All the info that this doc contains is from course https://paidcoursesforfree.com/jenkins-from-zero-to-hero-become-a-devops-jenkins-master/ git hub related to this course by the content creator has be forked <https://github.com/sasankkathera/jenkins-resources.git>

install docker and docker compose using official website

**https://docs.docker.com/engine/install/debian/** -- docker

**https://docs.docker.com/compose/install/** -- docker-compose

docker-compose is used to execute or apply yaml files.

Below starting the steps make sure all the files are in the specific location for the yml file to work properly or else change the yml contents to accordingly. For this files are represented in below format.

/

-->home

------>admin

------------>Debian

-------------------> Dockerfile

-------------------> id\_rsa.pub (public key file generated should be in RSA)

------------>docker-compose.yml

------------>jenkins\_home

understanding the below yml file contents (<https://github.com/sasankkathera/udemy-Download.git>)

version: '3'

services:

  jenkins:

    container\_name: jenkins

    image: jenkins/jenkins

    ports:

      - "8080:8080"

    volumes:

      - "/home/admin/jenkins\_home:/var/jenkins\_home"

    networks:

      - net

  sshtesting:

    container\_name: sshcreation

    image: sshtestin

    build:

      context: Debian

    networks:

      - net

networks:

  net:

In services section we have jenkins and sshtesting where it is as follows.

Jenkins:

jenkins (3rd line) is a hostname of container jenkins(4th line) which container jenkins/jenkins image.

mapping the hostport to docker container port for end user to access the portal

and then sharing the volumes into /home/admin/jenkins\_home

this shares all the files in /var/jenkins\_home of conatiner to /home/admin/jenkins\_home in host . Any changes made in any directory will reflect on to other

Networks section is creating a close network for all the containers that are being deployed or runned.

sshtesting

this container is created using a dockerfile which consists as below and resides in the directory Debian (context: Debian)

Not much to explain in above content, last 3 lines is creating a key in the docker container and starting the service and -D represents service to run in background/foreground try running it to know more

container\_name tag (it is what it is)

naming the image as ssht built by the dockerfile

and then creating all these in the same network as jenkins is on. So that each conatiners can communicate to each other. (try creating all the above steps with out using yml and follow below steps to understand more)

Execution:

create the files as mentioned at the start for the yml file to work properly.

Docker-compose.yml:

paste the content into the yml file mentioned in this document or else clone from git repo https://github.com/sasankkathera/udemy-Download.git

Jenkins\_home:

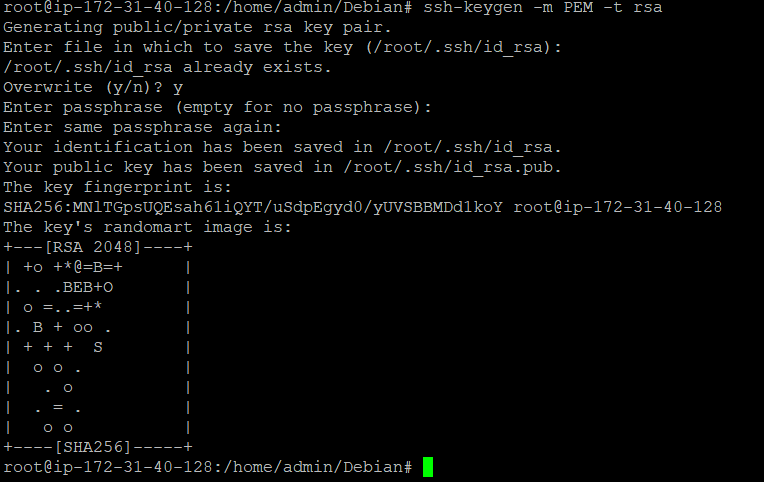
just create a directory and do not put any files into it.

Debian:

put dockerfile in this directory.

copying id\_rsa.pub in docker file ie., COPY id\_rsa.pub /home/ssh\_user/.ssh/authorized\_keys, if running on aws then you'll have to create a sshkey in RSA and then use for example.

NOTE: make sure to run this as root if you run this as an admin user then once you logout from the aws console then you wont be able to login unless you use the newly generated key.



**Using ssh-keygen -m PEM -t rsa will be explained later**

Copy the id\_rsa.pub (public key file) into /home/admin/Debian for the dockerfile to use it.

