**Create Jenkins in docker**

Commands:

create jenkins :

docker pull jenkins/jenkins

docker run -d -u root --name jenkins -p 8080:8080 -p 50000:50000 -v /root/jenkins\_2112:/var/jenkins\_home jenkins/jenkins:lts

**or**

docker run --name jen-test -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts

**with link the socat**

docker run -d --restart=always -u root --name jenkins --link socat:socat -p 8080:8080 -p 50000:50000 -v /root/jenkins\_2112:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkins/jenkins:latest

**create socat : (1st)**

docker pull alpine/socat

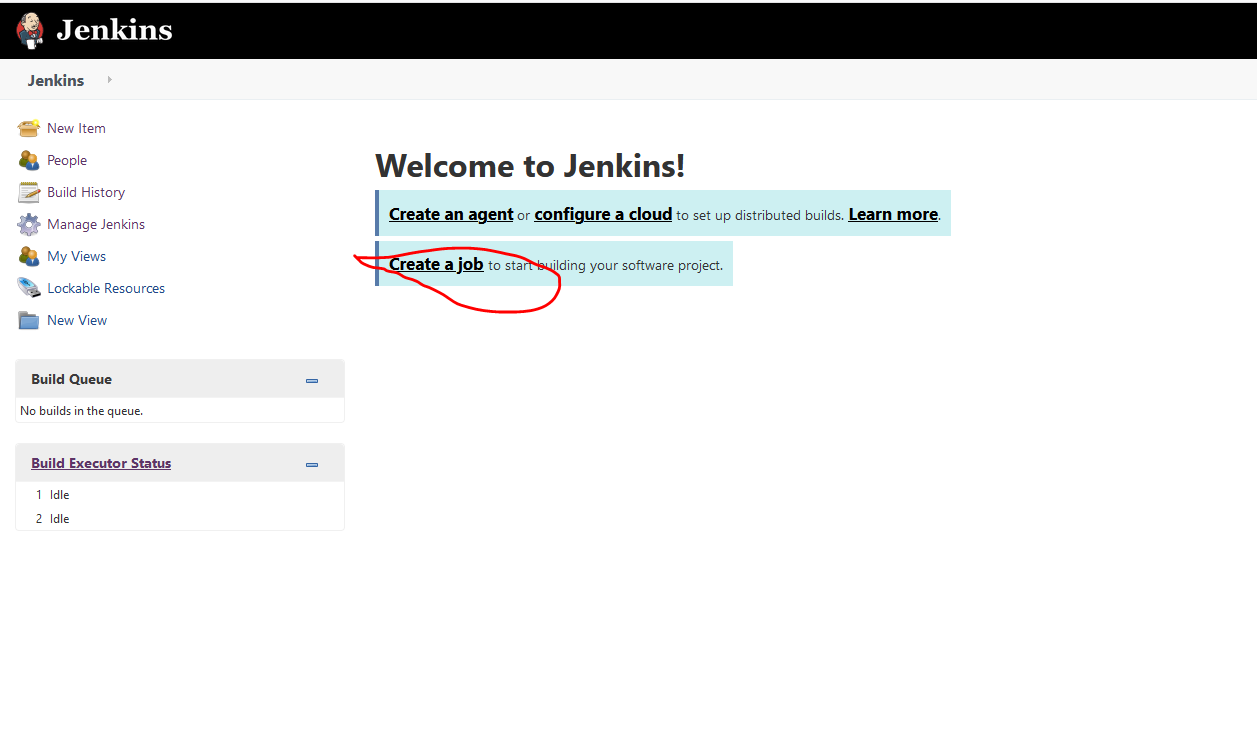
docker run -d --restart=always -p 127.0.0.1:2376:2375 -v /var/run/docker.sock:/var/run/docker.sock alpine/socat tcp-listen:2375,fork,reuseaddr unix-connect:/var/run/docker.sock

