Gradle Build Tool:

- Gradle is a Build tool, it is a build upon the concept of Apache Ant and Apache Maven
- Gradle build files use a Domain Specific Languages (DSL) to define custom build login
- It supports Groovy language.
- In Gradle we can create any custom task and any login to build our projects. In Gradle we write build file which contains logic, custom method to build any Java based source code.
- High Performance- Gradle avoids unnecessary work by only running the tasks that need to run because their inputs or outputs have changes.

For example, if developers have written 3 different modules

M1- Login

M2 – Home Page

M3 – Payment Gateway

(If developer has updated payment Gateway Module to support UPI payment then through Gradle we can compile and build only M3 module)

- Google also uses Gradle as build tool for Android Studio

Official Website: https://gradle.org/

What is Gradle: https://docs.gradle.org/current/userguide/what_is_gradle.html

User guide: https://docs.gradle.org/current/userguide/userguide.html

How to install: https://docs.gradle.org/current/userguide/installation.html

Current Releases: https://gradle.org/releases/

Gradle has several fixed build phases

It's important to understand that Gradle evaluates and executes build scripts in three phases:

1. Initialization

   Sets up the environment for the build and determine which projects will take part in it.

2. Configuration

   Constructs and configures the task graph for the build and then determines which tasks need to run and in which order, based on the task the user wants to run.

3. Execution

   Runs the tasks selected at the end of the configuration phase.

Maven vs Gradle:

Maven uses XML and Gradle use DSL.

In Gradle we have concept of -

- **build.gradle :** is a file which contains actual flow.

- **Project and task:** Project is nothing but one activity or lengthy task which you want to achieve, Gradle build can have one or more projects. Each project can have multiple tasks. Each task can be responsible for some activity.

Configure Jenkins for Gradle

1. Install Gradle package on your Jenkins Instance: -

```
wget https://services.gradle.org/distributions/gradle-6.1.1-bin.zip -P /tmp

sudo unzip -d /opt/gradle /tmp/gradle-6.1.1-bin.zip
```

PATH Environment variable should include Gradle directory. So we should create gradle.sh file inside /etc/profile.d/ directory. To create file run following command:

```
sudo vi /etc/profile.d/gradle.sh
```

Now paste following code inside above file.

```
export GRADLE_HOME=/opt/gradle/gradle-6.1.1

export PATH=${GRADLE_HOME}/bin:${PATH}
```

Save the file and close the file.

Now make script file executable using following command:

```
sudo chmod +x /etc/profile.d/gradle.sh
```

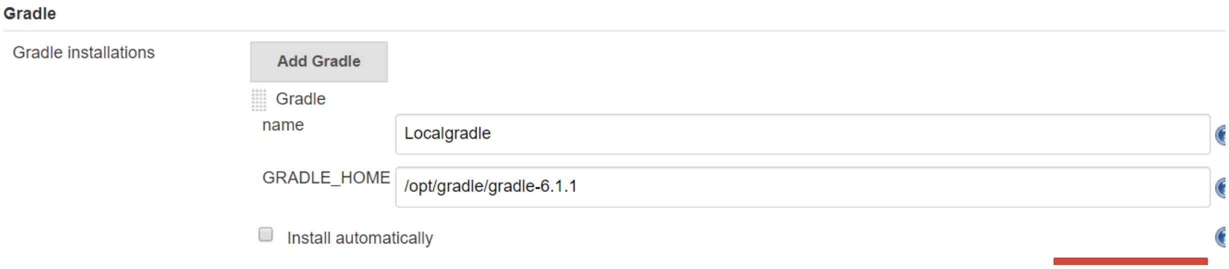
Load the environment variables using following command:

```
source /etc/profile.d/gradle.sh
```

Now verify the installation of Gradle using following command.

```
gradle -v
```

2. Configure gradle build tool under Global Tool Configuration. (refer Configure Jenkins for Gradle step 1: GRADLE_HOME=/opt/gradle/gradle-6.1.1)