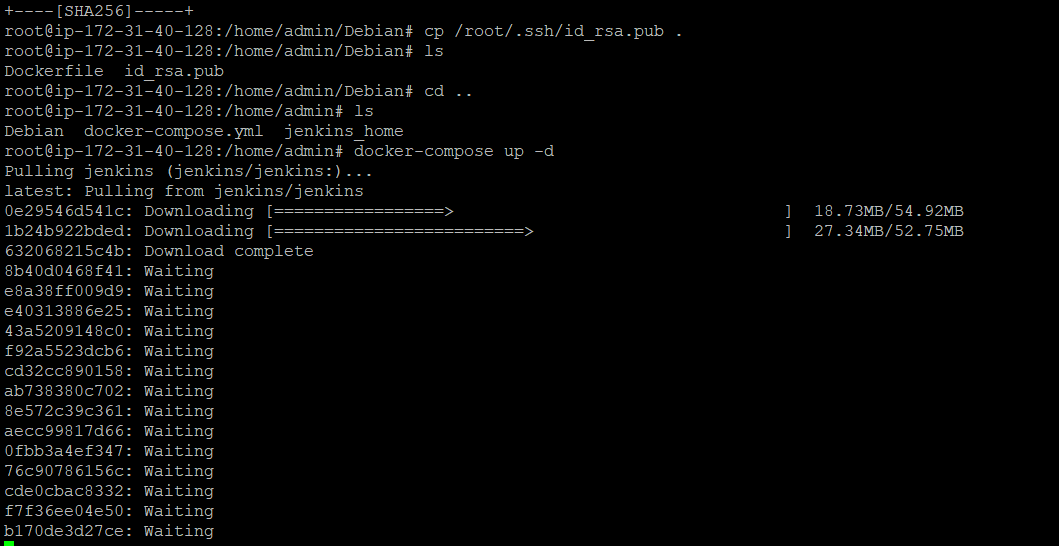
Once the setup is done we’ll have to use docker-compose command at the location of docker-compose.yml

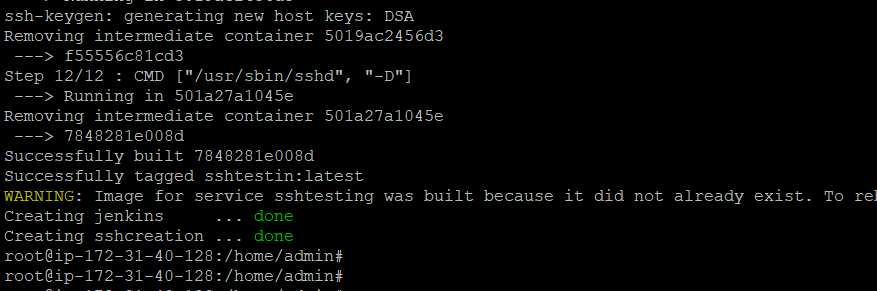
Command: docker-compose build --> to build images if there are any.

Command: docker-compose up -d --> to the yml file( it builds images and also creates containers)

NOTE: docker-compose.yml is the default name for docker-compose if you want to change the name of the file then you have use -f option and the file name example see below

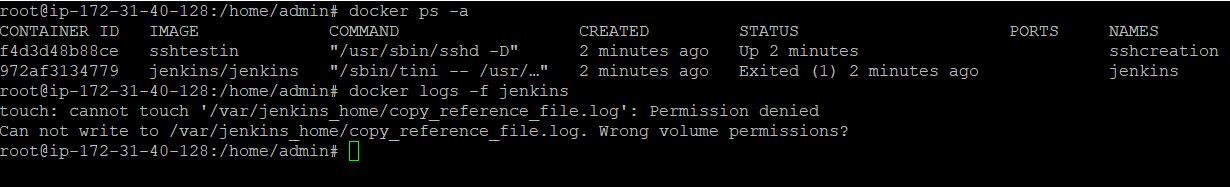
Command: docker-compose -f <file\_name>.yml up -d.



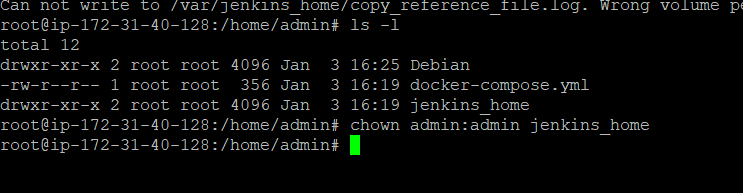


Once the above commands are executed accordingly then see the containers are created but if you see the running containers then only sshcreation container is up and running but Jenkins failed you can view the failed logs by using below command.

