**Jenkins Master**

1. Take EC2 Instance with Ubuntu free tier
2. Connect using Mobaxtern – Master machine public IP
3. Go to root user and run “**apt-get update**”
4. Install Java using “**apt install default-jre**”
5. Set permanent home path setting for java using below cmd

**cat >> /etc/environment <<EOL**

**JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64**

**EOL**

1. Installation of Jenkins in Master machine
   1. First, Get jenkins packages in local master machine

Using below cmd you will download key

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

* 1. Now once you got key, Update the repository using

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

* 1. Now again update the box, to see jenkins site is connecting or not. Using

“**apt-get update**”

* 1. Now, Install Jenkins using “**apt-get install jenkins - y**”

1. Go to Security group to enable 8080 port
2. Now to go to jenkins portal use “**MASTER PUBLIC IP:8080**”
3. First time it will ask for adminstration password for that go to

/var/lib/jenkins/secrets/initialAdminPassword

using “**cat /var/lib/jenkins/secrets/initialAdminPassword**”

1. Now copy key and give in jenkins portal and process to **install suggested plugins**.
2. Fill below form as required

User Name: Jenkins-Admin

Password: admin

Full Name: Jenkins Administrator

Email: [bogasan24041998@gmail.com](mailto:bogasan24041998@gmail.com)

1. Now, set up port number for agent in **Manage jenkins > Configure Global Security**

TCP port for inbound agents > Fixed - Example Port No: 3000

1. Now give Master Private IP in **Manage Jenkins > Configure System** in **jenkins location** section so that agents can connect to master using this private ip, as all are in the same vpc.
2. Now Create a node i.e. “worker machine” **Manage Jenkins > Nodes**
   1. Set as Permanent
   2. Create jenkins\_jobs directory in worker machine connect using mobaxtern and create in **/opt/jenkins\_jobs** 🡪 make sure directory has full access
   3. Using this directory in remote directory which is asked while creating node.
3. Now you will see worker-01 is not connected by X symbol – Means its offline. Now to make it online we need to setup or configure agent in worker machine
4. Create agent directory -> “sudo mkdir agent” and give chmod 777 agent full access and also to jenkins\_jobs folder which create already
5. Setup agent in worker machine using in agent directory

**wget http://172.31.95.148:8080/jnlpJars/agent.jar**

1. **Install Java and set environmental path** in worker machine and then run below cmd

Run this command in @root

**java -jar /opt/agent/**[agent.jar](http://54.86.198.191:8080/jnlpJars/agent.jar) **-jnlpUrl http://172.31.95.148:8080/manage/computer/Jenkins%2DWorker%2D01/jenkins-agent.jnlp -secret defb1145d3b258ef81ef7fbb703c93697415278c9edcc3ee81adac3448651331 -workDir "/opt/jenkins\_jobs"**

This command is given by jenkins worker machine when clicked on offline status

Once run above cmd it should show **connected**

1. Now if we see **Manage Jenkins > Nodes** worker-01 machine will be connected. That means agent now can connect to master machine.

**Note: Now if we disconnect the machine in mobaxtern or aws instance or worker machine in jenkins portal will be disconnected and again when we connect we have to again run jar and above step. Manually from (15 – 17)**

**Solution: RUN AGENT.JAR AS SERVICE**

1. To automatically connect the worker-01 **we need to run the worker agent as service**
2. **Create shell script file as below**

**Create this file within agent folder vi agent-service.sh** (press enter)

Press i – for insert and insert below cmd

**java -jar /opt/agent/**[agent.jar](http://54.86.198.191:8080/jnlpJars/agent.jar) **-jnlpUrl http://172.31.95.148:8080/manage/computer/Jenkins%2DWorker%2D01/jenkins-agent.jnlp -secret defb1145d3b258ef81ef7fbb703c93697415278c9edcc3ee81adac3448651331 -workDir "/opt/jenkins\_jobs"**

then press esc + :wq! + press enter

1. Now we need to create a **service** **file**(agent.service) which will run the above shell script file in background in **/etc/systemd/system/agent-service.service**
2. **Vi /etc/systemd/system/ agent-service.service**

[Unit]

Description=JenkinsSlave

[Service]

User=root

WorkingDirectory=/opt/jenkins\_jobs

ExecStart=/bin/bash /opt/agent/**agent-service**.sh

[Install]

WantedBy=multi-user.target

1. Enable the daemon with the following command.

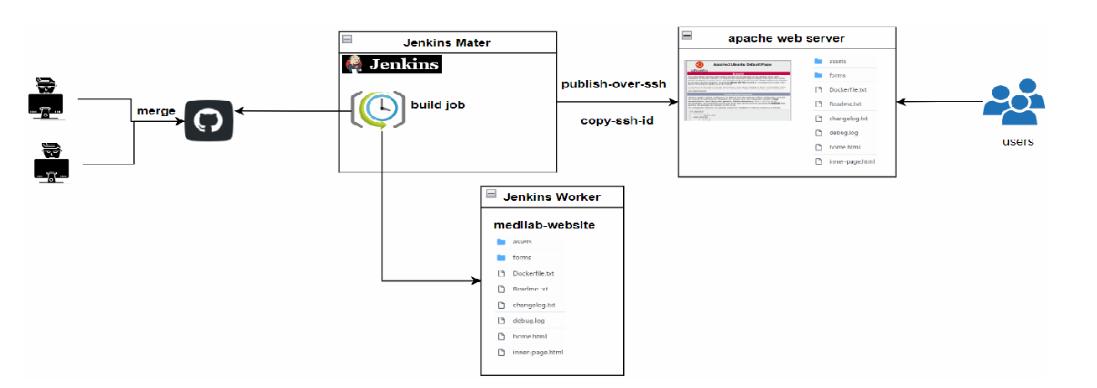
**sudo systemctl enable agent-service.service**

