SOFTWARE DEVELOPMENT PROPOSAL

PREPARED FOR

Usama Musharaf

PREPARED BY

Umair Maratab

P180045

Summary:

Java based application is deployed minikube (kubernetes) using docker containers, 3 pods are created on a node and each pod has 1 service and main three services and replicas are defined as 1 in YAML file:
Services will be:

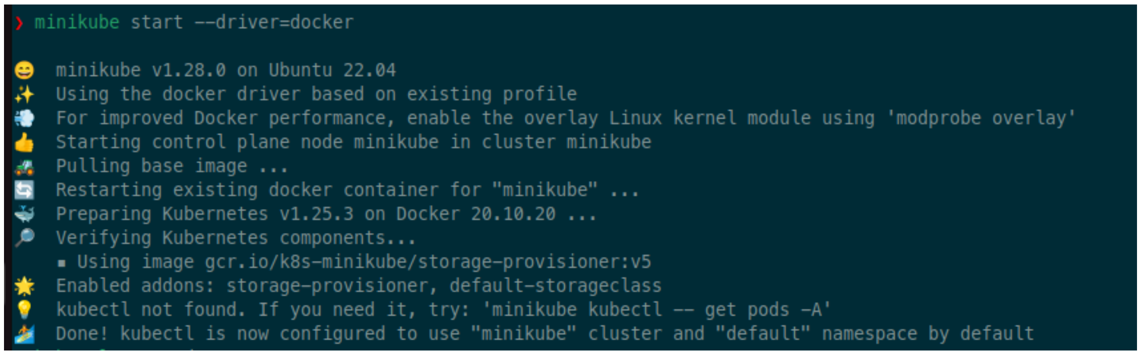
- Shopfront
- Product Catalogue
- StockManager

Start Minikube Cluster

Command:

```
minikube start --driver=docker
```

Output:



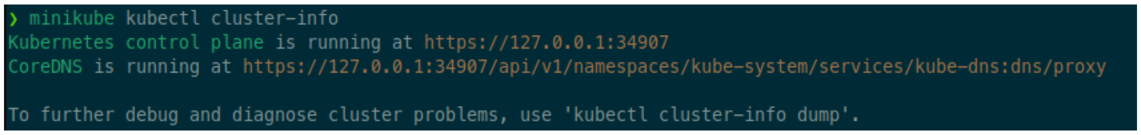
```
❯ minikube start --driver=docker

😄  minikube v1.28.0 on Ubuntu 22.04
✨  Using the docker driver based on existing profile
💨  For improved Docker performance, enable the overlay Linux kernel module using 'modprobe overlay'
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
💡  kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

To check execute:

```
minikube kubectl cluster-info
```

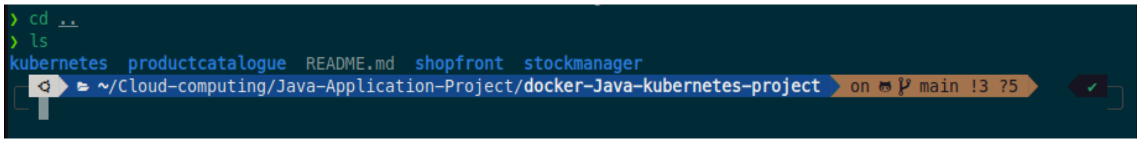
Output:



```
❯ minikube kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:34907
CoreDNS is running at https://127.0.0.1:34907/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Project Directories structure:

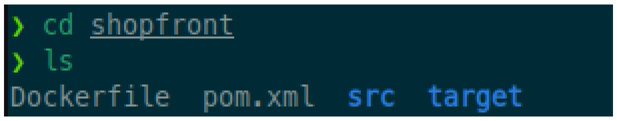


```
❯ cd ..
❯ ls
kubernetes  productcatalogue  README.md  shopfront  stockmanager
  ~/Cloud-computing/Java-Application-Project/docker-Java-kubernetes-project  on  main !3 ?5   ✔
```

Services will be:

- Shopfront
- Product Catalogue
- StockManager

→ We will go to shopfront and build there:

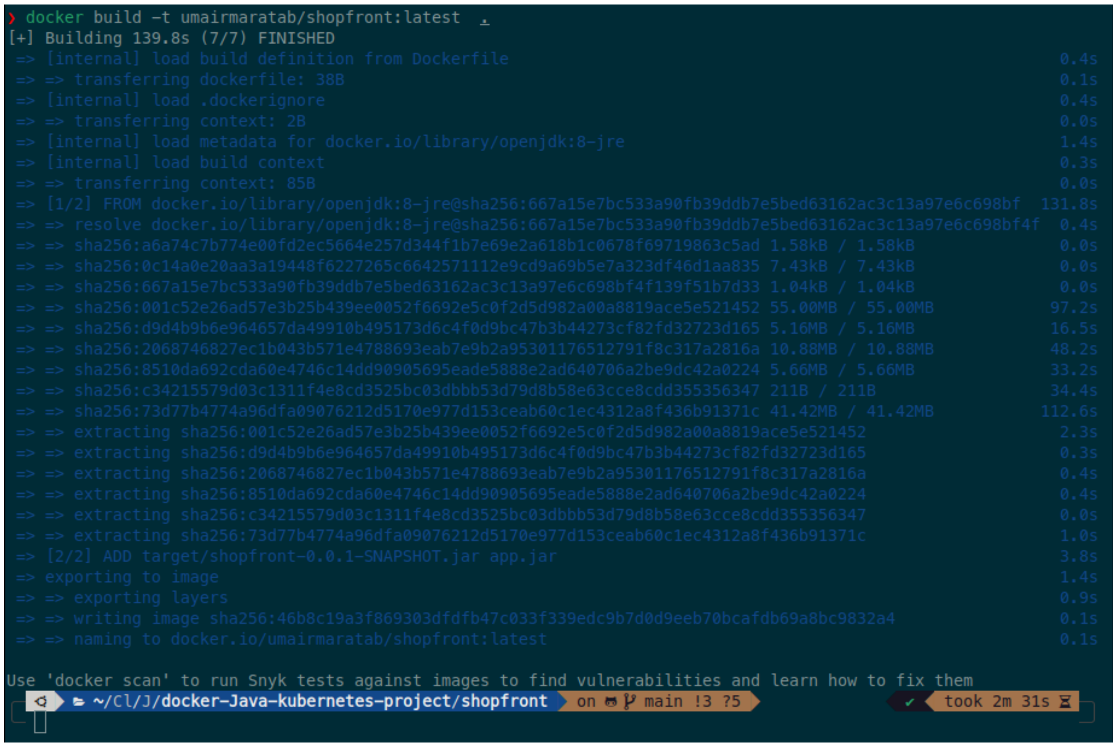


```
❯ cd shopfront
❯ ls
Dockerfile  pom.xml  src  target
```

