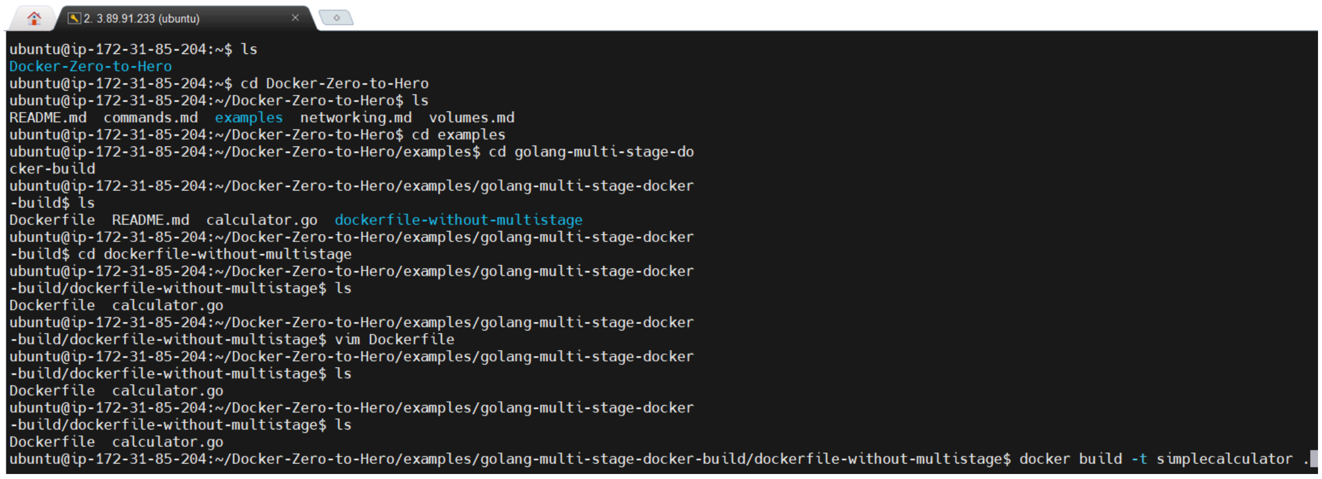
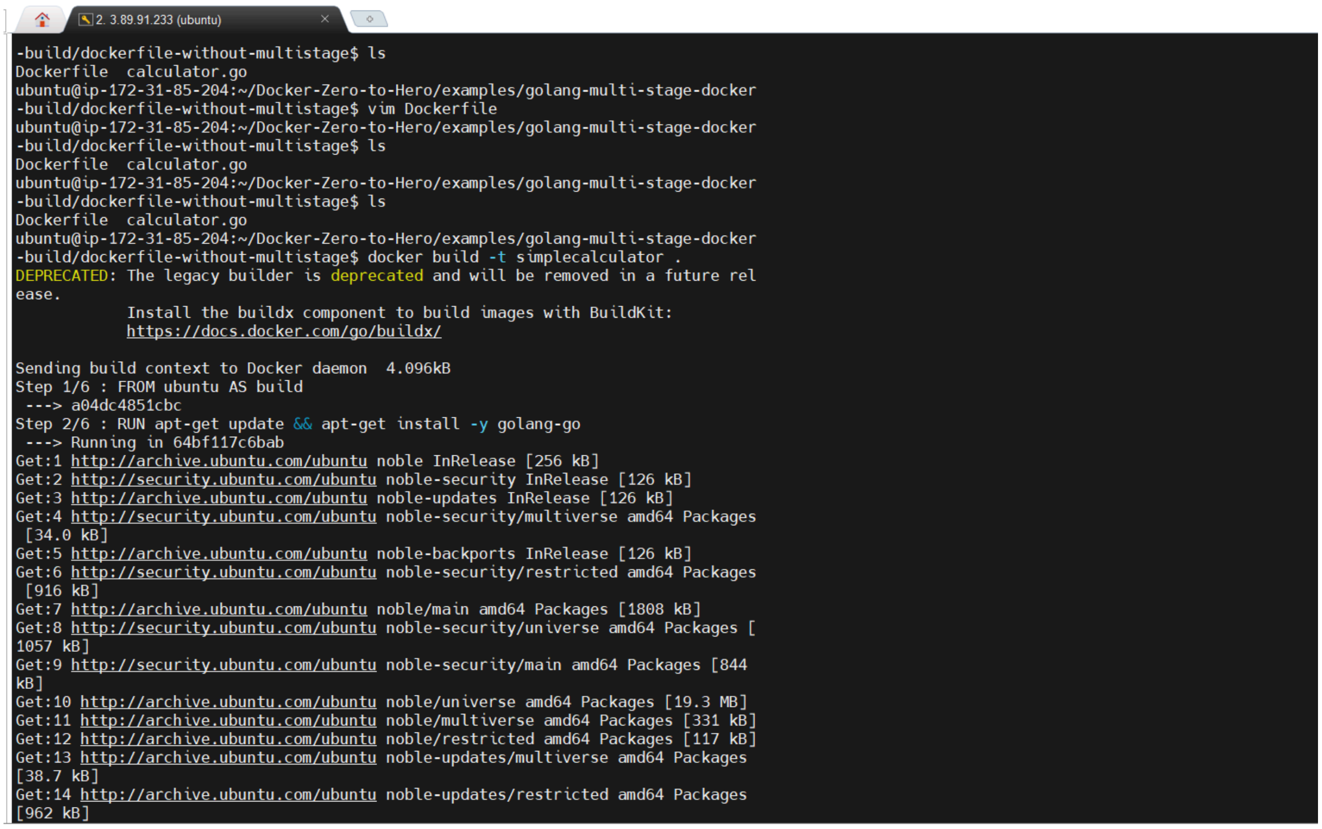
Docker - Multi Stage Build and Distroless Image



```
ubuntu@ip-172-31-85-204:~$ ls
Docker-Zero-to-Hero
ubuntu@ip-172-31-85-204:~$ cd Docker-Zero-to-Hero
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero$ ls
README.md  commands.md  examples  networking.md  volumes.md
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero$ cd examples
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples$ cd golang-multi-stage-do
cker-build
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build$ ls
Dockerfile  README.md  calculator.go  dockerfile-without-multistage
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build$ cd dockerfile-without-multistage
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ ls
Dockerfile  calculator.go
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ vim Dockerfile
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ ls
Dockerfile  calculator.go
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ ls
Dockerfile  calculator.go
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker-build/dockerfile-without-multistage$ docker build -t simplecalculator .
```

