

PES UNIVERSITY

UE19CS352

Cloud Computing

Lab - 02

Docker on Linux/Windows/Play-with-Docker

Name : Suhan B Revankar

SRN : PES2UG19CS412

Section : G Section

Deliverables:

The following screenshots are to be submitted:

- **1a.jpg:** Screenshot of running docker hello-world.

```
#####
#                               #
#   WARNING!!!!                #
# This is a sandbox environment. Using personal credentials   #
# is HIGHLY! discouraged. Any consequences of doing so are   #
# completely the user's responsibilities.                      #
#                                                             #
# The FWD team.                                              #
#####
[node1] (local) root@192.168.0.18 ~
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:97a379f4f88575512824f3b352bc03cd75e239179eea0fecc38e597b2209f49a
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

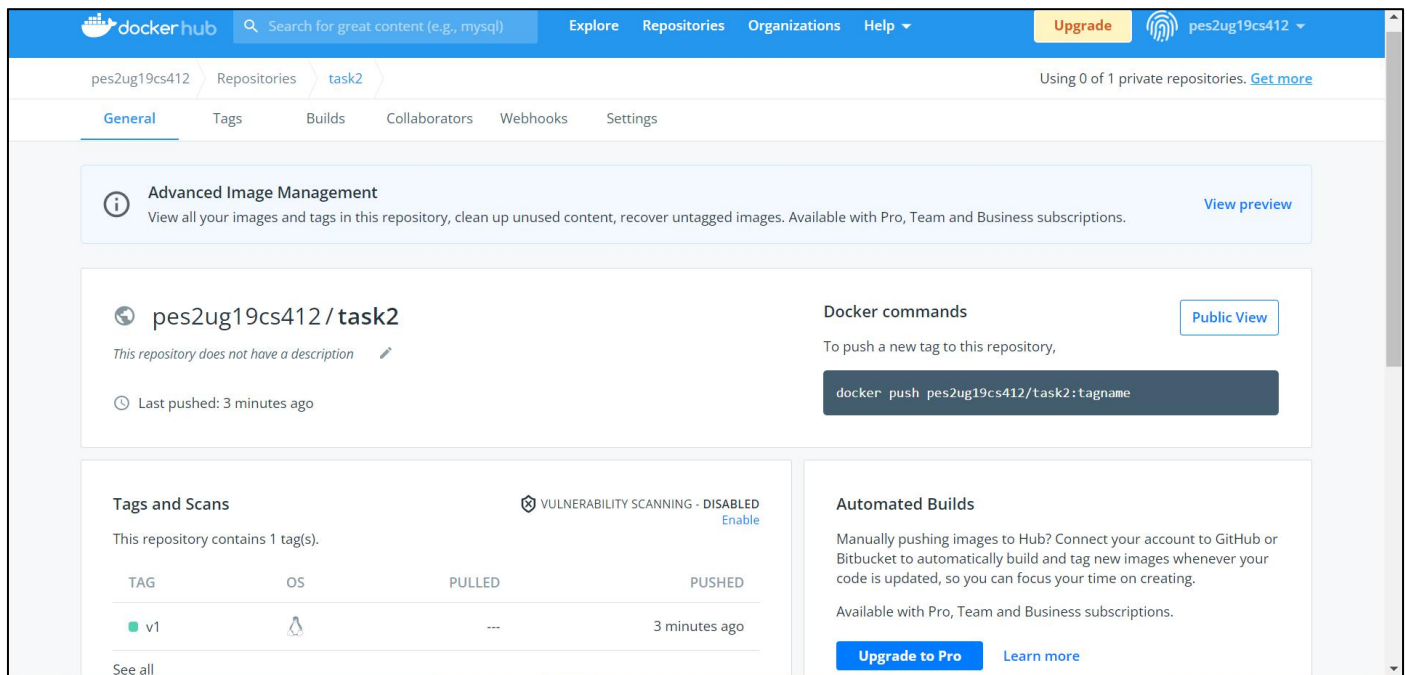
For more examples and ideas, visit:
https://docs.docker.com/get-started/

[node1] (local) root@192.168.0.18 ~
```

