**Client requirement:**

1. We have our product code base written in Java and built with Maven. Our developers currently build the jar and give to network team. They use a python script to check network dependencies and run a bash script to get a report.
2. Jar should be executed every 3 mins, Python Script every 5 mins and Shell script every 10 mins.

**As client cannot share the their code, we are creating the dummy/sample code as below**

1. Create a sample Java Project it should every 3mins.

2. With a sample Python Script, it should every 5mins.

3. With Sample Shell script, it should every 10mins.

4. Do continuous Integration

5. Run them in sequence

**Solution:**

Logically, given above requirement does not work properly with given timings (Java-3mins, Python-5Mins, Shell-10mins). Please find the brief explanation with timing diagram below:

a) We are missing maven (4 runs) and python (1 run) runs for shell job indicate with X in the above diagram.

b) We need to check with client what have to do with those missing runs?

c) Before completion of shell job we are completing maven 4times and python 1time is this according to the requirement

**Brief explanation**

**Step1:**

Suppose maven job first run is started at 3rd min and post the dependency after completion to the python job, so python job started at (3rd min) then after completion it will post the dependency to shell

**Step2:**

Again maven job 2nd run starts after 3 mins(6th min) and post the dependency after completion to python job, but python's/shell scripts first run are still running, so maven 2nd run dependency is not achieved.

**Step3:**

Again maven job 3rd run starts after 3 mins (9th min) and post the dependency after completion to python job, but python's 2nd run is already started / shell scripts first run is still running, so maven 3rd run dependency is not achieved.

**Step4:**

Again maven job 4th run starts after 3 mins (12th min) and post the dependency after completion to python job, but python's second run is still running/ shell scripts first run is still running, so maven 4th run dependency is not achieved.

**Step5:**

Again maven job 5th run starts after 3 mins (15th min) and post the dependency after completion to python job, but python's 2nd run is already completed but python 2nd run dependency to the shell is not achieved, still shell scripts first run is still running ,so here maven 4th run and python 2nd run

Dependency is not achieved.

**Step6**:

Again maven job 6th run starts after 3 mins(18th min) and post the dependency after completion to python job, here python's 3rd run was started and shell scripts first run completed and 2nd run is started ,so here maven 6th run and python 3rd run and shell 2nd run dependency is achieved.

Before completion of 1strun shell job we are completing maven 4runs and python

1run as per the given requirement.

Here we are missing maven (4 runs) and python (1 run) for the shell scripts 1st run.

**Possibilities**

1. To achieve given requirement i.e,( timing with dependencies )we need to modify as below:
2. Timings
   1. Change the Maven job to every 3mins to 15mins
   2. Change the Python job to every 5mins to 7mins
   3. Change the Maven job to every 10mins to 7mins

