\*\*\*\*\*\*\*\* Docker with Jenkins \*\*\*\*\*\*\*\*\*\*\*\*\*

create instance on AWS with ubuntu 20.01 OS

once the instance started connect it with the help of MobaXterm

-------------------------------------

Step 1:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Docker installation on Ubuntu OS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> sudo apt-get update

> sudo apt-get install ca-certificates curl gnupg lsb-release

>curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

> echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list /dev/null

\*\*\*\*\* Install Docker Engine \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> sudo apt-get update

> sudo apt-get install docker-ce docker-ce-cli containerd.io

\*\*\*\*\*\*\*\*\* let's check docker installed or not \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> sudo docker -v or sudo docker --version

> sudo docker info

Step 2: Install JDK

-----------------------------

>sudo apt-get update

>sudo apt install default-jdk -y

Step 3: install maven

-----------------------------

>sudo apt install maven -y

Step 4: install jenkins

------------------------------------

> curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

> echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

> sudo apt-get update

> sudo apt-get install jenkins

\*\*\*\*\*start jenkins service\*\*\*\*\*

> sudo service jenkins start

\*\*\*\*\*to check jenkins service\*\*\*\*\*

> sudo service jenkins status

> ctrl + c

>sudo cat /var/lib/jenkins/secrets/initialAdminPassword

(copy the initialAdminPassword)

open publicIp:8080

enter password and install suggested Plugins

---------------------------------------------------------------------------------------------

set up login

add username and password (remember) , add full name and email

continue and you will be redirected to jenkins Dashboard

------------------------------------------------------------------------------------------

Step 5: install Docker plugin in jenkins

manage jenkins> manage plugins> available >docker > install without retstart

Step 6: configure Docker cloud

manageJenkins> manage nodes and clouds> check lefthand side (configure cloud)>

>click on add a new cloud> select docker > click on cloud details

>Enter Host URI: unix:///var/run/docker.sock

test connection if getting error of access denied

go to you instance console and execute below command

>sudo chmod 777 /var/run/docker.sock

>refresh jenkins

and test it again

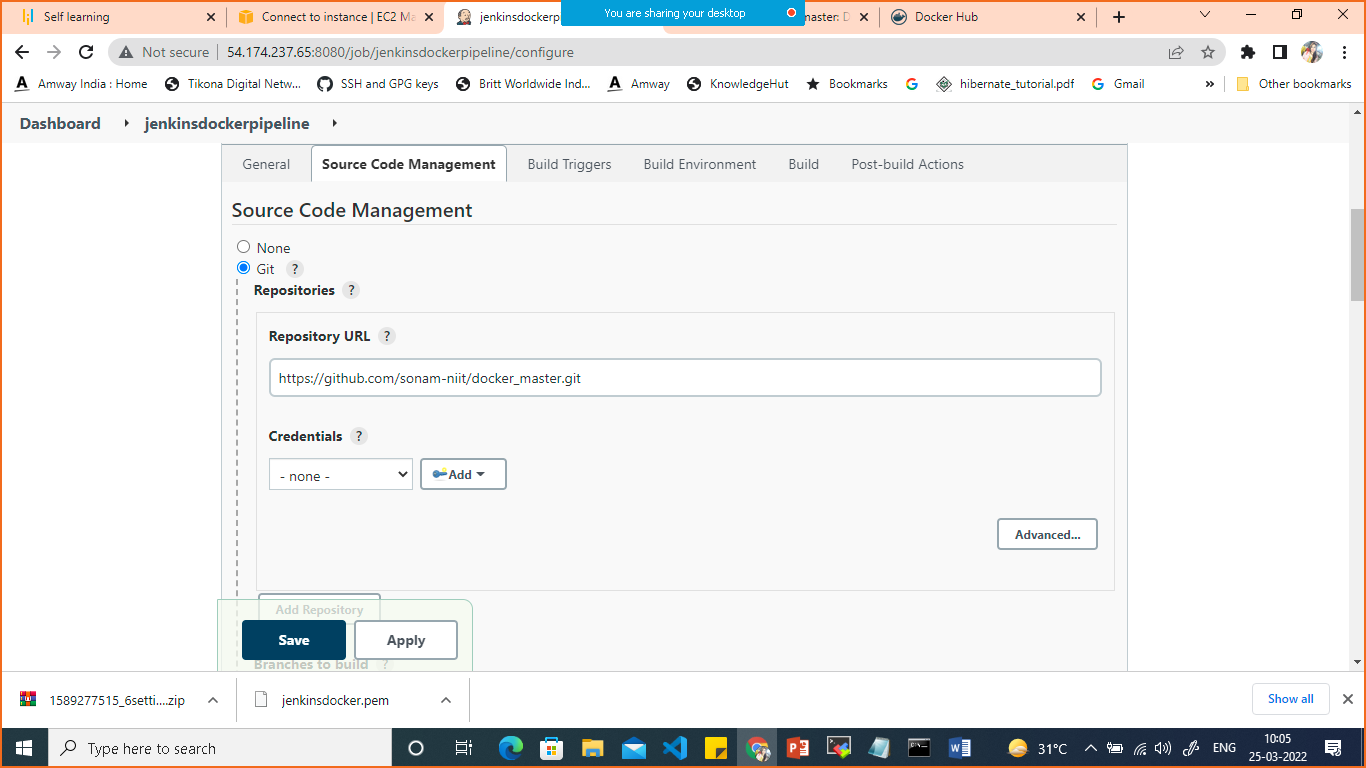
----------------------------------------------------------------------------