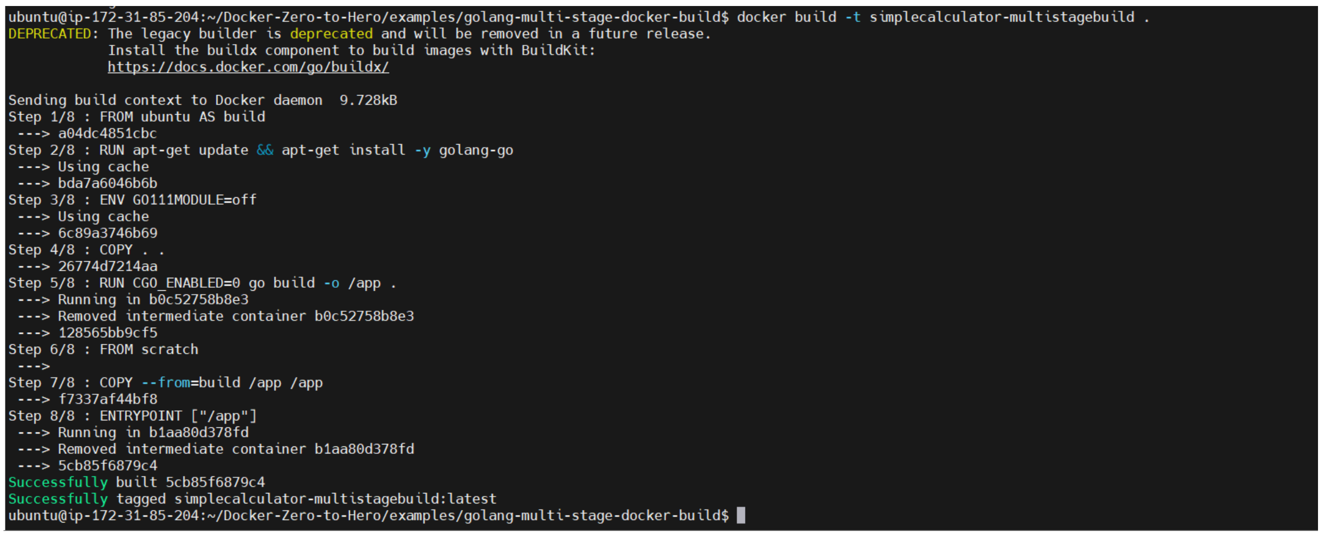
The command docker build -t simplecalculator .

- What It Does?
  Builds a Docker image named `simplecalculator` using the Dockerfile in the current directory.



