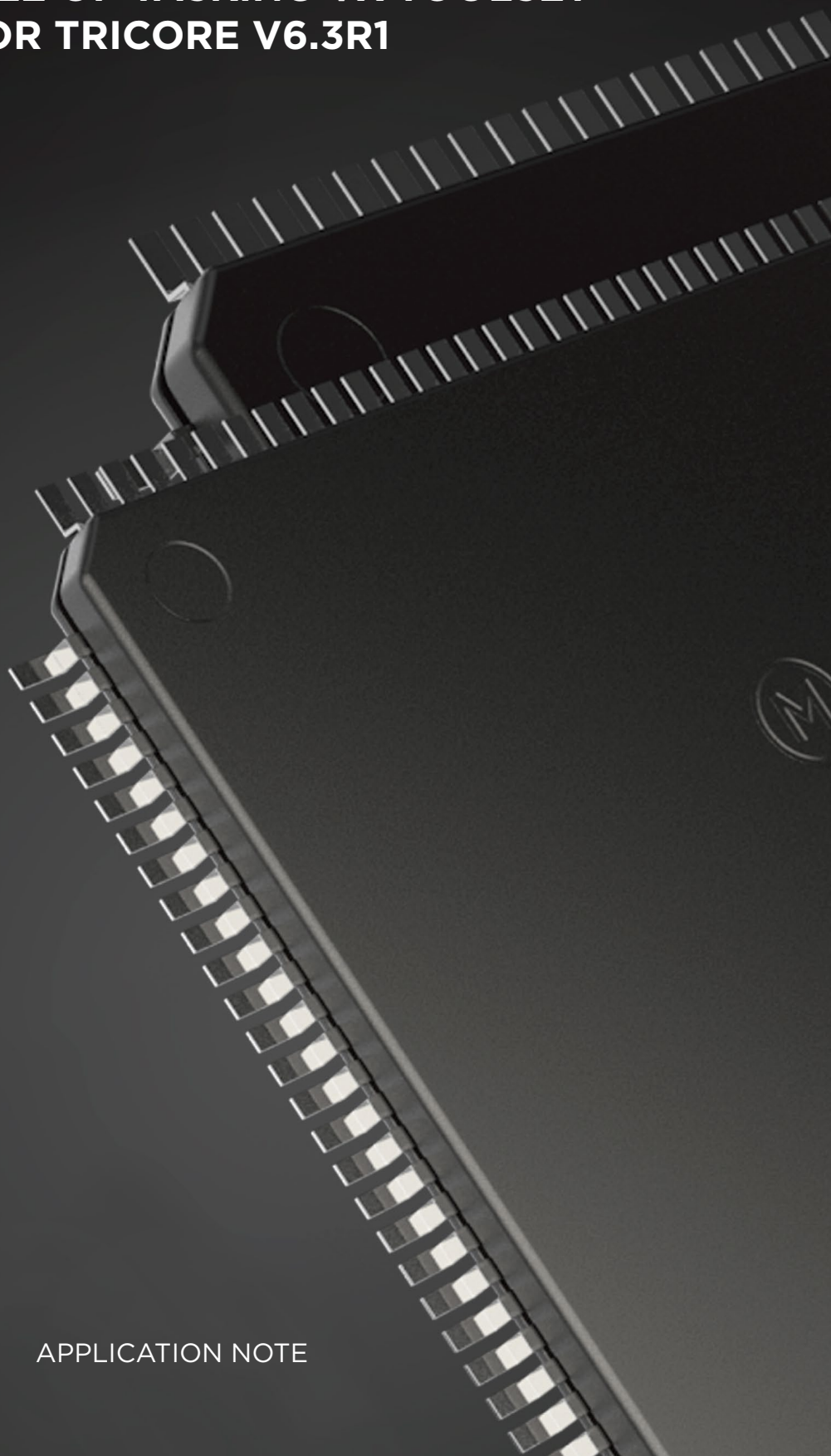


***TASKING***<sup>®</sup>

**HOW TO BUILD AND RUN  
A WINDOWS DOCKER IMAGE - USING  
THE EXAMPLE OF TASKING VX-TOOLSET  
FOR TRICORE V6.3R1**

APPLICATION NOTE



## HOW TO BUILD AND RUN A WINDOWS DOCKER IMAGE – USING THE EXAMPLE OF TASKING VX-TOOLSET FOR TRICORE V6.3R1

### INTRODUCTION

In this Application Note, we will explain how to Build and run **Windows Docker Image** for a TASKING product on Local Disk and Remote Repository. We will take the product “TASKING VX-toolset for TriCore v6.3r1” with a floating license as an example. Node-locked licenses are not recommended to be used for a Docker image.

