Cheat Sheet to Docker- Important Doc Commands for Software Developers

May 19, 2020

Considering, the widespread use of Docker, we have crea includes widely-used Docker commands for making you productive & efficient.

Undoubtedly, Docker has become hotter than hot! Wher in 2013, it didn't take long for Docker to garner all the atte the world.

All the noise was happening because of all the benefits D nutshell, Docker makes it easy for anyone to create conta in no time; it allows you to run more apps than anything servers; and it also helps in managing or deploying any a hassle. In this article, we have covered some of the most widely used Docker commands.

Though there are several Docker commands you must k covered some of the most essential commands for Dock 11 Docker commands that will help you in saving your tir commands will greatly help you to have a clear picture o ut any difficulties.

first understand the fundamentals of Docker…

Various cases in which you can use Docker

You can use Docker for-

- Distributing the OS of your app with a team as a version control syste
- In development phases like “Development”, “Testing” & “QA”.
- While running your code locally to replicate the server’s environment

Locally setting up Docker?

- First, you have to download Docker and its toolbox.
- After downloading, now you have to check that the AMD-V, Virtualiza enabled in your BIOS.
- Lastly, simply running the setup after installing the extension pack in do the job.

How to use a Docker?

Just like the snapshots of virtual machines, Docker has ir nothing but a file having several layers. It is used to execu and can be used as it is or customized. To create a Docke use the default execution driver i.e. libcontainer. Moreove Docker Hub for searching for all the Docker images.

- “*$ docker search <image>*” – You can use it for searching any image i
- *ocker images*” – You can use it for checking the total images on y
- “ *ocker run hello world*” – You can use it for downloading a “hello v ker container.

Read More: *Learn How To Use Docker Images?*

In case, if you don’t know what Docker is or looking to lea can begin with “*Docker for Dummies – The Complete A Guide*”.

11 Docker commands that you can use right

Since its inception, Docker has grown tremendously offe as per various virtualization needs. Because of this, you ca commands for Docker CLI.

1. To create a container with images from Docker Hub

```
$ docker create -it ubuntu:xenial bash
```

Docker Hub is basically an official repository for Docker ir
many pre-built images that were created by enterprises a
can use the above command that will help you to pull th
Xenial from Docker Hub. It will create a container using t

2. Listing containers in Docker

After creating the container, now you have to check whe
or not. The below command will inform the Docker to list
ainers.

Shares

```
cker ps
```

There will be a high chance that the "*ps*" command won'
containers. This might be due to the fact that you have n
Docker container. Don't worry – you can use the "*-a*" flag
Docker to list all the containers.

```
$ docker ps -a
```

It will help you to see the container that you have createc
command.

3. Initiating your container

You have to start your container so that you can use it in
command will help you start your container in seconds.

```
$ docker start CONTAINER_ID
```

In this command, “CONTAINER ID” will be the id of your
get this ID while running the “*ps -a*” command.

Read More: *Learn How To Create And Start Docker Con*

4. Connecting your container

Now you have started your container but you still need to
container to use it. To do so, you have to use the Docker “

```
cker attach CONTAINER_ID
```

lead to a change in your prompt and now you are i
ıl-based container. Here, whatever you do will remai
container.

5. Terminating your container

Just like starting a container was simple, terminating a c
simple. Without any issues, you can simply terminate an
Docker “*stop*” command. When it comes to the newbies,
useful commands because it allows you to exit easily fror
container.

```
$ docker stop CONTAINER_ID
```

However, the catch here is that you have to run this comı
different terminal as the original terminal is running the
not contain Docker. Still, if it is difficult to figure out then

container by pressing “Ctrl+D” or just typing “exit” inside

6. Deleting your container

One of the major reasons for a significant increase in the container is its ability that helps you in deleting any cont affecting the host machine. You can use the below comr

```
$ docker rm CONTAINER_ID
```

ommand will instantly delete your container, but w configurations and files will be deleted too. So, if you e, it is better to move them before you delete the co

7. Killing your container

After reading the title, I am sure the 1st thing that came i “How stopping or terminating a container is different fro container”, right? Well, in Docker, there is a very subtle di the two. When you terminate the container, it will 1st sto the container and then it will stop the container.

Whereas, with the “*kill*” command, you kill the entire pro running in the host system. It might lead to corruption ir processes or can cause unwanted misconfigurations. So, below command, be very cautious about killing a contair

```
$ docker kill CONTAINER_ID
```

Read More: *Learn How To Stop, Kill And Clean Up Dock*

8. Containers detachment

This command will be useful for you in a certain situation already attached to any container but because of some r leave the container running and want to use terminal ses docker commands for stopping or killing a container will container, you won't be able to use them.

```
Ctrl+P Ctrl+Q
```

Instead, you can just hold the combination of the above k inside a running container. It will help you in detaching f For instance, you can just hold down the Ctrl button and P & Q one after another.

9. How to copy contents from containers to the file sys

The need may arise to copy the configuration files from t system or other containers. You can use the "*cp*" comma action and it copies content recursively.

```
$ docker cp CONTAINER_ID:/test_file test_file
```

The above command helps in copying the test_file from the CONTAINER_ID to the host system. And you can use command for copying the file from the host to a contain

```
$ docker cp test_file CONTAINER_ID:/test_file
```

10. Listing all the images of Docker

As stated earlier, Docker images are very essential for dev the building block of any container. Do not forget that th containers is created using a Docker image. Because of t keep multiple images of Docker across their systems. Wit command, you can check all the images in your system i

```
$ docker images
```

Executing this Docker command will help you check all t images with its repository, tags, and size. If you want to ju IMAGE_ID then you can simply use *-q* option.

11. Removing Docker images

A time will arise when you might not require any particul cases, you can remove one or more Docker images with t for Docker.

```
$ docker rmi <IMAGE_ID>
```

In case, if any particular image is tagged with multiple re have to delete it with the following Docker command.

```
$ docker rmi REPOSITORY:TAG
```

And again, you can find the required information when y command Docker images.

Wrapping Up!

No doubt Docker has become an integral part of moderr development. It is packed with a myriad of Docker comm check in its documentation. However, in this article, we h some of the most basics and important Docker comman for your upcoming project without any difficulties.

Shares

e that all these Docker commands will be useful for soon we will be releasing some of the more advanced co so stay tuned and keep reading! And meanwhile, if you w about Docker then we have also listed some of the *top b* you can read right away.