**for view the frist password**

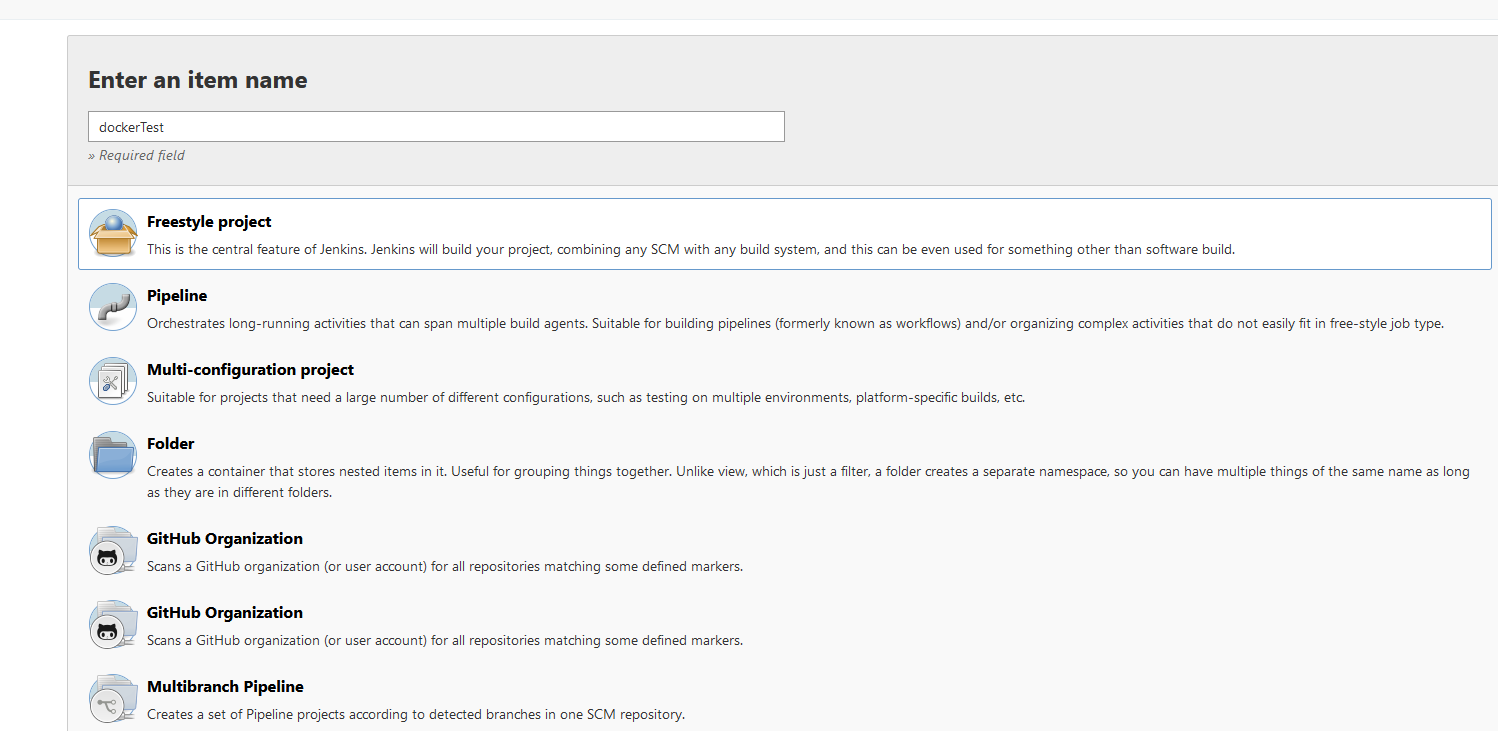
docker exec -it <container-name> cat /var/jenkins\_home/secrets/initialAdminPassword

