

Creating Own image using Dockerfile

Description: We're going to start building our own Docker images.

Dockerfile:

A Dockerfile is a text file that contains instructions for building a Docker image. An image is a read-only template that includes all the files, dependencies, and configuration needed to run a container. A Dockerfile provides a way to automate the process of building an image, ensuring that the resulting image is consistent and reproducible.



KEYWORDS:

FROM: The FROM instruction specifies the base image to use for the Docker container. This is the starting point for building the image, and all subsequent instructions will build on top of it.

Syntax:

FROM <image-

name>**Example:FROM**

RUN: The RUN instruction executes a command in the Docker container during the build process. This can be used to install packages, update software, or perform other tasks necessary for your application.

Syntax:

```
RUN<command>
```

Example:

```
RUN apt-get update && apt-get install -y git
```

MAINTAINER: In a Dockerfile, the MAINTAINER keyword is used to specify the name and email address of the person or organization that is responsible for maintaining the Docker image.

Syntax:

```
MAINTAINER<name>[<email>]
```

Example:

```
MAINTAINER krishna<krishna04.b@gmail.com>
```

CMD: The CMD instruction specifies the command that should be executed when the container starts running. This is typically used to start the main process in the container, such as a web server or application.

Syntax:

```
CMD["executable","param1","param2",...]
```

Example:

```
CMD["python","app.py"]
```

ENTRYPOINT: The ENTRYPOINT instruction specifies the command that should be executed when the container starts running, just like CMD. However, the difference is that the commands specified in ENTRYPOINT cannot be overridden by the command line arguments passed when starting the container.

Syntax:

```
ENTRYPOINT["executable","param1","param2",...]
```

Example:

```
ENTRYPOINT["python"]  
CMD["app.py"]
```

ADD/COPY: The ADD and COPY instructions are used to add files from the host system to the Docker image. COPY is preferred over ADD, as it is simpler and less powerful.

Syntax:

COPY<src><dest>

Example:

COPYapp.py/app/

EXPOSE: The EXPOSE instruction informs Docker that the container will listen on the specified network ports at runtime. This does not actually publish the port, but simply documents the intended usage.

Syntax:

EXPOSE<port>

Example:

EXPOSE8080

Procedure:

```
[root@ip-172-31-30-197 ~]# history
1  yum install docker -y
2  systemctl restart docker
3  systemctl status docker
4  docker info
```

vim index.html

```
<html>
<head>
  <title>HOME</title>
</head>
<body bgcolor="pink">
<marquee> WELCOME TO AITS-RAJAMPET</marquee>
<center>
  <h2>Employee Details</h2>

  <table border="1" cellspacing="0" cellpadding="8">
    <thead>
      <tr>
        <th>Name</th>
        <th>Department</th>
        <th>Position</th>
        <th>Salary</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Krishna</td>
        <td>IT</td>
        <td>Software Developer</td>
        <td>$70,000</td>
      </tr>
      <tr>
        <td>Milky</td>
        <td>HR</td>
        <td>HR Manager</td>
        <td>$80,000</td>
      </tr>
      <tr>
        <td>Cherry</td>
        <td>Finance</td>
        <td>Accountant</td>
        <td>$65,000</td>
      </tr>
    </tbody>
  </table>
</center>
</body>
</html>
```

vim Dockerfile

```
ARG NGINX_VERSION=latest
FROM nginx:${NGINX_VERSION}
ADD index.html /usr/share/nginx/html/index.html
WORKDIR ["nginx","-g","daemon off;"]
EXPOSE 80
LABEL description="krishna"
ENV APP_VERSION=1.0
```

```
[root@ip-172-31-30-197 ~]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
myimage       latest    f6c7dbda9465   16 minutes ago 187MB
nginx         latest    2a36393edaf1   33 hours ago   187MB
```

docker build -t myimage .

docker run -d -p 81:80 --name mycontainer myimage:latest

```
docker ps
[root@ip-172-31-30-197 ~]# docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS
PORTS         NAMES
5bba9b26bed3   myimage:latest       "/docker-entrypoint...." 18 minutes ago Up 18 minutes
0.0.0.0:81->80/tcp, :::81->80/tcp   mycontainer
```

RESULT:

The screenshot shows a web browser window with the URL 34.235.87.66:81. The page has a pink background and displays a welcome message 'WELCOME TO AITS-RAJAMPET'. Below the message is a table titled 'Employee Details' with the following data:

Name	Department	Position	Salary
Krishna	IT	Software Developer	\$70,000
Milky	HR	HR Manager	\$80,000
Cherry	Finance	Accountant	\$65,000