

# Cloud Computing Lab 5 Report

PES1201800366

Aditeya Baral

## 1 Installing Docker Engine on EC2 Instance

```
ubuntu@ip-172-31-78-204:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:95dddb6c31407e84e91a986b004aee40975cb0bda14b5949f6faac5d2deadb4b9
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/

ubuntu@ip-172-31-78-204:~$
```

Figure 1: Running Docker Hello-World

## 2 Docker Images and Docker Files

```
ubuntu@ip-172-31-78-204:~/task 2$ docker run pes1201800366
Running this inside a docker container!
My SRN is PES1201800366
ubuntu@ip-172-31-78-204:~/task 2$
```

Figure 2: Running C Program inside Container

### 3 Exposing Ports and Docker Networks



Figure 3: Accessing web page on browser

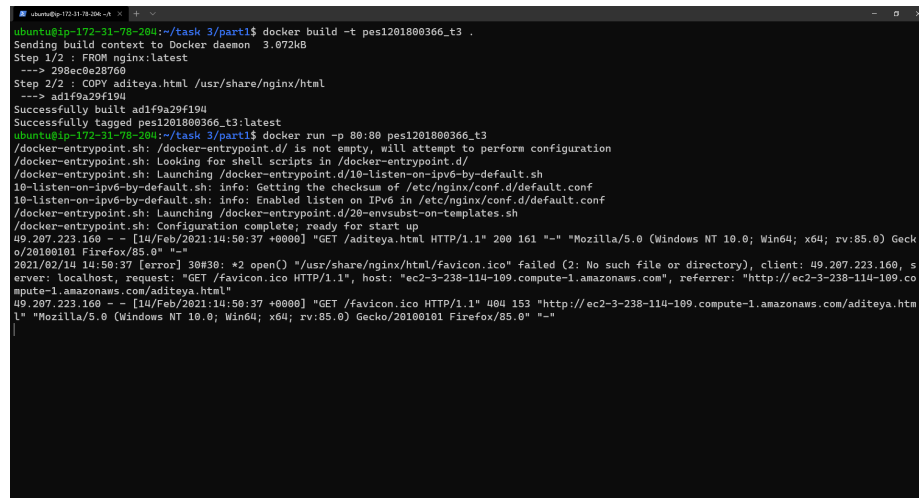


Figure 4: Docker container running nginx

```

ubuntu@ip-172-31-78-204:~/task 3/part2$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
30d264a28c4a   mongo:latest "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes   0.0.0.0:27017->27017/tcp  adoring_goldwasser

ubuntu@ip-172-31-78-204:~/task 3/part2$ docker run pes1201800366_t3_p2
Inserted into the MongoDB database!
Fetched from MongoDB: {'_id': ObjectId('60293a43c5afe0090f61d54b'), 'Name': 'ADITEYABARAL', 'SRN': 'PES1201800366'}
ubuntu@ip-172-31-78-204:~/task 3/part2$

```

Figure 5: Read and write from MongoDB

```

ubuntu@ip-172-31-78-204:~/task 3/part3$ docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
946c22860211   bridge   bridge  local
c1ac8bcc843c   host     host    local
a5f208865a0c   none     null    local

ubuntu@ip-172-31-78-204:~/task 3/part3$ docker network create --driver=bridge my-bridge-network
320050b042f0fc0dc26b5befb2afe21f83c02c38f6840c1e4fae9d4083cdfd083

ubuntu@ip-172-31-78-204:~/task 3/part3$ docker run -d -p 27107:27107 --network=my-bridge-network --name=mongodb
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

ubuntu@ip-172-31-78-204:~/task 3/part3$ docker run -d -p 27107:27107 --network=my-bridge-network --name=mongodb mongo:latest
62fd4d021e8b5a48b49f405151bd5295564a325b890192d769bb8691111af6bb7
ubuntu@ip-172-31-78-204:~/task 3/part3$

```

Figure 6: MongoDB running within network

```

ubuntu@ip-172-31-78-204:~/task 3/part3$ docker run --network=my-bridge-network pes1201800366_t3_p3
Inserted into the MongoDB database!
Fetched from MongoDB: {'_id': ObjectId('60293c29cae39df06e8d5d06'), 'Name': 'ADITEYABARAL', 'SRN': 'PES1201800366'}
ubuntu@ip-172-31-78-204:~/task 3/part3$

