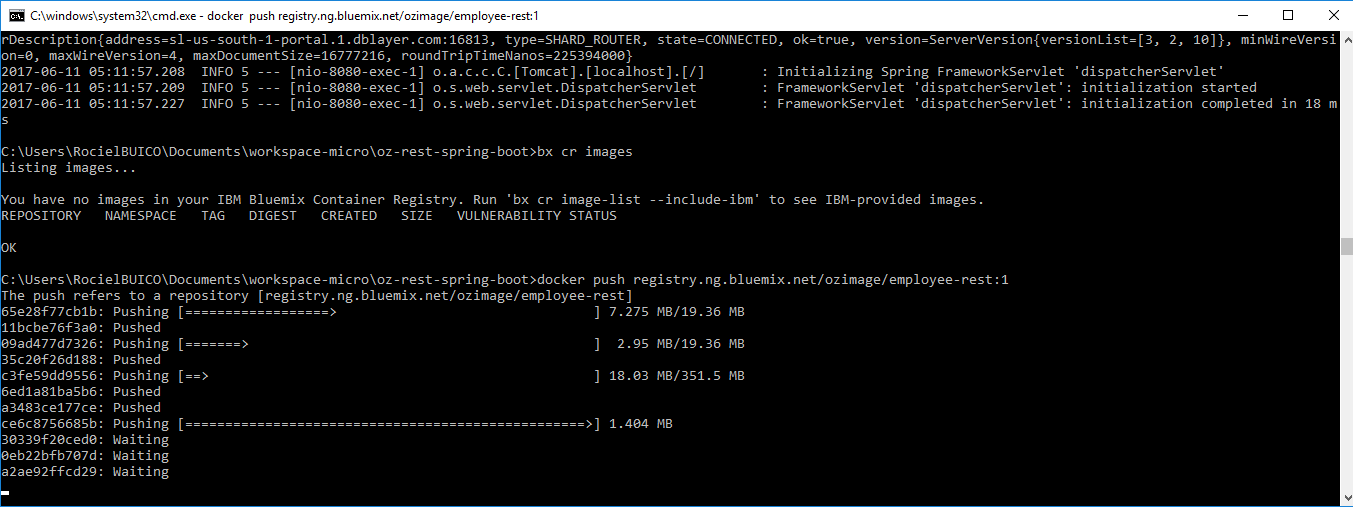
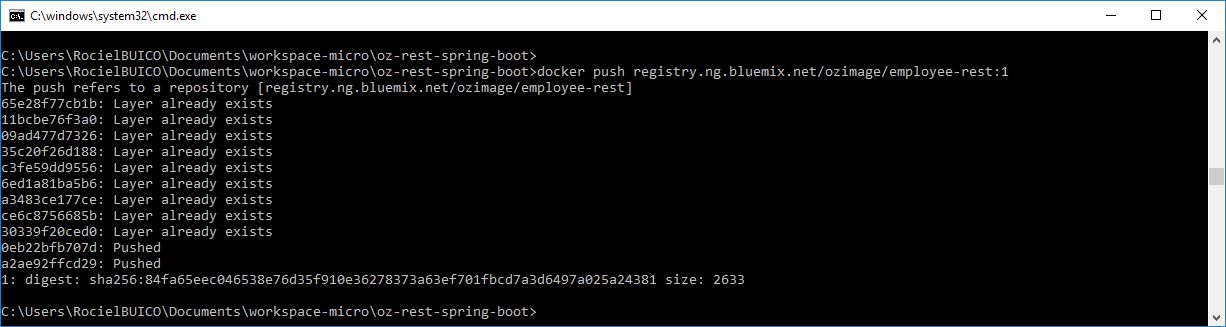
* 1. Run >docker ps to see the processes and confirm if employee-rest is running.
  2. Run >docker logs <container\_id> to see the logs generated.

