

Step 1

Create a Dockerfile

```
$ mkdir Dockerfiles
```

```
$ cd Dockerfiles
```

```
$ vim Dockerfile
```

Type i - to insert text

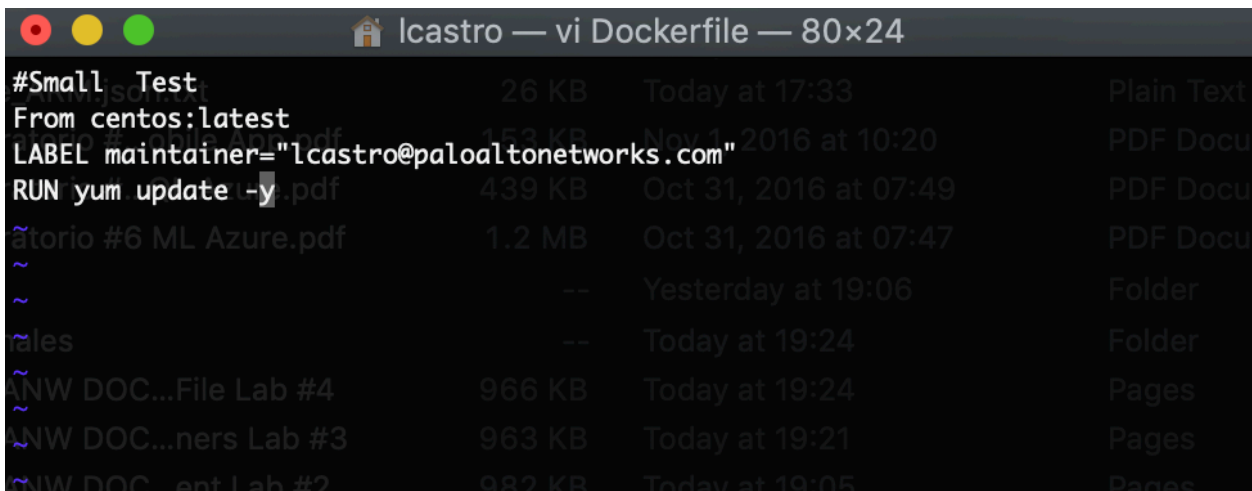
```
# Small Test
```

```
FROM centos:latest
```

```
LABEL maintainer="lcastro@paloaltonetworks.com"
```

```
RUN yum update -y
```

Type ESC : wq! + Enter to exit vim



The screenshot shows a terminal window titled "lcastro — vi Dockerfile — 80x24". The terminal displays the content of the Dockerfile being edited in vim. The text is as follows:

```
#Small Test
From centos:latest
LABEL maintainer="lcastro@paloaltonetworks.com"
RUN yum update -y
```

The terminal also shows a file manager sidebar on the right with various files and folders, including "26 KB", "Today at 17:33", "Plain Text", "153 KB", "Nov 1, 2016 at 10:20", "PDF Docu", "439 KB", "Oct 31, 2016 at 07:49", "PDF Docu", "1.2 MB", "Oct 31, 2016 at 07:47", "PDF Docu", "--", "Yesterday at 19:06", "Folder", "--", "Today at 19:24", "Folder", "966 KB", "Today at 19:24", "Pages", "963 KB", "Today at 19:21", "Pages", and "982 KB", "Today at 19:05", "Pages".

Step 2

Create a Custom Image from Docker File

```
$ docker build -t custom_image:v1 .
```

```
[ec2-user@ip-172-31-50-64 Dockerfiles]$ docker build -t custom_image:v2 .
Sending build context to Docker daemon 2.048kB
Step 1/3 : FROM centos:latest
--> 470671670cac
Step 2/3 : LABEL maintainer="lcastro@paloaltonetworks.com"
--> Using cache
--> 905267e0f338
Step 3/3 : RUN yum update -y
--> Using cache
--> 0eb45d09432c
Successfully built 0eb45d09432c
Successfully tagged custom_image:v2
[ec2-user@ip-172-31-50-64 Dockerfiles]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
custom_image	v1	0eb45d09432c	52 seconds ago	377MB
custom_image	v2	0eb45d09432c	52 seconds ago	377MB
ubuntu	latest	4e5021d210f6	11 days ago	64.2MB
centos	latest	470671670cac	2 months ago	237MB
mongo	3.3.11	97d969a5eca5	3 years ago	425MB

```
[ec2-user@ip-172-31-50-64 Dockerfiles]$
```

```
$ docker images
```

```
Luis-MacBook-Air:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
customimage	v1	30df6ac8fc35	2 minutes ago	4.15MB

Step 3

Rebuild image using Cache

```
$ docker build -t custom_image:v2 .
```

```
$ docker images
```

```
[ec2-user@ip-172-31-50-64 Dockerfiles]$ docker build -t custom_image:v2 .
Sending build context to Docker daemon 2.048kB
Step 1/3 : FROM centos:latest
--> 470671670cac
Step 2/3 : LABEL maintainer="lcastro@paloaltonetworks.com"
--> Using cache
--> 905267e0f338
Step 3/3 : RUN yum update -y
--> Using cache
--> 0eb45d09432c
Successfully built 0eb45d09432c
Successfully tagged custom_image:v2
[ec2-user@ip-172-31-50-64 Dockerfiles]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
custom_image	v1	0eb45d09432c	52 seconds ago	377MB
custom_image	v2	0eb45d09432c	52 seconds ago	377MB
ubuntu	latest	4e5021d210f6	11 days ago	64.2MB
centos	latest	470671670cac	2 months ago	237MB
mongo	3.3.11	97d969a5eca5	3 years ago	425MB

```
[ec2-user@ip-172-31-50-64 Dockerfiles]$
```

Step 4

Remove Dockerfile

rm Dockerfile

Step 5

Create an image from Docker Hub

\$ sudo yum install git

\$ git clone https://github.com/nigelpoulton/psweb.git

\$ cd psweb

\$ ls -l

```
-rw-r--r-- 1 root root 341 Sep 29 16:26 app.js
-rw-r--r-- 1 root root 216 Sep 29 16:26 circle.yml
-rw-r--r-- 1 root root 338 Sep 29 16:26 Dockerfile
-rw-r--r-- 1 root root 421 Sep 29 16:26 package.json
-rw-r--r-- 1 root root 370 Sep 29 16:26 README.md
drwxr-xr-x 2 root root 4096 Sep 29 16:26 test
drwxr-xr-x 2 root root 4096 Sep 29 16:26 views
```

\$ cat Dockerfile

```
FROM alpine
LABEL maintainer="nigelpoulton@hotmail.com"
RUN apk add --update nodejs nodejs-npm
COPY . /src
WORKDIR /src
RUN npm install
EXPOSE 8080
ENTRYPOINT ["node", "./app.js"]
```

\$ docker image build -t web:latest .

\$ docker image ls

REPO	TAG	IMAGE ID	CREATED	SIZE
web	latest	fc69fdc4c18e	10 seconds ago	64.4MB

\$ docker image inspect web:latest

ID	IMAGE	COMMAND	STATUS	PORTS
49..	web:latest	"node ./app.js"	UP 6 secs	0.0.0.0:80->8080/tcp

