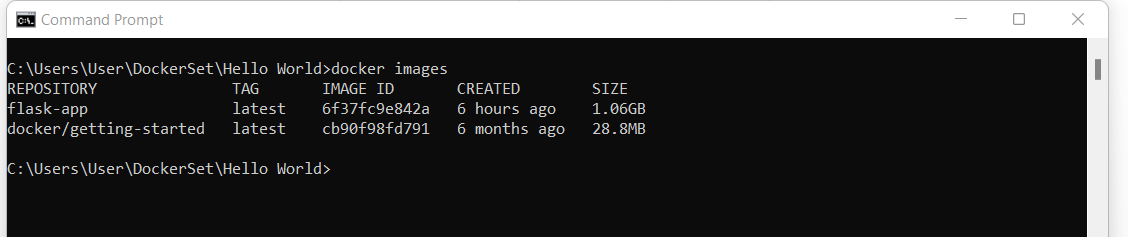
# **DOCKER COMMANDS – USAGE WITH SCREENSHOTS**

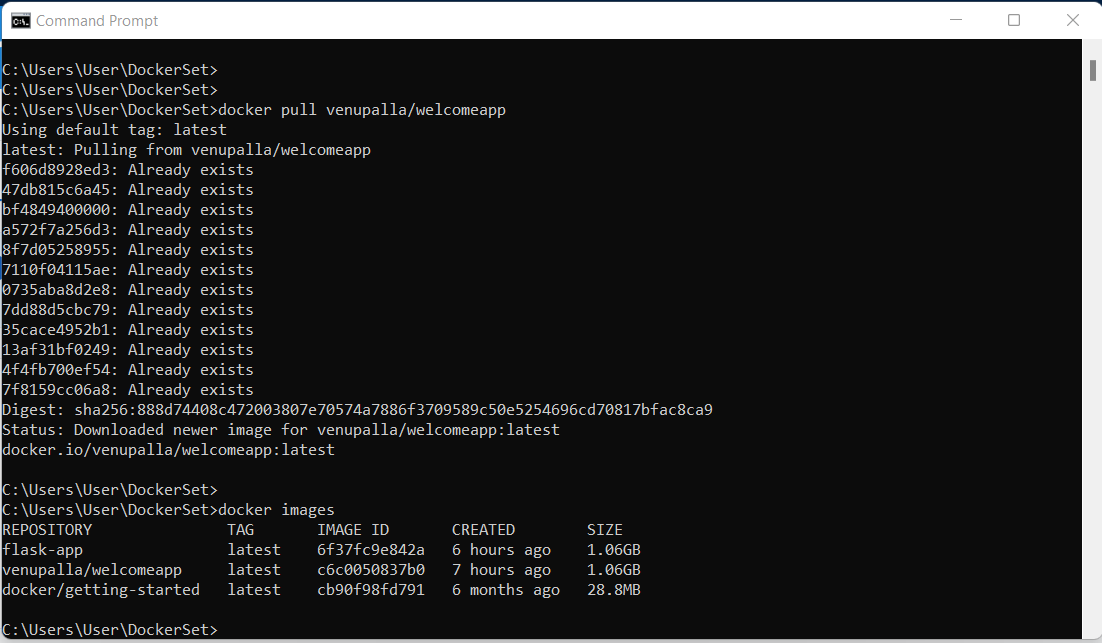
1. **$ docker images**

Lists all the docker images available in your local system.