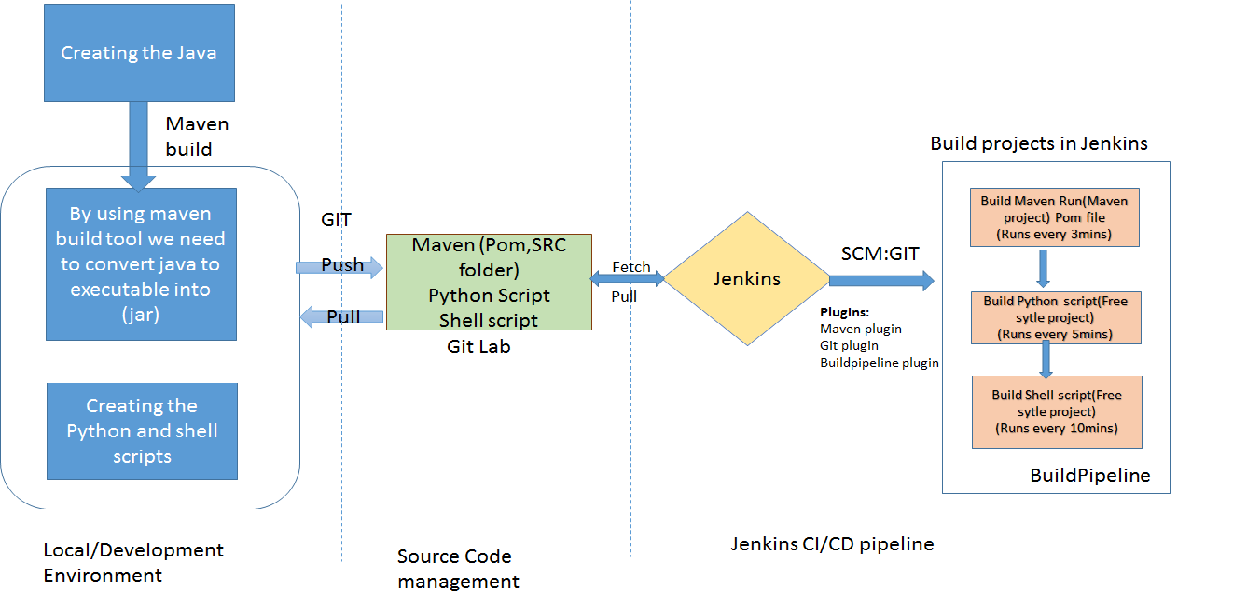
ii) We need to define the trigger (i.e., Pre-deployment conditions) to the jobs that will start deployment after completion of dependency jobs.

iii) For every jobs we need to select the option at job level properties (Run this job): only when all previous jobs have succeeded.

**Prerequisites: (Installed)**

* Java
* Python
* Maven
* Git
* Git lab(url: [**https://reposvr.tetrasoft.us/users/sign\_in**](https://reposvr.tetrasoft.us/users/sign_in))
* Jenkins(Web portal: [**https://jenkins.tetrasoft.us:8443/**](https://jenkins.tetrasoft.us:8443/))

**Flow of the project:**



**Steps for project**

a. Create the simple codes of **Java, python and shell**

b. Convert the Java project to executable jar file by using **build tool**

c. Push the code to **GITLAB**

d. Configure the **CI/CD** pipelines using **jenkins**

**a. Created of sample codes of Java, python and shell**

**Helloworld.java**

**Sample\_python.py**

**Shell\_test.sh**

**b. Convert the Java project to executable jar file**

By using maven Built tool we are making java project to executable into jar file

**mvn archetype:generate -DgroupId=com.varshika -DartifactId=HelloWorld -DarchetypeArtifactId=maven-archetype-quickstart - DinteractiveMode=false**

HelloWorld contains:

**Pom.xml**

**Src**

**mvn package command**

Pom.xml

Src

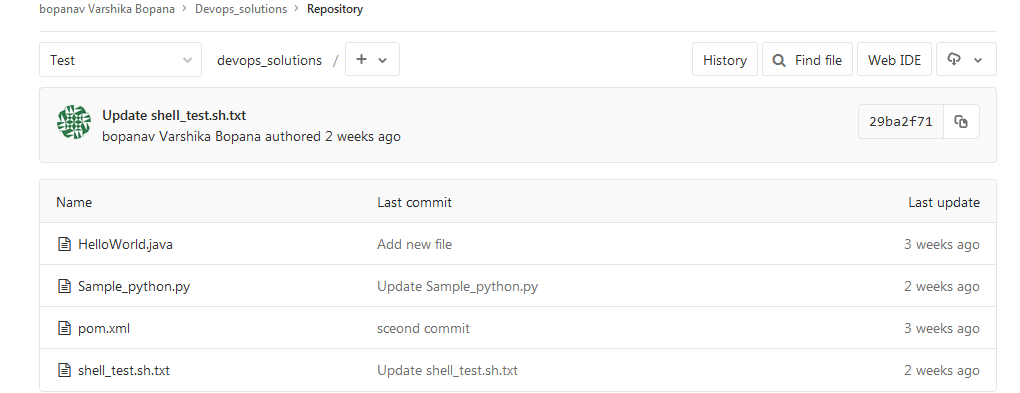
target

**c. Push the code to GITLAB**

By using the source code management tool (SCM) :**GIT**,



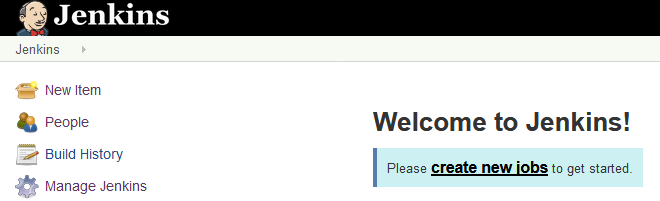
Push the code to **gitlab**



**d. Configure the CI/CD pipelines using jenkins**

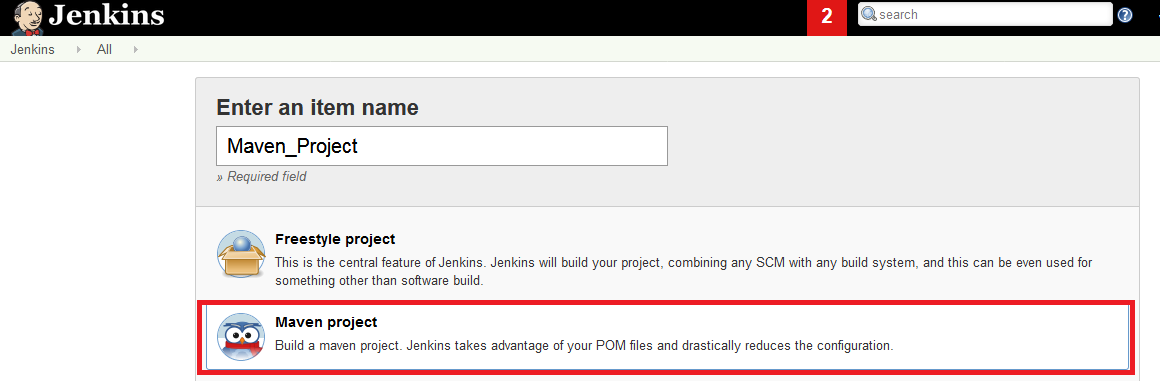
1. login to Jenkins web browser **https://jenkins.tetrasoft.us:8443**

