

Introduction to Docker

**Build once, run anywhere with Docker -
the key to portable and scalable
applications**



What is Docker?



- Docker is a platform for building, shipping, and running applications in containers
- Containers are lightweight, portable, and self-contained environments that package an application along with its dependencies and configuration into a single unit that can run consistently across different environments

Benefits of Docker



- **Portability:** Docker allows you to package your application into a container that can run anywhere, regardless of the underlying infrastructure
- **Consistency:** Docker ensures that your application runs the same way in any environment, without worrying about dependencies or compatibility issues

- **Scalability:** Docker allows you to easily scale your application by running multiple containers in parallel
- **Collaboration:** Docker allows you to easily share and deploy your application with others, without worrying about environment setup or configuration

Creating a Dockerfile



- A Dockerfile is a configuration file that describes how to build a Docker image, which is a snapshot of a container that includes the application code and its dependencies
- We can create a Dockerfile for our web application that starts from an existing image of a web server and copies our static files into it:

```
FROM nginx:alpine
```

```
COPY static /usr/share/nginx/html
```

Building the Image

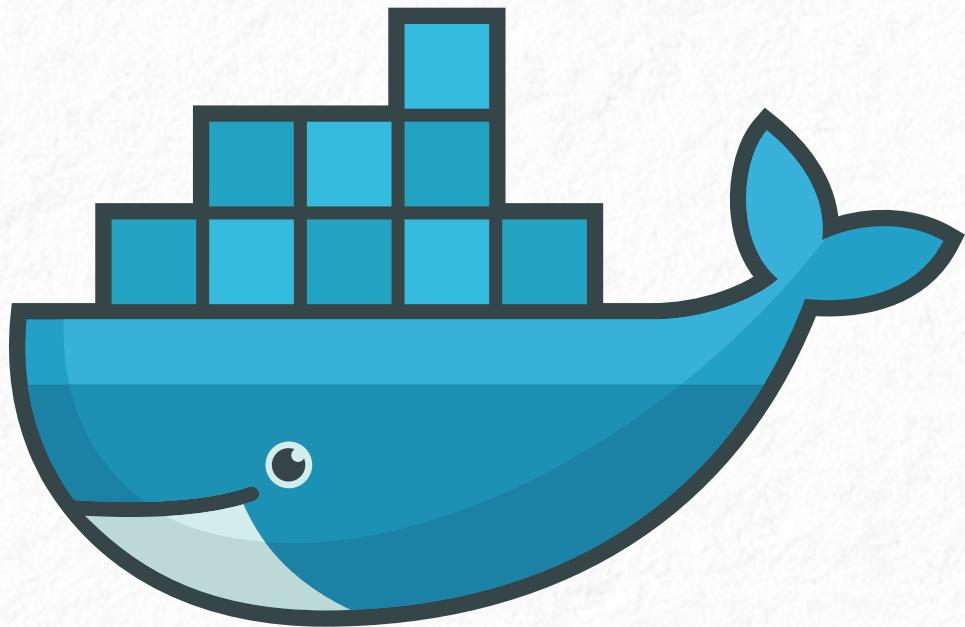


We can use the Docker CLI to build the image from the Dockerfile, by running the following command in the same directory where the Dockerfile is located:

```
docker build -t mywebapp .
```

This will create a new image called "mywebapp" based on the instructions in the Dockerfile

Running the Container



Now that we have an image, we can use it to start a new container that runs our web application

We can do this by running the following command:

```
docker run -d -p 80:80 mywebapp
```

This will start a new container based on the "mywebapp" image, and map the container's port 80 to the host machine's port 80, so that we can access the web application through a web browser

Testing the web Application



We can now open a web browser and navigate to `http://localhost` to see our static content served by the web server running inside the Docker container.

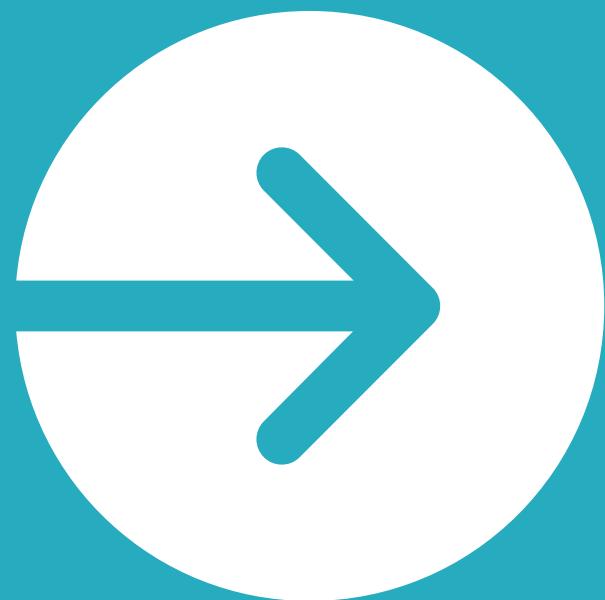
By using Docker, we have created a portable and reproducible environment for running our web application.

Docker allows us to scale our application by running multiple containers in parallel, and manage them with tools such as Docker Compose or Kubernetes

Docker makes collaboration and deployment of our application easier and more efficient.

Enjoyed
this?

One favor
to ask...



Sharing

=

caring



"Be a good friend.

**Support free content with
a repost."**



@devops-expert