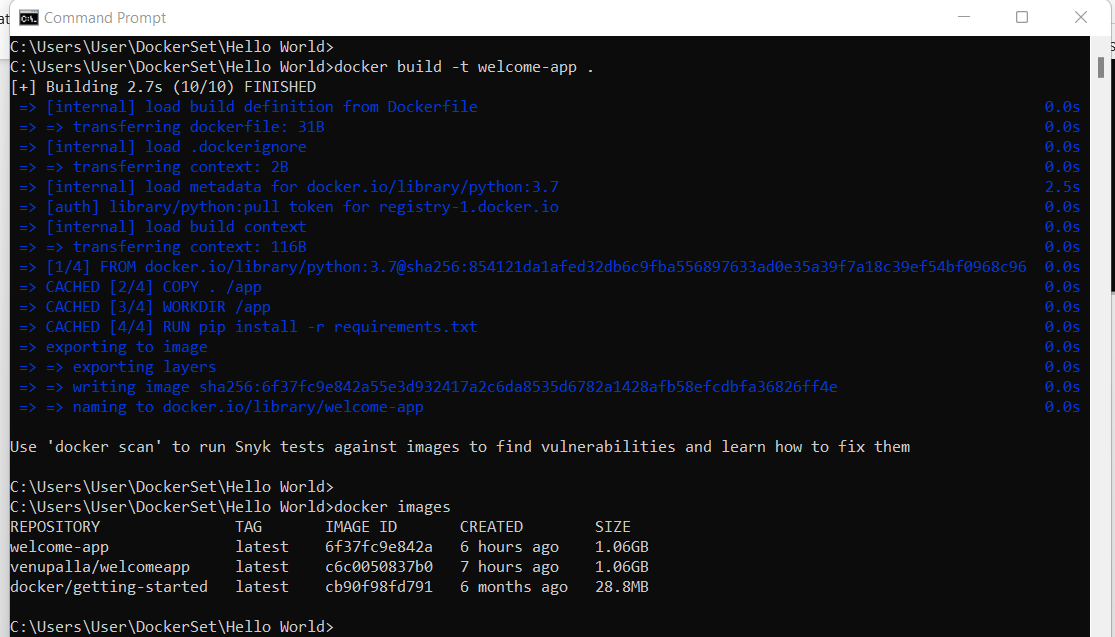
1. **$ docker pull <image-url>**

This command will pull the image from given Docker hub url into local machine.

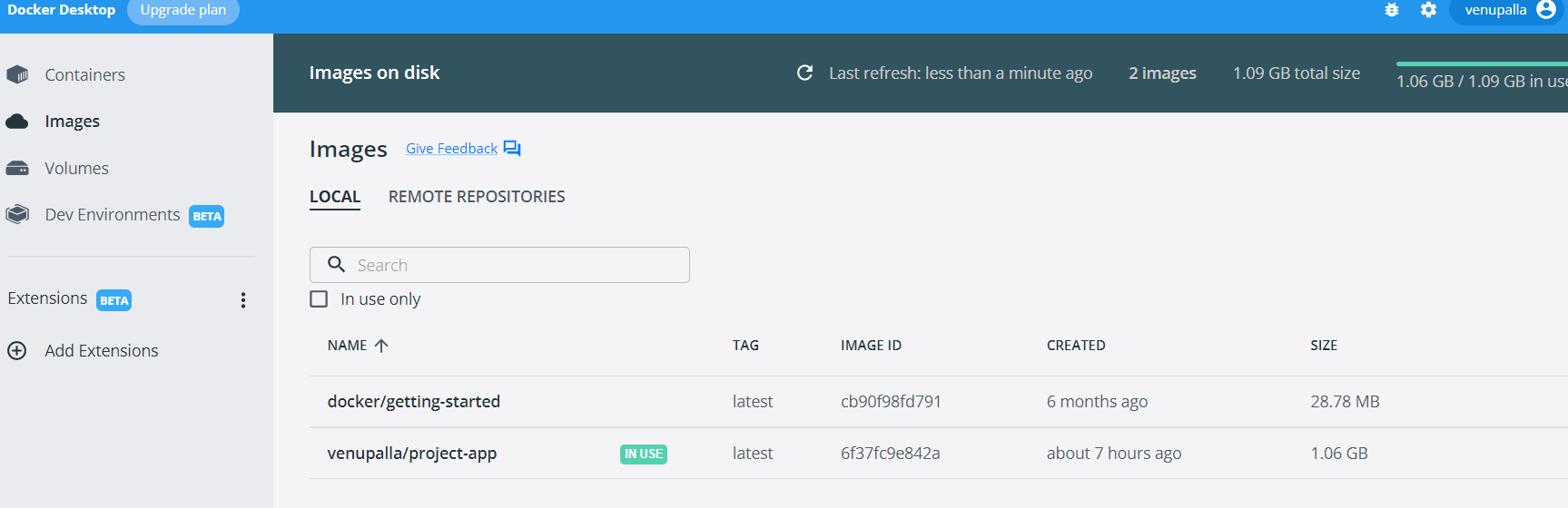


1. **$ docker build -t hello-app .**

This command builds Docker images from a Dockerfile and a “context” from local folder.