Gradle

Gradle installations

Gradle

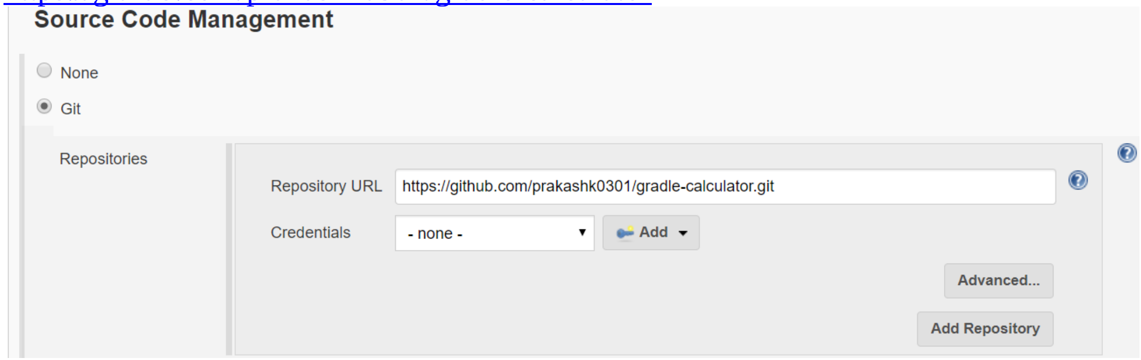
name: Localgradle

GRADLE_HOME: /opt/gradle/gradle-6.1.1

□ Install automatically

3. Create Jenkins freestyle Job. You can use Github project

https://github.com/prakashk0301/gradle-calculator



Source Code Management

○ None

● Git

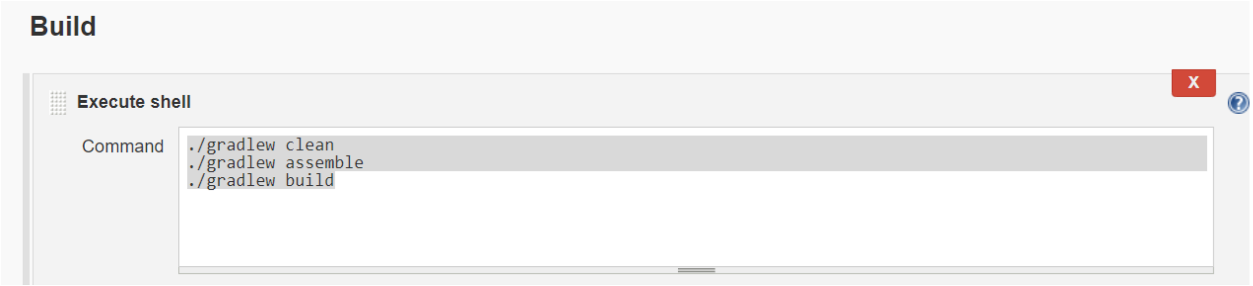
Repositories

Repository URL: https://github.com/prakashk0301/gradle-calculator.git

Credentials: - none -

4. Under the build section, please choose Execute shell-

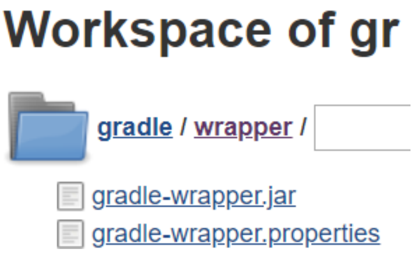
```
./gradlew clean

./gradlew assemble

./gradlew build

./ gradlew jar
```



Command

```
./gradlew clean
./gradlew assemble
./gradlew build
```

5. Save and build your job. (Gradle will assemble the project and creates a jar artifact file gradle-wrapper.jar which can be deployed)



gradle-wrapper.jar
gradle-wrapper.properties

6. Done 😊

We also can create pipeline using declarative method

Step 1: You need to create Jenkinsfile .

```
pipeline
{
agent any
stages
{
 stage ('scm checkout')
 {
    steps { git branch: 'master', url: 'https://github.com/prakashk0301/gradle-calculator/'  }
 }
 stage ('run-gradle-command-to-build-and-geneate-artifacts')

 {
   steps { sh './gradlew clean'
         sh './gradlew assemble'
         sh './gradlew build'
         sh './gradlew jar'
         }
 }


}

}
```