**Configure project in Jenkins from git :**

**Step 1 :**

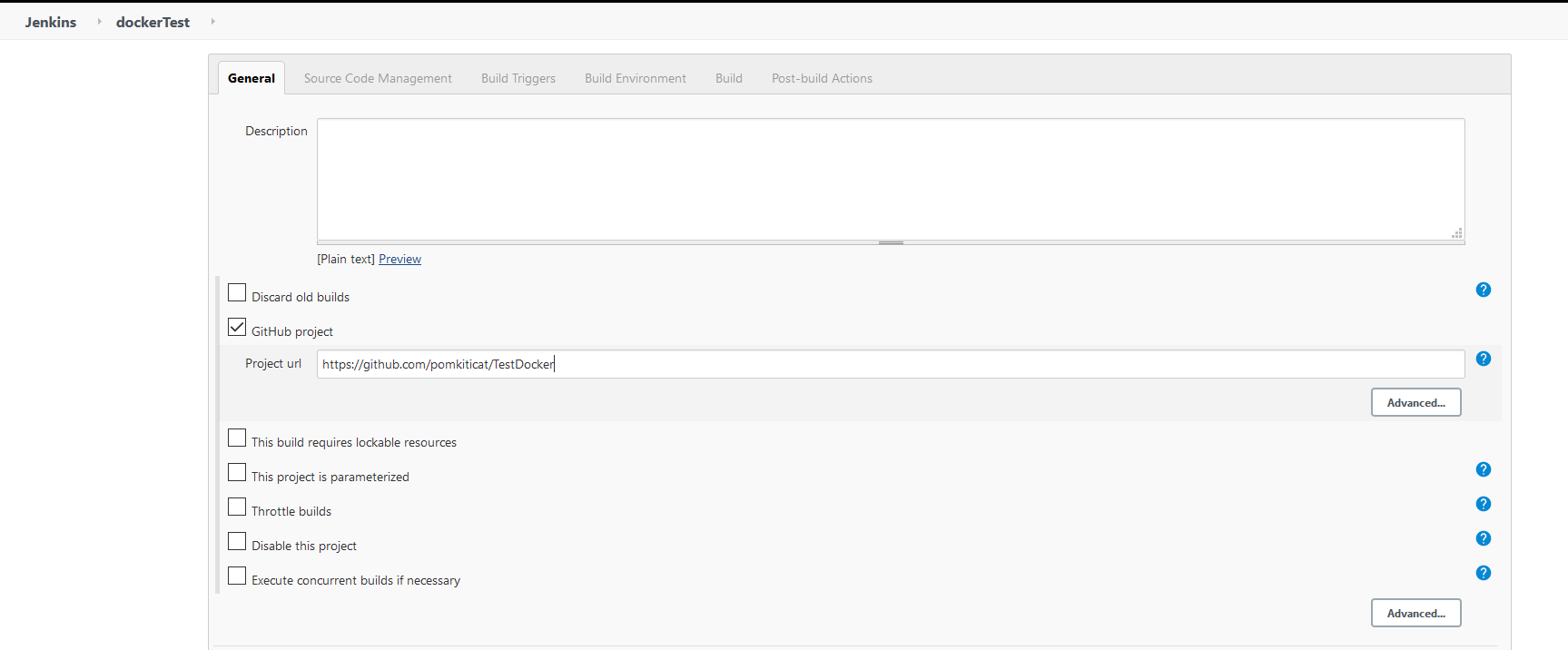


**Step 2:**

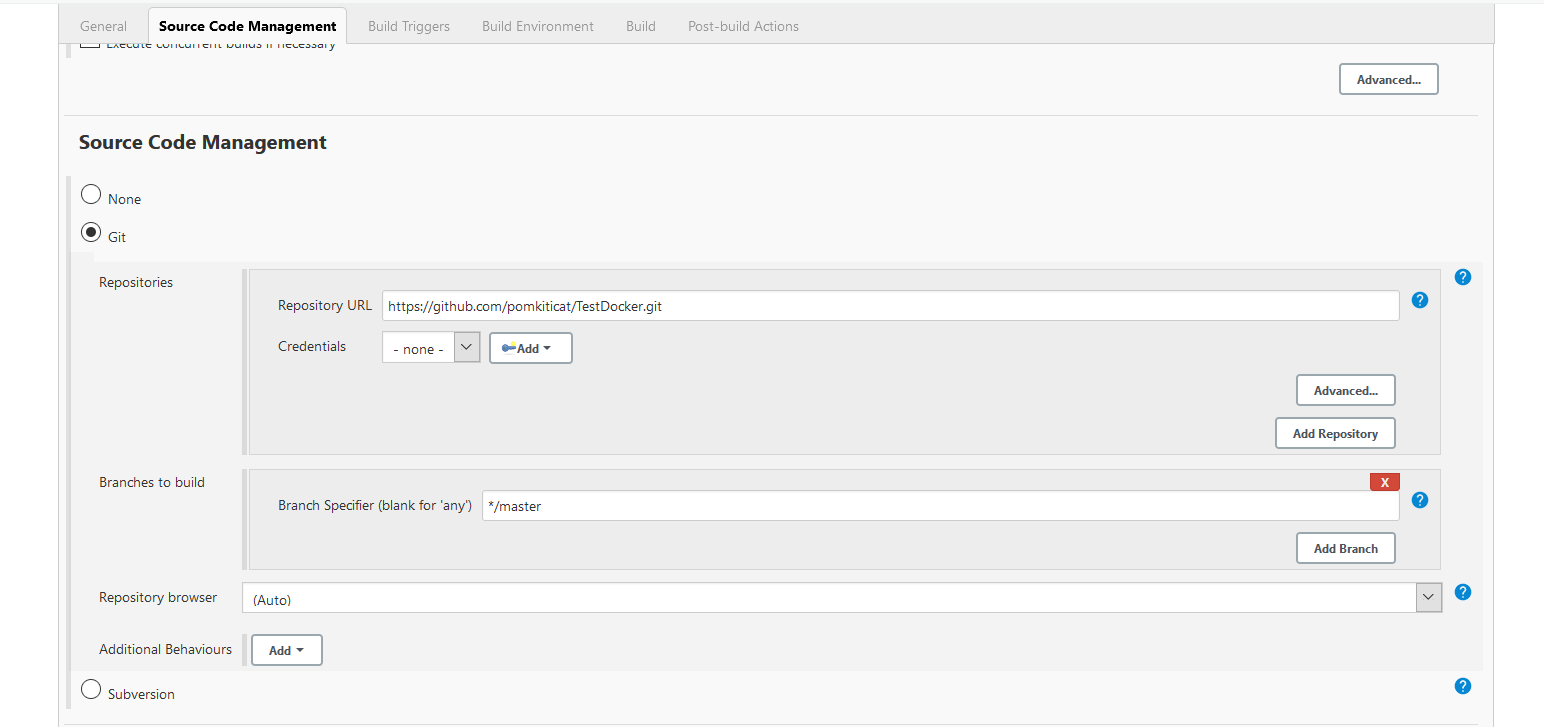


Click OK

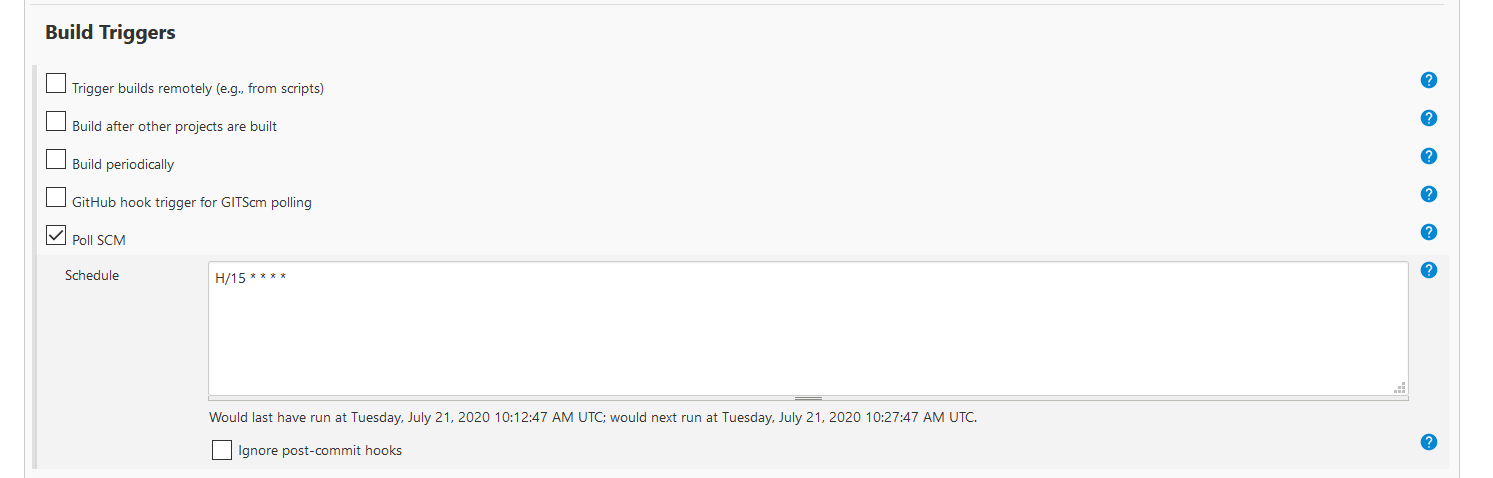
