Continuous integration using jenkins with Github, Ansible, Apache httpd.

Here we are doing a setup for deployment code in webserver automatically using jenkins & ansible, including Git and Github. It's basically CI/CD.

So we created 4 servers with names in aws or in azure vm:

Jenkins-server
Ansible-server
Web-server
Dev-server

1] Login into jenkins-server:

When we login into server with ssh using putty with pub and priv key:
—---
For mac:
Ssh -i jenkins-key.pem ec2-user@<ipaddress of ec2 jenkins server>
—--
After login we can seen that

[ec2-user@ip—---------]$ <here we can login into root user so> sudo su -

[root@<ip—-----> ec2-user]# <here we can change ip address name into jenkins or any>
hostnamectl set-hostname jenkins

Exit from the root and again login into root to see the changes.

[root@jenkins ec2-user]#
===
2] Login into Ansible-server:

When we login into server with ssh with putty using pub and priv key:
—---
For mac:
Ssh -i jenkins-key.pem ec2-user@<ipaddress of ec2 Ansible server>
—--
After login we can seen that

[ec2-user@ip—---------]$ <here we can login into root user so> sudo su -

[root@<ip—------> ec2-user]# <here we can change ip address name into jenkins or any>
hostnamectl set-hostname Ansible

Exit from the root and again login into root to see the changes.

[root@Ansible ec2-user]#
==================================

3] Login into web-server:

When we login into server with ssh with putty using pub and priv key:

—---

For mac:

```
Ssh -i jenkins-key.pem ec2-user@<ipaddress of ec2 web-server server>
```

—--

After login we can seen that

```
[ec2-user@ip—--------]$ <here we can login into root user so> sudo su -

[root@<ip—-----> ec2-user]# <here we can change ip address name into jenkins or any>
                                        hostnamectl set-hostname web-server
```

Exit from the root and again login into root to see the changes.

```
[root@web-server ec2-user]#
```

===

4] Login into dev-server:

When we login into server with ssh with putty using pub and priv key:

—---

For mac:

```
Ssh -i jenkins-key.pem ec2-user@<ipaddress of ec2 dev-server server>
```

—--

After login we can seen that

```
[ec2-user@ip—--------]$ <here we can login into root user so> sudo su -

[root@<ip—-----> ec2-user]# <here we can change ip address name into jenkins or any>
                                        hostnamectl set-hostname dev-server
```

Exit from the root and again login into root to see the changes.

```
[root@dev-server ec2-user]#
```

==

Now our servers are ready but not running.
For jenkins-server we need to update packages and install jenkins.

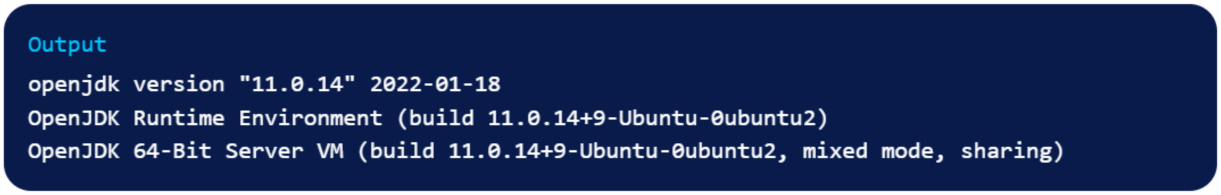
NOTE: Before installing jenkins we have to install Java because jenkins is running on java.

For ubuntu : sudo apt update
For redhat or centos : yum update -y

Next we can install jenkins:
Here am using ubuntu :

Java installation:

```
$ sudo apt install default-jre
$ java -version
```



```
Output
openjdk version "11.0.14" 2022-01-18
OpenJDK Runtime Environment (build 11.0.14+9-Ubuntu-0ubuntu2)
OpenJDK 64-Bit Server VM (build 11.0.14+9-Ubuntu-0ubuntu2, mixed mode, sharing)
```

You may need the JDK in addition to the JRE in order to compile and run some specific Java-based software. To install the JDK, execute the following command, which will also install the JRE:

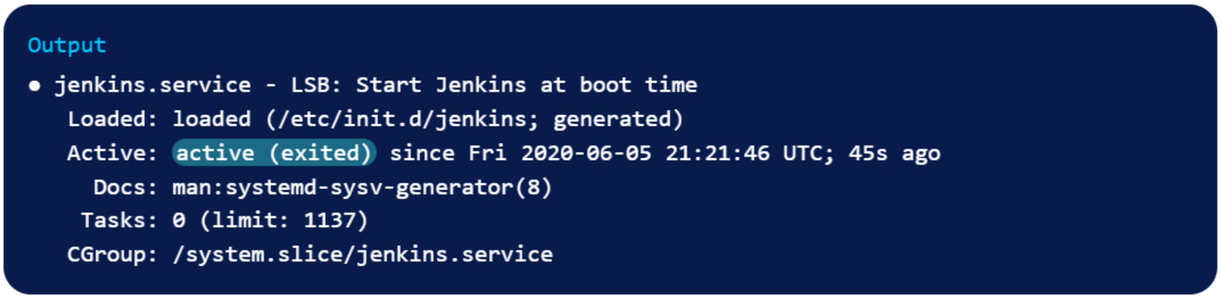
```
$ sudo apt install default-jdk
$ javac -version
```

Jenkins Installation:

```
$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'

$ sudo apt update
$ sudo apt install jenkins
$ sudo systemctl status jenkins
```



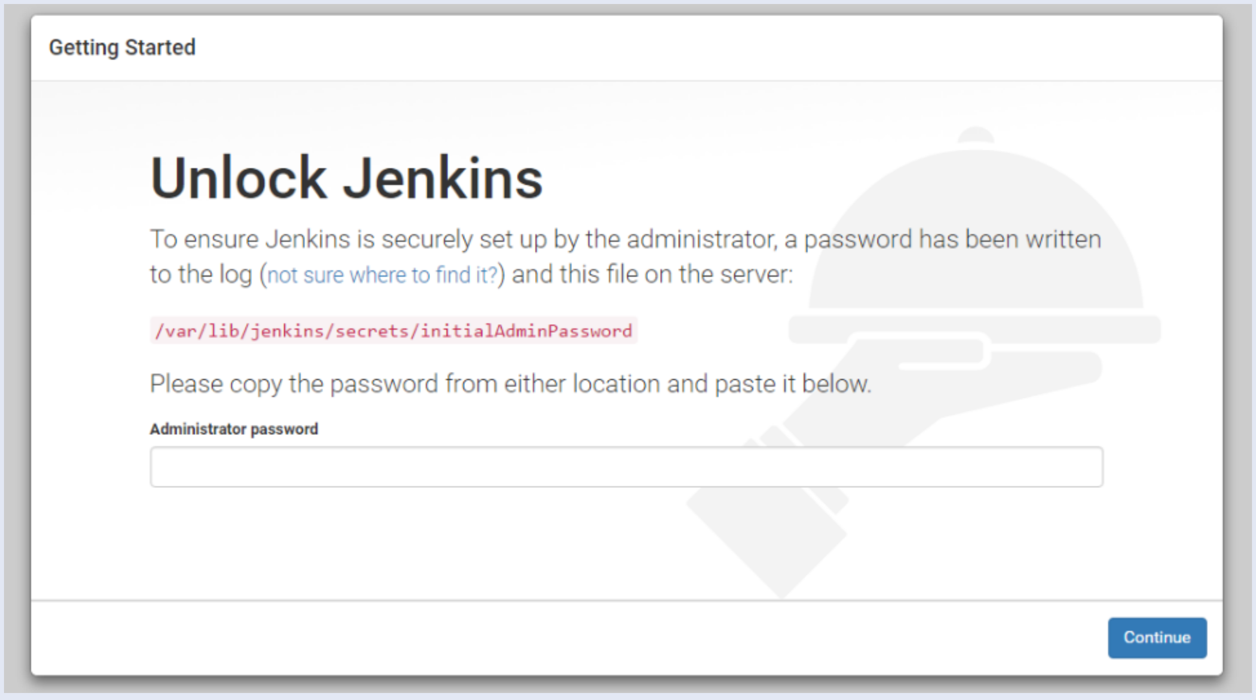
```
Output
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Fri 2020-06-05 21:21:46 UTC; 45s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1137)
   CGroup: /system.slice/jenkins.service
```

We can check the status of jenkins server is Active : active (running)

And remember we have to allow port number 8080 in nsg and in aws security group
We create inbound port rules and tcp port 8080 and anywhere from ip set these in security groups.
Bydefault jenkins is running in port 8080.

