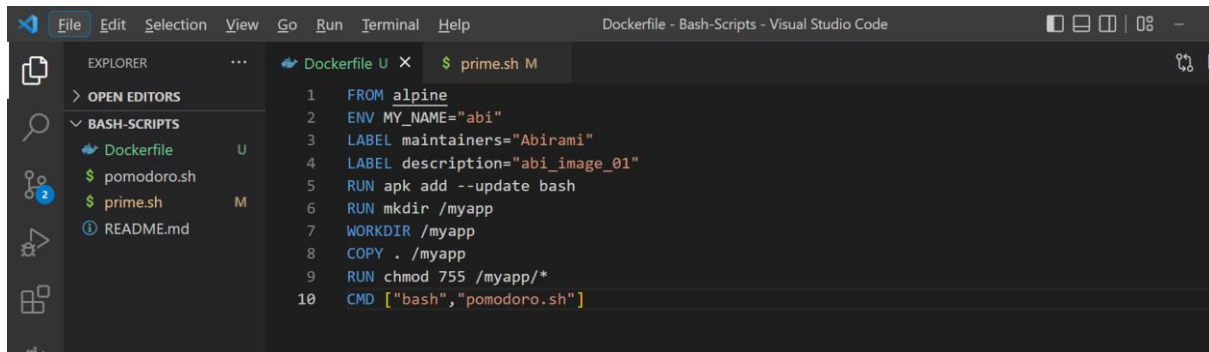


Ques 1: Create a container for the below bash script(**pomodoro.sh**).

1. Create a Dockerfile
2. Create a Docker Image
3. Push the image to the docker hub
4. Pull it and test it.

Find the script from here <https://github.com/DARK-art108/Bash-Scripts>

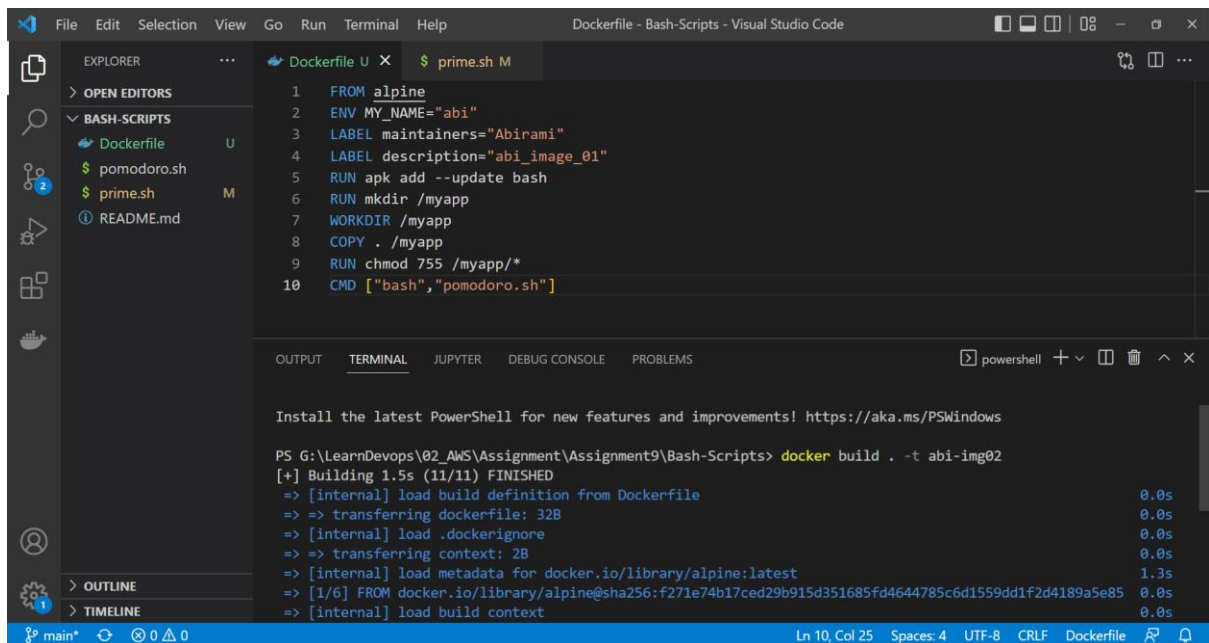
## Create a Dockerfile



The screenshot shows the Visual Studio Code interface with a Dockerfile being created. The Explorer panel on the left shows a project named 'BASH-SCRIPTS' with files 'Dockerfile', 'pomodoro.sh', 'prime.sh', and 'README.md'. The Dockerfile editor shows the following content:

```
1 FROM alpine
2 ENV MY_NAME="abi"
3 LABEL maintainers="Abirami"
4 LABEL description="abi_image_01"
5 RUN apk add --update bash
6 RUN mkdir /myapp
7 WORKDIR /myapp
8 COPY . /myapp
9 RUN chmod 755 /myapp/*
10 CMD ["bash", "pomodoro.sh"]
```

## Create a Docker Image



The screenshot shows the Visual Studio Code interface with the Dockerfile editor and the Terminal panel. The Dockerfile content is the same as in the previous screenshot. The Terminal panel shows the command `docker build . -t abi-img02` being executed, and the output of the build process:

```
PS G:\LearnDevops\02_AWS\Assignment\Assignment9\Bash-Scripts> docker build . -t abi-img02
[+] Building 1.5s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 32B                                              0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/alpine:latest                1.3s
=> [1/6] FROM docker.io/library/alpine@sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e85 0.0s
=> [internal] load build context                                                0.0s
```

