Learn Docker , Kubernetes, AWS

Docker-compose to Kubernetes deployment object

https://kubernetes.io/docs/tasks/configure-pod-container/translate-compose-kubernetes/
https://medium.com/skillshare-team/from-docker-compose-to-minikube-d94cbe97acda

Deploy spring boot application in kubernetes

https://www.kindsonthegenius.com/deploy-springboot-with-mysql-to-kubernetes-minikube-step-by-step-tutorial/

Steps to install minikube

https://www.kindsonthegenius.com/setup-kubernetes-locally-deploy-springboot-application-step-by-step-tutorial/

//Deploy Spring Boot in EKS

https://katharharshal1.medium.com/deploy-spring-boot-application-into-aws-eks-using-jenkins-cicd-2ced0e0d894c

//K8S basic

https://www.checkpoint.com/cyber-hub/cloud-security/what-is-kubernetes/what-is-a-kubernetes-cluster/#:~:text=A%20Kubernetes%20(K8s)%20cluster%20is,%2C%20virtual%2C%20and%20cloud%20servers.

https://enterprisersproject.com/article/2020/9/pod-cluster-container-what-is-difference

https://www.theserverside.com/answer/Kubectl-apply-vs-create-Whats-the-difference

https://www.nginx.com/blog/deploying-nginx-nginx-plus-docker/

https://www.kubermatic.com/blog/introduction-to-kubernetes-replicasets/

https://www.containiq.com/post/kubectl-delete

https://www.stacksimplify.com/aws-eks/kubernetes-for-absolute-beginners/create-kubernetes-pods-with-kubectl/

Convert any docker compose to kubernetes deployment file

command :**kompose -f docker-compose.yml convert**

```
C:\Users\Dhruva's PC>kompose -f docker-compose.yml convert
INFO Kubernetes file "mysqldb-service.yaml" created
INFO Kubernetes file "springboot-docker-container-service.yaml" created
INFO Kubernetes file "mysqldb-deployment.yaml" created
INFO Kubernetes file "springboot-docker-container-deployment.yaml" created

C:\Users\Dhruva's PC>
```

//RDS

https://www.stacksimplify.com/aws-eks/kubernetes-storage/aws-eks-storage-with-aws-rds-database/

//EKS

https://www.stacksimplify.com/aws-eks/eks-cluster/install-aws-eksctl-kubectl-cli/

Command to run the docker compose

docker-compose up

For running docker the updated Java file

docker-compose up --build

https://stackoverflow.com/questions/62193878/docker-container-not-updating-on-code-change

Steps to push images in docker hub

https://www.section.io/engineering-education/docker-push-for-publishing-images-to-docker-hub/

Command to start the minikube

minikube start

INFORMATIO N ABOUT AWS:

https://pilotcoresystems.com/insights/aws-fargate-vs-ecs-which-one-is-best-for-my-workload/

https://www.ques10.com/p/50544/run-your-application-with-docker-compose-on-aws-ec/?

