



docker

# Docker Tutorials

---

Creating optimized  
Docker Images using  
Multi-Stage Builds

# Introduction

- In this scenario you will learn how to use the multi-stage build functionality to make smaller, more optimised images.
- The feature is ideal for deploying languages such as Golang as containers. By having multi-stage builds, the first stage can build the Golang binary using a larger Docker image as the base. In the second stage, the newly built binary can be deployed using a much smaller base image. The end result is an optimised Docker Image.

# Step 1 - Create Dockerfile❶

- The Multi-Stage feature allows a single Dockerfile to contain multiple stages in order to produce the desired, optimised, Docker Image.
- Previously, the problem would have been solved with two Dockerfiles. One file would have the steps to build the binary and artifacts using a development container, the second would be optimised for production and not include the development tools.
- By removing development tooling in the production image, you reproduce the attack surface and improve the deployment time.

# Step 1 - Create Dockerfile②

## Multi-Stage Dockerfile

Dockerfile.multi

```
1  # First Stage
2  FROM golang:1.6-alpine
3
4  RUN mkdir /app
5  ADD . /app/
6  WORKDIR /app
7  RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o main .
8
9  # Second Stage
10 FROM alpine
11 EXPOSE 80
12 CMD ["/app"]
13
14 # Copy from first stage
15 COPY --from=0 /app/main /app
```

# Step 2 - Build Multi-Stage Docker Image

- Create the desired Docker Image using the build command below.
  - *docker build -f Dockerfile.multi -t golang-app*
- The result will be two images. One untagged that was used for the first stage and the second, smaller image, our target image.
  - *docker images*

# Step 3 - Test Image

- The image can be launched and deployed without any changes required.
  - *docker run -d -p 80:80 golang-app*
  - *curl localhost*

# Reference

- <https://www.katacoda.com/courses/docker/multi-stage-builds>