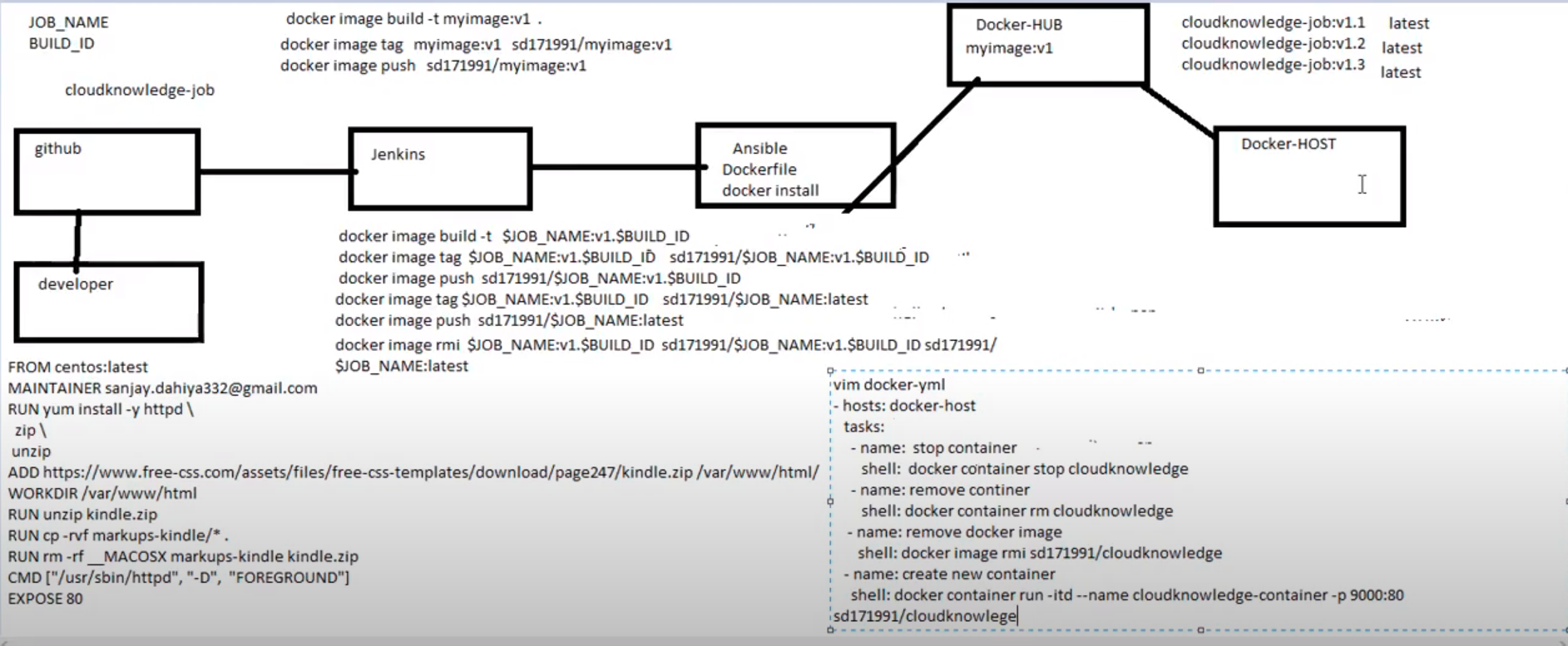
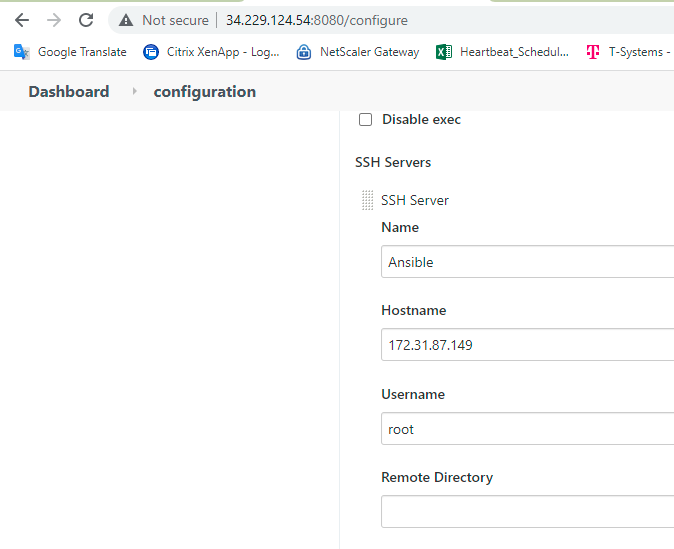
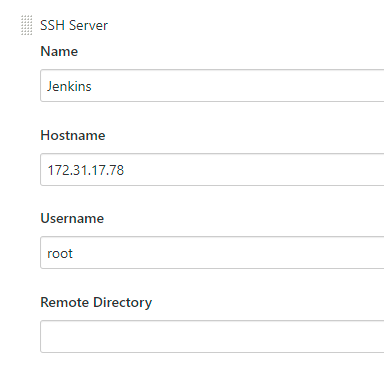
To deploy website developed by developer on Docker container.