1.go to ec2
2. launch instances
3. create a key value pair to generate ssh

sudo chmod 666 /var/run/docker.sock

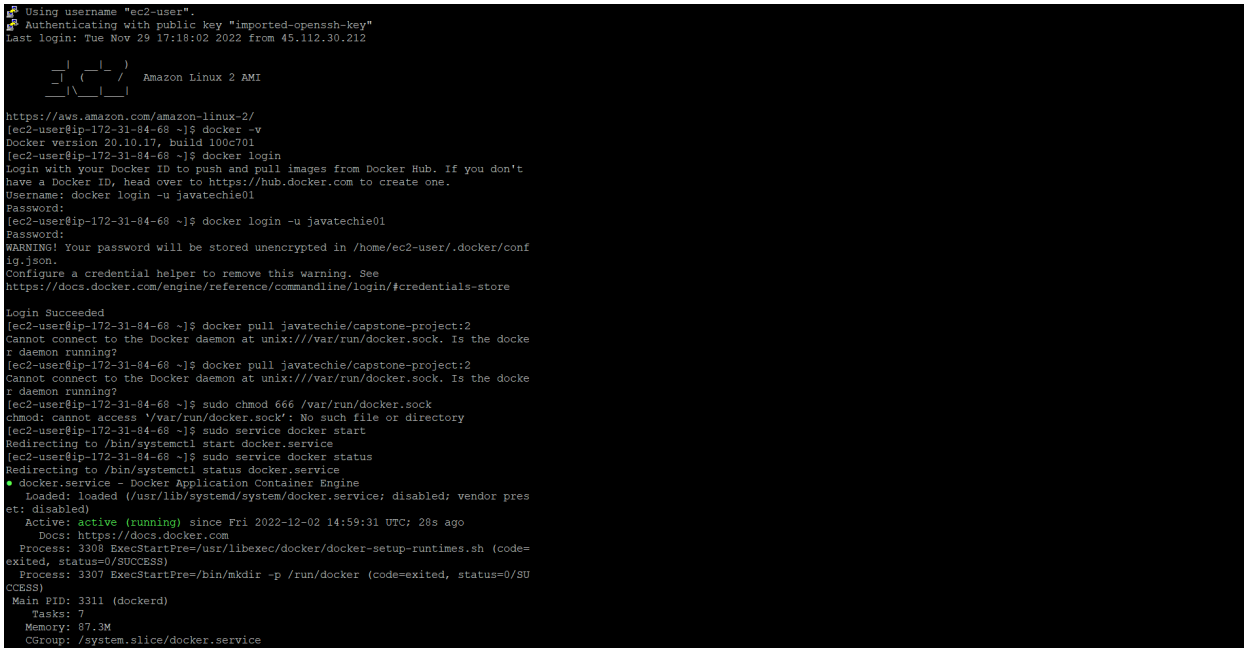
chmod 777 /var/run/docker.sock - when login failed

sudo service docker start

docker-compose --version

https://www.edureka.co/community/50699/how-do-i-install-docker-compose-on-linux

AWS EC2 instance Screenshots



```
CGESS)
 Main PID: 3311 (dockerd)
    Tasks: 7
   Memory: 87.3M
   CGroup: /system.slice/docker.service
           └─3311 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont...

Dec 02 14:59:30 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:30 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:30 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:30 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:31 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:31 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:31 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:31 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Dec 02 14:59:31 ip-172-31-84-68.ec2.internal systemd[1]: Started Docker Appli...
Dec 02 14:59:31 ip-172-31-84-68.ec2.internal dockerd[3311]: time="2022-12-02T...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-84-68 ~]$ sudo chmod 666 /var/run/docker.sock
[ec2-user@ip-172-31-84-68 ~]$ docker pull javatechie/capstone-project:2
Error response from daemon: pull access denied for javatechie/capstone-project,
repository does not exist or may require 'docker login': denied: requested acces
s to the resource is denied
[ec2-user@ip-172-31-84-68 ~]$ docker login -u javatechie01
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/conf
ig.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-84-68 ~]$ docker pull javatechie/capstone-project:2
Error response from daemon: pull access denied for javatechie/capstone-project,
repository does not exist or may require 'docker login': denied: requested acces
s to the resource is denied
[ec2-user@ip-172-31-84-68 ~]$ docker pull javatechie01/capstone-project:2
2: Pulling from javatechie01/capstone-project
9a0b9cd2dfe6: Pull complete
c637408ee7df: Pull complete
4c517093c276: Pull complete
301cc7d68c2a: Pull complete
17ca9bf9231a: Pull complete
9ae101e5c786: Pull complete
04baa409344e: Pull complete
f0b6015bf853: Pull complete
6005bb052ef8: Pull complete
99f303d57050: Pull complete
```