Build Command:

```
docker build -t "image-name" .
```

Output:



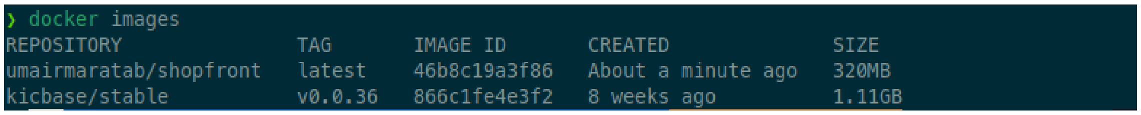
```
❯ docker build -t umairmaratab/shopfront:latest .
[+] Building 139.8s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                                                        0.4s
 => => transferring dockerfile: 38B                                                                         0.1s
 => [internal] load .dockerignore                                                                           0.4s
 => => transferring context: 2B                                                                             0.0s
 => [internal] load metadata for docker.io/library/openjdk:8-jre                                            1.4s
 => [internal] load build context                                                                           0.3s
 => => transferring context: 85B                                                                            0.0s
 => [1/2] FROM docker.io/library/openjdk:8-jre@sha256:667a15e7bc533a90fb39ddb7e5bed63162ac3c13a97e6c698bf  131.8s
 => => resolve docker.io/library/openjdk:8-jre@sha256:667a15e7bc533a90fb39ddb7e5bed63162ac3c13a97e6c698bf4f  0.4s
 => => sha256:a6a74c7b774e00fd2ec5664e257d344f1b7e69e2a618b1c0678f69719863c5ad 1.58kB / 1.58kB              0.0s
 => => sha256:0c14a0e20aa3a19448f6227265c6642571112e9cd9a69b5e7a323df46d1aa835 7.43kB / 7.43kB              0.0s
 => => sha256:667a15e7bc533a90fb39ddb7e5bed63162ac3c13a97e6c698bf4f139f51b7d33 1.04kB / 1.04kB              0.0s
 => => sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452 55.00MB / 55.00MB           97.2s
 => => sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b44273cf82fd32723d165 5.16MB / 5.16MB             16.5s
 => => sha256:2068746827ec1b043b571e4788693eab7e9b2a95301176512791f8c317a2816a 10.88MB / 10.88MB           48.2s
 => => sha256:8510da692cda60e4746c14dd90905695eade5888e2ad640706a2be9dc42a0224 5.66MB / 5.66MB             33.2s
 => => sha256:c34215579d03c1311f4e8cd3525bc03dbbb53d79d8b58e63cce8cdd355356347 211B / 211B                 34.4s
 => => sha256:73d77b4774a96dfa09076212d5170e977d153ceab60c1ec4312a8f436b91371c 41.42MB / 41.42MB          112.6s
 => => extracting sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452                   2.3s
 => => extracting sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b44273cf82fd32723d165                   0.3s
 => => extracting sha256:2068746827ec1b043b571e4788693eab7e9b2a95301176512791f8c317a2816a                   0.4s
 => => extracting sha256:8510da692cda60e4746c14dd90905695eade5888e2ad640706a2be9dc42a0224                   0.4s
 => => extracting sha256:c34215579d03c1311f4e8cd3525bc03dbbb53d79d8b58e63cce8cdd355356347                   0.0s
 => => extracting sha256:73d77b4774a96dfa09076212d5170e977d153ceab60c1ec4312a8f436b91371c                   1.0s
 => [2/2] ADD target/shopfront-0.0.1-SNAPSHOT.jar app.jar                                                   3.8s
 => exporting to image                                                                                      1.4s
 => => exporting layers                                                                                     0.9s
 => => writing image sha256:46b8c19a3f869303dfdfb47c033f339edc9b7d0d9eeb70bcafdb69a8bc9832a4                0.1s
 => => naming to docker.io/umairmaratab/shopfront:latest                                                    0.1s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
   ~/Cl/J/docker-Java-kubernetes-project/shopfront  on  main !3 ?5                       ✓  took 2m 31s ⧗
```

Check images now by:

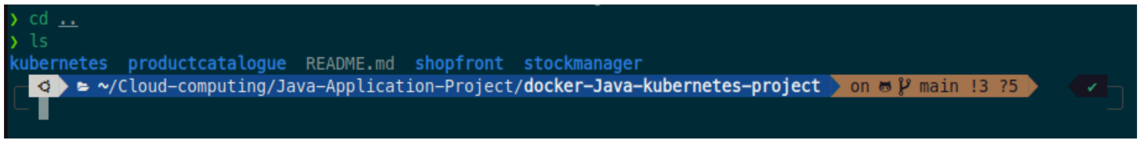
```
docker images
```

Output:



```
❯ docker images
REPOSITORY                TAG       IMAGE ID       CREATED              SIZE
umairmaratab/shopfront    latest    46b8c19a3f86   About a minute ago   320MB
kicbase/stable            v0.0.36   866c1fe4e3f2   8 weeks ago          1.11GB
```

Go back and go to second service directory and build there:

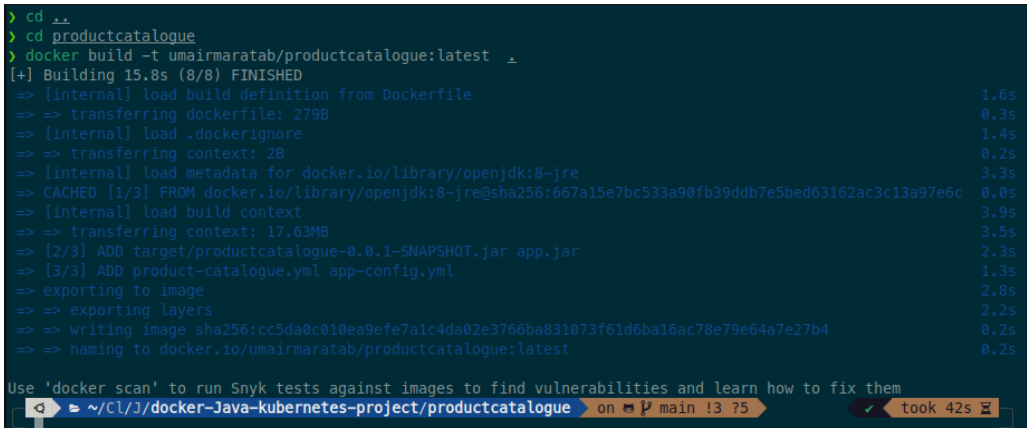


```
❯ cd ..
❯ ls
kubernetes  productcatalogue  README.md  shopfront  stockmanager
 ~/Cloud-computing/Java-Application-Project/docker-Java-kubernetes-project  on  main !3 ?5                ✓
```