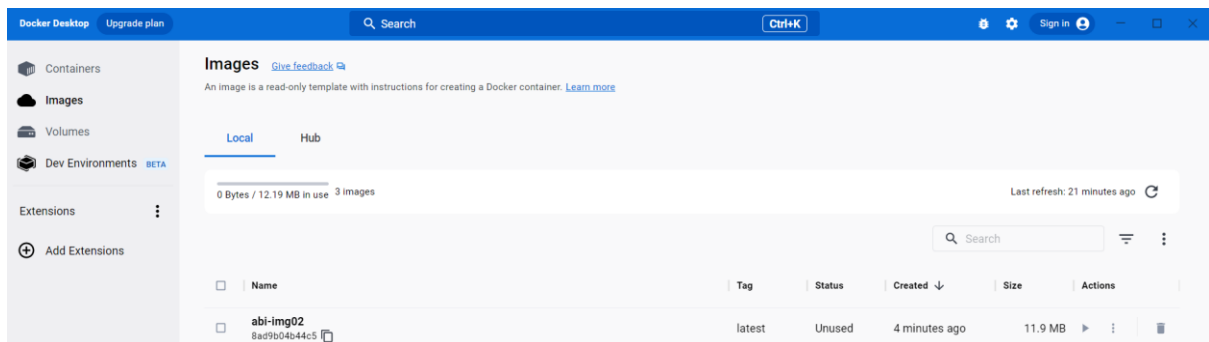
The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following instructions:

```
1 FROM alpine
2 ENV MY_NAME="abi"
3 LABEL maintainers="Abirami"
4 LABEL description="abi_image_01"
5 RUN apk add --update bash
6 RUN mkdir /myapp
7 WORKDIR /myapp
8 COPY . /myapp
9 RUN chmod 755 /myapp/*
10 CMD ["bash", "pomodoro.sh"]
```

The terminal output shows the build process:

```
=> CACHED [2/6] RUN apk add --update bash 0.0s
=> CACHED [3/6] RUN mkdir /myapp 0.0s
=> CACHED [4/6] WORKDIR /myapp 0.0s
=> CACHED [5/6] COPY . /myapp 0.0s
=> CACHED [6/6] RUN chmod 755 /myapp/* 0.0s
=> exporting to image 0.0s
=> => writing image sha256:8ad9b04b44c56ea50b0b96a56658404dc3a6da687c673b27b15390b04f80fe46 0.0s
=> naming to docker.io/library/abi-img02 0.0s
```

At the bottom, a message suggests using 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them.



Push the image to the docker hub

The screenshot shows the Visual Studio Code interface with the terminal open. The following commands are executed:

```
PS G:\LearnDevops\02_AWS\Assignment\Assignment9\Bash-Scripts> docker tag abi-img02 abiramikarthick/abi-img02
PS G:\LearnDevops\02_AWS\Assignment\Assignment9\Bash-Scripts> docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
abiramikarthick/abi-img02  latest     8ad9b04b44c5  34 minutes ago  11.9MB
abi-img02           latest     8ad9b04b44c5  34 minutes ago  11.9MB
abi-img             latest     2907f7131bce  53 minutes ago  11.9MB
```

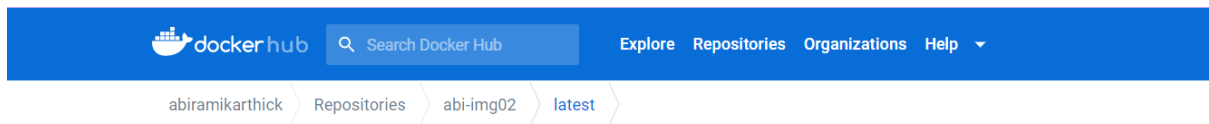
Then, the image is pushed to Docker Hub:


```
PS G:\LearnDevops\02_AWS\Assignment\Assignment9\Bash-Scripts> docker push abiramikarthick/abi-img02
Using default tag: latest
The push refers to repository [docker.io/abiramikarthick/abi-img02]
6653992f2ec1: Pushed
ec52f9b223cb: Pushed
```

The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following instructions:

```
1 FROM alpine
2 ENV MY_NAME="abi"
3 LABEL maintainers="Abirami"
4 LABEL description="abi_image_01"
5 RUN apk add --update bash
6 RUN mkdir /myapp
7 WORKDIR /myapp
8 COPY . /myapp
9 RUN chmod 755 /myapp/*
10 CMD ["bash", "pomodoro.sh"]
```

The terminal output shows the command `docker push abiramikarthick/abi-img02` being executed, resulting in a successful push of the latest image. The output includes the image ID, digest, and size.



			
<b>abiramikarthick/abi-img02:latest</b>			
DIGEST: sha256:a16fc593d548203f4766f19990456b79180d43b79864b2f89de1d16d8687ce32			
OS/ARCH	COMPRESSED SIZE ⓘ	LAST PUSHED	TYPE
linux/amd64	6.51 MB	3 minutes ago by <a href="#">abiramikarthick</a>	Image

Pull it and test it.

The screenshot shows the Visual Studio Code interface with the same Dockerfile as before. The terminal output shows the command `docker pull abiramikarthick/abi-img02` being executed, resulting in the image being pulled successfully. The output includes the image ID, digest, and size.