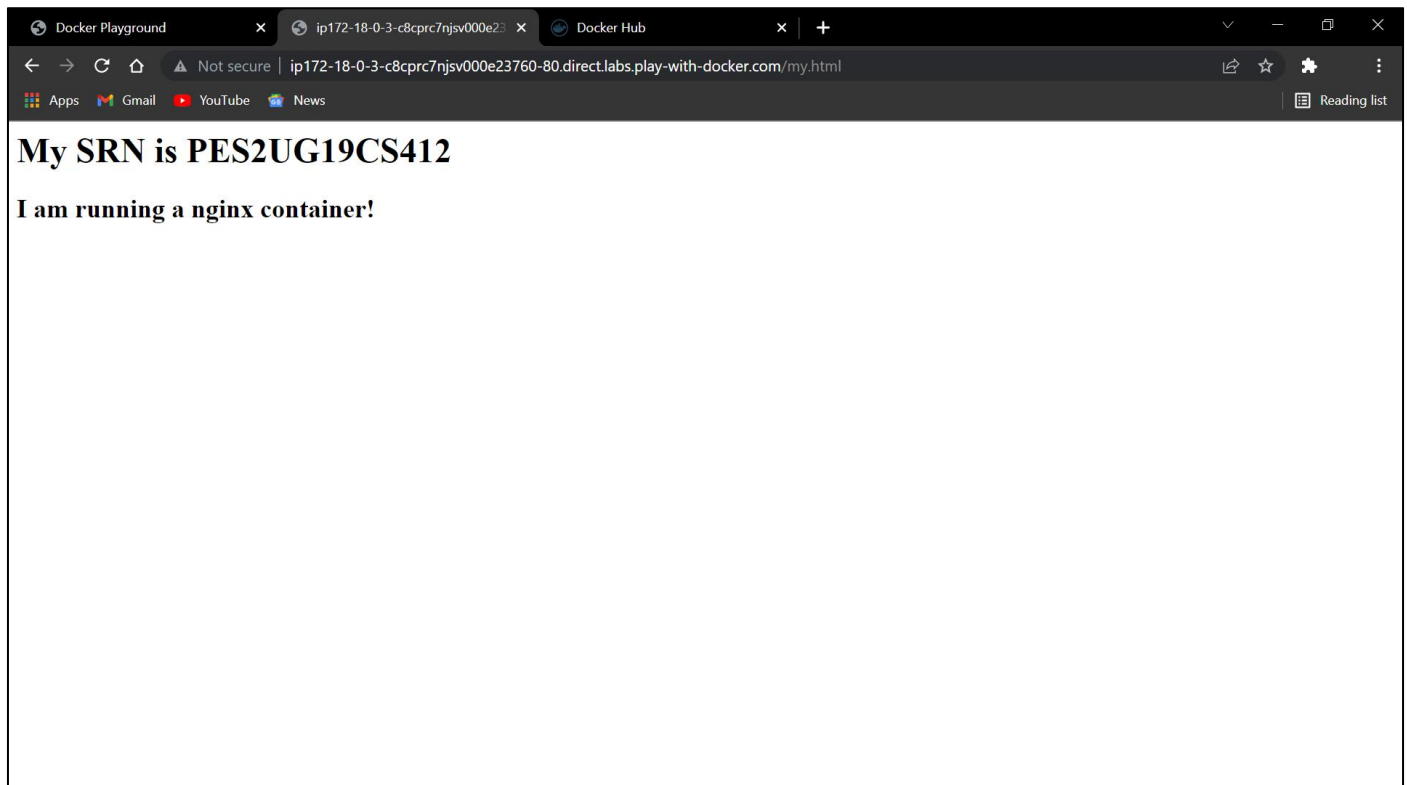
- **2a.jpg:** Screenshot of C Program successfully run inside the container.

```
[node1] (local) root@192.168.0.18 ~
$ nano program.c
[node1] (local) root@192.168.0.18 ~
$ docker build -t task2 .
Sending build context to Docker daemon   47MB
Step 1/6 : FROM ubuntu:18.04
--> dcf4d4bef137
Step 2/6 : RUN apt-get update
--> Using cache
--> 2b74a4b434f8
Step 3/6 : RUN apt-get install gcc -y
--> Using cache
--> e2fc78eb5d9e
Step 4/6 : COPY program.c program.c
--> f9065f444942
Step 5/6 : RUN gcc program.c
--> Running in 9e4cc95e15be
Removing intermediate container 9e4cc95e15be
--> 742f232afdbc
Step 6/6 : CMD ["/a.out"]
--> Running in 5b998723ce8e
Removing intermediate container 5b998723ce8e
--> 29cb380f71c6
Successfully built 29cb380f71c6
Successfully tagged task2:latest
[node1] (local) root@192.168.0.18 ~
$ docker run task2
Running this inside a container !
My SRN is PES2UG19CS412
[node1] (local) root@192.168.0.18 ~
$
```

- **2b.jpg:** Screenshot of the image pushed to Dockerhub.



