# 一、tomcat 自动化发布：

## 1.客户端环境准备:

### （1）.java安装

JDK1.8.161（java默认安装在/usr/java/jdk1.8.0\_161/）

编辑全局环境变量文件/etc/profile 末尾添加

#set java environment

JAVA\_HOME=/usr/java/jdk1.8.0\_161

JRE\_HOME=$JAVA\_HOME/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

### （2）.tomcat安装

Tomcat 8.5.30（tomcat 默认安装在/usr/apache-tomcat-8.5.30/）

以上搭建步骤略过

## 2.新建jenkins用户

新建ansible 本地操作用户jenkins

useradd jenkins

passwd jenkins

更改tomcat 属主为 jenkins 用户

chown -R jenkins:root /usr/apache-tomcat-8.5.30/ （本例的tomcat部署再/usr下面的目录）

## 3.将tomcat配置成systemd启动

由于ansible需要用systemd模块才能操作进程的启停，所以我们需要配置tomcat成service，

sudo vim /etc/systemd/system/tomcat.service

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

[Unit]

Description=Apache Tomcat Web Application Container

After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/java/jdk1.8.0\_161

Environment=CATALINA\_PID=/usr/apache-tomcat-8.5.30/temp/tomcat.pid

Environment=CATALINA\_HOME=/usr/apache-tomcat-8.5.30

Environment=CATALINA\_BASE=/usr/apache-tomcat-8.5.30

Environment='CATALINA\_OPTS=-Xms2048M -Xmx2048M -server -XX:+UseParallelGC'

#Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecReload=/bin/kill -s HUP $MAINPID

ExecStop=/bin/kill -s QUIT $MAINPID

PrivateTmp=true

ExecStart=/usr/apache-tomcat-8.5.30/bin/catalina.sh start

[Install]

WantedBy=multi-user.target

重新加载systemctl配置

systemctl daemon-reload

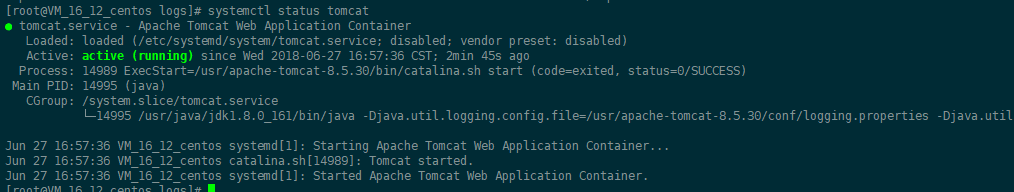
关闭当前正在运行的tomcat

ps -ef|grep tomcat|grep -v grep|awk '{print $2}' |xargs kill -9

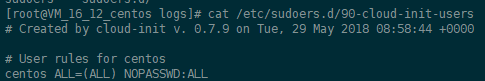
使用systemctl 来测试启动和关闭是否可用。

systemctl start tomcat.service

查看状态



修改sudoer.d文件夹下面的 90-cloud-init-users (默认腾讯云自带该文件)文件，添加 jenkins可以以root权限运行的命令。



打开写入权限

chmod 640 /etc/sudoers.d/90-cloud-init-users

在文件末尾处添加

jenkins ALL=(ALL) NOPASSWD: ALL

将权限修改回来

chmod 440 /etc/sudoers.d/90-cloud-init-users

## 4.配置SSHD服务，打开公钥免密登陆

修改sshd配置文件

vim /etc/ssh/sshd\_config



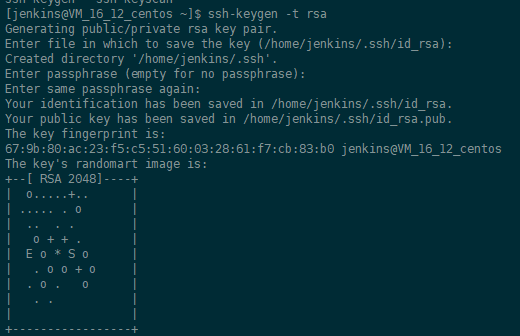
重新启动sshd

systemctl restart sshd

切换成jenkins用户，建立自己的密钥对

su – jenkins

ssh-keygen -t rsa (创建过程中回车即可)



