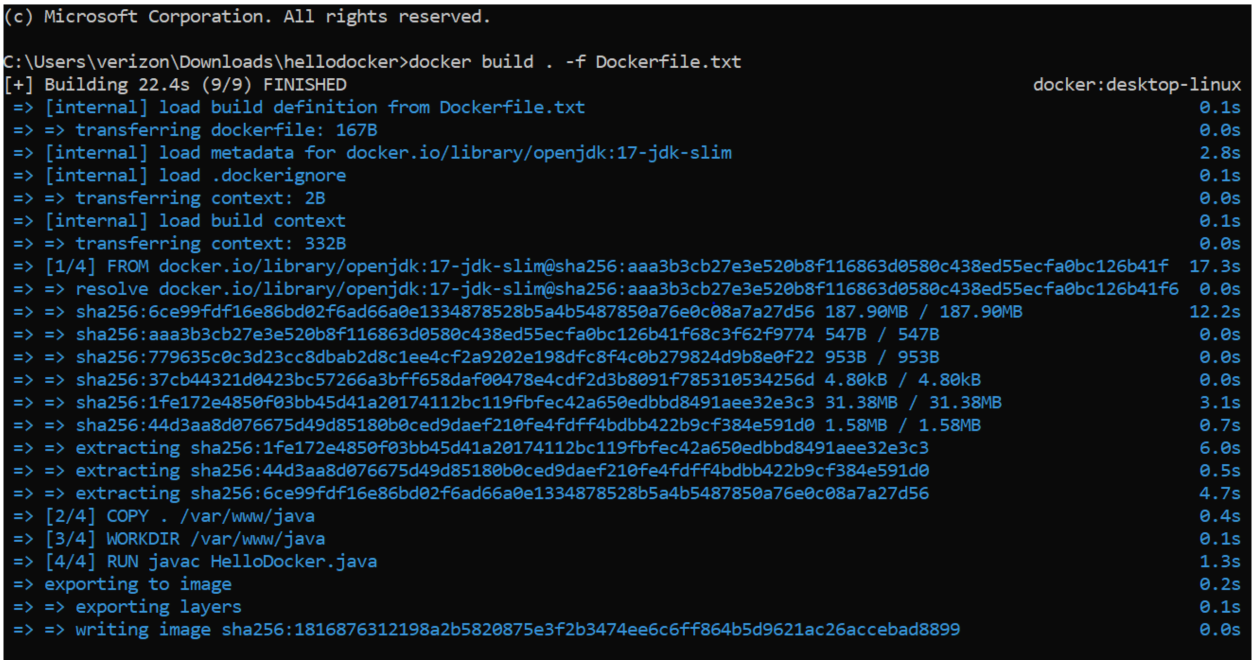
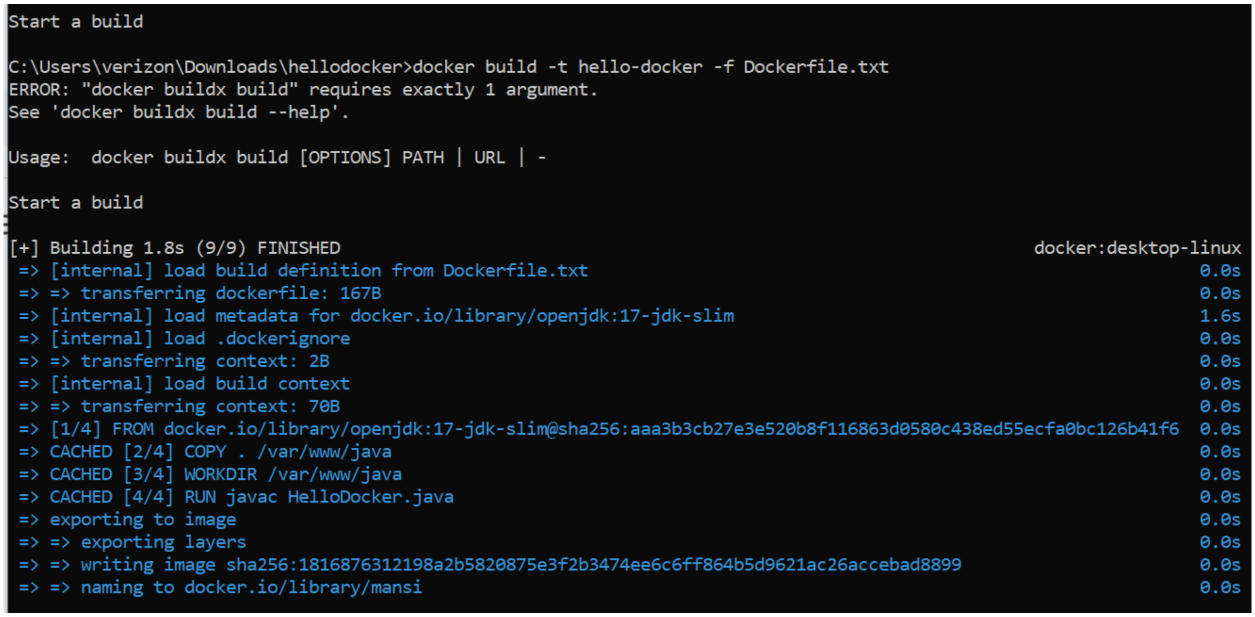
1. docker build . -f Dockerfile.txt



