**Dockerfile Key instructions**

There are number of Instructions used in Dockerfile but here most frequent used are covered.

**From:** It is the first instruction which is used to build image in Docker and basically used to set the base image on which subsequent instructions will be applied on this base image.

**COPY vs ADD:** These both instructions are designed to add directories and files into your Docker image. **ADD <src> . . . <dest>** or **COPY <src> <dest>.**

* ADD has extra features like it can pull files from **URL** sources which COPY cannot.
* ADD can extract compressed file
* COPY is used simpler solution for copying files in easier way

**WORKDIR:** A convenient way to define the working directory. If the directory doesn’t exist, Docker will create for you. It is used along subsequent RUN, CMD, ENTRYPOINT, COPY and ADD instructions. You can specify **WORKDIR** multiple times.

**RUN:** It is used to execute commands in your image. You have to use a lot of apt-get and other similar commands to add new packages to your image.

* RUN commands has two forms
  + RUN <command>

**e.g**: **RUN apt-get update**

* + RUN [“executable”, “param1”, “param2”]

**e.g: RUN ["/bin/bash", "-c", "echo hello"]**

**ENV:** It is used to define environment variables. Using environment variables, you can set/specify value in the Dockerfile.

**VOLUMES**:This Instruction used to tell Docker that the stuff you want to store in specific directory should be stored on **host file system** (Machine on which containers are running) not in container. The main benefit of using this is to **persist** your data if unluckily container crashes.

You can also share volumes from one container to another using command

**docker run --volumes-from**

Commands used in Volumes

* **docker volume ls**
* **dokcer volume inspect**

**EXPOSE:** Instruction used to tell users the port on which your application listening on. It is not used to publish the port.

**CMD and ENTRYPOINT:**

**CMD** instruction used to specify what component is to be run by your image. e.g **CMD [“executable”, “param1”, “param2”].**

* You can specify only single CMD instruction in Dockerfile
* It can be override
* CMD [“echo”, “Hello world”] when your container will start this instruction will be executed.

**ENTRYPOINT** instruction specify an entry point, you use **ENTRYPOINT** as the main executable of your image. Whatever you specify in ***CMD*** will be added to ***ENTRYPOINT*** as parameter. It can’t be override.

ENTRYPOINT [“echo”, “Hello”]

CMD [“world”]

Output 🡺 Hello world. As world which is defined in CMD become parameter of ENTRYPOINT.

**ONBUILD:** You specify the instructions with ONBUILD that will be executed when your image will be used as the base image. e.g

**ONBUILD** **COPY** . /usr/src/app

**ONBUILD** **RUN** /usr/src/app/mybuild.sh