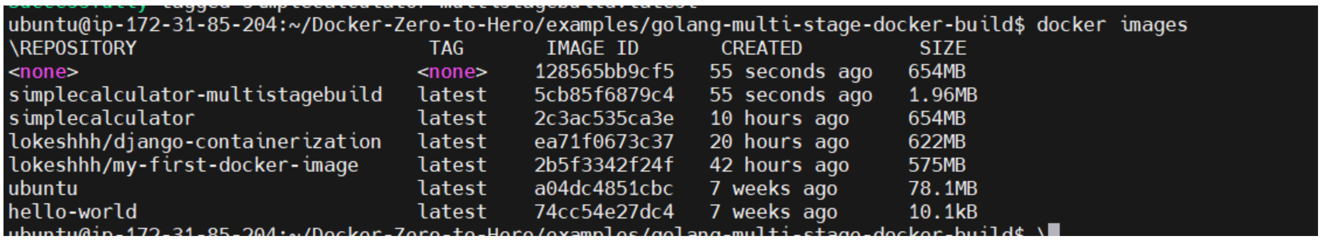
```
-build/dockerfile-without-multistage$ ls
Dockerfile  calculator.go
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ vim Dockerfile
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ ls
Dockerfile  calculator.go
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ ls
Dockerfile  calculator.go
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker
-build/dockerfile-without-multistage$ docker build -t simplecalculator .
DEPRECATED: The legacy builder is deprecated and will be removed in a future rel
ease.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.096kB
Step 1/6 : FROM ubuntu AS build
 ---> a04dc4851cbc
Step 2/6 : RUN apt-get update && apt-get install -y golang-go
 ---> Running in 64bf117c6bab
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages
 [34.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages
 [916 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [
1057 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [844
kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages
[38.7 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages
[962 kB]
```



```
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker-build$ docker build -t simplecalculator-multistagebuild .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  9.728kB
Step 1/8 : FROM ubuntu AS build
 ---> a04dc4851cbc
Step 2/8 : RUN apt-get update && apt-get install -y golang-go
 ---> Using cache
 ---> bda7a6046b6b
Step 3/8 : ENV GO111MODULE=off
 ---> Using cache
 ---> 6c89a3746b69
Step 4/8 : COPY . .
 ---> 26774d7214aa
Step 5/8 : RUN CGO_ENABLED=0 go build -o /app .
 ---> Running in b0c52758b8e3
 ---> Removed intermediate container b0c52758b8e3
 ---> 128565bb9cf5
Step 6/8 : FROM scratch
 --->
Step 7/8 : COPY --from=build /app /app
 ---> f7337af44bf8
Step 8/8 : ENTRYPOINT ["/app"]
 ---> Running in b1aa80d378fd
 ---> Removed intermediate container b1aa80d378fd
 ---> 5cb85f6879c4
Successfully built 5cb85f6879c4
Successfully tagged simplecalculator-multistagebuild:latest
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker-build$ 
```



```
ubuntu@ip-172-31-85-204:~/Docker-Zero-to-Hero/examples/golang-multi-stage-docker-build$ docker images
\REPOSITORY                          TAG       IMAGE ID       CREATED          SIZE
<none>                              <none>    128565bb9cf5   55 seconds ago   654MB
simplecalculator-multistagebuild    latest    5cb85f6879c4   55 seconds ago   1.96MB
simplecalculator                    latest    2c3ac535ca3e   10 hours ago     654MB
lokeshhh/django-containerization    latest    ea71f0673c37   20 hours ago     622MB
lokeshhh/my-first-docker-image      latest    2b5f3342f24f   42 hours ago     575MB
ubuntu                              latest    a04dc4851cbc   7 weeks ago      78.1MB
hello-world                         latest    74cc54e27dc4   7 weeks ago      10.1kB
```

Output :

- The command docker build -t simplecalculator .

Docker goes through each step of the Dockerfile, copying files, compiling the app, and finally exporting layers.

- This results in a **full-sized image** that includes Go compiler, dependencies, etc. (~654MB).

- And when we used command docker build -t simplecalculator-multistagebuild . this builds the Docker image using the optimized multi-stage Dockerfile.

- Docker performs the build in stages:

- First, compiles the app in a temporary container.

- Then creates a small production-ready image with just the binary.
- Result is a clean, secure image (~1.96MB) with no extra tools.

By This we achieve

1.Drastically Smaller Image Size

- From 654MB (normal) → ~1.96MB (multi-stage + distroless)
- Faster deployment, quicker image pulls, and less storage usage.

2.Improved Security

- Distroless images remove package managers, shells, and extra tools.
- This minimizes attack surface and follows the principle of least privilege.

3.Cleaner Separation of Concerns

- Multi-stage builds keep build tools in one stage and only copy the final artifact into the runtime stage.
- Results in clean, production-ready images.