Setting up jenkins:

`http://your_server_ip_or_domain:8080  or localhost:8080 (type in your browser)we can seen below like this:`



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

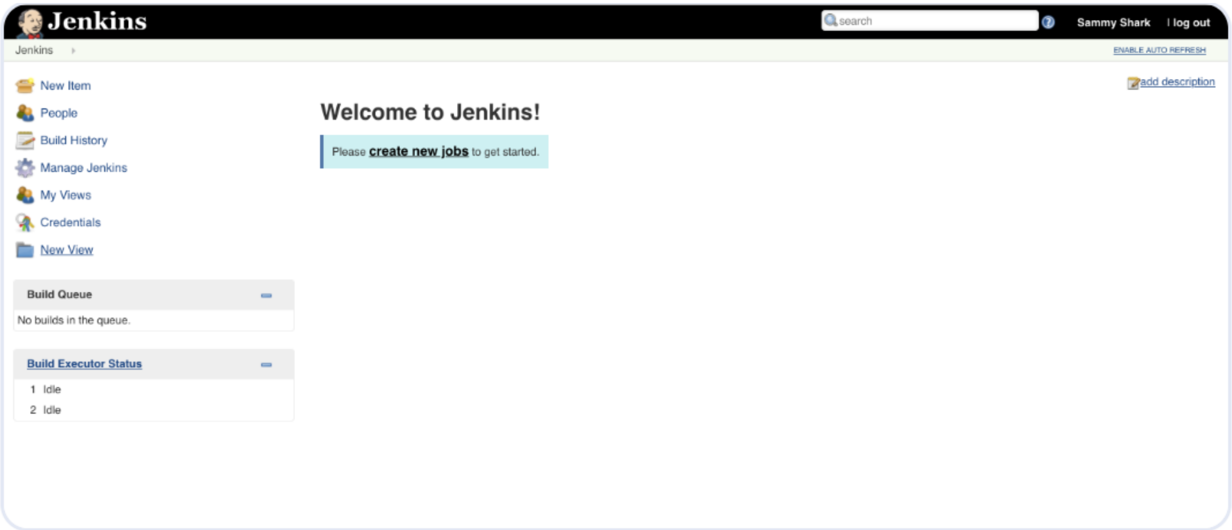
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Go to terminal and we can unlock with password
$ sudo cat <paste the above mentioned path /var/lib>

Then we can copy the password and paste it in the “Administrator password”
Click continue and install suggested plugins and create your own username and password and we all set.



Welcome to Jenkins!

Please **create new jobs** to get started.

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

We can follow the documentation for redhat installation:
https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos

Ansible installation:

What is Ansible?

. written in python, ansible is an opensource IT automation tool that allows you to automate Configuration management, application deployment and other manual operations on multiple nodes.

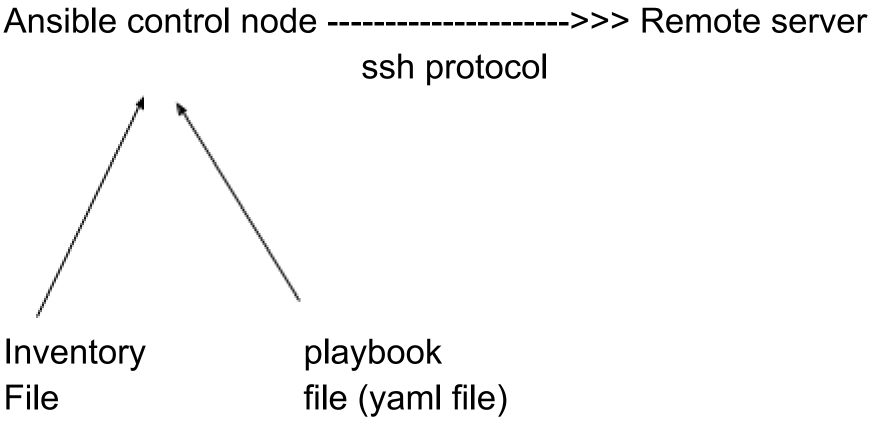
How to find the ipadress of the control node? $ ip a