step 7:

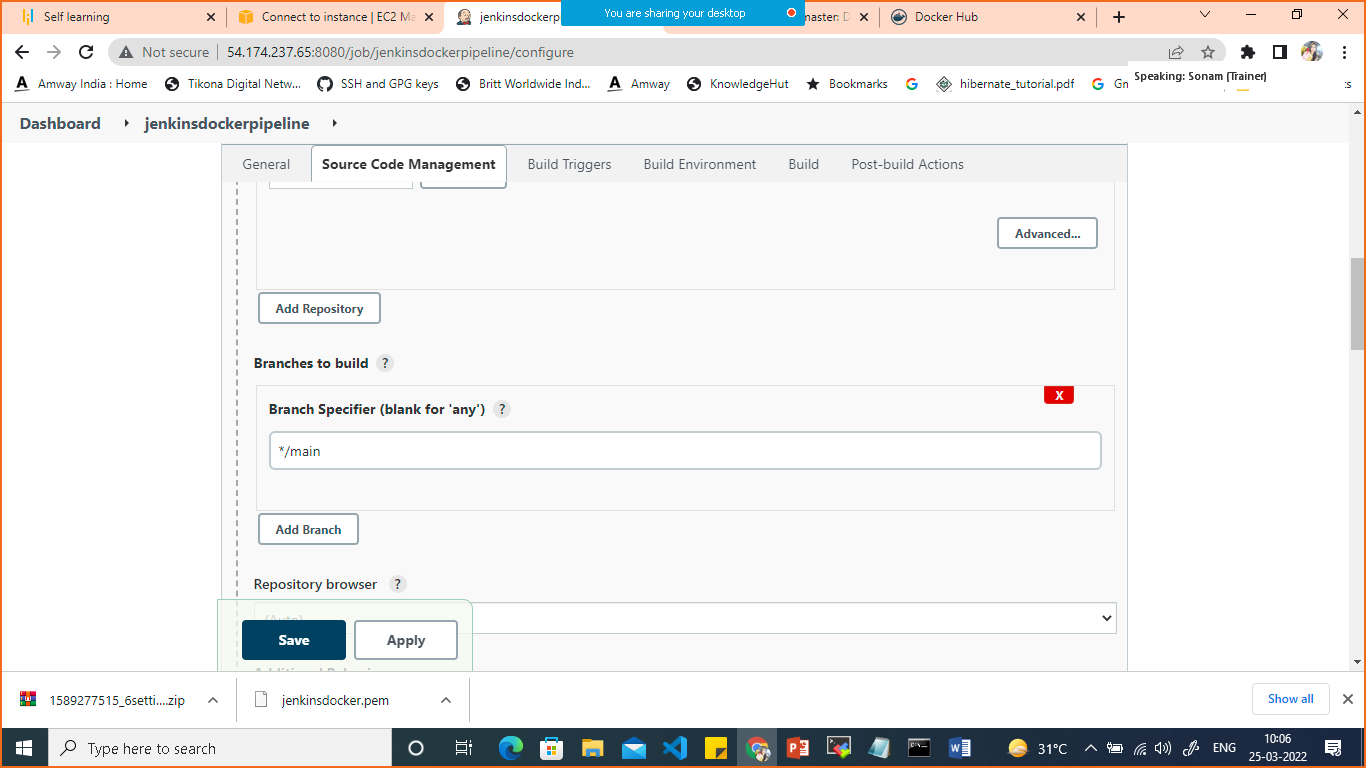
click on new Item

jenkinsdockerpipeline> freestyle project>ok

click on configure

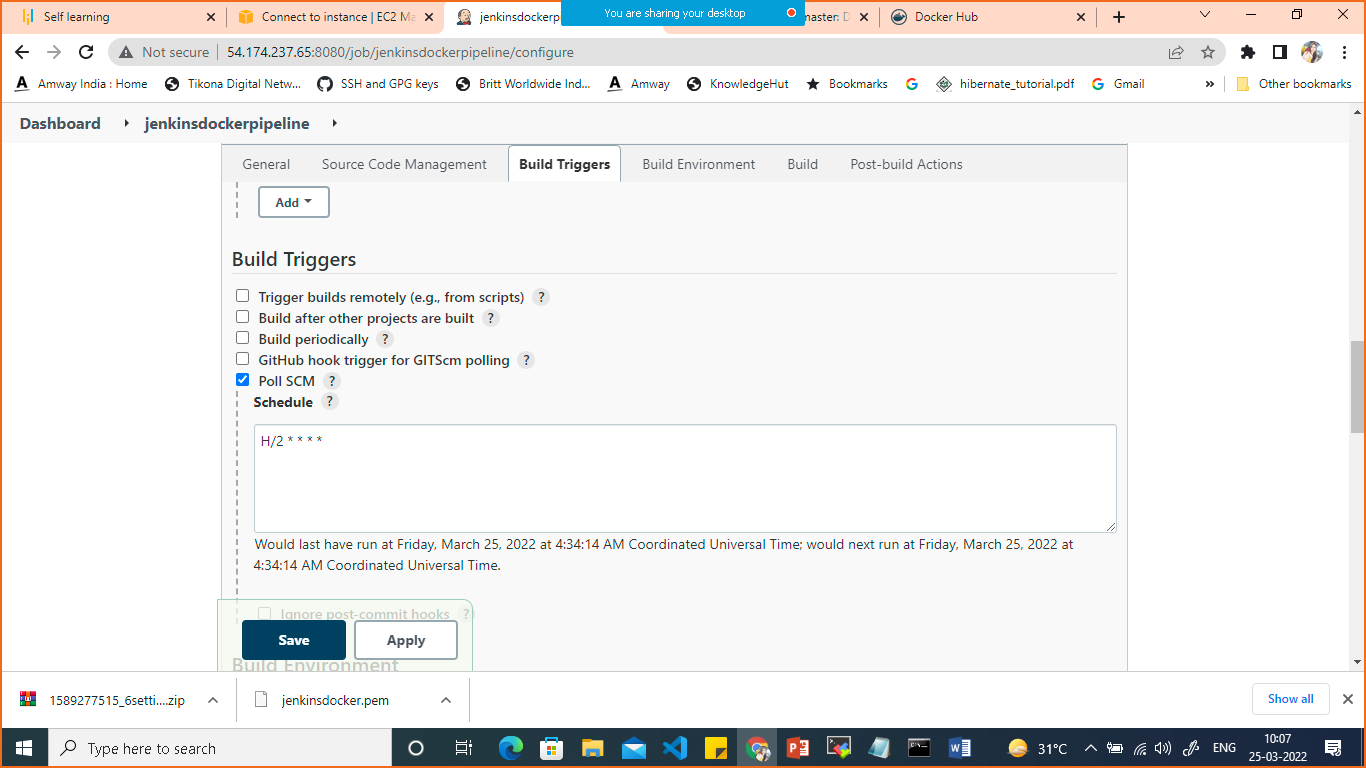


In Source Code management add git repo URL

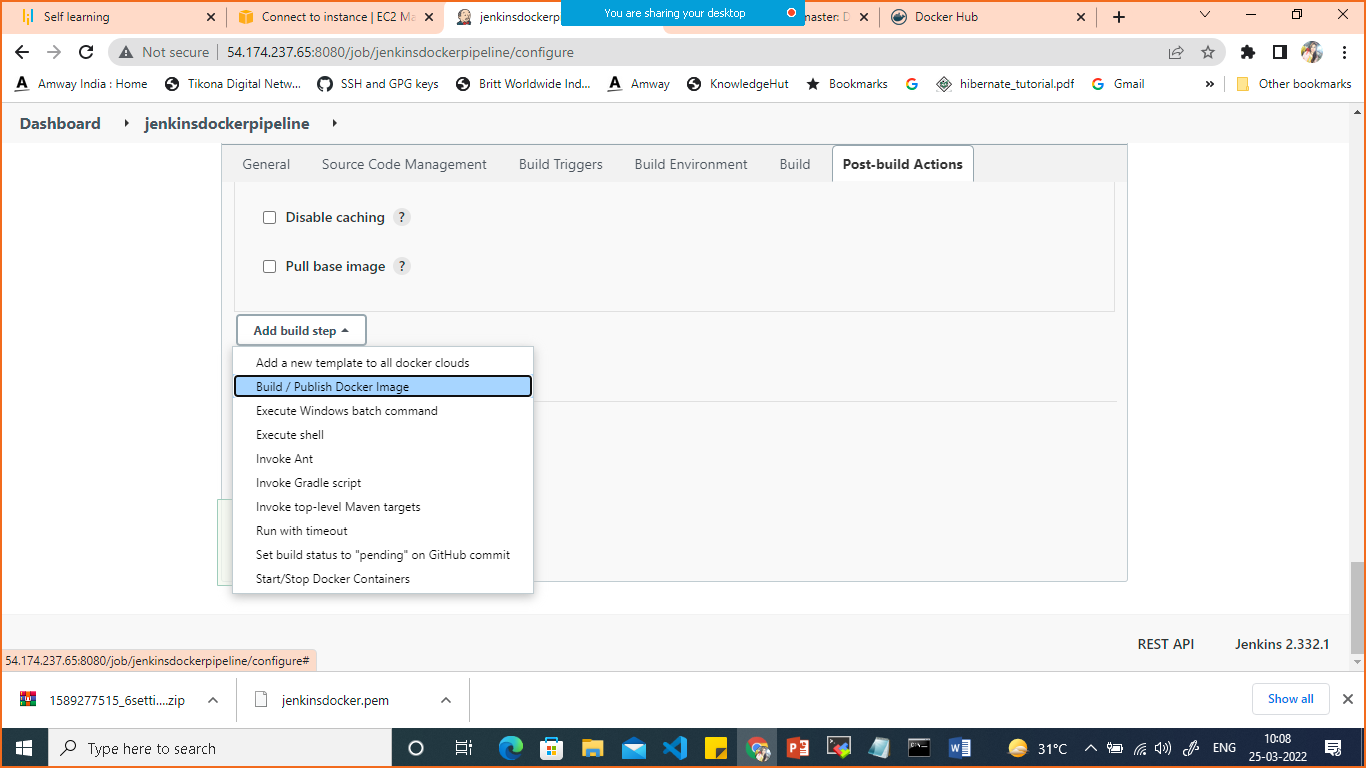


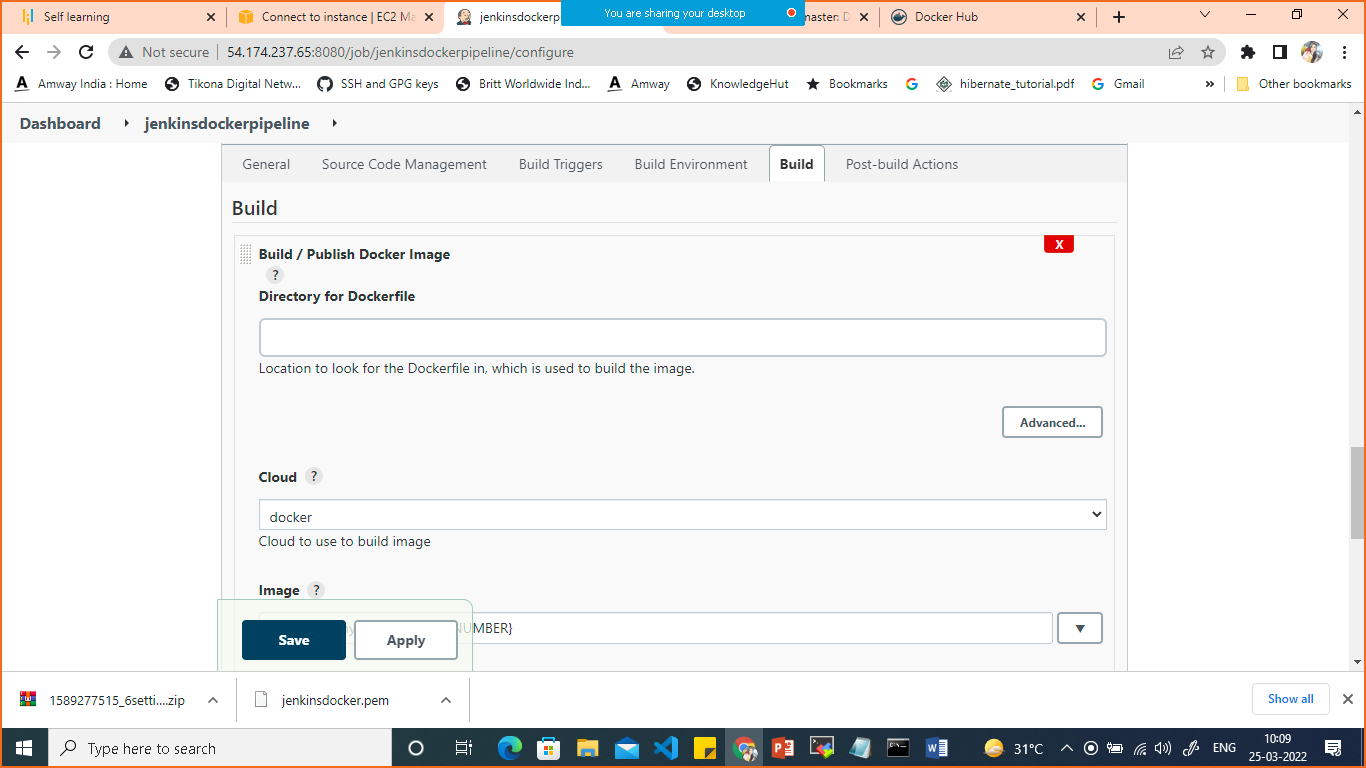
