DevOps

Brief Introduction

This is a Step-by-step tutorial.
After this tutorial, you will be able to:
1. Create a docker server on your server.
2. Deploy NodeJs application to Docker container via Jenkins.
3. Auto deploy application when you push code to your git repo.

Prerequisites

1. A server: Local pc (for practice) or VPS
2. Basic knowledge of Docker & Jenkins
3. Linux command (basic)

Install

Install Docker

You could skip this step if you have already have the docker server installed in your server. Or you could uninstall the old one, then follow this guide.

You could check the installation guide @ Docker Document

As most of you use CentOS, I will introduce the step for install docker on CentOS, for other system, please Kindly check above Docker document link.

There is two ways to install Docker on CentOS,
1. Install from repository
2. Install from a package
Here we use the first way to do it.

Firstly, we need to set up the repository:
1. Install required packages:
   sudo yum install -y yum-utils \ device-mapper-persistent-data \ lvm2

2. Set up the stable repository:
   sudo yum-config-manager \ --add-repo \ https://download.docker.com/linux/centos/docker-ce.repo

Then, install Docker CE
   sudo yum install docker-ce

Finally, Startup Docker
   sudo systemctl start docker

Install & Run Jenkins

Firstly, let's install the jenkins docker image
   sudo docker pull jenkins

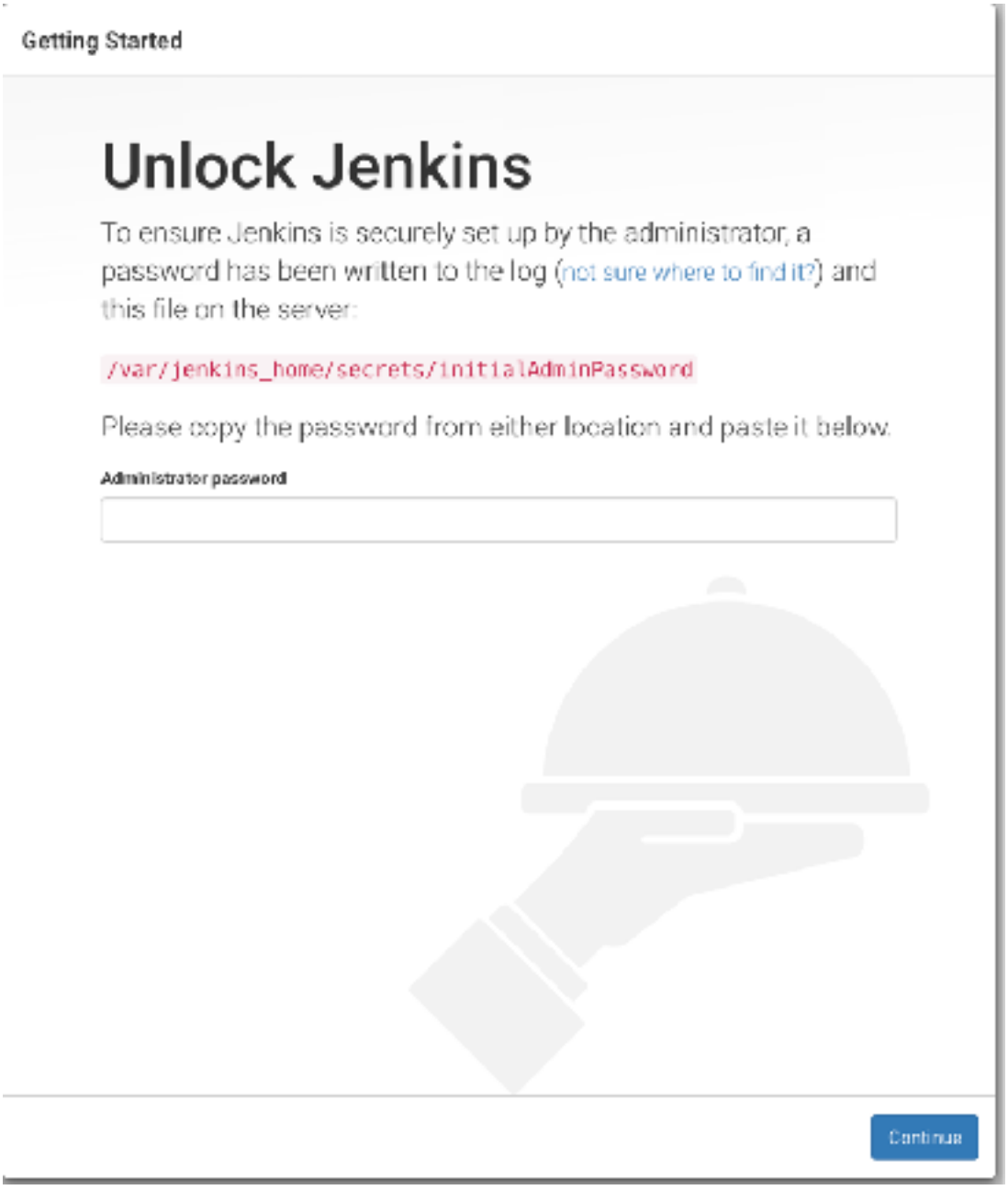
Then, we need to create a jenkins work directory
   mkdir /var/jenkins_node

