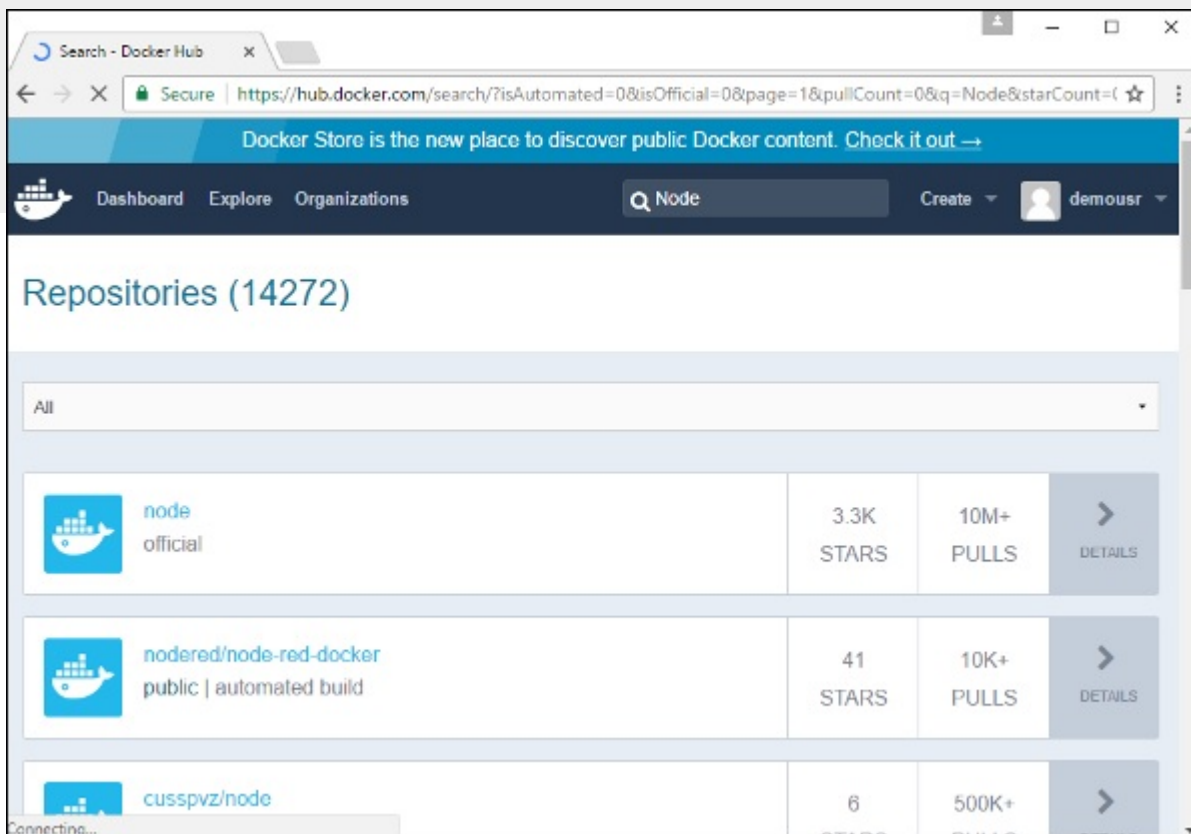


Docker - Setting Node.js

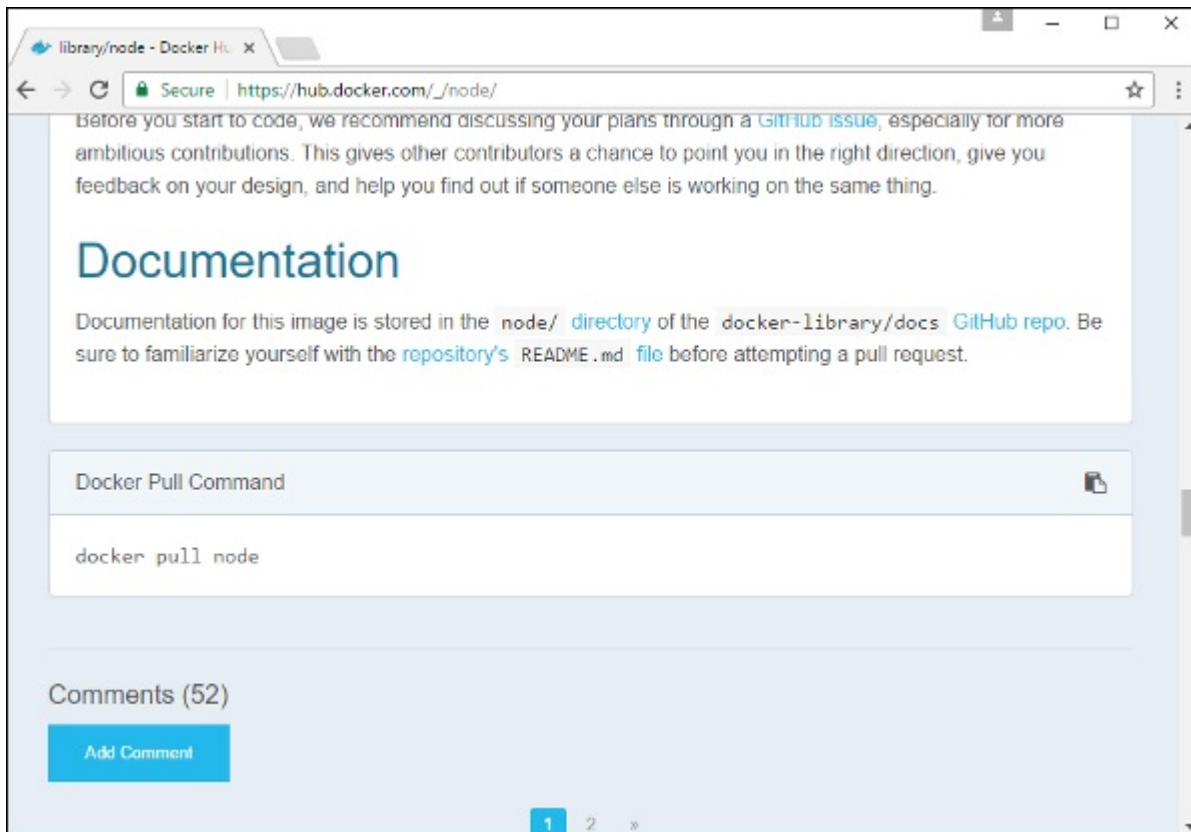
Node.js is a JavaScript framework that is used for developing server-side applications. It is an open source framework that is developed to run on a variety of operating systems. Since Node.js is a popular framework for development, Docker has also ensured it has support for Node.js applications.

We will now see the various steps for getting the Docker container for Node.js up and running.

Step 1 – The first step is to pull the image from Docker Hub. When you log into Docker Hub, you will be able to search and see the image for Node.js as shown below. Just type in Node in the search box and click on the node (official) link which comes up in the search results.



Step 2 – You will see that the Docker **pull** command for node in the details of the repository in Docker Hub.



Step 3 – On the Docker Host, use the Docker **pull** command as shown above to download the latest node image from Docker Hub.

```
demo@buntudemo:~$ sudo docker pull node_
```

Once the **pull** is complete, we can then proceed with the next step.

```
demo@buntudemo:~$ sudo docker pull node
Using default tag: latest
latest: Pulling from library/node
75a822cd7888: Downloading 31.54 MB/39.73 MB
75a822cd7888: Pull complete
57de64c72267: Pull complete
4306be1e8943: Pull complete
871436ab7225: Pull complete
0110c26a367a: Pull complete
1f04fe713f1b: Pull complete
723bac39028e: Pull complete
Digest: sha256:08d77f1984cf79739ba7c987636cb871fd69745754200e5891a0c7ee2d9965b0
Status: Downloaded newer image for node:latest
demo@buntudemo:~$
demo@buntudemo:~$
```

