Architectural anti-patterns when delivering a software ecosystem with Kubernetes

Contents

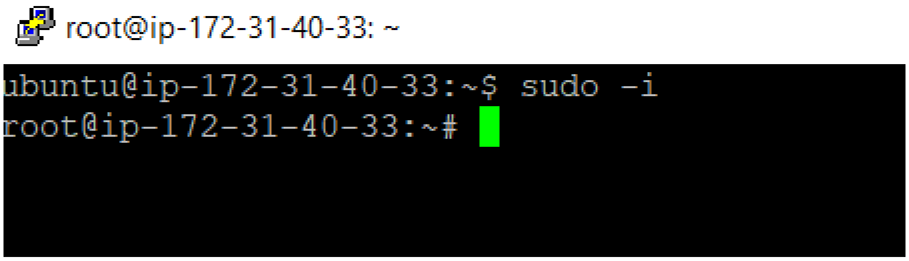
1. Setup

In order to be able to solve all the examples we must first set up our environments.

The first step is to create your own EC2 virtual machine in the space provided by the trainer. You can as well choose to use your own computer if you have the needed privileges. However, be aware that the commands might be different in some situations if you use your own computer – mostly depending on the OS that you have installed. The EC2 will have installed an Ubuntu server 16.04, so the commands in this document are focused on this OS version.

After first accessing the EC2 virtual machine, let's make sure we have root permissions. You can do this by running:

sudo -i

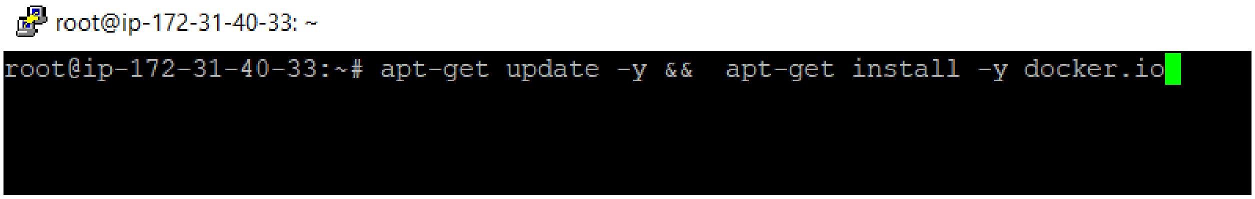


```
ubuntu@ip-172-31-40-33:~$ sudo -i
root@ip-172-31-40-33:~#
```

Next step is to install Docker. To achieve this, please run the following command:

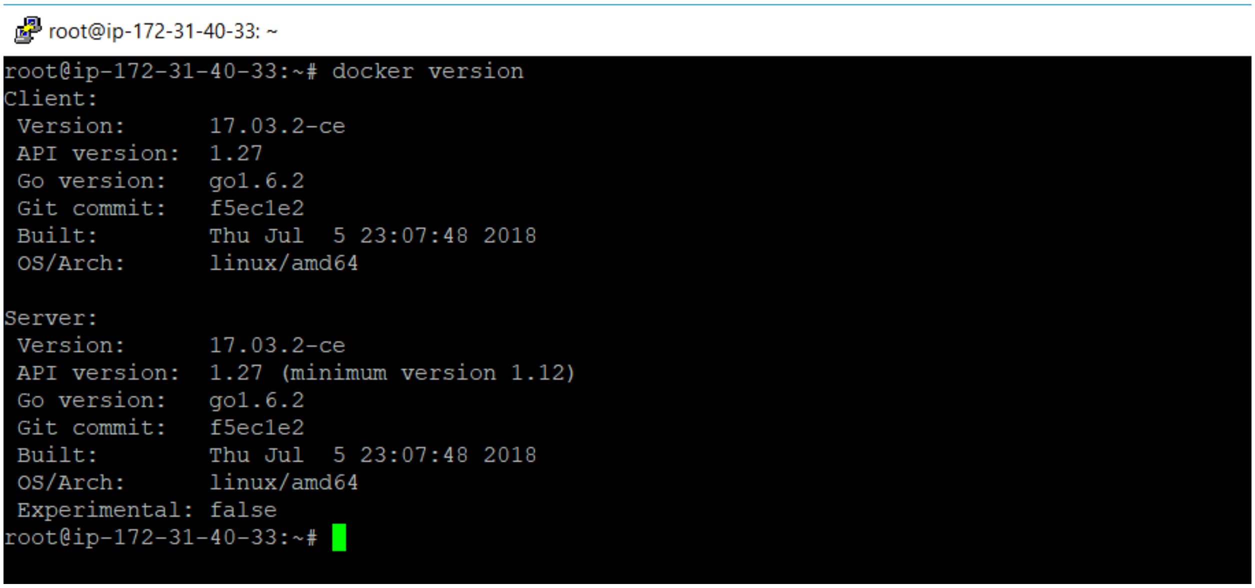
apt-get update -y

apt-get install -y docker.io



```
root@ip-172-31-40-33:~# apt-get update -y &&  apt-get install -y docker.io
```

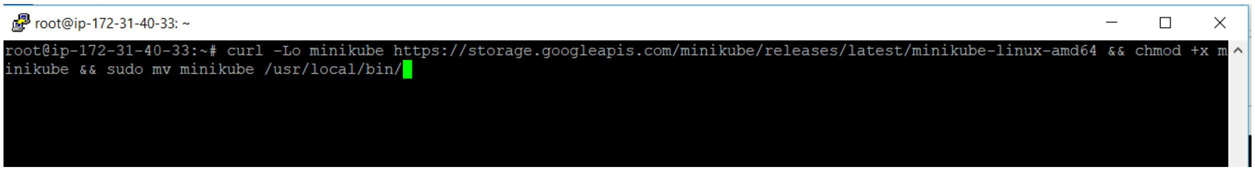
If there is no error message, please check your installation using the *docker version* command as shown below:



```
root@ip-172-31-40-33:~# docker version
Client:
 Version:      17.03.2-ce
 API version:  1.27
 Go version:   go1.6.2
 Git commit:   f5ec1e2
 Built:        Thu Jul  5 23:07:48 2018
 OS/Arch:      linux/amd64

Server:
 Version:      17.03.2-ce
 API version:  1.27 (minimum version 1.12)
 Go version:   go1.6.2
 Git commit:   f5ec1e2
 Built:        Thu Jul  5 23:07:48 2018
 OS/Arch:      linux/amd64
 Experimental: false
root@ip-172-31-40-33:~# 
```

If everything is fine, we should now install Minikube:

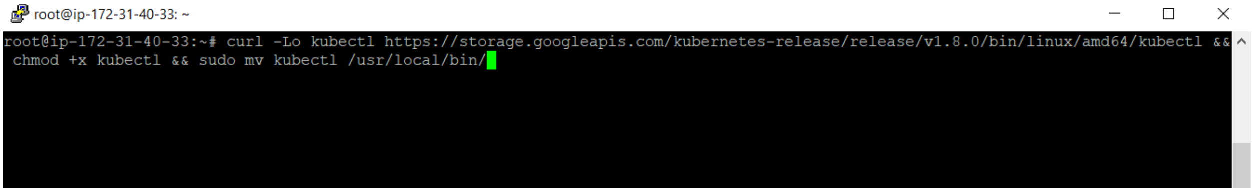
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/



```
root@ip-172-31-40-33:~# curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x m
inikube && sudo mv minikube /usr/local/bin/
```

In order to use our Kubernetes cluster on Minikube, we need the Kubernetes command line tool installed. This is kubectl. To install kubectl, run this command:

curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.10.0/bin/linux/amd64/kubectl && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl



```
root@ip-172-31-40-33:~# curl -Lo kubectl https://storage.googleapis.com/kubernetes-release/release/v1.8.0/bin/linux/amd64/kubectl &&
 chmod +x kubectl && sudo mv kubectl /usr/local/bin/
```

Next, we should install git, maven and the JDK:

apt-get install -y git

apt-get install -y maven

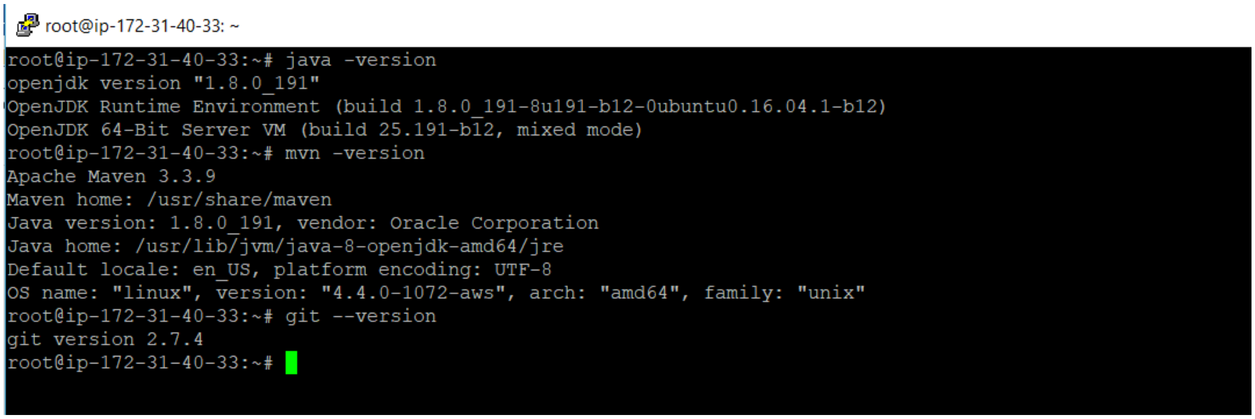
apt-get install -y default-jdk

Check that they have been installed correctly using:

java -version

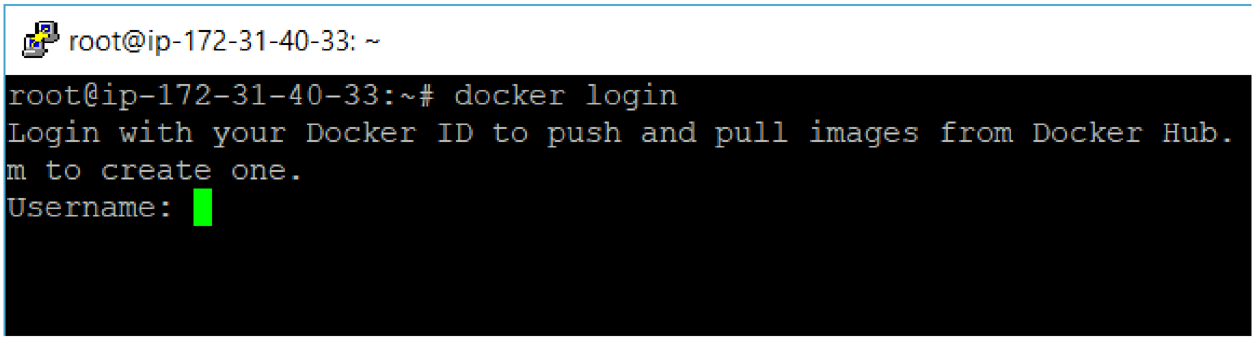
mvn -version

git --version



```
root@ip-172-31-40-33:~# java -version
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-0ubuntu0.16.04.1-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
root@ip-172-31-40-33:~# mvn -version
Apache Maven 3.3.9
Maven home: /usr/share/maven
Java version: 1.8.0_191, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.4.0-1072-aws", arch: "amd64", family: "unix"
root@ip-172-31-40-33:~# git --version
git version 2.7.4
root@ip-172-31-40-33:~#
```

At this moment we could also login to the Docker account where we will push our images.

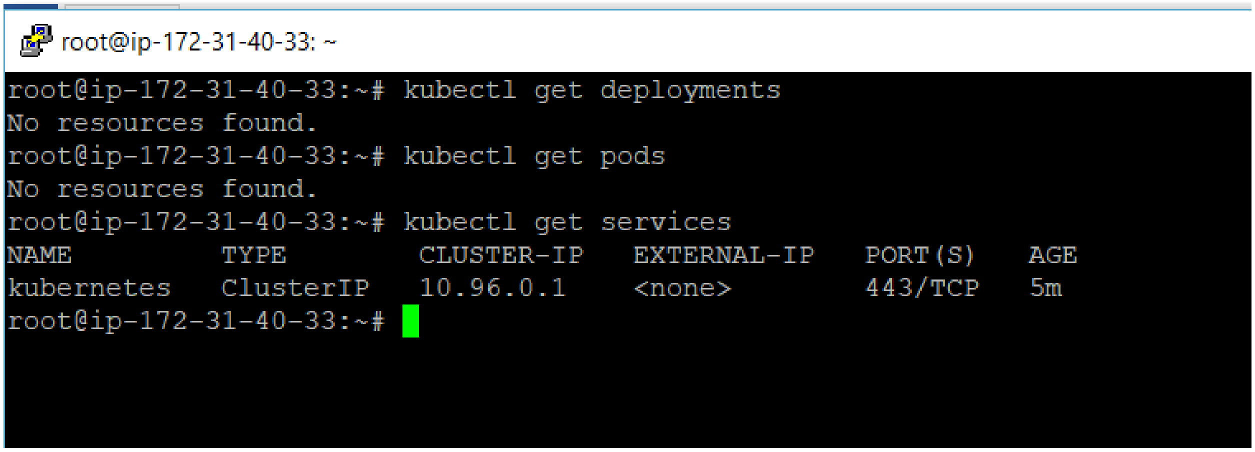


```
root@ip-172-31-40-33:~# docker login
Login with your Docker ID to push and pull images from Docker Hub.
m to create one.
Username:
```

Start minikube with this command.

minikube start --memory=8192 --cpus=4 --vm-driver=none

Then check that the service works fine by giving some commands to check the current deployments, pods and services:



```
root@ip-172-31-40-33:~# kubectl get deployments
No resources found.
root@ip-172-31-40-33:~# kubectl get pods
No resources found.
root@ip-172-31-40-33:~# kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   5m
root@ip-172-31-40-33:~#
```

During the tutorial we will need metrics, especially when dealing with horizontal auto scaling. First step is to install the metrics server on Kubernetes. In order to do this, you have to enable metrics-server, clone the following git repository and apply the files in the deploy/1.8+ folder:

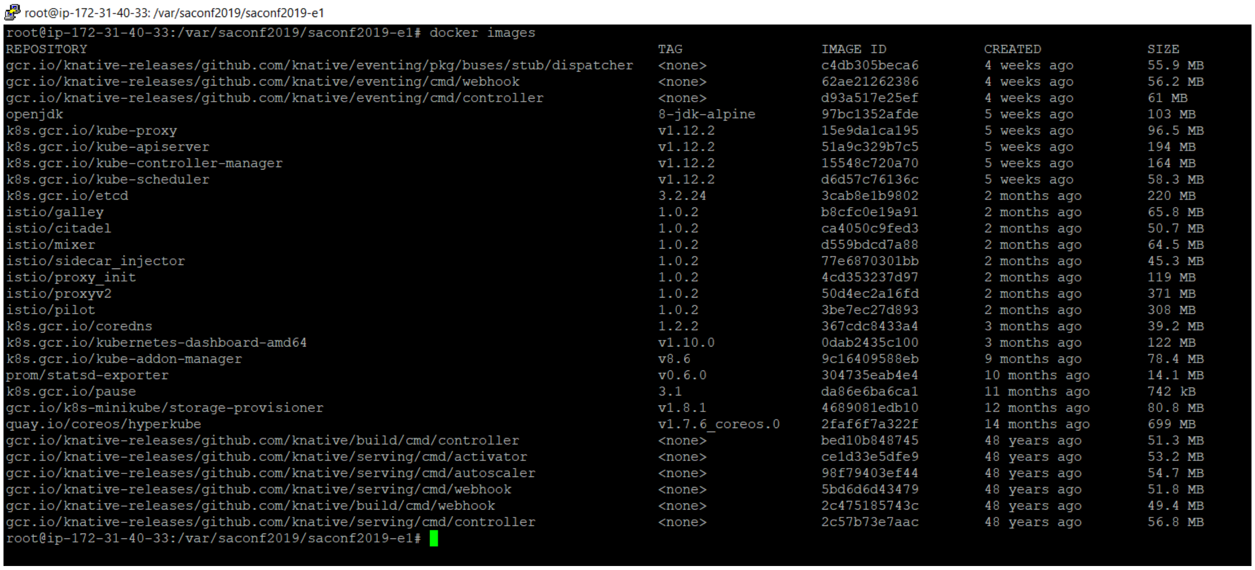
minikube addons enable metrics-server

git clone https://github.com/kubernetes-incubator/metrics-server.git

kubectl apply -f deploy/1.8+

2. Saying hello – Kubernetes components, orchestration and more…

First, we can check all *docker images.*



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

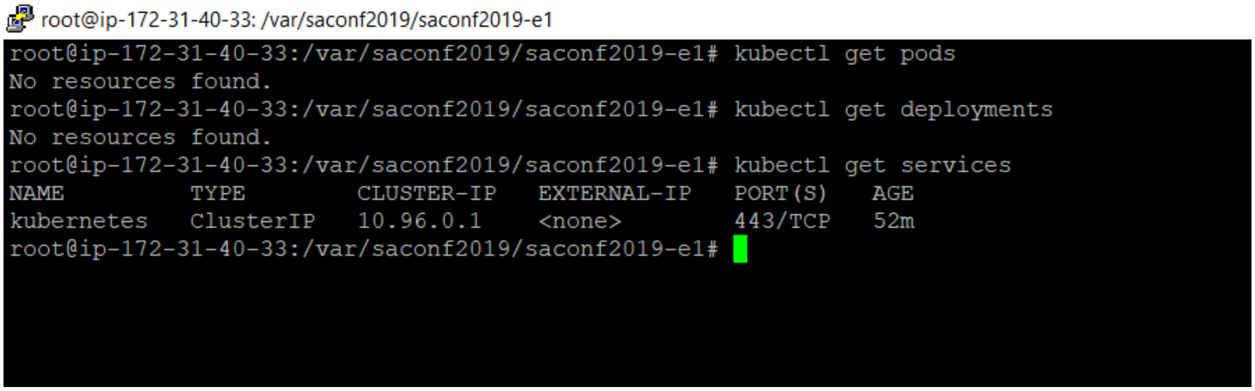
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# docker images
REPOSITORY                                                                      TAG                 IMAGE ID            CREATED             SIZE
gcr.io/knative-releases/github.com/knative/eventing/pkg/buses/stub/dispatcher   <none>              c4db305beca6        4 weeks ago         55.9 MB
gcr.io/knative-releases/github.com/knative/eventing/cmd/webhook                 <none>              62ae21262386        4 weeks ago         56.2 MB
gcr.io/knative-releases/github.com/knative/eventing/cmd/controller              <none>              d93a517e25ef        4 weeks ago         61 MB
openjdk                                                                         8-jdk-alpine        97bc1352afde        5 weeks ago         103 MB
k8s.gcr.io/kube-proxy                                                           v1.12.2             15e9da1ca195        5 weeks ago         96.5 MB
k8s.gcr.io/kube-apiserver                                                       v1.12.2             51a9c329b7c5        5 weeks ago         194 MB
k8s.gcr.io/kube-controller-manager                                              v1.12.2             15548c720a70        5 weeks ago         164 MB
k8s.gcr.io/kube-scheduler                                                       v1.12.2             d6d57c76136c        5 weeks ago         58.3 MB
k8s.gcr.io/etcd                                                                 3.2.24              3cab8e1b9802        2 months ago        220 MB
istio/galley                                                                    1.0.2               b8cfc0e19a91        2 months ago        65.8 MB
istio/citadel                                                                   1.0.2               ca4050c9fed3        2 months ago        50.7 MB
istio/mixer                                                                     1.0.2               d559bdcd7a88        2 months ago        64.5 MB
istio/sidecar_injector                                                          1.0.2               77e6870301bb        2 months ago        45.3 MB
istio/proxy_init                                                                1.0.2               4cd353237d97        2 months ago        119 MB
istio/proxyv2                                                                   1.0.2               50d4ec2a16fd        2 months ago        371 MB
istio/pilot                                                                     1.0.2               3be7ec27d893        2 months ago        308 MB
k8s.gcr.io/coredns                                                              1.2.2               367cdc8433a4        3 months ago        39.2 MB
k8s.gcr.io/kubernetes-dashboard-amd64                                           v1.10.0             0dab2435c100        3 months ago        122 MB
k8s.gcr.io/kube-addon-manager                                                   v8.6                9c16409588eb        9 months ago        78.4 MB
prom/statsd-exporter                                                            v0.6.0              304735eab4e4        10 months ago       14.1 MB
k8s.gcr.io/pause                                                                3.1                 da86e6ba6ca1        11 months ago       742 kB
gcr.io/k8s-minikube/storage-provisioner                                         v1.8.1              4689081edb10        12 months ago       80.8 MB
quay.io/coreos/hyperkube                                                        v1.7.6_coreos.0     2faf6f7a322f        14 months ago       699 MB
gcr.io/knative-releases/github.com/knative/build/cmd/controller                 <none>              bed10b848745        48 years ago        51.3 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/activator                <none>              ce1d33e5dfe9        48 years ago        53.2 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/autoscaler               <none>              98f79403ef44        48 years ago        54.7 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/webhook                  <none>              5bd6d6d43479        48 years ago        51.8 MB
gcr.io/knative-releases/github.com/knative/build/cmd/webhook                    <none>              2c475185743c        48 years ago        49.4 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/controller               <none>              2c57b73e7aac        48 years ago        56.8 MB
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1#
```

And then we can also check the deployments and the services again. On the default namespace, there should be no deployment yet and we should be able to see only the kubernetes service itself.

kubectl get pods

kubectl get deployments

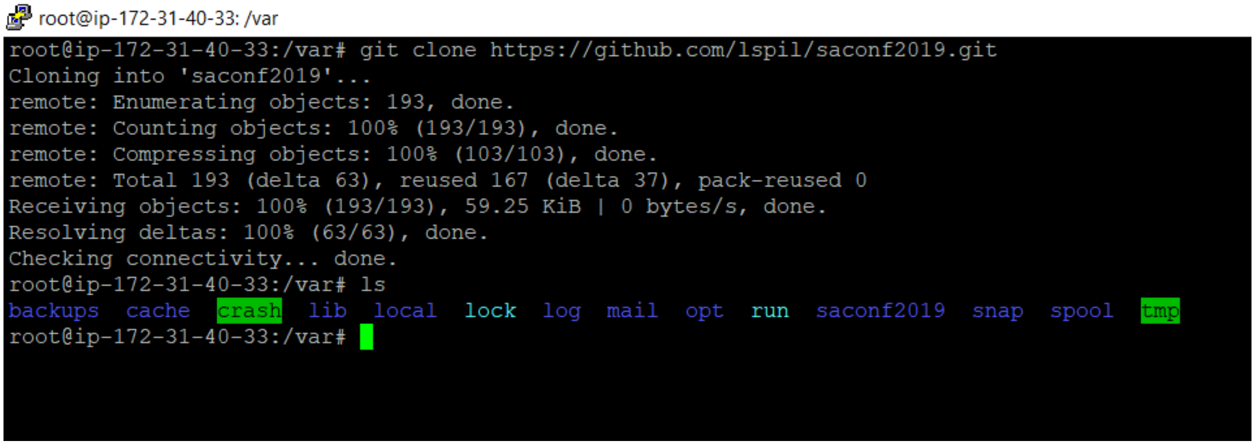
kubectl get services



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

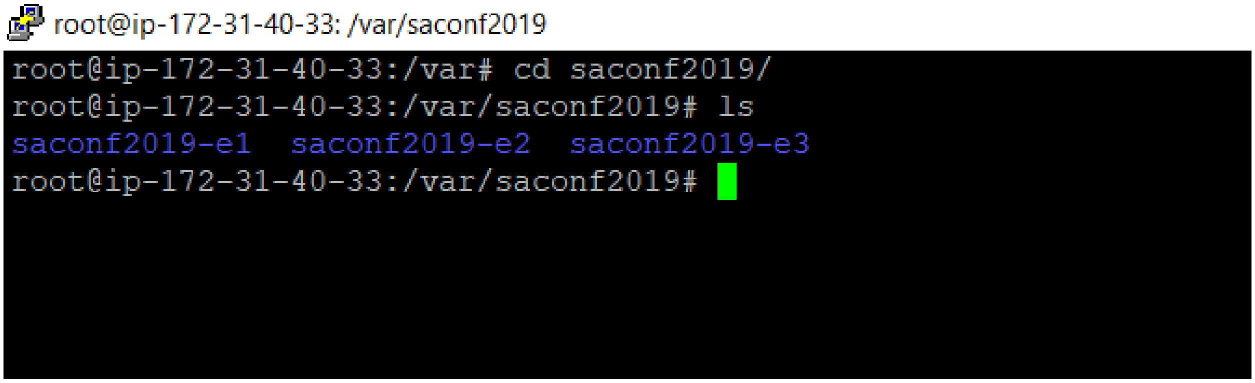
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get pods
No resources found.
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get deployments
No resources found.
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   52m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1#
```

Now, let's change the directory to the /var and clone the git repository provided:



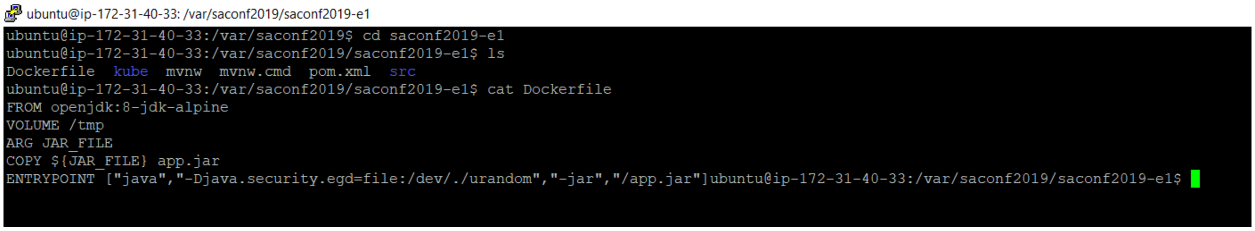
```
root@ip-172-31-40-33:/var# git clone https://github.com/lspil/saconf2019.git
Cloning into 'saconf2019'...
remote: Enumerating objects: 193, done.
remote: Counting objects: 100% (193/193), done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 193 (delta 63), reused 167 (delta 37), pack-reused 0
Receiving objects: 100% (193/193), 59.25 KiB | 0 bytes/s, done.
Resolving deltas: 100% (63/63), done.
Checking connectivity... done.
root@ip-172-31-40-33:/var# ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  saconf2019  snap  spool  tmp
root@ip-172-31-40-33:/var#
```

You can change dir now to saconf2019 where you should see all the examples:



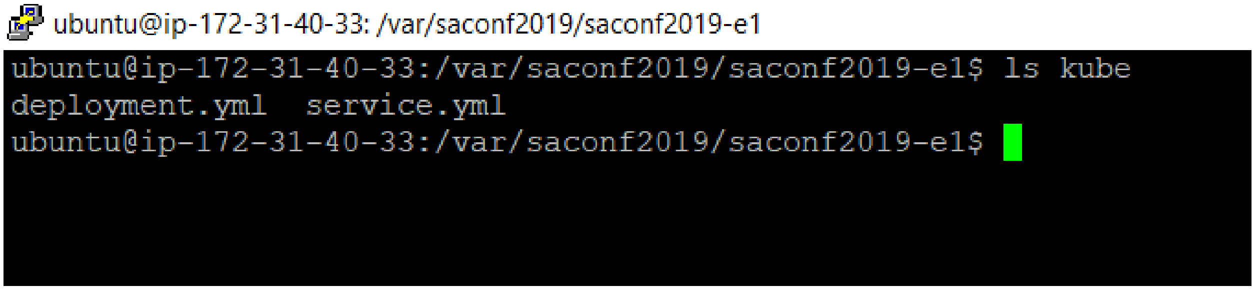
```
root@ip-172-31-40-33:/var# cd saconf2019/
root@ip-172-31-40-33:/var/saconf2019# ls
saconf2019-e1  saconf2019-e2  saconf2019-e3
root@ip-172-31-40-33:/var/saconf2019#
```

In the folder of the first example you can find a Dockerfile. The content of the Dockerfile presents a very simple setup of the image that we wish to create. The image starts from the initial layer of the openjdk 8 alpine and adds a jar file provided through a build argument. Then it starts the application.



```
ubuntu@ip-172-31-40-33:/var/saconf2019$ cd saconf2019-e1
ubuntu@ip-172-31-40-33:/var/saconf2019/saconf2019-e1$ ls
Dockerfile  kube  mvnw  mvnw.cmd  pom.xml  src
ubuntu@ip-172-31-40-33:/var/saconf2019/saconf2019-e1$ cat Dockerfile
FROM openjdk:8-jdk-alpine
VOLUME /tmp
ARG JAR_FILE
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]ubuntu@ip-172-31-40-33:/var/saconf2019/saconf2019-e1$
```

The kube folder contains the yml files with the description of the deployments, services, secrets etc.

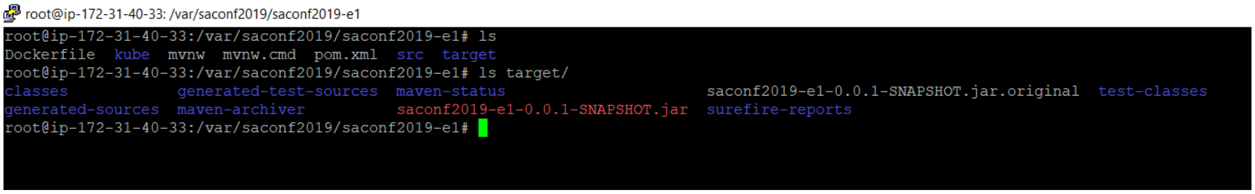


```
ubuntu@ip-172-31-40-33:/var/saconf2019/saconf2019-e1$ ls kube
deployment.yml  service.yml
ubuntu@ip-172-31-40-33:/var/saconf2019/saconf2019-e1$
```

Next step needed for running an application in Kubernetes is having the Docker image that will be used to create the running containers. Providing a Docker image, Kubernetes will create the pods according to the deployment yml file.