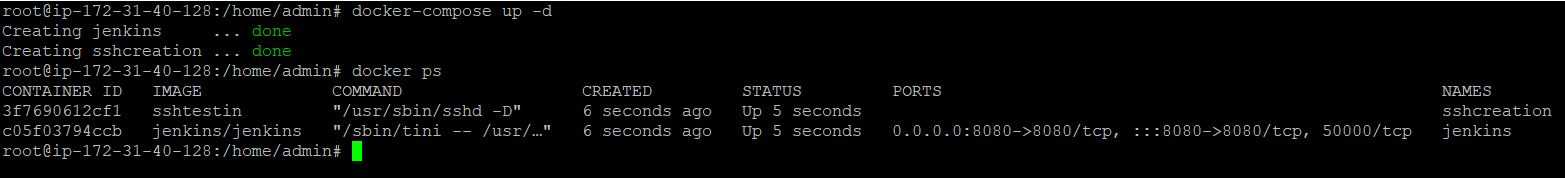
Command: docker logs -f (container\_name) Jenkins



To overcome this you have to change the ownership of Jenkins\_home file in admin directory

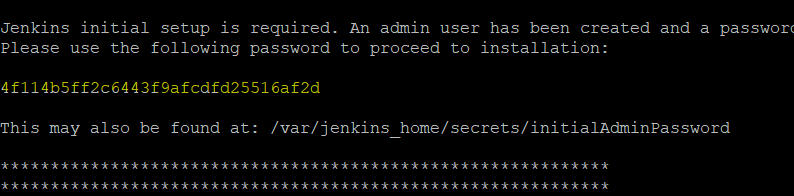


Then again use docker-compose after deleting the containers. Once that is done you should see Jenkins containers should be up and running.

