

# Docker container-docker run

---

`docker run` is a command in the Docker CLI that creates and starts a new container from an image. Here's a summary:

## Basic Syntax:

```
docker run [options] <image>[:<tag>] [<command>]
```

- `<image>` is the name of the image to use as a base for the new container.
- `[:<tag>]` is optional and specifies a specific tag (version) of the image.
- `[<command>]` is optional and specifies a command to run in the new container.

## Common Options:

- `-d`: Run the container in detached mode, i.e., in the background.
- `-p <port>`: Expose a port from the container's network stack to the host machine.
- `-v <volume>`: Mount a volume (directory) from the host machine into the container.

## Example:

Let's say we have an image called `nginx` and we want to run it with the default web server configuration. We can use the following command:

```
docker run -d --name my-nginx -p 8080:80 nginx
```

Here:

- `-d` runs the container in detached mode.
- `--name my-nginx` sets a name for the new container.
- `-p 8080:80` exposes port 8080 on the host machine and maps it to port 80 inside the container (the default web server port for Nginx).
- `nginx` is the image to use as a base.

After running this command, you should be able to access the Nginx web server by visiting `http://localhost:8080` in your web browser.

If you want to run a specific command inside the container, you can add it after the image name. For example:

```
docker run -it --rm ubuntu /bin/bash
```

This will create an Ubuntu container and start a new Bash shell inside it. The `-it` options allow for interactive terminal access, and `--rm` removes the container when it exits.

Remember to replace `<image>` with the actual name of the image you want to use, and adjust the options as needed for your specific use case!

---

*Curated by Brajesh Kumar*