To create the Docker image, we will use the docker file. Running the docker build command, the only thing we need to provide as a parameter is the value of the JAR_FILE build argument. To obtain the jar file, we simply have to compile the application using Maven:

mvn clean install



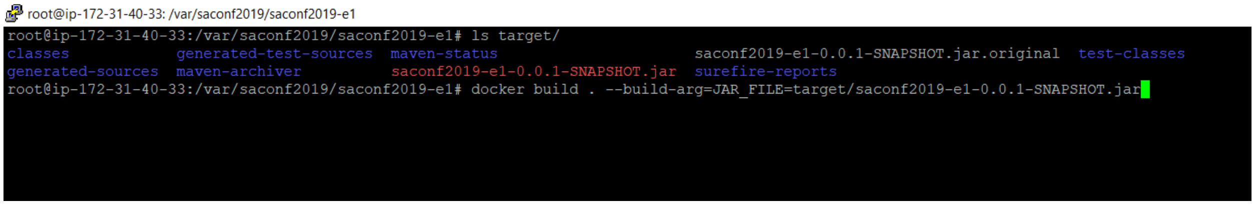
root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# ls
Dockerfile  kube  mvnw  mvnw.cmd  pom.xml  src  target
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# ls target/
classes            generated-test-sources  maven-status                      saconf2019-e1-0.0.1-SNAPSHOT.jar.original  test-classes
generated-sources  maven-archiver          saconf2019-e1-0.0.1-SNAPSHOT.jar  surefire-reports
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1#
```

After a successful build the target folder containing the fat Spring boot jar should appear.

We can use it to create the Docker image

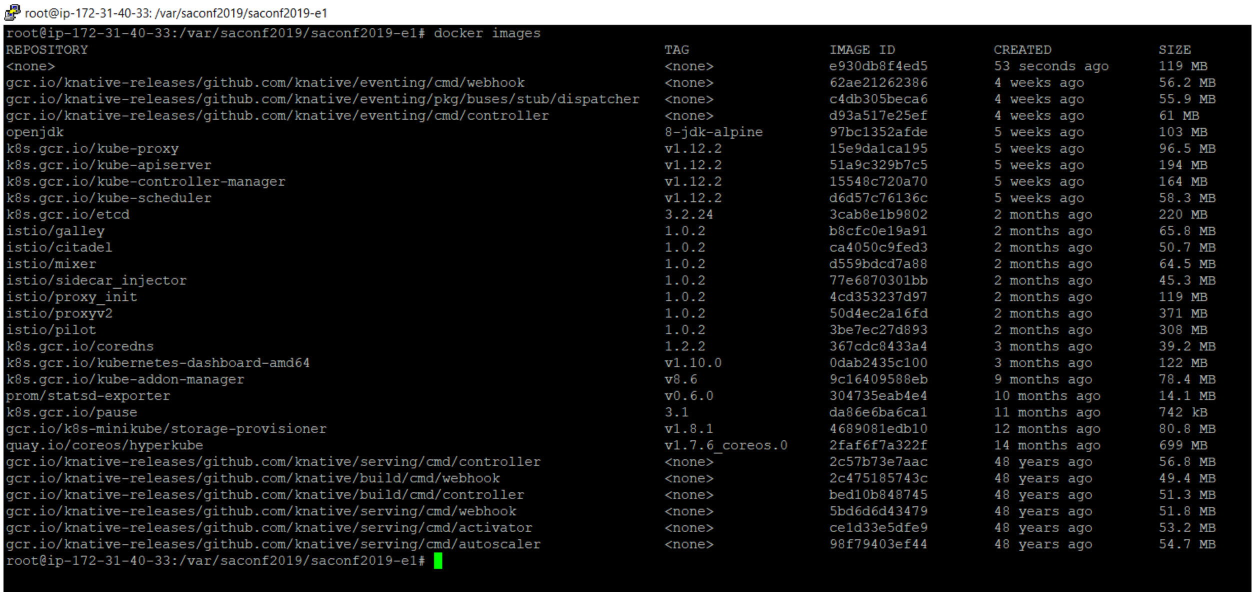
docker build . –build-arg=JAR_FILE=target/saconf2019-e1-0.0.1-SNAPSHOT.jar



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# ls target/
classes            generated-test-sources  maven-status                      saconf2019-e1-0.0.1-SNAPSHOT.jar.original  test-classes
generated-sources  maven-archiver          saconf2019-e1-0.0.1-SNAPSHOT.jar  surefire-reports
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# docker build . --build-arg=JAR_FILE=target/saconf2019-e1-0.0.1-SNAPSHOT.jar
```

Then we should be able to see it within the *docker images*

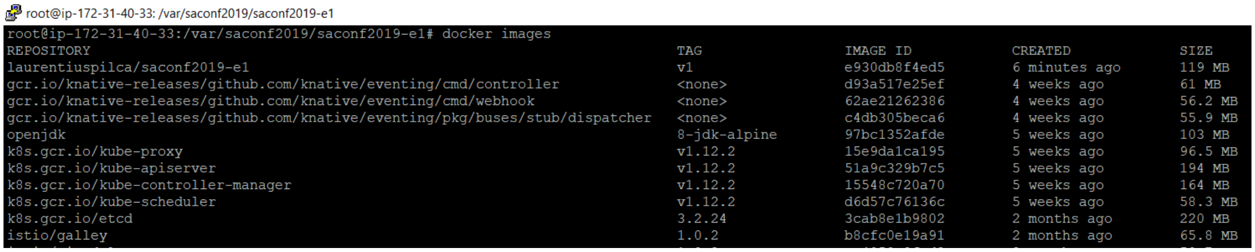


root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# docker images
REPOSITORY                                                                      TAG                 IMAGE ID            CREATED             SIZE
<none>                                                                          <none>              e930db8f4ed5        53 seconds ago      119 MB
gcr.io/knative-releases/github.com/knative/eventing/cmd/webhook                 <none>              62ae21262386        4 weeks ago         56.2 MB
gcr.io/knative-releases/github.com/knative/eventing/pkg/buses/stub/dispatcher   <none>              c4db305beca6        4 weeks ago         55.9 MB
gcr.io/knative-releases/github.com/knative/eventing/cmd/controller              <none>              d93a517e25ef        4 weeks ago         61 MB
openjdk                                                                         8-jdk-alpine        97bc1352afde        5 weeks ago         103 MB
k8s.gcr.io/kube-proxy                                                           v1.12.2             15e9da1ca195        5 weeks ago         96.5 MB
k8s.gcr.io/kube-apiserver                                                       v1.12.2             51a9c329b7c5        5 weeks ago         194 MB
k8s.gcr.io/kube-controller-manager                                              v1.12.2             15548c720a70        5 weeks ago         164 MB
k8s.gcr.io/kube-scheduler                                                       v1.12.2             d6d57c76136c        5 weeks ago         58.3 MB
k8s.gcr.io/etcd                                                                 3.2.24              3cab8e1b9802        2 months ago        220 MB
istio/galley                                                                    1.0.2               b8cfc0e19a91        2 months ago        65.8 MB
istio/citadel                                                                   1.0.2               ca4050c9fed3        2 months ago        50.7 MB
istio/mixer                                                                     1.0.2               d559bdcd7a88        2 months ago        64.5 MB
istio/sidecar_injector                                                          1.0.2               77e6870301bb        2 months ago        45.3 MB
istio/proxy_init                                                                1.0.2               4cd353237d97        2 months ago        119 MB
istio/proxyv2                                                                   1.0.2               50d4ec2a16fd        2 months ago        371 MB
istio/pilot                                                                     1.0.2               3be7ec27d893        2 months ago        308 MB
k8s.gcr.io/coredns                                                              1.2.2               367cdc8433a4        3 months ago        39.2 MB
k8s.gcr.io/kubernetes-dashboard-amd64                                           v1.10.0             0dab2435c100        3 months ago        122 MB
k8s.gcr.io/kube-addon-manager                                                   v8.6                9c16409588eb        9 months ago        78.4 MB
prom/statsd-exporter                                                            v0.6.0              304735eab4e4        10 months ago       14.1 MB
k8s.gcr.io/pause                                                                3.1                 da86e6ba6ca1        11 months ago       742 kB
gcr.io/k8s-minikube/storage-provisioner                                         v1.8.1              4689081edb10        12 months ago       80.8 MB
quay.io/coreos/hyperkube                                                        v1.7.6_coreos.0     2faf6f7a322f        14 months ago       699 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/controller               <none>              2c57b73e7aac        48 years ago        56.8 MB
gcr.io/knative-releases/github.com/knative/build/cmd/webhook                    <none>              2c475185743c        48 years ago        49.4 MB
gcr.io/knative-releases/github.com/knative/build/cmd/controller                 <none>              bed10b848745        48 years ago        51.3 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/webhook                  <none>              5bd6d6d43479        48 years ago        51.8 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/activator                <none>              ce1d33e5dfe9        48 years ago        53.2 MB
gcr.io/knative-releases/github.com/knative/serving/cmd/autoscaler               <none>              98f79403ef44        48 years ago        54.7 MB
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# 
```

The last image, created a couple of seconds ago, is currently untagged and it is the one created by the previous command. Now it can be tagged and pushed to the repository.

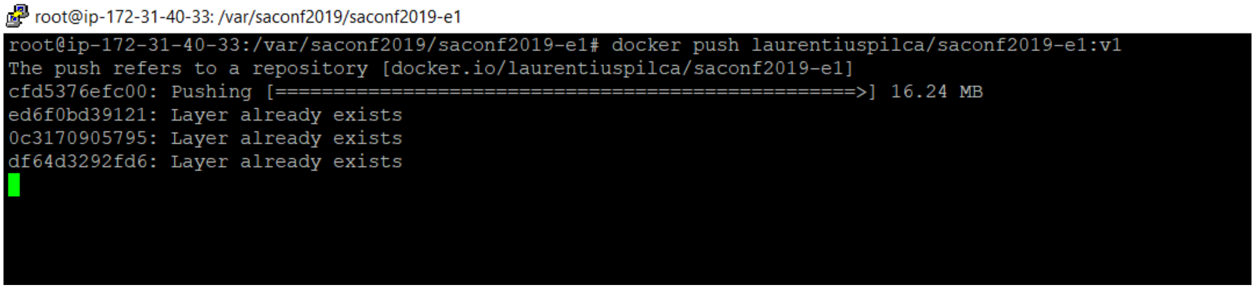
docker tag e930db8f4ed5 laurentiuspilca/saconf2019-e1:v1



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# docker images
REPOSITORY                                                                      TAG                 IMAGE ID            CREATED             SIZE
laurentiuspilca/saconf2019-e1                                                   v1                  e930db8f4ed5        6 minutes ago       119 MB
gcr.io/knative-releases/github.com/knative/eventing/cmd/controller              <none>              d93a517e25ef        4 weeks ago         61 MB
gcr.io/knative-releases/github.com/knative/eventing/cmd/webhook                 <none>              62ae21262386        4 weeks ago         56.2 MB
gcr.io/knative-releases/github.com/knative/eventing/pkg/buses/stub/dispatcher   <none>              c4db305beca6        4 weeks ago         55.9 MB
openjdk                                                                         8-jdk-alpine        97bc1352afde        5 weeks ago         103 MB
k8s.gcr.io/kube-proxy                                                           v1.12.2             15e9da1ca195        5 weeks ago         96.5 MB
k8s.gcr.io/kube-apiserver                                                       v1.12.2             51a9c329b7c5        5 weeks ago         194 MB
k8s.gcr.io/kube-controller-manager                                              v1.12.2             15548c720a70        5 weeks ago         164 MB
k8s.gcr.io/kube-scheduler                                                       v1.12.2             d6d57c76136c        5 weeks ago         58.3 MB
k8s.gcr.io/etcd                                                                 3.2.24              3cab8e1b9802        2 months ago        220 MB
istio/galley                                                                    1.0.2               b8cfc0e19a91        2 months ago        65.8 MB
```

docker push laurentiuspilca/saconf2019-e1:v1

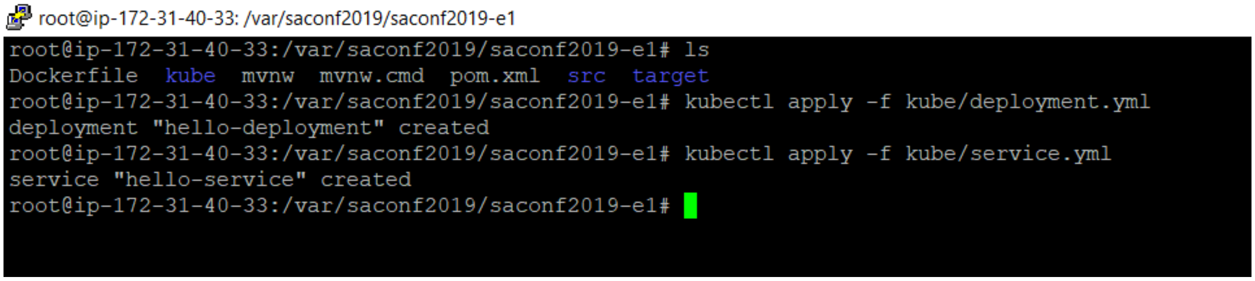


root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# docker push laurentiuspilca/saconf2019-e1:v1
The push refers to a repository [docker.io/laurentiuspilca/saconf2019-e1]
cfd5376efc00: Pushing [==================================================>] 16.24 MB
ed6f0bd39121: Layer already exists
0c3170905795: Layer already exists
df64d3292fd6: Layer already exists
```

kubectl apply –f target/deployment.yml

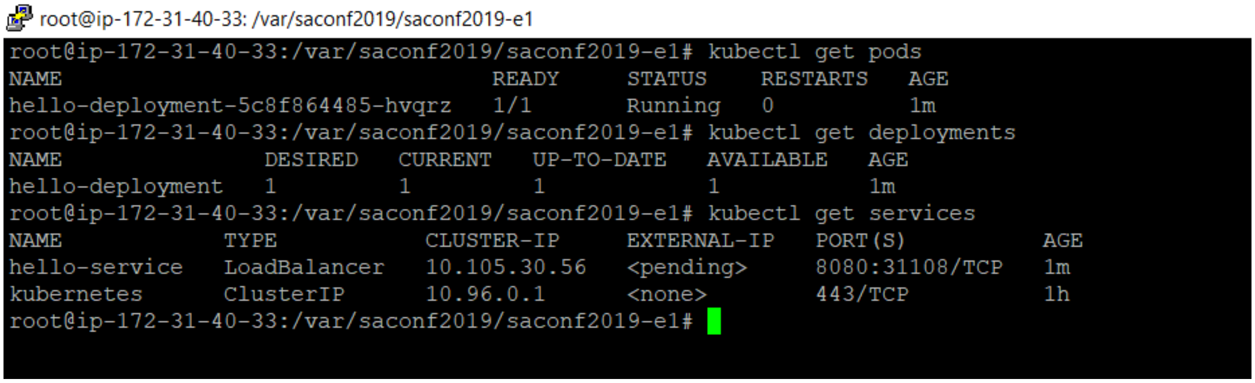
kubectl apply –f target/service.yml



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# ls
Dockerfile  kube  mvnw  mvnw.cmd  pom.xml  src  target
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl apply -f kube/deployment.yml
deployment "hello-deployment" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl apply -f kube/service.yml
service "hello-service" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1#
```

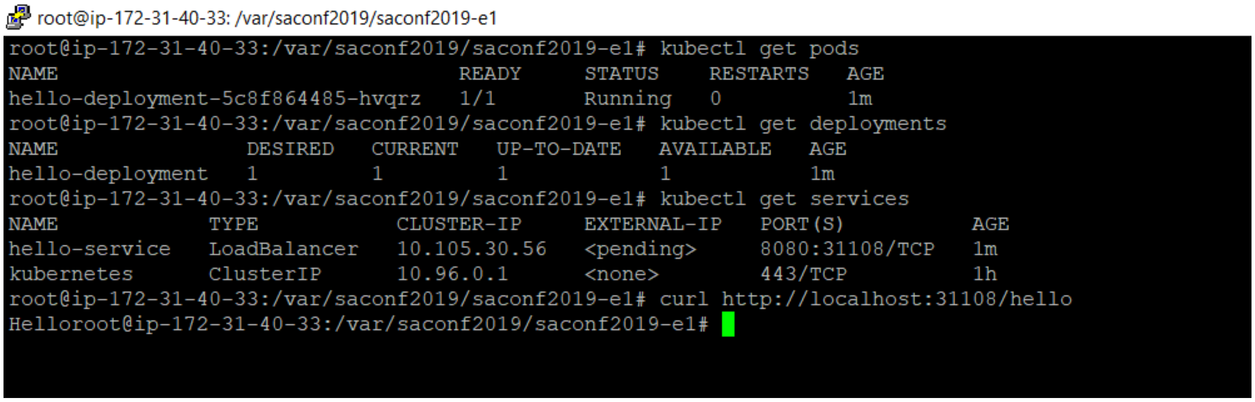
Let's check again the deployment, pods, and services.