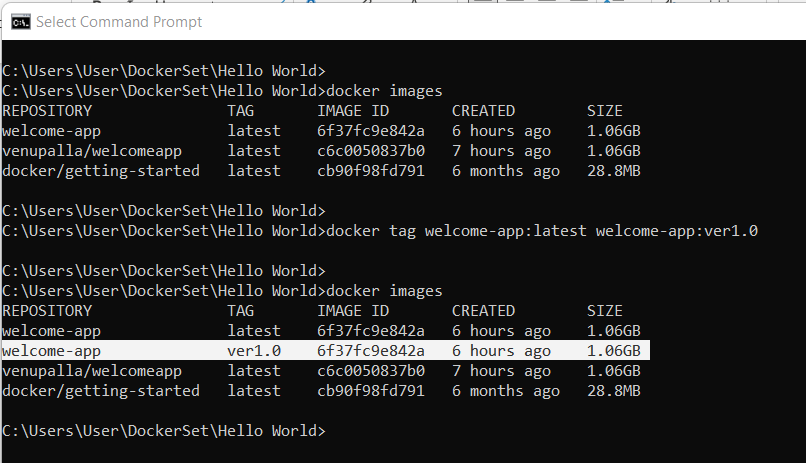


The images can be seen in Docker desktop tool.



1. **$docker tag <image-name>:<TAG> <iamge-name>:<NEW-TAG>**

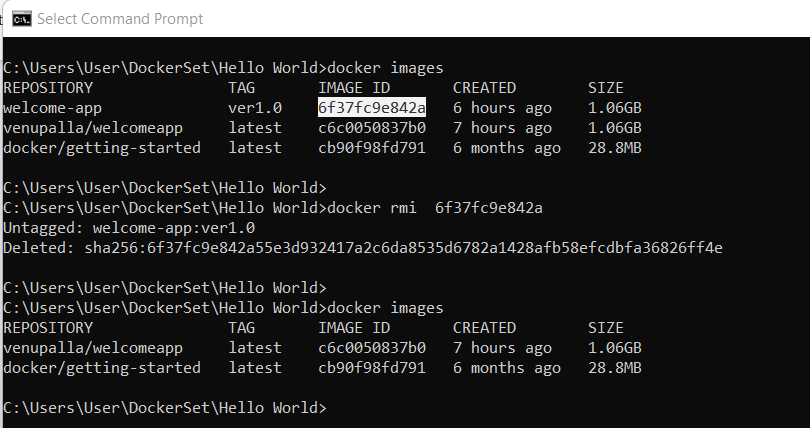
This command creates a new tag for an image. Actually  It does not create a new image. The tag points to the same image and is just another way to reference the image.



1. **$ docker rmi <image-id>**

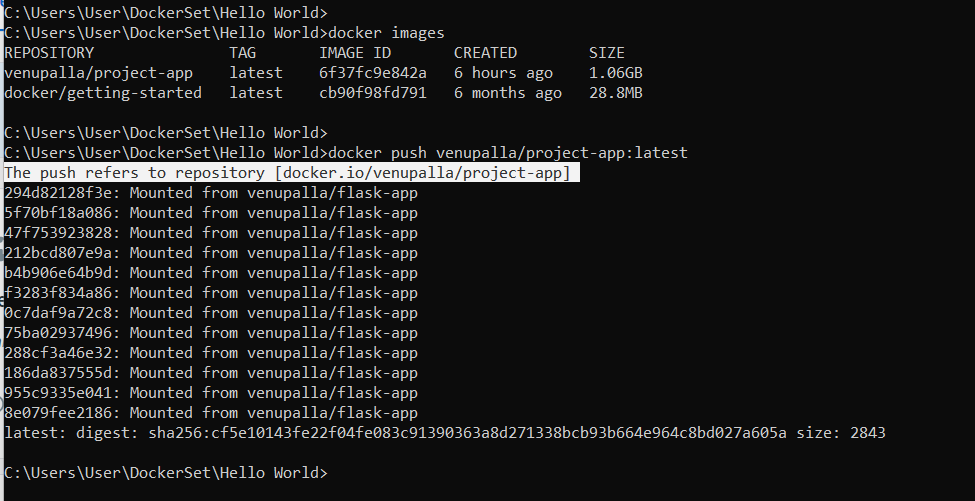
This command removes the image from local machine.