- **3a.jpg:** Sample.html showing the web page on the browser.



- **3b.jpg:** Screenshot of docker container running nginx

```
[node1] (local) root@192.168.0.18 ~
$ nano my.html
[node1] (local) root@192.168.0.18 ~
$ nano Dockerfile
[node1] (local) root@192.168.0.18 ~
$ docker build -t task3a .
Sending build context to Docker daemon   47MB
Step 1/2 : FROM nginx
--> c316d5a335a5
Step 2/2 : COPY my.html /usr/share/nginx/html
--> a867ab9d55b6
Successfully built a867ab9d55b6
Successfully tagged task3a:latest
[node1] (local) root@192.168.0.18 ~
$ docker run -p 80:80 task3a
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/02/26 04:07:40 [notice] 1#1: using the "epoll" event method
2022/02/26 04:07:40 [notice] 1#1: nginx/1.21.6
2022/02/26 04:07:40 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/02/26 04:07:40 [notice] 1#1: OS: Linux 4.4.0-96-generic
2022/02/26 04:07:40 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/02/26 04:07:40 [notice] 1#1: start worker processes
2022/02/26 04:07:40 [notice] 1#1: start worker process 33
2022/02/26 04:07:40 [notice] 1#1: start worker process 34
2022/02/26 04:07:40 [notice] 1#1: start worker process 35
2022/02/26 04:07:40 [notice] 1#1: start worker process 36
2022/02/26 04:07:40 [notice] 1#1: start worker process 37
2022/02/26 04:07:40 [notice] 1#1: start worker process 38
2022/02/26 04:07:40 [notice] 1#1: start worker process 39
2022/02/26 04:07:40 [notice] 1#1: start worker process 40
2022/02/26 04:08:00 [notice] 1#1: signal 28 (SIGWINCH) received
2022/02/26 04:08:02 [notice] 1#1: signal 28 (SIGWINCH) received
```

- **3c.jpg:** Screenshot of python application successfully writing and reading from the MongoDB database

```

Reading package lists...
Removing intermediate container 37cdfec67830
---> fcd22011afab
Step 3/5 : RUN pip install pymongo
---> Running in c5d5a5c75fa5
Collecting pymongo
  Downloading pymongo-4.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (460 kB)
Installing collected packages: pymongo
Successfully installed pymongo-4.0.1
WARNING: Running pip as the 'root' user can result in broken permissions and c
onflicting behaviour with the system package manager. It is recommended to use
 a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.2.4; however, version 22.0.3 is availabl
e.
You should consider upgrading via the '/usr/local/bin/python -m pip install --
upgrade pip' command.
Removing intermediate container c5d5a5c75fa5
---> 0fac39e90a4a
Step 4/5 : COPY sample.py sample.py
---> fbb7d8f37610
Step 5/5 : CMD ["python","sample.py"]
---> Running in 2d776af38ccb
Removing intermediate container 2d776af38ccb
---> 3e9fcd882966
Successfully built 3e9fcd882966
Successfully tagged task3b:latest
[node1] (local) root@192.168.0.18 ~
$ docker run task3b
Inserted into the MongoDB database!
Fetched from MOngoDB: {'_id': ObjectId('6219acef1e4c7a942fe2ed4b'), 'Name:':
'Suhan B', 'SRN': 'PES2UG19CS412'}
[node1] (local) root@192.168.0.18 ~
$ █

```

- **3d.jpg:** Screenshot showing mongoddb being run within the network(docker command has to be clearly highlighted)

```

[node1] (local) root@192.168.0.18 ~
$ ^C
[node1] (local) root@192.168.0.18 ~
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
[node1] (local) root@192.168.0.18 ~
$ docker network create my-bridge-network
Error response from daemon: network with name my-bridge-network already exists
[node1] (local) root@192.168.0.18 ~
$ docker run -dp 27017:27017 --network=my-bridge-network --name=mongoddb mongo:latest
docker: Error response from daemon: Conflict. The container name "/mongoddb" is already in use by container "a988c89c5a54ff28eba25fc01ab771beffabd09b02aceaf7e82281213d4675dd". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~
$ docker rm /mongoddb
/mongoddb
[node1] (local) root@192.168.0.18 ~
$ docker run -dp 27017:27017 --network=my-bridge-network --name=mongoddb mongo:latest
0df54538e0919a2e90f62cd51727bd1aa54bbc0942b831c8e387cd3904d6bb31
[node1] (local) root@192.168.0.18 ~
$ █

