

Unit Testing and Test Coverage and CI

The screenshot shows two tabs in Sublime Text: 'build.gradle' and 'CouponServletTest.java'. The 'CouponServletTest.java' tab is active, displaying Java test code for a 'CouponServletTest' class using Mockito annotations.

```
3 import org.junit.Before;
4 import org.junit.Test;
5 import org.mockito.Mock;
6 import org.mockito.MockitoAnnotations;
7
8 import javax.servlet.RequestDispatcher;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import java.io.PrintWriter;
12 import java.io.StringWriter;
13
14 import static org.junit.Assert.assertEquals;
15 import static org.mockito.Mockito.*;
16
17 public class CouponServletTest {
18     @Mock private HttpServletRequest request;
19     @Mock private HttpServletResponse response;
20     @Mock private RequestDispatcher requestDispatcher;
21 }
```

The screenshot shows the same Sublime Text interface as above, with the 'CouponServletTest.java' tab active. It displays the implementation of the test methods 'doGet()' and 'doPost()'.

```
21
22     @Before
23     public void setUp() throws Exception {
24         MockitoAnnotations.initMocks(this);
25     }
26
27     @Test
28     public void doGet() throws Exception{
29         StringWriter stringWriter = new StringWriter();
30         PrintWriter printWriter = new PrintWriter(stringWriter);
31         when(response.getWriter()).thenReturn(printWriter);
32         new CouponServlet().doGet(request,response);
33         assertEquals("SUPERSALE",stringWriter.toString());
34     }
35
36     @Test
37     public void doPost() throws Exception{
38         when(request.getParameter("coupon")).thenReturn("SUPERSALE");
39         when(request.getRequestDispatcher("response.jsp")).thenReturn(requestDispatcher);
40         new CouponServlet().doPost(request,response);
41         verify(request).setAttribute("discount","Discount for coupon SUPERSALE is 50%");
42         verify(requestDispatcher).forward(request,response);
43     }
44
45 }
```

The screenshot shows the 'build.gradle' file in Sublime Text. It includes dependencies for Java, Gretty, Jacoco, and SonarQube, and a repository configuration for JCenter.

```
10     // Apply the java plugin to add support for Java
11     id 'war'
12     id 'org.gretty' version '3.0.3'
13     id 'jacoco'
14     id 'org.sonarqube' version '3.0'
15 }
16
17 repositories {
18     // Use jcenter for resolving dependencies.
19     // You can declare any Maven/Ivy/file repository here.
20     jcenter()
21 }
22 }
```

Jacoco tells Test coverage

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Ctry	Missed	Lines	Missed	Methods	Missed	Classes
CouponServlet	100%	n/a	0	3	0	7	0	3	0	3	0	1
Total	0 of 24	100%	0 of 0	n/a	0	3	0	7	0	3	0	1

sonarqube generate report to tell bugs, and vulnerabilities and etc

firstWebApp Passed

Last analysis: September 4, 2020, 7:15 AM

0 A Bugs | 2 B Vulnerabilities | - A Hotspots Reviewed | 2 A Code Smells | 0.0% Coverage | 0.0% Duplications

1 of 1 shown

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support migrating your data out of it into a different database engine.

```
// https://mvnrepository.com/artifact/junit/junit
testImplementation group: 'junit', name: 'junit',
version: '4.4'
```

It is no more testCompile now - it became TestImplementation

```

4 * This generated file contains a sample Java project to get you started.
5 * For more details take a look at the Java Quickstart chapter in the Gradle
6 * User Manual available at https://docs.gradle.org/6.5.1/userguide/tutorial_java_projects.html
7 */
8
9 plugins {
10     // Apply the java plugin to add support for Java
11     id 'war'
12     id 'org.gretty' version '3.0.3'
13 }
14
15 repositories {
16     // Use Jcenter for resolving dependencies.
17     // You can declare any Maven/Ivy/file repository here.
18     jcenter()
19 }
20
21 dependencies {
22     providedCompile 'javax.servlet:javax.servlet-api:3.1.0'
23     compile group: 'com.fasterxml.jackson.core', name: 'jackson-core', version: '2.11.2'
24     testCompile group: 'junit', name: 'junit', version: '4.4'
25     testCompile group: 'org.mockito', name: 'mockito-core', version: '3.5.9'
26
27 }
28

