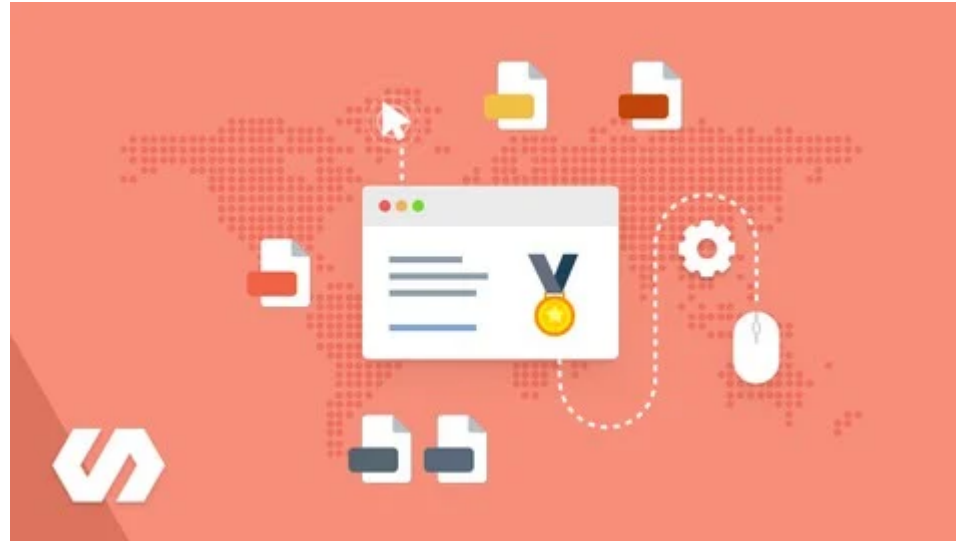


Course: Docker and Kubernetes: The Complete Guide

 [udemy.com/course/docker-and-kubernetes-the-complete-guide/learn/lecture/22776989](https://www.udemy.com/course/docker-and-kubernetes-the-complete-guide/learn/lecture/22776989)



Buildkit for Docker Desktop

Students who have the most recent versions of Docker will now have **Buildkit** enabled by default. If so, you will notice a slightly different output in your terminal when building from a Dockerfile.

The most important difference for students will be the final step in the build process. As shown in the lecture the last step would say:

1. ---> fc60771eaa08
2. Successfully Built fc60771eaa08

Now, with Buildkit, the final step would say:

1. => => exporting layers
2. 0.0s => => writing image sha256:ee59c34ada9890ca09145cc88ccb25d32b677fc3b61e921 0.0s

Both ***fc60771eaa08*** and ***ee59c34ada9890ca09145cc88ccb25d32b677fc3b61e921*** are the resulting image ID's that you would use to run a container.

eg:

1. docker run fc60771eaa08

or

1. docker run ee59c34ada9890ca09145cc88ccb25d32b677fc3b61e921

Also, the CMD instruction is now omitted from the visible output. This change is rather unimportant and does not affect the image in any way.

Using the Progress Flag to see more verbose output

Buildkit will hide away much of its progress which is something the legacy builder did not do. We will be discussing some messages and errors later in Section 4 that will be hidden by default. To see this output, you will want to pass the progress flag to the build command:

```
docker build --progress=plain .
```

Additionally, you can pass the no-cache flag to disable any caching:

```
docker build --no-cache --progress=plain .
```

Note - Do not try to use the no-cache flag with Lecture 47 Minimizing Cache Busting

Disabling Buildkit to match course output

To disable Buildkit, you can just pass the following variable to the build command:

```
DOCKER_BUILDKIT=0 docker build .
```

Buildkit Features and Documentation

If you want to learn more about features Buildkit has to offer, please check out the following pages:

https://docs.docker.com/develop/develop-images/build_enhancements/

<https://docs.docker.com/engine/reference/commandline/build/#specifying-external-cache-sources>

<https://www.docker.com/blog/advanced-dockerfiles-faster-builds-and-smaller-images-using-buildkit-and-multistage-builds/>