**3Installation of Jenkins on Ubuntu Machine**

**Links:**

1. <https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-18-04>
2. <https://linuxize.com/post/how-to-install-jenkins-on-ubuntu-18-04/>

**Implement CI/CD Build Pipeline for Java JSP Application on Master/Standalone Jenkins Server**

Log in to Jenkins Server.

Create a Free Style Jenkins Project.

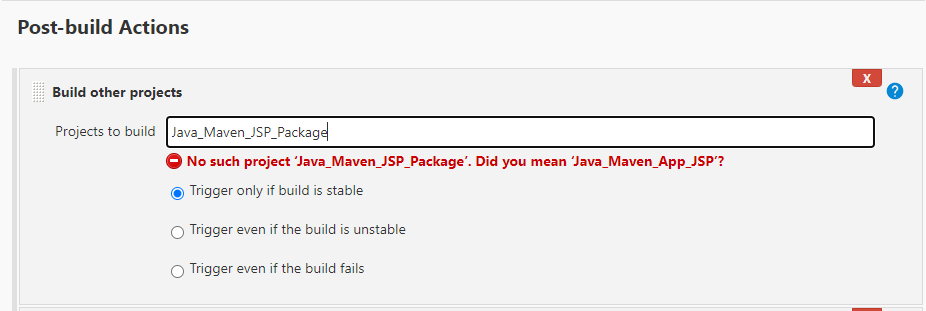
Provide a name to Project for Example: Java\_Maven\_JSP\_Build

GitHub Repo: <https://github.com/sreepathysois/MyMavenApp.git>

Build using: Invoke Top Level Maven Targets



After Building the application. In post Build Actions Invoke Another Free Style Jenkins Project Name Java\_Maven\_JSP\_Package.



Create New Free Style Project Titled: Java\_Maven\_JSP\_Package.

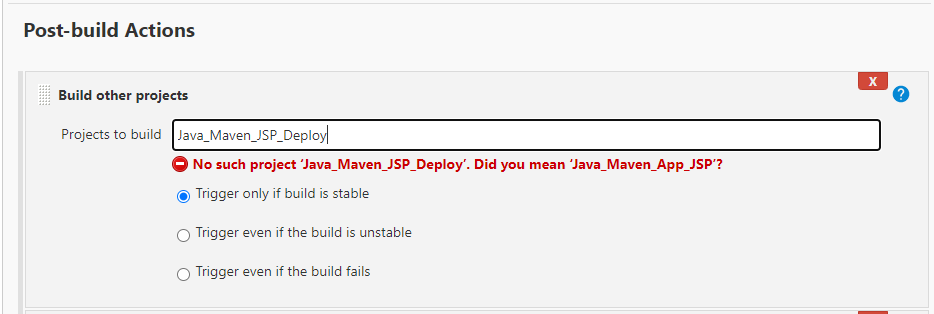
Use the Same Github Repo: <https://github.com/sreepathysois/MyMavenApp.git>

Build using: Invoke Top Level Maven Targets



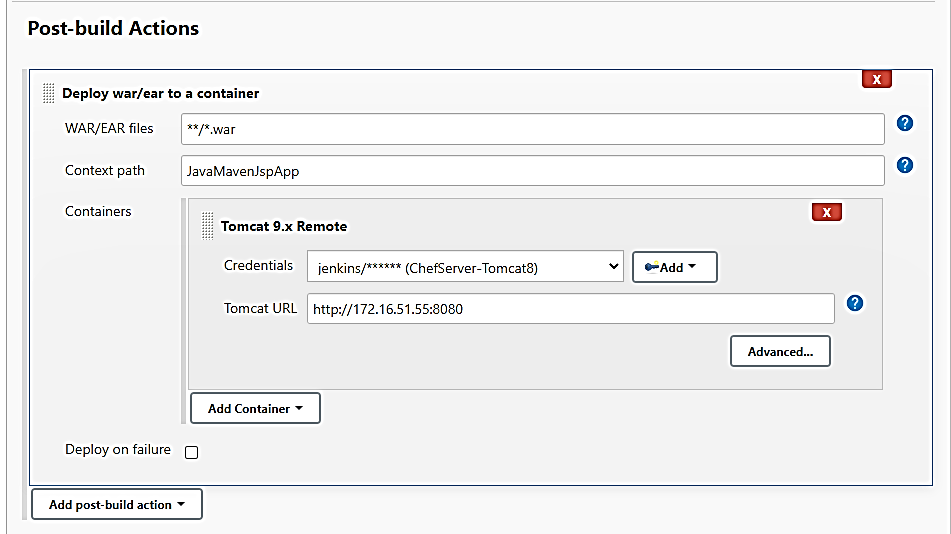
Using Maven Package all the Dependencies of Application in to a war Package.