We also need to grant the operation access to the jenkins user
   sudo chown -R 1000 /var/jenkins_node

Finally, you should start up a jenkins container via jenkins images
docker run -d --privileged=true --name myjenkins -p 49001:8080 -v /var/jenkins_node:/var/jenkins_home jenkins

After this, you could use docker ps -a to check whether your jenkins container is up or not,
If not, you could use docker logs <containerID> to check the error log

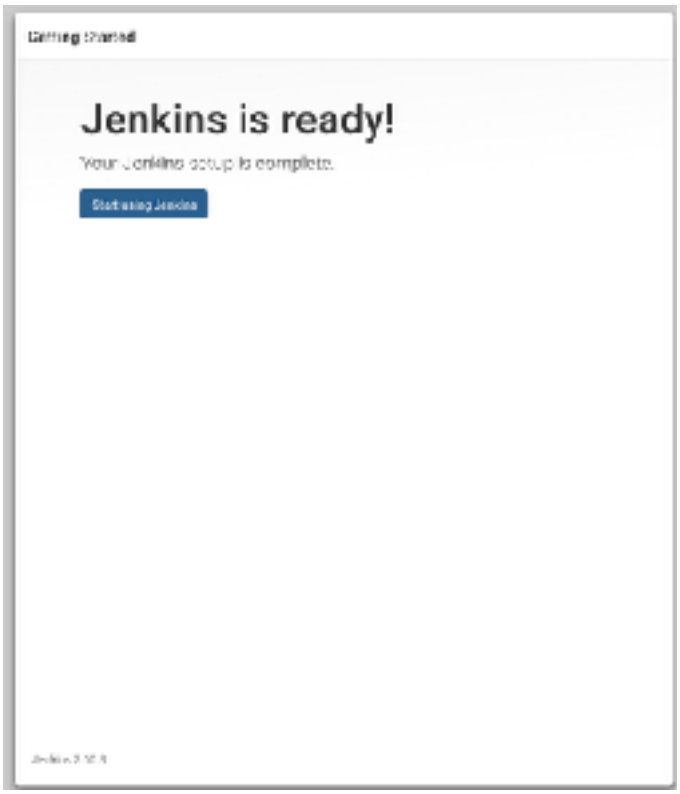
As a result, when you visit http://<your server ip>:49001, you will get below screen



Then, let's find the initial password and fill it it.
cat /var/jenkins_node/secrets/initialAdminPassword

And we could install the suggested plugins

You should create a first admin account, then the installation of jenkins is done.