2. Create new jobs in Jenkins(i.e, Maven project,Python project,Shell project)



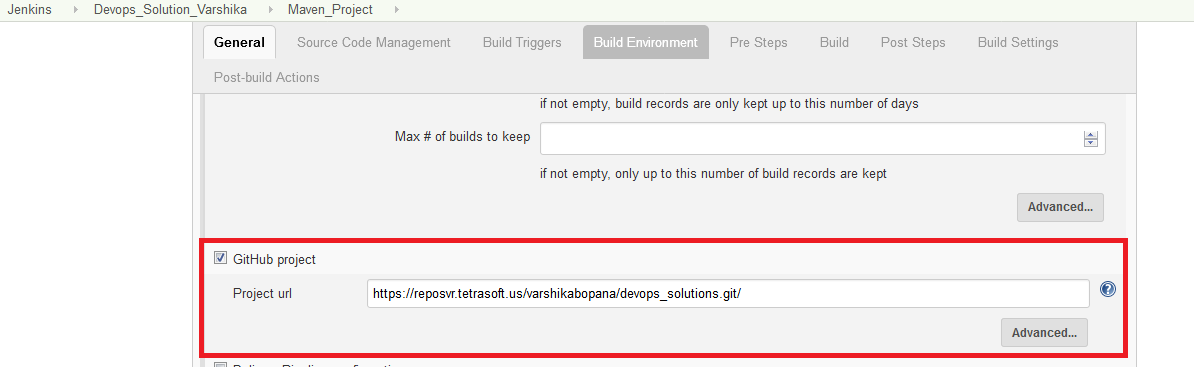
**MAVEN**

1. Select the **maven project** for **Maven**

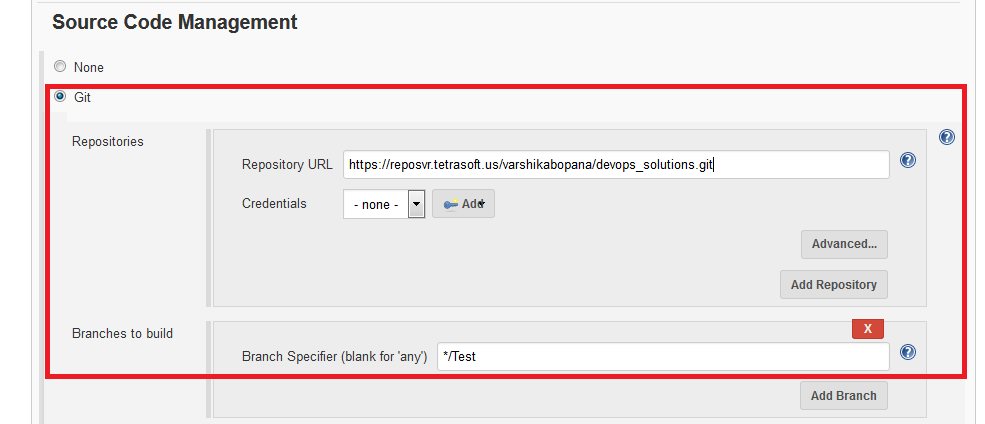


**Configure below options**

* General : **GitHub project**

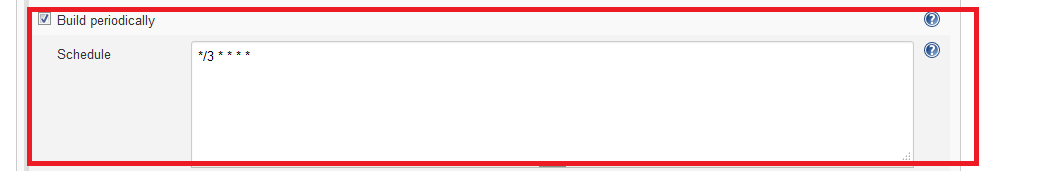