Step 4 – On the Docker Host, let's use the **vim** editor and create one Node.js example file. In this file, we will add a simple command to display "HelloWorld" to the command prompt.

```
demo@ubuntudemo:~$ vim HelloWorld.js
```

In the Node.js file, let's add the following statement –

```
console.log('Hello World');
```

This will output the “Hello World” phrase when we run it through Node.js.

```
console.log('Hello World');_
```

Ensure that you save the file and then proceed to the next step.

Step 5 – To run our Node.js script using the Node Docker container, we need to execute the following statement –

```
sudo docker run -it -rm -name = HelloWorld -v "$PWD":/usr/src/app  
-w /usr/src/app node node HelloWorld.js
```

The following points need to be noted about the above command –

- The **-rm** option is used to remove the container after it is run.
- We are giving a name to the container called “HelloWorld”.
- We are mentioning to map the volume in the container which is **/usr/src/app** to our current present working directory. This is done so that the node container will pick up our HelloWorld.js script which is present in our working directory on the Docker Host.
- The **-w** option is used to specify the working directory used by Node.js.
- The first node option is used to specify to run the node image.
- The second node option is used to mention to run the node command in the node container.
- And finally we mention the name of our script.

We will then get the following output. And from the output, we can clearly see that the Node container ran as a container and executed the HelloWorld.js script.

```
demo@ubuntudemo:~$ sudo docker run -it --rm --name=HelloWorld -v "$PWD":/usr/s  
/app -w /usr/src/app node node HelloWorld.js  
Hello World  
demo@ubuntudemo:~$
```