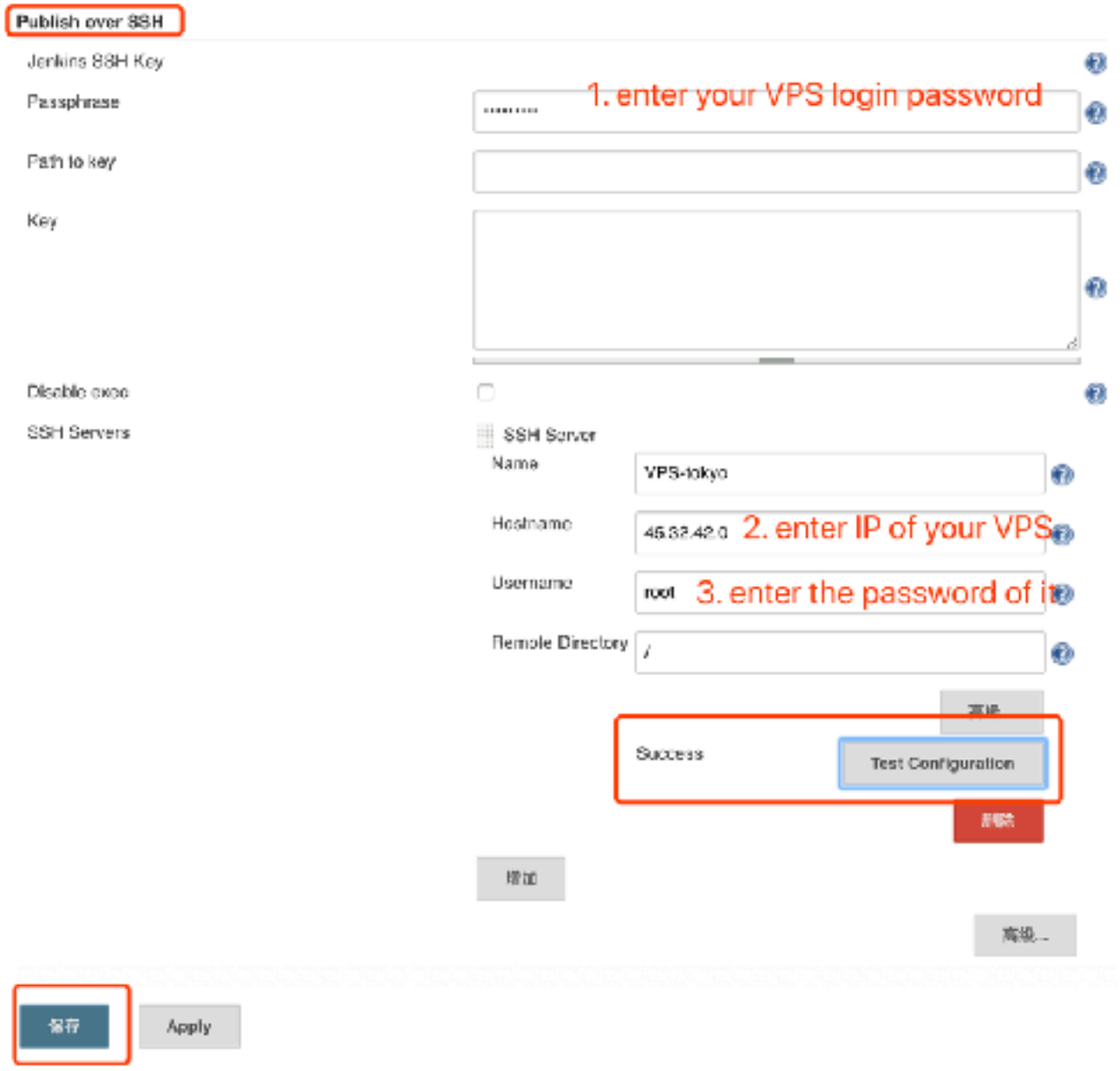


Connect your VPS server via jenkins over ssh

Firstly we should install the ‘Publish over SSH’ plugin in jenkins

Once the plugin is done for the installation, the jenkins will restart.
Then let’s try to connect to your VPS via the plugin

If you got success after click ‘test configuration’, it means it could reach your VPS.
Then, click ‘save’



Install basic docker image for NodeJs

Since it’s a NodeJs project, so it’s necessary to install a basic NodeJs image via jenkins.
This is the test git repo: https://github.com/MeatPieYan/BasicNodeJsDockerImage