* Source Code Management : **GIT**

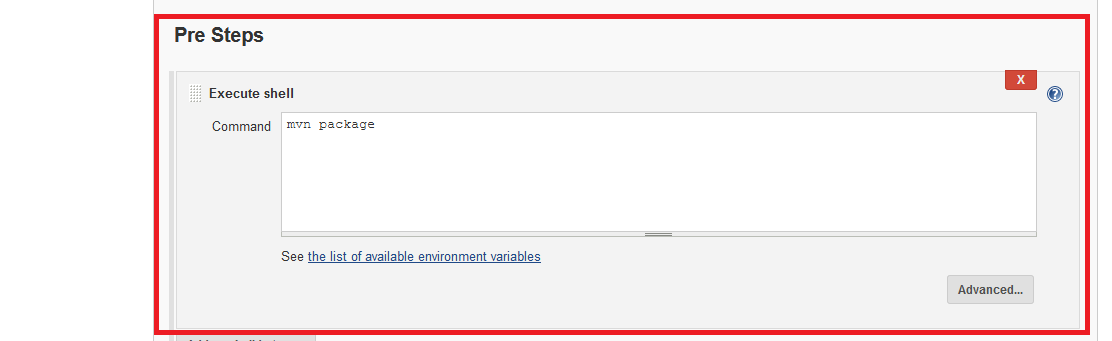
****

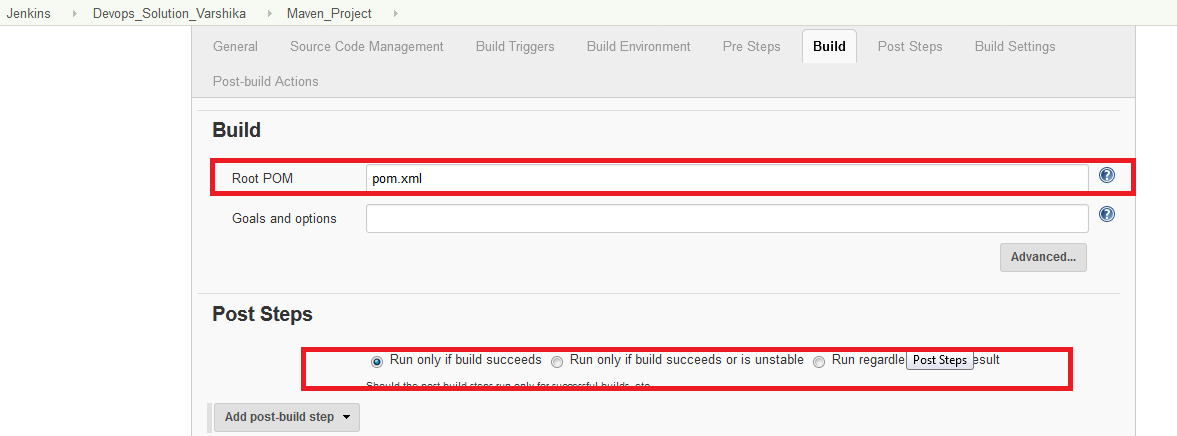
* Build Triggers

Build periodically(Check): **\*/3 \* \* \* \***



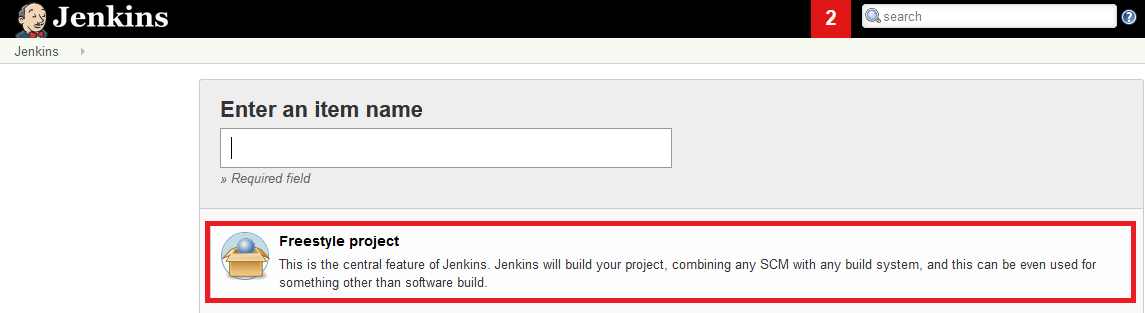
* Build : **Execute shell(**As Remote server is Linux machine)