In Post Build Actions Invoke other Jenlkins Project Java\_Maven\_JSP\_Deploy Project.



Deploy the application on to Tomcat Server using Post Build Action

Install Deploy to EAR/WAR Plugin



Create a Build Pipeline using Build Pipeline plugin.

Java\_Maven\_JSP\_Compile à Java\_Maven\_JSP\_Package Project à Java\_Maven\_JSP\_Deploy Project.

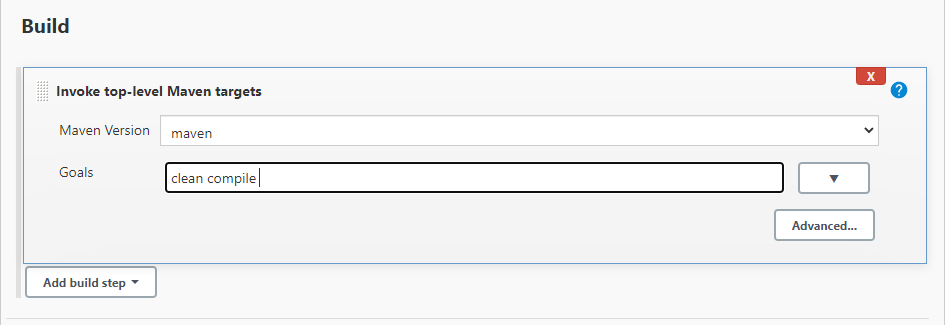
**Implement CI/CD Build Pipeline for Java Application on Master/Standalone Jenkins Server**

Create a Free Style Jenkins Project.

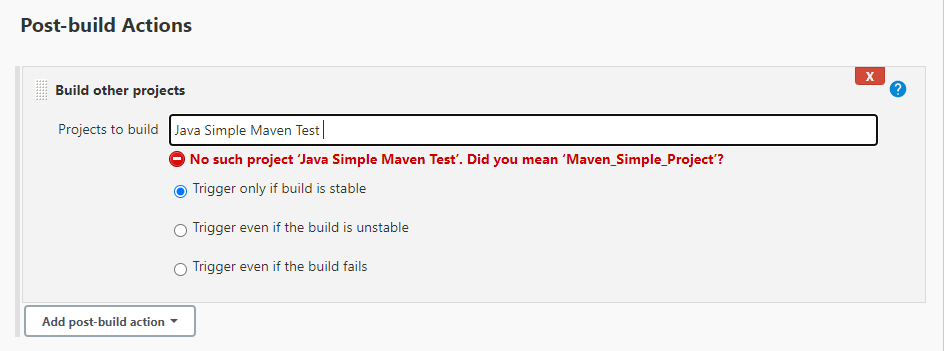
Provide a name to Project for Example: Java Simple Maven Compile Project