```


- **3e.jpg:** Screenshot showing python file being run within the network and successfully writing and reading from MongoDB(docker command has to be clearly highlighted)

```
[node1] (local) root@192.168.0.18 ~
$ docker run -dp 27017:27017 --network=my-bridge-network --name=mongodb mongo:
latest
aae31b1509eaaf88b0eb7066d69fe73778bf7b784f94df38f43691cef44988ec
[node1] (local) root@192.168.0.18 ~
$ nano sample.py
[node1] (local) root@192.168.0.18 ~
$ docker build -t task3c .
Sending build context to Docker daemon 47.01MB
Step 1/5 : FROM python
---> e4bdbb3cacf1
Step 2/5 : RUN apt-get update
---> Using cache
---> fcd22011afab
Step 3/5 : RUN pip install pymongo
---> Using cache
---> 0fac39e90a4a
Step 4/5 : COPY sample.py sample.py
---> 850b73cebebd
Step 5/5 : CMD ["python","sample.py"]
---> Running in f637e50ce02f
Removing intermediate container f637e50ce02f
---> 0d2a03b2555a
Successfully built 0d2a03b2555a
Successfully tagged task3c:latest
[node1] (local) root@192.168.0.18 ~
$ docker run --network=my-bridge-network task3c
Inserted into the MongoDB database!
Fetched from MOngoDB: {'_id': ObjectId('6219afa80e992e60fc689ba9'), 'Name':
'Suhan B', 'SRN': 'PES2UG19CS412'}
[node1] (local) root@192.168.0.18 ~
$
```

- **4a.jpg:** Screenshot of python-mongodb application running as a docker-compose application(logs of the application)

```
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.811+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":
main", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
"}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.814+00:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":
main", "msg":"No TransportLayer configured during NetworkInterface startup"}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.814+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":
main", "msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFast
OpenClient, and tcpFastOpenQueueSize."}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.817+00:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":
main", "msg":"No TransportLayer configured during NetworkInterface startup"}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.817+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":
main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationDonorService", "n
s":"config.tenantMigrationDonors"}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.817+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":
main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationRecipientService
", "ns":"config.tenantMigrationRecipients"}}
```


- **4b.jpg:** Screenshot of 3 python application writes and reads from MongoDB after scaling the python application.

```
[node1] (local) root@192.168.0.18 ~
$ docker-compose up --scale pycode=3
Starting root_mongodb_1 ... done
Starting root_pycode_1 ... done
Creating root_pycode_2 ... done
Creating root_pycode_3 ... done
Attaching to root_mongodb_1, root_pycode_1, root_pycode_3, root_pycode_2
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.811+00:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}}}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.811+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.814+00:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.814+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"main","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.817+00:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.817+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","ns":"config.tenantMigrationDonors"}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:10.817+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","ns":"config.tenantMigrationRecipients"}}
```

1st python application writes and reads from mongoDB after scaling the python application

```
:"PyMongo","version":"4.0.1"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"4.4.0-96-generic"},"platform":"CPython 3.10.2.final.0"}}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.493+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn8","msg":"client metadata","attr":{"remote":"172.20.0.3:51964","client":"conn8","doc":{"driver":{"name":"PyMongo","version":"4.0.1"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"4.4.0-96-generic"},"platform":"CPython 3.10.2.final.0"}}}}
pycode_1 | Inserted into the MongoDB database!
pycode_1 | Fetched from MongoDB: {'_id': ObjectId('6219b11e69fe8f8c0c820155'), 'Name': 'Suhan B', 'SRN': 'PES2UG19CS412'}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.919+00:00"},"s":"I", "c":"-", "id":20883, "ctx":"conn1","msg":"Interrupted operation as its client disconnected","attr":{"opId":13}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.920+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn1","msg":"Connection ended","attr":{"remote":"172.20.0.4:42564","uid":"0cfde78d-6e97-4f86-87c1-4c08be2aa76a","connectionId":1,"connectionCount":6}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.928+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn2","msg":"Connection ended","attr":{"remote":"172.20.0.4:42566","uid":"c2646049-a6c2-4ce6-b822-d59fc3cc43d8","connectionId":2,"connectionCount":7}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.928+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn3","msg":"Connection ended","attr":{"remote":"172.20.0.4:42568","uid":"19b29ea7-d3f8-46f6-82f4-f9c708a893ce","connectionId":3,"connectionCount":8}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.938+00:00"},"s":"I", "c":"-", "id":20883, "ctx":"conn4","msg":"Interrupted operation as its client disconnected","attr":{"opId":19}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.939+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn4","msg":"Connection ended","attr":{"remote":"172.20.0.5:37636","uid":"c9885d4d-358a-448d-bed8-106ec529466b","connectionId":4,"connectionCount":5}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.950+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn6","msg":"Connection ended","attr":{"remote":"172.20.0.5:37640","uid":"02cfc014-8c42-405f-8fae-702c030"}}
```


