# How To Containerize MERN Stack with Podman

-> A step-by-step guide with an example project

Containerization is a popular method for packaging and deploying applications in a self-contained environment, isolated from the host operating system. Podman is an open-source container engine that provides a similar interface to Docker but without the need for a daemon process, making it easier to use on Windows machines. In this blog, we will walk you through the process of containerizing a MERN (MongoDB, Express.js, React.js, and Node.js) application using Podman on Windows.

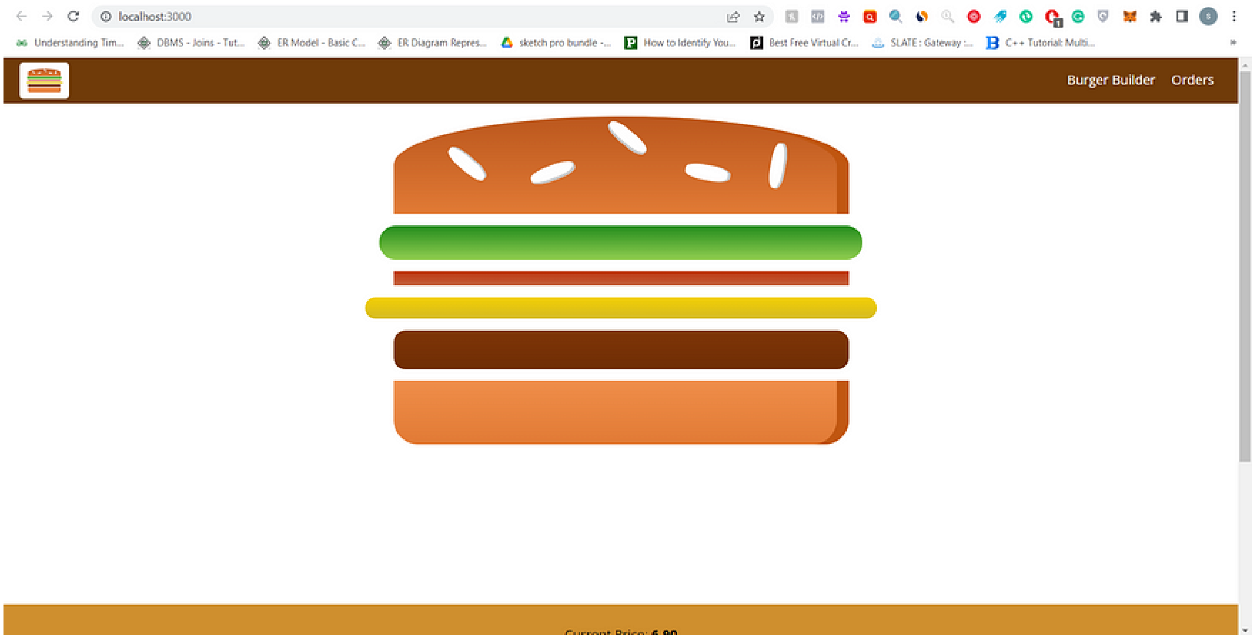
## Prerequisites

Before we begin, you will need to have the following software installed on your Windows machine:

- Podman: Podman can be installed from the official website at https://podman.io/getting-started/installation.html#windows

- Git: Git can be installed from https://git-scm.com/downloads

You will also need a MERN application. If you don’t have one, you can use the sample MERN application available at https://github.com/SaadBajwa24/Mern-Podman/tree/Feature-Branch



## Step 1: Build the MERN Application

Navigate to your MREN directory and build the application using the following commands:

```
cd react-burger-app
```

```
npm install
```

```
npm run build
```

## Step 2: Create a Dockerfile

Create a file named `Dockerfile` in the root directory of your MERN application with the following content

```
FROM node:14-alpine

WORKDIR /app

COPY package*.json ./

RUN npm install --production
```

```
COPY . .
```

```
CMD [ "npm", "start" ]
```

This Dockerfile sets up a Node.js environment, installs production dependencies, and runs the start command of the application.

## Step 3: Build the Docker image

To build the Docker image for the MERN application, run the following command:

```
podman build -t my-mern-app .
```

This command builds the Docker image with the tag `my-mern-app`.

## Step 4: Create the Podman container

To create a Podman container from the Docker image, run the following command:

```
podman create -p 8080:8080 --name my-mern-container my-mern-app
```

This command creates a Podman container with the name `my-mern-container`, mapping port 8080 of the container to port 8080 of the host machine.

## Step 5: Start the Podman container

To start the Podman container, run the following command:

```
podman start my-mern-container
```

This command starts the container with the name `my-mern-container`.

## Step 6: Test the MERN application

Open a web browser and navigate to [http://localhost:8080](http://localhost:8080) to test the MERN application. You should see the application running in the browser.

That’s it! You have successfully containerized your MERN application with Podman and created a Podman container to run it. You can now deploy and run your application on any machine with Podman installed.

## Conclusion

In this blog, we have shown you how to containerize a MERN application using Podman on Windows. We have walked you through cloning the MERN application, building the application, creating a Podman image, running the Podman container, and testing the application. With containerization, you can quickly deploy and run your application on any machine with Podman installed, without worrying about dependencies or environmental issues.