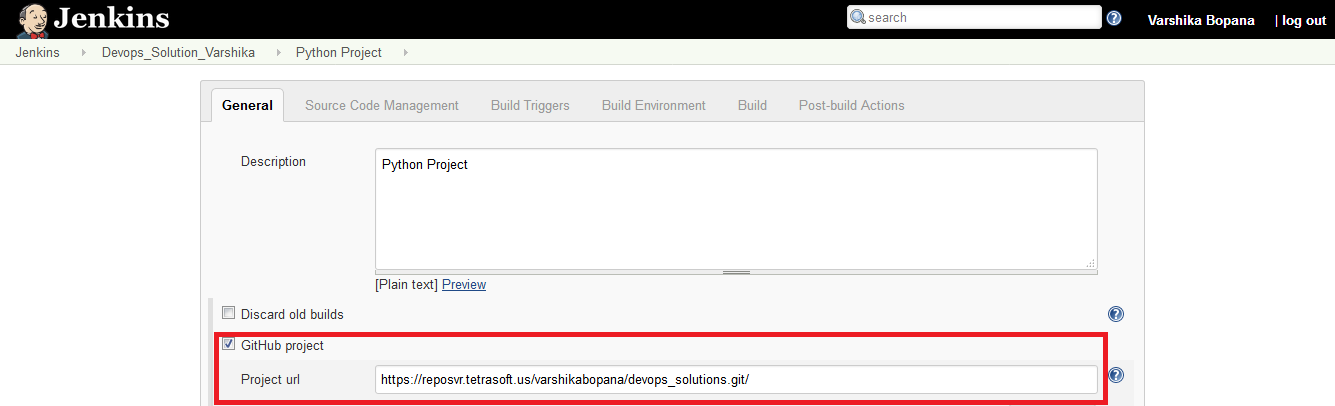
**PYTHON PROJECT**

1. Select the project type **freestyle for Python Project**

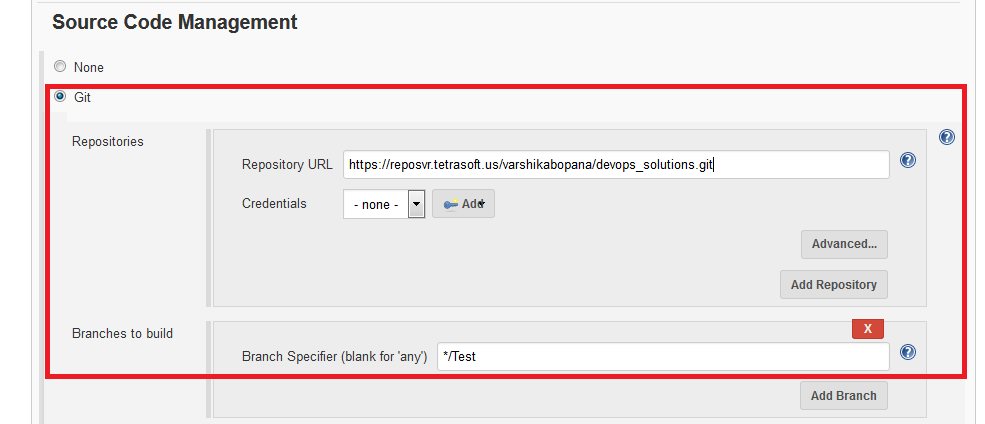


**Configure below options**

* + General : **GitHub project**



* + Source Code Management : **GIT**



* + **Build Triggers**

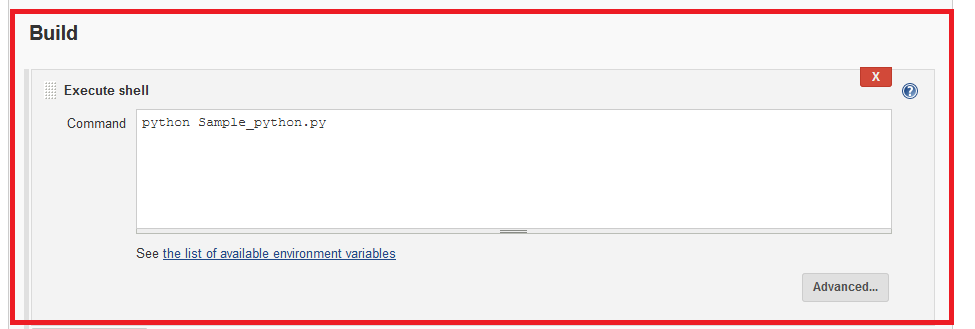
Build after other projects are built(Check): **Maven Project**

Trigger only if build is stable(Radio button)

