How to install OpenedX on ubuntu20.04 with help of tutor website

Step 1: Update System Packages

Start by updating the system packages to their latest versions using the following commands:-

sudo apt update

sudo apt upgrade

Step 2: Install Prerequites

Open Edx requires certain prerequisites to be installed. Using the following commands to install them:-

sudo apt install pip3

sudo apt install python3-dev python3-pip python3-yaml npm gcc g++ make python3-venv

sudo apt install python3-dev python3-setuptools python3-venv build-essential software-properties-common curl git libxml2 libxmlsec1 libxmlsec1-dev python3-lxml libyaml-dev wget

Step 3: Create a New User

For security purposes, its recommended to create a new system user to run Open edX. You can create a new user named “edxapp” using the following command:-

sudo adduser --disabled-password --gecos "" edxapp

Step 4 : Install OpenedX

Switch to the “edXapp” user and navigate to the home directory:-

sudo su - edxapp

This is not a necessory

cd ~

clone the Open edX repository from gitHub or gitlab:-

git clone <https://github.com/edx/edx-platform.git> or yourself

Create and activate a virtual environment:-

python3 -m venv edxapp\_env

source edxapp\_env/bin/activate

install the required Python Packages:-

cd edx-platform

pip3 install -r requirements/edx/base.txt

Generate the configuration Files:

make lms-envs

Run the following command to install the assets:-

paver update\_assets lms --settings=devstack\_docker

Sometims will coming error that time we install docker with the help off tutor Website

Step 5: Install docker (using docker documentation).

Overview

Select Linux

Ubuntu

Step 6: Install Docker desktop

sudo apt install gnome-terminal

1.Uninstall the tech preview or beta version of docker desktop for linx

sudo apt remove docker-desktop

2.For a complete cleanup, remove configuration and data files at

rm -r $HOME/.docker/desktop

sudo rm /usr/local/bin/com.docker.cli

sudo apt purge docker-desktop

3.Set up the docker repository

* Update the apt package index and install packages to allow apt to use a repository over HTTPS

sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

* Add Docker Official GPG key

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

* Use following commands to set up the repository

echo \

"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

"$(. /etc/os-release && echo "$VERSION\_CODENAME")" stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

4. Install the package with apt as follows

sudo apt-get update

sudo apt-get install ./docker-desktop-<version>-<arch>.deb

start docker desktop :-

systemctl --user start docker-desktop

Check to the docker version:-

docker compose version

Docker Compose version v2.17.3

docker --version

Docker version 23.0.5, build bc4487a

docker version

Client: Docker Engine - Community

Cloud integration: v1.0.31

Version: 23.0.5

API version: 1.42

Containerised application:-

Clone the getting-started repository using the following command

git clone <https://github.com/docker/getting-started.git>

Build the apps container image:-

1. Change the directory—

cd /path/to/app

1. Create an empty file named Dockerfile.

touch Dockerfile

Using a tect editor or code editor, add the following contents to the Dokerfile:-

# syntax=docker/dockerfile:1

FROM node:18-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]

EXPOSE 3000

1. Build the container image (sometime will happen error that the time if we move next step)

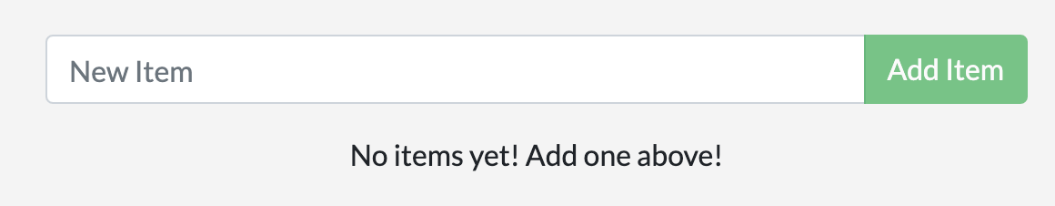
docker build -t getting-started .

Start an app container:

1. Start your container using the docker run command:-

docker run -dp 127.0.0.1:3000:3000 getting-started

1. After a few seconds, open your web browser to <http://localhost:3000>



1. List your container using this command docker ps

docker ps

O/p:

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

df784548666d getting-started "docker-entrypoint.s…" 2 minutes ago Up 2 minutes 127.0.0.1:3000->3000/tcp priceless\_mcclintock

Update the source code

1. In the src/static/js/app.js file, update line 56 to use the new empty text.

- <p className="text-center">No items yet! Add one above!</p>

+ <p className="text-center">You have no todo items yet! Add one above!</p>

1. Build your update version of the image, using the same docker build command

docker build -t getting-started .

