

# Installation Instruction for Linux

---

The document contains detailed instructions to install the recommendation system project on a Linux OS.

This document will consider Ubuntu 18.04 LTS as an example Linux OS.

## Step 1: Install Docker

The following links would be helpful to guide you in smooth installation of docker on your Linux OS.

<https://docs.docker.com/engine/install/>

<https://docs.docker.com/engine/install/ubuntu/>

```
rahul@rahul-VirtualBox:~$ docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "/home/rahul/.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
```

Check if you get similar output as shown above.

**Try to run a sample hello-world container. Follow these steps:**

## **##DOCKER PERMISSION DENIED ISSUE**

**#NOTE :** To run a container or perform operations on a existing container the following operations need to be done in the following order:

1. Create a group docker if it does not exist.

Command: **sudo groupadd docker**

```
rahul@rahul-VirtualBox:~$ sudo groupadd docker
[sudo] password for rahul:
groupadd: group 'docker' already exists
rahul@rahul-VirtualBox:~$
```

2. Add your user to the docker group.

Command: **sudo usermod -aG docker \$USER**

```
rahul@rahul-VirtualBox:~$ sudo usermod -aG docker $USER
rahul@rahul-VirtualBox:~$
```

3. Change to a new group.

Command: **newgrp docker**

```
rahul@rahul-VirtualBox:~$ sudo groupadd docker
[sudo] password for rahul:
groupadd: group 'docker' already exists
rahul@rahul-VirtualBox:~$ sudo usermod -aG docker $USER
rahul@rahul-VirtualBox:~$ newgrp docker
rahul@rahul-VirtualBox:~$
```

After these 3 commands you can run any docker containers **IN THAT DIRECTORY**. When in a new directory make sure to follow these steps once.

**Follow the above steps whenever you get a**

*“Got permission denied when connecting to docker daemon ..”*

**Error.**

Now that the permission denied issue is sorted you can try running a sample hello world container by pulling the hello world image from docker hub.

Command: docker run hello-world

```
rahul@rahul-VirtualBox:~$ docker run hello-world

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/

rahul@rahul-VirtualBox:~$
```

## Step 2: Build image

Once docker is installed and working fine we need to open the terminal in the downloaded project folder:

```
rahul@rahul-VirtualBox:~$ ls
Desktop      Music          Templates
Documents    Pictures       'Untitled Document'
Downloads    Public         Videos
examples.desktop  recommendation_system_project-master
rahul@rahul-VirtualBox:~$ cd recommendation_system_project-master
rahul@rahul-VirtualBox:~/recommendation_system_project-master$
```

Once inside we need to build the image using docker –compose

**#NOTE :** Here docker compose needs to be installed as follows:

Command: `sudo apt install docker-compose`

```
rahul@rahul-VirtualBox:~$ ls
Desktop      Music          Templates
Documents    Pictures       'Untitled Document'
Downloads    Public         Videos
examples.desktop  recommendation_system_project-master
rahul@rahul-VirtualBox:~$ cd recommendation_system_project-master
rahul@rahul-VirtualBox:~/recommendation_system_project-master$ sudo apt install
docker-compose
[sudo] password for rahul:
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker-compose is already the newest version (1.17.1-2).
The following packages were automatically installed and are no longer required:
  libcapnp-0.5.3 libhardware2 libhybris libhybris-common1 libllvm6.0 libmedia1
  libwebpdemux1 linux-generic linux-generic-hwe-16.04 linux-headers-generic
  linux-headers-generic-hwe-16.04 linux-image-generic-hwe-16.04
  snapd-login-service xserver-xorg-input-evdev-hwe-16.04
  xserver-xorg-input-synaptics-hwe-16.04 xserver-xorg-legacy-hwe-16.04
  xserver-xorg-video-amdgpu-hwe-16.04 xserver-xorg-video-qxl-hwe-16.04
  xserver-xorg-video-vesa-hwe-16.04
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
rahul@rahul-VirtualBox:~/recommendation_system_project-master$
```

It had been already installed in my system. Now we can proceed to build the image of our application