Build Command:

```
docker build -t "image-name" .
```

Output:



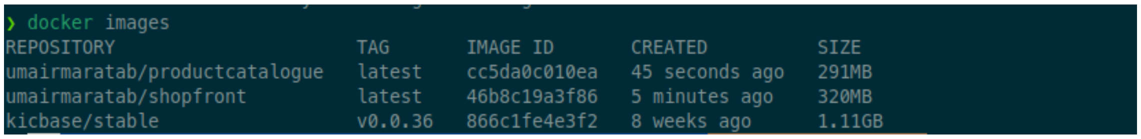
```
❯ cd ..
❯ cd productcatalogue
❯ docker build -t umairmaratab/productcatalogue:latest  .
[+] Building 15.8s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                                                   1.6s
 => => transferring dockerfile: 279B                                                                   0.3s
 => [internal] load .dockerignore                                                                      1.4s
 => => transferring context: 2B                                                                        0.2s
 => [internal] load metadata for docker.io/library/openjdk:8-jre                                      3.3s
 => CACHED [1/3] FROM docker.io/library/openjdk:8-jre@sha256:667a15e7bc533a90fb39ddb7e5bed63162ac3c13a97e6c  0.0s
 => [internal] load build context                                                                      3.9s
 => => transferring context: 17.63MB                                                                   3.5s
 => [2/3] ADD target/productcatalogue-0.0.1-SNAPSHOT.jar app.jar                                      2.3s
 => [3/3] ADD product-catalogue.yml app-config.yml                                                    1.3s
 => exporting to image                                                                                 2.8s
 => => exporting layers                                                                                2.2s
 => => writing image sha256:cc5da0c010ea9efe7a1c4da02e3766ba831073f61d6ba16ac78e79e64a7e27b4           0.2s
 => => naming to docker.io/umairmaratab/productcatalogue:latest                                       0.2s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
 ~/Cl/J/docker-Java-kubernetes-project/productcatalogue  on  main !3 ?5          ✓  took 42s
```

List images again by:

```
docker images
```

Output:



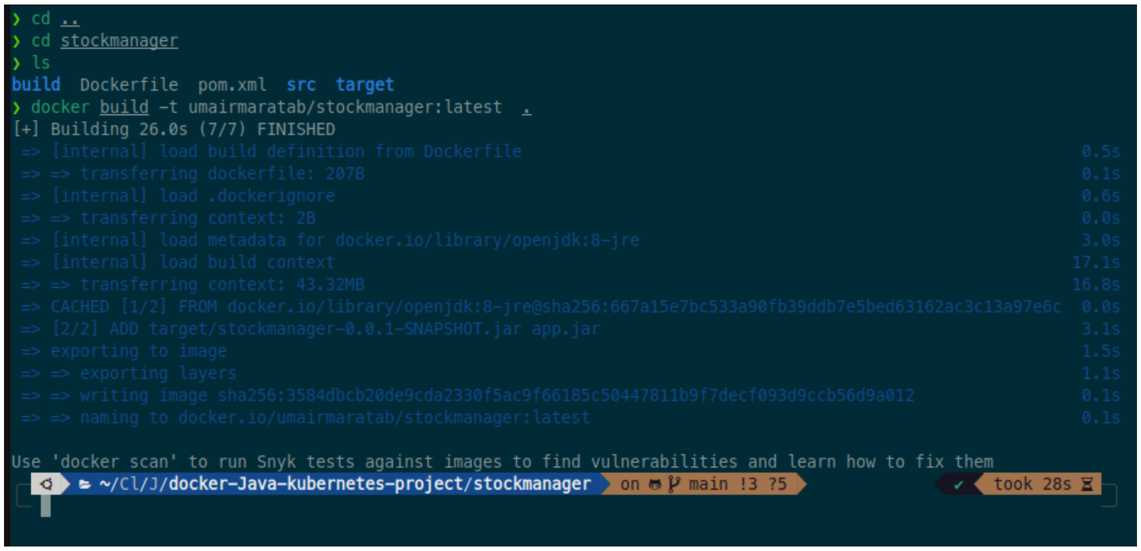
```
❯ docker images
REPOSITORY                     TAG       IMAGE ID       CREATED          SIZE
umairmaratab/productcatalogue  latest    cc5da0c010ea   45 seconds ago   291MB
umairmaratab/shopfront         latest    46b8c19a3f86   5 minutes ago    320MB
kicbase/stable                 v0.0.36   866c1fe4e3f2   8 weeks ago      1.11GB
```

Now we will build our last service using docker-build:

Build Command:

```
docker build -t "image-name" .
```

Output:

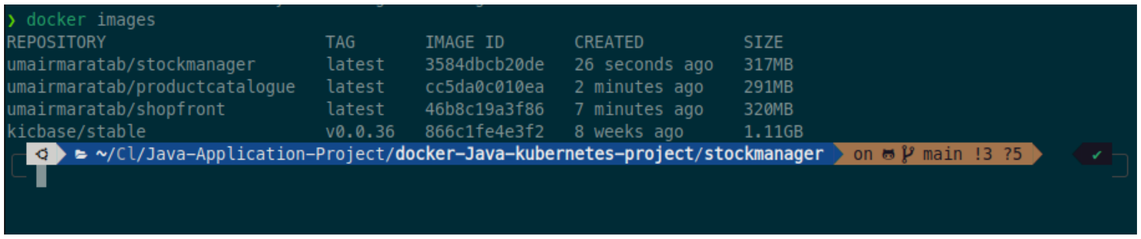


```
❯ cd ..
❯ cd stockmanager
❯ ls
build  Dockerfile  pom.xml  src  target
❯ docker build -t umairmaratab/stockmanager:latest  .
[+] Building 26.0s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                                                    0.5s
 => => transferring dockerfile: 207B                                                                    0.1s
 => [internal] load .dockerignore                                                                       0.6s
 => => transferring context: 2B                                                                         0.0s
 => [internal] load metadata for docker.io/library/openjdk:8-jre                                        3.0s
 => [internal] load build context                                                                      17.1s
 => => transferring context: 43.32MB                                                                   16.8s
 => CACHED [1/2] FROM docker.io/library/openjdk:8-jre@sha256:667a15e7bc533a90fb39ddb7e5bed63162ac3c13a97e6c  0.0s
 => [2/2] ADD target/stockmanager-0.0.1-SNAPSHOT.jar app.jar                                            3.1s
 => exporting to image                                                                                  1.5s
 => => exporting layers                                                                                 1.1s
 => => writing image sha256:3584dbcb20de9cda2330f5ac9f66185c50447811b9f7decf093d9ccb56d9a012            0.1s
 => => naming to docker.io/umairmaratab/stockmanager:latest                                             0.1s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
 ~/Cl/J/docker-Java-kubernetes-project/stockmanager  on  main !3 ?5                    ✔  took 28s
```

