

Dockerfile Instruction	Explanation
FROM	To specify the base image which can be pulled from a container registry(Docker hub, GCR, Quay, ECR, etc)
RUN	Executes commands during the image build process.
ENV	Sets environment variables inside the image. It will be available during build time as well as in a running container. If you want to set only build-time variables, use ARG instruction.
COPY	Copies local files and directories to the image
EXPOSE	Specifies the port to be exposed for the Docker container.
ADD	It is a more feature-rich version of the COPY instruction. It also allows copying from the URL that is the source and tar file auto-extraction into the image. However, usage of COPY command is recommended over ADD. If you want to download remote files, use curl or get using RUN.
WORKDIR	Sets the current working directory. You can reuse this instruction in a Dockerfile to set a different working directory. If you set WORKDIR, instructions like RUN, CMD, ADD, COPY, or ENTRYPOINT gets executed in that directory.
VOLUME	It is used to create or mount the volume to the Docker container
USER	Sets the user name and UID when running the container. You can use this instruction to set a non-root user of the container.
LABEL	It is used to specify metadata information of Docker image
ARG	Is used to set build-time variables with key and value. the ARG variables will not be available when the container is running. If you want to persist a variable on a running container, use ENV.
CMD	It is used to execute a command in a running container. There can be only one CMD, if multiple CMD there then it only applies to the last one. It can be overridden from the Docker CLI.
ENTRYPOINT	Specifies the commands that will execute when the Docker container starts. If you don't specify any ENTRYPOINT, it defaults to <code>/bin/sh -c</code> . You can also override ENTRYPOINT using the <code>--entrypoint</code> flag using CLI. Please refer CMD vs ENTRYPOINT for more information.