```

Add line 24 and 25

Line number 24 is outdated one use TestImplementation instead

```

tasks - Displays the tasks runnable from root project 'firstWebApp'.
-----
Verification tasks
-----
check - Runs all checks.
jacocoTestCoverageVerification - Verifies code coverage metrics based on specified rules for the test task.
jacocoTestReport - Generates code coverage report for the test task.
test - Runs the unit tests.

```

If we have failure in our test results and you tried multiple times below command. It won't generate a jacoco report.

```
nthippireddy$ gradle clean build jTR
```

It wont work. Add the below line 34.

```

27
28
29 }
30
31 gretty.httpPort=8090
32
33
34 test.finalizedBy jacocoTestReport;
35 jacocoTestReport.dependsOn test

```

If we have line number 35 whenever we run jacocoTestReport task it always runs the task test.

```

bharath@MacBook-Pro:firstWebApp bharaththippireddy$ gradle clean build jTR
> Task :compileTestJava
Note: /Users/bharaththippireddy/Documents/gradle/firstWebApp/src/test/java/com/bharath/gradle/CouponServletTest.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

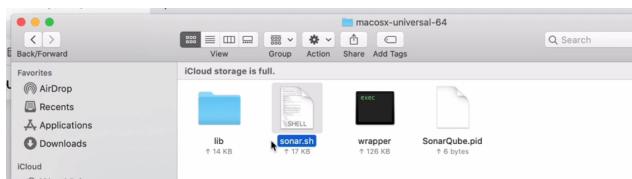
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.5.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 1s
5 actionable tasks: 5 executed

```

Install and launch Sonar

The screenshot shows the SonarQube download page. At the top, there's a navigation bar with links like 'Product', 'What's New', 'Documentation', and 'Community'. Below the header, there's a section for 'FREE & OPEN SOURCE' featuring the 'Community EDITION'. It describes it as 'The starting point for adopting code quality in your CI/CD' and includes a 'Download Community Edition' button. To the right, there's a 'Developer EDITION' section with the tagline 'Maximum Application Security' and 'Maximum value across branches & PRs', which includes a 'Request a Free Trial License' button and a 'Download Developer Edition' button.



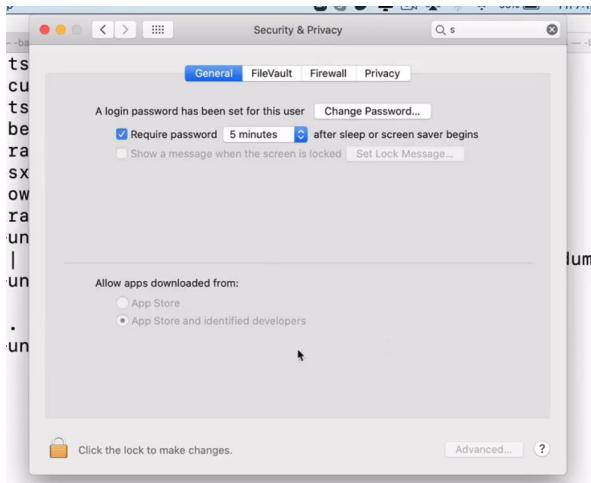
To run sonarqube in your local.

```

bharaths-MacBook-Pro:Documents bharaththippireddy$ pwd
/Users/bharaththippireddy/Documents
bharaths-MacBook-Pro:Documents bharaththippireddy$ cd sonarqube-8.4.2.36762/
bharaths-MacBook-Pro:sonarqube-8.4.2.36762 bharaththippireddy$ cd bin
bharaths-MacBook-Pro:bin bharaththippireddy$ ls
jsw-license          macosx-universal-64
linux-x86-64         windows-x86-64
bharaths-MacBook-Pro:bin bharaththippireddy$ cd macosx-universal-64/
bharaths-MacBook-Pro:macosx-universal-64 bharaththippireddy$ ./sonar.sh
Usage: ./sonar.sh { console | start | stop | force-stop | restart | status | dump }
bharaths-MacBook-Pro:macosx-universal-64 bharaththippireddy$ ./sonar.sh start
Starting SonarQube...
SonarQube is already running.
bharaths-MacBook-Pro:macosx-universal-64 bharaththippireddy$ 