* Pre-requisites :
* Need 3 different VM’s Jenkins, Ansible and Docker Host
* Need Github and Dockerhub accounts.
* Below is the POC for the project :

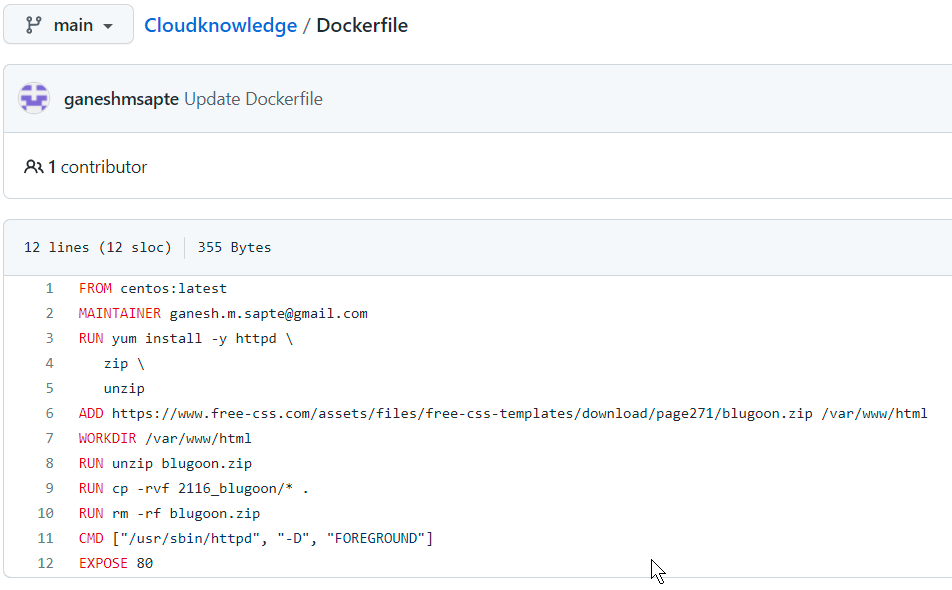


* Configuration :
* Create 3 VM’s and login to them using Putty.exe
* Install Jenkins, Git on Jenkins Server.
* Install Ansible, Docker on Ansible Server, login to Docker hub account.
* Install Docker on Docker Host.
* Login to git-hub and create webhook with Jenkins server “ipaddress:8080/github/webhooks”.
* Reset root account password on Ansible and Docker host.
* Generate public ssh key on Jenkins and Ansible servers and copy it to Ansible and Docker host respectively.
* Open the Jenkins dashboard and login to it by creating user.
* Add Public over SSH plugin on Jenkins.
* Go to manager Jenkins and add Jenkins and Ansible as ssh servers with Private IP Addresses :

(Adv settings >> add password authentication and provide pswd of root user)

* Create new repository on Github and create Dockerfile in it with below contents :



=============================EXAMPLE 1========================================

FROM centos:latest

MAINTAINER ganesh.m.sapte@gmail.com

RUN yum install -y httpd \

zip \

unzip

ADD https://www.free-css.com/assets/files/free-css-templates/download/page247/kindle.zip /var/www/html

WORKDIR /var/www/html

RUN unzip kindle.zip

RUN cp -rvf markups-kindle/\* .