**Note:** But we need to remove all dependent containers before removing an image.

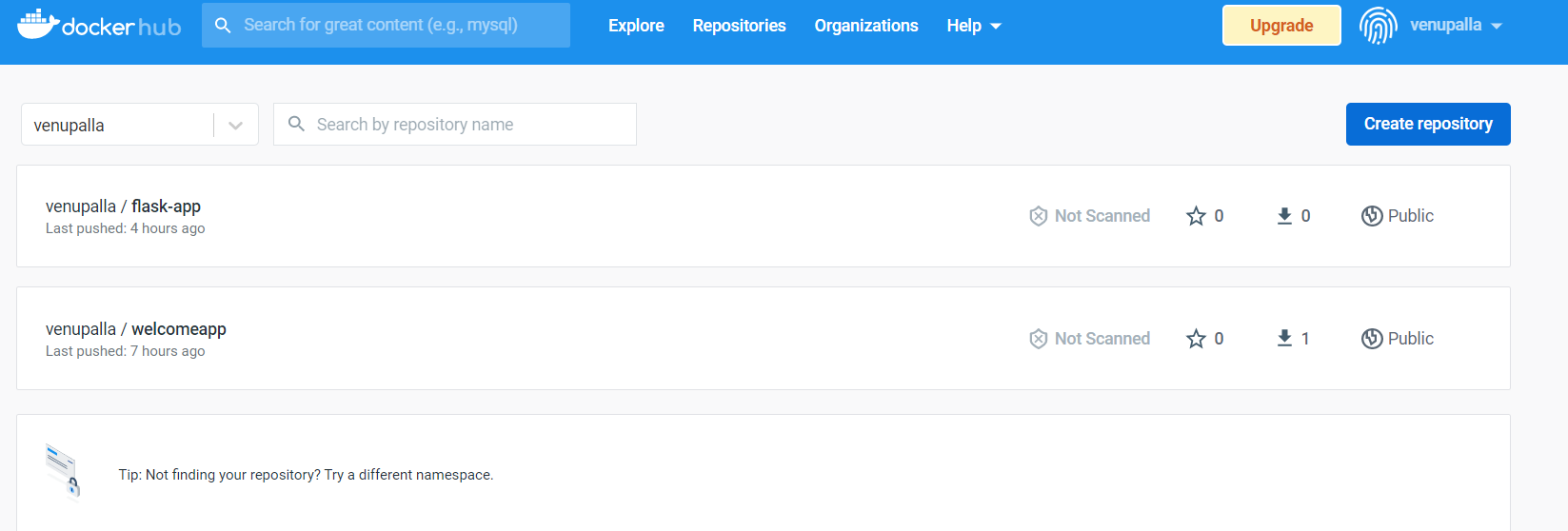


1. **$ docker push <userid>/<image-name>:<tagname>**

This command pushes the image from local machine to Docker hub repository.



This repository can be seen under user account in docker hub website.



1. **$ docker run -d --name <container name> -p <localhost ip>:<cotainerhost ip> <image-id>**

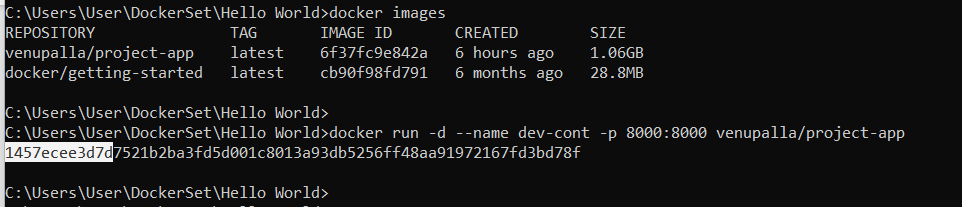
This command runs the image and creates the container out of it.