## 5.服务端配置：

备注：以下设置使用的是jenkins用户登录进行配置。

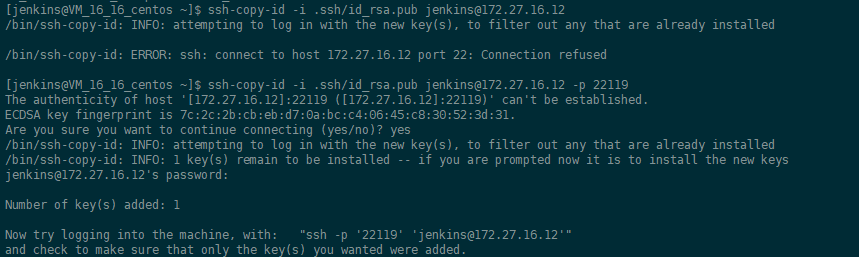
### （1）服务端公钥免密登陆。

登陆jenkins-ansible管理平台服务器，并将jenkins用户的公钥发送到客户端服务器。

ssh-copy-id -i .ssh/id\_rsa.pub jenkins@172.27.16.12 -p 22119

备注:172.27.16.12为客户端服务器IP地址

运行结果如下：



测试公钥登陆是否成功。

ssh -p 22119 [jenkins@172.27.16.12](mailto:jenkins@172.27.16.12)

### （2）安装maven

本例使用maven进行打包，所以需要在服务端安装maven工具。

wget <http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz>

解压到/home/jenkins目录

tar zxvfpP apache-maven-3.5.4-bin/tar.gz -C /home/jenkins

编辑jenkins用户的个人环境变量添加

vim ~/.bashrc

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

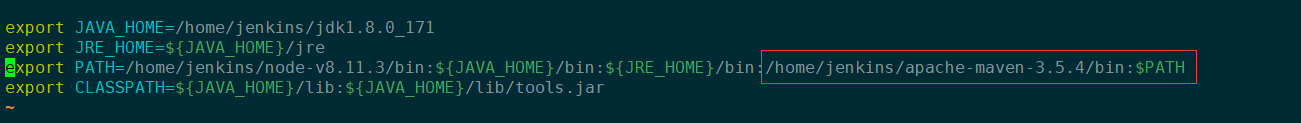
export JAVA\_HOME=/home/jenkins/jdk1.8.0\_171

export JRE\_HOME=${JAVA\_HOME}/jre

export PATH=/home/jenkins/node-v8.11.3/bin:${JAVA\_HOME}/bin:${JRE\_HOME}/bin:/home/jenkins/apache-maven-3.5.4/bin:$PATH

export CLASSPATH=${JAVA\_HOME}/lib:${JAVA\_HOME}/lib/tools.jar

如下图所示



source ~/.bashrc

mvn -v测试是否安装成功。

### （3）配置ansbile

在/etc/ansible/hosts文件中，配置clinic\_user组。

vim /etc/ansible/hosts

并且添加ansible\_ssh\_port,ansible\_ssh\_user

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

[clinic\_user]

172.27.16.12 ansible\_ssh\_port=22119 ansible\_ssh\_user=jenkins

### （4）编写playbook

vim /home/jenkins/playbook/clinic\_user\_12.yml

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

---

- hosts: clinic\_user

vars:

war\_file: /home/jenkins/.jenkins/workspace/clinic\_user/clinic-user/target/clinic-user.war

tomcat\_path: /usr/apache-tomcat-8.5.30/webapps

home\_path: /home/jenkins

tasks:

- name: stop tomcat

systemd:

name: tomcat

state: stopped

- name: general date variable

shell: date +%Y\_%m\_%d\_%H

register: date\_time

ignore\_errors: True

- name: archive old tomcat webapps

archive:

path: "{{ tomcat\_path }}/clinic\_user"

dest: "/home/jenkins/clinic\_user\_{{ date\_time.stdout }}.tgz"

remove: yes

- name: copy war\_file to dest server.

copy:

src: "{{ war\_file }}"

dest: "{{ home\_path }}"

- name: create clinic\_file direcoty, if exist exit this step.

file:

path: "{{ tomcat\_path }}/clinic\_user"

state: directory

mode: 0755

owner: jenkins

group: jenkins

- name: unarchive war\_file to tomcat webapps!!!

unarchive:

remote\_src: yes

src: "{{ home\_path }}/clinic-user.war"

dest: "{{ tomcat\_path }}/clinic\_user"

owner: jenkins

group: jenkins

mode: 0755

- name: startup tomcat

systemd:

name: tomcat

state: restarted

- name: create a variables to restore tomcat server status

shell: ps -ef|grep tomcat|grep -v grep

register: tomcat\_status

ignore\_errors: True

- name: check tomcat status

debug: msg="The tomcat\_status is {{ tomcat\_status.stdout }}"