**Step 3:**



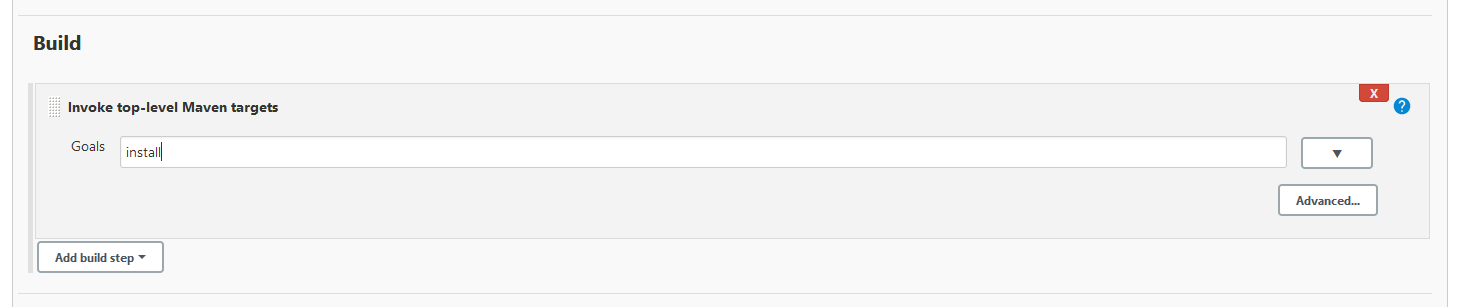
**Step 4:**



**Step 5 :**



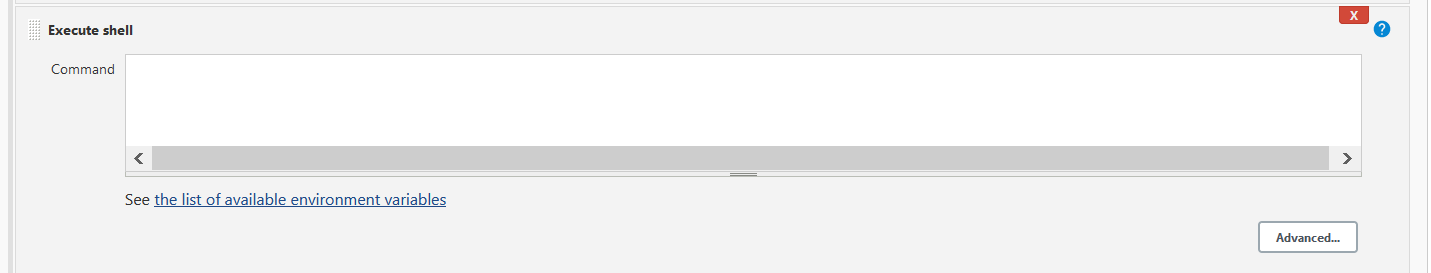
**Step 6:**



For maven build

