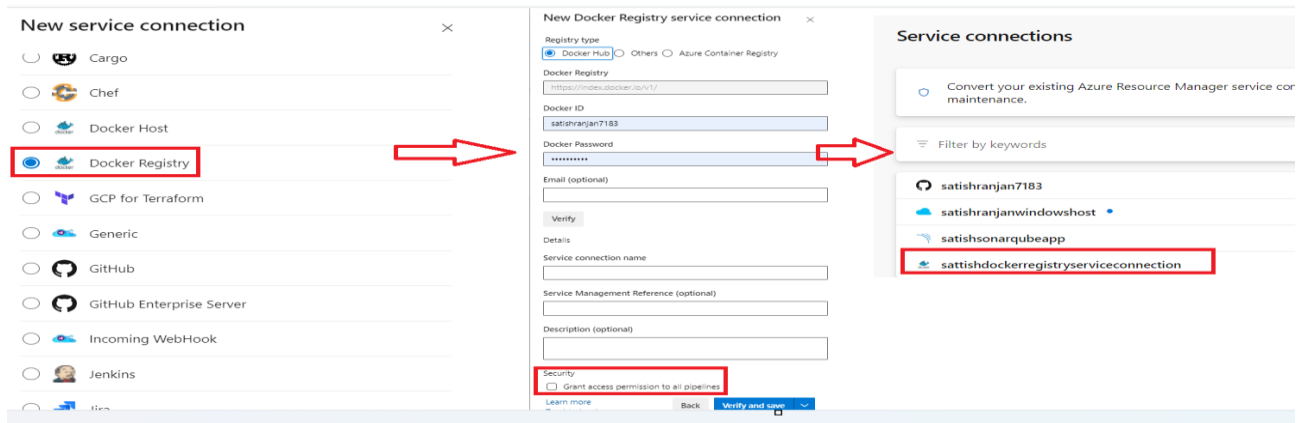


## Docker Pipeline

### Prerequisite :

1. Need to create one Self-hosted agent which have docker installed.
2. One "Docker Registry" Service Connection need to create for docker hub where our image can pull or push



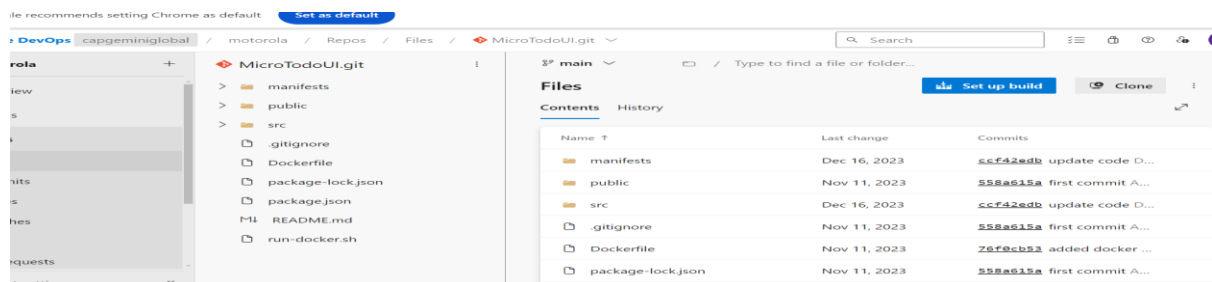
### Step 1: To create docker Service connection

Login to dev.azure.com, then go to service connection -> New Service Connection -> select docker registry -> Next

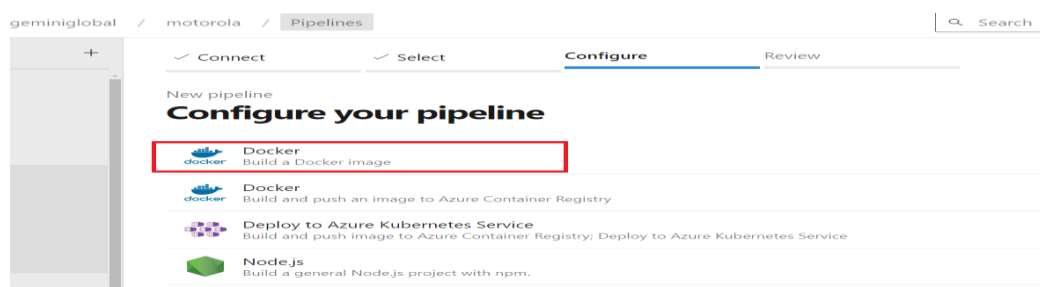
In Registry Type -> Select docker hub -> Put docker id and password -> and then click to verify to test.

Put any service connection name and click on to grant permission of all pipeline.

### Step2: Now go to repo and import docker source file in our azure repo



### Step3: Now need to create one pipeline for building image and push this image to docker hub.



#### Step4: Code for pipeline.

```
1 # Docker
2 # Build a Docker image
3 # https://docs.microsoft.com/azure/devops/pipelines/languages/docker
4
5 trigger:
6 - main
7
8 resources:
9 - repo: self
10
11 variables:
12 - tag: '$(Build.BuildId)'
13
14 stages:
15 - stage: Build
16   displayName: Build image
17   jobs:
18   - job: Build
19     displayName: Build
20     pool: satishpool
21     steps:
22     - task: Docker@2
23       inputs:
24         containerRegistry: 'sattishdockerregistryserviceconnection'
25         repository: 'satishranjan7183/microtodoui'
26         command: 'buildAndPush'
27         dockerfile: '**/Dockerfile'
```

#### < Docker

Container Repository

Container registry

Container repository

Commands

Command \*

Dockerfile \*

Build context

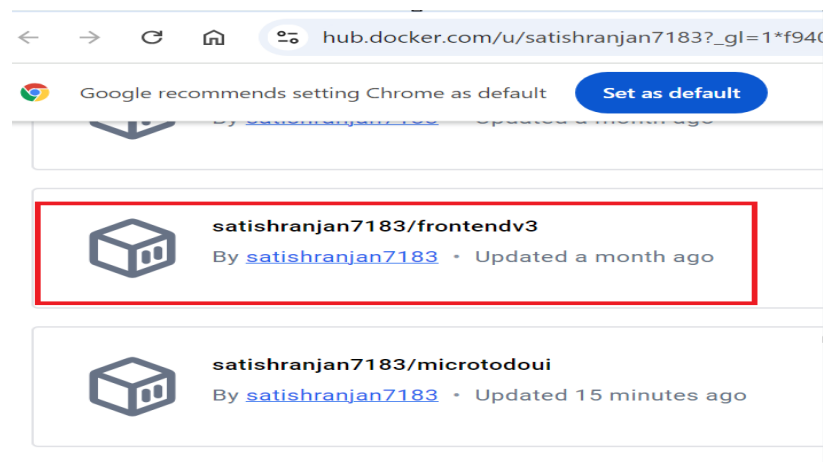
Tags

☒ Add Pipeline metadata to image(s)

☒ Add base image metadata to image(s)

About this task [Add](#)

#### Step5: Now we can run pipeline and then we can check our image present on our docker hub registry.



#### Step6: Now we build one container and expose it.

we can run this command from self hosted docker agent. Our container will expose on port no 8000.

Docker images

Docker run -d -p 8000:80 <imageid>

We can check 192.168.1.38:9000