Give the branch name as main

Click on Poll HCM

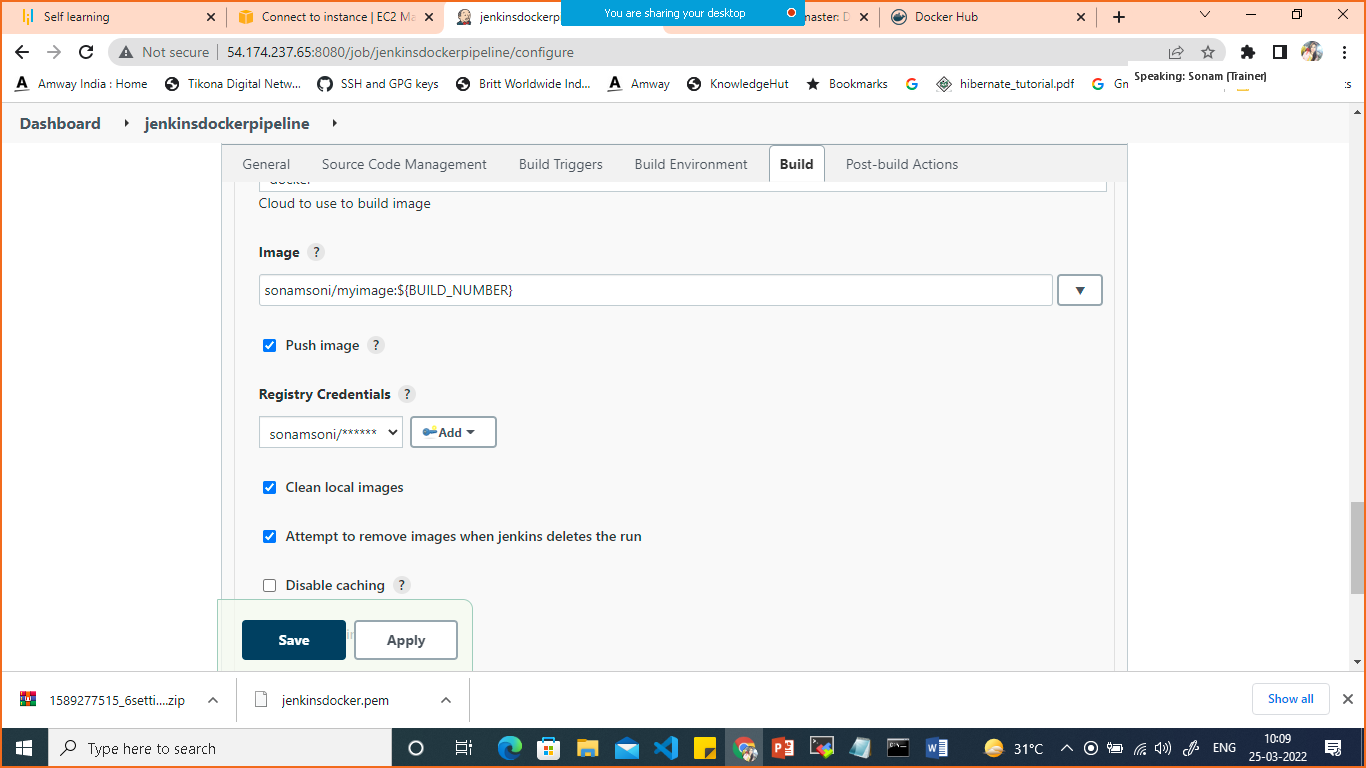


Scroll down in build select add and publish docker





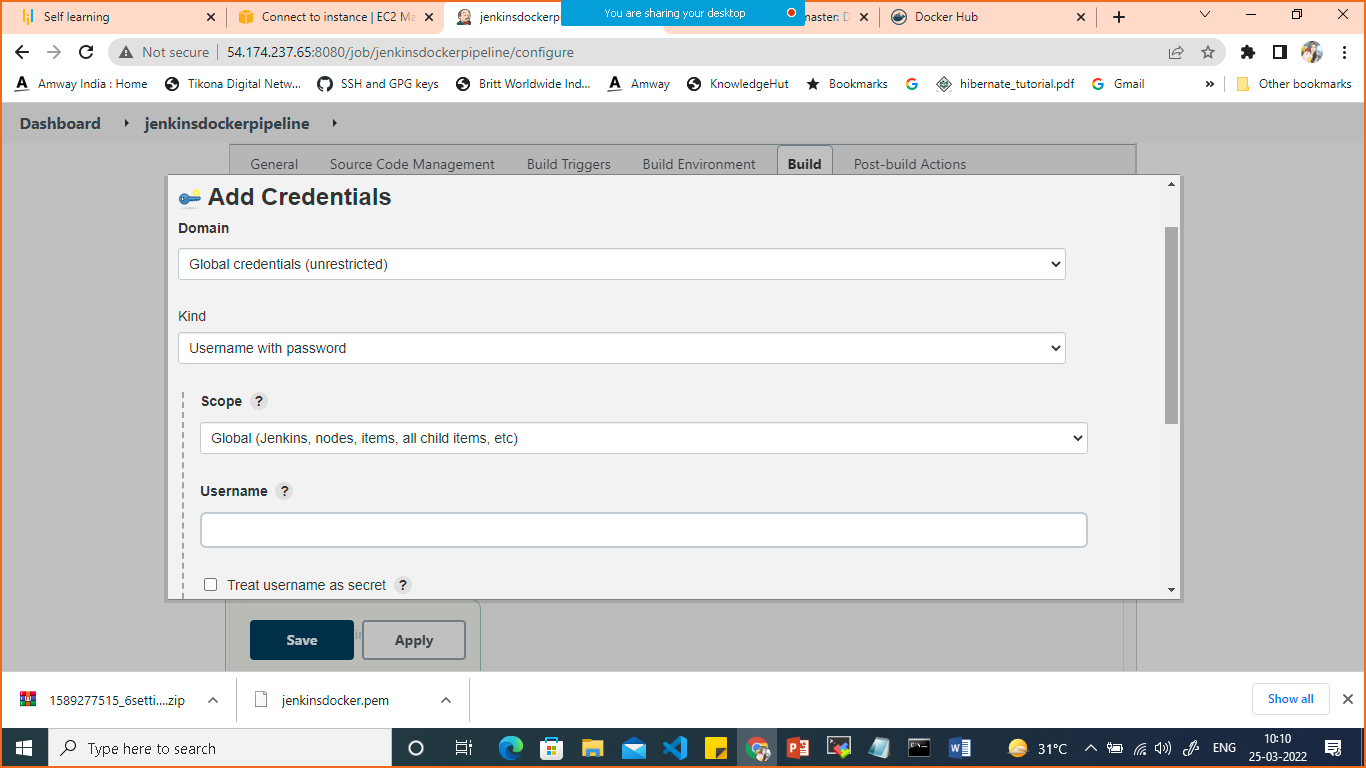
In cloud select docker



For adding docker credentials

Click on add credentials >Jenkins

Select Kind> username and password



Give dockerhub username and password

Then select your credentials in registry credentials which you have added.

Save.

Click on buid now

It will build the image and push it on dockerhub

If you make any changes in your github automatically it will build a new image.

\*\*\*\*\*\* run image on docker swarm \*\*\*\*\*\*\*

swarm means managing multiple conatiners running

on multiple hosts

------------------------------------

2 node : 1 node is manager and another node is worker

>docker swarm init

(it wil make my node as manager node and give one command, if you execute that command on another instaance --> that instance will connect to your manager node as worker.)

once swarm initialized check the running node

>docker node ls

Let's run the service on docker swarm

>docker service create -p 80:80 --name webserver myimage

>docker service ls (check running service)

>curl localhost:80

\*\*\*\*\*\* Scale the container with Docker swarm \*\*\*\*\*\*\*\*\*\*\*

>docker service scale webserver=3

(this will scale your service with 3 services running)

to check service running

>docker service ls

to check howmany same services running

>docker service ps webserver

(you will be able to see 3 running services)