Github Repo: <https://github.com/sreepathysois/simple-java-maven-app.git>

Build using: Invoke Top Level Maven Targets



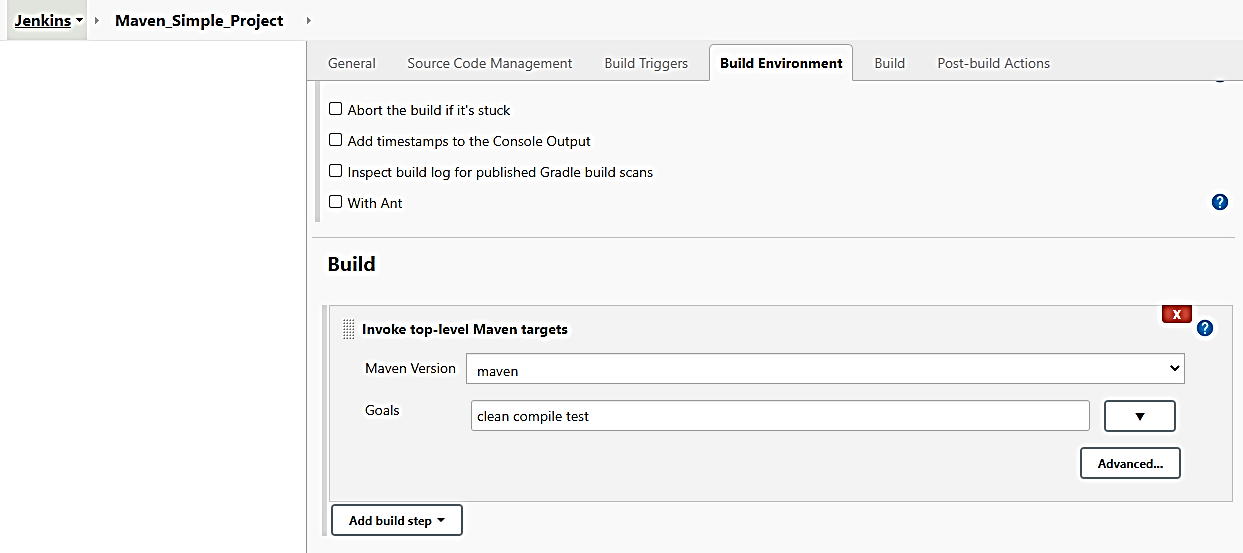
After Successful Build in Post Build Actions Invoke Java Simple Maven Test Project



Create FreeStyle Jenkins Project Titled: Java Simple Maven Test

Github Repo: <https://github.com/sreepathysois/simple-java-maven-app.git>

Build using: Invoke Top Level Maven Targets



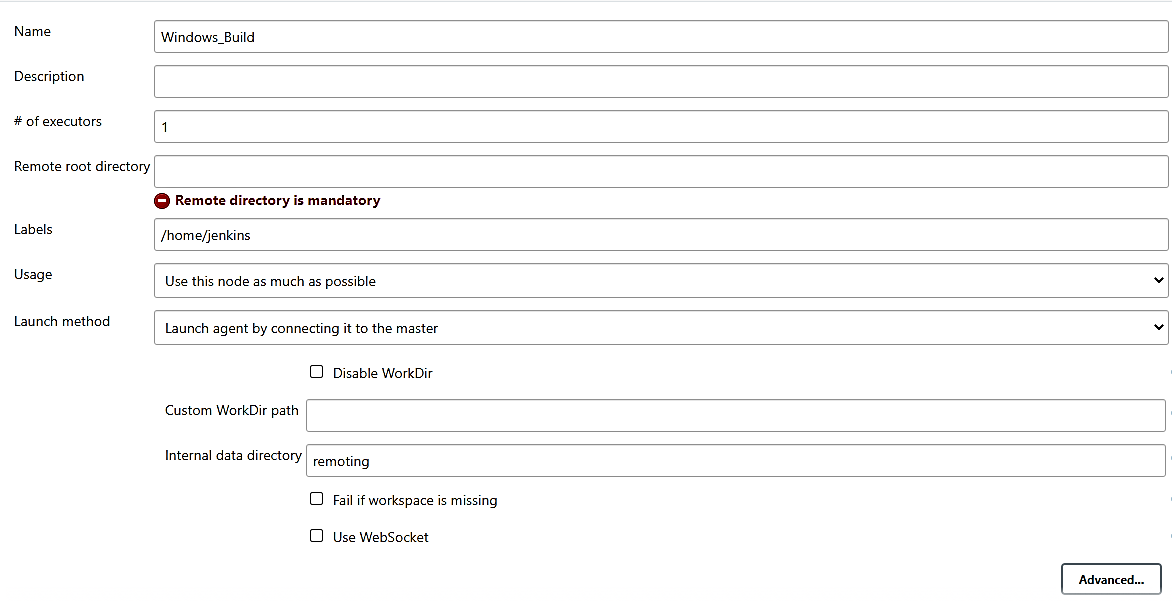
Output the Test Result to Test Result Report Plugin or HTML Publish Plugin etc.