Options:

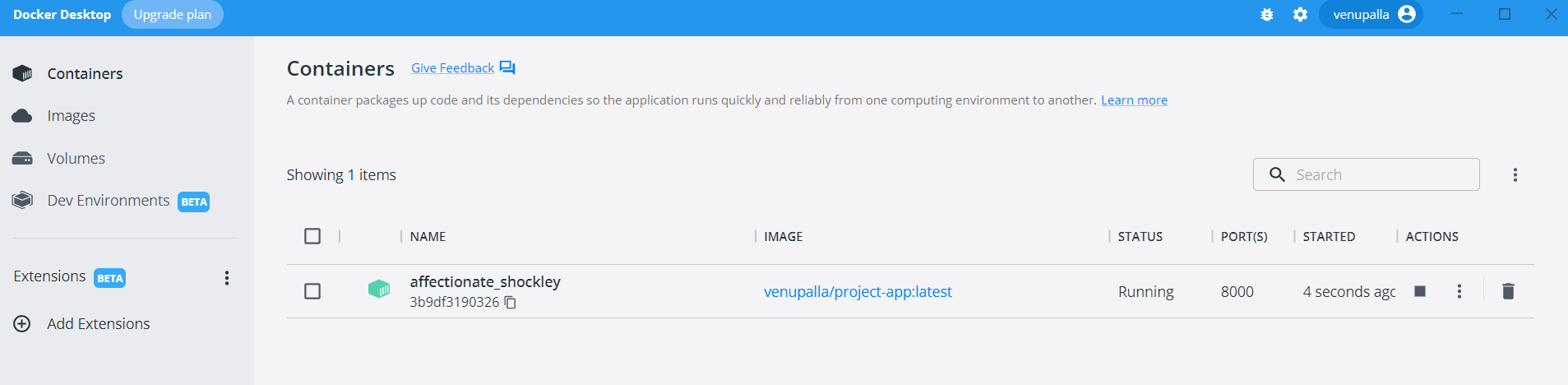
-d : detach mode

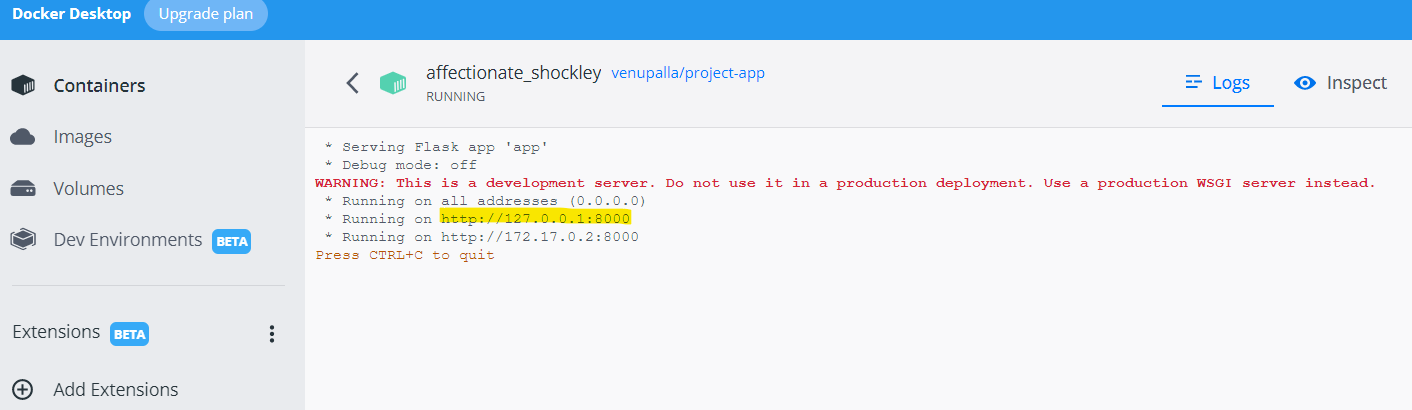
--name: container name

-p : port values

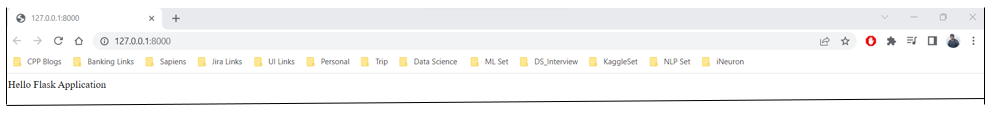


The container can be seen in Docker desktop tool.