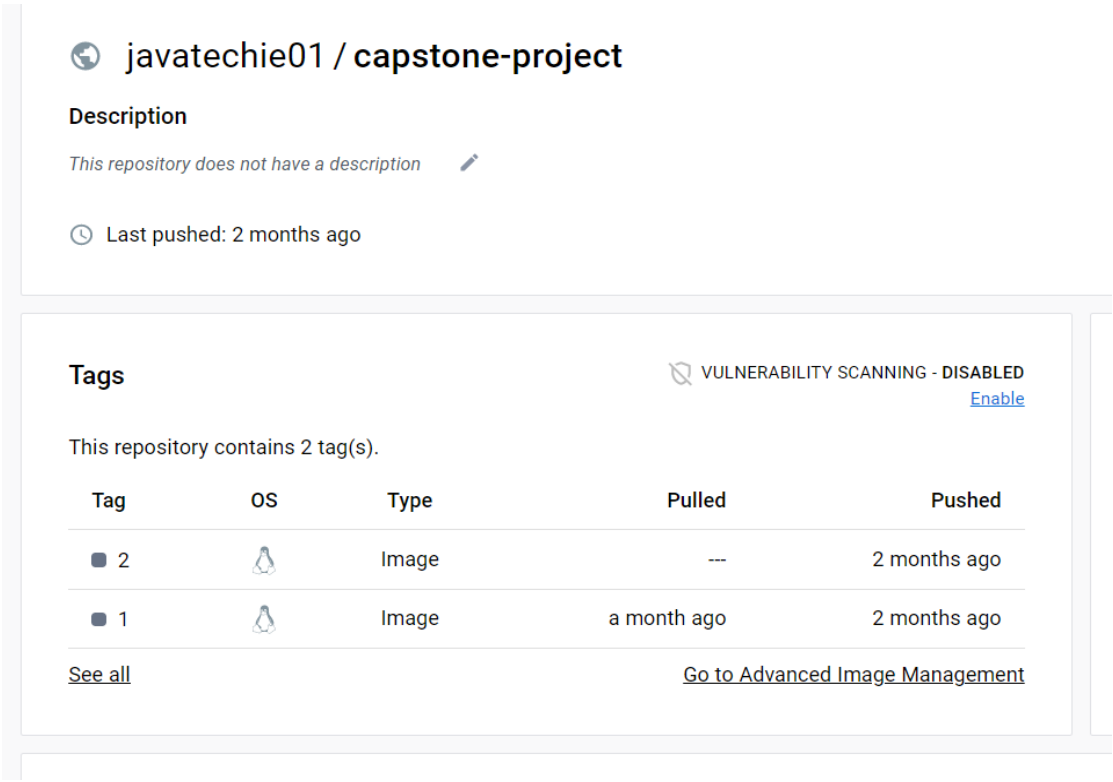
```
Login Succeeded
[ec2-user@ip-172-31-84-68 ~]$ docker pull javatechie/capstone-project:2
Error response from daemon: pull access denied for javatechie/capstone-project,
repository does not exist or may require 'docker login': denied: requested acces
s to the resource is denied
[ec2-user@ip-172-31-84-68 ~]$ docker pull javatechie01/capstone-project:2
2: Pulling from javatechie01/capstone-project
9a0b9cd2dfe6: Pull complete
c637408ee7df: Pull complete
4c517093c276: Pull complete
301cc7d68c2a: Pull complete
17ca9bf9231a: Pull complete
9ae101e5c786: Pull complete
04baa409344e: Pull complete
f0b6015bf853: Pull complete
6005bb052ef8: Pull complete
99f303d57050: Pull complete
307a9a80c1df: Pull complete
Digest: sha256:55d2f4aa17fd27821a7fc575d2921485681a8aa5ac8411b75d7a163d895dfba1
Status: Downloaded newer image for javatechie01/capstone-project:2
docker.io/javatechie01/capstone-project:2
[ec2-user@ip-172-31-84-68 ~]$ docker pull javatechie01/capstone-project:1
1: Pulling from javatechie01/capstone-project
001c52e26ad5: Pull complete
d9d4b9b6e964: Pull complete
2068746827ec: Pull complete
9daef329d350: Pull complete
d85151f15b66: Pull complete
52a8c426d30b: Pull complete
8754a66e0050: Pull complete
09fbf4ff9ffc: Pull complete
Digest: sha256:6555cde68b86066955ad09116777cf47fe949d00b567677499e432072aea77f7
Status: Downloaded newer image for javatechie01/capstone-project:1
docker.io/javatechie01/capstone-project:1
[ec2-user@ip-172-31-84-68 ~]$ vim docker-compose.yml
[ec2-user@ip-172-31-84-68 ~]$ docker compose up
docker: 'compose' is not a docker command.
See 'docker --help'
[ec2-user@ip-172-31-84-68 ~]$ docker-compose --version
-bash: docker-compose: command not found
[ec2-user@ip-172-31-84-68 ~]$
```