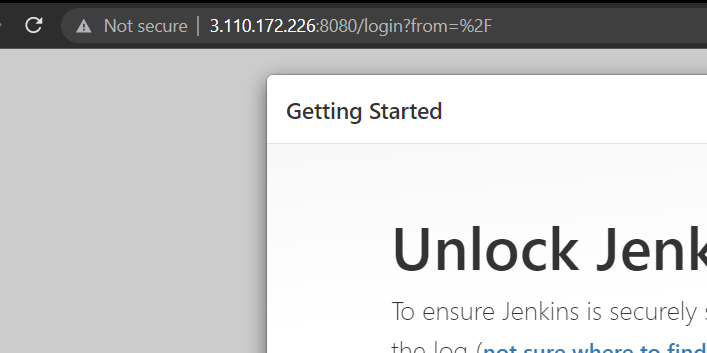


To view Jenkins portal logs ie., secret password you can use

Command: docker logs -f Jenkins

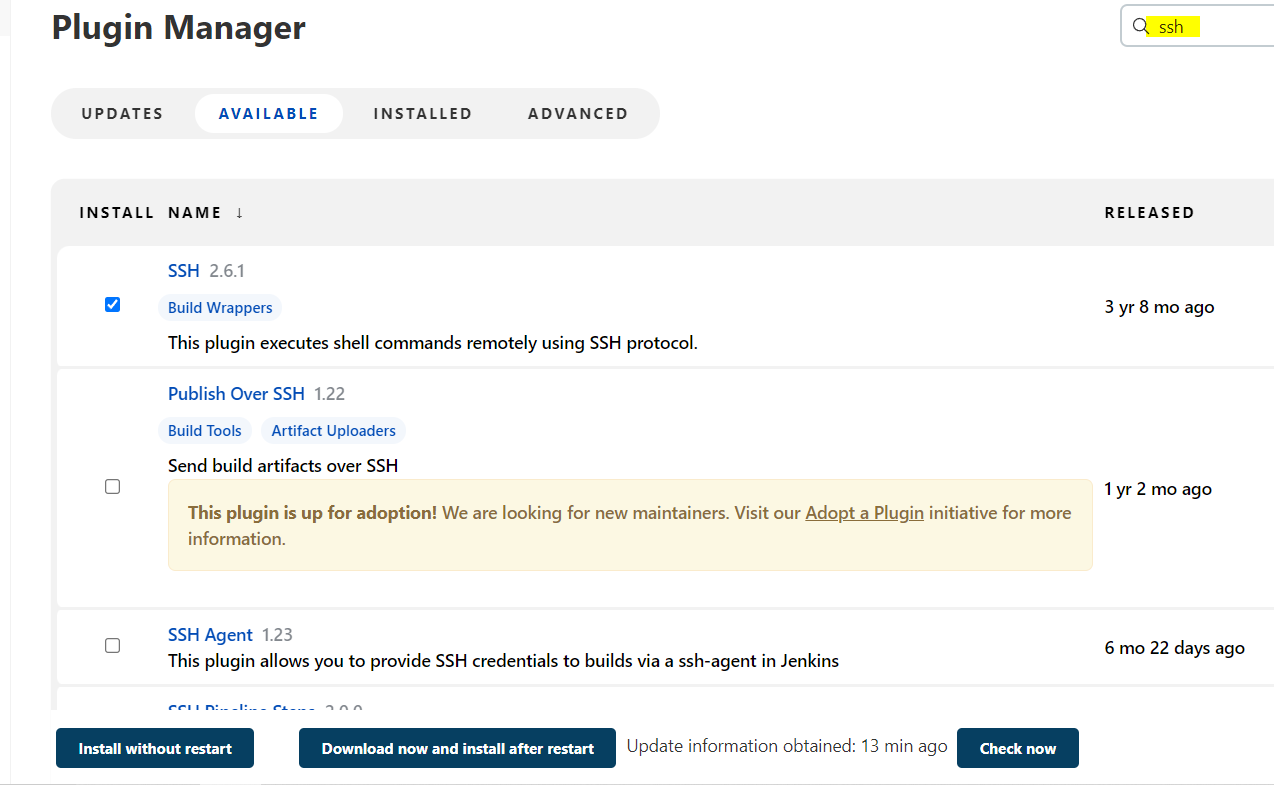


You can use your instance <ip\_addr>:8080 to access Jenkins portal running on Jenkins container.

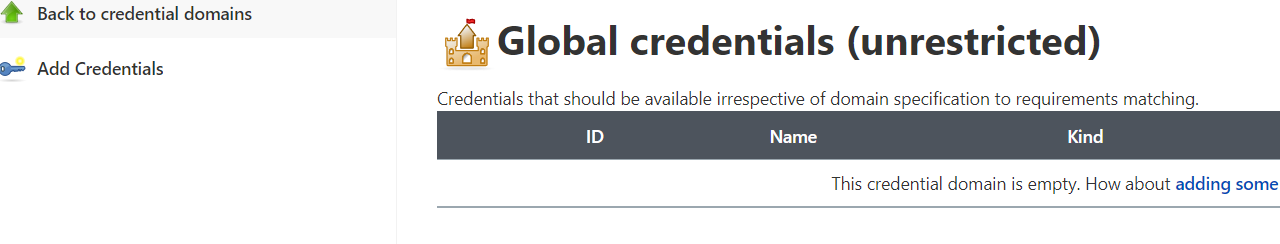


Paste the password and then procced by installing all the suggested plugins and setting up user and password if you want or else you would continue with admin with password: admin.