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get pods
NAME                                READY     STATUS    RESTARTS   AGE
hello-deployment-5c8f864485-hvqrz   1/1       Running   0          1m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get deployments
NAME               DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-deployment   1         1         1            1           1m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get services
NAME            TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-service   LoadBalancer   10.105.30.56   <pending>     8080:31108/TCP   1m
kubernetes      ClusterIP      10.96.0.1      <none>        443/TCP          1h
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1#
```

curl http://localhost:port/hello



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e1

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get pods
NAME                                READY     STATUS    RESTARTS   AGE
hello-deployment-5c8f864485-hvqrz   1/1       Running   0          1m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get deployments
NAME               DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-deployment   1         1         1            1           1m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# kubectl get services
NAME            TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-service   LoadBalancer   10.105.30.56   <pending>     8080:31108/TCP   1m
kubernetes      ClusterIP      10.96.0.1      <none>        443/TCP          1h
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e1# curl http://localhost:31108/hello
Helloroot@ip-172-31-40-33:/var/saconf2019/saconf2019-e1#
```

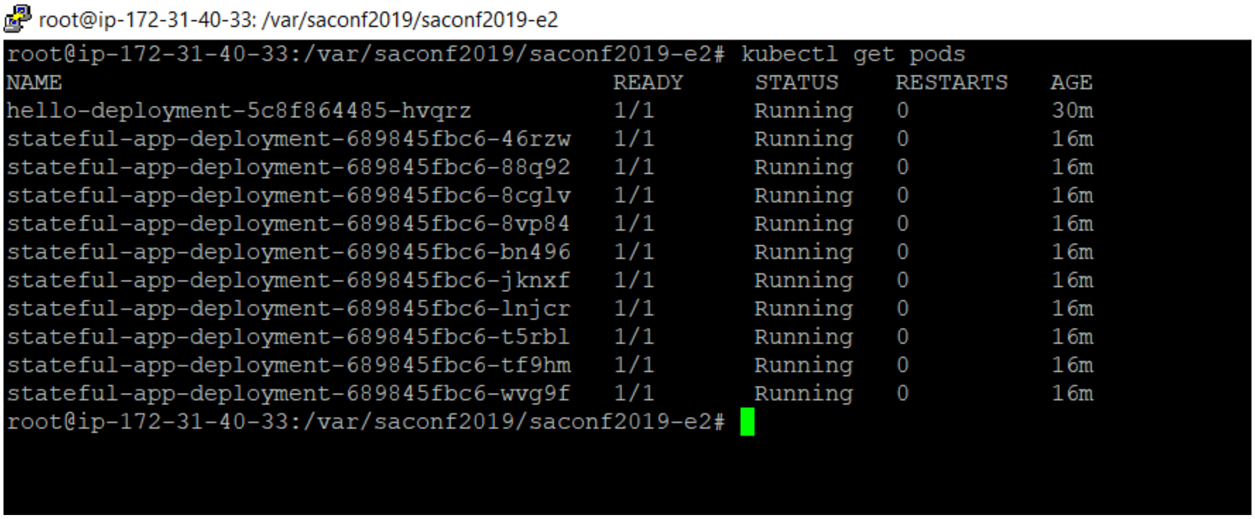
3. Statefulness – the first evil of all microservices architectures

To run the second example simply assume the Docker images are there. Just apply the deployment and the service.

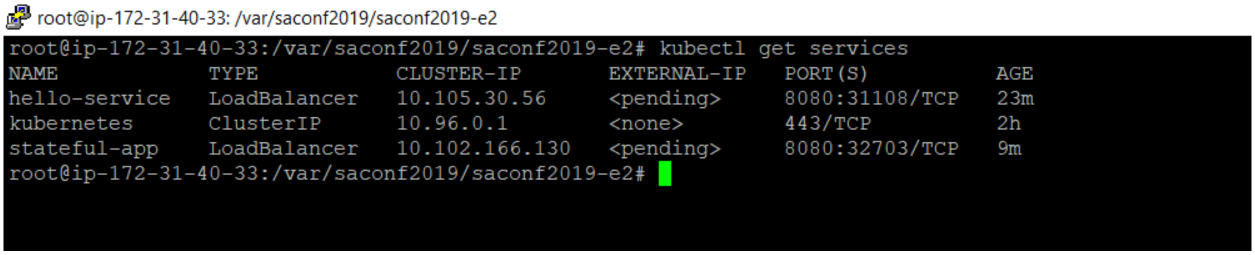
kubectl apply -f target/deployment.yml

kubectl apply -f target/service.yml

After running the commands you will see 10 replicas of the same pod starting. You can play with the number of replicas to make it smaller or bigger by changing the replicas parameter in the deployment.yml file.



```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e2# kubectl get pods
NAME                                       READY     STATUS    RESTARTS   AGE
hello-deployment-5c8f864485-hvqrz          1/1       Running   0          30m
stateful-app-deployment-689845fbc6-46rzw   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-88q92   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-8cglv   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-8vp84   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-bn496   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-jknxf   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-lnjcr   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-t5rbl   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-tf9hm   1/1       Running   0          16m
stateful-app-deployment-689845fbc6-wvg9f   1/1       Running   0          16m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e2#
```



```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e2# kubectl get services
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-service   LoadBalancer   10.105.30.56     <pending>     8080:31108/TCP   23m
kubernetes      ClusterIP      10.96.0.1        <none>        443/TCP          2h
stateful-app    LoadBalancer   10.102.166.130   <pending>     8080:32703/TCP   9m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e2#
```

The second example has two endpoints. One of them is setting a name. The other is returning hello to that name. The name is stored in the state of the container. Let’s see what happens when we set a name and then say hello multiple times using more than one replica.

curl -XPOST http://localhost:32703/name/John

curl http://localhost:32703/hello

The observation should state the conclusion that keeping state on the application will result in keeping state only on one pod. But, as we want an architecture horizontally scalable this becomes a problem. When we scale our pods to multiple instances, requests are now spread over the replicas. This means that two consecutive requests might not reach the same pod.

4. Privacy – mind your secrets

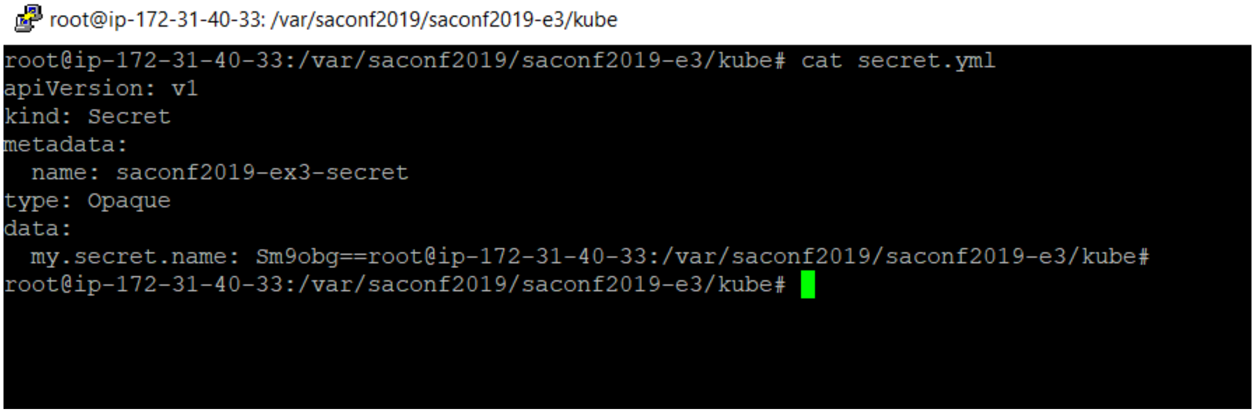
When we deploy the system using an orchestration tool in a cloud it is always important to know where to keep the sensitive data like users, password or encryption keys used by the deployed applications.

In Kubernetes we keep such data in secrets. In this case, to be easier to use, the secret is also defined as a yml file. But you would not do this in a real case scenario and of course neither you should store them in git.

In the following example you can see one way in which the application can read the secret and use it.

cat secret.yml

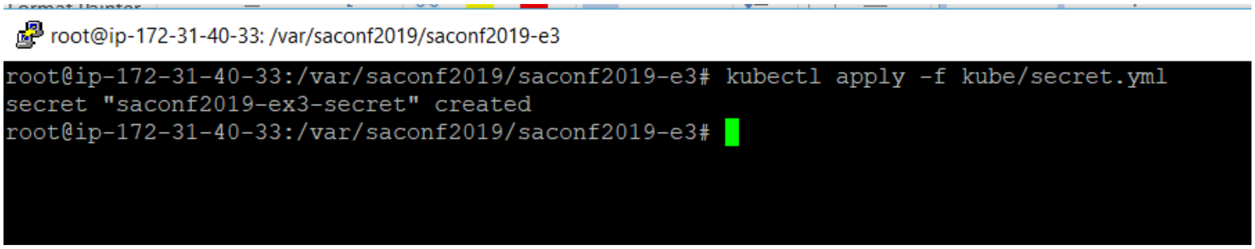
Observe in the yml file that the value is base 64 encoded. Copy the value and decode it.



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e3/kube

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3/kube# cat secret.yml
apiVersion: v1
kind: Secret
metadata:
  name: saconf2019-ex3-secret
type: Opaque
data:
  my.secret.name: Sm9obg==root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3/kube#
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3/kube#
```

kubectl apply -f kube/secret.yml

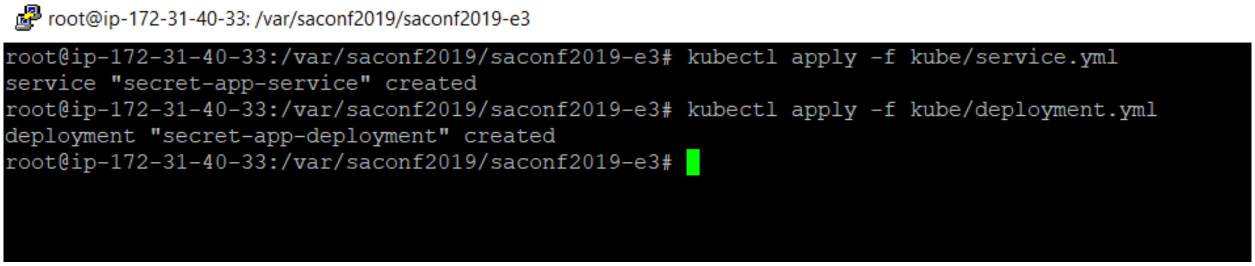


```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3# kubectl apply -f kube/secret.yml
secret "saconf2019-ex3-secret" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3#
```

Once the secret is created, apply the yml configuration for deployment and service.

kubectl apply -f kube/service.yml

kubectl apply -f kube/deployment.yml

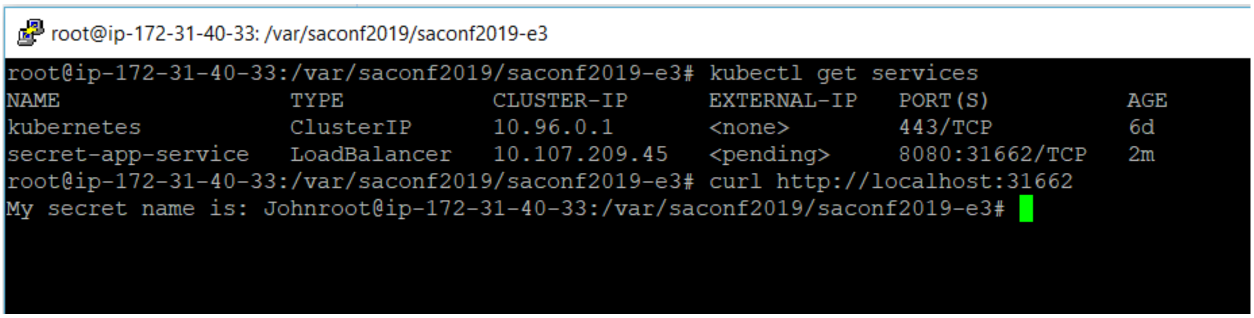


```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3# kubectl apply -f kube/service.yml
service "secret-app-service" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3# kubectl apply -f kube/deployment.yml
deployment "secret-app-deployment" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3#
```

kubectl get services

After calling the endpoint in the application, you can see that the value of the secret is used.

curl http://localhost:<port>