```

Incase of any issue in running sonarqube, allow it from the window below.



My Favorites All

Perspective: Overall Status Sort by:

Filters

Quality Gate

Passed: 1 | Failed: 0

firstWebApp Passed

Last analysis: September 2, 2020, 7:01 PM

0 A Bugs | 2 B Vulnerabilities | - A Hotspots Review

```

1 Tone For Sonar:
2
3 My Account -> Security
4
5 gradle sonar -Dsonar.login={token_from_my-account->security}
6
7 Example:
8
9 gradle sonar -Dsonar.login=d9485adba92198844074498dd8082c40808e0zbb

```

```

1 gradle sonar -Dsonar.login={your_generated_token}
2
3
4 sonar {
5     properties {
6         property "sonar.token", "{your_generated_token}"
7     }
8 }

```

Line 4 to 7 in build.gradle file

FIRST, INCLUDE THE SCANNER IN YOUR BUILD IN build.gradle

```

plugins {
    id "org.sonarqube" version "3.0"
}

```

More details on <https://plugins.gradle.org/plugin/org.sonarqube>

```

6   * User manual available at https://docs.gradle.org/3.1/userguide
7   */
8
9 plugins {
10     // Apply the java plugin to add support for Java
11     id 'war'
12     id 'org.gretty' version '3.0.3'
13     id 'jacoco'
14     id 'org.sonarqube' version '3.0'
15 }
16

```

```

firstWebApp -- bash 90x23
~/Documents/gradle/firstWebApp --bash
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$ pwd
/Users/bharaththippireddy/Documents/gradle/firstWebApp
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$ gradle sonarqube
Starting a Gradle Daemon (subsequent builds will be faster)

> Task :sonarqube
Missing blame information for the following files:
 * src/main/java/com/bharath/gradle/CouponServlet.java
 * src/test/java/com/bharath/gradle/CouponServletTest.java
This may lead to missing/broken features in SonarQube

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0
.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.5.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 50s
3 actionable tasks: 1 executed, 2 up-to-date
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$ 

```

It give all code smells and other summary.

The screenshot shows the SonarQube interface with the following details:

- Project:** firstWebApp (master branch)
- Last Analysis:** September 4, 2020, 7:15 AM, Version unspecified
- Issues Tab:** Selected, showing 2/4 issues (50 minutes old).
- Issues List:**
 - Add the '@Override' annotation above this method signature (Why is this an issue? 2 days ago L12%)
 - Handle the following exception that could be thrown by "getWriter": IOException. (Why is this an issue? 2 days ago L14%)