2nd python application writes and reads from mongoDB after scaling the python application

```
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.5:37640", "uuid": "02cfc014-8c42-405f-8fae-702c03092350", "connectionId": 6, "connectionCount": 6}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.441+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
conn6", "msg": "client metadata", "attr": {"remote": "172.20.0.5:37640", "client": "conn6", "doc": {"driver": {"name":
:"PyMongo", "version": "4.0.1"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "version": "4.4.0-
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
pycode_2 | Inserted into the MongoDB database!
pycode_2 | Fetched from MOngoDB: {'_id': ObjectId('6219b11e69fe8f8c0c820155'), 'Name': 'Suhan B', 'SRN'
: 'PES2UG19CS412'}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.483+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.3:51962", "uuid": "aff0668-f06d-4464-b4ae-4
6a6b4b72054", "connectionId": 7, "connectionCount": 7}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.484+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
conn7", "msg": "client metadata", "attr": {"remote": "172.20.0.3:51962", "client": "conn7", "doc": {"driver": {"name":
:"PyMongo", "version": "4.0.1"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "version": "4.4.0-
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.488+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.3:51964", "uuid": "2051bb05-9704-4ae3-8283-5
3d180b33d66", "connectionId": 8, "connectionCount": 8}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.491+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.3:51966", "uuid": "2425b77a-3b5f-4a98-b54e-4
eb7e595545b", "connectionId": 9, "connectionCount": 9}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.491+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
conn9", "msg": "client metadata", "attr": {"remote": "172.20.0.3:51966", "client": "conn9", "doc": {"driver": {"name":
:"PyMongo", "version": "4.0.1"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "version": "4.4.0-
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.493+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
```

3rd python application writes and reads from mongoDB after scaling the python application

```
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
pycode_3 | Inserted into the MongoDB database!
pycode_3 | Fetched from MOngoDB: {'_id': ObjectId('6219b11e69fe8f8c0c820155'), 'Name': 'Suhan B', 'SRN'
: 'PES2UG19CS412'}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.434+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.5:37636", "uuid": "c9885d4d-358a-448d-bed8-1
06ec529466b", "connectionId": 4, "connectionCount": 4}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.435+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
conn4", "msg": "client metadata", "attr": {"remote": "172.20.0.5:37636", "client": "conn4", "doc": {"driver": {"name":
:"PyMongo", "version": "4.0.1"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "version": "4.4.0-
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.438+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.5:37638", "uuid": "70219305-0221-4e1c-8f1f-3
b9b8accf9cb", "connectionId": 5, "connectionCount": 5}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.440+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
conn5", "msg": "client metadata", "attr": {"remote": "172.20.0.5:37638", "client": "conn5", "doc": {"driver": {"name":
:"PyMongo", "version": "4.0.1"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "version": "4.4.0-
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.440+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":
listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.5:37640", "uuid": "02cfc014-8c42-405f-8fae-7
02c03092350", "connectionId": 6, "connectionCount": 6}}
mongodb_1 | {"t":{"$date":"2022-02-26T04:51:13.441+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":
conn6", "msg": "client metadata", "attr": {"remote": "172.20.0.5:37640", "client": "conn6", "doc": {"driver": {"name":
:"PyMongo", "version": "4.0.1"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "version": "4.4.0-
96-generic"}, "platform": "CPython 3.10.2.final.0"}}}
pycode_2 | Inserted into the MongoDB database!
pycode_2 | Fetched from MOngoDB: {'_id': ObjectId('6219b11e69fe8f8c0c820155'), 'Name': 'Suhan B', 'SRN'
```