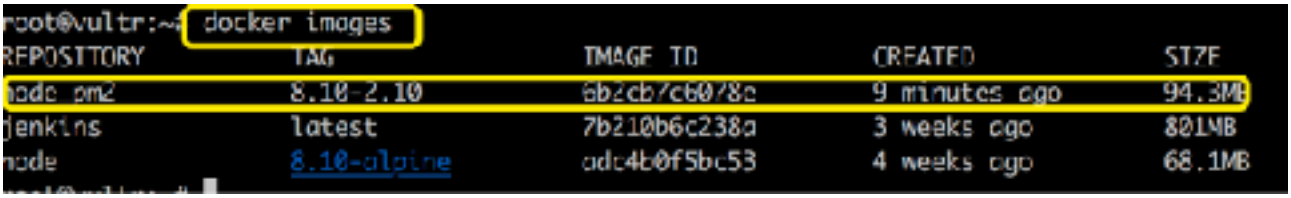
<== enter Git related info

<== the script is as below

```
cd /var/jenkins_node/
workspace/NodeImageBasic
\
 && docker build --rm --no-
cache=true  -t
node_pm2:8.10-2.10  - <
Dockerfile
```

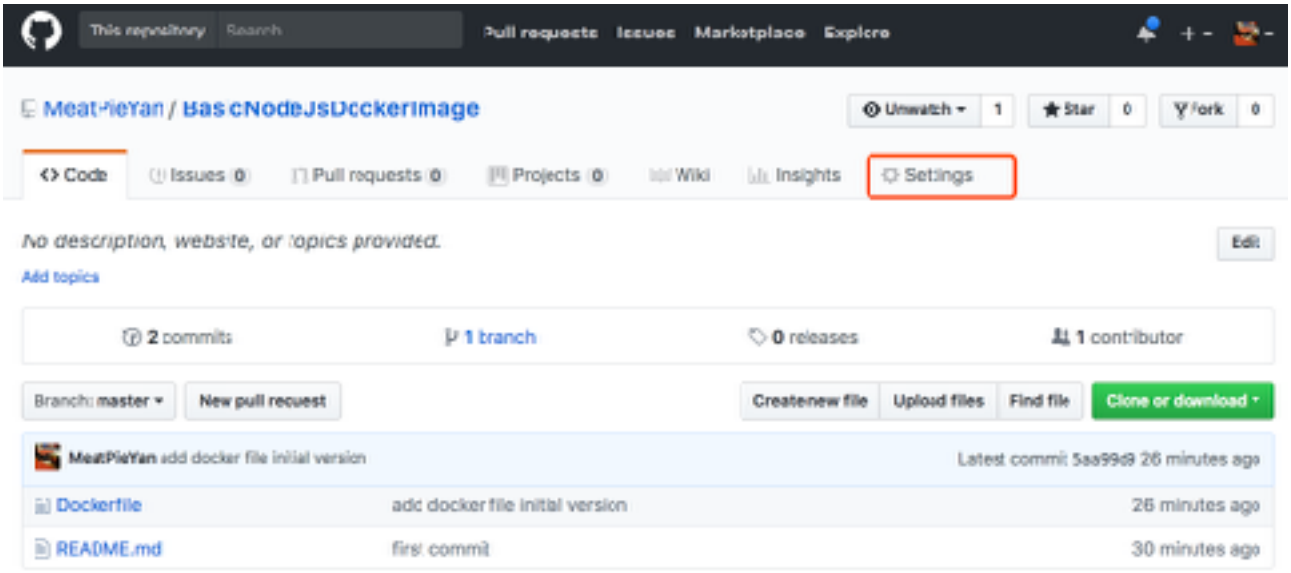
After save it, you could try to run it.

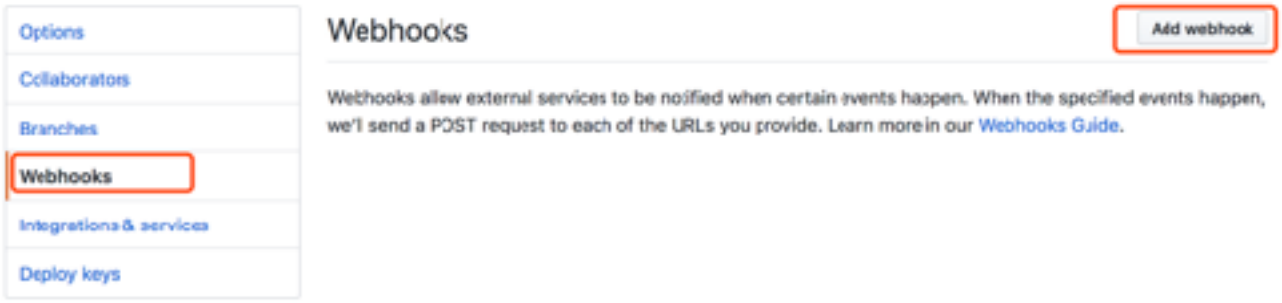
Once the build is successful, we could check whether we have create a basic image in docker
If build status is unstable of failure, you could check the console of this build.