1. Start a new conatiner using the update code.

docker run -dp 127.0.0.1:3000:3000 getting-started

You probably saw an error like this (the IDs will be different):-

docker: Error response from daemon: driver failed programming external connectivity on endpoint laughing\_burnell

(bb242b2ca4d67eba76e79474fb36bb5125708ebdabd7f45c8eaf16caaabde9dd): Bind for 127.0.0.1:3000 failed: port is already allocated.

Then if you remove old container

1. Get the ID of the container by using the docker ps command.

docker ps

1. Use the docker stop command to stop the container. Replace conatiner ID.

docker stop <the-container-id>

1. Onece the conatiner has stoped, you can remove it by using the docker rm command.

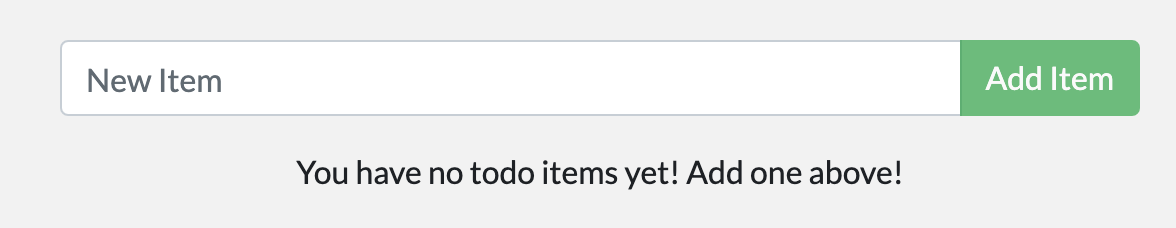
docker rm <the-container-id>

Start the updated app conatiner

1. Now start yoyr update app using the docker run command.

docker run -dp 127.0.0.1:3000:3000 getting-started

1. Refresh your browser on http://localhost:3000 and you see your update help text.



Create a repository:-

1. Login docker hub
2. Create repository

Push the image:

docker push docker/getting-started

The push refers to repository [docker.io/docker/getting-started]

An image does not exist locally with the tag: docker/getting-started

docker tag getting-started YOUR-USER-NAME/getting-started

docker push YOUR-USER-NAME/getting-started

docker build --platform linux/amd64 -t YOUR-USER-NAME/getting-started .

docker run -dp 0.0.0.0:3000:3000 YOUR-USER-NAME/getting-started

Installing tutor:-

Choose one of the installation methods bellow. If you install tutor in differemt ways, you will end up with multiple tutor exicutable, which is going to be very confusing. At any time, you can check the path to your tutor exicutable by running which tutor.

Python package:

pip install "tutor[full]"

sudo apt install python3 python3-pip libyaml-dev

Binary release:

<https://github.com/overhangio/tutor/releases>

sudo curl -L "https://github.com/overhangio/tutor/releases/download/v16.0.1/tutor-$(uname -s)\_$(uname -m)" -o /usr/local/bin/tutor

sudo chmod 0755 /usr/local/bin/tutor

Installing from source :-

To inspect the tutor source code, install turoto from the git repository:

git clone https://github.com/overhangio/tutor

cd tutor

pip install -e .

Upgrading :-

To upgrade Open edX or benefit from the latest features and bug fixes, you should simply upgrade Tutor. Start by upgrading the “tutor” package and its dependencies:

pip install --upgrade "tutor[full]"

Then run the launch command again. Depending on your deployment target, run one of:

tutor local launch *# for local installations*

*{You get some error install some requirments.*

*Change the directory from open/edx-platform/edx-platform/requirments/edx/........*

*Then install requirments...........}:------extra adding*

tutor dev launch *# for local development installations*

*{again error: getting credentials –err: docker credential-desktop.exe resolves to executable in currunt directory} extra adding \_\_\_\_\_\_\_\_\_\_then if you install docker-desktop in this directory...........*

*Above steps we alredy done on top of the doc if you do same thigs in this time also.......*