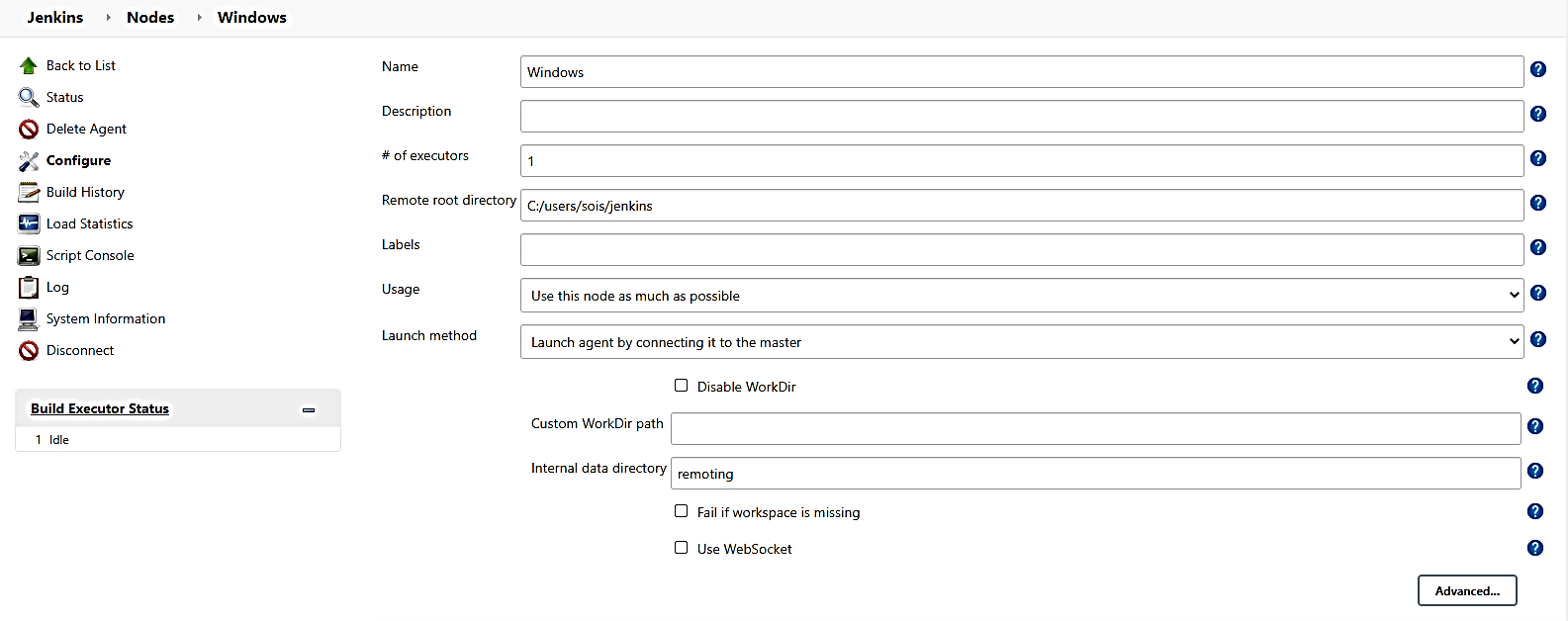
**Implement CI/CD Build Pipeline for ASP.Net Application on Jenkins Master – Slave Server**

Preparation and adding a slave windows node to Jenkins Master

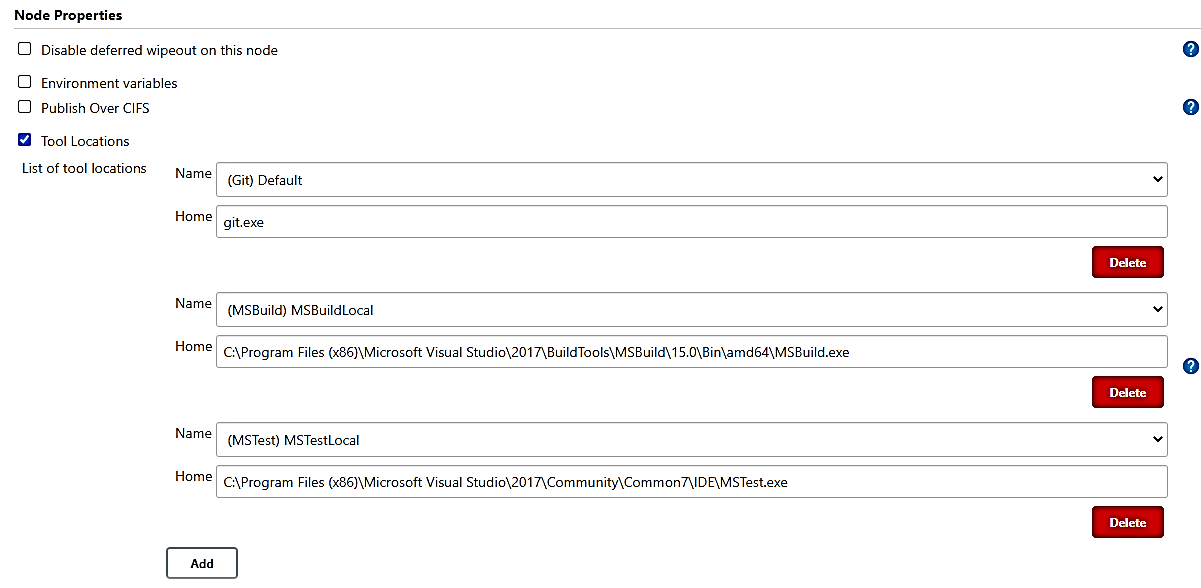
Go to Manage Jenkins à Manage Nodes and Cloud and Add a new node.