```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3# kubectl get services
NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes           ClusterIP      10.96.0.1       <none>        443/TCP          6d
secret-app-service   LoadBalancer   10.107.209.45   <pending>     8080:31662/TCP   2m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e3# curl http://localhost:31662
My secret name is: Johnroot@ip-172-31-40-33:/var/saconf2019/saconf2019-e3#
```

5. The undo rollout – or how to deal with the dependencies at undo rollout

When thinking about the architecture of the system, it is always important to understand what happens with the dependencies at an update. An orchestration tool like Kubernetes allows us to rollout undo the deployment. Because orchestration is “playing” with images, we store the last snapshot of our application so it is fairly easy to undo to an earlier point. However, depending on the architecture of the system we must understand if rollout and undo of the java code is just enough.

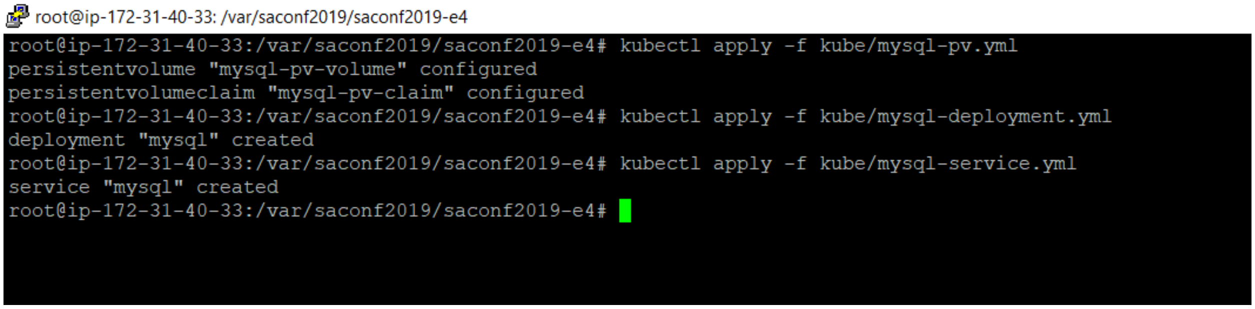
In the below example you find a situation in which the rollout undo is not enough.

We start by deploying a mysql server.

kubectl apply -f kube/mysql-pv.yml

kubectl apply -f kube/mysql-deployment.yml

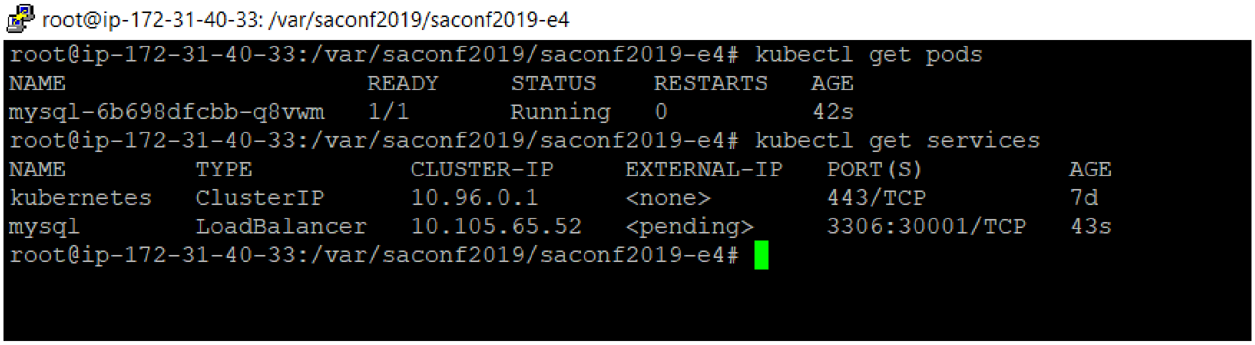
kubectl apply -f kube/mysql-service.yml



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/mysql-pv.yml
persistentvolume "mysql-pv-volume" configured
persistentvolumeclaim "mysql-pv-claim" configured
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/mysql-deployment.yml
deployment "mysql" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/mysql-service.yml
service "mysql" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```

kubectl get pods



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

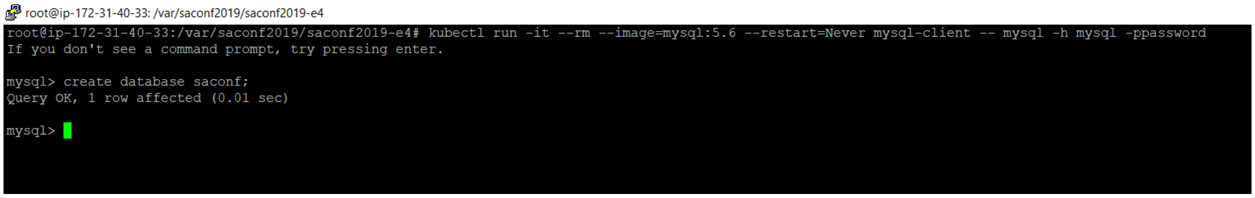
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl get pods
NAME                     READY     STATUS    RESTARTS   AGE
mysql-6b698dfcbb-q8vwm   1/1       Running   0          42s
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP      10.96.0.1       <none>        443/TCP          7d
mysql        LoadBalancer   10.105.65.52    <pending>     3306:30001/TCP   43s
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```

Let’s now use a client to connect to the server.

kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql -ppassword

So we can create a database which we connect to.

create database saconf;



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql -ppassword
If you don't see a command prompt, try pressing enter.

mysql> create database saconf;
Query OK, 1 row affected (0.01 sec)

mysql>
```

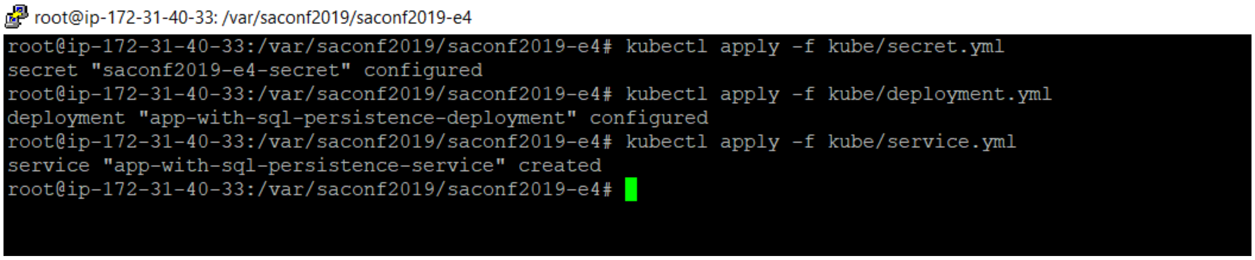
Check the secret.yml file. You might need to change the cluster IP address which you connect to.

Then deploy the application and check that the connection works properly.

kubectl apply -f kube/secret.yml

kubectl apply -f kube/deployment.yml

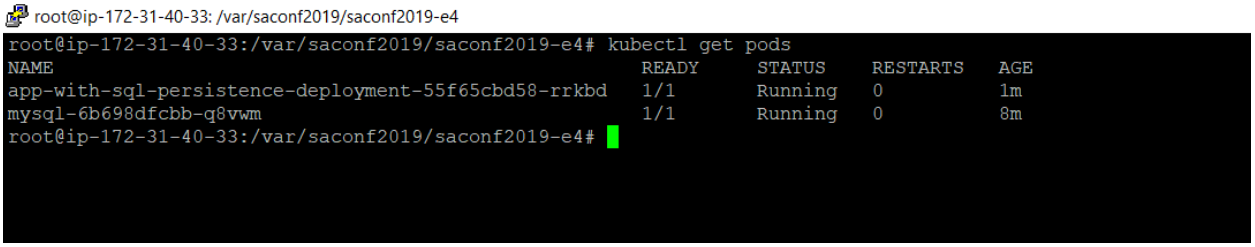
kubectl apply -f kube/service.yml



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/secret.yml
secret "saconf2019-e4-secret" configured
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/deployment.yml
deployment "app-with-sql-persistence-deployment" configured
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/service.yml
service "app-with-sql-persistence-service" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```

kubectl get pods

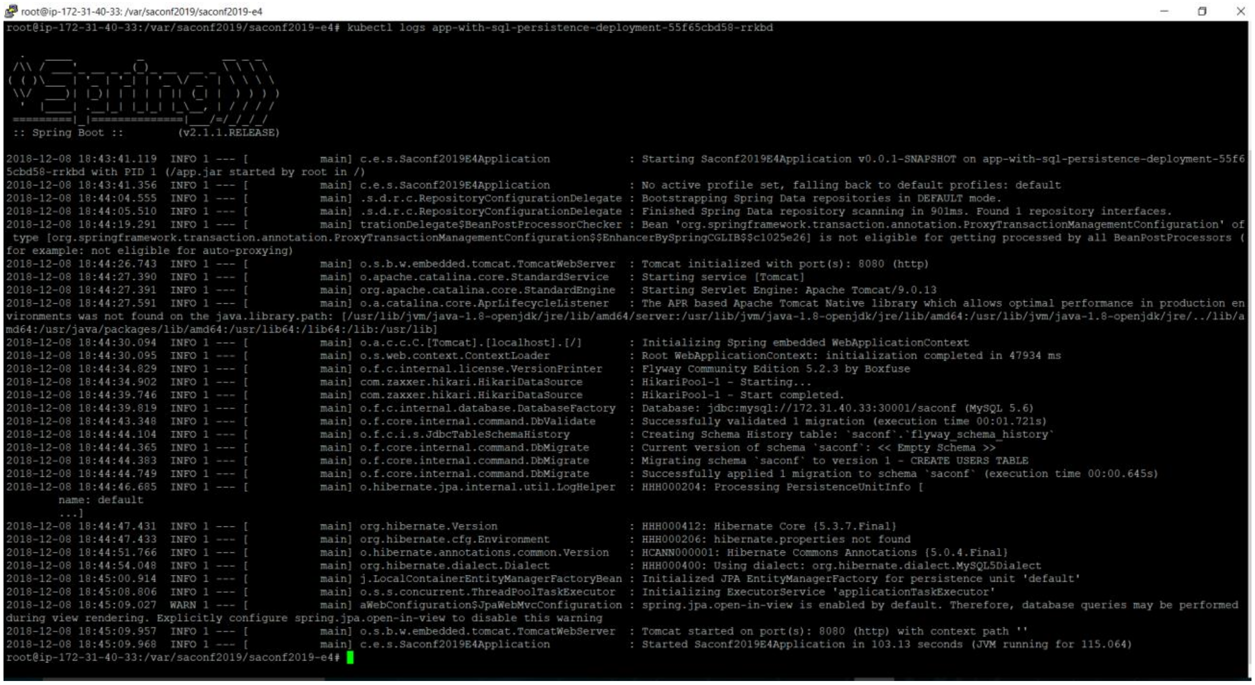


root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl get pods
NAME                                                READY     STATUS    RESTARTS   AGE
app-with-sql-persistence-deployment-55f65cbd58-rrkbd   1/1       Running   0          1m
mysql-6b698dfcbb-q8vwm                               1/1       Running   0          8m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# 
```

Display logs:

kubectl logs <name_of_the_pod>



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

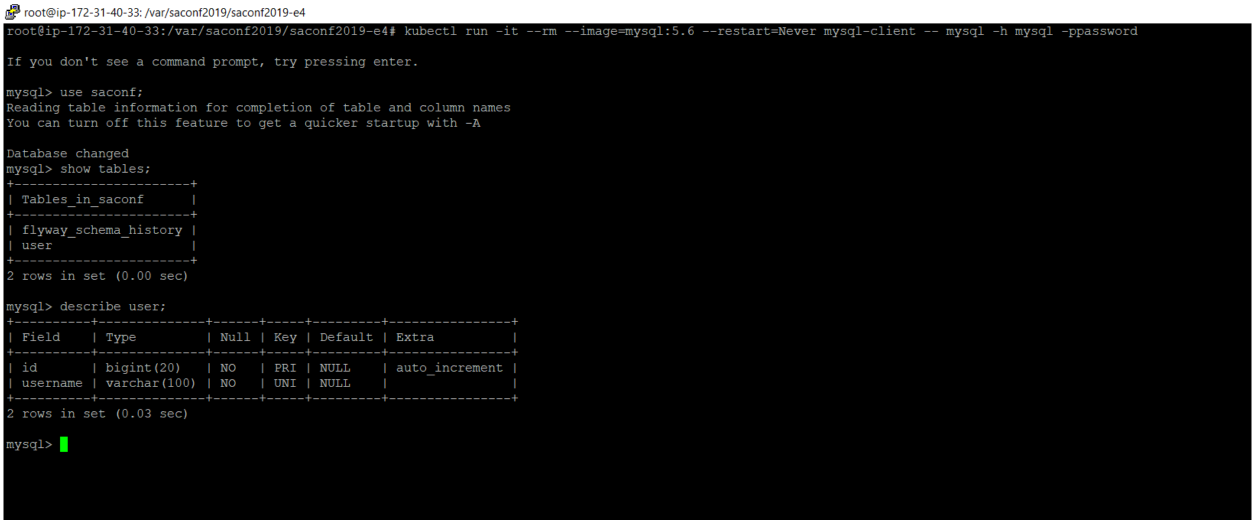
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl logs app-with-sql-persistence-deployment-55f65cbd58-rrkbd

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.1.RELEASE)

2018-12-08 18:43:41.119  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : Starting Saconf2019E4Application v0.0.1-SNAPSHOT on app-with-sql-persistence-deployment-55f6
5cbd58-rrkbd with PID 1 (/app.jar started by root in /)
2018-12-08 18:43:41.356  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : No active profile set, falling back to default profiles: default
2018-12-08 18:44:04.555  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mode.
2018-12-08 18:44:05.510  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 901ms. Found 1 repository interfaces.
2018-12-08 18:44:19.291  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of
 type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIB$$c1025e26] is not eligible for getting processed by all BeanPostProcessors (
for example: not eligible for auto-proxying)
2018-12-08 18:44:26.743  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2018-12-08 18:44:27.390  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2018-12-08 18:44:27.391  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/9.0.13
2018-12-08 18:44:27.591  INFO 1 --- [           main] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance in production en
vironments was not found on the java.library.path: [/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64/server:/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64:/usr/lib/jvm/java-1.8-openjdk/jre/../lib/a
md64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]
2018-12-08 18:44:30.094  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2018-12-08 18:44:30.095  INFO 1 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 47934 ms
2018-12-08 18:44:34.829  INFO 1 --- [           main] o.f.c.internal.license.VersionPrinter    : Flyway Community Edition 5.2.3 by Boxfuse
2018-12-08 18:44:34.902  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2018-12-08 18:44:39.746  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2018-12-08 18:44:39.819  INFO 1 --- [           main] o.f.c.internal.database.DatabaseFactory  : Database: jdbc:mysql://172.31.40.33:30001/saconf (MySQL 5.6)
2018-12-08 18:44:43.348  INFO 1 --- [           main] o.f.core.internal.command.DbValidate     : Successfully validated 1 migration (execution time 00:01.721s)
2018-12-08 18:44:44.104  INFO 1 --- [           main] o.f.c.i.s.JdbcTableSchemaHistory         : Creating Schema History table: `saconf`.`flyway_schema_history`
2018-12-08 18:44:44.365  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Current version of schema `saconf`: << Empty Schema >>
2018-12-08 18:44:44.383  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Migrating schema `saconf` to version 1 - CREATE USERS TABLE
2018-12-08 18:44:44.749  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Successfully applied 1 migration to schema `saconf` (execution time 00:00.645s)
2018-12-08 18:44:46.685  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [
        name: default
        ...]
2018-12-08 18:44:47.431  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate Core {5.3.7.Final}
2018-12-08 18:44:47.433  INFO 1 --- [           main] org.hibernate.cfg.Environment            : HHH000206: hibernate.properties not found
2018-12-08 18:44:51.766  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
2018-12-08 18:44:54.048  INFO 1 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
2018-12-08 18:45:00.914  INFO 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2018-12-08 18:45:08.806  INFO 1 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2018-12-08 18:45:09.027  WARN 1 --- [           main] aWebConfiguration$JpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed
during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2018-12-08 18:45:09.957  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2018-12-08 18:45:09.968  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : Started Saconf2019E4Application in 103.13 seconds (JVM running for 115.064)
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# 
```

kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql -ppassword

use saconf;

show tables;