RUN rm -rf \_MACOSX markups-kindle kindle.zip

CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]

EXPOSE 80

=============================EXAMPLE 2=========================================

FROM centos:latest

MAINTAINER ganesh.m.sapte@gmail.com

RUN yum install -y httpd \

zip \

unzip

ADD https://www.free-css.com/assets/files/free-css-templates/download/page271/blugoon.zip /var/www/html

WORKDIR /var/www/html

RUN unzip blugoon.zip

RUN cp -rvf 2116\_blugoon/\* .

RUN rm -rf blugoon.zip

CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]

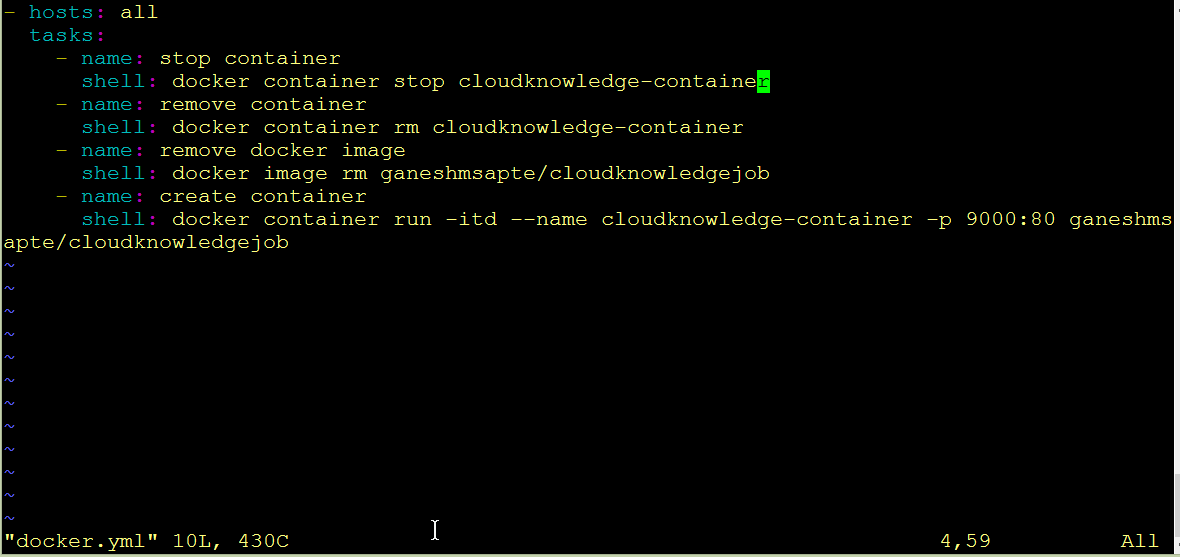
EXPOSE 80

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

