**Install Docker Ubuntu | How to install Docker in Ubuntu 18.0.4 | Setup Docker on Ubuntu**

Please find steps needed for installing Docker in Ubuntu 18.0.4 instance. You can install in Docker in many ways. But try only one option.  
  
**Docker installation steps using default repository from Ubuntu**  
Update local packages by executing below command:  
sudo apt-get update  
  
**Install the below packages**  
sudo apt install gnupg2 pass -y

*gnupg2 is tool for secure communication and data storage. It can be used to encrypt data and to create digital signatures*

**Install docker**  
sudo apt install docker.io -y  
  
**Add Ubuntu user to Docker group**  
sudo usermod -aG docker $USER  
  
**We need to reload shell in order to have new group settings applied. Now you need to logout and log back in command line or execute the below command:**  
newgrp docker  
The Docker service needs to be setup to run at startup. To do so, type in each command followed by enter:  
  
sudo systemctl start docker

sudo systemctl enable docker  
sudo systemctl status docker

Graphical user interface, text

Description automatically generated

The above screenshot should confirm that Docker daemon is up and running.

**How to upload Docker images to Azure Container Registry | Hosting Docker images in Azure Container Registry**

Azure Container Registry is a managed, private docker registry service. You can create and maintain Azure container registries to store and manage your private Docker container images.  
  
It is alternative to Docker Hub. Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments.  
  
Let us see how to upload a docker images from your VM into ACR.

Diagram

Description automatically generated

**Pre-requistes:**

Make sure you have docker installed on your VM.

**Step 1 - Create Azure Container Registry (ACR)**  
  
Go to <https://portal.azure.com/>  
Create a Resource, Give container registry as a name

Graphical user interface, text, application, email

Description automatically generated

Click on Create

[Graphical user interface, application

Description automatically generated](https://1.bp.blogspot.com/-GDTc8telFB0/Xd9LmKlnx_I/AAAAAAAABTQ/f2cra2ccn7cpwt_G4mwDPep1GVFw59wmwCLcBGAsYHQ/s1600/Screen%2BShot%2B2019-11-27%2Bat%2B10.22.17%2BPM.png)

Enter values as mentioned below:

Give registry name, resource group and choose Enable admin user and SKU as Basic

Graphical user interface, text, application

Description automatically generated

 Click on Review and then create, now your container registry is been created.

**Step 2 - Download sample Python App**

Go to your machine where you have docker images stored. perform below command to download sample pythonapp which is already dockerized.

git clone https://github.com/femilv/Dockerapp-sample.git

cd  Dockerapp-sample/pythonApp

**Step 3 - Create docker image**

docker build . -t mypythonapp

type below command:

sudo docker images

this should show docker images you have created locally.

Now go to ACR, click on docker registry you created. Click on access keys under Settings.

[Graphical user interface, text, application, chat or text message

Description automatically generated](https://1.bp.blogspot.com/-6SGS_PBEMts/Xd9PTr3_R8I/AAAAAAAABTo/PDziOEF_SqIf_icf6dP7r-V7LPCzYEHeACLcBGAsYHQ/s1600/Screen%2BShot%2B2019-11-27%2Bat%2B10.37.58%2BPM.png)

Copy value under login server

Login to Azure Container Registry through command line.

sudo docker login mydockerazureregistry.azurecr.io

Username: mydockerazureregistry

Text

Description automatically generated

Enter password by copying value below password

**Step 4 - Tag and Push your docker image to ACR**

Now tag the docker image per as below:

sudo docker tag mypythonapp mydockerazureregistry.azurecr.io/mypythonapp

sudo docker push mydockerazureregistry.azurecr.io/mypythonapp

That's it..Docker image is pushed into ACR.

You can see the Docker image uploaded under ***Services--> repositories***

Graphical user interface, application

Description automatically generated

If you want to run your Docker container in the background, like a service, you can use the -d option (detached mode). This starts the container in the background and leaves it running until it's stopped explicitly.

Here's how you can do it:

sudo docker run -d -p 8081:5000 femipy

then browse the app http://20.51.191.227:8081/