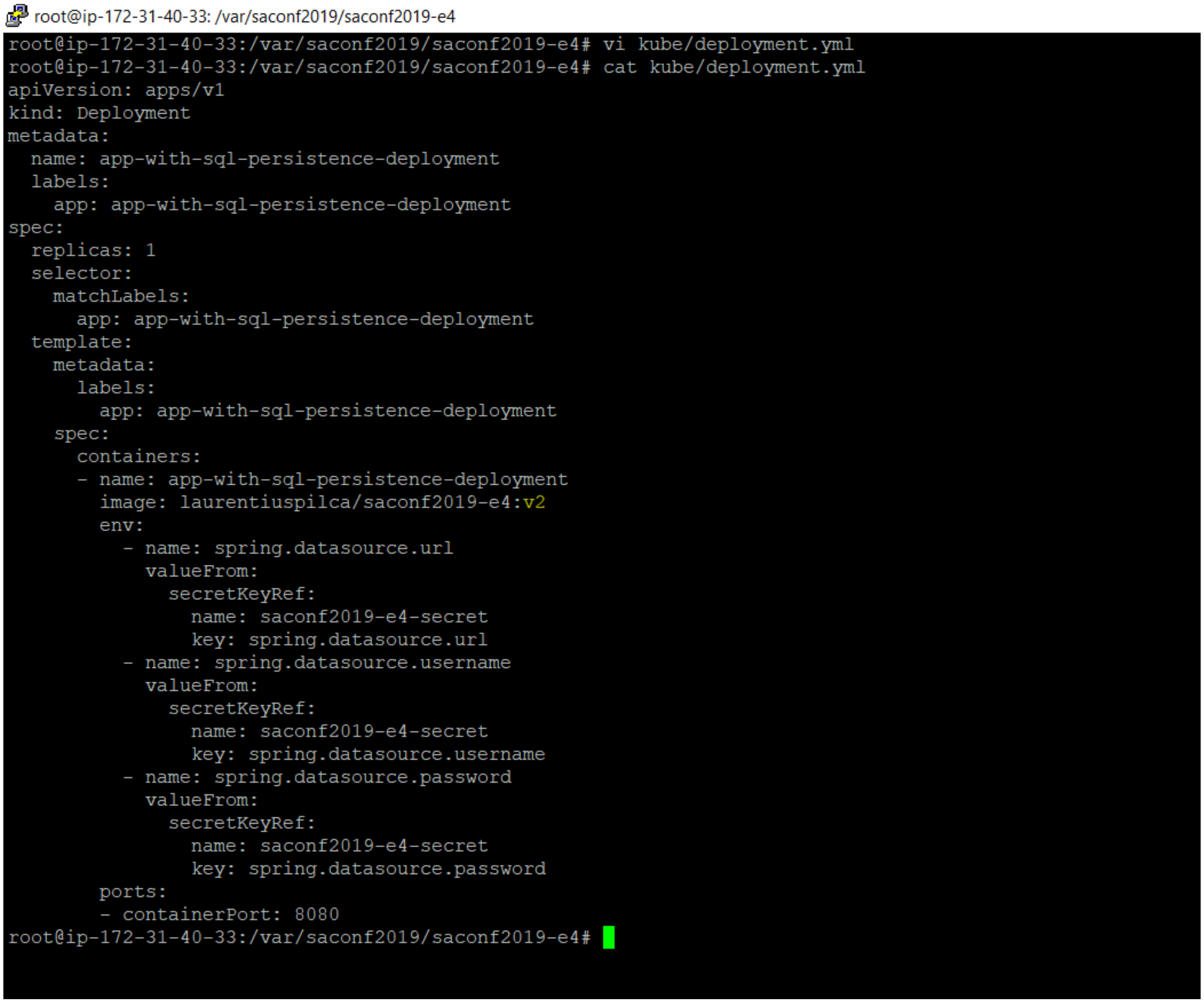
describe user;



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql -ppassword

If you don't see a command prompt, try pressing enter.

mysql> use saconf;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_saconf      |
+-----------------------+
| flyway_schema_history |
| user                  |
+-----------------------+
2 rows in set (0.00 sec)

mysql> describe user;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| username | varchar(100) | NO   | UNI | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
2 rows in set (0.03 sec)

mysql> 
```

Use vi to change the version number in the deployment file to upgrade to version v2.

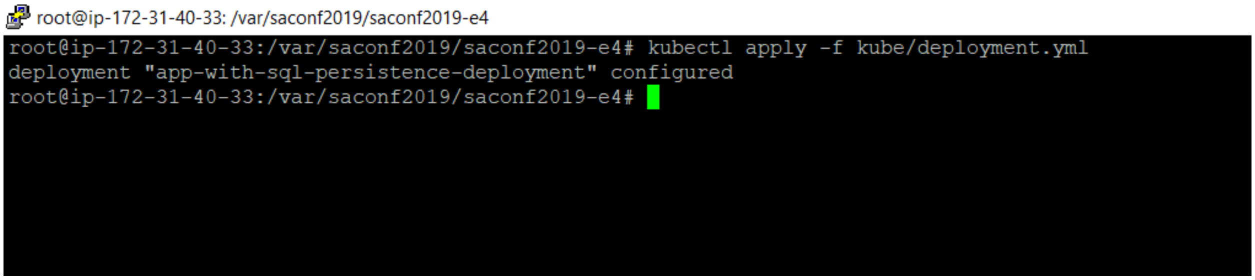


root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# vi kube/deployment.yml
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# cat kube/deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-with-sql-persistence-deployment
  labels:
    app: app-with-sql-persistence-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: app-with-sql-persistence-deployment
  template:
    metadata:
      labels:
        app: app-with-sql-persistence-deployment
    spec:
      containers:
      - name: app-with-sql-persistence-deployment
        image: laurentiuspilca/saconf2019-e4:v2
        env:
          - name: spring.datasource.url
            valueFrom:
              secretKeyRef:
                name: saconf2019-e4-secret
                key: spring.datasource.url
          - name: spring.datasource.username
            valueFrom:
              secretKeyRef:
                name: saconf2019-e4-secret
                key: spring.datasource.username
          - name: spring.datasource.password
            valueFrom:
              secretKeyRef:
                name: saconf2019-e4-secret
                key: spring.datasource.password
        ports:
        - containerPort: 8080
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# 
```

Apply again the deployment:

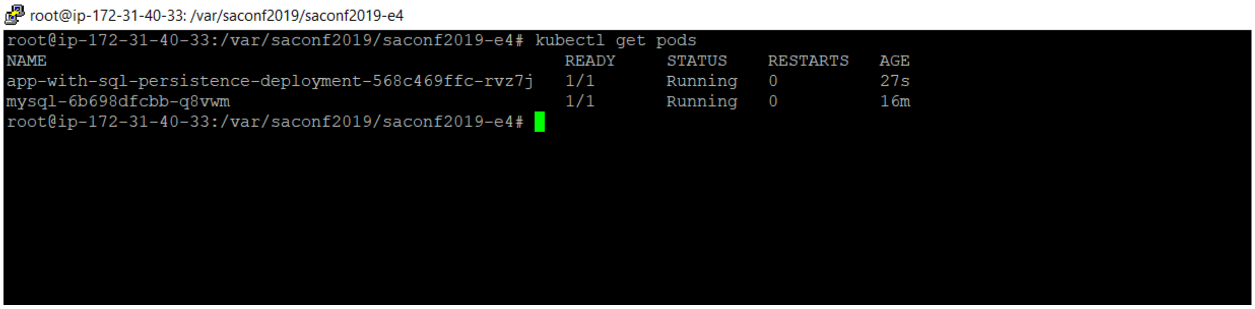
kubectl apply -f kube/deployment.yml



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

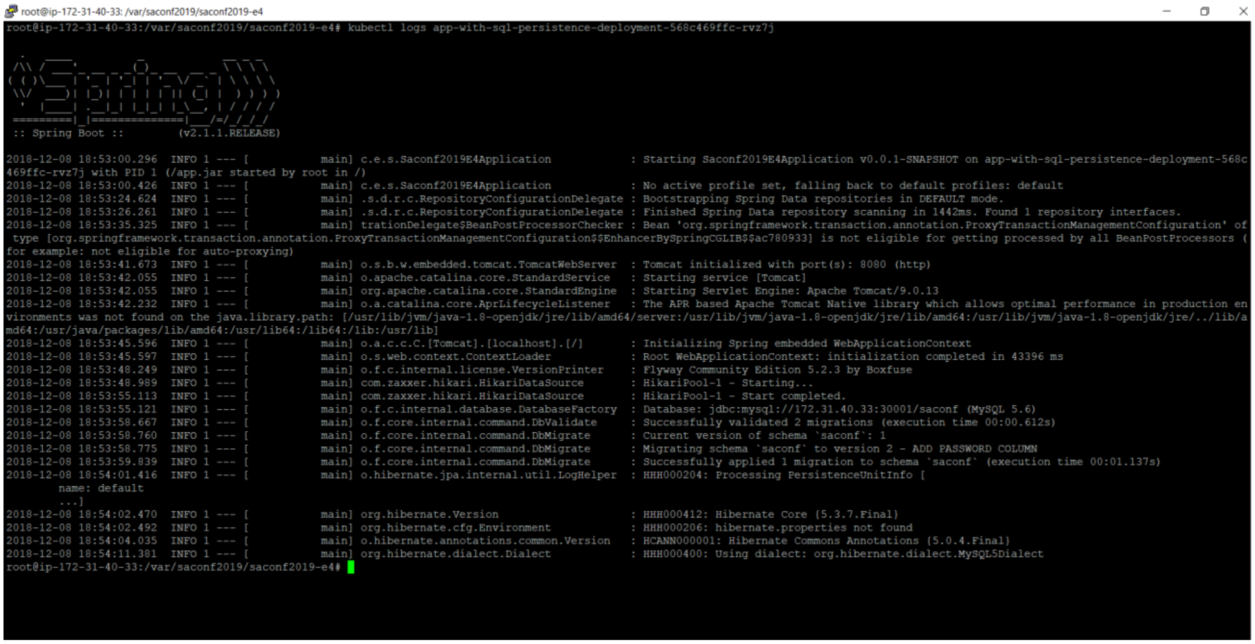
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl apply -f kube/deployment.yml
deployment "app-with-sql-persistence-deployment" configured
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```

Check the pods:



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

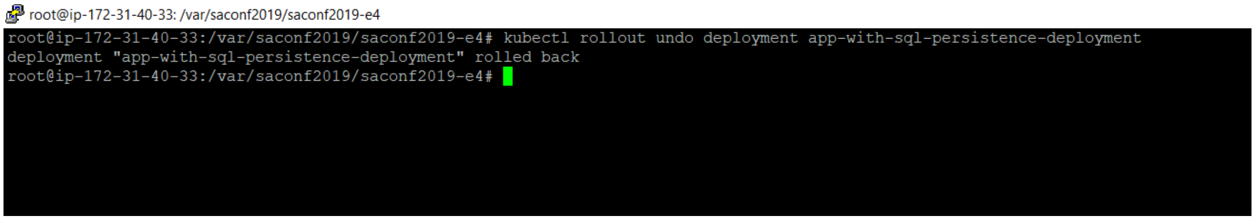
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl get pods
NAME                                                  READY     STATUS    RESTARTS   AGE
app-with-sql-persistence-deployment-568c469ffc-rvz7j   1/1       Running   0          27s
mysql-6b698dfcbb-q8vwm                                1/1       Running   0          16m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```



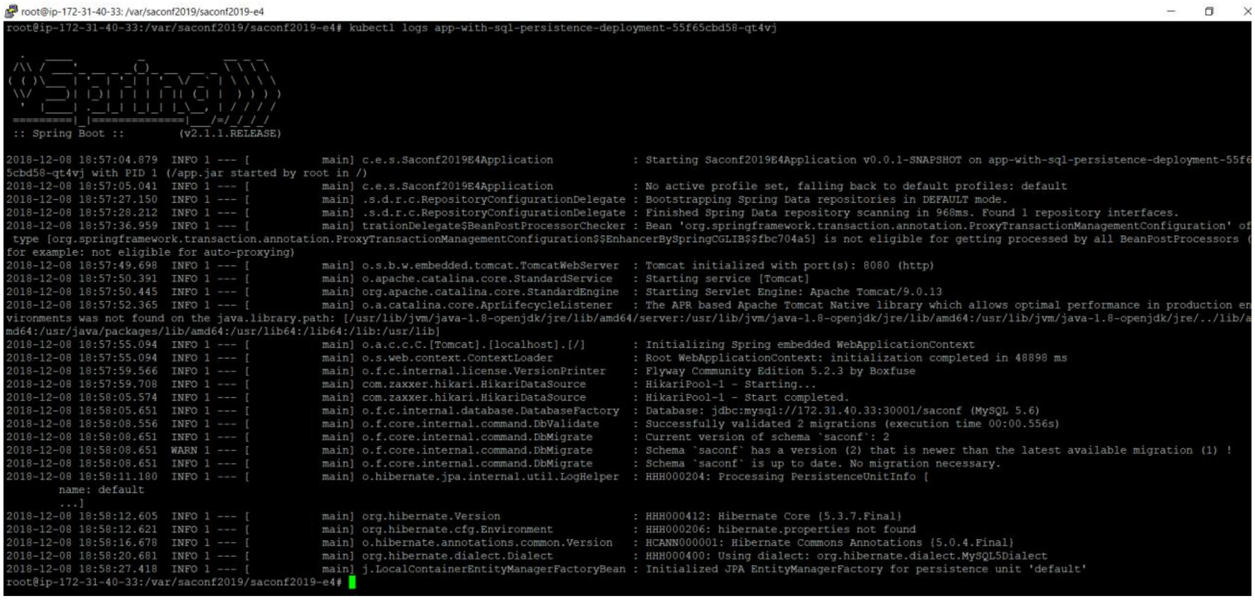
root@ip-172-31-40-33: /var/saconf2019/saconf2019-e4

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl logs app-with-sql-persistence-deployment-568c469ffc-rvz7j

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.1.RELEASE)

2018-12-08 18:53:00.296  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : Starting Saconf2019E4Application v0.0.1-SNAPSHOT on app-with-sql-persistence-deployment-568c
469ffc-rvz7j with PID 1 (/app.jar started by root in /)
2018-12-08 18:53:00.426  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : No active profile set, falling back to default profiles: default
2018-12-08 18:53:24.624  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mode.
2018-12-08 18:53:26.261  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 1442ms. Found 1 repository interfaces.
2018-12-08 18:53:35.325  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of
 type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIB$$ac780933] is not eligible for getting processed by all BeanPostProcessors (
for example: not eligible for auto-proxying)
2018-12-08 18:53:41.673  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2018-12-08 18:53:42.055  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2018-12-08 18:53:42.055  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/9.0.13
2018-12-08 18:53:42.232  INFO 1 --- [           main] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance in production en
vironments was not found on the java.library.path: [/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64/server:/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64:/usr/lib/jvm/java-1.8-openjdk/jre/../lib/a
md64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]
2018-12-08 18:53:45.596  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2018-12-08 18:53:45.597  INFO 1 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 43396 ms
2018-12-08 18:53:48.249  INFO 1 --- [           main] o.f.c.internal.license.VersionPrinter    : Flyway Community Edition 5.2.3 by Boxfuse
2018-12-08 18:53:48.989  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2018-12-08 18:53:55.113  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2018-12-08 18:53:55.121  INFO 1 --- [           main] o.f.c.internal.database.DatabaseFactory  : Database: jdbc:mysql://172.31.40.33:30001/saconf (MySQL 5.6)
2018-12-08 18:53:58.667  INFO 1 --- [           main] o.f.core.internal.command.DbValidate     : Successfully validated 2 migrations (execution time 00:00.612s)
2018-12-08 18:53:58.760  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Current version of schema `saconf`: 1
2018-12-08 18:53:58.775  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Migrating schema `saconf` to version 2 - ADD PASSWORD COLUMN
2018-12-08 18:53:59.839  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Successfully applied 1 migration to schema `saconf` (execution time 00:01.137s)
2018-12-08 18:54:01.416  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [
        name: default
        ...]
2018-12-08 18:54:02.470  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate Core {5.3.7.Final}
2018-12-08 18:54:02.492  INFO 1 --- [           main] org.hibernate.cfg.Environment            : HHH000206: hibernate.properties not found
2018-12-08 18:54:04.035  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
2018-12-08 18:54:11.381  INFO 1 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```

kubectl rollout undo deployment app-with-sql-persistence-deployment



```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl rollout undo deployment app-with-sql-persistence-deployment
deployment "app-with-sql-persistence-deployment" rolled back
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```



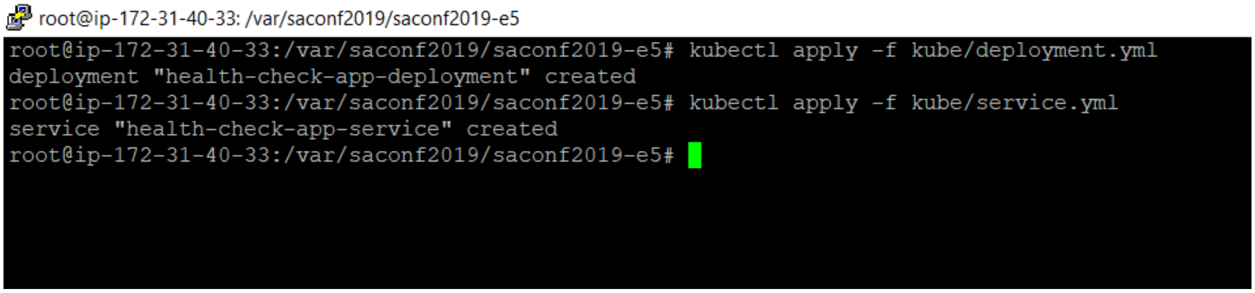
```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4# kubectl logs app-with-sql-persistence-deployment-55f65cbd58-qt4vj