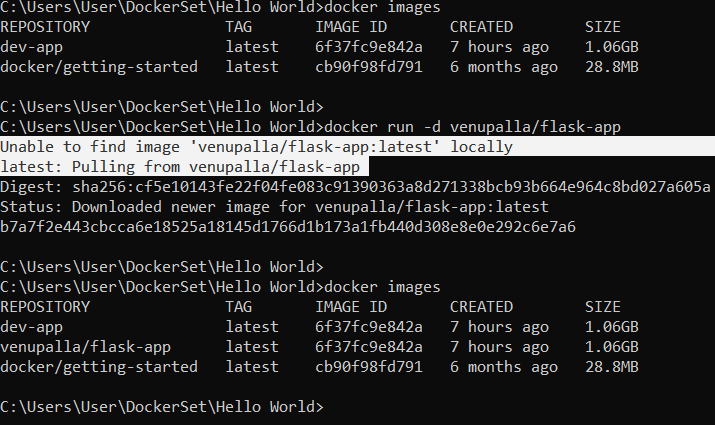


By clicking on local host url, it gives the application output message.



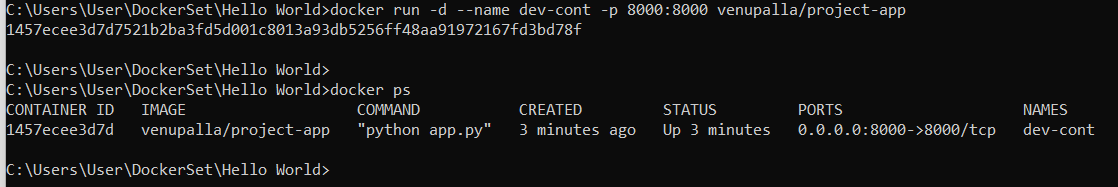
1. **$ docker run <dockerhub image-url>**

This command directly creates the image and container from docker hub repository url.



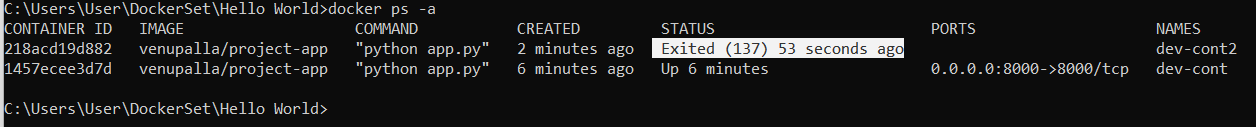
1. **$ docker ps**

This command lists only active containers in local machine.



1. **$ docker ps -a**

This command lists all active and EXITED containers in local machine.



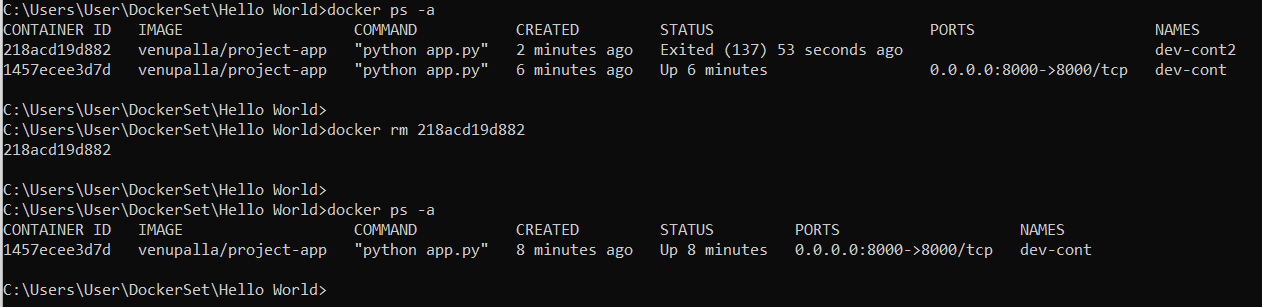
1. **$ docker stop <container id>**

This command stops the container.

