Dockerfile Keywords - Detailed Explanation and Scenarios

Certainly! Dockerfiles use various keywords (or instructions) to define how a Docker image is built and how the resulting container behaves. Each keyword serves a specific purpose and is used in different scenarios. Below is a detailed explanation of the most commonly used Dockerfile keywords, their differences, and the scenarios in which they are used:

1. FROM

Purpose: Specifies the base image for the Docker image.

Syntax: FROM <image>:<tag>

Example: FROM ubuntu:20.04

Scenario: Used at the beginning of every Dockerfile to define the starting point for the image.

2. RUN

Purpose: Executes commands during the image build process.

Syntax: RUN <command>

Example: RUN apt-get update && apt-get install -y python3

Scenario: Used to install dependencies or perform setup tasks.

3. COPY

Purpose: Copies files or directories from the host machine into the Docker image.

Syntax: COPY <src> <dest>

Example: COPY . /app

Scenario: Used to add application code or resources to the image.

4. ADD

Purpose: Similar to COPY but also supports downloading files and extracting tar archives.

Syntax: ADD <src> <dest>

Example: ADD https://example.com/file.tar.gz /app

Scenario: When you need to download or auto-extract files.

5. ENV

Purpose: Sets environment variables.

Syntax: ENV <key>=<value>

Example: ENV APP_HOME=/app

Scenario: Pass runtime config values or simplify builds.

6. WORKDIR

Purpose: Sets the working directory.

Syntax: WORKDIR <path>

Example: WORKDIR /app

Scenario: Used to define where commands will be run.

7. EXPOSE

Purpose: Documents the port the app listens on.

Syntax: EXPOSE <port>

Example: EXPOSE 8080

Scenario: Inform about ports; does not publish them.

8. CMD

Purpose: Specifies default command at container start.

Syntax: CMD ["executable", "param1"]

Example: CMD ["python3", "app.py"]

Scenario: Define the container's default process.

9. ENTRYPOINT

Purpose: Makes container run like an executable.

Syntax: ENTRYPOINT ["executable"]

Example: ENTRYPOINT ["python3"]

Scenario: Used for always-running commands, accepts args.

10. ARG

Purpose: Build-time variable.

Syntax: ARG <name>=<value>

Example: ARG PYTHON_VERSION=3.8

Scenario: Pass values like versions during build.

11. VOLUME

Purpose: Declare mount point for persistent data.

Syntax: VOLUME <path>

Example: VOLUME /data

Scenario: Persist or share data outside containers.

12. LABEL

Purpose: Adds metadata.

Syntax: LABEL <key>=<value>

Example: LABEL maintainer="you@example.com"

Scenario: Image info, search, and management.

13. USER

Purpose: Sets user to run container.

Syntax: USER <user>

Example: USER appuser

Scenario: Avoid root; use secure users.

14. HEALTHCHECK

Purpose: Monitors container health.

Syntax: HEALTHCHECK CMD <command>

Example: HEALTHCHECK CMD curl -f http://localhost:8080/ || exit 1

Scenario: Auto-checks app health, restarts if needed.

15. ONBUILD

Purpose: Trigger for downstream builds.

Syntax: ONBUILD <instruction>

Example: ONBUILD COPY . /app

Dockerfile Keywords - Detailed Explanation and Scenarios

Scenario: Reusable base images that auto-setup next builds.

16. SHELL

Purpose: Changes default shell.

Syntax: SHELL ["/bin/bash", "-c"]

Example: SHELL ["/bin/bash", "-c"]

Scenario: If you need bash features or different shell behavior.

Summary Table:

FROM - Define base image

RUN - Install/setup during build

COPY - Add files to image

ADD - Like COPY + downloads/extracts

ENV - Set environment variables

WORKDIR - Set directory for next commands

EXPOSE - Document listening port

CMD - Default command

ENTRYPOINT - Main executable with arg support

ARG - Build-time variables

VOLUME - Persistent storage

LABEL - Metadata

USER - Run as non-root user

HEALTHCHECK - Monitor container health

ONBUILD - Triggers for child builds

SHELL - Change shell used in RUN/CMD