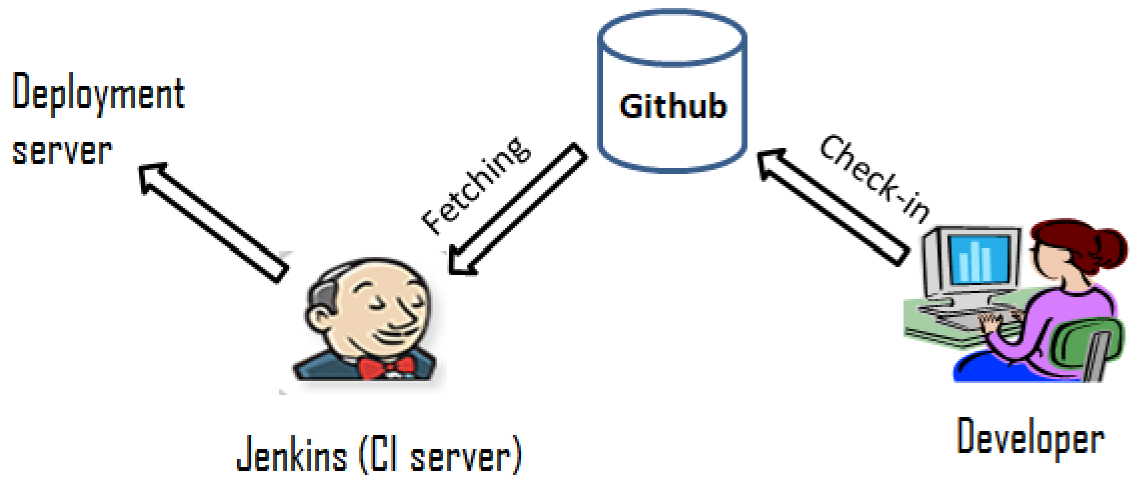
CI/CD for PHP using Jenkins as CI server and Apache2 as deployment server

anusha sharma Follow
Aug 16, 2018 · 4 min read

This blog will specify the steps to conduct CI/CD for PHP using Jenkins as CI server and Apache2 as deployment server. In this use case, both are installed on separate AWS ec2 Ubuntu 16.04 instances.

General Workflow:

Our PHP code resides on a SCM (Source code Management) server, which is in this case is Github. Jenkins (CI server) will fetch the code from the github account automatically upon check-in. Jenkins will run the build and will deploy the code in the /var/www/html directory of the Apach2 web server. To accomplish this Jenkins instance has to connect to the Apache2 instance via SSH to trigger the deployment process.



Pre-requisite:

1. Two AWS ec2 Ubuntu 16.04 instances.

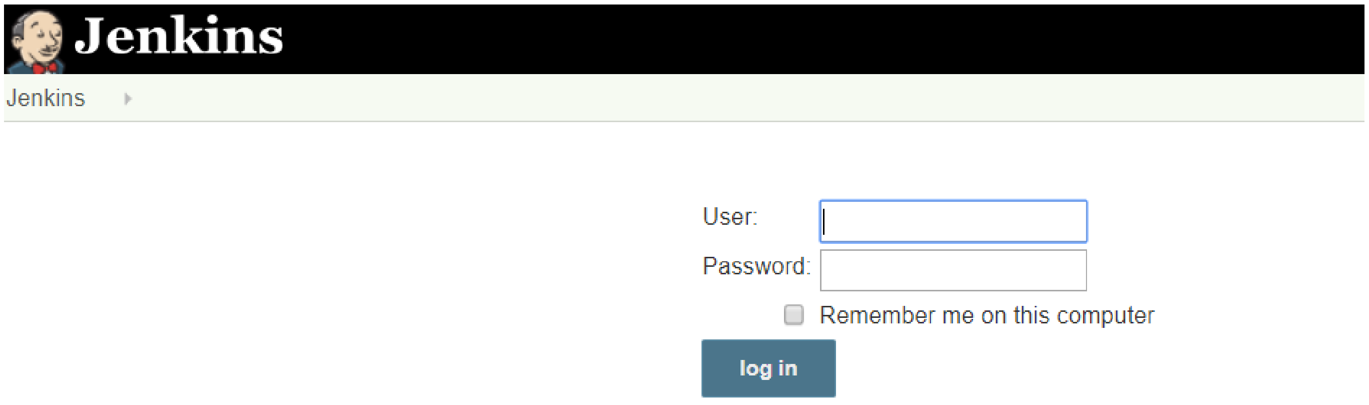
2. Jenkins and Java installation on Ubuntu instance one (Click **here**).