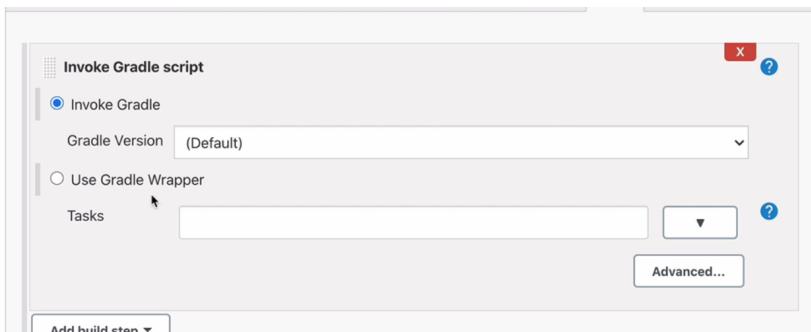
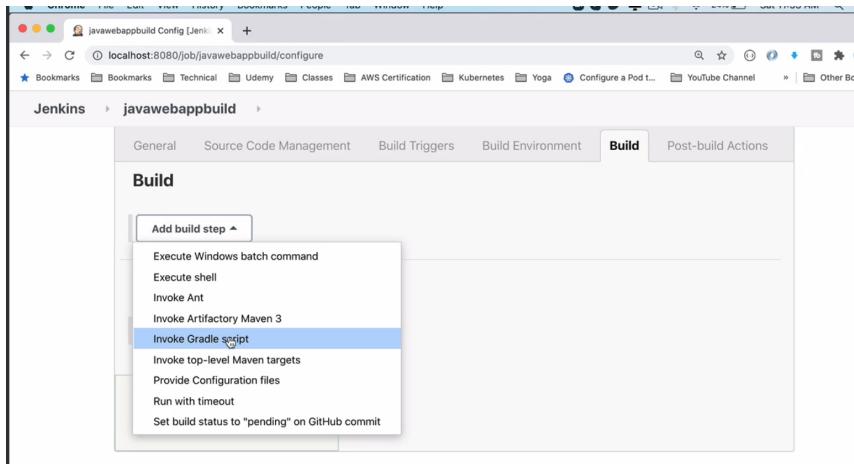
Configure a jenkins gradle build

The screenshot shows the Jenkins Plugin Manager with the following details:

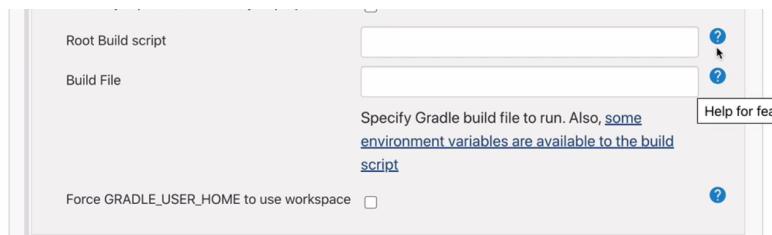
- Plugin:** Gradle (version 1.36)
- Description:** This plugin allows Jenkins to invoke Gradle build scripts directly.
- Install:** Uninstall button is visible.

The screenshot shows the Jenkins Global Tool Configuration for Gradle with the following details:

- Gradle:** A new installation is being configured.
- Name:** GRADLE
- GRADLE_HOME:** /usr/local/Cellar/gradle/6.5.1
- Install automatically:** Unchecked checkbox.
- Buttons:** Add Gradle, Delete Gradle, Save, Apply.



In the advance tab we may need specify root build script and project specific build script.



The screenshot shows the Jenkins job configuration page for 'javawebappbuild'. The 'Build' tab is selected. Under 'Root Build script', the path 'ppreddy/Documents/gradle/firstWebApp/build.gradle' is specified. A note below says: 'Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)'. There are other tabs like General, Source Code Management, Build Triggers, Build Environment, and Post-build Actions.

Or we just clean and build as shown below.

The screenshot shows the 'Build' configuration dialog. It has a single step: 'Invoke Gradle script'. The 'Invoke Gradle' option is selected, and the 'Gradle Version' dropdown is set to 'GRADLE'. The 'Tasks' field contains 'clean build'. There is an 'Advanced...' button at the bottom right.

And run the build. You will see gradle command and other task and so on.

The screenshot shows the Jenkins console output for build #1. The log starts with 'Started by user test' and 'Running as SYSTEM'. It shows the workspace is '/Users/bharaththippireddy/.jenkins/workspace/javawebappbuild'. The log then lists the executed Gradle tasks: 'Task :clean', 'Task :compileJava', 'Task :processResources NO-SOURCE', 'Task :classes', 'Task :war', 'Task :assemble', 'Task :compileTestJava', 'Note: /Users/bharaththippireddy/Documents/gradle/firstWebApp/src/test/java/com/bharath/gradle/CouponServletTest.java uses or overrides a deprecated API. Note: Recompile with -Xlint:deprecation for details.', 'Task :processTestResources NO-SOURCE', 'Task :testClasses', 'Task :test', and 'Task :jacocoTestReport'.