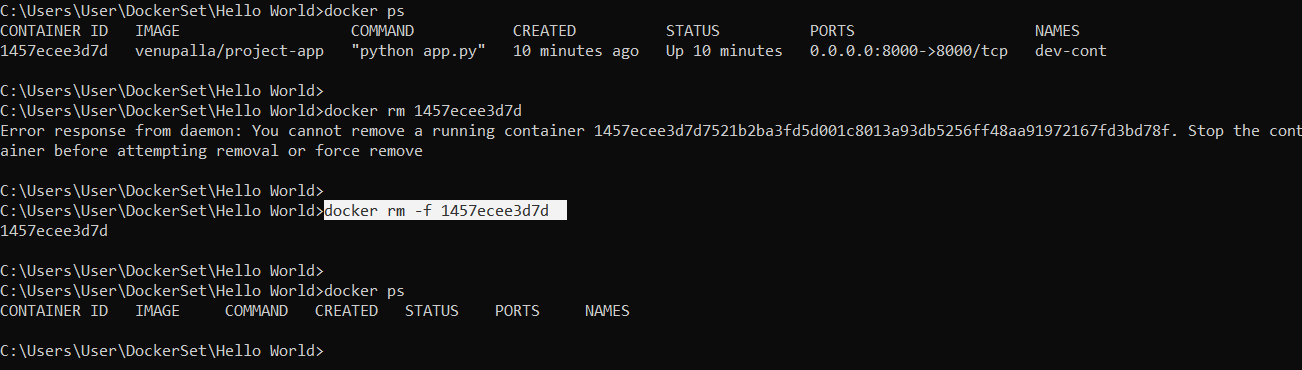


1. **$ docker rm <container id>**

This command removes the stopped/inactive container from local machine.

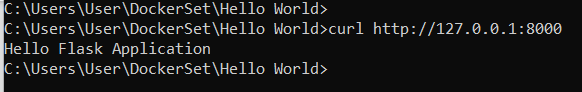


1. **$ docker rm -f < container id> : removes the active container forcefully without stopping it.**



1. **$ curl** [**http://127.0.0.1:8000**](http://127.0.0.1:8000)

This command connects to the application running inside of our container on port 8000 and shows the result.



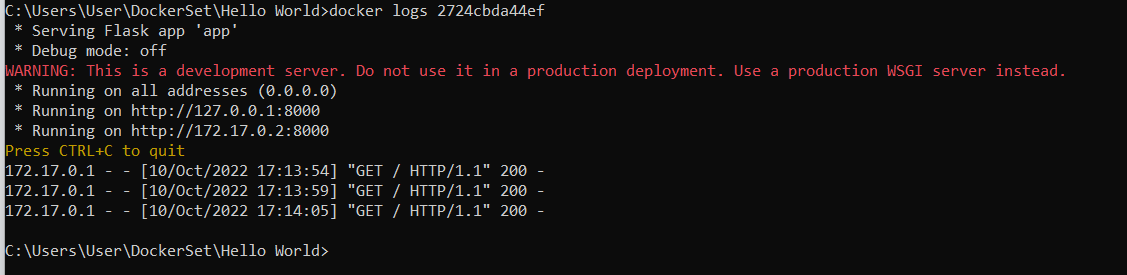
1. **$ docker restart <container-id>**

This command restart the container which we have stopped earlier.



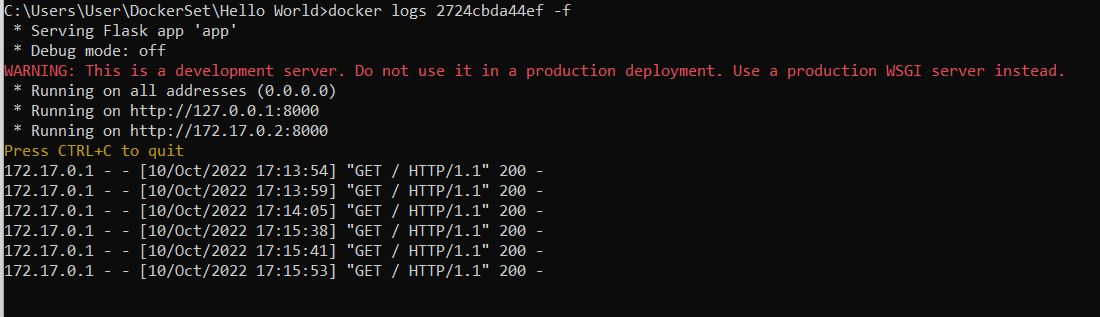
1. **$ docker logs <container id>**

This command lists the log of given container.



**$ docker logs <container id> -f**

This command lists the log of given container in real time (live) without exiting the command.



Type ‘Ctrl+C’ to quit.