Build periodically (Check): **\*/5\* \* \* \***

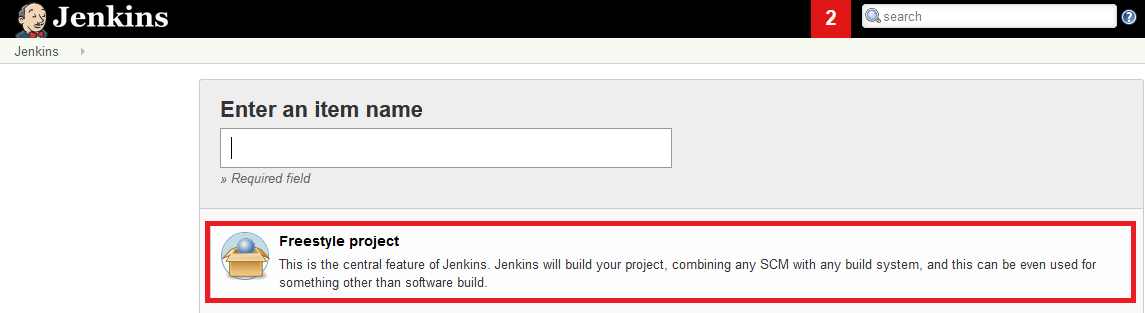


* + Build : **Execute shell**(As Remote server is Linux machine



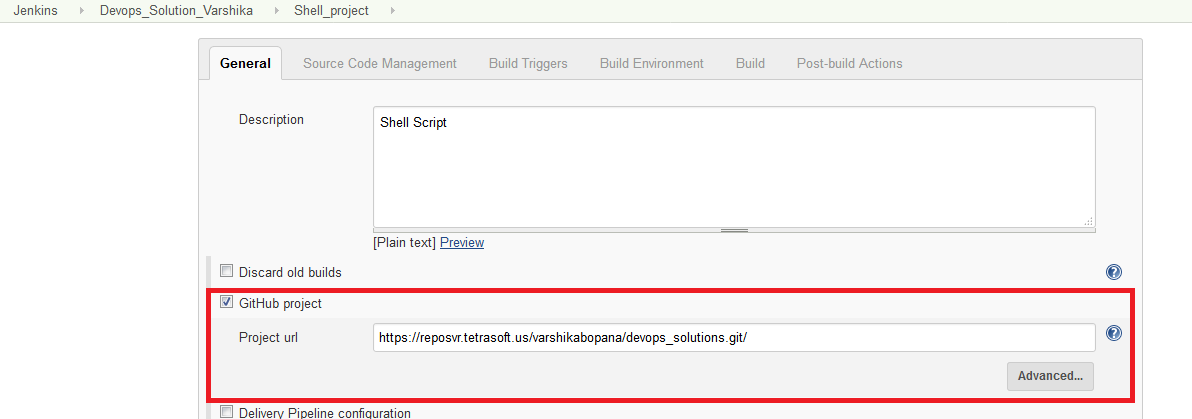
**SHELL PROJECT**

1. Select the project type **freestyle for Shell project**

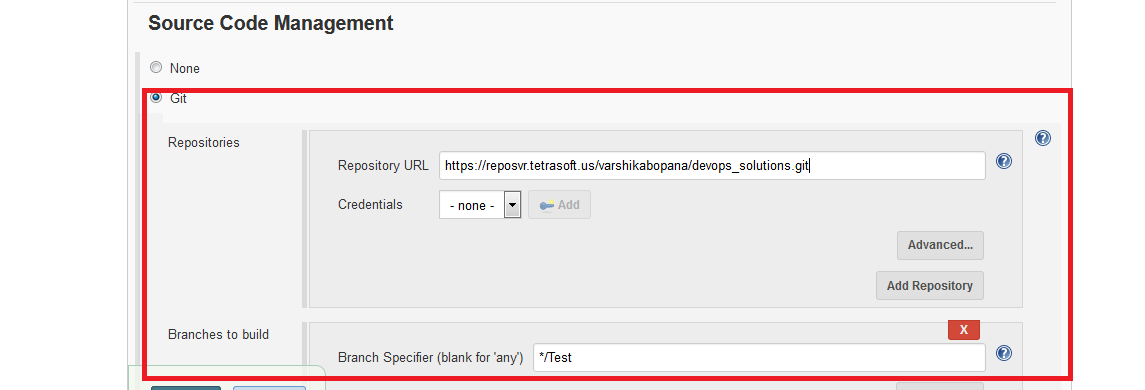
****

**i) Configure the below properties**

* General : **GitHub project**

****

* Source Code Management : **GIT**

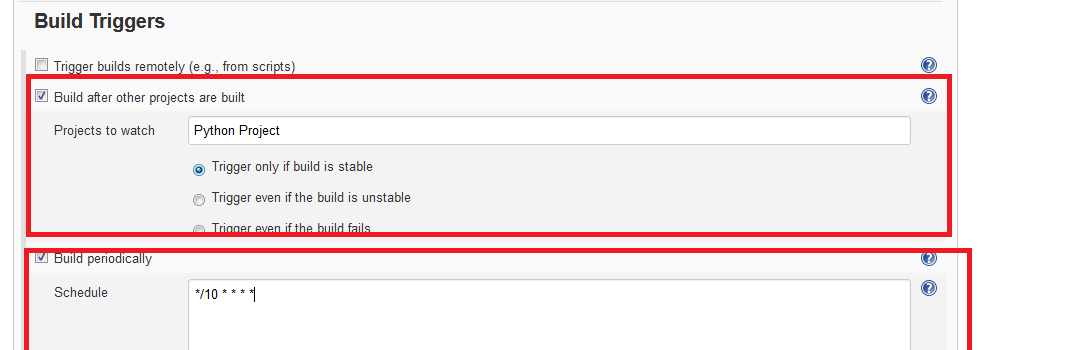