. It is agentless : There is no need to install an agent (piece of software) on remote nodes in order to work with ansible. This results in low overhead during deployment.

. it uses SSH protocol and python to establish a connection with the target nodes.

. Ansible uses playbook files to define the tasks to be executed on the target system. Playbooks are written in yaml format.

. information about target nodes is stored in an inventory file which, by default is
/ect/ansible/hosts.



Ansible control node --------------------->>> Remote server
ssh protocol

Inventory
File

playbook
file (yaml file)

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository ppa:ansible/ansible --yes --update
$ sudo apt install ansible-core
$ ansible --version
# sudo apt purge ansible ansible-core (to remove ansible)
```

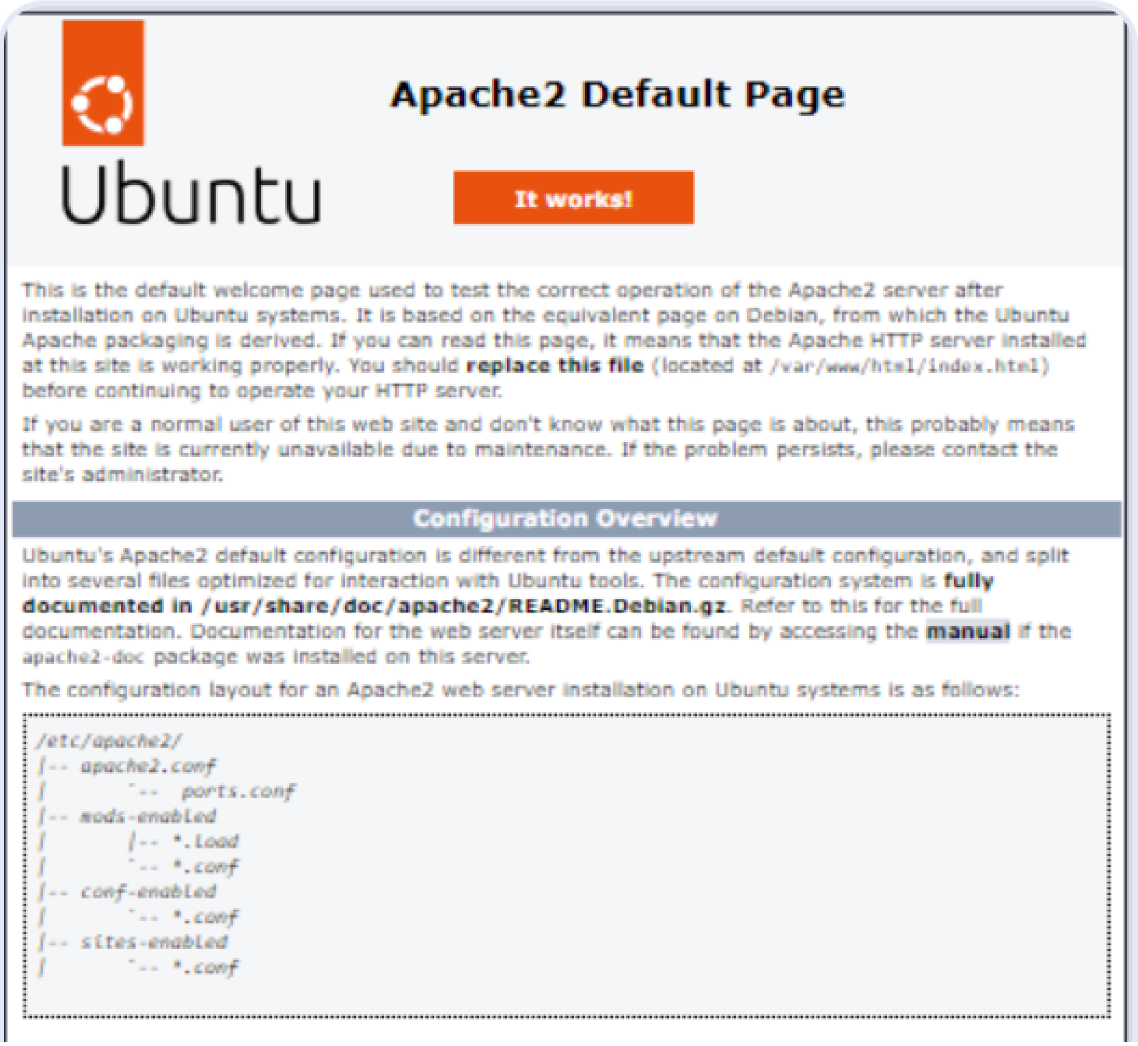
Install ansible in ansible server ..
Then start jenkins service.
Install web server that is apache httpd in web-server
Follow below commands to install webserver.