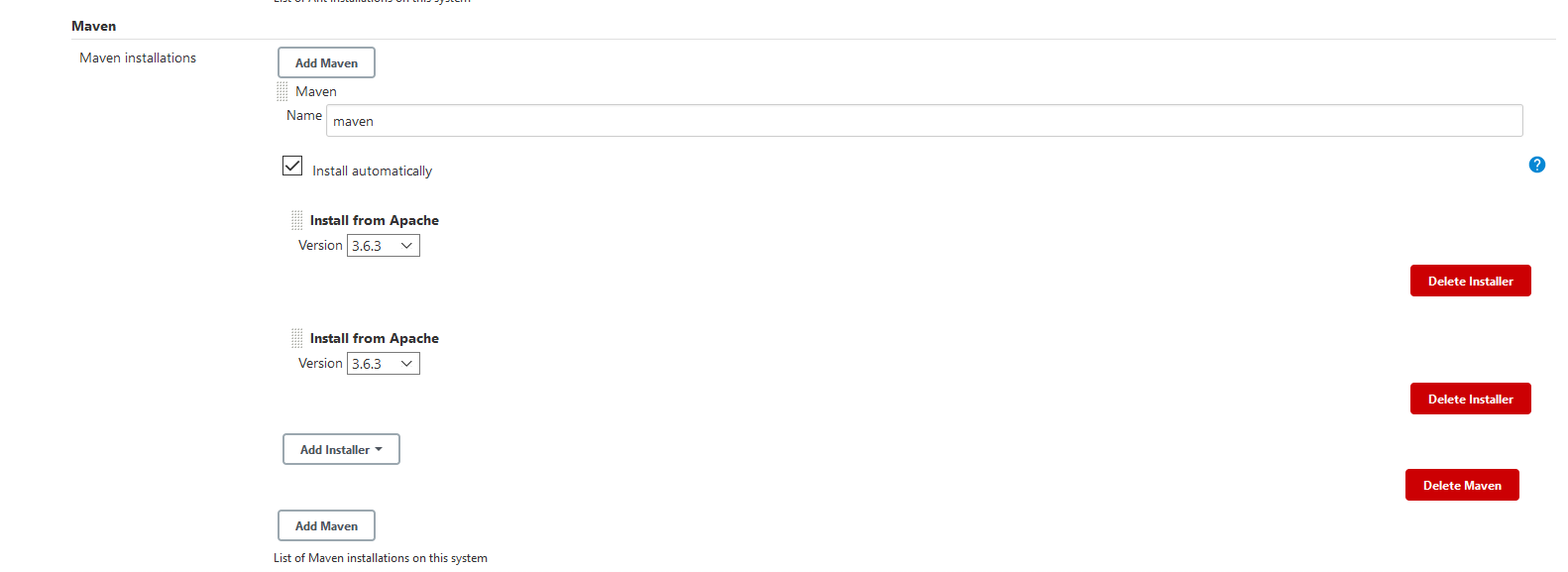
================================

For docker command

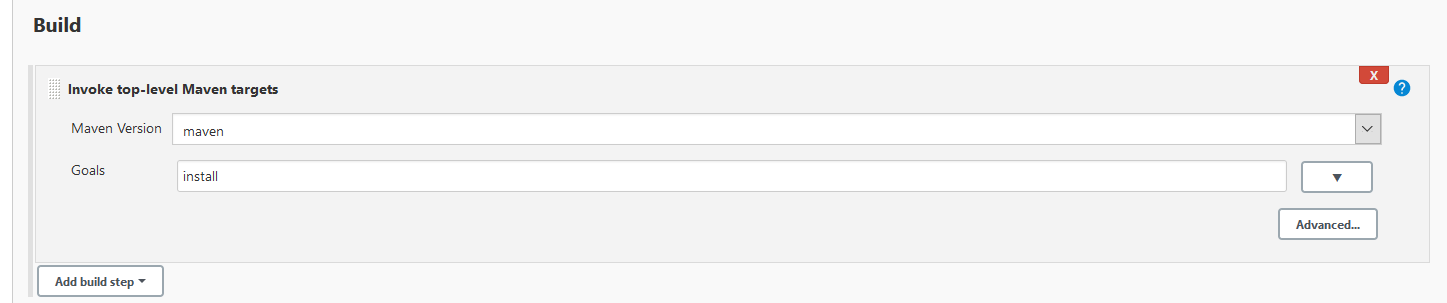


**For maven setting :**

Jenkins-> Global Tool Configuration ->maven



