

Day 3 – Multi-stage Docker Builds

1. What is Multi-stage Docker Build?

Ans) Multi-stage Docker Build is a technique in Docker that allows you to use multiple FROM statements in your Dockerfile, enabling you to create clean, optimized images by separating the build environment from the final runtime environment.

Why Use Multi-Stage Builds?

Without multi-stage builds:

-> You might include unnecessary build tools (e.g., compilers) in your final image.

-> This increases image size, reduces security, and makes deployment less efficient.

With multi-stage builds:

-> Only the essential artifacts (like compiled binaries or built files) are copied to the final image.

-> The result is a smaller, cleaner, and more secure image.

How it Works (Example)

Let's say you're building a Go application:

```
dockerfile

# Stage 1: Build
FROM golang:1.21 AS builder
WORKDIR /app
COPY . .
RUN go build -o myapp

# Stage 2: Final
FROM alpine:latest
WORKDIR /app
COPY --from=builder /app/myapp .
ENTRYPOINT ["/myapp"]
```

-> Stage 1 (golang): A full Go environment to compile the code.

-> Stage 2 (alpine): A minimal runtime image that contains only the binary.