Here's a step-by-step tutorial on creating a Python "Hello World" project and pushing it to the Docker Hub repository:

Step 1: Set up the project directory

1. Create a new directory for your project. You can name it anything you like.
2. Inside the project directory, create a new file called **app.py**. This file will contain the Python code for your "Hello World" program.

Step 2: Write the Python code

1. Open **app.py** in a text editor or an integrated development environment (IDE).
2. Write the following code in **app.py**:

```
print("Hello, World!")
```

3. Save the file.

Step 3: Create a Dockerfile

1. In the project directory, create a new file called **Dockerfile** (without any file extension).
2. Open **Dockerfile** in a text editor or IDE.
3. Add the following content to the **Dockerfile**:

```
# Use the official Python image from the Docker Hub
FROM python:3

# Copy the application code into the container
COPY app.py /app.py

# Set the default command to run the application
CMD [ "python", "/app.py" ]
```

4. Save the file.

Step 4: Build the Docker image

1. Open a terminal or command prompt.

2. Navigate to the project directory using the **cd** command.
3. Run the following command to build the Docker image:

```
docker build -t your-username/hello-world .
```

Replace **your-username** with your Docker Hub username. Don't forget the period (**.**) at the end of the command, which indicates the current directory.

4. Docker will now build the image based on the instructions in the Dockerfile. It may take a few moments.

Step 5: Push the Docker image to Docker Hub

1. Login to Docker Hub using the following command:

```
docker login
```

2. Enter your Docker Hub username and password when prompted.
3. Push the image to Docker Hub using the following command:

```
docker push your-username/hello-world
```

Replace **your-username** with your Docker Hub username.

4. Docker will upload the image to your Docker Hub repository. The process may take some time, depending on the image size and your internet connection.

Step 6: Verify the pushed image

1. Visit the Docker Hub website (https://hub.docker.com) and login to your account.
2. Navigate to your repository. You should see the **hello-world** image listed there.

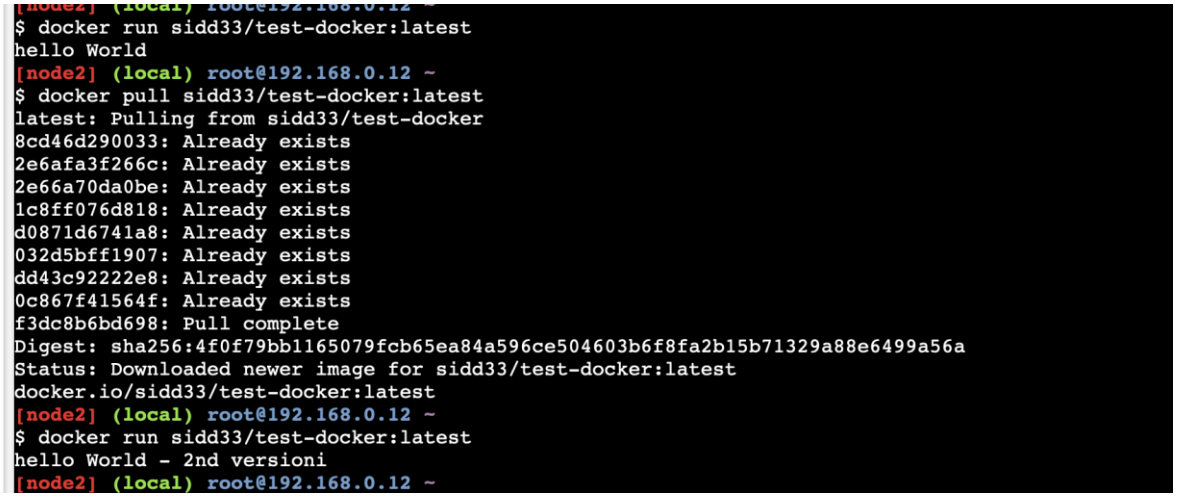
Congratulations! You have successfully created a Python "Hello World" project, built a Docker image, and pushed it to the Docker Hub repository. Now, you can pull this image on any machine and run it using Docker.

RE Do the change to App.py and reverify via push and then pull to another node

```
[node1] (local) root@192.168.0.13 ~
$
[node1] (local) root@192.168.0.13 ~
$ cat app.py
print("hello World - 2nd versioni")
[node1] (local) root@192.168.0.13 ~
$
```

```
latest: digest: sha256:d26edb4526220aaddd0d3e70410f0f9c9e2d1c423dea4febbeec38ddf2dd30f8 size: 2214
[node1] (local) root@192.168.0.13 ~
$ docker images
REPOSITORY           TAG       IMAGE ID       CREATED         SIZE
sidd33/test-docker   latest    04e99e227959   4 minutes ago   1.01GB
[node1] (local) root@192.168.0.13 ~
$ ls
app.py      dockerfile
[node1] (local) root@192.168.0.13 ~
$ cat app.py
print("hello World")
[node1] (local) root@192.168.0.13 ~
$ vi app.py
[node1] (local) root@192.168.0.13 ~
$ docker build -t sidd33/test-docker .
[+] Building 0.4s (8/8) FINISHED                                                                   docker:default
 => [internal] load build definition from dockerfile                                                          0.0s
 => => transferring dockerfile: 268B                                                                          0.0s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [internal] load metadata for docker.io/library/python:3                                                   0.2s
 => [auth] library/python:pull token for registry-1.docker.io                                                 0.0s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 69B                                                                              0.0s
 => CACHED [1/2] FROM docker.io/library/python:3@sha256:11aa4b620c15f855f66f02a7f3c1cd9cf843cc10f3edbcf158e5ebcd98d1f549  0.0s
 => [2/2] COPY app.py /app.py                                                                                 0.1s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:7565aa8e58b915af0cc4819f1bcf38efc16548e85105103770d3dd83ef9e7fc9                  0.0s
 => => naming to docker.io/sidd33/test-docker                                                                 0.0s
[node1] (local) root@192.168.0.13 ~
$ docker push sidd33/test-docker
Using default tag: latest
The push refers to repository [docker.io/sidd33/test-docker]
e313ea579ac5: Pushed
bc20776edade: Layer already exists
fe44cca5f6d4: Layer already exists
5b6a3f461af3: Layer already exists
d96d277eb872: Layer already exists
3a8081ce85fa: Layer already exists
045d8b74bf0d: Layer already exists
25879f85bbb0: Layer already exists
6abe10f2f601: Layer already exists
latest: digest: sha256:4f0f79bb1165079fcb65ea84a596ce504603b6f8fa2b15b71329a88e6499a56a size: 2214
[node1] (local) root@192.168.0.13 ~
$
```

FROM THE OTHER MACHINE(NODE2)


$ docker run sidd33/test-docker:latest
hello World
[node2] (local) root@192.168.0.12 ~
$ docker pull sidd33/test-docker:latest
latest: Pulling from sidd33/test-docker
8cd46d290033: Already exists
2e6afa3f266c: Already exists
2e66a70da0be: Already exists
1c8ff076d818: Already exists
d0871d6741a8: Already exists
032d5bff1907: Already exists
dd43c92222e8: Already exists
0c867f41564f: Already exists
f3dc8b6bd698: Pull complete
Digest: sha256:4f0f79bb1165079fcb65ea84a596ce504603b6f8fa2b15b71329a88e6499a56a
Status: Downloaded newer image for sidd33/test-docker:latest
docker.io/sidd33/test-docker:latest
[node2] (local) root@192.168.0.12 ~
$ docker run sidd33/test-docker:latest
hello World - 2nd versioni
[node2] (local) root@192.168.0.12 ~