List images now:

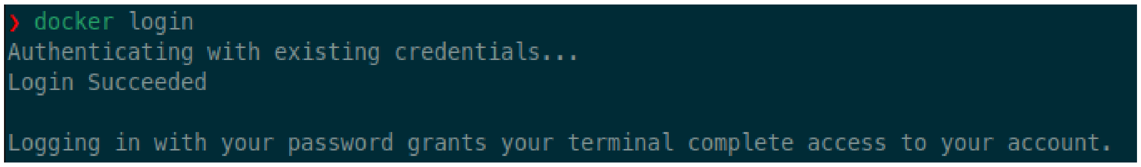
```
docker images
```



```
❯ docker images
REPOSITORY                      TAG       IMAGE ID       CREATED          SIZE
umairmaratab/stockmanager       latest    3584dbcb20de   26 seconds ago   317MB
umairmaratab/productcatalogue   latest    cc5da0c010ea   2 minutes ago    291MB
umairmaratab/shopfront          latest    46b8c19a3f86   7 minutes ago    320MB
kicbase/stable                  v0.0.36   866c1fe4e3f2   8 weeks ago      1.11GB
~/Cl/Java-Application-Project/docker-Java-kubernetes-project/stockmanager on main !3 ?5
```

All images have been built successfully.

Now we will push these on the docker hub.

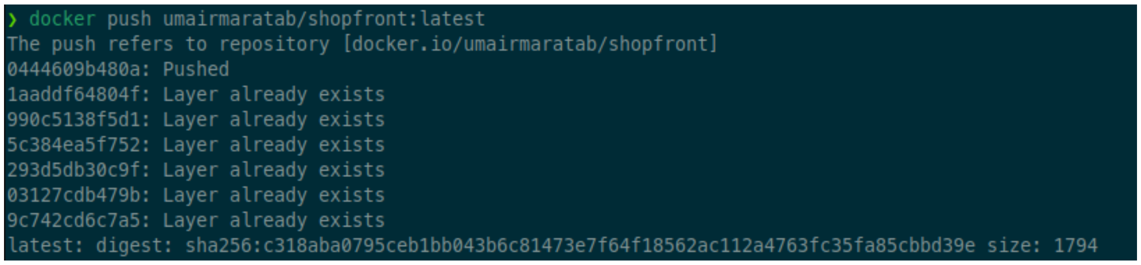


```
❯ docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
```

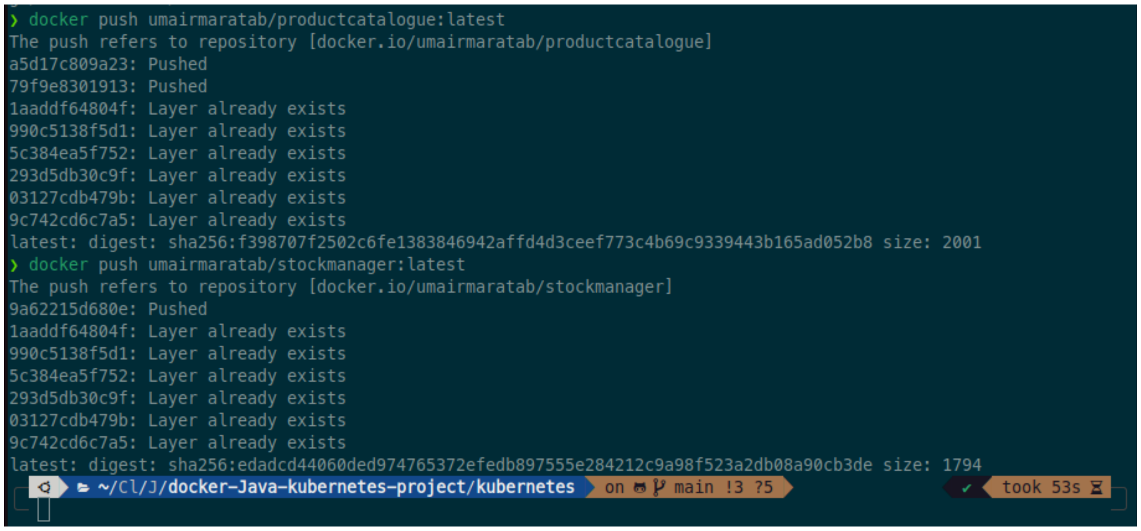
Push using docker command i.e docker push

```
docker push "image-name"
```



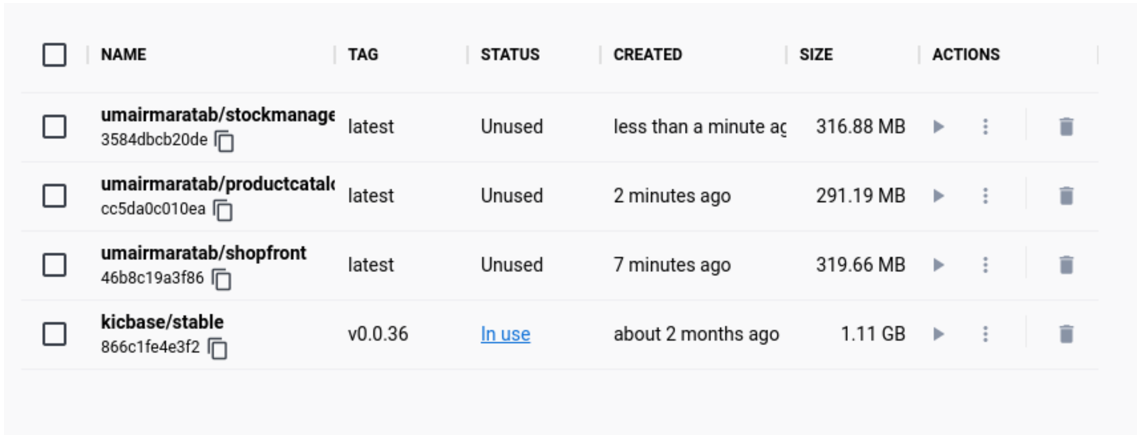
```
❯ docker push umairmaratab/shopfront:latest
The push refers to repository [docker.io/umairmaratab/shopfront]
0444609b480a: Pushed
1aaddf64804f: Layer already exists
990c5138f5d1: Layer already exists
5c384ea5f752: Layer already exists
293d5db30c9f: Layer already exists
03127cdb479b: Layer already exists
9c742cd6c7a5: Layer already exists
latest: digest: sha256:c318aba0795ceb1bb043b6c81473e7f64f18562ac112a4763fc35fa85cbbd39e size: 1794
```

