

## Docker Task

### Dockerfile Concept

#### optimize a Docker image

Sure, here's a step-by-step guide to optimize a Docker image

#### Step 1: Create a new directory and navigate into it

Open your terminal.

Create a new directory named `kajal`

```
ubuntu $ mkdir kajal
```

Navigate into the `kajal` directory:

```
ubuntu $ cd kajal
```

#### Step 2: Create and edit a Dockerfile

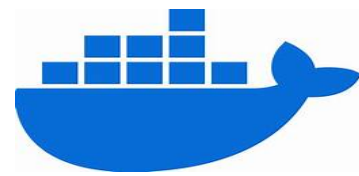
Create a new Dockerfile using a text editor (e.g., `vi`, `nano`, etc.):

```
ubuntu $ vi Dockerfile
```

Add the necessary instructions in the Dockerfile.

Save and exit the editor (`:wq` for `vi`)

```
FROM ubuntu
RUN apt update -y
RUN apt install apache2 -y
RUN useradd kajal
RUN touch MYFILE.TXT
COPY . /mnt/test.txt
```



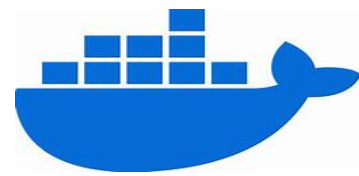
### Step 3: Build the Docker image

Build the Docker image and tag it as **myapp**

```
ubuntu $ docker build -t myapp .
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  2.048kB
Step 1/6 : FROM ubuntu
latest: Pulling from library/ubuntu
9c704ecd0c69: Pull complete
Digest: sha256:2e863c44b718727c860746568e1d54afd13b2fa71b160f5cd9058fc436217b
30
Status: Downloaded newer image for ubuntu:latest
--> 35a88802559d
Step 2/6 : RUN apt update -y
--> Running in 5c56fba9fb21

done.
Removing intermediate container a39cda372e95
--> 8f14964c8972
Step 4/6 : RUN useradd kajal
--> Running in 33b779191541
Removing intermediate container 33b779191541
--> e104b41f6a01
Step 5/6 : RUN touch MYFILE.TXT
--> Running in 30d563d6f8b1
Removing intermediate container 30d563d6f8b1
--> f88bade477ec
Step 6/6 : COPY . /mnt/test.txt
--> 144b37c25692
Successfully built 144b37c25692
Successfully tagged myapp:latest
ubuntu $
```



## Step 4: List Docker images

List all Docker images to verify that **myapp** has been created

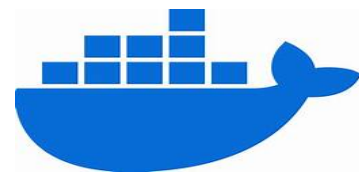
```
ubuntu $ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
myapp         latest    144b37c25692   About a minute ago  223MB
ubuntu        latest    35a88802559d   6 weeks ago    78.1MB
ubuntu $
```

## Step 5: Inspect the Docker image

Inspect the **myapp** Docker image

```
ubuntu $ docker inspect myapp
```

```
dca960ebd29bb273962793056b76d5e0697af71a/work"
    },
    "Name": "overlay2"
  },
  "RootFS": {
    "Type": "layers",
    "Layers": [
      "sha256:a30a5965a4f7d9d5ff76a46eb8939f58e95be844de1ac4a4b452d5d31158fdea",
      "sha256:652644a8afa8baffac571487fd70c1fa6e8cb34a9ed32e49a5bbe79537471d2",
      "sha256:3dc046e279703bff5d189f4c3590bb8d66148d61bd6212d227c3c07fff4fb1a7",
      "sha256:c223dbfba534214dc8a726dfa621bf92d5246489af7c673ba6d03c14681c05ee",
      "sha256:ffe8c48d7665856d174d96cd19ba5fa74540de0266e27c81d926ffa1350dbee9",
      "sha256:d4d9540b63e1d66f2562fe000b398f9d5ac96e4aedd1f7d39b76090a351ae80d"
    ]
  },
  "Metadata": {
    "LastTagTime": "2024-07-23T05:46:20.427940999Z"
  }
}
```



## Step 6: Create another directory and navigate into it (optimize a Docker image)

Create a new directory named `payal`

Navigate into the `payal` directory

```
ubuntu $ mkdir payal
ubuntu $ cd payal
ubuntu $
```

## Step 7: Create and edit another Dockerfile

Create another Dockerfile using a text editor:

```
ubuntu $ vi Dockerfile
ubuntu $
```

Add the necessary instructions in the new Dockerfile.  
Save and exit the editor (`:wq` for `vi`).

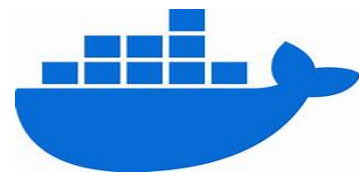
```
FROM ubuntu
RUN apt update -y && apt install apache2 -y && useradd kajal && touch MYFILE.
COPY . /mnt/test.txt
```

## Step 8: Build the new Docker image

Build the Docker image and tag it as `myapp1`

```
ubuntu $ docker build -t myapp1 .
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  2.048kB
Step 1/3 : FROM ubuntu
--> 35a88802559d
Step 2/3 : RUN apt update -y && apt install apache2 -y && useradd kajal && touch MYFILE.TXT
--> Running in 7c7506a8c85b
```



```
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Removing intermediate container 7c7506a8c85b
---> a541f89fe519
Step 3/3 : COPY . /mnt/test.txt
---> bcf1726324a0
Successfully built bcf1726324a0
Successfully tagged myapp1:latest
```

## Step 9: List Docker images

List all Docker images to verify that **myapp1** has been created

```
ubuntu $ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
myapp1        latest    bcf1726324a0   22 seconds ago 223MB
myapp         latest    144b37c25692   8 minutes ago  223MB
ubuntu        latest    35a88802559d   6 weeks ago   78.1MB
ubuntu $
```

## Step 10: Inspect the new Docker image

Inspect the **myapp1** Docker image:

```
ubuntu $ docker inspect myapp1
[
  {
    "RootFS": {
      "Type": "layers",
      "Layers": [
        "sha256:a30a5965a4f7d9d5ff76a46eb8939f58e95be844de1ac4a4b452d5d31158fdea",
        "sha256:98304a1b31ea2064421a134c303c02b8d4d3d4356632e710c17b6e951d8b8b39",
        "sha256:6fe903a657fab97e75b34fcbca060f6c321bfe1fc77527ce371d6453c190500b"
      ]
    },
    "Metadata": {
      "LastTagTime": "2024-07-23T05:54:36.901789219Z"
    }
  }
]
```

By following these steps, you'll create and build two separate Docker images with different setups, and inspect them to understand their configurations.