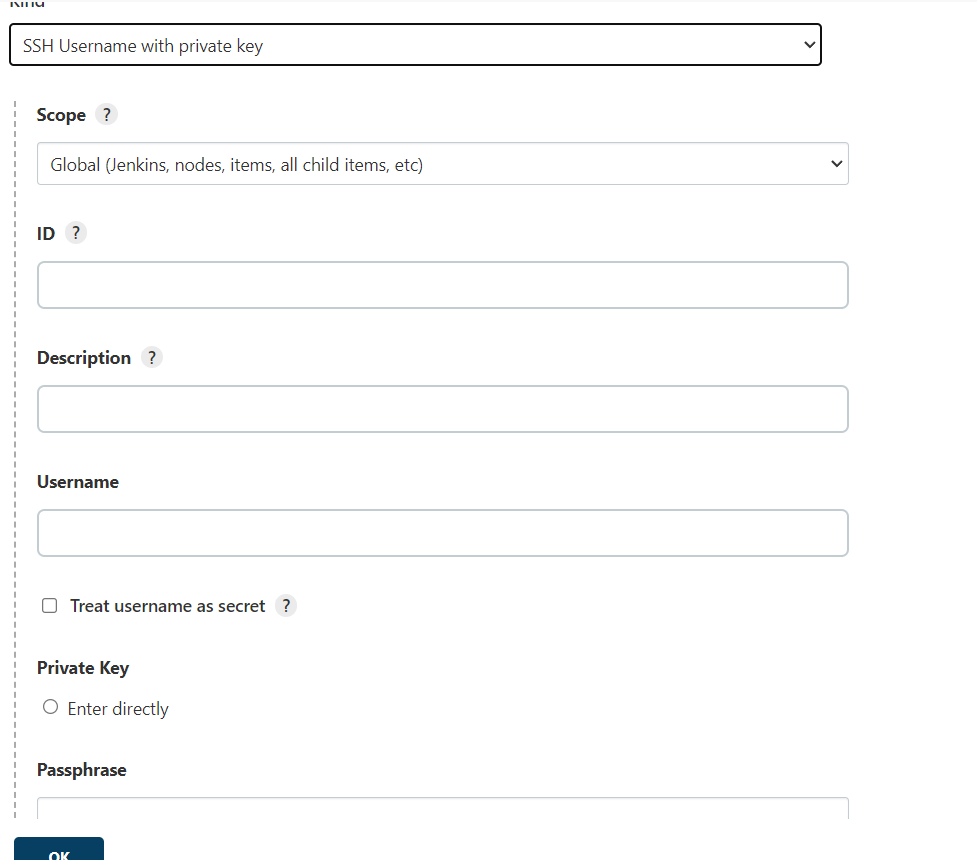
Once done install ssh plugin. Head to Manage Jenkins--> manage Plugins🡪 available

then click on install without restart

Set up our credentials to connect to remotehost (sshcreation container). Manager Jenkins🡪 manage credentials🡪 Jenkins🡪 globalcredentials🡪addcredentials



Select as below and paste id\_rsa (private key file )