### WHAT IS A DOCKER IMAGE?

A Docker image is a read-only template that contains a set of instructions for creating a container that can run on a Docker platform. It provides a convenient way to package up applications and preconfigured server environments, which you can use for your own private use or share publicly with other Docker users. For more information about Docker, please refer to <https://www.docker.com>.

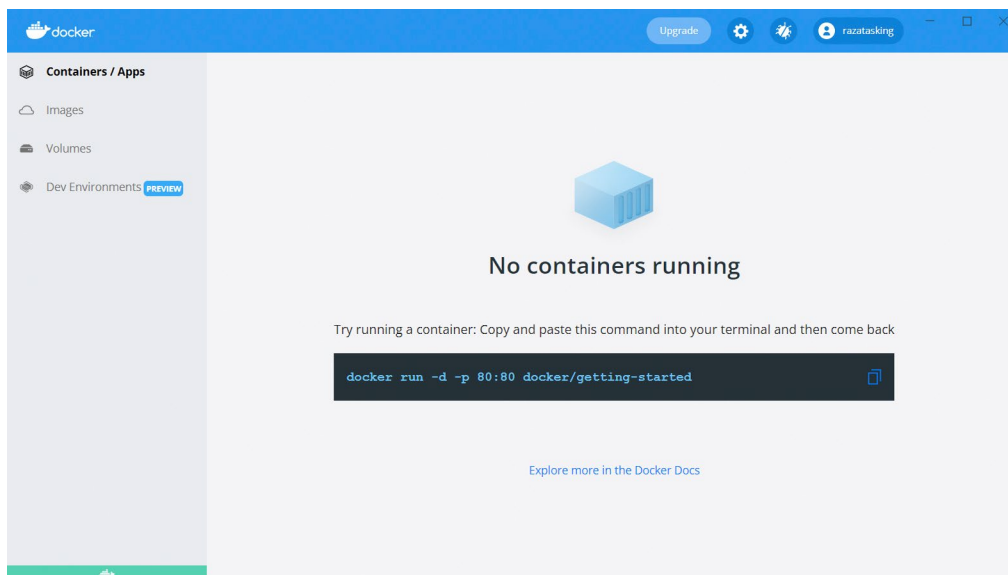
### CREATE TASKING DOCKER IMAGE ON LOCAL DISK

#### Setting Up Docker Desktop application

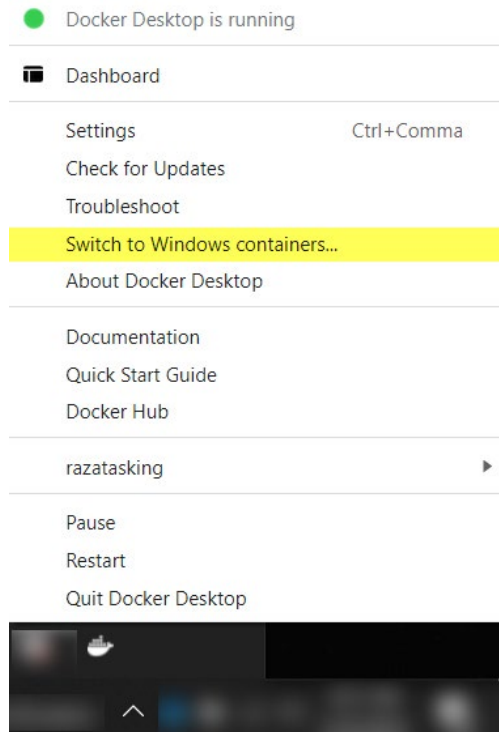
1. Download the latest Dockers Desktop App and install it from: <https://www.docker.com/products/docker-desktop>

For this Application Note Docker Desktop 4.5.1 (74721) will be used.