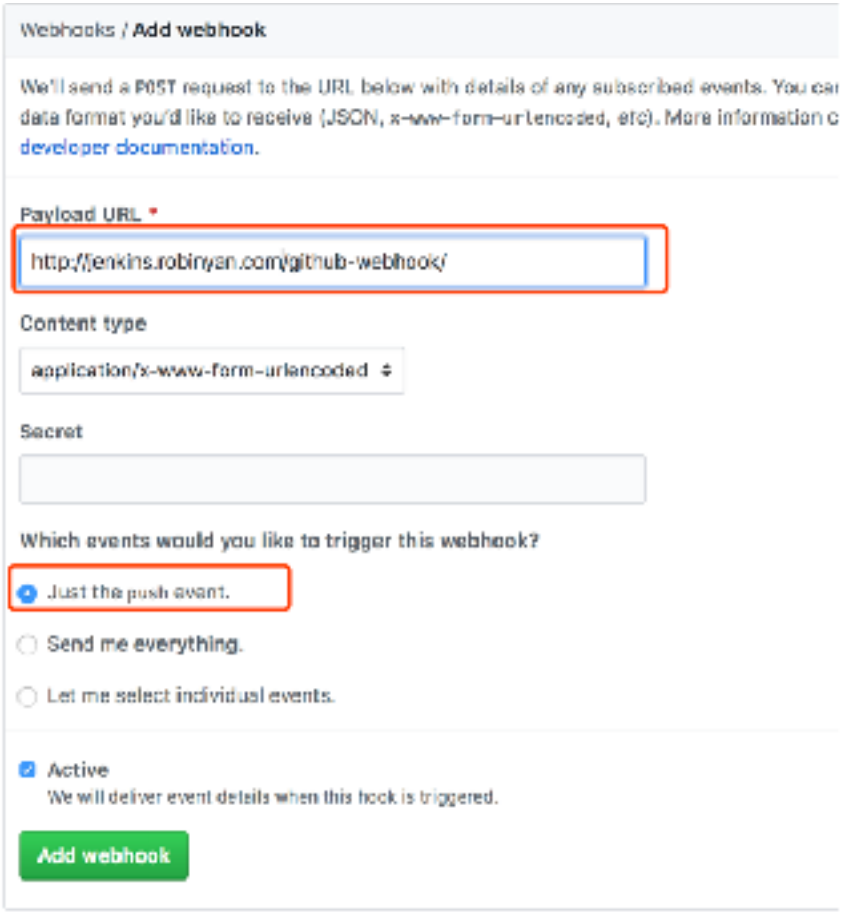


Install basic docker image for NodeJs automatically

As of now, we have done image creation, but if we want to update our basic image, how shall we do it. We could update the dockerfile in the git repo, and trigger the jenkins task again.
But can we do it automatically? The answer is yes.