Change Step 6 after setting maven



**For Docker :**

Jenkins -> Plugin Manager

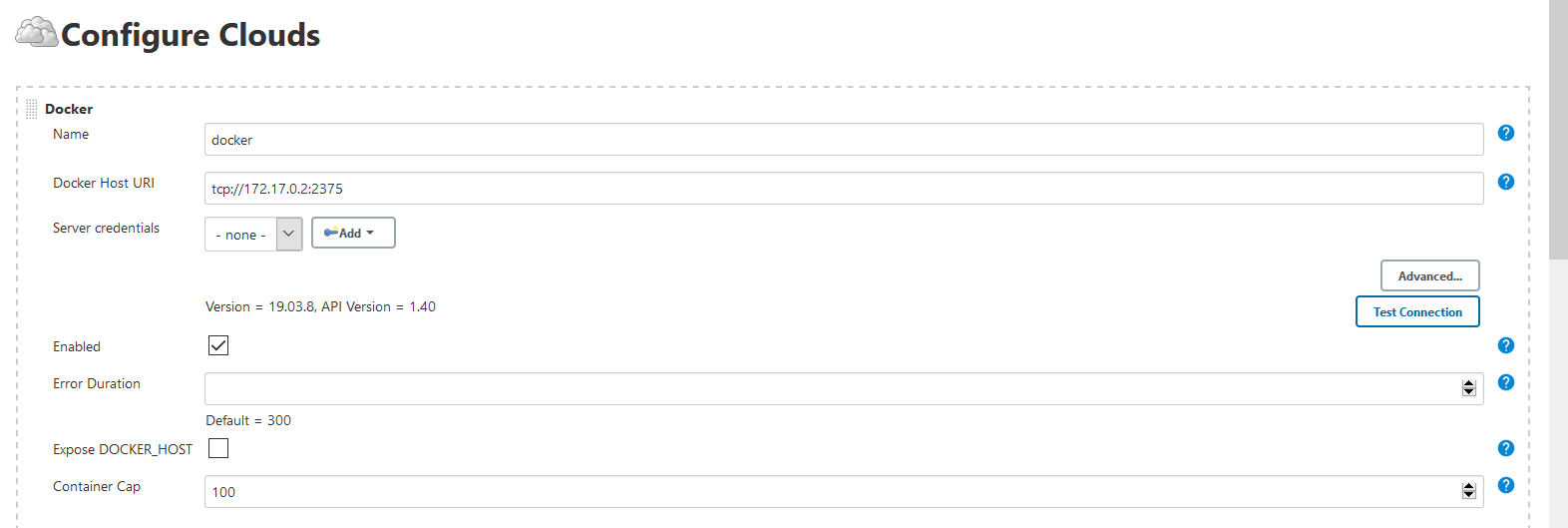
Search docker plugin is install or not . if not then install