Likewise push other 2 images as well.



```
❯ docker push umairmaratab/productcatalogue:latest
The push refers to repository [docker.io/umairmaratab/productcatalogue]
a5d17c809a23: Pushed
79f9e8301913: Pushed
1aaddf64804f: Layer already exists
990c5138f5d1: Layer already exists
5c384ea5f752: Layer already exists
293d5db30c9f: Layer already exists
03127cdb479b: Layer already exists
9c742cd6c7a5: Layer already exists
latest: digest: sha256:f398707f2502c6fe1383846942affd4d3ceef773c4b69c9339443b165ad052b8 size: 2001
❯ docker push umairmaratab/stockmanager:latest
The push refers to repository [docker.io/umairmaratab/stockmanager]
9a62215d680e: Pushed
1aaddf64804f: Layer already exists
990c5138f5d1: Layer already exists
5c384ea5f752: Layer already exists
293d5db30c9f: Layer already exists
03127cdb479b: Layer already exists
9c742cd6c7a5: Layer already exists
latest: digest: sha256:edadcd44060ded974765372efedb897555e284212c9a98f523a2db08a90cb3de size: 1794
   ~/Cl/J/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5                          ✓  took 53s
```

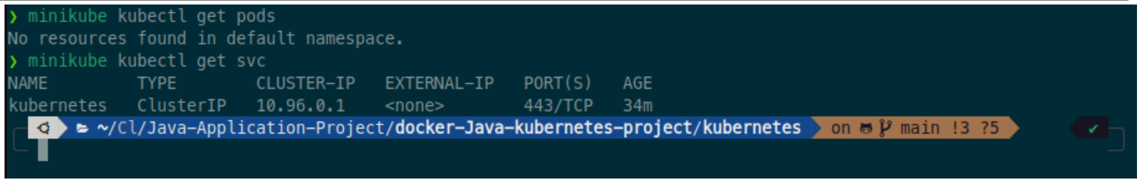
To confirm go to your browser login there and see if images are there:



	NAME	TAG	STATUS	CREATED	SIZE	ACTIONS
☐	**umairmaratab/stockmanage** 3584dbcb20de	latest	Unused	less than a minute ag	316.88 MB	
☐	**umairmaratab/productcatal** cc5da0c010ea	latest	Unused	2 minutes ago	291.19 MB	
☐	**umairmaratab/shopfront** 46b8c19a3f86	latest	Unused	7 minutes ago	319.66 MB	
☐	**kicbase/stable** 866c1fe4e3f2	v0.0.36	In use	about 2 months ago	1.11 GB	

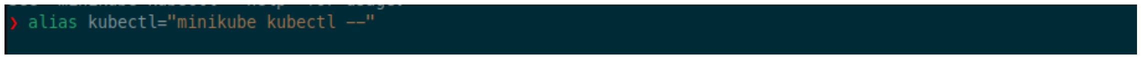
List minikube Pods and services by:

```
minikube kubectl get pods,svc
```



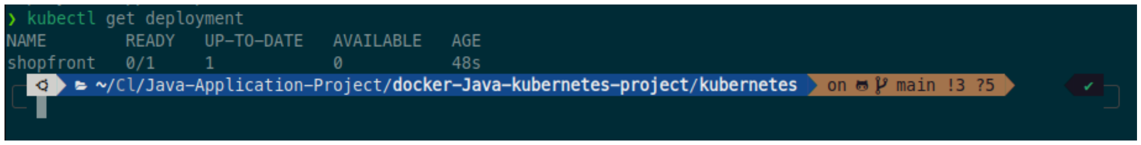
```
❯ minikube kubectl get pods
No resources found in default namespace.
❯ minikube kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   34m
 ~/Cl/Java-Application-Project/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5  ✔
```

You can do alias so you donot have to specify minikube again and again to access kubectl by:



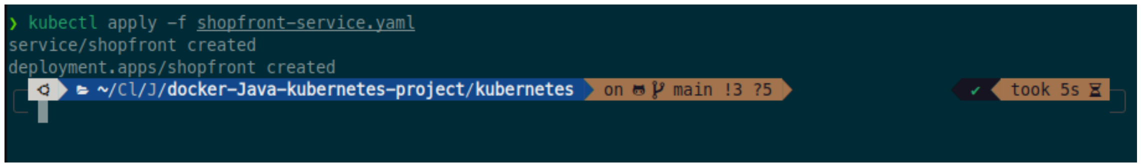
```
❯ alias kubectl="minikube kubectl --"
```

List deployments:

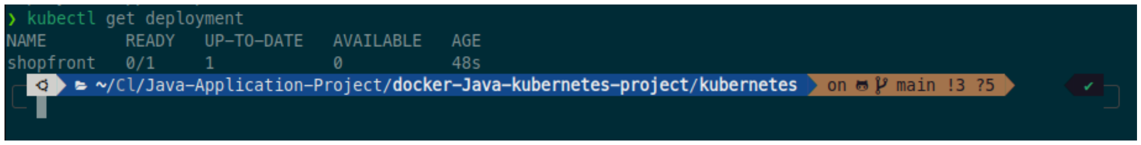


```
❯ kubectl get deployment
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
shopfront   0/1     1            0           48s
 ~/Cl/Java-Application-Project/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5  ✔
```

Go to kubernetes directory and create service:

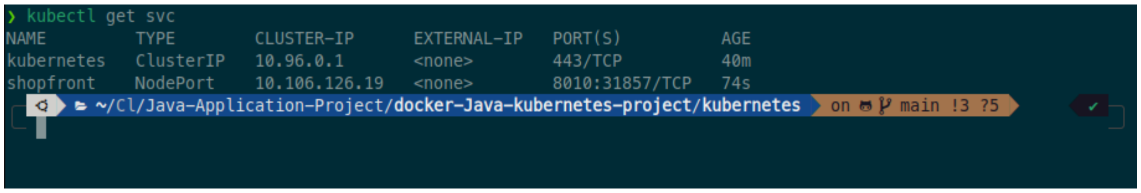


```
❯ kubectl apply -f shopfront-service.yaml
service/shopfront created
deployment.apps/shopfront created
 ~/Cl/J/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5  ✔  took 5s
```