2. Launch the Docker Desktop app. Once Docker Engine had started successful it will look like this (Docker Icon turn to Green):



3. Make sure to “Switch to Window Containers...” via right clicking on the Docker icon present in “Show Hidden Icons” from Windows task bar:



If you are creating an image for Linux OS leave it to Linux Container (this is the default).

### Prepare Docker file

For this App Note, let's assume that TASKING VX-toolset TriCore v6.3r1 is placed at the following installation directory:

```
C:\Program Files\TASKING\TriCore v6.3r1
```

1. Create a new text file(.txt) file at the installation directory of v6.3r1.

For now, we will refer this .txt file as “**Dockerfile.txt**”.

2. Place the following text in Dockerfile:

```
# Getting base image Windows
FROM mcr.microsoft.com/windows/servercore:ltsc2019
RUN echo „Making a New Directory Called Tricore“
RUN mkdir Tricore
RUN echo „Now Adding Folders“
ADD . /Tricore
# Setting the Required Environment Variable
ENV TSK_OPTIONS_FILE_SW160800v6_3r1 „C:\Tricore\etc\licopt.txt“
# Sets a command or process that will run each time a container is
# run from the new image.
CMD [ „cmd“ ]
```

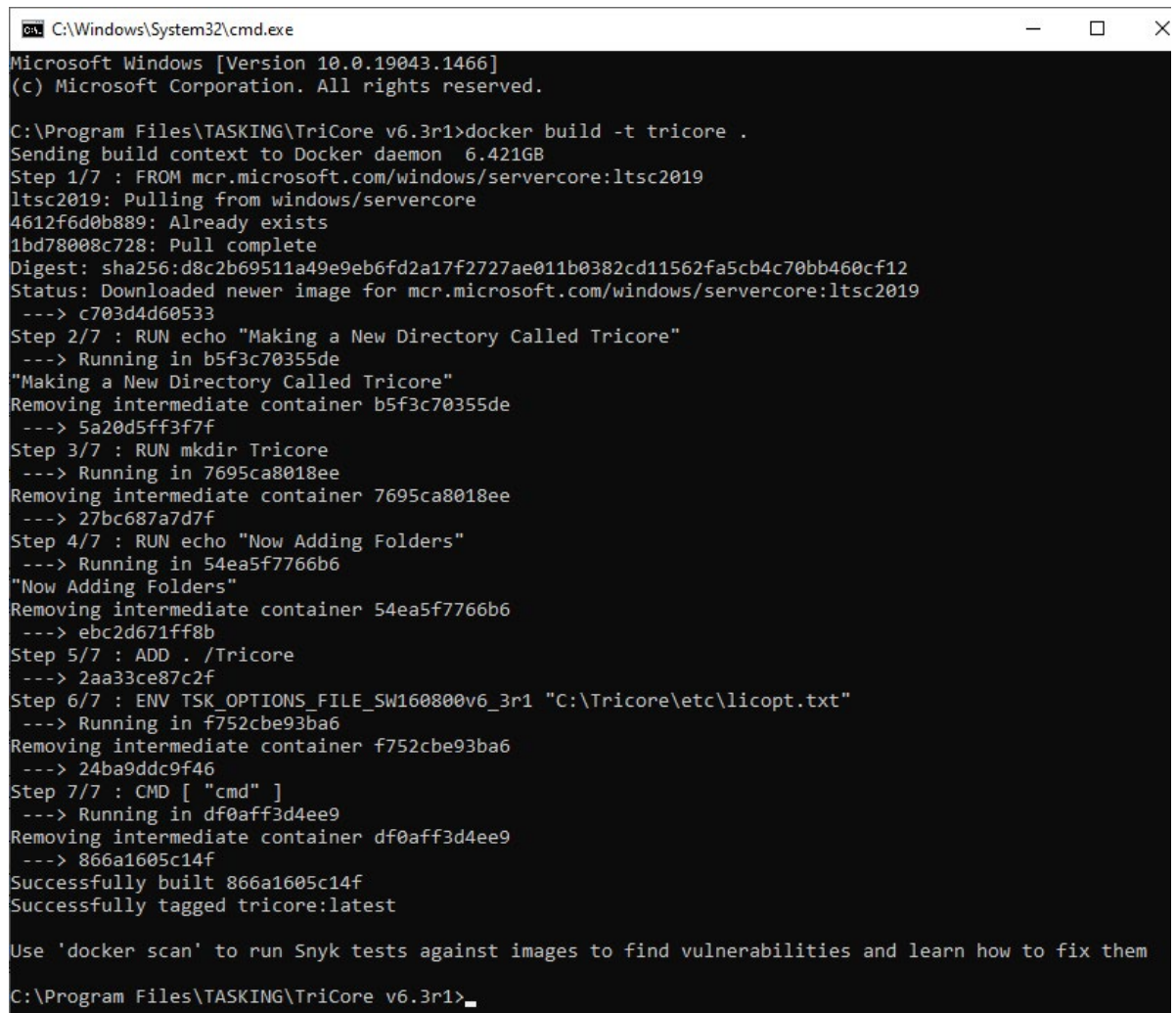
3. Save and Dockerfile.txt and close it.
4. Rename the **Dockerfile.txt** to **Dockerfile**  
(Remove the extension .txt as Docker engine do not accept the .txt format)