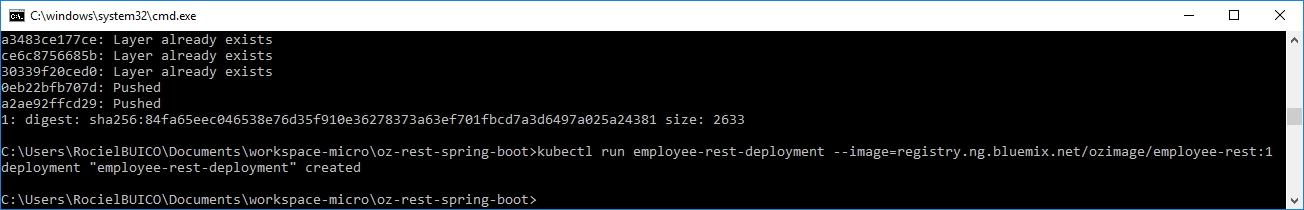


1. Create kubernetes deployment
   1. Push the newly created image to bluemix.

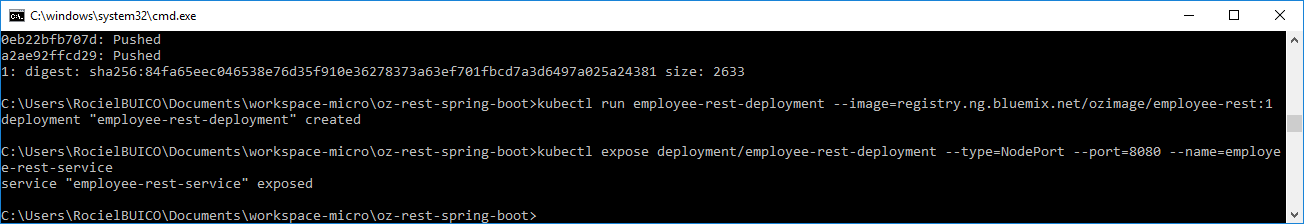
>bx cr images

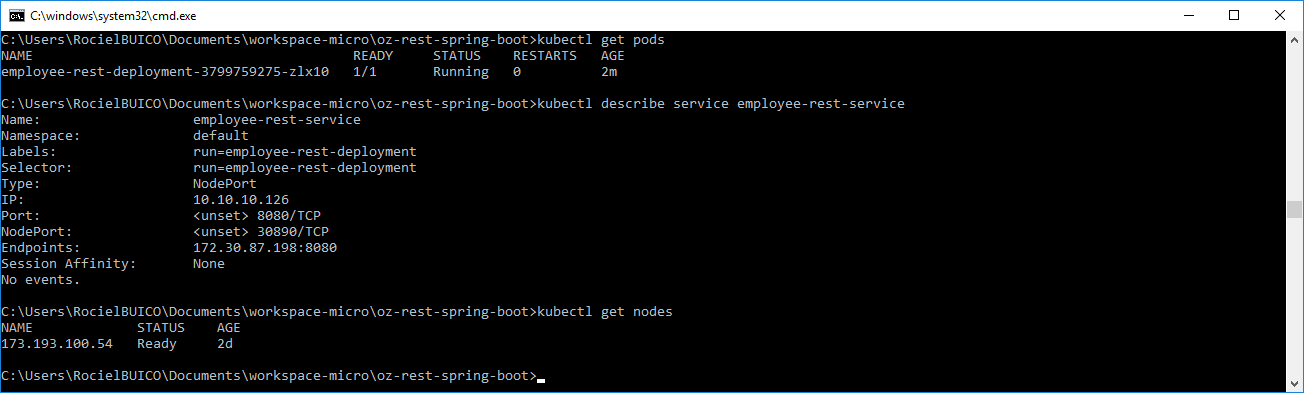
> docker push registry.ng.bluemix.net/ozimage/employee-rest:1

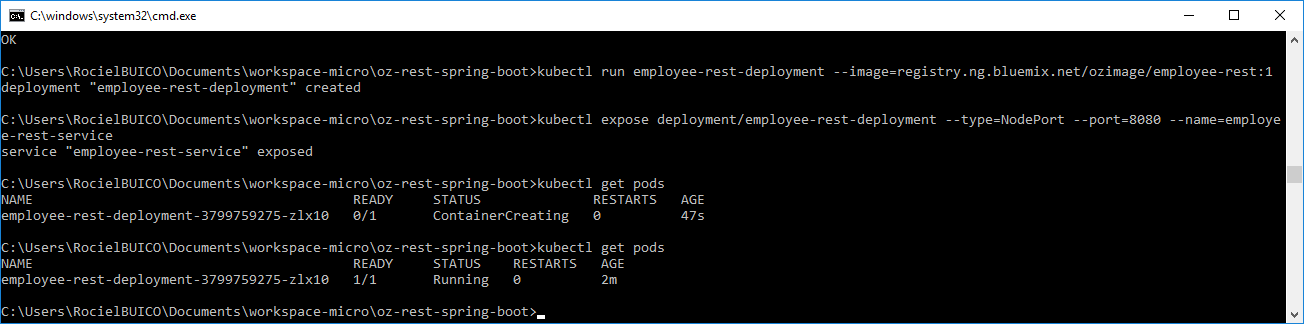
  


* 1. kubectl run employee-rest-deployment --image=registry.ng.bluemix.net/ozimage/employee-rest:1  
     