3. Apache2 and PHP5.6 installation on Ubuntu instance second(Click **here**).

Following steps have to be employed:

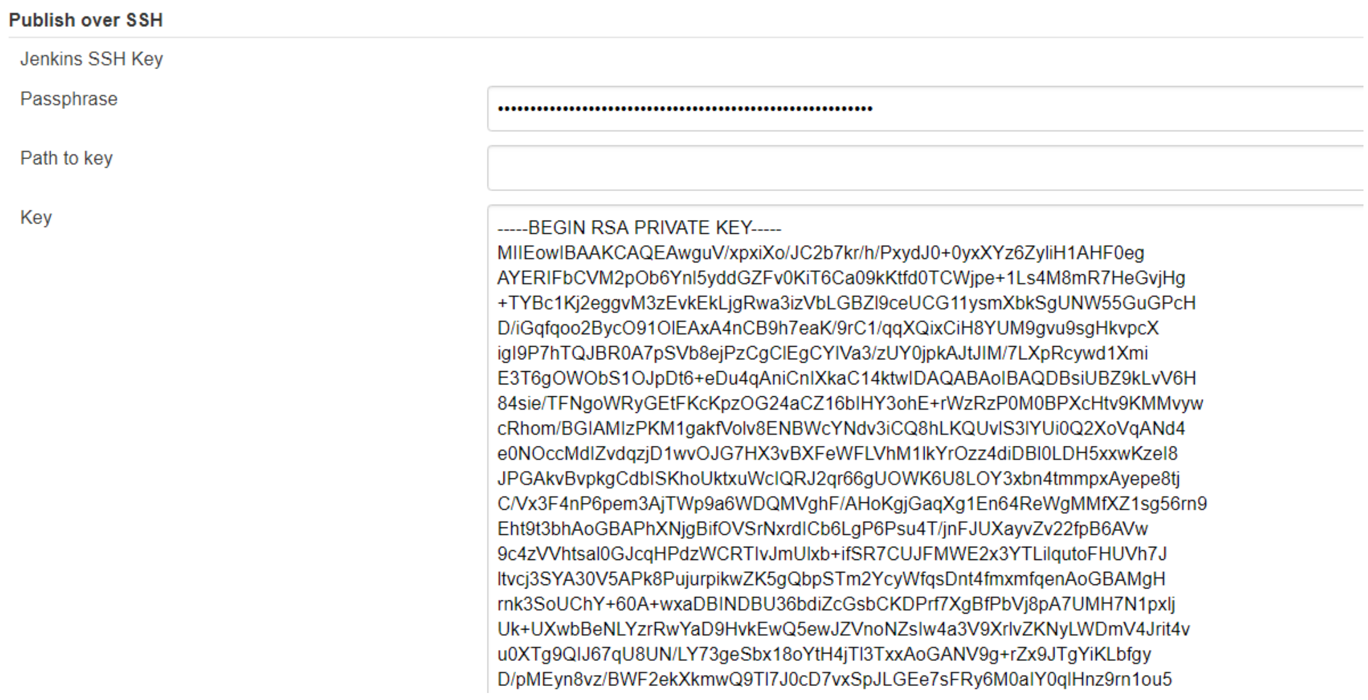
Configure Jenkins:

Step 1. Go to browser and start Jenkins server at default port 8080



Step 2. After login, Go to Manage Jenkins → Manage Plugins → Available → search for "Publish over SSH" plugin → install without restart.