```

Figure 7: Python file reading and writing from MongoDB within network

## 4 Docker Compose

```
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.539+00:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"
/tmp/mongodb-27017.sock"}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.539+00:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"
0.0.0.0"}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.539+00:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"
port":27017
"ssl":"off"}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.559+00:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"LogicalSessionCacheRefresh","msg":"Index build: do
ne building","attr":{"buildUUID":"null","namespace":"config.system.sessions","index":{"id":"commitTimestamp":{"t":{"i":0}}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.559+00:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"LogicalSessionCacheRefresh","msg":"Index build: do
ne building","attr":{"buildUUID":"null","namespace":"config.system.sessions","index":{"lsidTTLIndex","commitTimestamp":{"t":{"i":0}}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.569+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"rem
ote":"172.19.0.3:46034","connectionId":1,"connectionCount":1}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.569+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"client metadata","attr":{"remote":"1
72.19.0.3:46034","client":{"conn1","doc":{"driver":{"name":"PyMongo","version":"3.11.2"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version
":"5.4.0-1037-aws"},"platform":"CPython 3.9.1.final.0"}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.569+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"rem
ote":"172.19.0.3:46036","connectionId":2,"connectionCount":2}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.569+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"rem
ote":"172.19.0.3:46038","connectionId":3,"connectionCount":3}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.569+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn2","msg":"client metadata","attr":{"remote":"1
72.19.0.3:46036","client":{"conn2","doc":{"driver":{"name":"PyMongo","version":"3.11.2"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version
":"5.4.0-1037-aws"},"platform":"CPython 3.9.1.final.0"}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.569+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn3","msg":"client metadata","attr":{"remote":"1
72.19.0.3:46038","client":{"conn3","doc":{"driver":{"name":"PyMongo","version":"3.11.2"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version
":"5.4.0-1037-aws"},"platform":"CPython 3.9.1.final.0"}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.576+00:00"},"s":"I", "c":"STORAGE", "id":20320, "ctx":"conn3","msg":"createCollection","attr":{"namespace
":"sample.db.sample.collection","uidDisposition":"generated","uid":{"build":{"id":"60d84b6d-9e9d-4b3a-8022-076d137f2513"},"options":{"i}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:54.576+00:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"conn3","msg":"Index build: done building","attr":{"
"buildUUID":"null","namespace":"sample.db.sample.collection","index":{"id":"commitTimestamp":{"t":{"i":0}}}}}
pycode_1 | Inserted into the MongoDB database
pycode_1 | Fetched from MongoDB: {'_id': ObjectId('60293d7e68981c2b1b2e5f2'), 'Name': 'ADITEVABARAL', 'SRN': 'PES1201800366'}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:55.363+00:00"},"s":"I", "c":"CONTROL", "id":20883, "ctx":"conn1","msg":"Interrupted operation as its client
disconnected","attr":{"opId":19}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:55.363+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn1","msg":"Connection ended","attr":{"remote":"
172.19.0.3:46034","connectionId":1,"connectionCount":2}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:55.367+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn3","msg":"Connection ended","attr":{"remote":"
172.19.0.3:46038","connectionId":3,"connectionCount":1}}
mongo_1 | {"t":{"$date":"2021-02-14T15:10:55.368+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn2","msg":"Connection ended","attr":{"remote":"
172.19.0.3:46036","connectionId":2,"connectionCount":0}}
ubuntu_pycode_1 exited with code 0
```

Figure 8: Python-MongoDB application running as docker-compose

```
ubuntu@ip-172-31-78-204:~$ docker-compose up --scale pycode=3
Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/compose-native-build/
Creating network ubuntu_default with the default driver
Creating ubuntu_mongodb_1 ... done
Creating ubuntu_pycode_1 ... done
Creating ubuntu_pycode_2 ... done
Creating ubuntu_pycode_3 ... done
Attaching to ubuntu_mongodb_1, ubuntu_pycode_2, ubuntu_pycode_1, ubuntu_pycode_3
pycode_2 | Inserted into the MongoDB database
pycode_2 | Fetched from MongoDB: {'_id': ObjectId('60293d7e68981c2b1b2e5f2'), 'Name': 'ADITEVABARAL', 'SRN': 'PES1201800366'}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.663+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"Automatically disabling T
LS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.665+00:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main","msg":"No TransportLayer configu
red during NetronInterface startup"}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.666+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"main","msg":"Implicit TCP FastOpen una
vailable. IF TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.667+00:00"},"s":"I", "c":"STORAGE", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting
","attr":{"pid":1,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"f7162a70d4f01"}}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.667+00:00"},"s":"I", "c":"CONTROL", "id":23003, "ctx":"initandlisten","msg":"Build Info","att
r":{"buildInfo":{"version":"4.4.3","gitVersion":"913d6b62acfb34dddb1b16f4161360acd8fd13","opensslVersion":"OpenSSL 1.1.1 11 Sep 2018","modules
":["allocator","tcmalloc","environment":{"distmod":"ubuntu1804","distarch":"x86_64","target_arch":"x86_64"}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.667+00:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Operating System
","attr":{"os":{"name":"Ubuntu","version":"18.04"}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.667+00:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Options set by c
ommand line","attr":{"options":{"net":{"bindip":"*"}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.675+00:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS fi
lesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem","tags":["startupw
arnings"]}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:56.677+00:00"},"s":"I", "c":"STORAGE", "id":22215, "ctx":"initandlisten","msg":"Opening WiredTig
er","attr":{"config":{"createCacheSize":1048576,"sessionMax":33000,"eviction":{"threads_min":0,"threads_max":0,"config_base":false,"statistics":{"fast":e
nabled,true,"archive":true,"path":journal,"compressor":"snappy"},"file_manager":{"close_idle_time":100000,"close_scan_interval":10,"close_handle_minumum":250},"st
atistics_log":{"wait":0,"verbose":["recovery_progress","checkpoint_progress","compact_progress"]}}}}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:57.602+00:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger messa
ge","attr":{"message":"[1613315577:602492][1:0x7f58bf522ac0], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery tim
estamp: (0, 0)"}
mongo_1 | {"t":{"$date":"2021-02-14T15:12:57.602+00:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger messa
```