Docker Compose 👍

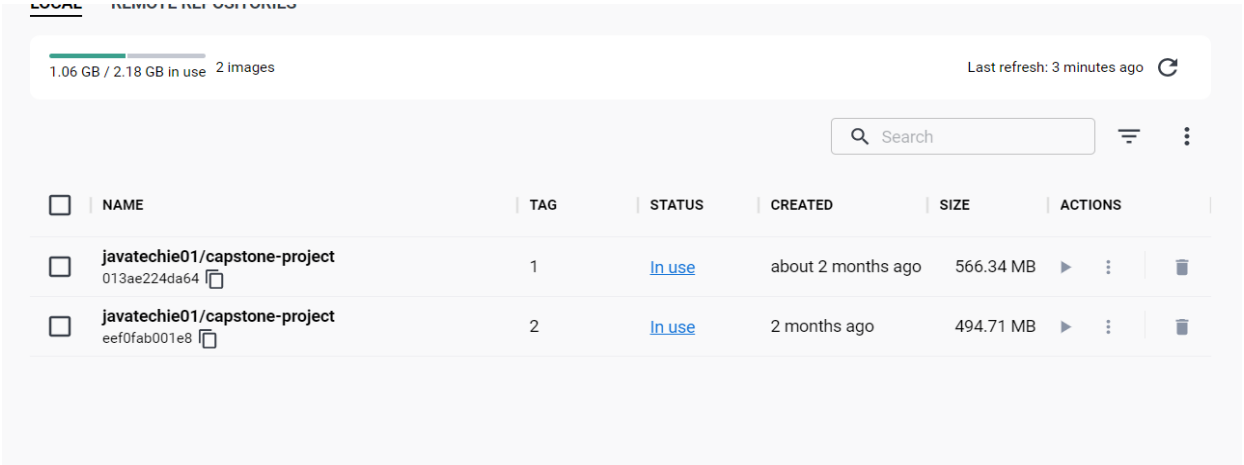
```yaml
version: '3'
services:
  mysqldb:
    image: 'mysql:5.7'
    environment:
       MYSQL_ROOT_PASSWORD: [redacted]
       MYSQL_DATABASE: springjdbc
       MYSQL_ROOT_USER : root
       MYSQL_PASSWORD: [redacted]
    ports:
      - "3307:3306"
  springboot-docker-container:
    image: springboot-docker-container
    ports:
      - "8080:8080"
    environment:
      SPRING_DATASOURCE_URL: jdbc:mysql://mysqldb:3306/springjdbc?autoReconnect=true&useSSL=false
      SPRING_DATASOURCE_USERNAME: "root"
      SPRING_DATASOURCE_PASSWORD: "[redacted]"
    build:
      context: "./"
      dockerfile: "Dockerfile"
    depends_on:
      - mysqldb
```

```
FROM openjdk:8
EXPOSE 8080
ADD target/spring-boot-docker.jar spring-boot-docker.jar
ENTRYPOINT ["java","-jar","/spring-boot-docker.jar"]
```