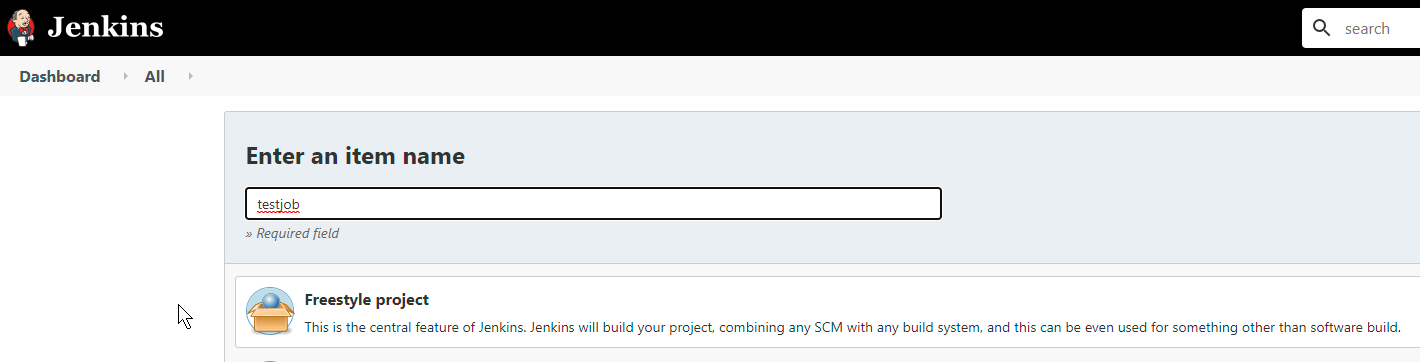
* Go to Ansible server >> /etc/hosts : add the tas as [dockerhost] and make entry of private IP addess in the file and save the changes.



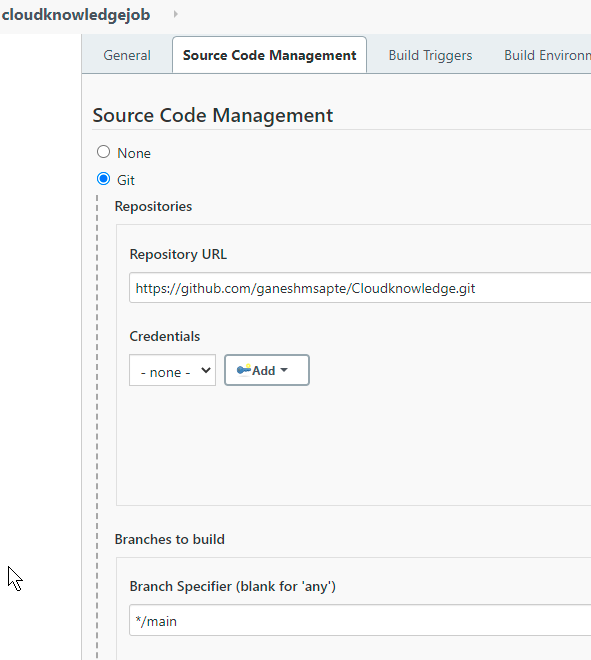
* On Ansible server : create new folder as sourcecode and create ‘docker.yml’ file with below contents : */sourcecode/docker.yml*



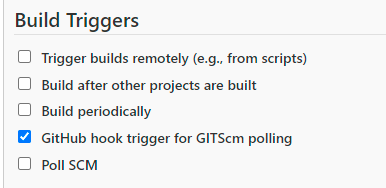
* Go to Jenkins dashboards and create new job as below :



* Add SCM as Git and provide the path of the repository which was created earlier :



* Mark build trigger as below, so when if there is any new commit to the repository on GitHub, it will trigger the configured Job :