Select **Docker** from **Install** list .And **Install without restart**

**Step 1:**

Manage Jenkins -> configure -> select cloud

**Step 2:**



Tcp://ip-socket:2376

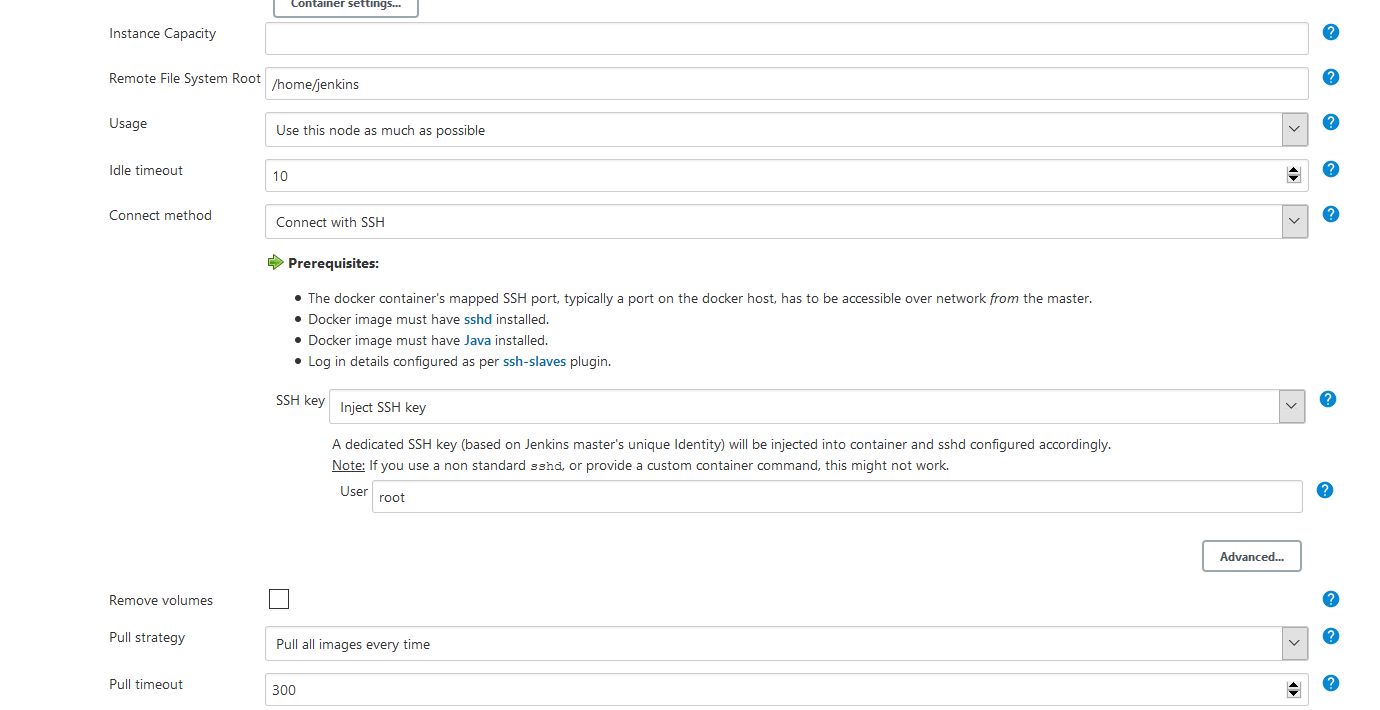
**Get ip of the socat :**

docker inspect --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" <socat container>

Click Enable

**For Docker Agent templates**





**Set values:**

1. Label name: docker-agent
2. Docker Image : benhall/dind-jenkins-agent:v2
3. Under **Container Settings**, In the "Volumes" text box enter /var/run/docker.sock:/var/run/docker.sock. This allows our build container to communicate with the host.
4. For **Connect Method** select **Connect with SSH**. The image is based on the Jenkins SSH Slave image meaning the default Inject SSH key will handle the authentication.
5. Make sure it is Enabled.

