**Machine Details  
172.17.5.39  
P@ssw0rd  
  
  
Link: http://172.17.5.39:32771/job/pipeline1/configure**

**sudo docker run -dit --restart=always -P -v jenkins\_home:/var/jenkins\_home\_talha -v /var/run/docker.sock:/var/run/docker.sock jenkins-lab-talha**

**--port**32771  
  
  
--to check port  
check container   
sudo docker ps  
  
--set different volume  
**:/var/jenkins\_home\_talha**

**--github setup  
  
--clone the git repo of nodejs on VM**

**--setup git repo on VM  
-- git push to your local repo  
-- for password generate new token through developer settings  
  
--saved token**ghp\_ps27IAH1cv8eLysFTCvL0NNSvM4wRE2tXZme  
  
  
Tasks:

1. Install docker (used command in jenkins video)

2. Install node (used nvm)

3. Install net-tools (already installed)

4. Install git (already installed)

5. Create user

-----

SSH

-----

ssh appadmin@172.17.5.38

P@ssw0rd

-----

INSTALL DOCKER

-----

RUN apt update && curl -fsSL https://get.docker.com | sh

-----

ADD USER

-----

sudo useradd -s /bin/bash -d /home/yousha/ -m -G sudo yousha

sudo passwd yousha

-----

INSTALL NODE

-----

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash

export NVM\_DIR="$HOME/.nvm" && [ -s "$NVM\_DIR/nvm.sh" ] && \. "$NVM\_DIR/nvm.sh" && [ -s "$NVM\_DIR/bash\_completion" ] && \. "$NVM\_DIR/bash\_completion"

or

source ~/.bashrc

nvm install 18

nvm use 18

-----

INSTALL DOCKER

-----

sudo usermod -aG docker $USER

newgrp docker (no logout)

-----

PASSWORDLESS LOGIN

---------

[PASTE your .pub key here]

nano /home/$USER/.ssh/authorized\_keys

(If .ssh folder does not exist)

mkdir .ssh

nano /home/$USER/.ssh/authorized\_keys

paste the public key here, so next time you can login without password

or (known\_hosts - not working for me)

-------

-----

#Container

docker run -dit --restart=always -p 8080:8080 -p 50000:50000 -v jenkins\_home:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkins-lab

#Dockerfile

FROM jenkins/jenkins:lts

USER root

RUN mkdir -p /tmp/download && \

curl -L https://download.docker.com/linux/static/stable/x86\_64/docker-18.03.1-ce.tgz | tar -xz -C /tmp/download && \

rm -rf /tmp/download/docker/dockerd && \

mv /tmp/download/docker/docker\* /usr/local/bin/ && \

rm -rf /tmp/download && \

groupadd -g 999 docker && \

usermod -aG staff,docker jenkins

USER jenkins

-----

docker build -t jenkins-lab-$USER .

docker run -dit --restart=always -p 8080:8080 -p 50000:50000 -v jenkins\_home:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkins-lab-$USER

RUN ON WEBPAGE

---------

http://172.17.5.38:8080/

----------

docker ps

get docker id e.g. 'a90e031df4b9' then run a shell inside it using:

sudo docker exec -it a90e031df4b9 bash

cat /var/jenkins\_home/secrets/initialAdminPassword

paste this password on the web page

-------

HOW TO SETUP GIT  
  
Certainly! Here are the basic commands for setting up a Git repository, cloning it, making changes, and pushing those changes:

**Clone a Repository:**

bash

git clone <repository\_url>

Replace <repository\_url> with the URL of the Git repository you want to clone.

**Setup a New Git Repository:**

If you're starting a new project and want to initialize a Git repository in an existing folder:

bash

cd /path/to/your/project

git init

**Add and Commit Changes:**

1. **Stage Changes:**

bash

 git add .

This command stages all changes. If you want to stage specific files, replace the . with the file names.

 **Commit Changes:**

bash

1. git commit -m "Your commit message"
2. Replace "Your commit message" with a concise description of the changes you made.