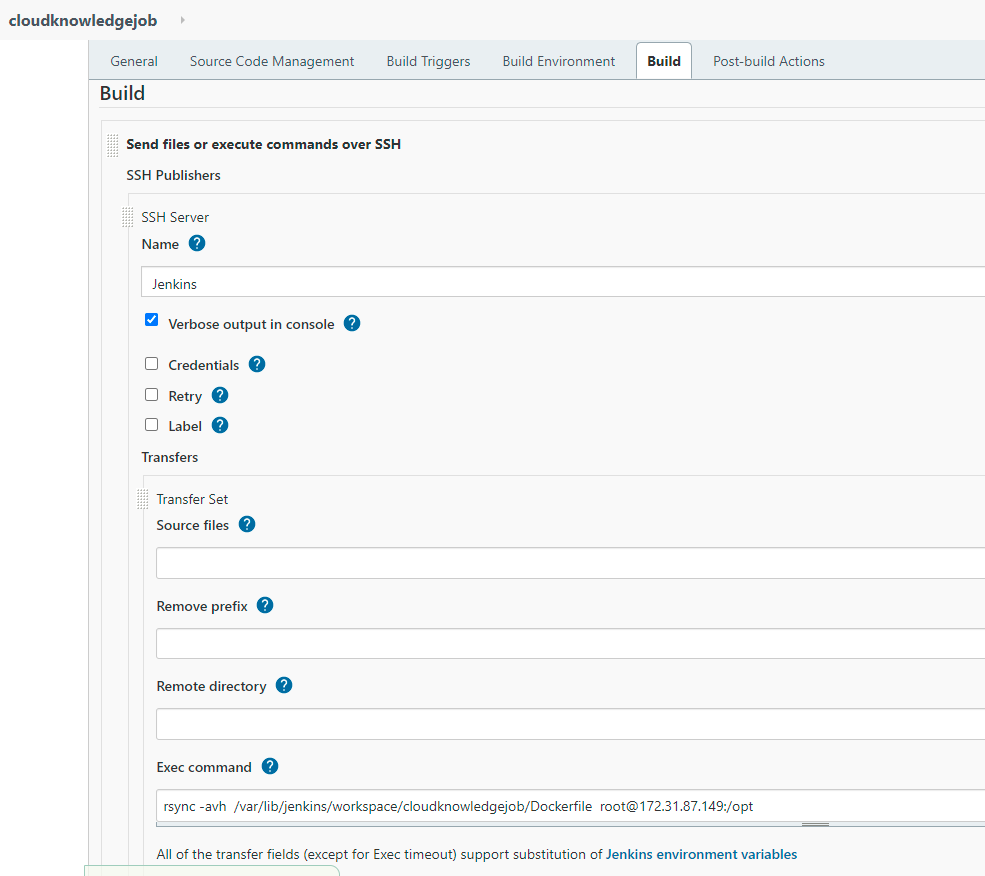
* Add build server as Jenkins and configure it as below :

So when Jenkins will compile and build the Dockerfile in Job’s Workspace folder,

It will be copied to /opt folder on Ansible server:

Additionally, click Verbose option to get detailed Console output:

*rsync -avh /var/lib/jenkins/workspace/cloudknowledgejob/Dockerfile root@172.31.87.149:/opt*



* Once the Dockerfile gets copied to /opt folder, Docker host installed on Ansible server will do the following :

Navigate to /opt folder:

Build two docker images with tags as Version nr and latest one, with latest Dockerfile in the /opt folder

Push the created Docker images to Docker hub

Delete the created images once it is pushed to Docker Hub.

cd /opt

docker image build -t $JOB\_NAME:v1.$BUILD\_ID .

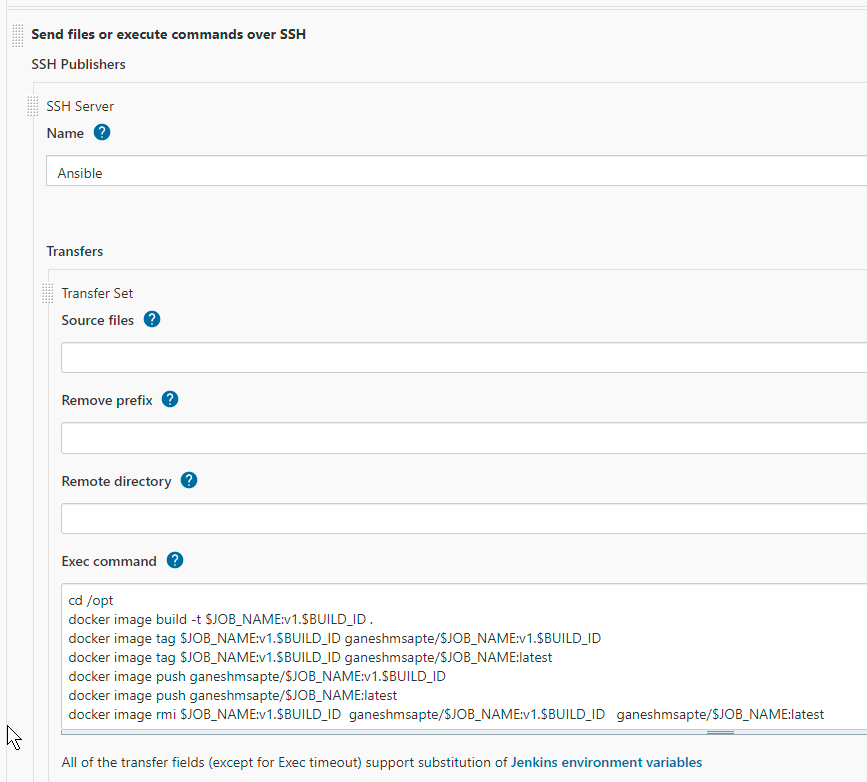
docker image tag $JOB\_NAME:v1.$BUILD\_ID ganeshmsapte/$JOB\_NAME:v1.$BUILD\_ID