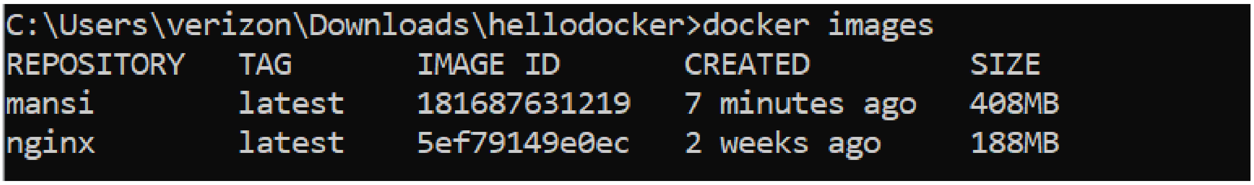
```
(c) Microsoft Corporation. All rights reserved.

C:\Users\verizon\Downloads\hellodocker>docker build . -f Dockerfile.txt
[+] Building 22.4s (9/9) FINISHED                                                                   docker:desktop-linux
 => [internal] load build definition from Dockerfile.txt                                                            0.1s
 => => transferring dockerfile: 167B                                                                                0.0s
 => [internal] load metadata for docker.io/library/openjdk:17-jdk-slim                                              2.8s
 => [internal] load .dockerignore                                                                                   0.1s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build context                                                                                   0.1s
 => => transferring context: 332B                                                                                   0.0s
 => [1/4] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f  17.3s
 => => resolve docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f6  0.0s
 => => sha256:6ce99fdf16e86bd02f6ad66a0e1334878528b5a4b5487850a76e0c08a7a27d56 187.90MB / 187.90MB                12.2s
 => => sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774 547B / 547B                         0.0s
 => => sha256:779635c0c3d23cc8dbab2d8c1ee4cf2a9202e198dfc8f4c0b279824d9b8e0f22 953B / 953B                         0.0s
 => => sha256:37cb44321d0423bc57266a3bff658daf00478e4cdf2d3b8091f785310534256d 4.80kB / 4.80kB                     0.0s
 => => sha256:1fe172e4850f03bb45d41a20174112bc119fbfec42a650edbbd8491aee32e3c3 31.38MB / 31.38MB                   3.1s
 => => sha256:44d3aa8d076675d49d85180b0ced9daef210fe4fdff4bdbb422b9cf384e591d0 1.58MB / 1.58MB                     0.7s
 => => extracting sha256:1fe172e4850f03bb45d41a20174112bc119fbfec42a650edbbd8491aee32e3c3                          6.0s
 => => extracting sha256:44d3aa8d076675d49d85180b0ced9daef210fe4fdff4bdbb422b9cf384e591d0                          0.5s
 => => extracting sha256:6ce99fdf16e86bd02f6ad66a0e1334878528b5a4b5487850a76e0c08a7a27d56                          4.7s
 => [2/4] COPY . /var/www/java                                                                                      0.4s
 => [3/4] WORKDIR /var/www/java                                                                                     0.1s
 => [4/4] RUN javac HelloDocker.java                                                                                1.3s
 => exporting to image                                                                                              0.2s
 => => exporting layers                                                                                             0.1s
 => => writing image sha256:1816876312198a2b5820875e3f2b3474ee6c6ff864b5d9621ac26accebad8899                       0.0s
```