Good job!

Question 1:

The tests should always be run for Jacoco to generate the reports

True

Good job!

Question 2:

Which of the following task will generate a Jacoco Test Report

jTR

Good job!

Question 3:

Which tasks runs the sonar scanner to scan our code and create a report when we are using sonar plugin

- sonarscanner
- sonarscan
- sonardo
- sonar

Good job!

Question 4:

To use Gradle in jenkins we should install the gradle plugin

True

Different task of spring boot plugin. And the version we are seeing at line number 3 is the gradle parent version. And line 3 will pull what are all the dependencies (line number 16 and 17 and soon) version we need to pull in order to sync with parent version 2.3.3

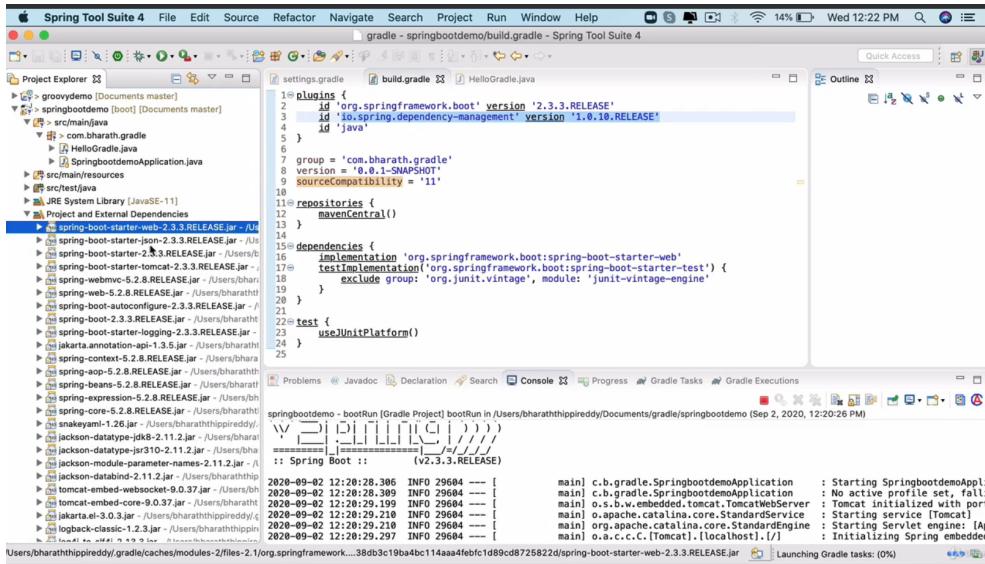
The screenshot shows an IDE interface with two main panes. The top pane displays the contents of a build.gradle file. The bottom pane is a Problems view, which lists various build tasks and their descriptions.

build.gradle

```
1plugins {
2    id 'org.springframework.boot' version '2.3.3.RELEASE'
3    id 'io.spring.dependency-management' version '1.0.10.RELEASE'
4    id 'java'
5 }
6
7 group = 'com.bharath.gradle'
8 version = '0.0.1-SNAPSHOT'
9 sourceCompatibility = '11'
10
11 repositories {
12     mavenCentral()
13 }
14
15 dependencies {
16     implementation 'org.springframework.boot:spring-boot-starter-web'
17     testImplementation 'org.springframework.boot:spring-boot-starter-test'
18 }
```