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

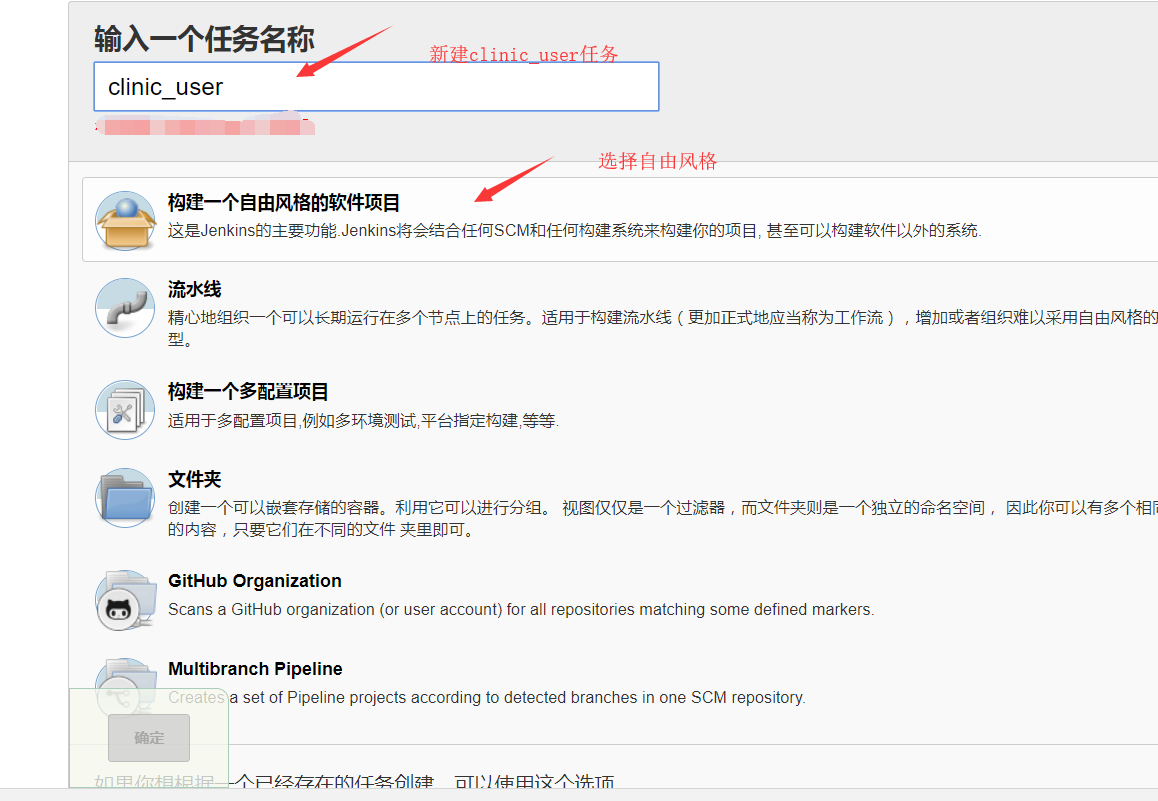
### （5）运行playbook

ansible-playbook /home/jenkins/clinic\_user\_12.yml -s

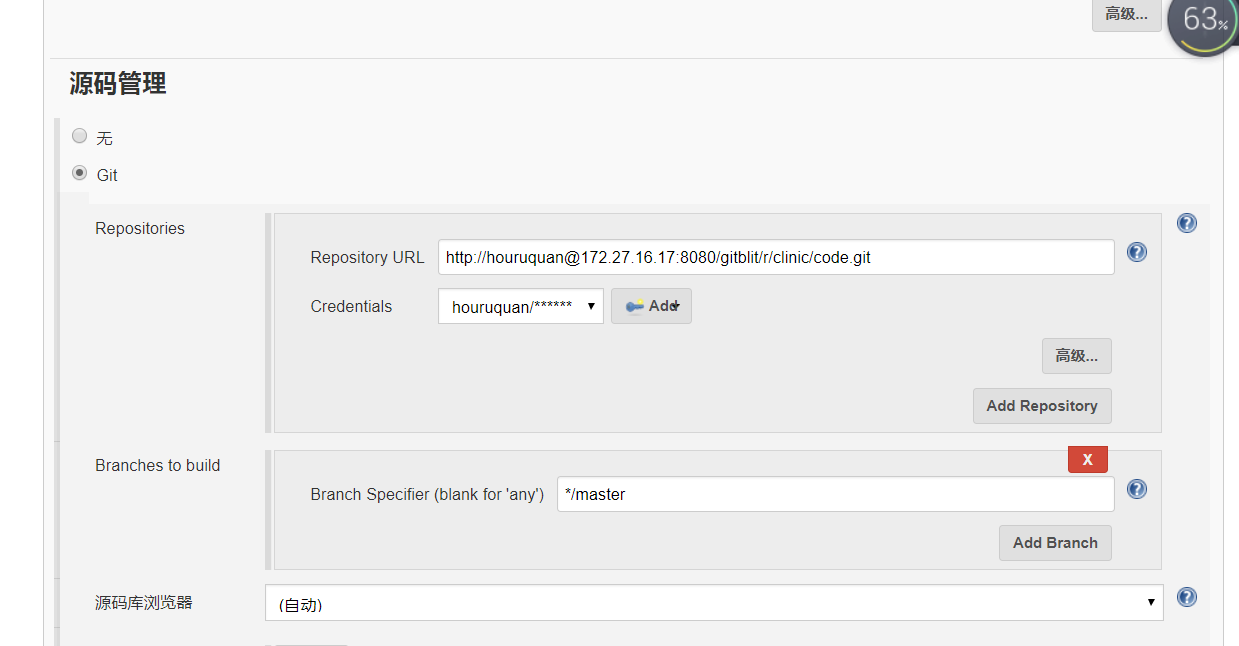


## 6、jenkins页面配置

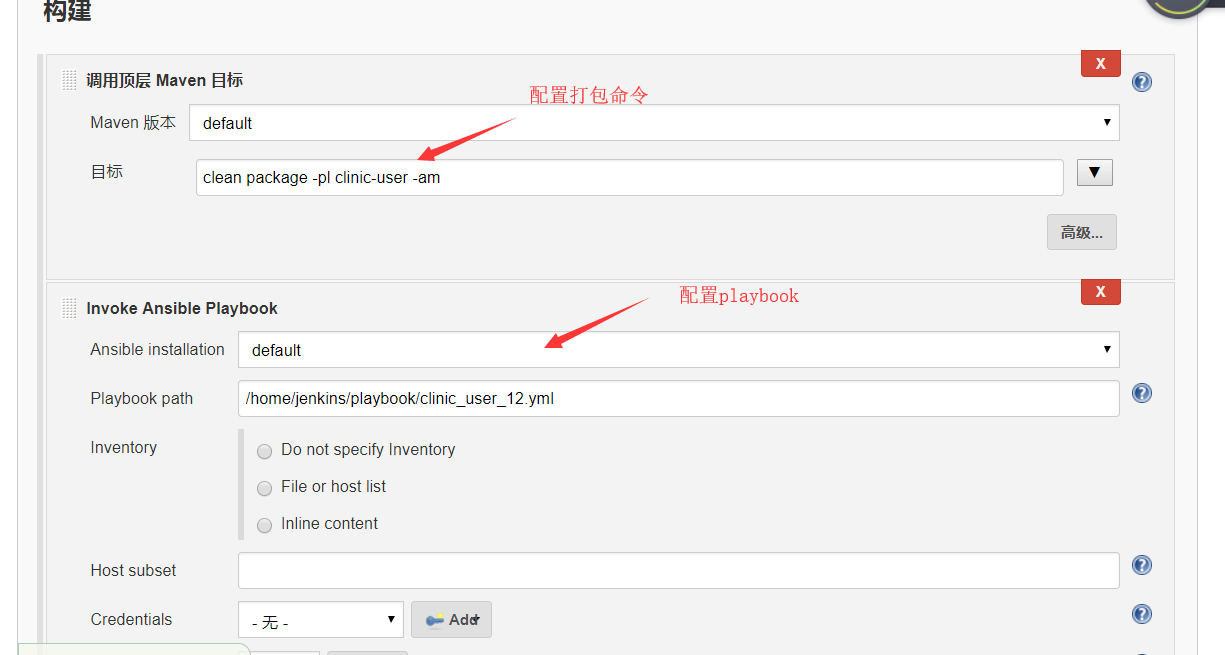
新建项目



配置项目源码



配置打包命令，playbook



保存

返回页面主页，点击构建项目，测试成功。



# 二、nginx 自动化发布：

## 1.客户端环境准备：

系统环境和nginx版本：

Centos 7.2

Nginx 1.14.0

### 下载并安装nginx

下载nginx源码包

wget <http://nginx.org/download/nginx-1.14.0.tar.gz>

tar zxvf nginx-1.14.0.tar.gz

mv nginx-1.14.0/ usr/local/nginx

## 2 配置systemctl 服务启动 nginx

注意：nginx不指定 PID文件位置，不适用/bin/kill HUP $MAINPID 进行重新启动和停止服务。不然会提示报错,提示 PID文件没有可读权限。适用以下配置即可。

编辑

vim /usr/lib/systemd/system/nginx.service

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

[Unit]