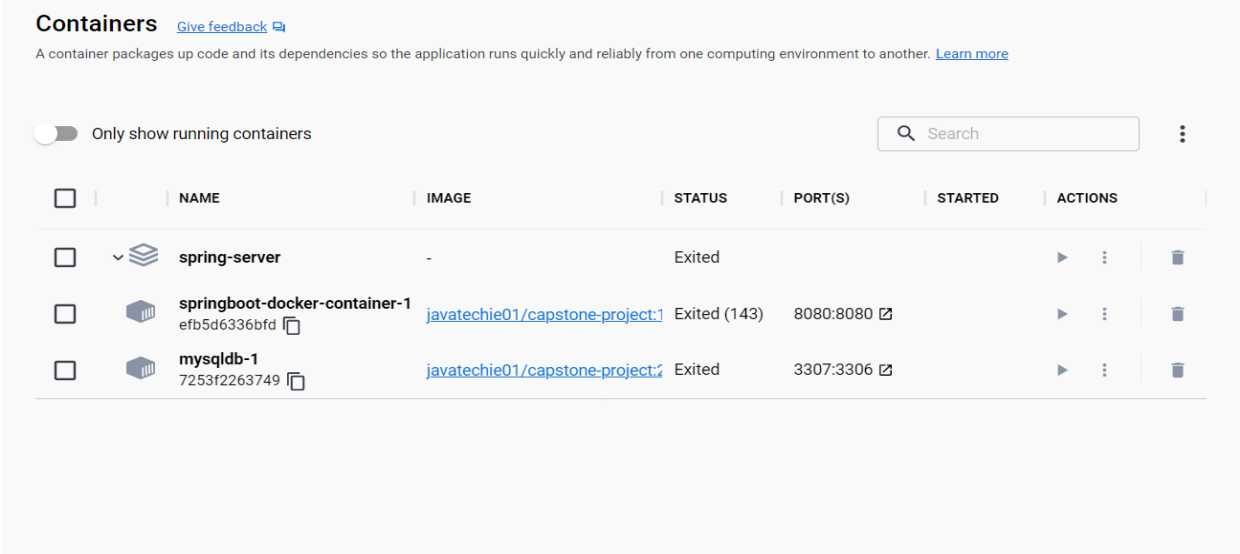
Docker Hub with the list of Images pushed in a repository