### Creating Docker Image Locally

1. Open the Command Prompt(cmd.exe) at Tricore Installation Directory  
and write the following command:

```
docker build -t tricore .
```

i.e.

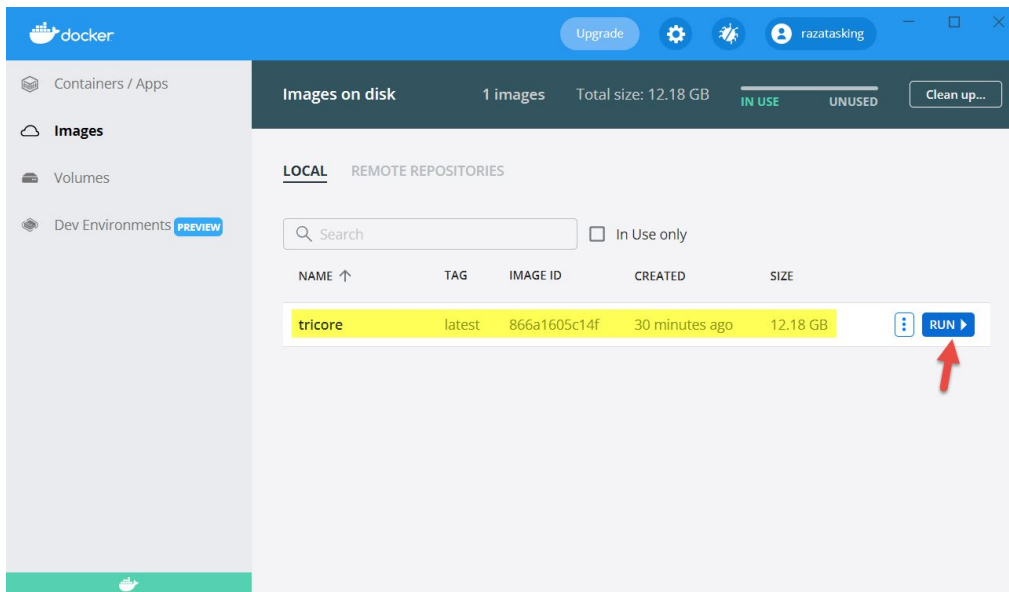


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\TASKING\TriCore v6.3r1>docker build -t tricore .
Sending build context to Docker daemon  6.421GB
Step 1/7 : FROM mcr.microsoft.com/windows/servercore:ltsc2019
ltsc2019: Pulling from windows/servercore
4612f6d0b889: Already exists
1bd78008c728: Pull complete
Digest: sha256:d8c2b69511a49e9eb6fd2a17f2727ae011b0382cd11562fa5cb4c70bb460cf12
Status: Downloaded newer image for mcr.microsoft.com/windows/servercore:ltsc2019
--> c703d4d60533
Step 2/7 : RUN echo "Making a New Directory Called Tricore"
--> Running in b5f3c70355de
"Making a New Directory Called Tricore"
Removing intermediate container b5f3c70355de
--> 5a20d5ff3f7f
Step 3/7 : RUN mkdir Tricore
--> Running in 7695ca8018ee
Removing intermediate container 7695ca8018ee
--> 27bc687a7d7f
Step 4/7 : RUN echo "Now Adding Folders"
--> Running in 54ea5f7766b6
"Now Adding Folders"
Removing intermediate container 54ea5f7766b6
--> ebc2d671ff8b
Step 5/7 : ADD . /Tricore
--> 2aa33ce87c2f
Step 6/7 : ENV TSK_OPTIONS_FILE_SW160800v6_3r1 "C:\Tricore\etc\licopt.txt"
