**1.Docker Instructions and Commands Documentation**

* Docker Instructions and Commands Documentation
* FROM ubuntu:latest

**2. RUN**

* Executes a command inside the container while building the image.
* Example:
* RUN apt-get update && apt-get install -y nginx

**3. CMD**

* Provides default executable and arguments for the container.
* Example:
* CMD ["nginx", "-g", "daemon off;"]

**4. ENTRYPOINT**

* Defines a command that will always run within the container.
* Example:
* ENTRYPOINT ["python", "app.py"]

**5. COPY**

* Copies files and directories from the host to the container.
* Example:
* COPY . /app

**6. ADD**

* Similar to COPY but allows remote URLs and archives extraction.
* Example:
* ADD sample.tar.gz /app/

**7. WORKDIR**

* Sets the working directory inside the container.
* Example:
* WORKDIR /app

**8. ENV**

* Sets an environment variable.
* Example:
* ENV APP\_ENV=production

**9. EXPOSE**

* Informs Docker that the container listens on the specified port.
* Example:
* EXPOSE 80

**10. VOLUME**

* Creates a mount point for persistent storage.
* Example:
* VOLUME /data

**11. LABEL**

* Adds metadata to an image.
* Example:
* LABEL maintainer="yourname@example.com"

**12. USER**

* Sets the user to run the container.
* Example:
* USER nonrootuser

**13. HEALTHCHECK**

* Defines a command to check container health.
* Example:
* HEALTHCHECK --interval=30s --timeout=10s CMD curl -f http://localhost || exit 1

**Common Docker Commands**

Docker commands help manage images, containers, and networks.

**1. Docker Version and Info**

* Check the installed Docker version:
* docker --version
* Get system-wide Docker information:
* docker info

**2. Image Management**

* List all images:
* docker images
* Pull an image from Docker Hub:
* docker pull ubuntu
* Remove an image:
* docker rmi image\_name

**3. Container Management**

* Run a container interactively:
* docker run -it ubuntu bash
* Start/Stop a container:
* docker start container\_id
* docker stop container\_id
* Remove a container:
* docker rm container\_id
* List running containers:
* docker ps
* List all containers:
* docker ps -a

**4. Docker Networks**

* List networks:
* docker network ls
* Create a network:
* docker network create my\_network
* Connect a container to a network:
* docker network connect my\_network container\_id

**5. Docker Volumes**

* Create a volume:
* docker volume create my\_volume
* List volumes:
* docker volume ls
* Remove a volume:
* docker volume rm my\_volume

**6. Building and Running Custom Images**

* Build an image from a Dockerfile:
* docker build -t my\_image .
* Run a container from the built image:
* docker run -d -p 8080:80 my\_image

**Conclusion**

Docker provides powerful features to build, manage, and deploy applications efficiently. Understanding its instructions and commands allows for better container orchestration and infrastructure automation.