```
$ sudo apt update
$ sudo apt install apache2
$ sudo systemctl status apache2
```

Follow this below link to complete understanding of apache httpd installation in ubuntu
https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-22-04

If you do not know your server's IP address, you can get it a few different ways from the command line.

```
$ hostname -I (to check server ip)
$ http://your_server_ip
```



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Go to browser and login into jenkins server

<ipaddress of jenkins server>:8080

Now ansible is running and web-server is also running.

Now in the dev-server we need to install git and jenkins because the developer pushes the code into github through git and triggers the jenkins so that we need to configure jenkins and git in the dev-server.

1] install git in dev-server (for ubuntu follow doc.)

```
$ sudo apt update
$ sudo apt install git
$ git --version
```

Follow this link to complete installation and git configuration.
https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-20-04

2] install jenkins in dev-server (follow the above steps for jenkins installation)

If you do not install git u r unable to integrate jenkins and github properly.

Now we have to configure ssh connection jenkins ----------> ansible--------> web-server (destination)
So for that..
1] First we generate ssh keys in jenkins and transfer it to ansible.
2] In ansible again we generate ssh keys and transfer it to the web-server.

For web-server:

```
[root@web-server …]# <to change or update the root password>
                              $ passwd root
```

Enter password and will be updated.
Now we have to to change some settings in web-server config file so for that we can type

```
$ vi /etc/ssh/ssh/sshd_config
```

In the file uncomment the permitroot login

```
#permitRootLogin yes
permitRootLogin yes
```

PasswordAuthentication <default no> set yes
Save and exit

Now restart the service
systemctl restart sshd (bcoz we did changes in sshd_config file)

Check again service status and if service is not started follow below doc.
https://askubuntu.com/questions/6358/how-do-you-restart-apache

For Ansible:

```
[root@ansible …]# <to change or update the root password>
                              $ passwd root
```

Enter password and will be updated.

Now we have to to change some settings in ansible-server config file so for that we can type

$ vi /etc/ssh/ssh/sshd_config

In the file uncomment the permitroot login

#permitRootLogin yes

permitRootLogin yes

PasswordAuthentication <default no> set yes

Save and exit and restart : $ systemctl restart sshd

https://askubuntu.com/questions/103889/how-do-i-restart-the-ssh-service

Check status of sshd

$ systemctl status sshd

For Jenkins-server:

[root@jenkins-server …]# <to change or update the root password>
$ passwd root

Enter password and will be updated.

Now we have to to change some settings in web-server config file so for that we can type

$ vi /etc/ssh/ssh/sshd_config

In the file uncomment the permitroot login

#permitRootLogin yes

permitRootLogin yes

PasswordAuthentication <default no> set yes

Save and exit

Now restart the service

systemctl restart sshd (bcoz we did changes in sshd_config file)

SSH-KEY GENERATION (for password less connectivity)

[root@jenkins…..]# ssh-keygen

[root@jenkins…..]# ssh-copy-id -i root@<private ipaddress of ansible>

Now we are configured jenkins with ansible server for confirmation do ssh

$ ssh root@<ansible private ipaddress>

Output: [root@ansible……..]

Now we can check and exit.

SSH-KEY GENERATION (for password less connectivity)