Docker Images



Docker Container

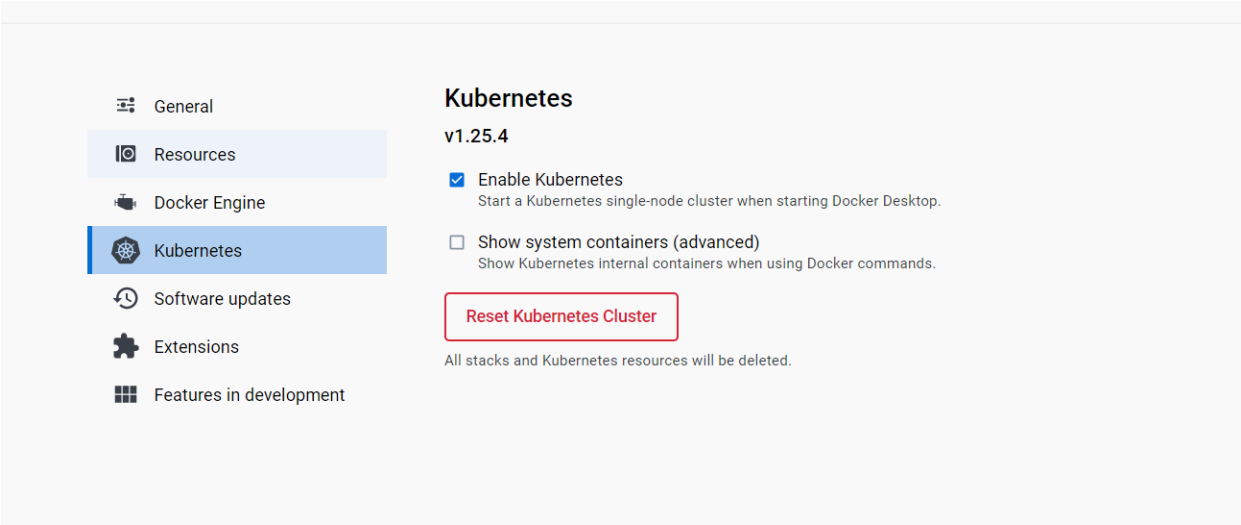


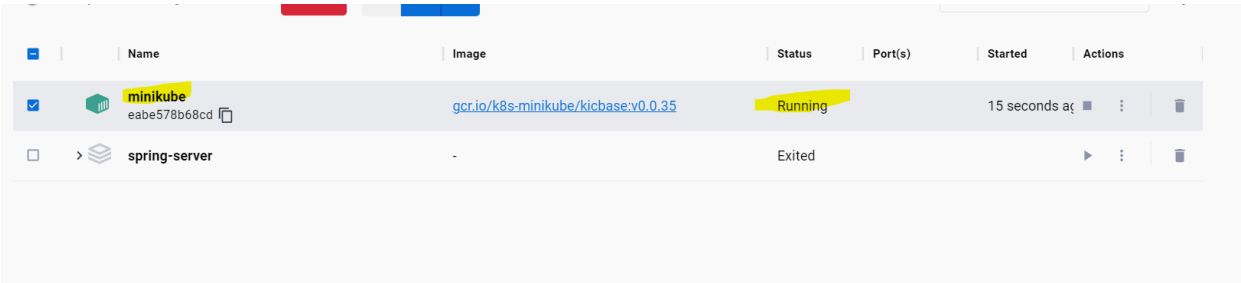
Kubernetes Deployment Commands:

1. Enable the kubernetes from docker desktop settings.
2. For Minikube to start : **minikube start.**
3. **kubectl apply -f mysqldb-deployment.yaml , kubectl apply -f mysqldb-service.yaml,**
kubectl apply -f springboot-docker-container-deployment.yaml , kubectl apply -f springboot-docker-container-service.yaml
4. Once the deployment objects are created , write : **kubectl get pods**
5. To delete a pod **: kubectl delete -f mysqldb-deployment.yaml**

```
D:\WorkspaceJava\spring-server>kubectl get pods
NAME                        READY   STATUS              RESTARTS   AGE
mysqldb-7f8dd9b57c-qfztl    0/1     ContainerCreating   0          60s

D:\WorkspaceJava\spring-server>kubectl apply -f mysqldb-service.yaml
service/mysqldb created

D:\WorkspaceJava\spring-server>kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
mysqldb-7f8dd9b57c-qfztl    1/1     Running   0          91s

D:\WorkspaceJava\spring-server>kubectl apply -f springboot-docker-container-deployment.yaml
deployment.apps/springboot-docker-container created

D:\WorkspaceJava\spring-server>kubectl apply -f springboot-docker-container-service.yaml
service/springboot-docker-container created

D:\WorkspaceJava\spring-server>kubectl get pods
NAME                                        READY   STATUS              RESTARTS   AGE
mysqldb-7f8dd9b57c-qfztl                    1/1     Running             0          2m7s
springboot-docker-container-699b4cc78b-hdwjg   0/1     ContainerCreating   0          18s

D:\WorkspaceJava\spring-server>kubectl get pods
NAME                                        READY   STATUS              RESTARTS   AGE
mysqldb-7f8dd9b57c-qfztl                    1/1     Running             0          2m41s
springboot-docker-container-699b4cc78b-hdwjg   0/1     ContainerCreating   0          52s
```

```
D:\WorkspaceJava\spring-server>kubectl delete -f springboot-docker-container
-deployment.yaml
deployment.apps "springboot-docker-container" deleted
```





