

# EPAM Winter 2020-2021

## Final Task

---

CREATING BASIC CI/CD CHAIN

# The main idea is

- ▶ The main idea of this project is to create a simple CI/CD chain. At the end of this project, we will have 1 Developer, 1 Web Server (PROD), 1 Jenkins-Master. I also decided to use Ansible and write a simple playbook in which I described the configuration of our Web Server. So I'm going to add an instance for Ansible-Master as well.

# Project Steps

1

Let's create four instances for our purpose.

2

Set up Git on Dev-workstation and connected it with GitHub repository

3

Install Ansible and write playbook to configure our Web Server.














4

Install Java and Jenkins. Connecting Jenkins and GitHub via Webhook.

5

Configure Build actions in Jenkins.

# Step 1. Let's create four instances for our purpose.

Instances (4) <a href="#">Info</a>								Connect	Instance state ▼	Actions
<input type="text" value="Filter instances"/>										
<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check					
<input type="checkbox"/>	WebServer-PROD	<a href="#">i-04964985f0e1694d8</a>	 Running  	t2.micro	-					
<input type="checkbox"/>	Jenkins_Master	<a href="#">i-0cdc425847fee8f79</a>	 Running  	t2.micro	-					
<input type="checkbox"/>	Dev-1	<a href="#">i-0c3e71cef8923e4ed</a>	 Running  	t2.micro	-					
<input type="checkbox"/>	Ansible_Master	<a href="#">i-05c05d386b6d458c7</a>	 Running  	t2.micro	-					

## Step 2. Set up Git on Dev-workstation and connected it with GitHub repository

- ▶ We going to use ec2 instances along this project. After we created instance for Developer we have to set up Git. Before that I created the repository to connect. For connecting to this repository I configured access via ssh.
- ▶ Let's check if it's works with push.

```
drwxrwxr-x 3 ubuntu ubuntu 4096 Mar 26 01:20 ./
drwxr-xr-x 7 ubuntu ubuntu 4096 Mar 26 01:14 ../
drwxrwxr-x 8 ubuntu ubuntu 4096 Mar 26 01:18 .git/
-rw-rw-r-- 1 ubuntu ubuntu 1 Mar 26 01:20 index.html
ubuntu@ip-172-31-32-158:~/WebServer$ git add .
ubuntu@ip-172-31-32-158:~/WebServer$ git commit -m "Add empty
[master 7e8f433] Add empty index.html
1 file changed, 1 insertion(+)
create mode 100644 index.html
ubuntu@ip-172-31-32-158:~/WebServer$ git push
Counting objects: 3, done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:eagle-vn/WebServer.git
f7e92ed..7e8f433 master -> master
```

## Step 3. Install Ansible and write playbook to configure our Web Server.

- ▶ Using build-in modules we configure the state of our Web Server.
- ▶ What this playbook is done? Install apache2, then configure it to be enabled with boot and change the owner of index.html so that we will be able to change it using Jenkins.

```
ubuntu@ip-172-31-43-191:~/ansible$ cat web.yml
---
- name: Install apache and change owner index.html file
  hosts: PROD_SERVERS
  become: yes

  tasks:
    - name: Install Apache WebServer
      apt: name=apache2 state=present

    - name: Start Apache and Enable it with boot
      service: name=apache2 state=started enabled=yes

    - name: Change owner index.html file
      file: path=/var/www/html/index.html owner=ubuntu
```

# Check how it works.

- ▶ I have run this playbook before so nothing has changed. Everything is already in the right state. And that's cool.

```
ubuntu@ip-172-31-45-191:~/ansible$ ansible-playbook web.yml

PLAY [Install apache and change owner index.html file] *****

TASK [Gathering Facts] *****
ok: [linux-1]

TASK [Install Apache WebServer] *****
ok: [linux-1]

TASK [Start Apache and Enable it with boot] *****
ok: [linux-1]

TASK [Change owner index.html file] *****
ok: [linux-1]

PLAY RECAP *****
linux-1                : ok=4    changed=0    unreachable=0    failed=0
```