Provide a name and select either Permanent agent or Copy Settings from Existing Node.

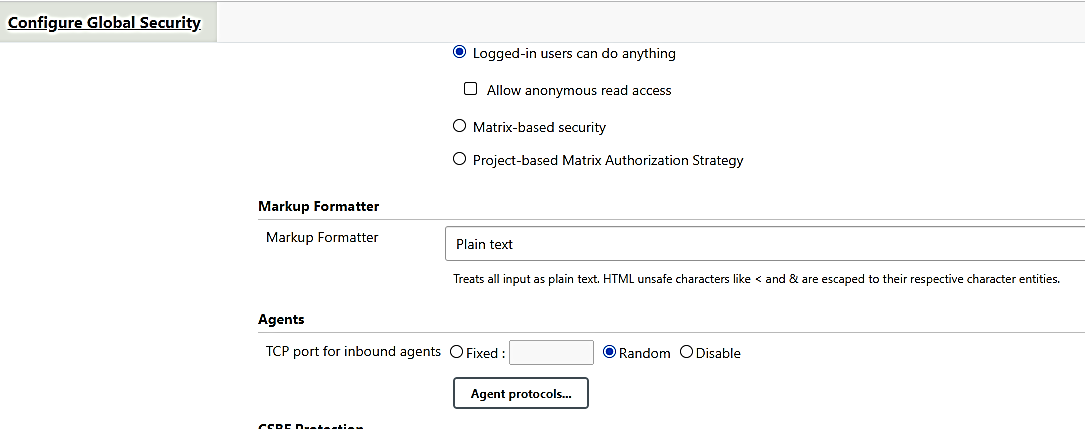




Add Additional Tool Location on Machine under Node Properties.



Go to Manage Jenkins – Configure Global Security and Enable Random or Fixed TCP port for Agents.



To Connect Slave Node to Master Jenkins:

Go to Slave Machine à Download agent.jar file to Machine. Install JDK. Finally run command similar to the following:

java -jar [agent.jar](http://172.16.51.27:8080/jnlpJars/agent.jar) -jnlpUrl http://172.16.51.27:8080/computer/Windows\_Build/slave-agent.jnlp -secret e04b81d9b6e88deece5037045667ff6ebe56fde00e114adb14334041c48fd138 -workDir "/home/jenkins/"

**Windows Slave Machine Environment Preparation for ASP.Net Application**:

In windows slave machine

**· Install git (in not already installed)**

https://git-scm.com/downloads

Open Windows PowerShell to Install the Following with Administrator Access.

· **Install nuget**

Invoke-WebRequest https://dist.nuget.org/win-x86-commandline/latest/nuget.exe -OutFile Nuget.exe

· **Install chocolatey: https://docs.chocolatey.org/en-us/choco/setup**

Set-ExecutionPolicy Bypass -Scope Process -Force;

[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

· **Install msbuild refer to: https://community.chocolatey.org/packages/microsoft-build-tools**

· **Install visualstudio web build tools**

choco install visualstudio2019-workload-webbuildtools

·Install IIS refer to: https://www.itechguides.com/install-iis-windows-10/

\*\* Enable all the ASPNET services without fail.