--> Running in f752cbe93ba6
Removing intermediate container f752cbe93ba6
--> 24ba9ddc9f46
Step 7/7 : CMD [ "cmd" ]
--> Running in df0aff3d4ee9
Removing intermediate container df0aff3d4ee9
--> 866a1605c14f
Successfully built 866a1605c14f
Successfully tagged tricore:latest

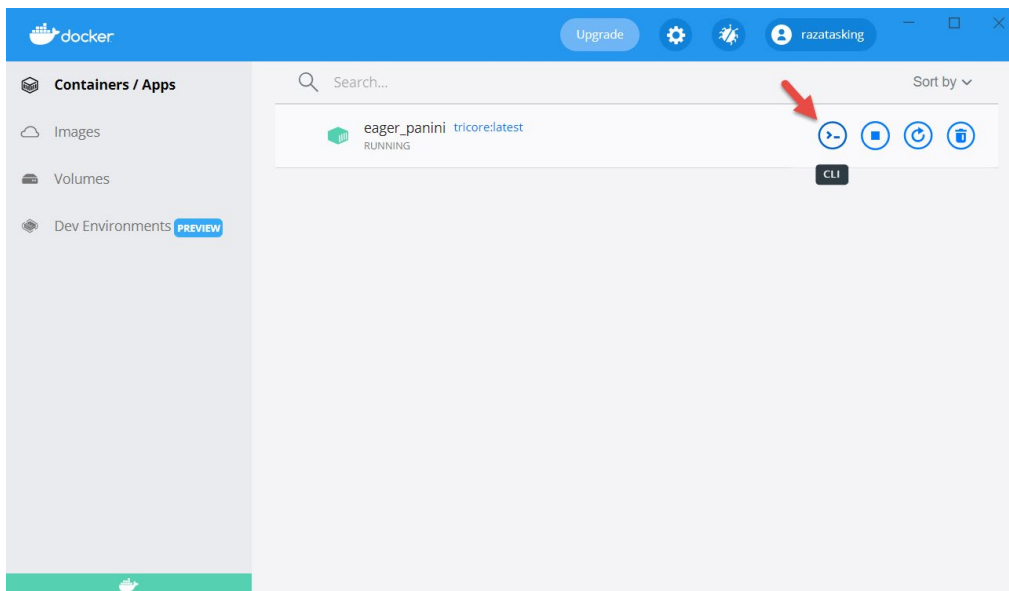
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Program Files\TASKING\TriCore v6.3r1>_
```

2. This will in return create an image named as “tricore” with the tag “latest” (default) like:



## Running the Docker Container

1. Click on the “Run” button (as indicated above) in-order to run the image.
2. Once the Docker image is running, click on the Command Line Interface (CLI)



3. Once the command line interface is open, change the working directory to `\Tricore\ctc\bin` and run the `"ctc -v"` command as indicated below:

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.2565]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\>cd Tricore\ctc\bin

C:\Tricore\ctc\bin>ctc -V
Get remote license info for ██████████
Connect to lic1.tasking.com:8080
Set local license info for ██████████
TASKING VX-toolset for TriCore: C compiler v6.3r1 Build 21092262
Copyright 2002-2021 TASKING BV

C:\Tricore\ctc\bin>
```