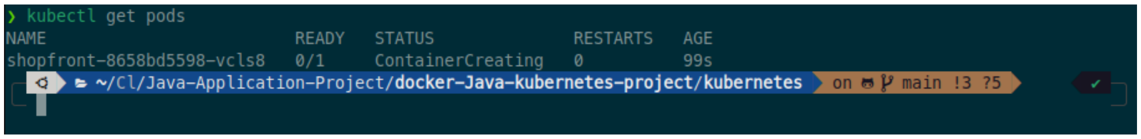


```
❯ kubectl get deployment
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
shopfront   0/1     1            0           48s
 ~/Cl/Java-Application-Project/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5  ✔
```

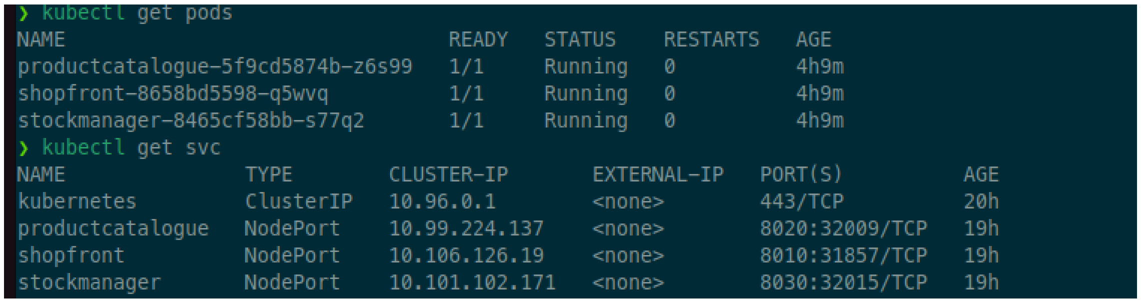
Check services and pods



```
❯ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          40m
shopfront    NodePort    10.106.126.19   <none>        8010:31857/TCP   74s
 ~/Cl/Java-Application-Project/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5          ✔
```

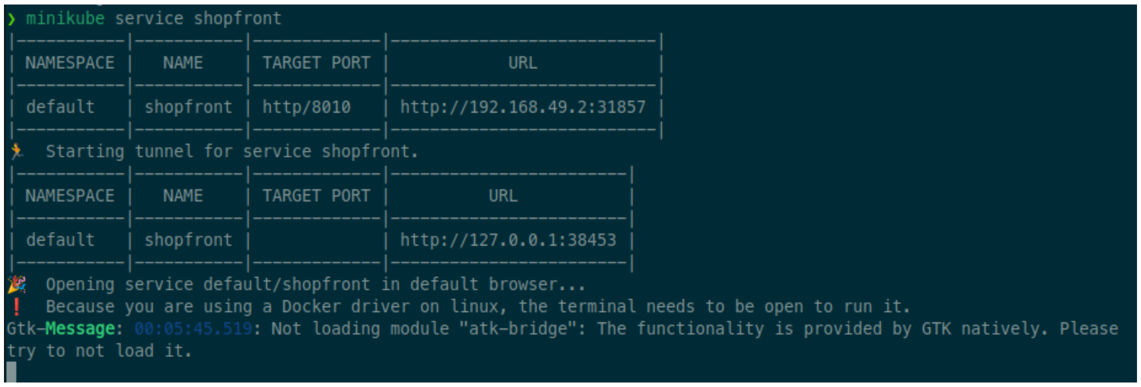


```
❯ kubectl get pods
NAME                         READY   STATUS              RESTARTS   AGE
shopfront-8658bd5598-vcls8   0/1     ContainerCreating   0          99s
 ~/Cl/Java-Application-Project/docker-Java-kubernetes-project/kubernetes  on  main !3 ?5          ✔
```

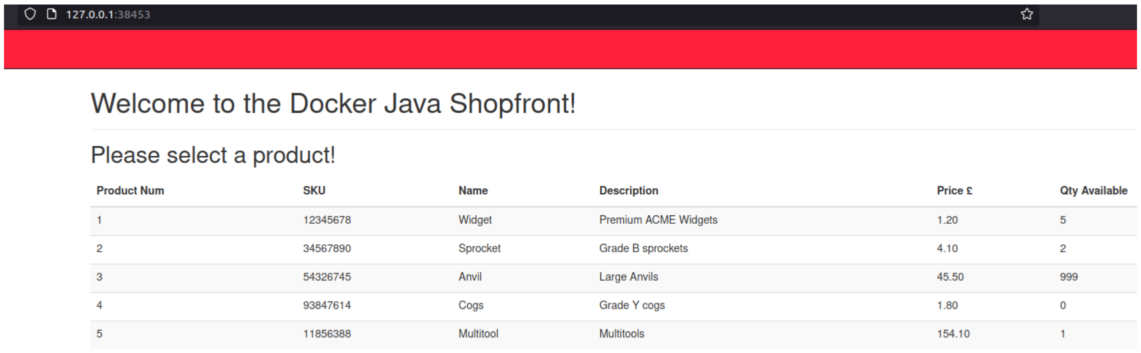


```
❯ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
productcatalogue-5f9cd5874b-z6s99 1/1     Running   0          4h9m
shopfront-8658bd5598-q5wvq        1/1     Running   0          4h9m
stockmanager-8465cf58bb-s77q2     1/1     Running   0          4h9m
❯ kubectl get svc
NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP          20h
productcatalogue   NodePort    10.99.224.137    <none>        8020:32009/TCP   19h
shopfront          NodePort    10.106.126.19    <none>        8010:31857/TCP   19h
stockmanager       NodePort    10.101.102.171   <none>        8030:32015/TCP   19h
```

Access service via browser:



```
❯ minikube service shopfront
|-----------|-----------|-------------|---------------------------|
| NAMESPACE |   NAME    | TARGET PORT |            URL            |
|-----------|-----------|-------------|---------------------------|
| default   | shopfront | http/8010   | http://192.168.49.2:31857 |
|-----------|-----------|-------------|---------------------------|
🏃  Starting tunnel for service shopfront.
|-----------|-----------|-------------|------------------------|
| NAMESPACE |   NAME    | TARGET PORT |          URL           |
|-----------|-----------|-------------|------------------------|
| default   | shopfront |             | http://127.0.0.1:38453 |
|-----------|-----------|-------------|------------------------|
🎉  Opening service default/shopfront in default browser...
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
Gtk-Message: 00:05:45.519: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please
try to not load it.
```


127.0.0.1:38453

Welcome to the Docker Java Shopfront!

