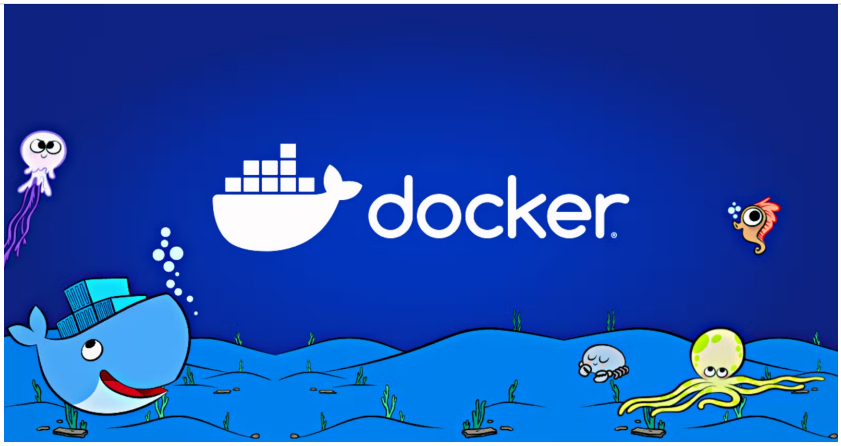


Docker for DevOps Engineers

Kunal Maurya · Mar 16, 2023 · 5 min read

TABLE OF CONTENTS

Docker is a powerful containerization platform that has revolutionized the way software applications are built and deployed. It provides a lightweight and portable way to package an application, along with its dependencies, into a single container that can be run anywhere, from a developer's laptop to a production server.

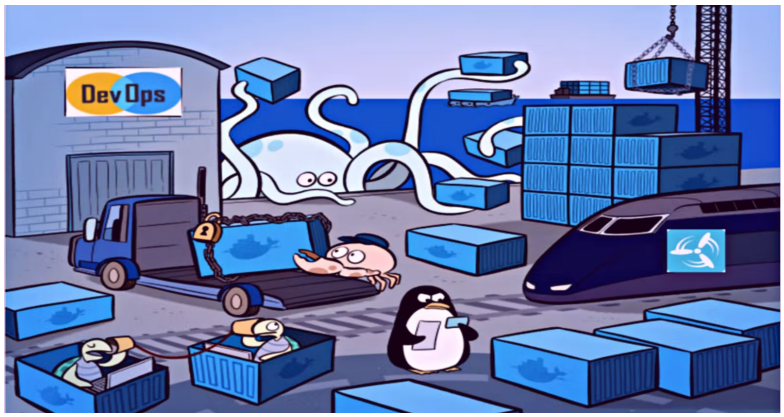
In this blog, we will explore what Docker is, how it works, and some of its key features.

What is Docker?

Docker is a containerization platform that enables developers to package an application and its dependencies into a single container. A container is a lightweight and standalone executable package that includes everything needed to run the application, including libraries, system tools, and runtime. Containers are isolated from each other and from the host system, which makes them secure and portable.

How does Docker work?

Docker uses a client-server architecture, where the Docker client communicates with the Docker daemon, which manages the containers. The Docker daemon runs on the host machine and is responsible for building, running, and managing the containers. The Docker client provides a command-line interface and a graphical user interface to interact with the Docker daemon.



What is Dockerfile?

To create a Docker container, you need to define a Dockerfile, which is a text file that contains the instructions to build the container. The Dockerfile specifies the base image, which is the operating system and runtime environment for the container, as well as any additional packages and configurations needed for the application. Once the Dockerfile is defined, you can use the docker build command to build the container.

Here is a table of some common Dockerfile instructions and their purpose:

Instruction	Purpose
FROM	Specifies the base image to use for the container
RUN	Runs a command during the image build process
CMD	Specifies the command to run when the container starts
LABEL	Adds metadata to the image
EXPOSE	Specifies which ports the container should listen on
ENV	Sets environment variables for the container
ADD	Copies files or directories from the build context into the image
COPY	Copies files or directories from the build context into the image
ENTRYPOINT	Specifies the executable to run when the container starts
VOLUME	Specifies a directory in the image that should be used as a volume

Key Features of Docker

1. Portability - Docker containers can run on any machine that supports Docker, which makes them highly portable and flexible.
2. Isolation - Docker containers are isolated from each other and from the host system, which makes them secure and reduces the risk of conflicts between applications.
3. Lightweight - Docker containers are lightweight and use minimal resources, which makes them ideal for deploying applications on resource-constrained environments.
4. Reproducibility - Docker containers provide a consistent environment for the application, which makes it easier to reproduce issues and debug problems.

Docker Commands

Check out how to create a dockerfile and containerized app:
https://docs.docker.com/get-started/02_our_app/#get-the-app

- **docker build** - This command is used to build a new image from a Dockerfile.
  Syntax:

```
docker build -t imageName .
```

- **docker images**- This command is used to list all available images.

Example:

```
docker images
```

- **docker run -** This command is used to create and start a new container.

- **docker ps** - This command is used to list all running containers.

- **docker inspect** - This command is used to get detailed information about a Docker object, such as a container, image, network, or volume.

Example:

```
docker inspect my-container
```

- **docker port**- This command is used to list the public-facing ports of a container.
  Example:

```
docker port my-container
```

  Output:

```
3306/tcp -> 0.0.0.0:3306
```

- **docker stats**-This command is used to display live system resource usage of one or more containers.
  Example:

```
docker stats my-container
```

- **docker save**:This command is used to save one or more Docker images to a tar archive. The syntax for the command is as follows:

```
docker save [-o|--output FILE] IMAGE [IMAGE...]
```

  - `-o|--output FILE`: Specifies the output file name and location. If not specified, the image is saved to stdout.
  - `IMAGE [IMAGE...]`: Specifies one or more images to save.

Example:

```
docker save -o my-image.tar my-image:latest
```

This command will save the `my-image:latest` Docker image to a tar archive named `my-image.tar`.

- **docker load**: The `docker load` command is used to load Docker images from a tar archive. The syntax for the command is as follows:

```
docker load [-i|--input FILE]
```

  - `-i|--input FILE`: Specifies the input file name and location. If not specified, the image is loaded from stdin.

Example:

```
docker load -i my-image.tar
```

This command will load the Docker image stored in the `my-image.tar` tar archive into the Docker registry.

- **docker stop**- This command is used to stop a running container.

Example:

```
docker stop my-container
```

Output:

The container named `my-container` is stopped.

- **docker exec**- This command is used to run a command within a running container.

Example:

```
docker exec my-container ls -l /
```

Output:

```
total 4
drwxr-xr-x   2 root root 4096 Mar 15 12:34 bin
drwxr-xr-x   2 root root 4096 Apr 13  2021 boot
drwxr-xr-x   4 root root 4096 Mar 16 11:32 dev
...
```

In this example, `docker exec` is used to run the `ls -l /` command within the running container named `my-container`. The output shows the directory listing of the root directory within the container.

Thank you for reading! Hope you find this article helpful.

~Kunal

Docker docker images Dockerfile containerization

MORE ARTICLES

Kunal Maurya

Python for Devops

Python has become a widely used programming language in the world of DevOps due to its versatility a...

Kunal Maurya

Git And Linux Cheatsheet

Git Cheatsheet CommandDescription git initInitialize a new git repository git clone [url]Clo...

Kunal Maurya

Day 11 : Advance Git for DevOps Engineers

Git Stash Git stash allows developers to save changes in a temporary location, which can be retrieve...