2. Docker build -t mansi -f Dockerfile.txt.



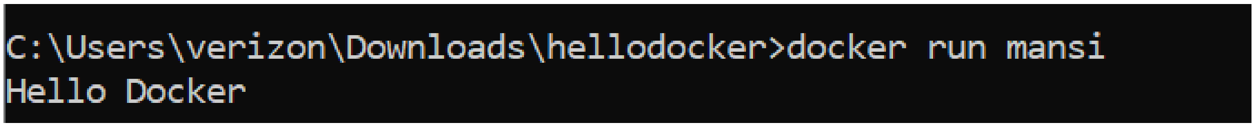
```
Start a build

C:\Users\verizon\Downloads\hellodocker>docker build -t hello-docker -f Dockerfile.txt
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Start a build

[+] Building 1.8s (9/9) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile.txt                                                            0.0s
 => => transferring dockerfile: 167B                                                                                0.0s
 => [internal] load metadata for docker.io/library/openjdk:17-jdk-slim                                              1.6s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 70B                                                                                    0.0s
 => [1/4] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f6  0.0s
 => CACHED [2/4] COPY . /var/www/java                                                                               0.0s
 => CACHED [3/4] WORKDIR /var/www/java                                                                              0.0s
 => CACHED [4/4] RUN javac HelloDocker.java                                                                         0.0s
 => exporting to image                                                                                              0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:1816876312198a2b5820875e3f2b3474ee6c6ff864b5d9621ac26accebad8899                       0.0s
 => => naming to docker.io/library/mansi                                                                            0.0s
```

3. Docker images



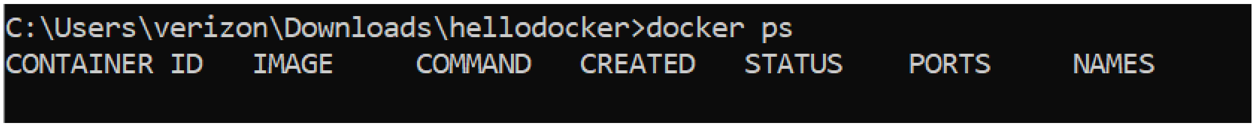
```
C:\Users\verizon\Downloads\hellodocker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
mansi        latest    181687631219   7 minutes ago   408MB
nginx        latest    5ef79149e0ec   2 weeks ago     188MB
```

4. Docker run mansi-docker



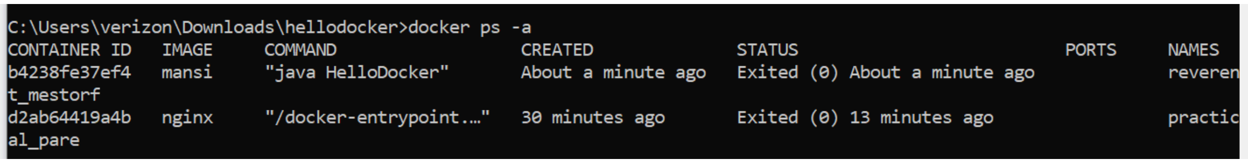
```
C:\Users\verizon\Downloads\hellodocker>docker run mansi
Hello Docker
```