```
[root@ansible…..]# ssh-keygen
```

```
[root@ansible…..]# ssh-copy-id  -i root@<private ipaddress of web-server>
```

Now we are configured jenkins with ansible server for confirmation do ssh

```
$ ssh root@<web-server private ipaddress>
```

Output: [root@web-server……..]

Now we can check and exit.

So now in the ansible server we put web-server private ipaddress in the inventory file:

```
$ vi /etc/ansible/hosts
In Ex 1 : ungrouped hosts,..................
## green.example………
## blue……………….
## <ipaddress>
## <ipaddress>
//Write here below//
[webserver]
<webserver private ip>
Save and exit.
```

We completed ssh connectivity.

Now we integrate github with jenkins:

1] So let's go to github account and create new repo:
2] lets create new simple file index.html and commit
3] go to jenkins and create a new job and create a freestyle project.
4] Jenkins----> source code management —-select Git.

Repository URL
<in the github copy code url and paste it in jenkins repo url>

Apply and save

Go to manage jenkins ——-> manage plugins ———> available plugins ———> search for “publish over ssh”
Install without restart ——-> after install restart the jenkins.

Go to manage jenkins —-> configure system —--> ssh servers —> click “Add” —-> <here we have to add jenkins private ip as well as web server private ip>

Name : jenkins
Hostname : jenkins <private ip>
Username : root

Enable use password authentication

Put root password in the box while we created it for jenkins.

Check “Test Configuration” —-----> it should be ‘success’.

Click “Apply”

And then add the same for ansible server.

Go to project—------>select configure and select Build steps —----> select “send files or execute cmds over ssh”

Ssh server
Name :
Select : jenkins

Exec command

```
rsync -avh /var/lib/jenkins/workspace/<your project name>/*.html root@<private ip of ansible>:/opt/index.html
```

We can write playbooks as well. For the above mentioned format is html and file name is index.html we can write and create any file format and mention a/c to your requirement.

Here we are configuring and transferring files that are built on jenkins to ansible.

Build Steps

Click on ‘Add post-build action’
Select “send build artifacts over SSH”

Ssh server
Name
Ansible

Exec command
```
ansible-playbook/sourcecode/deployment.yml
```

Go to terminal ansible server
```
[root@ansible …..]# mkdir /sourcecode
                              cd sourcecode
          sourcecode]# vi deployment.yml
```

```
# -name: pic file from source and push into destination

#   hosts: webserver
    tasks:
 #     - copy:
            src: /opt/index.html
            dest: /var/www/html
```

```
# cd /var/www/html
```

Go to github and open settings and click in webhooks and click on add webhook
Add payload url:
Here we have to add jenkins url so we can automate everything
http://<jenkinsweburl:8080>/github-webhook/

Content:
select **application/json**

Go to jenkins profile or username and select configure
In the API Token section
Click on add new token
And click on generate token and copy the code.
Again go back to webhook page in the secret tab and add copied token
And then click on add webhook
Go to jenkins dashboard and select project and configure and in the Build Triggers section select "Github hook trigger for Gitscm polling"
And click on apply and save.

NOW click on build now ()
Now we can copy the public ip of webserver and check on browser.

Now we can do automatically

```
[root@dev-server …..]# git clone <git repo url>
```

```
# ls
# index.html
# vi index.html
# edit file
# git add .
# git commit -m "first commit" index.html
set git config files
# git config –global user.email "your email id"
# git config –global user.name "your name"
```

After that it will ask you to login so we generate a token for automation.
Learn how to generate PAT token (personal access token)

```
# github_pat_  <token code>
# git push -u -f origin master.
```

check browser and refresh

====================================END===