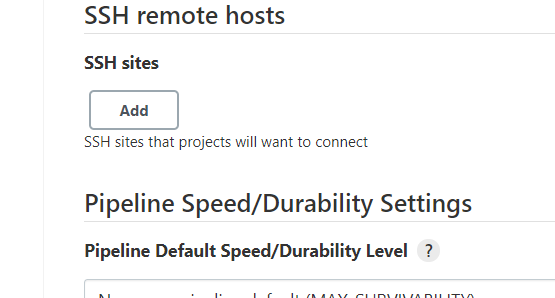


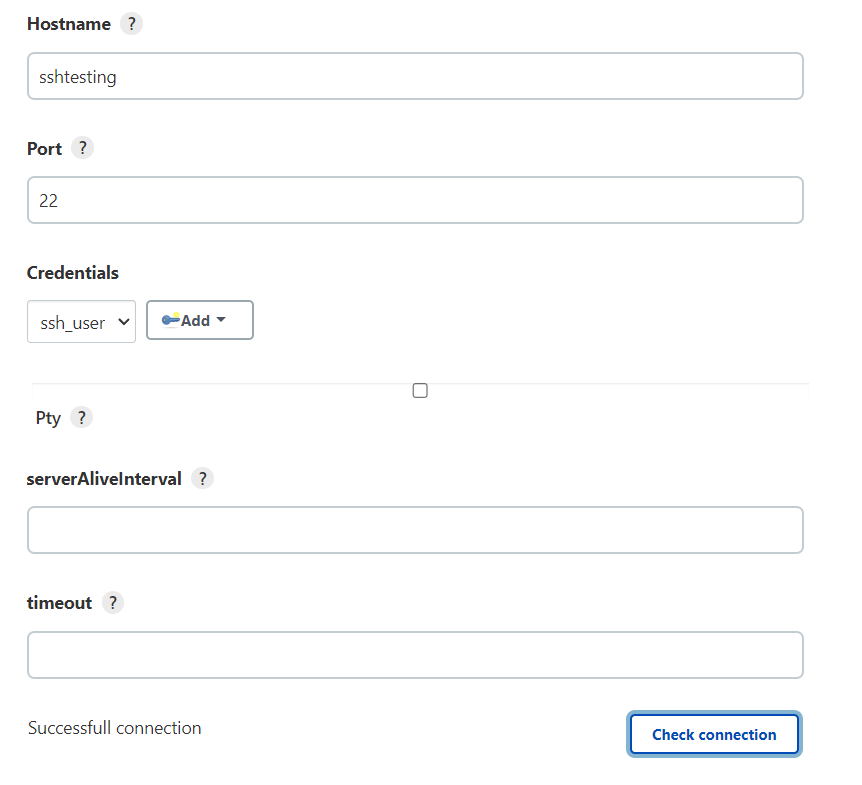
Username mentioned in dockerfile



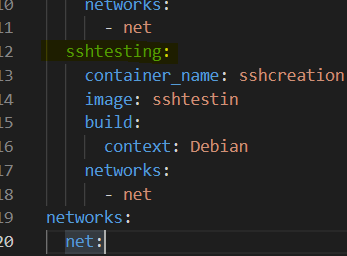
Head to: manage jenkins🡪 configure system

At ssh sites click on add

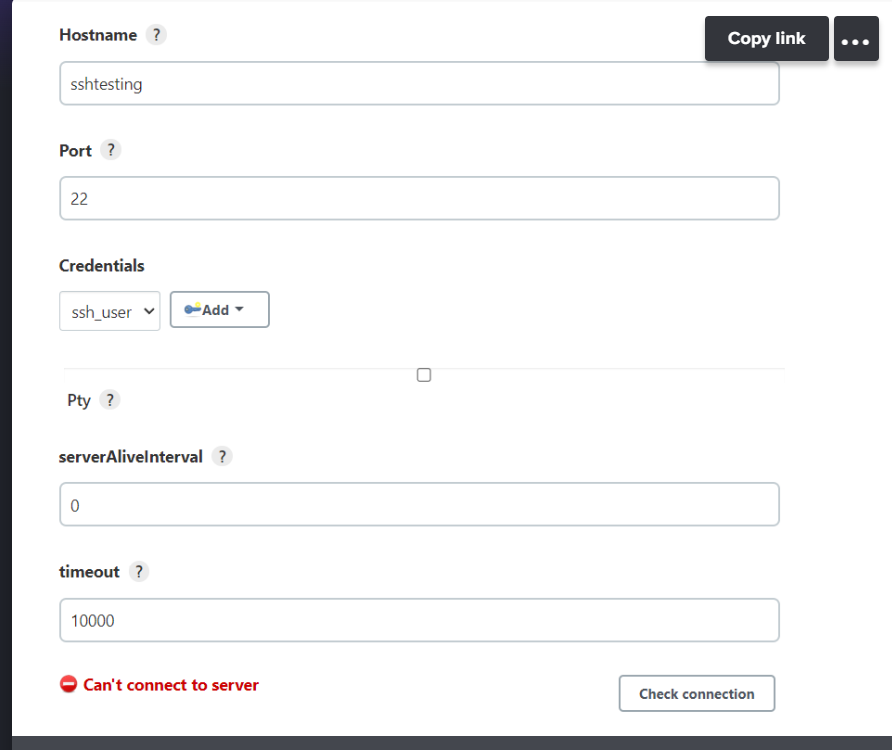


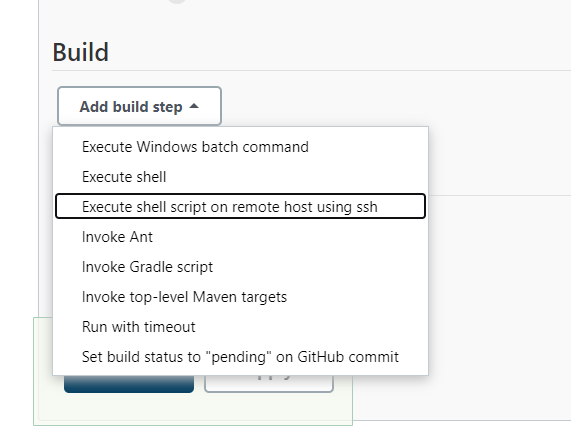


Here sshtesting is hostname mentioned in yml file



We used ssh-keygen -m PEM -t rsa at the time of creating key pairs is becase to create a pem file with rsa private key or else it would not connect



Once the connection is successfully established then you can create a free style project and at build section click on 

And write the commands that you want to execute in the remotehost container.

That’s it 😉