Once the pods are running properly,

1. **kubectl get service --all-namespaces (to check the services running)**

```
D:\WorkspaceJava\spring-server>kubectl get service --all-namespaces
NAMESPACE     NAME                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes                    ClusterIP   10.96.0.1        <none>        443/TCP                  29d
default       mysqldb                       ClusterIP   10.103.246.243   <none>        3307/TCP                 23m
default       springboot-docker-container   ClusterIP   10.105.39.243    <none>        8080/TCP                 20m
kube-system   kube-dns                      ClusterIP   10.96.0.10       <none>        53/UDP,53/TCP,9153/TCP   29d

D:\WorkspaceJava\spring-server>
```

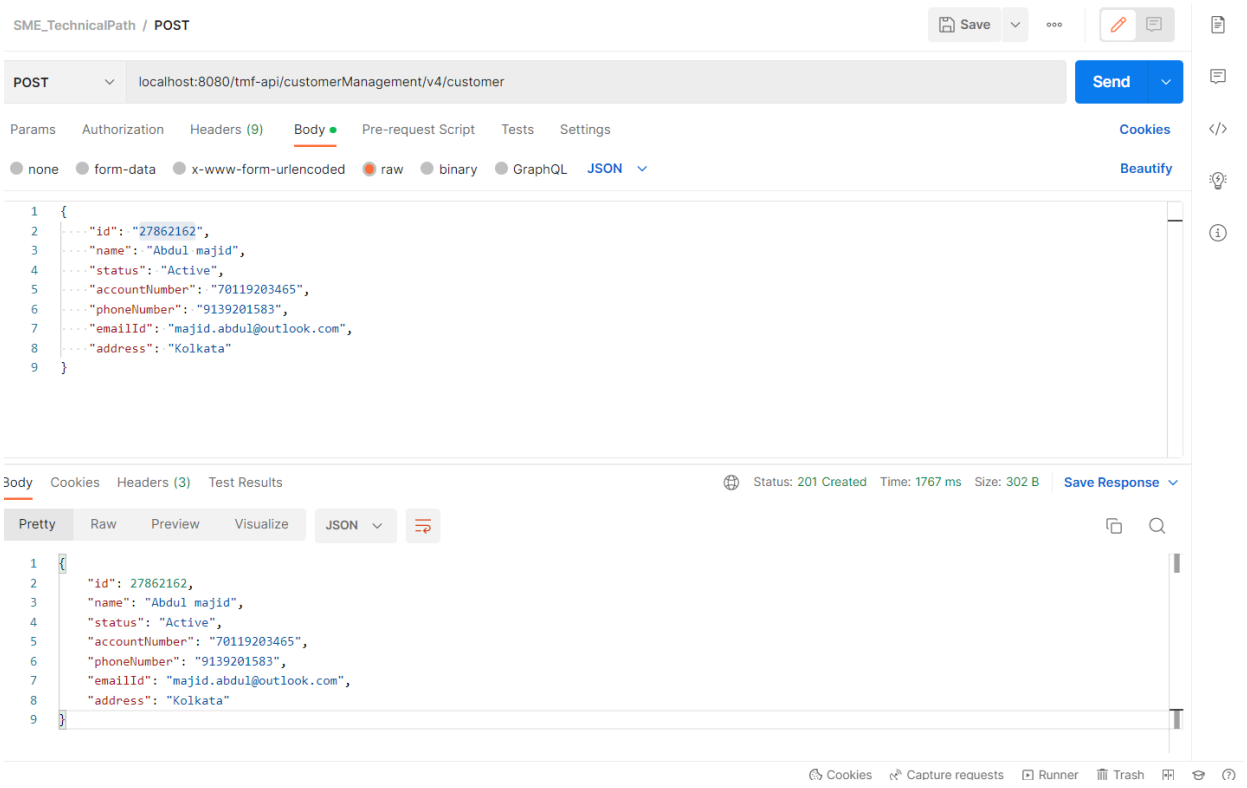
2. **To check the service logs**

```
D:\WorkspaceJava\spring-server>kubectl get pods springboot-docker-container-6dc5879cb7-xxtl5
NAME                                          READY   STATUS    RESTARTS   AGE
springboot-docker-container-6dc5879cb7-xxtl5   1/1     Running   0          19m

D:\WorkspaceJava\spring-server>kubectl logs springboot-docker-container-6dc5879cb7-xxtl5

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.5.9.RELEASE)

2023-02-14 03:41:39.240  INFO 1 --- [           main] io.swagger.Swagger2SpringBoot            : Starting Swagger2SpringBoot v1.0.0 on springboot-docker-con
tainer-6dc5879cb7-xxtl5 with PID 1 (/spring-boot-docker.jar started by root in /)
2023-02-14 03:41:39.256  INFO 1 --- [           main] io.swagger.Swagger2SpringBoot            : No active profile set, falling back to default profiles: de
fault
2023-02-14 03:41:40.748  INFO 1 --- [           main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.Annota
tionConfigEmbeddedWebApplicationContext@2cdf8d8a: startup date [Tue Feb 14 03:41:40 UTC 2023]; root of context hierarchy
2023-02-14 03:42:09.249  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTrans
actionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIB$$b1773e5b]
 is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
```