Description=The NGINX HTTP and reverse proxy server

After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

ExecStartPre=/usr/local/nginx/sbin/nginx -t -c /usr/local/nginx/conf/nginx.conf

ExecStart=/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

ExecReload=/usr/local/nginx/sbin/nginx -s reload

ExecStop=/usr/local/nginx/sbin/nginx -s stop

PrivateTmp=true

[Install]

WantedBy=multi-user.target

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

修改以后记得

systemctl daemon-reload

使用systemctl测试服务是否配置成功

systemctl status nginx

systemctl start nginx

## 3.新建jenkins用户

新建jenkins用户，并设置密码（同1-（1））

配置jenkins服务器免密登录(同2-（1）)

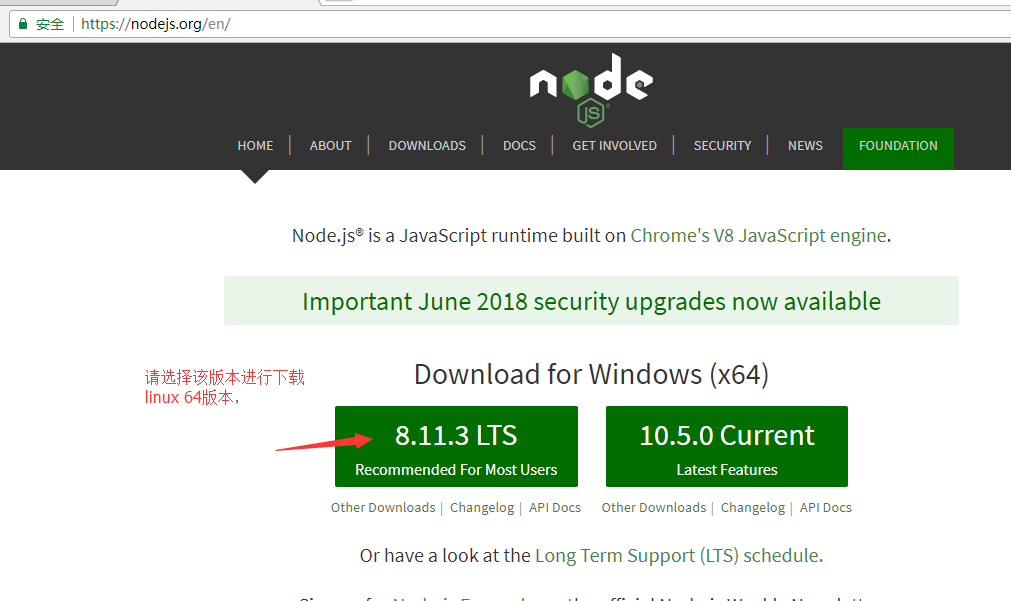
## 4.服务端配置：

### （1）node,npm安装

本例nginx 主要作为静态资源文件服务器，主要部署的是前端的代码如js,css,html,img等。

发布中使用了npm打包工具，

下载nodejs （包含npm打包工具）



下载的包是tar.xz后缀，需要安装xz 进行解压

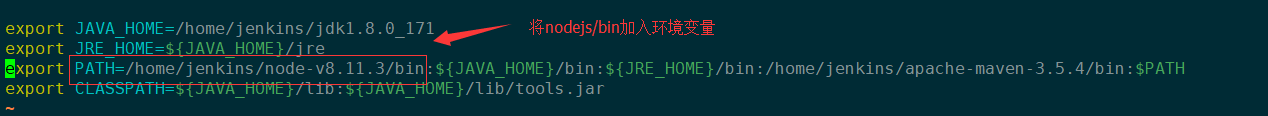
yum install xz

xz -d node-v8.11.3-linux-x64.tar.xz



配置当前jenkins 用户的环境变量。

vim ~/.bashrc



加载环境变量

source ~/.bashrc

检查版本信息

npm -v

node -v

安装淘宝镜像

npm install -g cnpm --registry=https://registry.npm.taobao.org

可以使用cnpm命令，关联淘宝镜像进行依赖下载和打包。

### （2）编写打包脚本

创建存放目录

mkdir -p /home/jenkins/bash

编辑脚本

vim /home/jenkins/bash/npm\_package.bash

#################################################################

!/bin/bash

project\_path=/home/jenkins/.jenkins/workspace/web\_static/sh-hospital

source /home/jenkins/.bashrc

cd "$project\_path"

cnpm install

cnpm install font-awesome

cnpm run build

if [ $? -ne 0 ];then

echo "building packages has some problem,please check"

exit 55

else

echo "buiding success"

fi

if [ -d ${project\_path}/dist ];then

echo "dist directory success"