## Step 4. Install Java and Jenkins. Connecting Jenkins and GitHub via Webhook.

- I have already done these steps earlier, so we only need to check if everything is up and running. And it does!

```
ubuntu@ip-172-31-32-220:~$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~18.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
ubuntu@ip-172-31-32-220:~$ sudo service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Fri 2021-03-26 11:36:24 EET; 37min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 787 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)

Mar 26 11:36:21 ip-172-31-32-220 systemd[1]: Starting LSB: Start Jenkins at boot t
Mar 26 11:36:22 ip-172-31-32-220 jenkins[787]: Correct java version found
Mar 26 11:36:23 ip-172-31-32-220 jenkins[787]: * Starting Jenkins Automation Serv
Mar 26 11:36:23 ip-172-31-32-220 su[914]: Successful su for jenkins by root
Mar 26 11:36:23 ip-172-31-32-220 su[914]: + ??? root:jenkins
Mar 26 11:36:23 ip-172-31-32-220 su[914]: pam_unix(su:session): session opened for
Mar 26 11:36:23 ip-172-31-32-220 su[914]: pam_unix(su:session): session closed for
Mar 26 11:36:24 ip-172-31-32-220 jenkins[787]: ...done.
Mar 26 11:36:24 ip-172-31-32-220 systemd[1]: Started LSB: Start Jenkins at boot ti
```



# Configuring a trigger via a webhook

- ▶ When our developer pushes something to the repository it will send a POST request to our Jenkins-Master. After that, Jenkins will be triggered to look if something new in this repository appears?

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

---

**Payload URL \***

**Content type**

**Secret**

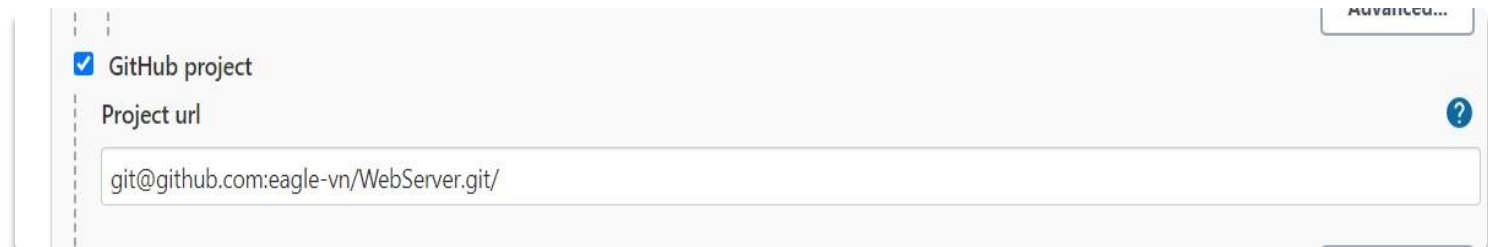
---

Which events would you like to trigger this webhook?

☒ Just the push event.

# Step 5. Configure Build actions in Jenkins

- ▶ We gonna use the following plugins: GitHub, Git, Publish Over SSH, SSH Agent. Let's make it done :)
- ▶ So, here we put our GitHub repository to make the link between the following repo and Jenkins. We already placed this link in Step 4.



A screenshot of the Jenkins configuration interface for a new project. The 'Project type' is set to 'GitHub project', which is indicated by a blue checkmark. Below this, the 'Project url' field is populated with the text 'git@github.com:eagle-vn/WebServer.git/'. To the right of the 'Project url' label is a blue question mark icon. In the top right corner of the configuration box, there is a button labeled 'Advanced...'. The entire configuration box has a light gray background and a subtle shadow.

# Pre-configuration for working with the remote repository.

Specify the repository to track. And also add the credentials for our repository.

Check the box to enable Triggers.

### Source Code Management

☐ None  
☒ Git

**Repositories**

Repository URL  
git@github.com:eagle-vn/WebServer.git

Credentials  
eagle-vn (ssh key github) Add

### Build Triggers

☐ Trigger builds remotely (e.g., from scripts)  
☐ Build after other projects are built  
☐ Build periodically  
☒ GitHub hook trigger for GITScm polling

If Jenkins will receive PUSH GitHub hook from repo defined in Git SCM section it will trigger Git SCM polling logic. So polling logic in fact belongs to Git SCM.  
(from [GitHub plugin](#))

☐ Poll SCM

# Add simple build actions, test and delivery to our Web Server.

## Build

### Execute shell

#### Command

```
echo "Build started"
ls -lah
echo "The name of this job is: $JOB_NAME and Build # $BUILD_ID" >> index.html
echo "Build finished"
```

### Execute shell

#### Command

```
echo "Test started"
test=`grep "Hello" index.html | wc -l`
echo "$test"
if [ "$test" = "1" ]
then
    echo "Test passed"
else
    echo "Test failed"
    exit 1
fi
```

## Post-build Actions

### Post-build Actions

#### Send build artifacts over SSH

##### SSH Publishers

##### SSH Server

##### Name

WebServer-PROD

# Let's check the result.

- ▶ I changed the background and pushed it to our repository. Then build started via Webhook, tested, and delivered to the Web Server.