**Remark:** During the preparation of Dockerfile, we had introduced a layer to create a new folder named "Tricore" in the Docker container (RUN `mkdir Tricore`). After that we have copied (ADD `./Tricore`) the full content of the root installation folder (`C:\Program Files\TASKING\TriCore v6.3r1`) to this newly created folder **Tricore (C:\Tricore)**.

4. This will show the current version number of Tricore being used in the Docker image which proves that the Tricore image has been created successfully

## CREATE A TASKING DOCKER IMAGE FOR DOCKER HUB REPOS

### Building an Image for Docker Repositories

1. In order to push an image to docker repository on Docker Hub, first sign up for the [Docker Hub](#) community and login at Docker Desktop App with your respective credentials. Create a repository in Docker Hub. In this App Note we are going to refer username as **taskingdocker** and repository as **tricore\_2022**. Once the repository has been created, go to the local installation directory of TASKING VX-toolset TriCore v6.3r1. For this App Note we consider that as:

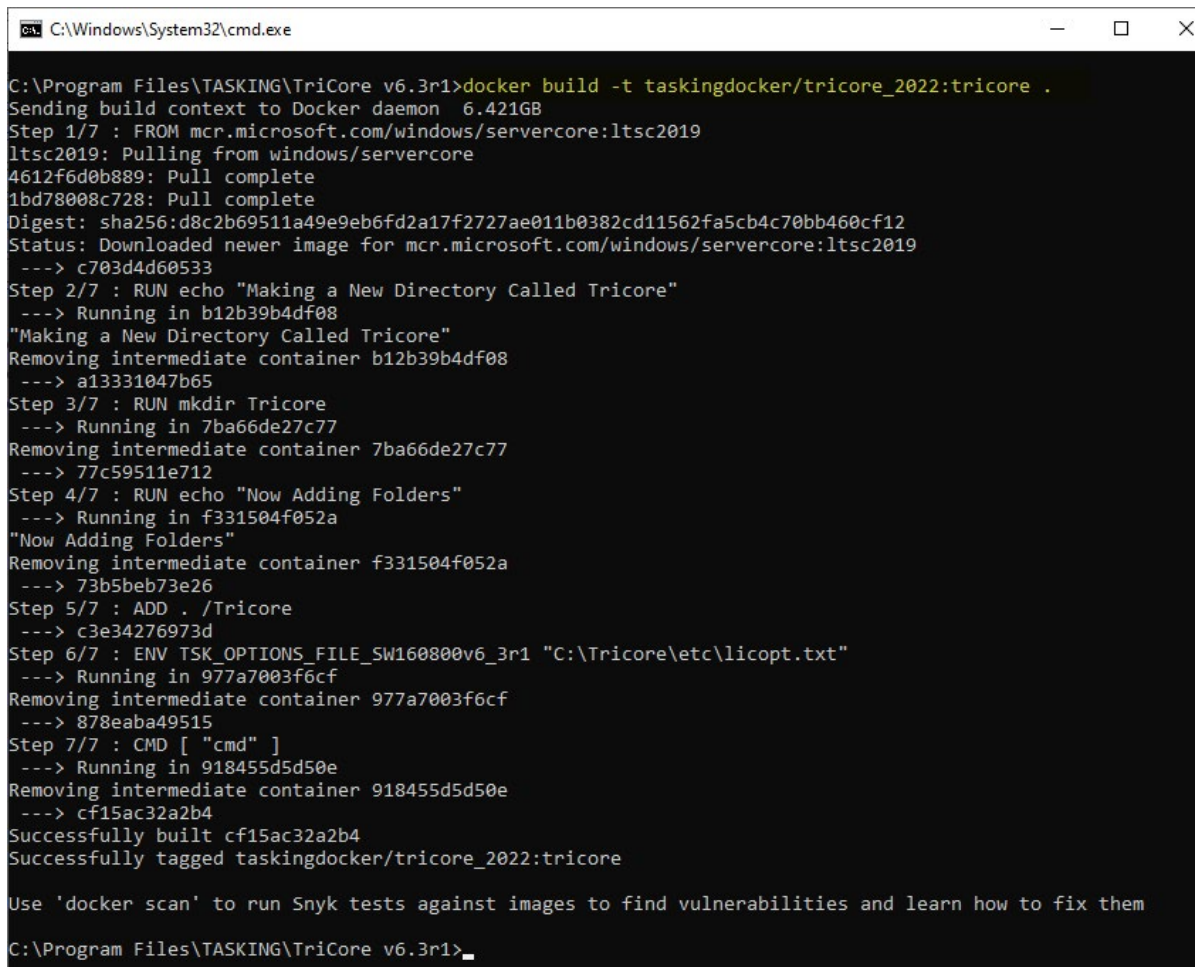
```
C:\Program Files\TASKING\TriCore v6.3r1
```