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.1.RELEASE)

2018-12-08 18:57:04.879  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : Starting Saconf2019E4Application v0.0.1-SNAPSHOT on app-with-sql-persistence-deployment-55f6
5cbd58-qt4vj with PID 1 (/app.jar started by root in /)
2018-12-08 18:57:05.041  INFO 1 --- [           main] c.e.s.Saconf2019E4Application            : No active profile set, falling back to default profiles: default
2018-12-08 18:57:27.150  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mode.
2018-12-08 18:57:28.212  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 968ms. Found 1 repository interfaces.
2018-12-08 18:57:36.959  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of
 type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIB$$fbc704a5] is not eligible for getting processed by all BeanPostProcessors (
for example: not eligible for auto-proxying)
2018-12-08 18:57:49.698  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2018-12-08 18:57:50.391  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2018-12-08 18:57:50.445  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/9.0.13
2018-12-08 18:57:52.365  INFO 1 --- [           main] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance in production en
vironments was not found on the java.library.path: [/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64/server:/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64:/usr/lib/jvm/java-1.8-openjdk/jre/../lib/a
md64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]
2018-12-08 18:57:55.094  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2018-12-08 18:57:55.094  INFO 1 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 48898 ms
2018-12-08 18:57:59.566  INFO 1 --- [           main] o.f.c.internal.license.VersionPrinter    : Flyway Community Edition 5.2.3 by Boxfuse
2018-12-08 18:57:59.708  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2018-12-08 18:58:05.574  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2018-12-08 18:58:05.651  INFO 1 --- [           main] o.f.c.internal.database.DatabaseFactory  : Database: jdbc:mysql://172.31.40.33:30001/saconf (MySQL 5.6)
2018-12-08 18:58:08.556  INFO 1 --- [           main] o.f.core.internal.command.DbValidate     : Successfully validated 2 migrations (execution time 00:00.556s)
2018-12-08 18:58:08.651  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Current version of schema `saconf`: 2
2018-12-08 18:58:08.651  WARN 1 --- [           main] o.f.core.internal.command.DbMigrate      : Schema `saconf` has a version (2) that is newer than the latest available migration (1) !
2018-12-08 18:58:08.651  INFO 1 --- [           main] o.f.core.internal.command.DbMigrate      : Schema `saconf` is up to date. No migration necessary.
2018-12-08 18:58:11.180  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [
        name: default
        ...]
2018-12-08 18:58:12.605  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate Core {5.3.7.Final}
2018-12-08 18:58:12.621  INFO 1 --- [           main] org.hibernate.cfg.Environment            : HHH000206: hibernate.properties not found
2018-12-08 18:58:16.678  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
2018-12-08 18:58:20.681  INFO 1 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
2018-12-08 18:58:27.418  INFO 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e4#
```

6. The actuator – check the healthiness of your container

In the early days of software, the application knew about the environment. Now the relationship is bidirectional. The environment must also know about the application. Frameworks designed for microservices architectures - like Spring Boot in java - come with implementations like the actuator to facilitate this.

kubectl apply -f kube/deployment.yml

kubectl apply -f kube/service.yml

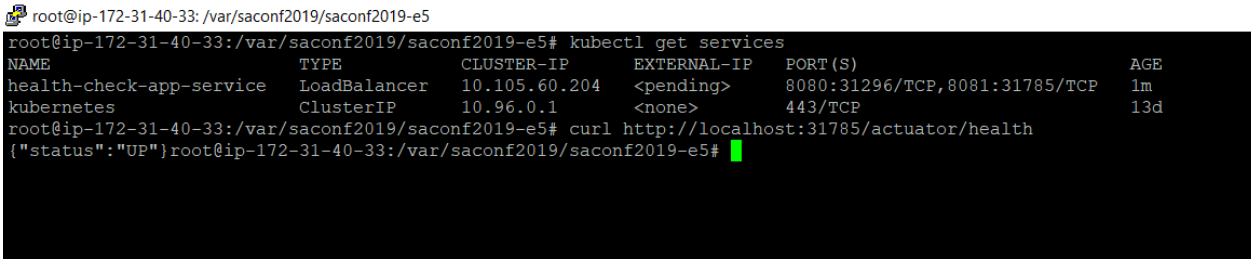


root@ip-172-31-40-33: /var/saconf2019/saconf2019-e5

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e5# kubectl apply -f kube/deployment.yml
deployment "health-check-app-deployment" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e5# kubectl apply -f kube/service.yml
service "health-check-app-service" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e5#
```

kubectl get services

curl http://localhost:<port>/actuator/health



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e5

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e5# kubectl get services
NAME                        TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                         AGE
health-check-app-service    LoadBalancer   10.105.60.204   <pending>     8080:31296/TCP,8081:31785/TCP   1m
kubernetes                  ClusterIP      10.96.0.1       <none>        443/TCP                         13d
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e5# curl http://localhost:31785/actuator/health
{"status":"UP"}root@ip-172-31-40-33:/var/saconf2019/saconf2019-e5#
```

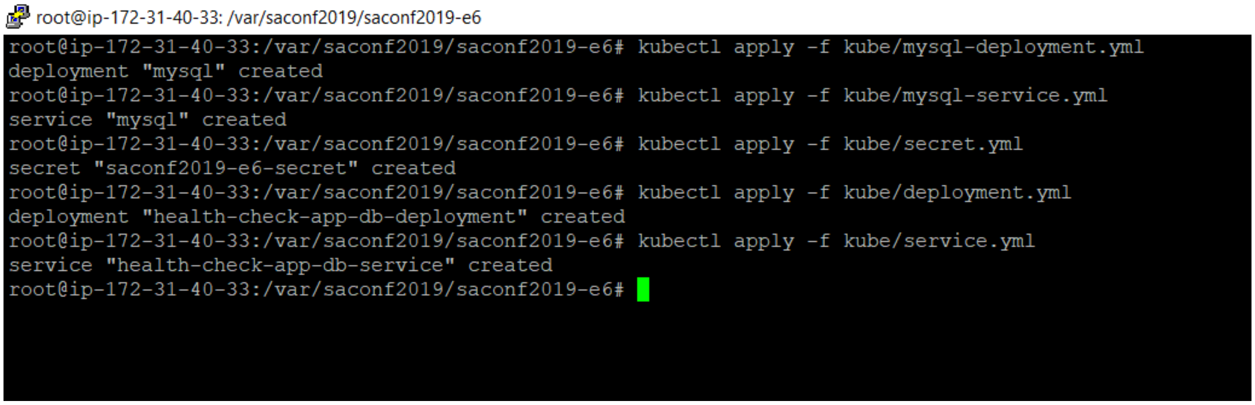
kubectl apply -f kube/mysql-deployment.yml

kubectl apply -f kube/mysql-service.yml

kubectl apply -f kube/secret.yml

kubectl apply -f kube/deployment.yml

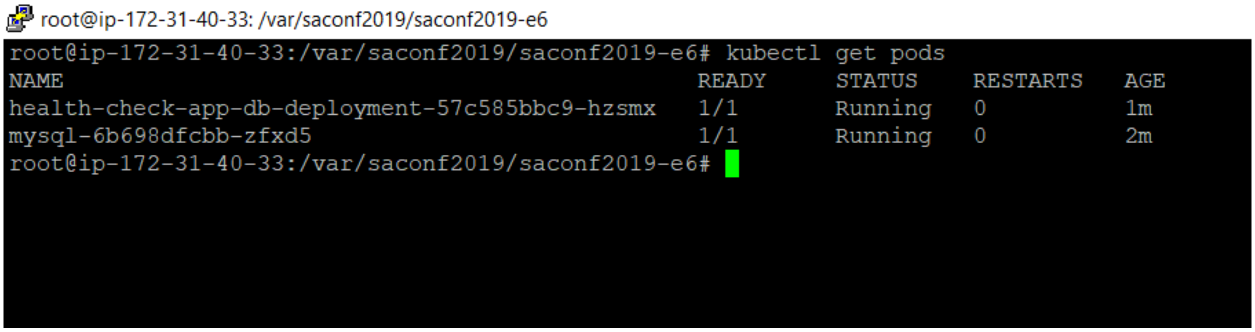
kubectl apply -f kube/service.yml



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e6

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl apply -f kube/mysql-deployment.yml
deployment "mysql" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl apply -f kube/mysql-service.yml
service "mysql" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl apply -f kube/secret.yml
secret "saconf2019-e6-secret" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl apply -f kube/deployment.yml
deployment "health-check-app-db-deployment" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl apply -f kube/service.yml
service "health-check-app-db-service" created
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6#
```

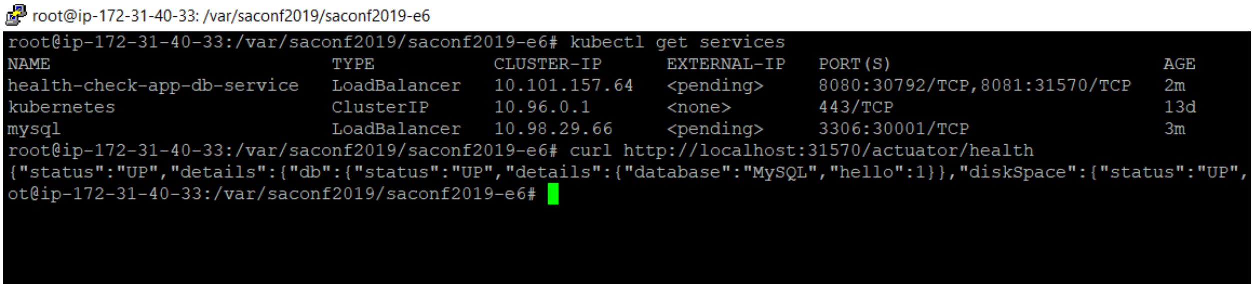
kubectl get pods



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e6

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl get pods
NAME                                              READY     STATUS    RESTARTS   AGE
health-check-app-db-deployment-57c585bbc9-hzsmx   1/1       Running   0          1m
mysql-6b698dfcbb-zfxd5                            1/1       Running   0          2m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6#
```

curl http://localhost:<port>/actuator/health



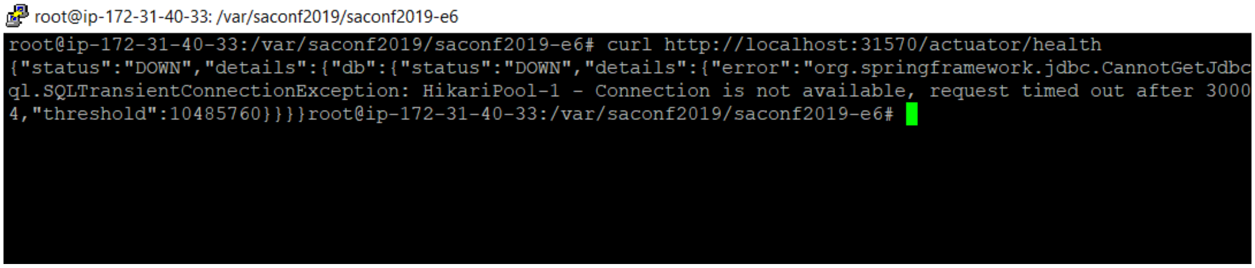
root@ip-172-31-40-33: /var/saconf2019/saconf2019-e6

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl get services
NAME                          TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                         AGE
health-check-app-db-service   LoadBalancer   10.101.157.64   <pending>     8080:30792/TCP,8081:31570/TCP   2m
kubernetes                    ClusterIP      10.96.0.1       <none>        443/TCP                         13d
mysql                         LoadBalancer   10.98.29.66     <pending>     3306:30001/TCP                  3m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# curl http://localhost:31570/actuator/health
{"status":"UP","details":{"db":{"status":"UP","details":{"database":"MySQL","hello":1}},"diskSpace":{"status":"UP",
ot@ip-172-31-40-33:/var/saconf2019/saconf2019-e6#
```

kubectl delete deployments mysql

kubectl delete services mysql

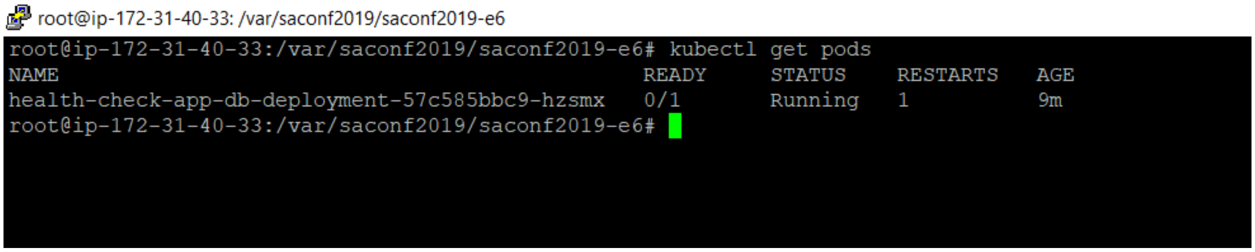
curl http://localhost:<port>/actuator/health



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e6

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# curl http://localhost:31570/actuator/health
{"status":"DOWN","details":{"db":{"status":"DOWN","details":{"error":"org.springframework.jdbc.CannotGetJdbc
ql.SQLTransientConnectionException: HikariPool-1 - Connection is not available, request timed out after 3000
4,"threshold":10485760}}}}root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# 
```

kubectl get pods



root@ip-172-31-40-33: /var/saconf2019/saconf2019-e6