MS Deploy Install:

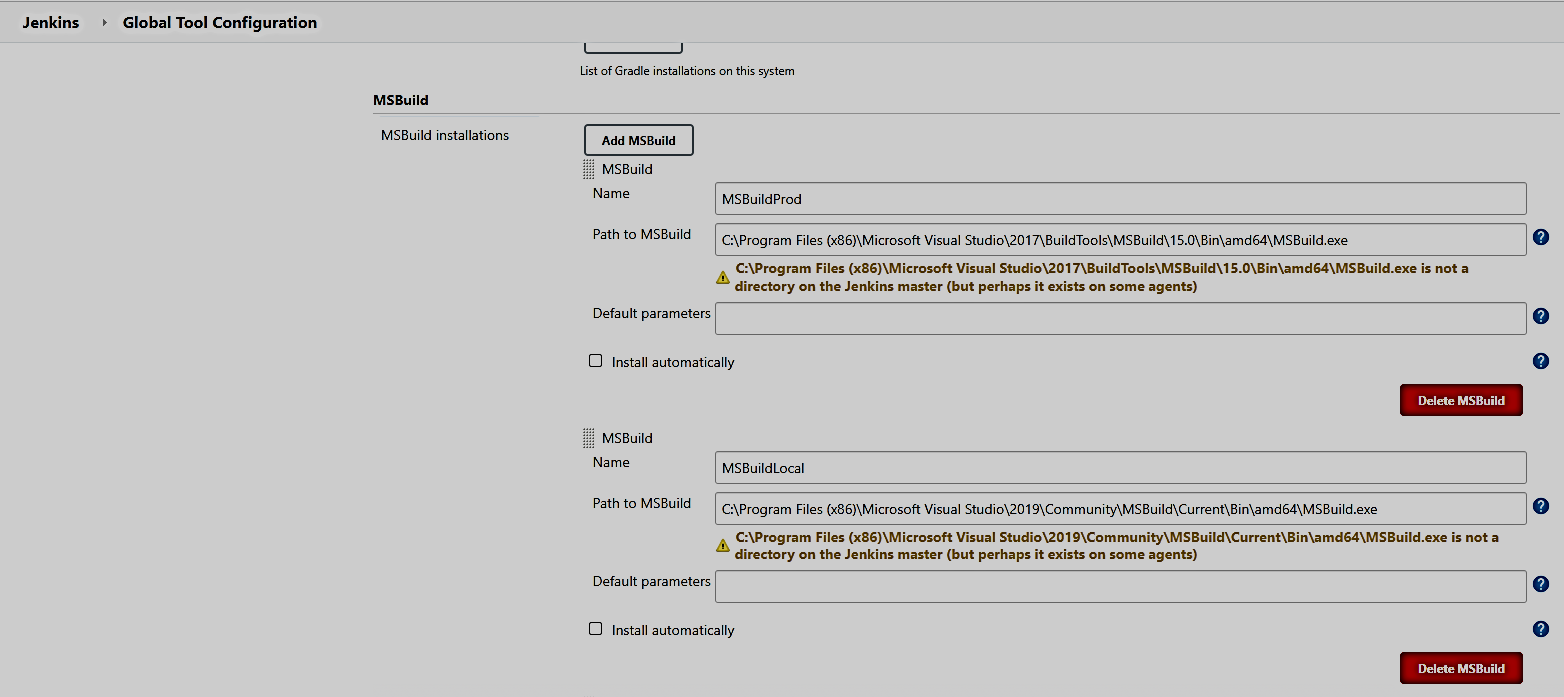
**Dotnet install:**

choco install dotnetcore-sdk

Search for Turn windows features on or off navigate to Internet Information Services -> World Wide Web Services ->Application Development Features -> Enable all

**In master Jenkins need to add MSBuild/MSTest tools in Global Tool Configurations.**

To configure a MSBuild: Goto Manage Jenkins -> Global Tool Configuration -> (Navigate to MSBuild) -> Add MSBuild

