Module-2: Image Management and Registry

Demo Document - 8

edureka!

DEMO-8: Multi-stage Builds

Note: All commands are executed as root.

To demonstrate how a multi-stage build is more efficient we will first create a normal image and then compare it to the one created with a multi-stage build.

1. Create a sample go application

```
$ vi sample.go
```

```
package main

import "fmt"

func main() {

    fmt.Println("All that we do, all that we are, begins and ends with ourselves")

}
```

2. Create a Dockerfile to run the application

```
$ vi Dockerfile
```

```
FROM golang:1.12.4

WORKDIR /sample

COPY sample.go .

RUN GOOS=linux go build -a -installsuffix cgo -o sample .

CMD ["./sample"]
```

3. Build the image and check the image size

```
$ docker build . -t nimage
```

```
root@docker-1:~/multiStage# docker build . -t nimage
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM golang:1.12.4
1.12.4: Pulling from library/golang
e79bb959ec00: Pull complete
d4b7902036fe: Pull complete
1b2a72d4e030: Pull complete
d54db43011fd: Pull complete
963c818ebafc: Pull complete
9eee6e7073aa: Pull complete
83e75b35417b: Pull complete
Digest: sha256:83e8267be041b3ddf6a5792c7e464528408f75c446745642db08cfe4e8d58d18
Status: Downloaded newer image for golang:1.12.4
 ---> b860ab44e93e
Step 2/5 : WORKDIR /sample
do ---> Running in 061df18e31d3
cRemoving intermediate container 061df18e31d3
 ---> 83cf5147e672
Step 3/5 : COPY sample.go .
ke ---> bb3aad41b3e2
Step 4/5 : RUN GOOS=linux go build -a -installsuffix cgo -o sample .
 ---> Running in 3fa6cf12b061
Removing intermediate container 3fa6cf12b061
 ---> e746d1ceeb1c
Step 5/5 : CMD ["./sample"]
 ---> Running in 9fe49eb74238
Removing intermediate container 9fe49eb74238
 ---> 1ce547157a20
Successfully built 1ce547157a20
Successfully tagged nimage:latest
root@docker-1:~/multiStage# docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
nimage                              latest              1ce547157a20        7 seconds ago       784MB
```

4. Now, let us edit the Dockerfile to create the multi-stage build

```
FROM golang:1.12.4 AS compiler

WORKDIR /sample

COPY sample.go .

RUN GOOS=linux go build -a -installsuffix cgo -o sample .


FROM alpine:3.9.3

WORKDIR /root

COPY --from=compiler /sample/sample.

CMD ["./sample"]
```

5. Build the multi-stage Dockerfile and check the image size

```
$ docker build . -t multistage
```

```
root@docker-1:~/multiStage# docker build . -t multistage
Sending build context to Docker daemon  3.072kB
Step 1/8 : FROM golang:1.12.4 AS compiler
 ---> b860ab44e93e
Step 2/8 : WORKDIR /sample
 ---> Using cache
 ---> 83cf5147e672
Step 3/8 : COPY sample.go .
 ---> Using cache
 ---> bb3aad41b3e2
Step 4/8 : RUN GOOS=linux go build -a -installsuffix cgo -o sample .
 ---> Using cache
 ---> e746d1ceeb1c
Step 5/8 : FROM alpine:3.9.3
3.9.3: Pulling from library/alpine
bdf0201b3a05: Pull complete
Digest: sha256:28ef97b8686a0b5399129e9b763d5b7e5ff03576aa5580d6f4182a49c5fe1913
Status: Downloaded newer image for alpine:3.9.3
 ---> cdf98d1859c1
Step 6/8 : WORKDIR /root
 ---> Running in a246528718d5
Removing intermediate container a246528718d5
 ---> fd21b7189950
Step 7/8 : COPY --from=compiler /sample/sample .
 ---> 57bb640f182f
Step 8/8 : CMD ["./sample"]
 ---> Running in 6f7c215514e8
Removing intermediate container 6f7c215514e8
 ---> 196425a181e9
Successfully built 196425a181e9
Successfully tagged multistage:latest
root@docker-1:~/multiStage# docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
multistage                          latest              196425a181e9        7 seconds ago       7.53MB
```