**Click Apply and Save**

**Build command:**

ls

docker info

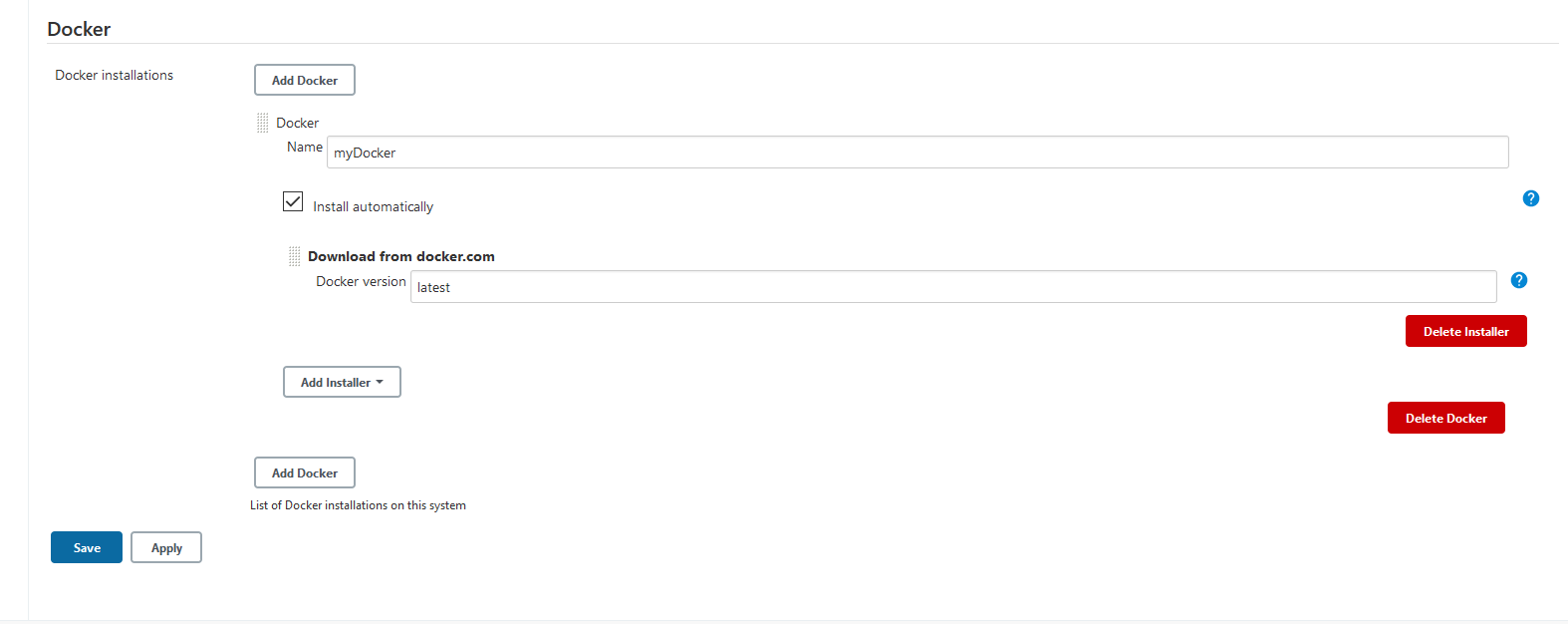
docker build -t katacoda/jenkins-demo:${BUILD\_NUMBER} .

docker tag katacoda/jenkins-demo:${BUILD\_NUMBER} katacoda/jenkins-demo:latest

docker images

**Install Docker :**

Manage Jenkins -> Global Tool Configuration -> Docker



**Dockerfile for install docker:**

FROM jenkins/jenkins:lts

MAINTAINER 4oh4

# Derived from https://github.com/getintodevops/jenkins-withdocker (miiro@getintodevops.com)

USER root

# Install the latest Docker CE binaries and add user `jenkins` to the docker group

RUN apt-get update && \

apt-get -y --no-install-recommends install apt-transport-https \

ca-certificates \

curl \

gnupg2 \

software-properties-common && \

curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg > /tmp/dkey; apt-key add /tmp/dkey && \

add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") \

$(lsb\_release -cs) \

stable" && \

apt-get update && \

apt-get -y --no-install-recommends install docker-ce && \

apt-get clean && \

usermod -aG docker jenkins

# drop back to the regular jenkins user - good practice

USER Jenkins

**Command:**

docker run -d --restart=always -u root --name jenkins-new --link socat:socat -p 8080:8080 -p 50000:50000 -v /root/jenkins\_2112:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkins-docker