Step 3. Go to Manage Jenkins → Configure System → Publish over SSH



Publish over SSH

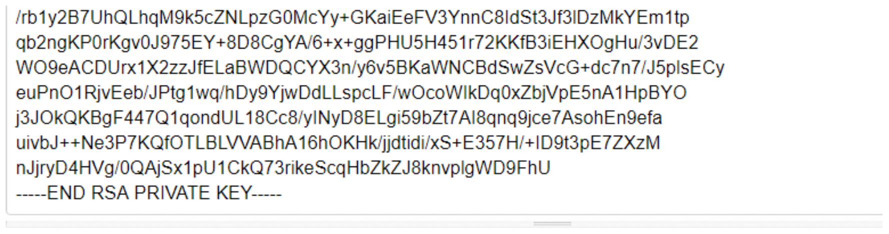
Jenkins SSH Key

Passphrase: ••

Path to key:

Key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAwguV/xpxiXo/JC2b7kr/h/PxydJ0+0yxXYz6ZyliH1AHF0eg
AYERIFbCVM2pOb6Ynl5yddGZFv0KiT6Ca09kKtfd0TCWjpe+1Ls4M8mR7HeGvjHg
+TYBc1Kj2eggvM3zEvkEkLjgRwa3izVbLGBZl9ceUCG11ysmXbkSgUNW55GuGPcH
D/iGqfqoo2BycO91OlEAxA4nCB9h7eaK/9rC1/qqXQixCiH8YUM9gvu9sgHkvpcX
igl9P7hTQJBR0A7pSVb8ejPzCgClEgCYlVa3/zUY0jpkAJtJIM/7LXpRcywd1Xmi
E3T6gOWObS1OJpDt6+eDu4qAniCnlXkaC14ktwIDAQABAoIBAQDBsiUBZ9kLvV6H
84sie/TFNgoWRyGEtFKcKpzOG24aCZ16bIHY3ohE+rWzRzP0M0BPXcHtv9KMMvyw
cRhom/BGIAMIzPKM1gakfVolv8ENBWcYNdv3iCQ8hLKQUvlS3lYUi0Q2XoVqANd4
e0NOccMdlZvdqzjD1wvOJG7HX3vBXFeWFLVhM1lkYrOzz4diDBl0LDH5xxwKzel8
JPGAkvBvpkgCdblSKhoUktxuWcIQRJ2qr66gUOWK6U8LOY3xbn4tmmpxAyepe8tj
C/Vx3F4nP6pem3AjTWp9a6WDQMVghF/AHoKgjGaqXg1En64ReWgMMfXZ1sg56rn9
Eht9t3bhAoGBAPhXNjgBifOVSrNxrdlCb6LgP6Psu4T/jnFJUXayvZv22fpB6AVw
9c4zVVhtsal0GJcqHPdzWCRTIvJmUlxb+ifSR7CUJFMWE2x3YTLilqutoFHUVh7J
ltvcj3SYA30V5APk8PujurpikwZK5gQbpSTm2YcyWfqsDnt4fmxmfqenAoGBAMgH
rnk3SoUChY+60A+wxaDBINDBU36bdiZcGsbCKDPrf7XgBfPbVj8pA7UMH7N1pxlj
Uk+UXwbBeNLYzrRwYaD9HvkEwQ5ewJZVnoNZslw4a3V9XrlvZKNyLWDmV4Jrit4v
u0XTg9QlJ67qU8UN/LY73geSbx18oYtH4jTl3TxxAoGANV9g+rZx9JTgYiKLbfgy
D/pMEyn8vz/BWF2ekXkmwQ9Tl7J0cD7vxSpJLGEe7sFRy6M0aIY0qlHnz9rn1ou5
```

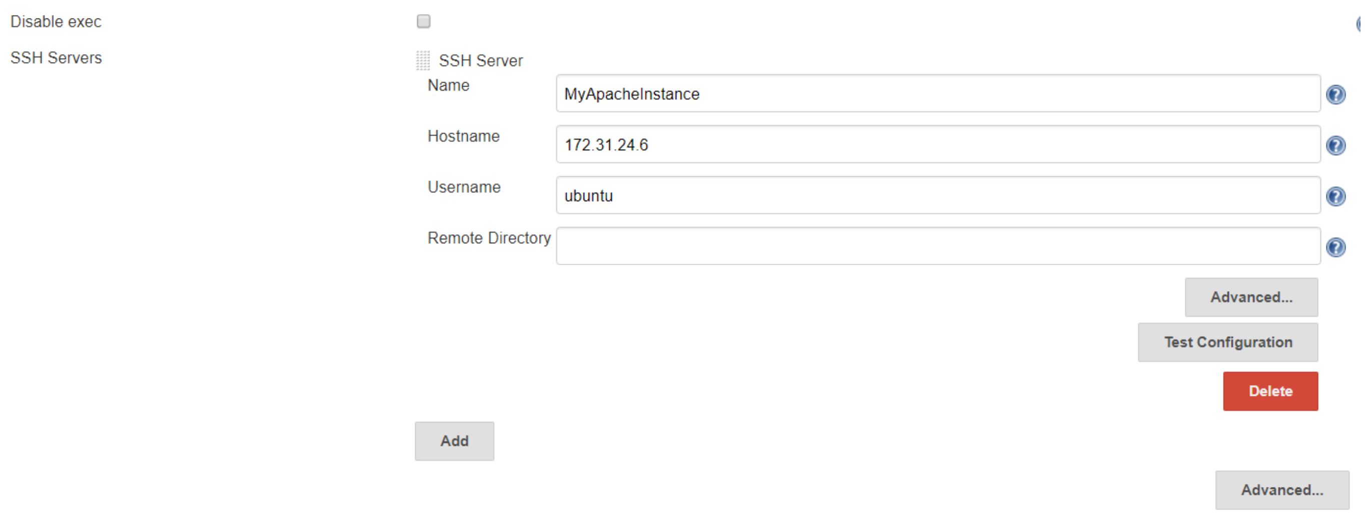