Command: docker-compose build

```
rahul@rahul-VirtualBox:~/recommendation_system_project-master$ docker-compose build
mongo uses an image, skipping
Building web
Step 1/7 : FROM python:3
---> 7f5b6ccd03e9
Step 2/7 : ENV PYTHONUNBUFFERED 1
---> Running in 3cd12079935f
Removing intermediate container 3cd12079935f
---> 215d4a77a946
Step 3/7 : RUN mkdir /code
---> Running in b37c6a5883c8
Removing intermediate container b37c6a5883c8
---> f517c03ed975
Step 4/7 : WORKDIR /code
---> Running in 647eb028d90a
Removing intermediate container 647eb028d90a
---> 75d203e38b7d
Step 5/7 : ADD . /code
---> d98b41970cff
Step 6/7 : COPY . /code
---> f105e73086be
Step 7/7 : RUN pip install -r requirements.txt
---> Running in 4bbe29f15ee7
```

```
Created wheel for pyyaml: filename=PyYAML-5.3.1-cp38-cp38-linux_x86_64.whl size=572459 sha256=4b839ec60d689ce027a83248792def264353b064aea0b0825afbade83e
Stored in directory: /root/.cache/pip/wheels/13/90/db/290ab3a34f2ef0b5a0f89235dc2d40fea83e77de8f05c
Building wheel for et-xmlfile (setup.py): started
Building wheel for et-xmlfile (setup.py): finished with status 'done'
Created wheel for et-xmlfile: filename=et_xmlfile-1.0.1-py3-none-any.whl size=8915 sha256=c8c3236ab4e5e484ba3c65b60d987655bf0e74ea1847647fe8e7c
Stored in directory: /root/.cache/pip/wheels/6e/df/38/abda47b884e3e25f9f9b6430e5ce44c47670758a50759
Successfully built django bson diff-match-patch markuppy odfpy pyyaml et-xmlfile
Installing collected packages: sqlparse, pytz, Django, six, python-dateutil, bson, pymongo, dataclasses, django, gunicorn, dnspython, django-crispy-forms, markuppy, defusedxml, odfpy, xlrd, xlwt, jdcal, et-xmlfile, openpyxl, pyyaml, tablib, diff-match-patch, django-import-export
Successfully installed Django-2.2 bson-0.5.8 dataclasses-0.6 defusedxml-0.6.0 diff-match-patch-2018.12.31 django-crispy-forms-1.9.1 django-import-export-2.0.2 django-1.3.0 dnspython-1.16.0 et-xmlfile-1.0.1 gunicorn-20.0.4 jdcal-1.4.1 markuppy-1.14 odfpy-1.4.1 openpyxl-3.0.4 pymongo-3.10.1 python-dateutil-2.8.0 pytz-2020.1 pyyaml-5.3.1 six-1.15.0 sqlparse-0.2.4 tablib-2.0.0 xlrd-1.2.0 xlwt-1.3.0
Removing intermediate container 4bbe29f15ee7
---> 72ee4806c678
Successfully built 72ee4806c678
Successfully tagged recommendationssystemprojectmaster_web:latest
mongo-express uses an image, skipping
rahul@rahul-VirtualBox:~/recommendation_system_project-master$
```

The docker image is successfully created.

### Step 3: Running the image

You can run *docker images* to check if the new image has been added. Now we need to run this image.

Command: `docker-compose up`

```
rahul@rahul-VirtualBox:~/recommendation_system_project-master$ docker-compose up
Creating mongo-express ...
Creating database ...
Creating mongo-express
Creating database ... done
Creating recommend_app ...
Creating recommend_app ... done
Attaching to mongo-express, database, recommend_app
mongo-express | Waiting for mongo:27017...
mongo-express | /docker-entrypoint.sh: connect: Connection refused
mongo-express | /docker-entrypoint.sh: line 14: /dev/tcp/mongo/27017: Connection refused
mongo-express | Fri Jun 26 06:38:26 UTC 2020 retrying to connect to mongo:27017 (2/5)
mongo-express | /docker-entrypoint.sh: connect: Connection refused
mongo-express | /docker-entrypoint.sh: line 14: /dev/tcp/mongo/27017: Connection refused
mongo-express | Fri Jun 26 06:38:27 UTC 2020 retrying to connect to mongo:27017 (3/5)
mongo-express | /docker-entrypoint.sh: connect: Connection refused
mongo-express | /docker-entrypoint.sh: line 14: /dev/tcp/mongo/27017: Connection refused
mongo-express | Fri Jun 26 06:38:28 UTC 2020 retrying to connect to mongo:27017 (4/5)
mongo-express | /docker-entrypoint.sh: connect: Connection refused
mongo-express | /docker-entrypoint.sh: line 14: /dev/tcp/mongo/27017: Connection refused
mongo-express | Fri Jun 26 06:38:29 UTC 2020 retrying to connect to mongo:27017 (5/5)
mongo-express | /docker-entrypoint.sh: connect: Connection refused
mongo-express | /docker-entrypoint.sh: line 14: /dev/tcp/mongo/27017: Connection refused
database | 2020-06-26T06:38:27.957+0000 I CONTROL [main] Automatically disabling TLS 1.0, to
```

Keep this terminal running.

Now, we need **to import the items from the dataset** into the database container using `script.sh`

We need to enable permission for the script first.

Command: `sudo chmod 774 script.sh`

```
rahul@rahul-VirtualBox:~/recommendation_system_project-master$ sudo chmod 774 script.sh
[sudo] password for rahul:
rahul@rahul-VirtualBox:~/recommendation_system_project-master$ ./script.sh
```

Then run the script using `./script.sh`

## Step 4: Check the output

Go to **localhost:8081** to access mongo-express.

A database named “recommendation\_system-db” will be created. Inside that database, collections containing movie, anime and books datasets can be viewed.

This makes sure that we imported the dataset using script.sh.

To access the main webpage, go to:

**Localhost:8000/recommendations\_webapp\_home/**

Suggestion:

On a Linux OS, to get the best experience of the web application, try and get the sidebar to the bottom and give full screen access to your browser.

To get the side bar on the bottom, type this command in the terminal:

```
gsettings set com.canonical.Unity.Launcher launcher-position  
Bottom
```

Preferred browsers:

Chrome, Opera, Firefox