fi

##################################################################

### (3)利用jenkins拉取项目源码。

登陆jenkins web页面，

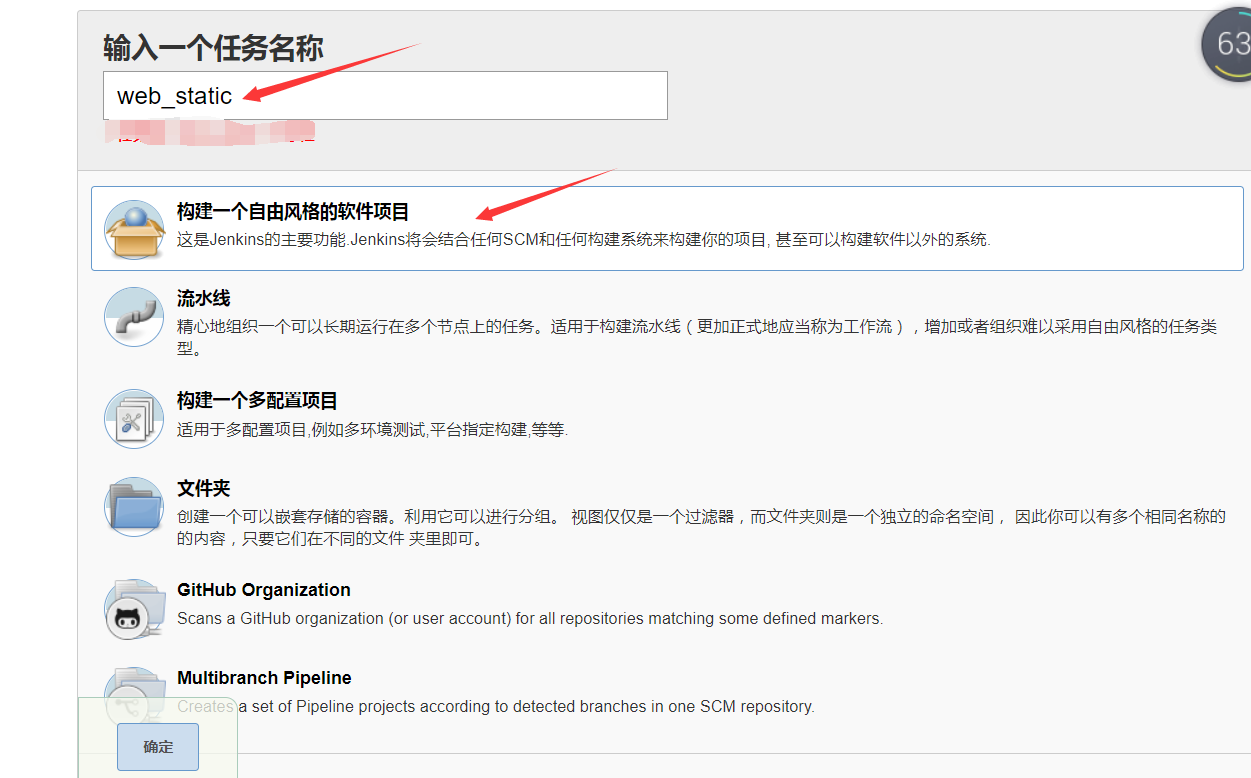
http://172.27.16.16:8080

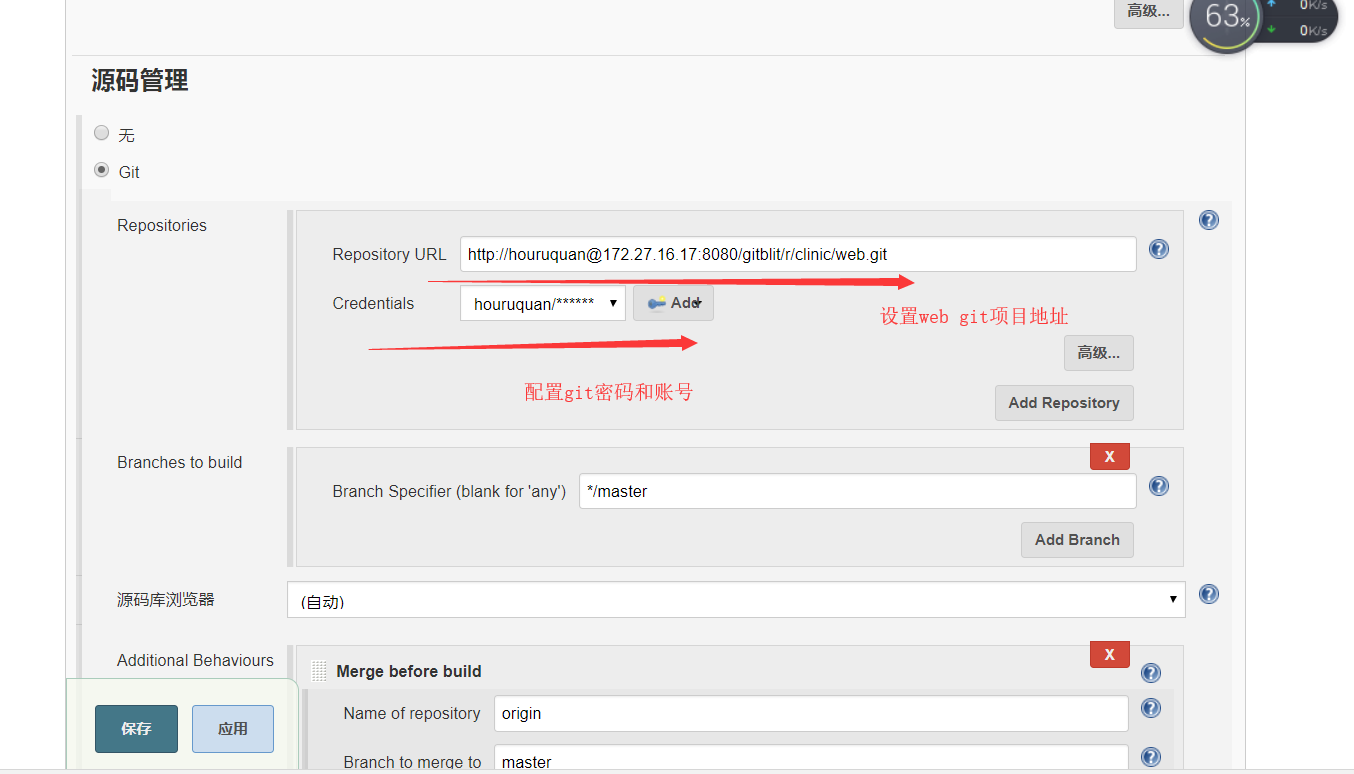


新建任务

输入任务名称web\_static

选择 自由风格的软件项目，并确认



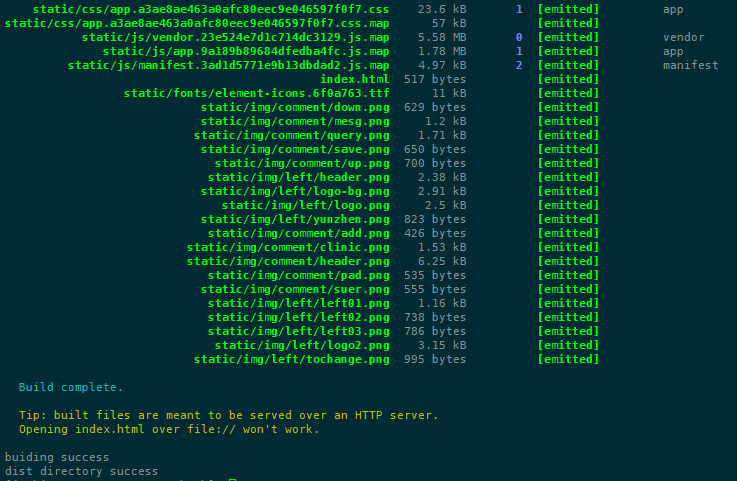


按照以上设置，并保存。执行构建，将会在本地/home/jenkins/.jenkins/workspace/web\_static/

拉取web前段项目的代码

### （4）测试打包脚本

/bin/bash /home/jenkins/bash/npm\_package.bash



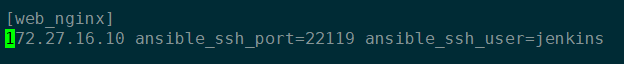
注意：如果这里报错不成功，请和前端沟通。

默认打包文件在/home/.jenkins/workspace/项目名/sh-hispital/dist/下面

### (5)配置ansible

首先在 inventory文件 /etc/hosts添加定义组。

添加内容如下：由于我们远程ssh不使用默认的22端口这里修改成22119



定义客户端 playbook需要使用的变量

新建文件夹

mkdir -p /etc/ansible/host\_vars

vim /etc/ansible/host\_vars/172.27.16.10

##################################################################

---

nginx\_path: "/usr/local/nginx/html"

package\_file: "/home/jenkins/.jenkins/workspace/web\_static/sh-hospital/dist"

files:

field1: static

field2: index.html

##################################################################

### （6）编写playbook

vim /home/jenkins/playbook/web\_static\_10.yml

---

- hosts:

web\_nginx

tasks:

- name: stop nginx

systemd:

name: nginx

state: stopped

- name: general date variable

shell: date +%Y\_%m\_%d\_%H

register: date\_time

- name: debug variables

debug: msg="echo nginx\_path variables {{ nginx\_path }} echo package\_file variables {{ package\_file }}"

- name: tar old nginx static directory

archive:

path:

- "{{ nginx\_path }}/{{ files['field1'] }}"

- "{{ nginx\_path }}/{{ files['field2'] }}"

dest: "/home/jenkins/nginx\_file\_{{ date\_time.stdout }}.tgz"

remove: yes

- name: copy nginx\_file to dest server.

copy:

src: "{{ item }}"

dest: "{{ nginx\_path }}"

with\_items:

- "{{ package\_file }}/{{ files['field1'] }}"

- "{{ package\_file }}/{{ files['field2'] }}"

- name: startup nginx

systemd:

name: nginx

state: restarted

- name: create a variables to restore nginx server status

shell: ps -ef|grep nginx|grep -v grep

register: nginx\_status

ignore\_errors: True

- name: check nginx status

debug: msg="The nginx\_status is {{ nginx\_status.stdout }}"

### （7）运行playbook

ansible-playbook web\_static\_10.yml -s –check

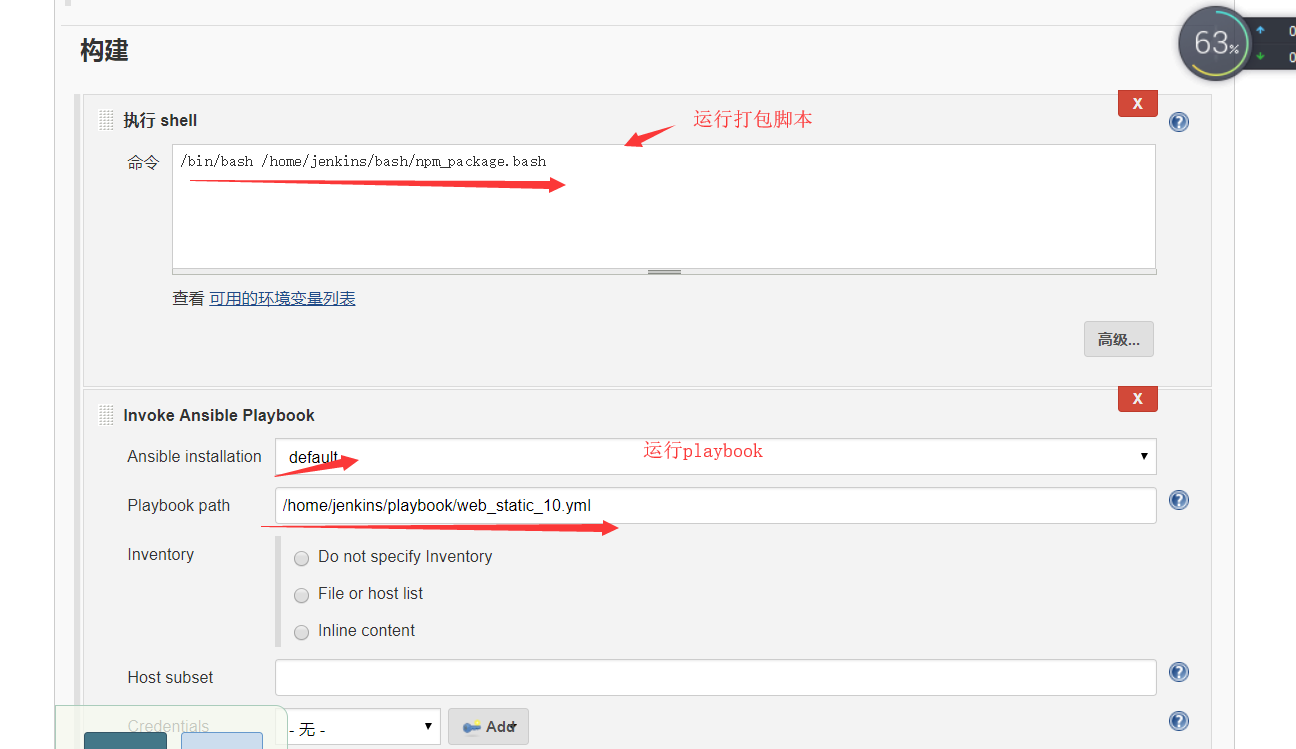
可能会包取不到值stdout ,取消—check即可通过。

## 5、jenkins 页面配置

打开jenkins web页面并登录

<http://172.27.16.16:8080>

继续编辑web\_static项目



保存

返回页面主页，点击构建项目，测试成功。