```
/rb1y2B7UhQLhqM9k5cZNLpzG0McYy+GKaiEeFV3YnnC8ldSt3Jf3lDzMkYEm1tp
qb2ngKP0rKgv0J975EY+8D8CgYA/6+x+ggPHU5H451r72KKfB3iEHXOgHu/3vDE2
WO9eACDUrx1X2zzJfELaBWDQCYX3n/y6v5BKaWNCBdSwZsVcG+dc7n7/J5plsECy
euPnO1RjvEeb/JPtg1wq/hDy9YjwDdLLspcLF/wOcoWlkDq0xZbjVpE5nA1HpBYO
j3JOkQKBgF447Q1qondUL18Cc8/yINyD8ELgi59bZt7Al8qnq9jce7AsohEn9efa
uivbJ++Ne3P7KQfOTLBLVVABhA16hOKHk/jjdtidi/xS+E357H/+ID9t3pE7ZXzM
nJjryD4HVg/0QAjSx1pU1CkQ73rikeScqHbZkZJ8knvplgWD9FhU
-----END RSA PRIVATE KEY-----
```

Either provide the path to the generated ssh key or paste it directly. It is important to paste everything including header and footer as shown in the above snip from my lab experiment.

If wondering how to generate a SSH key to establish connection between two servers, click **here** to get more details and steps.

Then click on ADD button in order to add a server to SSH with/ connect with.



Disable exec	☐
SSH Servers	SSH Server
Name	MyApacheInstance
Hostname	172.31.24.6
Username	ubuntu
Remote Directory	

Fill in the details, like;

- Name: Provide any logical name which can be used later on to configure job.
- Hostname: ip address of the server to connect to (in this case it is the instance second hosting Apache2)
- Username: name of the user to login to (in this case it is ubuntu)
- Remote Directory: path of any directory you want to deploy to (or can be leaved blank for later configuration within the job)