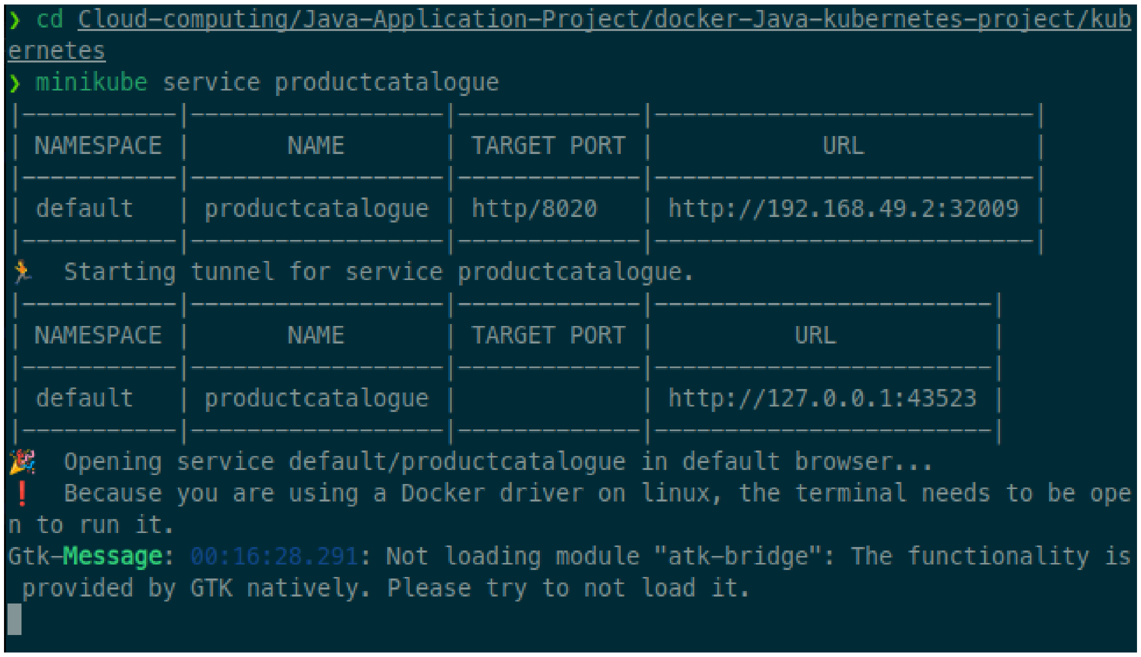
Please select a product!

Product Num	SKU	Name	Description	Price £	Qty Available
1	12345678	Widget	Premium ACME Widgets	1.20	5
2	34567890	Sprocket	Grade B sprockets	4.10	2
3	54326745	Anvil	Large Anvils	45.50	999
4	93847614	Cogs	Grade Y cogs	1.80	0
5	11856388	Multitool	Multitools	154.10	1

Start second service using command:

```
minikube service "service-name"
```

Output:

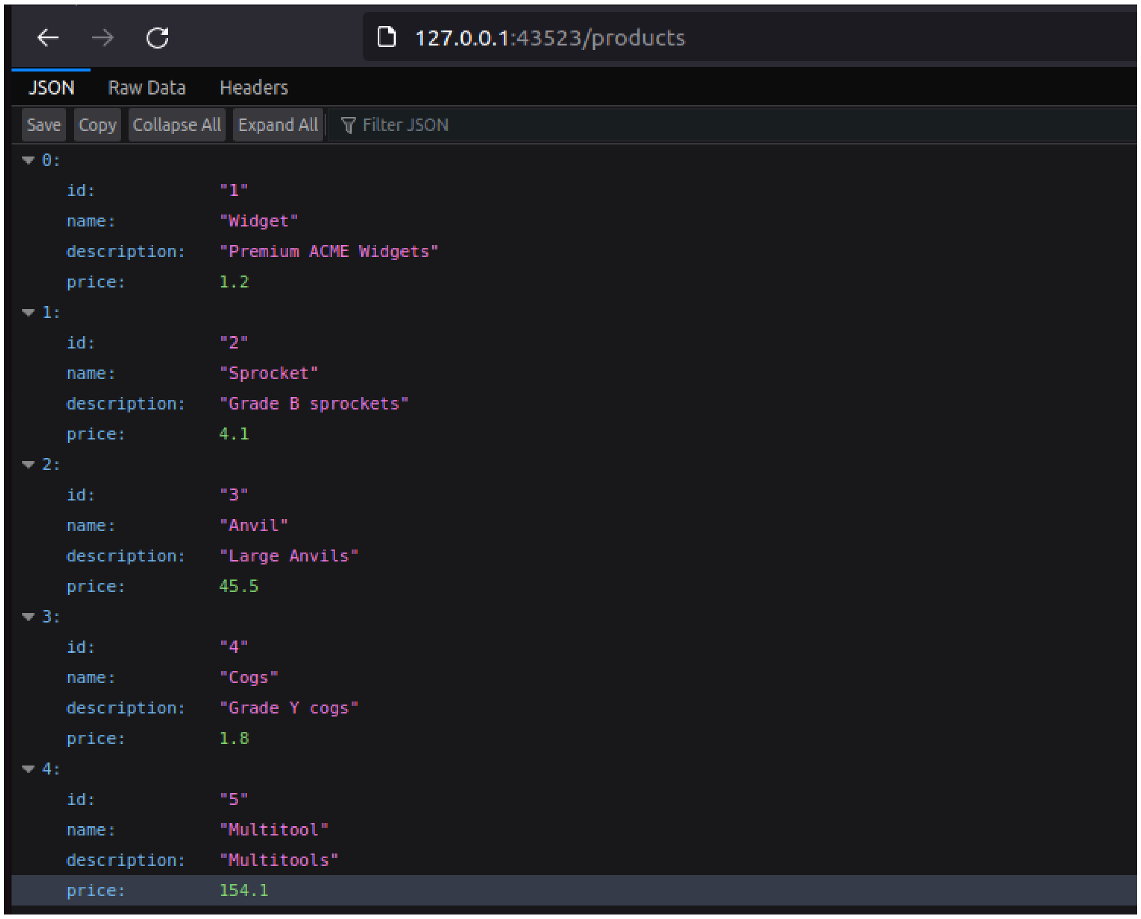


```
❯ cd Cloud-computing/Java-Application-Project/docker-Java-kubernetes-project/kub
ernetes
❯ minikube service productcatalogue
|-----------|-------------------|-------------|---------------------------|
| NAMESPACE |       NAME        | TARGET PORT |            URL            |
|-----------|-------------------|-------------|---------------------------|
| default   | productcatalogue  | http/8020   | http://192.168.49.2:32009 |
|-----------|-------------------|-------------|---------------------------|
🏃  Starting tunnel for service productcatalogue.
|-----------|-------------------|-------------|------------------------|
| NAMESPACE |       NAME        | TARGET PORT |          URL           |
|-----------|-------------------|-------------|------------------------|
| default   | productcatalogue  |             | http://127.0.0.1:43523 |
|-----------|-------------------|-------------|------------------------|
🎉  Opening service default/productcatalogue in default browser...
❗  Because you are using a Docker driver on linux, the terminal needs to be ope
n to run it.
Gtk-Message: 00:16:28.291: Not loading module "atk-bridge": The functionality is
 provided by GTK natively. Please try to not load it.
```