5. Docker ps



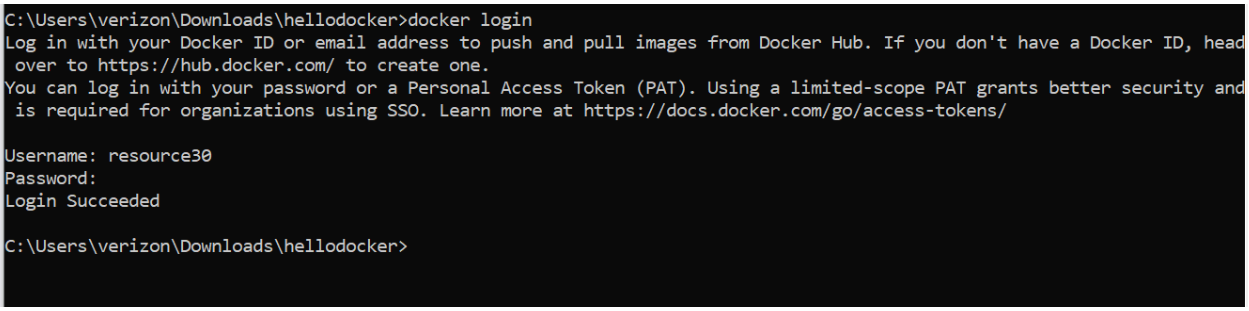
```
C:\Users\verizon\Downloads\hellodocker>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

6. Docker ps -a



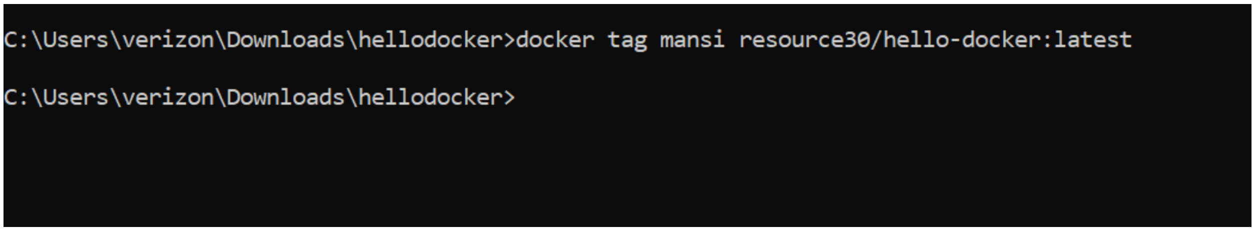
```
C:\Users\verizon\Downloads\hellodocker>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                          PORTS     NAMES
b4238fe37ef4   mansi     "java HelloDocker"       About a minute ago   Exited (0) About a minute ago             reveren
t_mestorf
d2ab64419a4b   nginx     "/docker-entrypoint.…"   30 minutes ago       Exited (0) 13 minutes ago                 practic
al_pare
```

7. Docker login



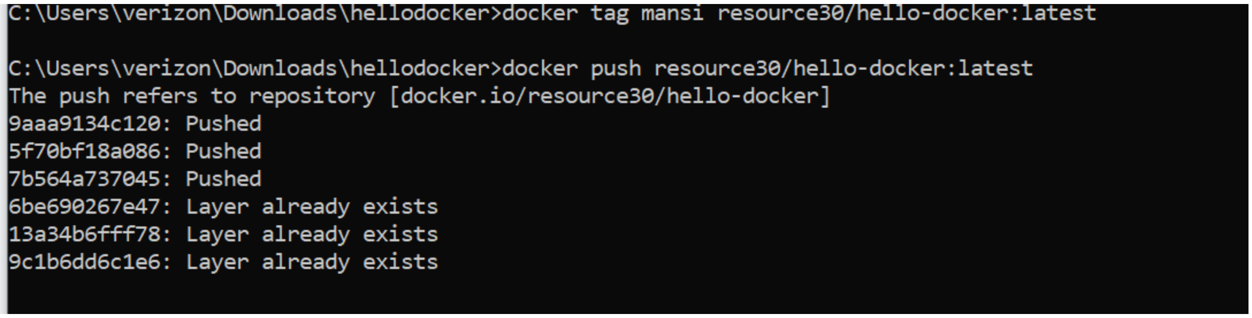
```
C:\Users\verizon\Downloads\hellodocker>docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head
 over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and
 is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: resource30
Password:
Login Succeeded

C:\Users\verizon\Downloads\hellodocker>
```

8. Docker tag mansi resource30/hello-docker:latest



```
C:\Users\verizon\Downloads\hellodocker>docker tag mansi resource30/hello-docker:latest

C:\Users\verizon\Downloads\hellodocker>
```

9. Docker push resource30/hello-docker:latest



```
C:\Users\verizon\Downloads\hellodocker>docker tag mansi resource30/hello-docker:latest

C:\Users\verizon\Downloads\hellodocker>docker push resource30/hello-docker:latest
The push refers to repository [docker.io/resource30/hello-docker]
9aaa9134c120: Pushed
5f70bf18a086: Pushed
7b564a737045: Pushed
6be690267e47: Layer already exists
13a34b6fff78: Layer already exists
9c1b6dd6c1e6: Layer already exists
```