2. Add the Docker file as mentioned in previous Chapter (please refer to Prepare Docker file).  
Open the command prompt(cmd.exe) here and use the following command to build an image for repository.

```
docker build -t [username]/[repository_name]:image_tag .
```

like:

```
docker build -t taskingdocker/tricore_2022:tricore .
```



```
C:\Windows\System32\cmd.exe

C:\Program Files\TASKING\TriCore v6.3r1>docker build -t taskingdocker/tricore_2022:tricore .
Sending build context to Docker daemon 6.421GB
Step 1/7 : FROM mcr.microsoft.com/windows/servercore:ltsc2019
ltsc2019: Pulling from windows/servercore
4612f6d0b889: Pull complete
1bd78008c728: Pull complete
Digest: sha256:d8c2b69511a49e9eb6fd2a17f2727ae011b0382cd11562fa5cb4c70bb460cf12
Status: Downloaded newer image for mcr.microsoft.com/windows/servercore:ltsc2019
--> c703d4d60533
Step 2/7 : RUN echo "Making a New Directory Called Tricore"
--> Running in b12b39b4df08
"Making a New Directory Called Tricore"
Removing intermediate container b12b39b4df08
--> a13331047b65
Step 3/7 : RUN mkdir Tricore
--> Running in 7ba66de27c77
Removing intermediate container 7ba66de27c77
--> 77c59511e712
Step 4/7 : RUN echo "Now Adding Folders"
--> Running in f331504f052a
"Now Adding Folders"
Removing intermediate container f331504f052a
--> 73b5beb73e26
Step 5/7 : ADD . /Tricore
--> c3e34276973d
Step 6/7 : ENV TSK_OPTIONS_FILE_SW160800v6_3r1 "C:\Tricore\etc\licopt.txt"
--> Running in 977a7003f6cf
Removing intermediate container 977a7003f6cf
--> 878eaba49515
Step 7/7 : CMD [ "cmd" ]
--> Running in 918455d5d50e
Removing intermediate container 918455d5d50e
--> cf15ac32a2b4
Successfully built cf15ac32a2b4
Successfully tagged taskingdocker/tricore_2022:tricore

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\Program Files\TASKING\TriCore v6.3r1>_
```