* Build Triggers

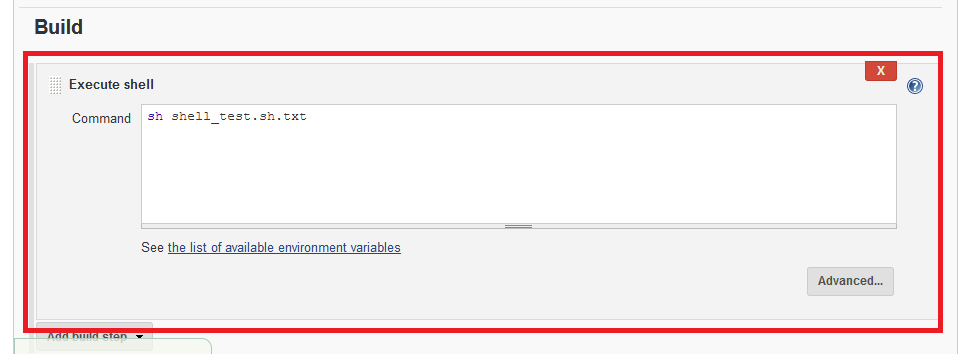
Build after other projects are built(Check): **Python Project**

Trigger only if build is stable(Radio button)

Build periodically(Check): **\*/10 \* \* \* \***

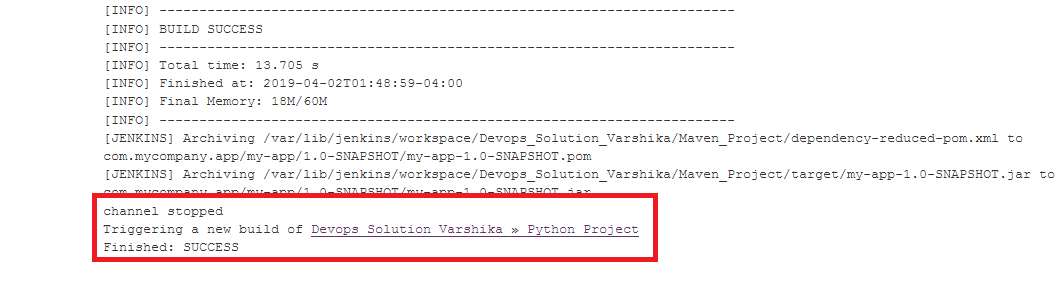


* Build **:** **Execute shell**(As Remote server is Linux machine).



**Outputs:**

* **Maven Console Output:.**



* **Python Console OutPut:**



* **Shell Console Output:**



**Dependency flow:**