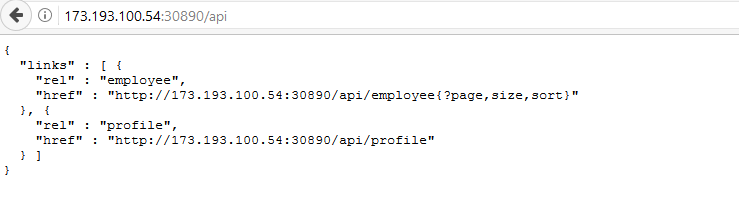
1. Expose kubernetes service
   1. kubectl expose deployment/employee-rest-deployment --type=NodePort --port=8080 --name=employee-rest-service



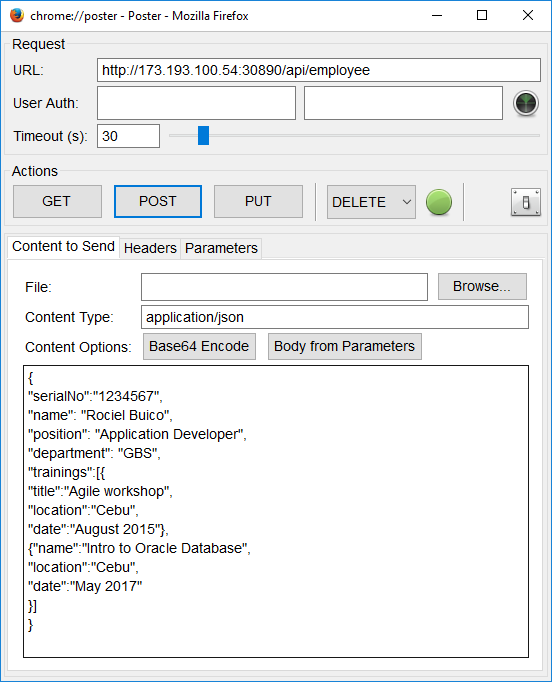
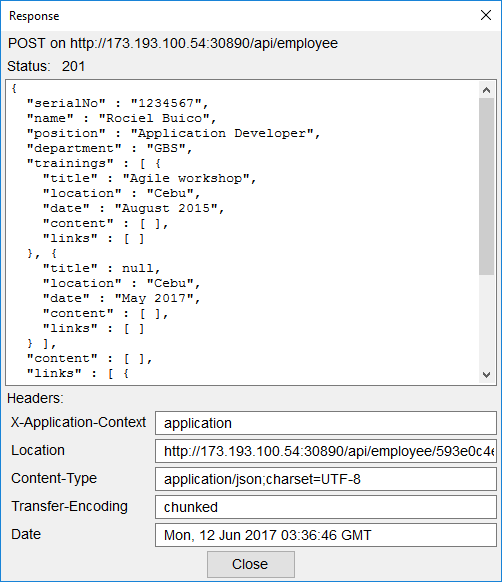
1. Access the rest api
   1. kubectl describe service employee-rest-service  
      
   2. run bx cs workers <cluster\_name>



* 1. Browse the application

http://173.193.100.54:30890/api   


* 1. Using poster firefox plugin as client, send post request to create new db entry.

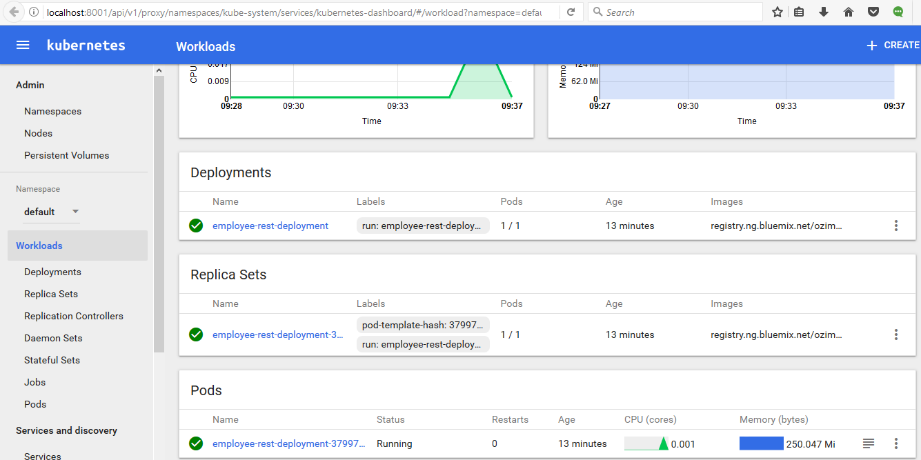
  


* 1. Browse the employee api



* 1. Run >kubectl proxy to view kubernetes UI

1. <http://localhost:8001/ui>



* 1. Get logs - >kubectl logs <name\_of\_pods>  
     