```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# kubectl get pods
NAME                                               READY     STATUS    RESTARTS   AGE
health-check-app-db-deployment-57c585bbc9-hzsmx    0/1       Running   1          9m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e6# 
```

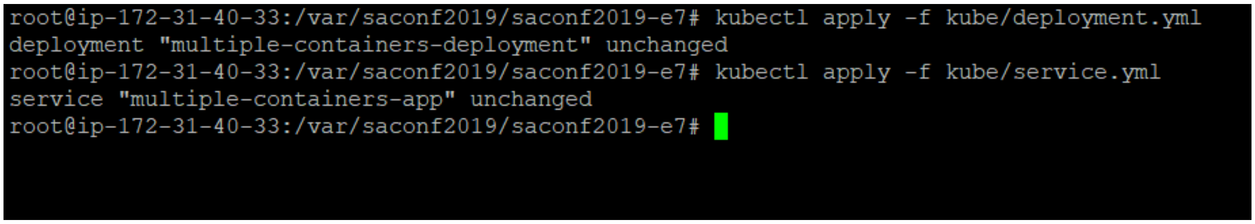
7. An unfortunate choice - having multiple containers in the same pod

We usually think of a pod as having only one container. But a pod can actually have multiple containers. Is it a good practice to deploy related containers in the same pod?

Let’s first start with seeing how is this possible.

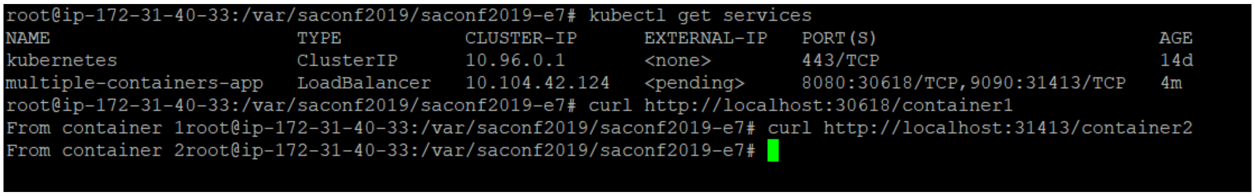
kubectl apply -f kube/deployment.yml

kubectl apply -f kube/service.yml



```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# kubectl apply -f kube/deployment.yml
deployment "multiple-containers-deployment" unchanged
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# kubectl apply -f kube/service.yml
service "multiple-containers-app" unchanged
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# █
```

kubectl get services



```
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# kubectl get services
NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                          AGE
kubernetes                ClusterIP      10.96.0.1       <none>        443/TCP                          14d
multiple-containers-app   LoadBalancer   10.104.42.124   <pending>     8080:30618/TCP,9090:31413/TCP    4m
root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# curl http://localhost:30618/container1
From container 1root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# curl http://localhost:31413/container2
From container 2root@ip-172-31-40-33:/var/saconf2019/saconf2019-e7# █
```

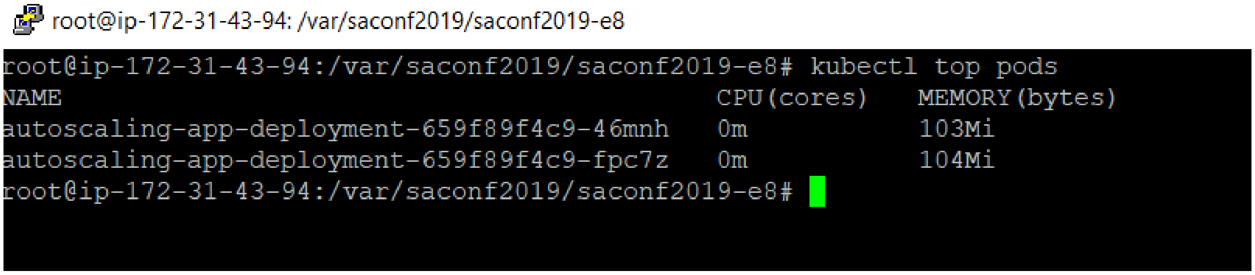
8. Limits and auto-scaling

Limitation is important. We always have to define the request and limit resources for our containers. This can be done in the deployment.yml file as shows in the example 8.

Apply the yml files from the kube folder to run the example. As you have observed we also have here the definition of an auto-scaler within the file hpa.yml

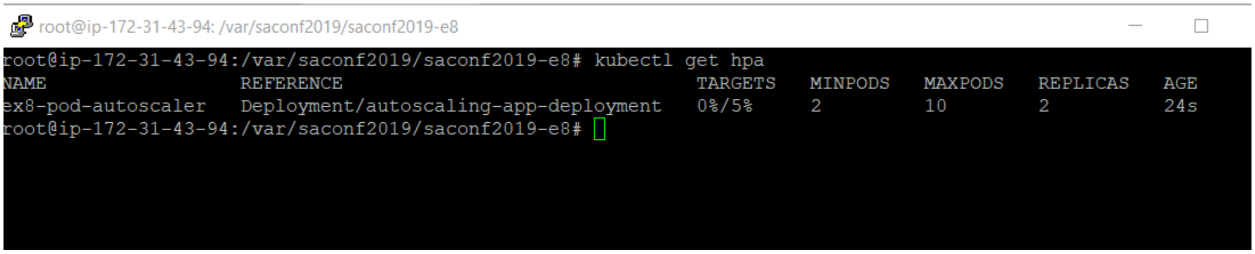
Once we have defined limits we can use the metrics server to display the current status.

kubectl top pods



root@ip-172-31-43-94: /var/saconf2019/saconf2019-e8

```
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# kubectl top pods
NAME                                         CPU(cores)   MEMORY(bytes)
autoscaling-app-deployment-659f89f4c9-46mnh   0m           103Mi
autoscaling-app-deployment-659f89f4c9-fpc7z   0m           104Mi
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8#
```

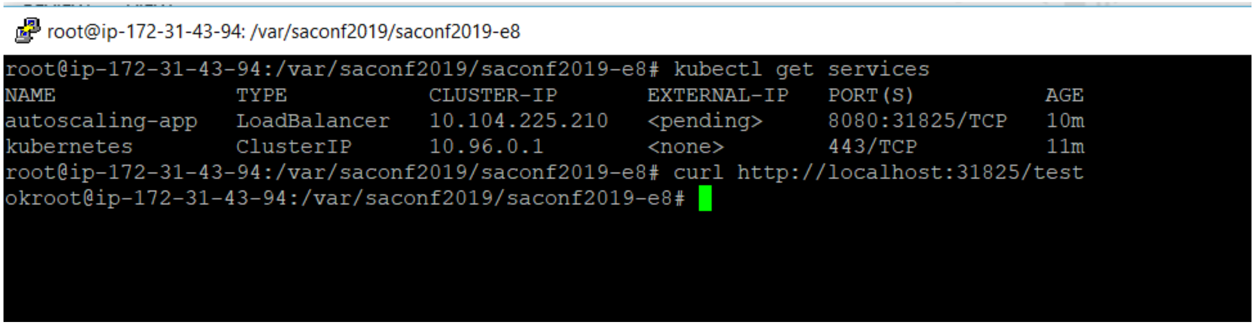


root@ip-172-31-43-94: /var/saconf2019/saconf2019-e8

```
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# kubectl get hpa
NAME                 REFERENCE                               TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
ex8-pod-autoscaler   Deployment/autoscaling-app-deployment   0%/5%     2         10        2          24s
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8#
```

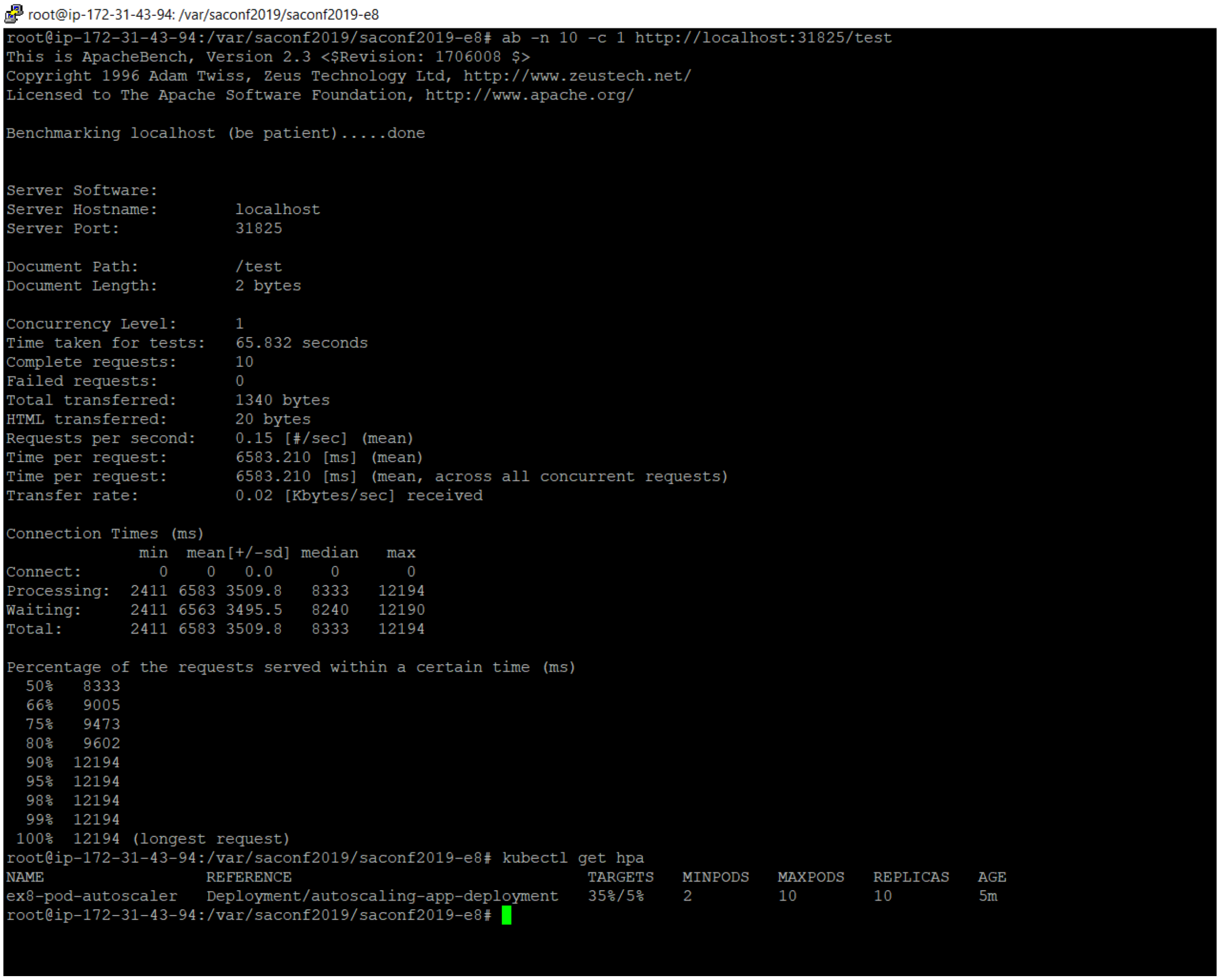
We can use Apache Benchmark to stress a little our service and see the horizontal pod auto-scaler in action.

apt-get install apache2-utils



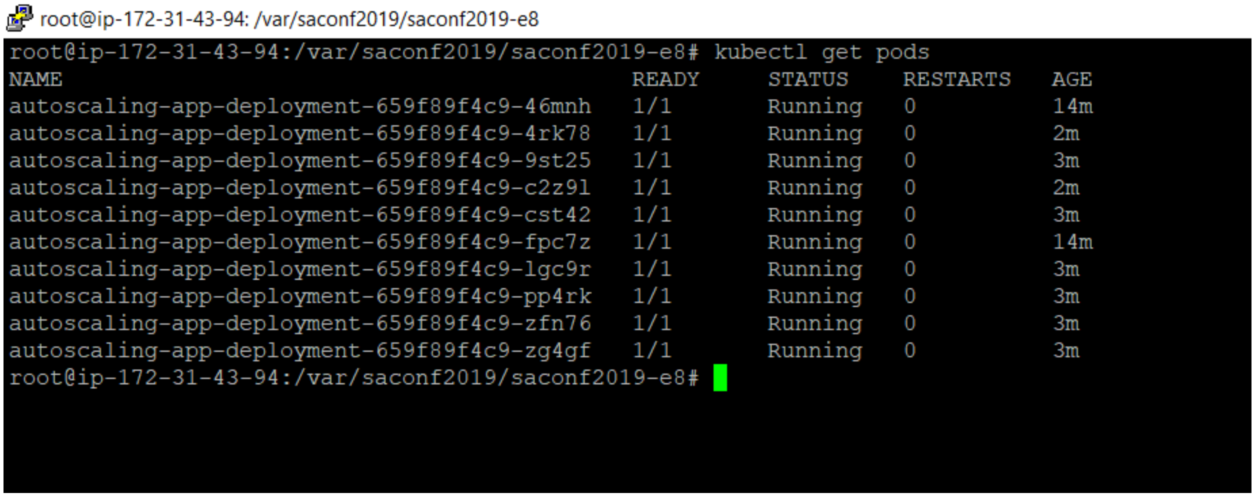
root@ip-172-31-43-94: /var/saconf2019/saconf2019-e8

```
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# kubectl get services
NAME               TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
autoscaling-app    LoadBalancer   10.104.225.210   <pending>     8080:31825/TCP   10m
kubernetes         ClusterIP      10.96.0.1        <none>        443/TCP          11m
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# curl http://localhost:31825/test
okroot@ip-172-31-43-94:/var/saconf2019/saconf2019-e8#
```



root@ip-172-31-43-94: /var/saconf2019/saconf2019-e8

```
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# ab -n 10 -c 1 http://localhost:31825/test
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done


Server Software:
Server Hostname:        localhost
Server Port:            31825

Document Path:          /test
Document Length:        2 bytes

Concurrency Level:      1
Time taken for tests:   65.832 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      1340 bytes
HTML transferred:       20 bytes
Requests per second:    0.15 [#/sec] (mean)
Time per request:       6583.210 [ms] (mean)
Time per request:       6583.210 [ms] (mean, across all concurrent requests)
Transfer rate:          0.02 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.0      0       0
Processing:  2411 6583 3509.8   8333   12194
Waiting:     2411 6563 3495.5   8240   12190
Total:       2411 6583 3509.8   8333   12194

Percentage of the requests served within a certain time (ms)
  50%   8333
  66%   9005
  75%   9473
  80%   9602
  90%  12194
  95%  12194
  98%  12194
  99%  12194
 100%  12194 (longest request)
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# kubectl get hpa
NAME                  REFERENCE                                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
ex8-pod-autoscaler    Deployment/autoscaling-app-deployment    35%/5%    2         10        10         5m
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8#
```



root@ip-172-31-43-94: /var/saconf2019/saconf2019-e8

```
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8# kubectl get pods
NAME                                          READY     STATUS    RESTARTS   AGE
autoscaling-app-deployment-659f89f4c9-46mnh   1/1       Running   0          14m
autoscaling-app-deployment-659f89f4c9-4rk78   1/1       Running   0          2m
autoscaling-app-deployment-659f89f4c9-9st25   1/1       Running   0          3m
autoscaling-app-deployment-659f89f4c9-c2z9l   1/1       Running   0          2m
autoscaling-app-deployment-659f89f4c9-cst42   1/1       Running   0          3m
autoscaling-app-deployment-659f89f4c9-fpc7z   1/1       Running   0          14m
autoscaling-app-deployment-659f89f4c9-lgc9r   1/1       Running   0          3m
autoscaling-app-deployment-659f89f4c9-pp4rk   1/1       Running   0          3m
autoscaling-app-deployment-659f89f4c9-zfn76   1/1       Running   0          3m
autoscaling-app-deployment-659f89f4c9-zg4gf   1/1       Running   0          3m
root@ip-172-31-43-94:/var/saconf2019/saconf2019-e8#
```