docker image tag $JOB\_NAME:v1.$BUILD\_ID ganeshmsapte/$JOB\_NAME:latest

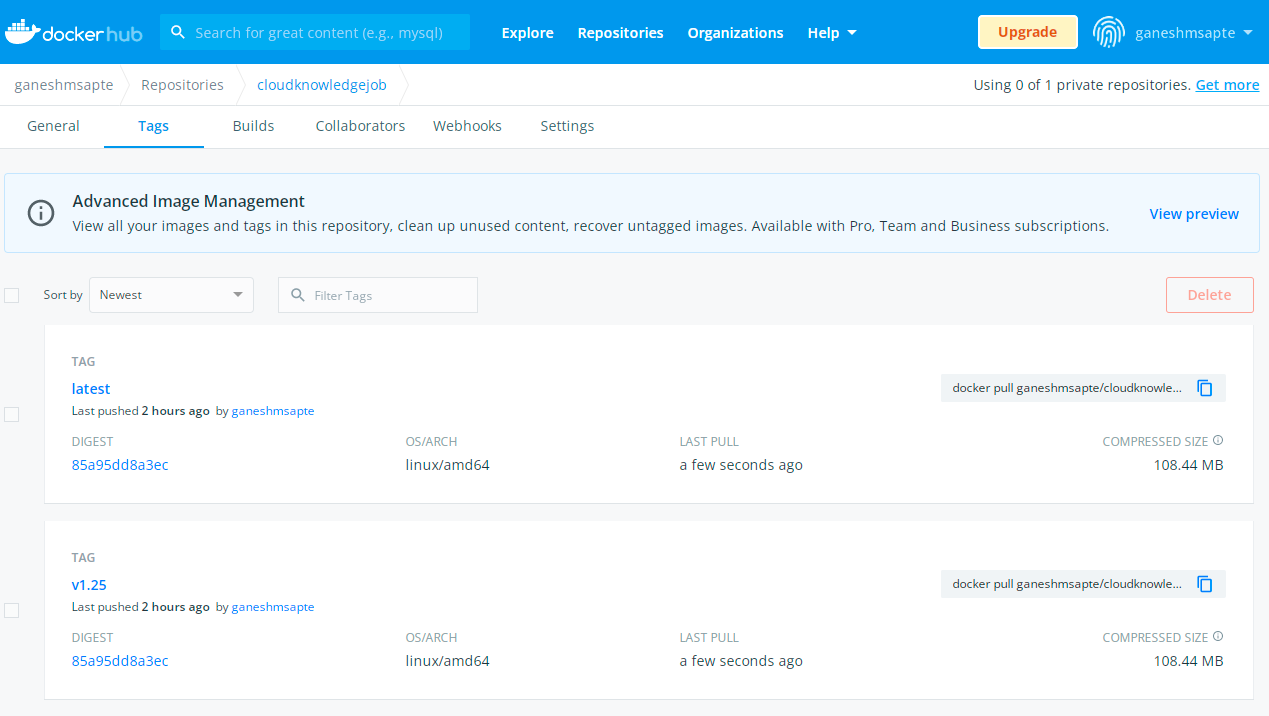
docker image push ganeshmsapte/$JOB\_NAME:v1.$BUILD\_ID

docker image push ganeshmsapte/$JOB\_NAME:latest

docker image rmi $JOB\_NAME:v1.$BUILD\_ID ganeshmsapte/$JOB\_NAME:v1.$BUILD\_ID ganeshmsapte/$JOB\_NAME:latest