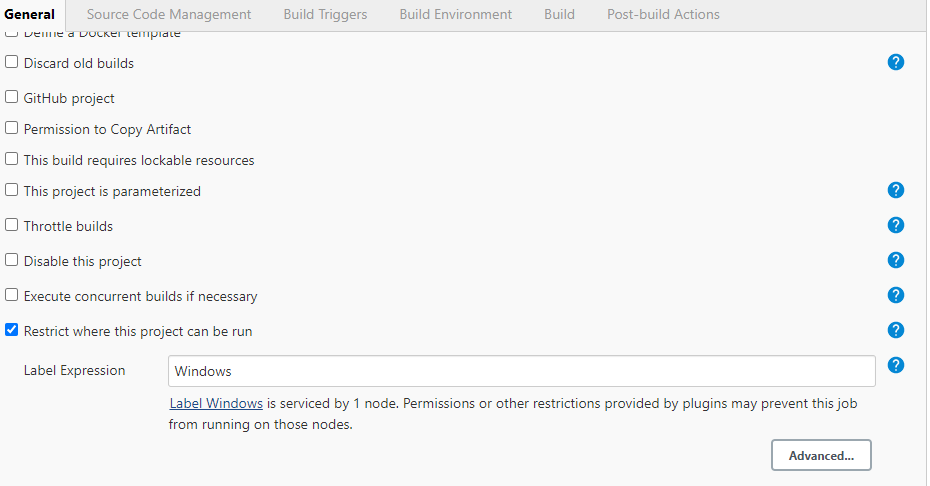


**Create ASP.Net Build Pipeline:**

Create a FreeStyle Jenkins Project.

Provide a name to project like: CalculatorAppAsp

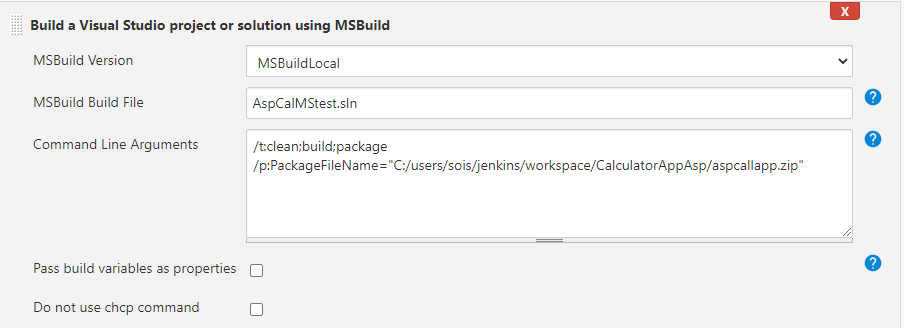
Github Repo: <https://github.com/sreepathysois/CalculatorMStest.git>



Install Dependencies of project using nugget on Windows batch command.



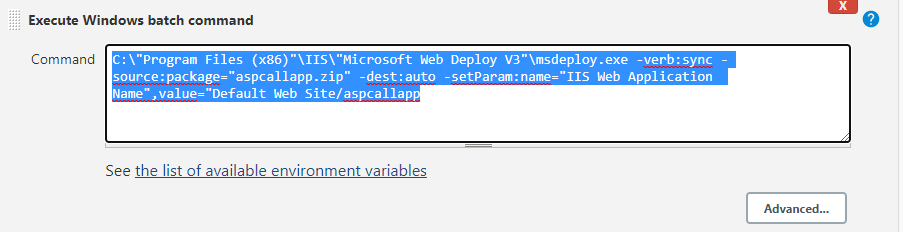
Build the project using MSBuild tool:



/t:clean;build;package /p:PackageFileName="C:/users/sois/jenkins/workspace/CalculatorAppAsp/aspcallapp.zip"

Deploy the Zipped/Packaged App on to IIS using following batch command:

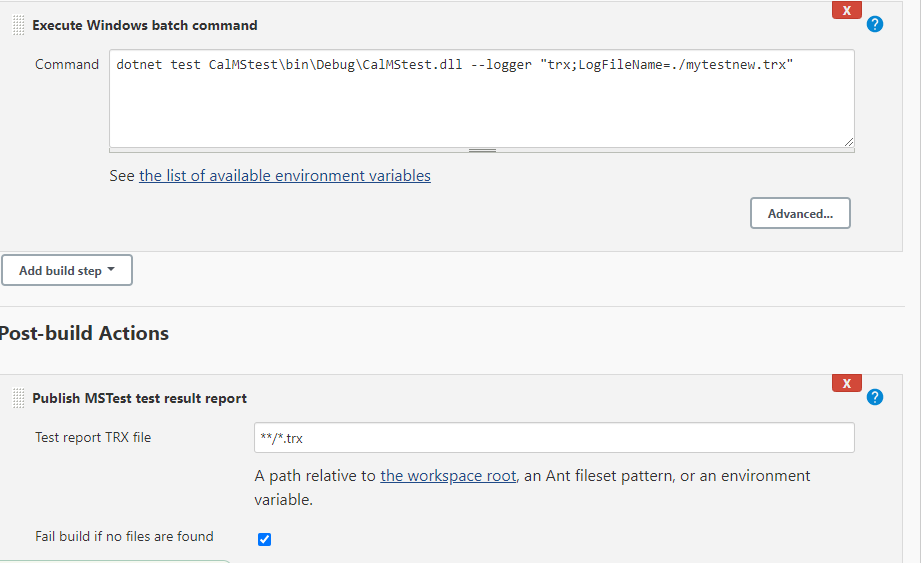
C:\"Program Files (x86)"\IIS\"Microsoft Web Deploy V3"\msdeploy.exe -verb:sync -source:package="aspcallapp.zip" -dest:auto -setParam:name="IIS Web Application Name",value="Default Web Site/aspcallapp



Run the Test Cases on the Application using dotnet test utility:

Use Publish MSTest Test Result Report Plugin.

dotnet test CalMStest\bin\Debug\CalMStest.dll –logger "trx;LogFileNme=./mytestreport.trx"



**Ansible - Configuration Management**

**Ansible Installation on Ubuntu Machine:**

sudo apt-add-repository ppa:ansible/ansible

sudo apt-get update -y

sudo apt install ansible

**Creating an Inventory for machines to be managed:**

/etc/ansible/hosts

[servers]

server1 ansible\_host=203.0.113.111

server2 ansible\_host=203.0.113.112

server3 ansible\_host=203.0.113.113

[all:vars]

ansible\_python\_interpreter=/usr/bin/python3

**ansible-inventory --list –y**

**Create a Ansible playbook to set-up** LAMP Stack on Ubuntu Machine:

Create folder on Ansible Machine: Lampansible

Change Directory to above created folder: cd Lampansible

Create file called lamp\_setup.yml

---

# Setup LAMP Stack

- hosts: lampserver

tasks:

- name: "Repo PHP 7.4"

apt\_repository:

repo="ppa:ondrej/php"

- name: install lamp stack

become: yes

become\_user: root

apt:

pkg:

- apache2

- mysql-server

- php7.2

- php7.2-mysql

state: present

update\_cache: yes

- name: start apache service

become: yes

become\_user: root

service:

name: apache2

state: started

enabled: yes

- name: start mysql service

become: yes

become\_user: root

service:

name: mysql

state: started

enabled: yes

- name: create target directory

file: path=/var/www/html state=directory mode=0755

- name: deploy index.html

become: yes

become\_user: root

copy:

src: /home/ansible/lampansible/index.html

dest: /var/www/html/index.html

Run the following command to run lamp\_setup.yml on servers listed in inventory.

**sudo ansible-playbook lamp\_setup.yml –u msis –ask-pass**

**sudo ansible-playbook lamp\_setup.yml –u msis –K**

**Create Ansible Playbook to deploy Static Website on Lamp stack:**

Download a static website from : <http://freehtml5.co>

For Example : multipurpose named website and keep code in same folder of playbook file

Create file lamp\_app\_deploy.yml

---

# Setup LAMP Stack

- hosts: lampserver

tasks:

- name: install lamp stack

become: yes

become\_user: root

apt:

pkg:

- apache2

- mysql-server

- php7.2

- php7.2-mysql

state: present

update\_cache: yes

- name: start apache service

become: yes

become\_user: root

service:

name: apache2

state: started

enabled: yes

- name: start mysql service

become: yes

become\_user: root

service:

name: mysql

state: started

enabled: yes

- name: create target directory

file: path=/var/www/html state=directory mode=0755

- name: deploy index.html

become: yes

become\_user: root

copy:

src: /home/ansible/lampansible/multipurpose/

dest: /var/www/html

**sudo ansible-playbook lamp\_app\_deply.yml –u msis –ask-pass**

**sudo ansible-playbook lamp\_app\_deply.yml –u msis –K**

**Ansible Playbook GitHub Repo:** <https://github.com/sreepathysois/ConfigurationManagment---Ansible.git>