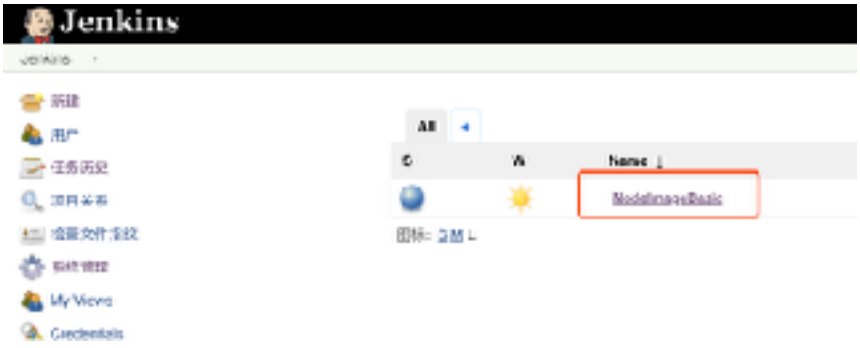


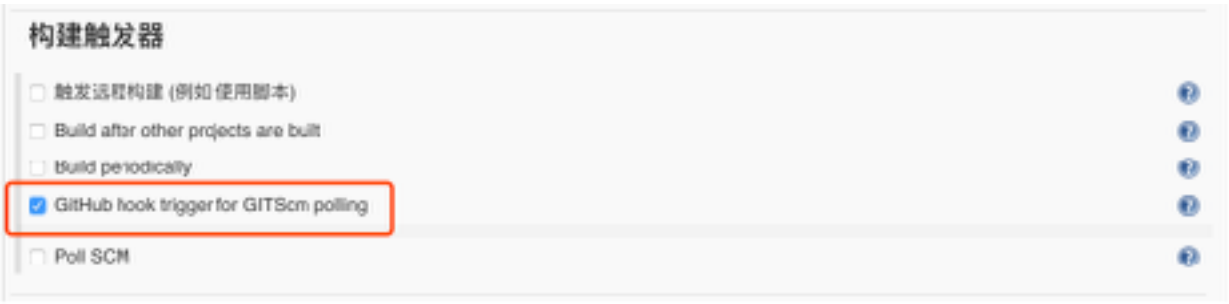
<== enter the url like this:

http://<you vps id>:<jenkins port>/github-webhook

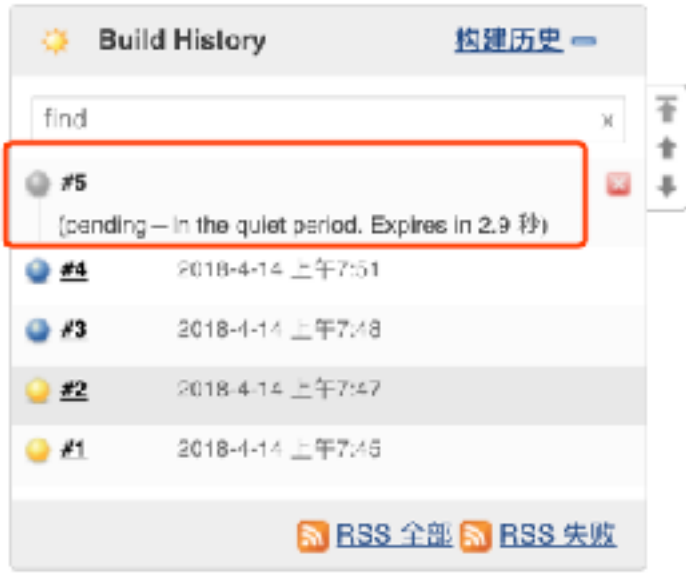
If you have a domain like me, you could also enter the domain instead of ip:port

For jenkins, we need to do some change to make it effective.





let's have a try with it
once you push any update into your git repo, a new jenkins build will be triggered.



Seems everything is ok, then let's build our project.

Project deployment

Brief introduction

After this section, we need to deploy the project into docker container

Create Jenkins task

Everything of this task is same like the one to build basic docker image except the script, below is the script. For git web hook, also the same.

```
docker stop container || true \
    && docker rm container || true \
    && cd /var/jenkins_node/workspace/Node_Test \
    && ls \
    && docker build --rm --no-cache=true  -t node_project_img  . \
    && docker run -d  --name container -p 3000:3000 -v /var/jenkins_node/workspace/Node_Test:/
home/project  node_project_img
```


General
Node_Test
GitHub project
Throttle builds
源码管理
None
Git
Repositories
Repository URL
Credentials
Add
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch
Additional Behaviours
Add
Subversion
构建触发器
Build after other projects are built
Build periodically
Poll SCM
构建环境
Delete workspace before build starts
Use secret text(s) or file(s)
Send files or execute commands over SSH after the build runs
SSH Publishers
SSH Server
Name
Transfers
Transfer Set
Source files
Remove prefix
Remote directory
Exec command
docker stop container && true \
&& docker rm container && true \
&& cd /var/jenkins_code/workspace/Node_Test \
&& ls \
&& docker build -t
Either Source files, Exec command or both must be supplied
Add Transfer Set
Add Server
Abort the build if it's stuck
Add timestamps to the Console Output
With Ant
构建
构建后操作
Save
Apply