

# Docker Images Without Docker Using Podman — A Practical Guide

**D**ocker is a powerful containerization platform used by developers to build and deploy applications in a containerized environment. However, not all users have access to Docker, especially those using Windows operating systems. Fortunately, there are alternatives to Docker that can be used to create and manage container images on Windows. In this blog, we will explore how to create Docker images without Docker, specifically for Windows users.

## ***Why Create Docker Images Without Docker?***

Docker provides a straightforward way to create, build, and manage container images. However, there are several reasons why a Windows user may want to create Docker images without using Docker. Some possible reasons are:

- Docker is not supported on their operating system or architecture.
- They want to use an alternative to Docker for creating container images.
- They want to use a lightweight solution that requires fewer resources than Docker.

Whatever the reason may be, the good news is that there are alternative tools that can be used to create and manage container images without Docker.

## Alternative Tools to Create Docker Images Without Docker

**B***uildah* is a lightweight tool that allows users to build and manage container images without Docker. It is designed to work in a non-privileged environment, making it ideal for developers who do not have access to root privileges on their systems. Buildah can build images in a variety of formats, including OCI and Docker, and it integrates well with Kubernetes and other container orchestration platforms.

**P***odman* is another tool that can be used to create and manage container images without Docker. It is designed to be a drop-in replacement for Docker, providing users with a similar experience to Docker. Podman can build and manage images in a variety of formats, including Docker, and it is also compatible with Kubernetes and other container orchestration platforms.

**K***aniko* is a tool developed by Google that allows users to build container images without requiring a Docker daemon. It is designed to

be used in a containerized environment, making it ideal for building images in a Kubernetes cluster. Kaniko can build images in Docker format, and it can also push images to a Docker registry.

This blog provides a detailed guide on “How to Create & Manage Docker Images Using Podman In Windows”.



### *Prerequisites:*

Before getting started, ensure that you have the following prerequisites in place:

- A Windows machine with Podman installed
- A text editor (such as Notepad++) installed

## *Step 1: Installing Podman*

To install Podman on Windows, follow the steps below:

1. Open PowerShell as an administrator.
2. Run the following command to install the Chocolatey package manager:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-  
Object  
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

3. Run the following command to install Podman:

```
choco install podman
```

## *Step 2: Creating a Docker Image*

To create a Docker image using Podman, follow the steps below:

1. Open PowerShell and create a new directory for your project:

```
mkdir my-project
```

2. Change into the new directory:

```
cd my-project
```

3. Create a new file named `Dockerfile` in your project directory using your preferred text editor:

```
touch Dockerfile
```

4. Open the `Dockerfile` in your text editor and add the following lines:

```
FROM alpine:3.10
RUN apk add --no-cache python3
COPY . /app
WORKDIR /app
CMD ["python3", "app.py"]
```

This `Dockerfile` specifies an image that uses the Alpine Linux distribution as its base image and installs Python 3 on top of it. It then copies the contents of the current directory into the `/app` directory within the image, sets the working directory to `/app`, and specifies the command to run when the container is started.

5. Save and close the `Dockerfile`.

6. Build the Docker image using the following command:

```
podman build -t my-image:latest .
```

This command will build a Docker image with the name `my-image` and tag `latest` using the `Dockerfile` in the current directory (`.`).

7. Verify that the image was built successfully by running the following command:

```
podman images
```

This command should display a list of images, including the one you just built (`my-image:latest`).

### *Step 3: Running a Docker Container*

To run a Docker container from the image you just created, follow the steps below:

1. Run the following command to start a container from the image:

```
podman run -p 5000:5000 my-image:latest
```

This command starts a new container from the `my-image` image, maps port 5000 from the container to port 5000 on the host machine, and runs the command specified in the `CMD` directive of the `Dockerfile`.

2. Verify that the container is running by opening a web browser and navigating to `http://localhost:5000`. You should see the output of the `app.py` file that was copied into the image in Step 2.

3. To stop the container, open another PowerShell window and run the following command:

```
podman stop $(podman ps -q)
```

This command stops all running containers.

**C**onclusion: In this guide, I have provided a detailed guide on how to create and manage Docker images using Podman in Windows. By using Podman, developers can use Docker-compatible commands without the need for a daemon, providing a more lightweight and flexible containerization solution.