**Push Changes to Remote Repository:**

bash

git push origin <branch\_name>

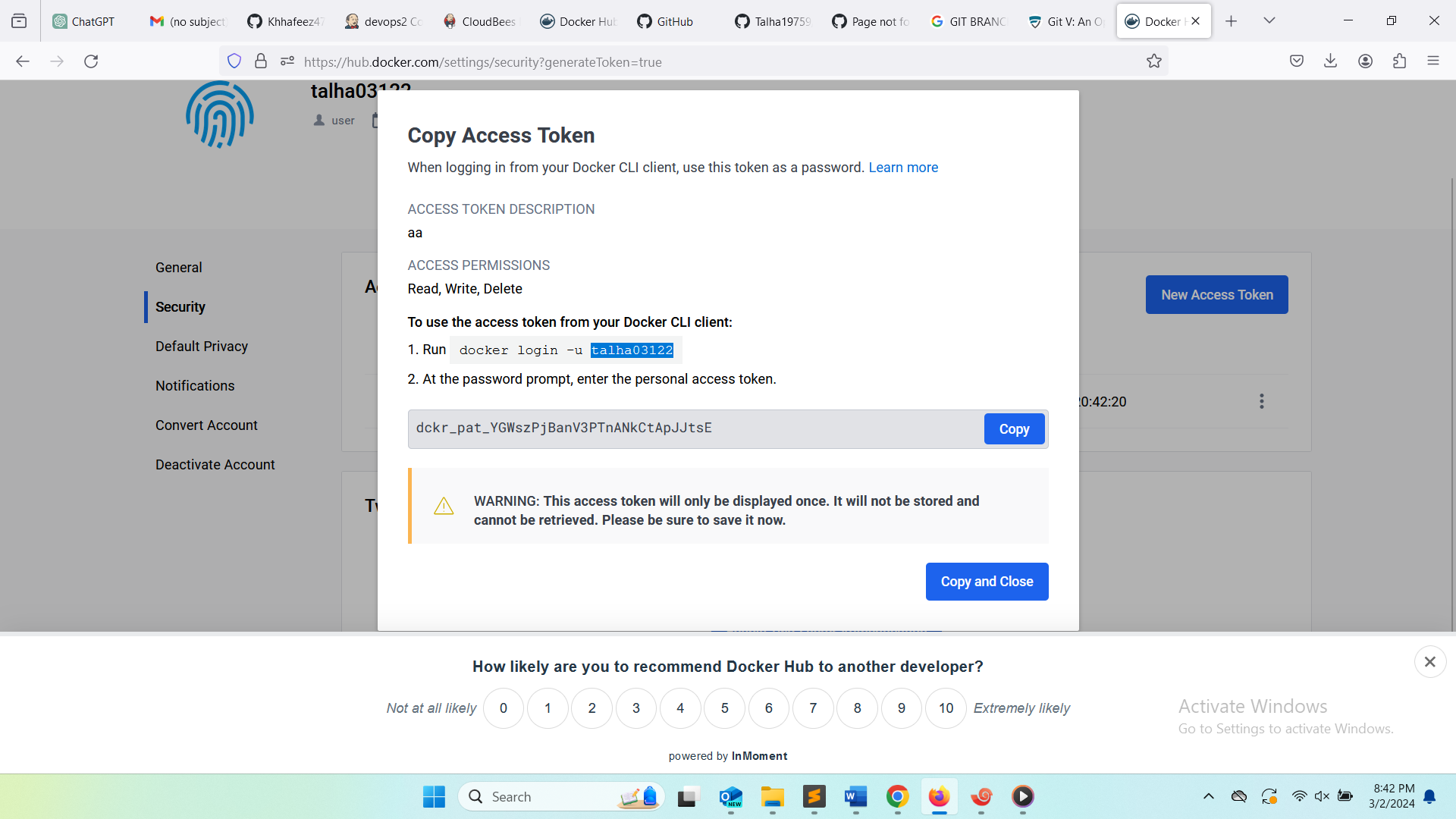
Replace <branch\_name> with the name of the branch you want to push. If you're pushing to the main branch, it's often named 'main' or 'master.'

Remember, if this is your first push to a new repository, you might need to set the upstream branch using the -u option:

bash

git push -u origin <branch\_name>

These commands cover the basic workflow of cloning a repository, initializing a new repository, adding and committing changes, and pushing those changes to a remote repository. Adjust the commands based on your specific use case and repository structure.

Docker:   
  
user: talha03122  
pass: dckr\_pat\_YGWszPjBanV3PTnANkCtApJJtsE  


Docker execiute commands  
  
#!/bin/bash

docker stop nodeapp

docker rm nodeapp

docker pull talha03122/devops:latest

docker run -d -p 5000:5000 --name nodeapp talha03122/devops:latest