Figure 9: Application read and write from MongoDB

```
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.135+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn4", "msg":"client metadata", "attr":{"remote":"172.20.0.5:48252", "client":"conn4", "doc":{"driver":{"name":"PyMongo", "version":"3.11.2"}, "os":{"type":"Linux", "name":"Linux", "architecture":"x86_64", "version":"5.4.0-1037-aws"}, "platform":"CPython 3.9.1.final.0"}}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.136+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener", "msg":"Connection accepted", "attr":{"remote":"172.20.0.5:48254", "connectionId":5, "connectionCount":5}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.137+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn5", "msg":"client metadata", "attr":{"remote":"172.20.0.5:48254", "client":"conn5", "doc":{"driver":{"name":"PyMongo", "version":"3.11.2"}, "os":{"type":"Linux", "name":"Linux", "architecture":"x86_64", "version":"5.4.0-1037-aws"}, "platform":"CPython 3.9.1.final.0"}}}]
pycode_1 | Inserted into the MongoDB database!
pycode_1 | Fetched from MongoDB: {'_id': ObjectId('60293df9bf641783491ceef5'), 'Name': 'ADITEYABARAL', 'SRN': 'PES1201800366'}
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.147+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener", "msg":"Connection accepted", "attr":{"remote":"172.20.0.4:48800", "connectionId":6, "connectionCount":6}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.148+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn6", "msg":"client metadata", "attr":{"remote":"172.20.0.4:48800", "client":"conn6", "doc":{"driver":{"name":"PyMongo", "version":"3.11.2"}, "os":{"type":"Linux", "name":"Linux", "architecture":"x86_64", "version":"5.4.0-1037-aws"}, "platform":"CPython 3.9.1.final.0"}}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.149+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener", "msg":"Connection accepted", "attr":{"remote":"172.20.0.4:48802", "connectionId":7, "connectionCount":7}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.150+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn7", "msg":"client metadata", "attr":{"remote":"172.20.0.4:48802", "client":"conn7", "doc":{"driver":{"name":"PyMongo", "version":"3.11.2"}, "os":{"type":"Linux", "name":"Linux", "architecture":"x86_64", "version":"5.4.0-1037-aws"}, "platform":"CPython 3.9.1.final.0"}}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.150+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener", "msg":"Connection accepted", "attr":{"remote":"172.20.0.4:48804", "connectionId":8, "connectionCount":8}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.151+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn8", "msg":"client metadata", "attr":{"remote":"172.20.0.4:48804", "client":"conn8", "doc":{"driver":{"name":"PyMongo", "version":"3.11.2"}, "os":{"type":"Linux", "name":"Linux", "architecture":"x86_64", "version":"5.4.0-1037-aws"}, "platform":"CPython 3.9.1.final.0"}}}]
pycode_3 | Inserted into the MongoDB database!
pycode_3 | Fetched from MongoDB: {'_id': ObjectId('60293df9bf641783491ceef5'), 'Name': 'ADITEYABARAL', 'SRN': 'PES1201800366'}
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.347+00:00"},"s":"I", "c":"-", "id":20883, "ctx":"conn1", "msg":"Interrupted operation as its client disconnected", "attr":{"opId":17}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.348+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn1", "msg":"Connection ended", "attr":{"remote":"172.20.0.3:60990", "connectionId":1, "connectionCount":7}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.353+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn2", "msg":"Connection ended", "attr":{"remote":"172.20.0.3:60992", "connectionId":2, "connectionCount":6}}]
mongo_1 | [{"t":{"$date":"2021-02-14T15:12:58.353+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn3", "msg":"Connection ended", "attr":{"remote":"172.20.0.3:60994", "connectionId":3, "connectionCount":5}}]
ubuntu.pycode_2 exited with code 0
```

Figure 10: Application read and write from MongoDB