3. Test the application to verify the proper response ,
   Kubectl port-forward <pod-name> 8080:8080 (example)

```
D:\WorkspaceJava\spring-server>kubectl port-forward springboot-docker-container-6dc5879cb7-sfnkc 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
```

Kubernetes test data screenshots:




```
    WHERE
        x.id = :id */ select
            count(*) as col_0_0_
        from
            customer customer0_
        where
            customer0_.id=?
2023-02-18 14:55:43.944 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [1] as [BIGINT] - [27862162]
2023-02-18 14:55:43.962 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicExtractor   : extracted value ([col_0_0_] : [BIGINT]) - [0]
Hibernate:
    /* load io.swagger.model.Customer */ select
        customer0_.id as id1_0_0_,
        customer0_.account_number as account_2_0_0_,
        customer0_.address as address3_0_0_,
        customer0_.email_id as email_id4_0_0_,
        customer0_.name as name5_0_0_,
        customer0_.phone_number as phone_nu6_0_0_,
        customer0_.status as status7_0_0_
    from
        customer customer0_
    where
        customer0_.id=?
2023-02-18 14:55:44.049 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [1] as [BIGINT] - [27862162]
Hibernate:
    /* insert io.swagger.model.Customer
        */ insert
        into
            customer
            (account_number, address, email_id, name, phone_number, status, id)
        values
            (?, ?, ?, ?, ?, ?, ?)
2023-02-18 14:55:44.279 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [1] as [VARCHAR] - [70119203465]
2023-02-18 14:55:44.280 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [2] as [VARCHAR] - [Kolkata]
2023-02-18 14:55:44.281 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [3] as [VARCHAR] - [majid.abdul@outlook.c
om]
2023-02-18 14:55:44.281 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [4] as [VARCHAR] - [Abdul majid]
2023-02-18 14:55:44.282 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [5] as [VARCHAR] - [9139201583]
2023-02-18 14:55:44.282 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [6] as [VARCHAR] - [Active]
2023-02-18 14:55:44.282 TRACE 1 --- [nio-8080-exec-3] o.h.type.descriptor.sql.BasicBinder      : binding parameter [7] as [BIGINT] - [27862162]
Hibernate:
```