## Recent Deliveries

✓  b9dde732-8e28-11eb-94f3-d6b43a9bc1c0 2021-03-26 14:45:13 ...

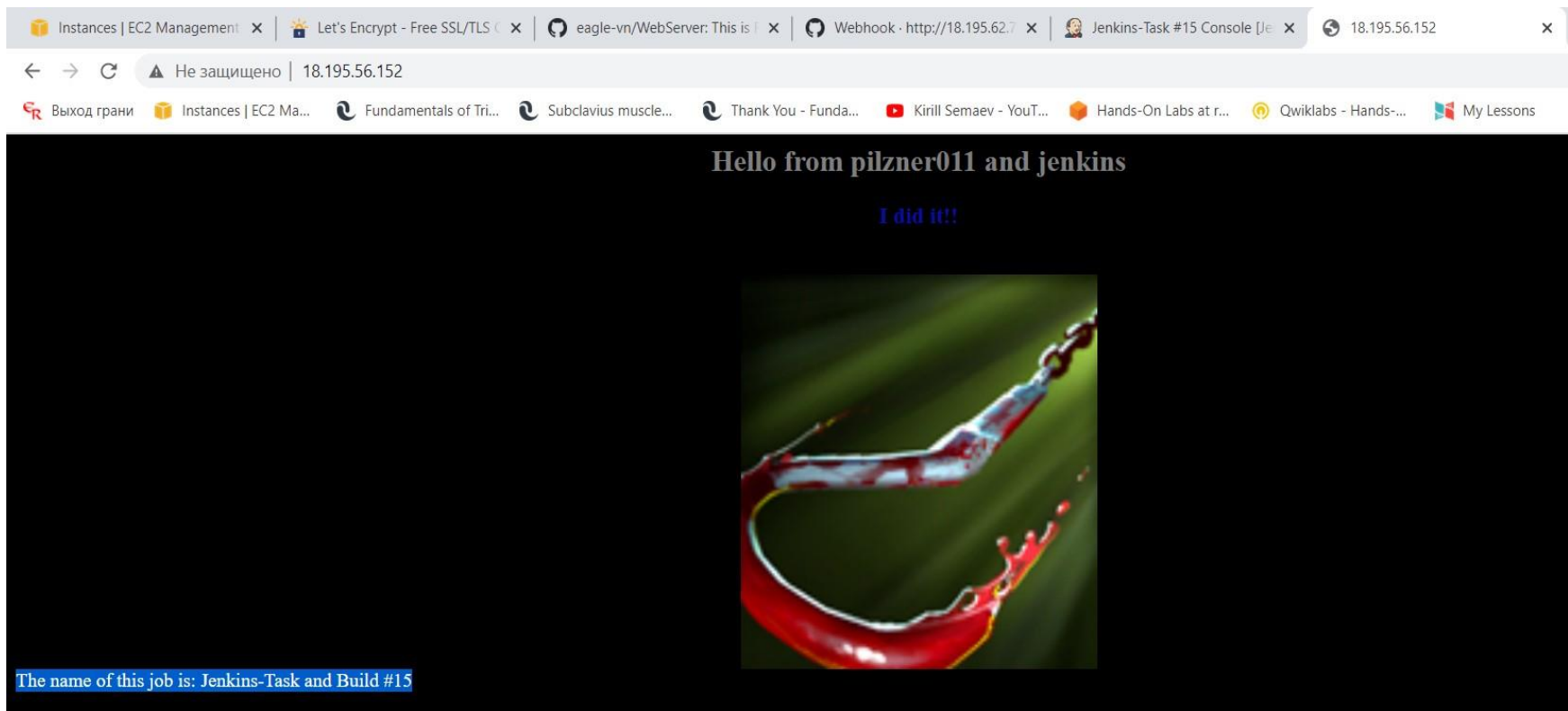
## Console Output

Started by GitHub push by eagle-vn

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/Jenkins-Task

# Thank you for attention!



**Project by Volodymyr  
Sheshulkov**