1. Now start the daemon with the following command.

**sudo systemctl start agent-service.service**

1. Check the status with the below command

**sudo service agent-service status**



**Till here we have setup master and slave(worker) machine**

**Web server Machine(EC2 Instance)**

1. Create t2.micro ec2 instance
2. Connect to instance and update box
3. Create AdminUser using “**sudo adduser apacheadmin**” and then add all privilages vi /etc/sudoers
4. Enable PasswordAuthentication to “Yes”

**vi /etc/ssh/sshd\_config**

**press i (insert)**

**scroll down to**

**PasswordAuthentication no -> make yes**

**Esc + :wq! + press enter**

1. Reload the above modifies service file “**service sshd reload**”
2. Install java and apache web server “**apt-get install apache2**”
3. Enable http i.e. 80 port in security group and check html file in browser it will work
4. **Build SSH connection between worker and web server machine**
   1. Generate the public and private RSA keys using the command: “**ssh-keygen**” in worker machine
   2. To View keys go to “**cd .ssh**” then “**ls - l**” you will se public and private key
   3. Copy public i.e. .pub file from worker to web server ec2 machine

**ssh-copy-id apacheadmin@private\_ip\_web\_server**

Give password when asked i.e. same password when you create user in 3 step -> admin

* 1. To connect for testing use **ssh apacheadmin@ private\_ip\_web\_server**

1. Now as you have build the connection between worker and web server you can always go to terminal and do manually for that install plugin called “**publish over ssh**”
2. Go to **Manage Jenkins > Configure System** scroll down to Publish over ssh section and add web server machine details

Name: Apache Web Server

Hostname: Private IP

Username: apacheadmin

Remote Directory: /var/www/html

Select password authentication and give password i.e. admin

Then check Test Configuration 🡪 should come as success

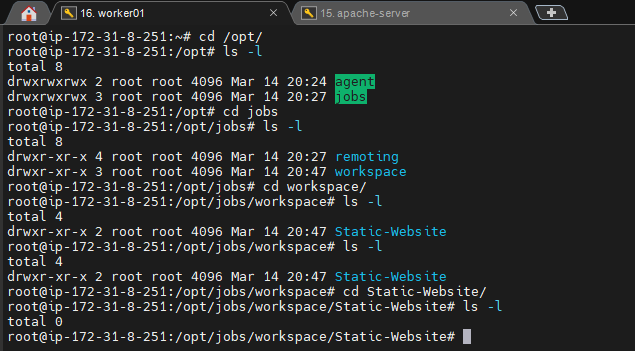
1. Make sure /html has full permission or else give using “**chmod 777 /var/www/html**” in Apache server

**Till here we have completed Creating Master, Worker and Web Server in our case apache.**

**Then we have also completed password authentication between worker and server**

**Now we will start will how to pull project from git and then publish in web server.**

1. Create a freestyle project by giving name example - “Static-Website”
2. For now to test wheather workspace directory is creating in worker machine or not apply and save and then **build now**



Now go back to project 🡪 click configure

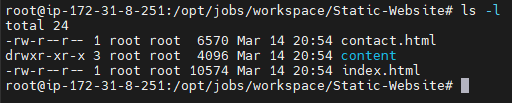
1. Go to source code management 🡪 click 🡪Git In Repositry URL give <“github project link”>

ex: <https://github.com/iamsanjayboga/AWS-Static-Website>

Scroll down and give the branch name may be “main” or “master”

**Then apply and Save**

1. Now if you build now and see workspace directory in jenkins portal or worker machine you must find the github files



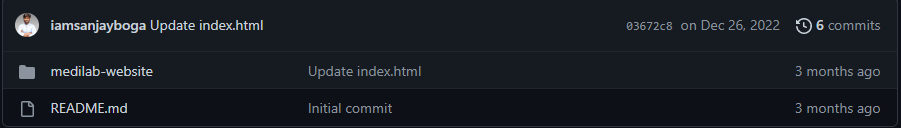
**Now as we have code in worker machine we need to deploy this code in web server (apache)**

1. To do this now go to project name in this case <Static-Website> 🡪 configure 🡪Build 🡪

Send file or execute command over SSH

1. Once we click above option we have to provide **source file** in this format

If you have directory in github ex



Then in this case source file will be **medilab-website/\*\*/\***

If you directly have file means no initial directly then **\*\*/\***

1. Now if you see, your application will be live
2. Now, Set up push mechanism i.e git will notify jenkins when new commit is available
3. Go to jenkins, people module and select Jekins-Adminstrator. Go to configure and then

Their generate a key shown below.

11d26daa1b1168d8438ee15da5a0369215

1. Once generated go to github and respective repository then go to setting and webhooks on left hand go to payload url section and give url/github-webhook/.
2. Give hosted url i.e. apache web server public ip ex “http:// 3.239.225.239:80/github-webhook/
3. Give above generated key in below password field and save. Now check the recent deliverables and done.