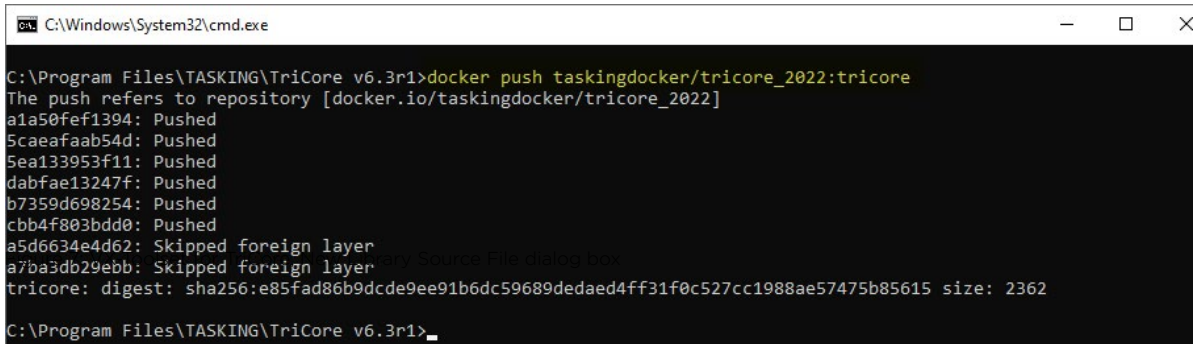
### Pushing an Image to Docker Repositories

1. Once the image had been successfully built for the repository, it can be pushed to repository with the following command

```
docker push [username]/[repository_name]:tagname
```

like:

```
docker push taskingdocker/tricore_2022:tricore
```



```
C:\Windows\System32\cmd.exe

C:\Program Files\TASKING\TriCore v6.3r1>docker push taskingdocker/tricore_2022:tricore
The push refers to repository [docker.io/taskingdocker/tricore_2022]
a1a50fef1394: Pushed
5caefaab54d: Pushed
5ea133953f11: Pushed
dabfae13247f: Pushed
b7359d698254: Pushed
cbb4f803bdd0: Pushed
a5d6634e4d62: Skipped foreign layer
a7ba3db29ebb: Skipped foreign layer
tricore: digest: sha256:e85fad86b9dcde9ee91b6dc59689dedaed4ff31f0c527cc1988ae57475b85615 size: 2362

C:\Program Files\TASKING\TriCore v6.3r1>_
```

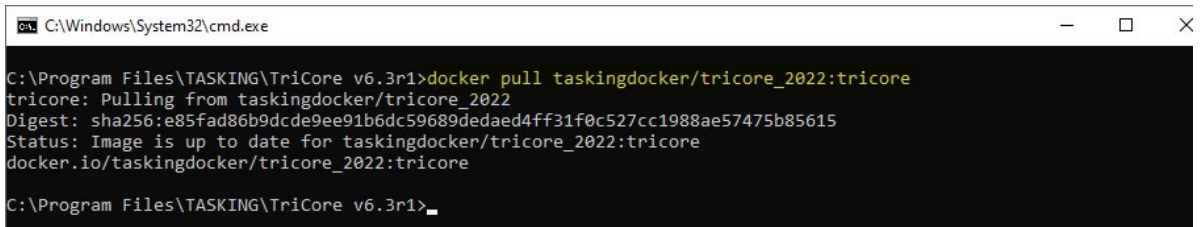
### Pulling an Image from Docker Repositories

1. Once the image had been successfully pushed in the repository, it can be pulled from the repository with the following command

```
docker pull [username]/[repository_name]:tagname
```

like:

```
docker pull taskingdocker/tricore_2022:tricore
```



```
C:\Windows\System32\cmd.exe

C:\Program Files\TASKING\TriCore v6.3r1>docker pull taskingdocker/tricore_2022:tricore
tricore: Pulling from taskingdocker/tricore_2022
Digest: sha256:e85fad86b9dcde9ee91b6dc59689dedaed4ff31f0c527cc1988ae57475b85615
Status: Image is up to date for taskingdocker/tricore_2022:tricore
docker.io/taskingdocker/tricore_2022:tricore

C:\Program Files\TASKING\TriCore v6.3r1>_
```



The information about successful pushing and pulling can be verified in the Docker Desktop App and also in the Docker Hub repository. i.e.