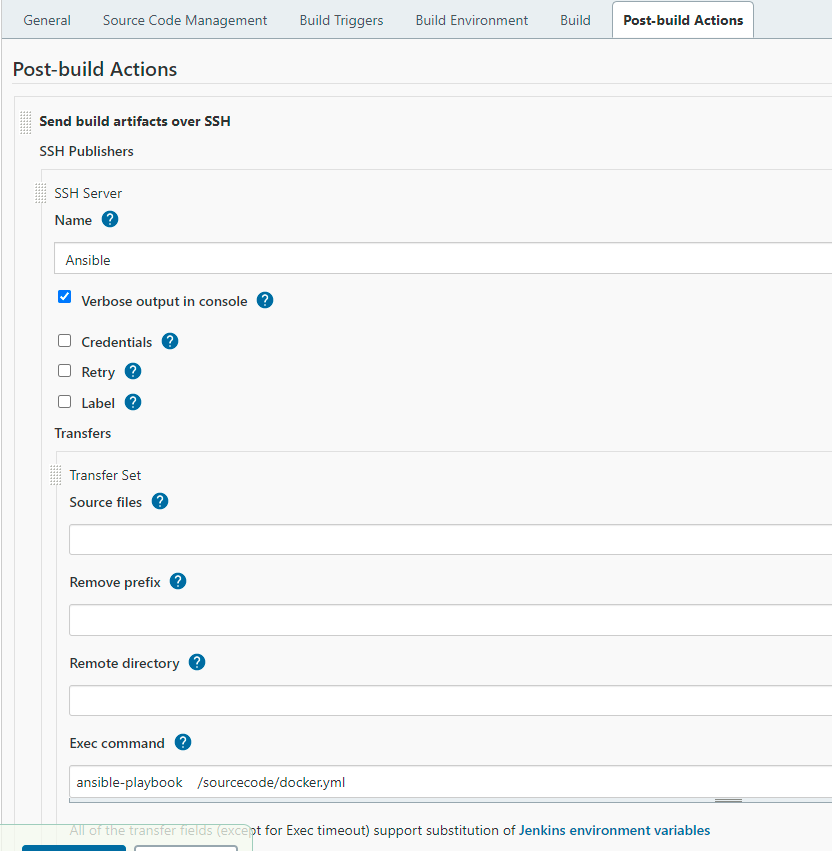
* On Docker Hub, images will be pushed as below :



* Once the build is done, configure the post build action as below:

It will trigger the Ansible playbook: /sourcecode/docker.yml

ansible-playbook /sourcecode/docker.yml

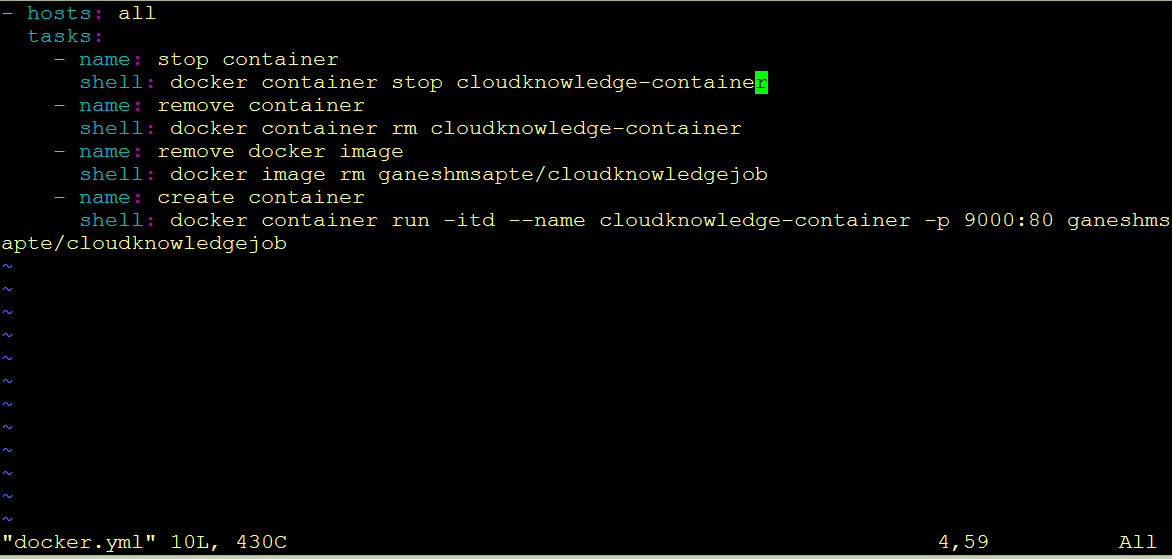


* Playbook as below :

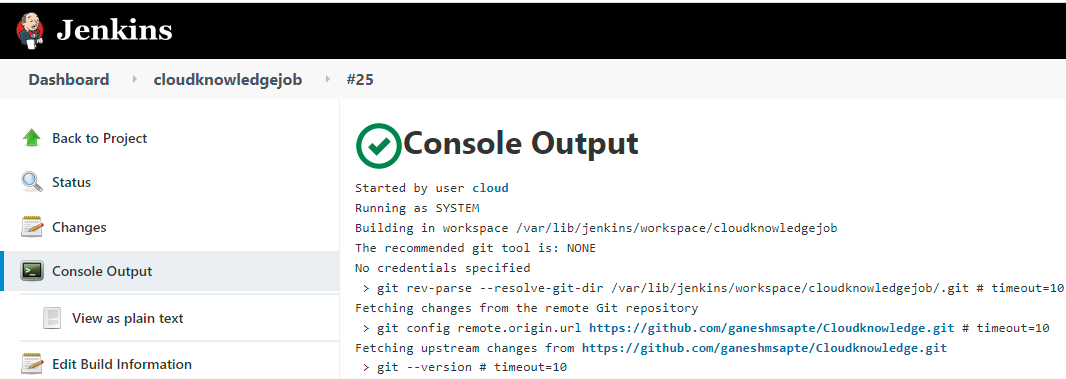
Playbook, will first check if existing container is present, if yes, it will stop and remove it.

Remove Docker images related to it.

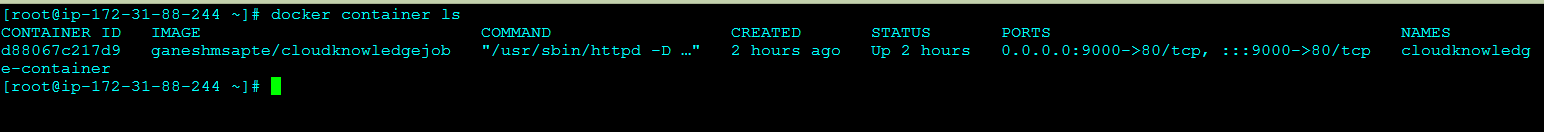
Create the new Container with port mapping in interactive mode with the help of latest image form Docker hub :



* Job Execution :
* Once the Dockerfile is committed on GitHub, job will get executed :



* Latest Dockerfile will be build and copied to Ansible server by Jenkins server.
* Docker on Ansible server, will create Docker image from it and push it to Docker hub.
* Docker Host will pull the latest image from Dockerhub and will create the docker container from it.



* By hitting the mapped port of Private IP address of Docker server, Website deployed on the Container can be accessed :