Step 4. Click 'Test Configuration' to confirm the connection and 'Save' at the bottom of the page.

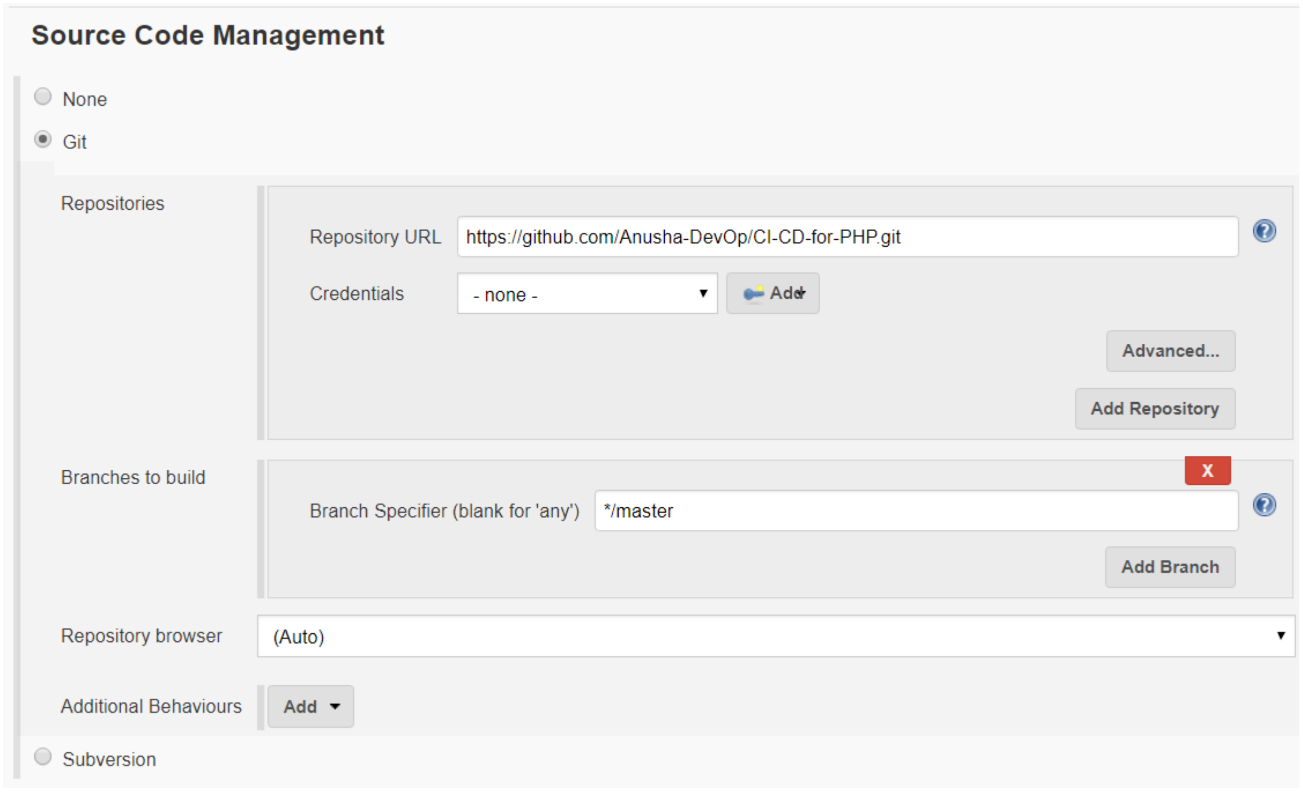
Note: Both servers must be up and running to test the configuration.

Create a Jenkins Job:

Step 1. Go to Jenkins Dashboard → Click on 'New Item' → Provide name of the project (e.g. CI-CD-PHP) → choose 'Freestyle Job' → Click 'OK'.

In the configuration window of the job:

Step 2. Provide the Git URL from where code has to be pulled from.


Source Code Management

None
[✓ Git]

Repositories
Repository URL: https://github.com/Anusha-DevOp/CI-CD-for-PHP.git
Credentials: - none -

Branches to build
Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

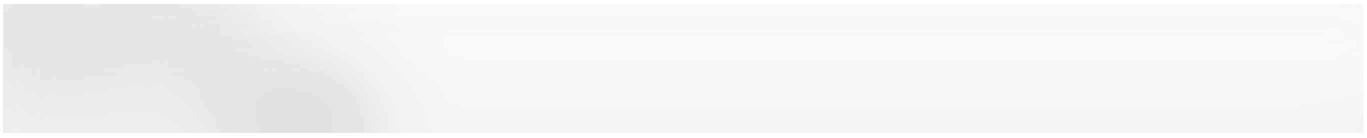
Additional Behaviours

Subversion

Step 3. In the Build Environment section choose:

a). Delete workspace before build starts.

b). Send files or execute commands over SSH after the build runs



Provide the Name of the server, source files and remote directory.

OR

Another strategy is to archive artifacts and then send build artifacts over SSH

Also specify the remote directory (Although, if already specified in the configuration system then not necessary to specify it here).

Step 4. Save the job and build it.

Step 5. Check the Apache2 server for the successful transfer of your files.

Hope you liked the summarized steps. For more technical blogs on DevOps tools and AWS, visit our website https://devops4solutions.com/

)

Jenkins Ssh Ubuntu Server