Go to browser and access via:

```
127.0.0.1:43523/products
```

Check here:

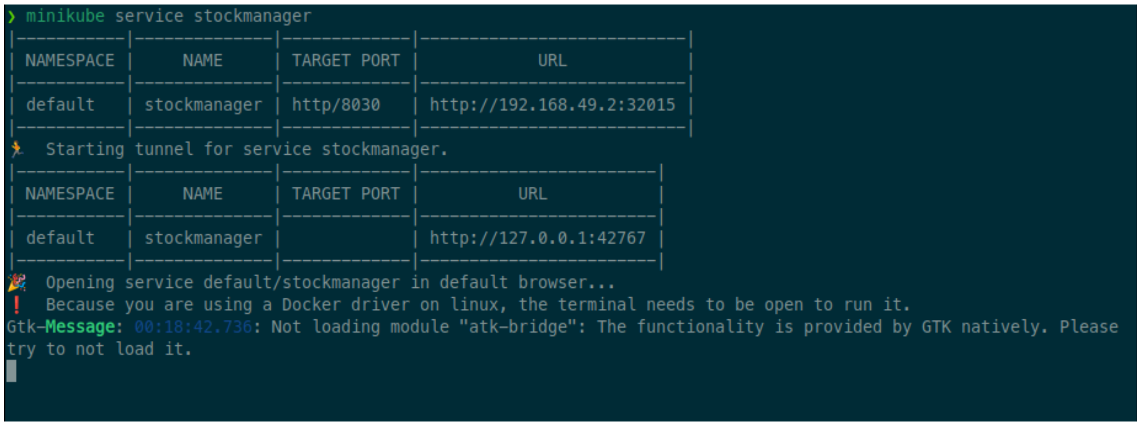


```
0:
    id:           "1"
    name:         "Widget"
    description:  "Premium ACME Widgets"
    price:        1.2
1:
    id:           "2"
    name:         "Sprocket"
    description:  "Grade B sprockets"
    price:        4.1
2:
    id:           "3"
    name:         "Anvil"
    description:  "Large Anvils"
    price:        45.5
3:
    id:           "4"
    name:         "Cogs"
    description:  "Grade Y cogs"
    price:        1.8
4:
    id:           "5"
    name:         "Multitool"
    description:  "Multitools"
    price:        154.1
```

Start third service using command:

```
minikube service "service-name"
```

Output:

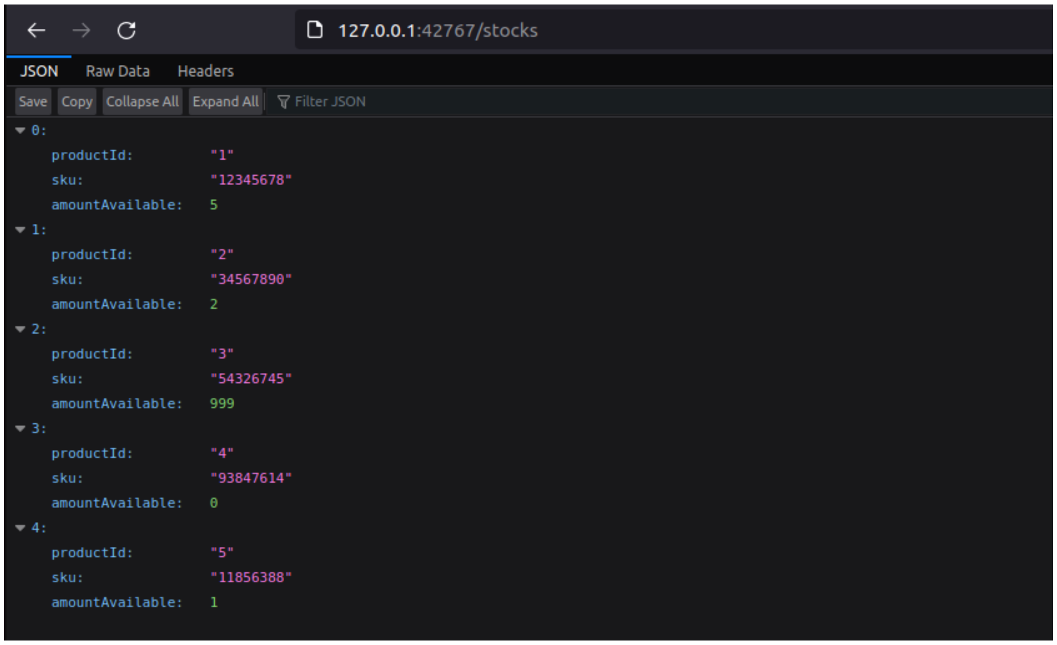


```
❯ minikube service stockmanager
|-----------|--------------|-------------|---------------------------|
| NAMESPACE |     NAME     | TARGET PORT |            URL            |
|-----------|--------------|-------------|---------------------------|
| default   | stockmanager | http/8030   | http://192.168.49.2:32015 |
|-----------|--------------|-------------|---------------------------|
🏃  Starting tunnel for service stockmanager.
|-----------|--------------|-------------|------------------------|
| NAMESPACE |     NAME     | TARGET PORT |          URL           |
|-----------|--------------|-------------|------------------------|
| default   | stockmanager |             | http://127.0.0.1:42767 |
|-----------|--------------|-------------|------------------------|
🎉  Opening service default/stockmanager in default browser...
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
Gtk-Message: 00:18:42.736: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please
try to not load it.
```

Go to browser and access via:

```
127.0.0.1:42767/stocks
```

Check here:



```
▼ 0:
    productId:        "1"
    sku:              "12345678"
    amountAvailable:  5
▼ 1:
    productId:        "2"
    sku:              "34567890"
    amountAvailable:  2
▼ 2:
    productId:        "3"
    sku:              "54326745"
    amountAvailable:  999
▼ 3:
    productId:        "4"
    sku:              "93847614"
    amountAvailable:  0
▼ 4:
    productId:        "5"
    sku:              "11856388"
    amountAvailable:  1
```