Problems

Name	Description
init	Initializes a new Gradle build.
wrapper	Generates Gradle wrapper files.
build	Assembles the outputs of this project.
assemble	Builds an OCI image of the application using the output of the bootJar task.
bootBuildImage	Assembles an executable jar archive containing the main classes and their dependencies.
bootJar	Assembles an executable jar archive containing the main classes and their dependencies.
build	Assembles and tests this project.
buildDepends	Assembles and tests this project and all projects that depend on it.
buildNeeded	Assembles and tests this project and all projects it depends on.
classes	Assembles main classes.
clean	Deletes the build directory.
jar	Assembles a jar archive containing the main classes.
testClasses	Assembles test classes.
documentation	Generates Javadoc API documentation for the main source code.
javadoc	Generates Javadoc API documentation for the main source code.
help	Displays all buildscript dependencies declared in root project 'springbootgradle'.
buildEnvironment	Displays all buildscript dependencies declared in root project 'springbootgradle'.



Its equivalent maven tags

Line 2 of gradle

```

13
14      <parent>
15          <groupId>org.springframework.boot</groupId>
16          <artifactId>spring-boot-starter-parent</artifactId>
17          <version>2.0.6.RELEASE</version>
18          <relativePath/> <!-- lookup parent from repository -->
19      </parent>
20

```

Line number 3 of gradle

```

39
40      <build>
41          <plugins>
42              <plugin>
43                  <groupId>org.springframework.boot</groupId>
44                  <artifactId>spring-boot-maven-plugin</artifactId>
45              </plugin>
46          </plugins>
47      </build>
48

```

How to migrate from maven to gradle

Just go to pom.xml file and run gradle init as shown below. It identify it's maven project ask for the option.

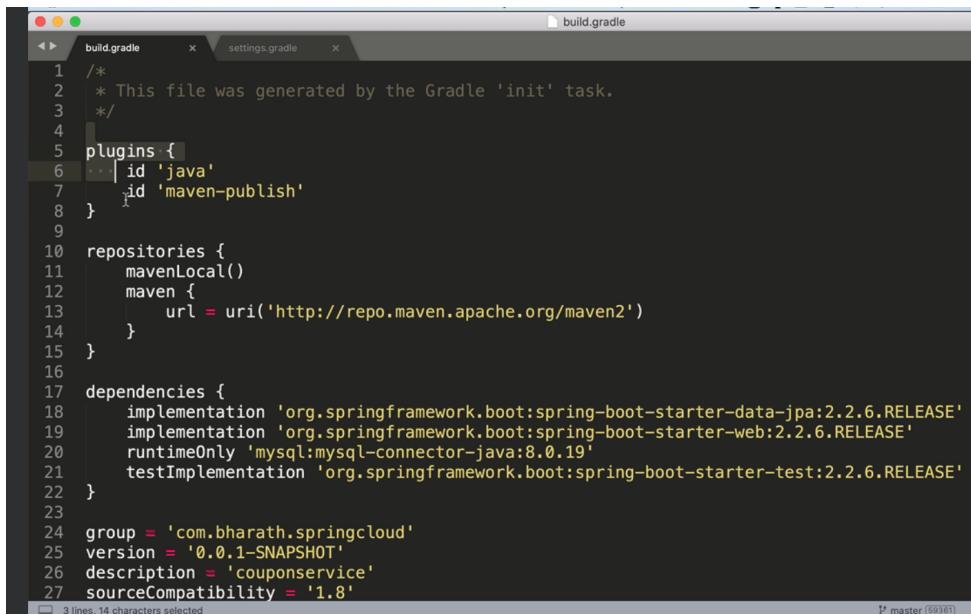
```
bharaths-MacBook-Pro:couponservice bharaththippireddy$ pwd
/Users/bharaththippireddy/Documents/gradle/couponservice
bharaths-MacBook-Pro:couponservice bharaththippireddy$ gradle init

Found a Maven build. Generate a Gradle build from this? (default: yes) [yes, no] yes

> Task :init
Maven to Gradle conversion is an incubating feature.
Get more help with your project: https://docs.gradle.org/6.5.1/userguide/migrating_from_maven.html

BUILD SUCCESSFUL in 11s
2 actionable tasks: 2 executed
bharaths-MacBook-Pro:couponservice bharaththippireddy$
```

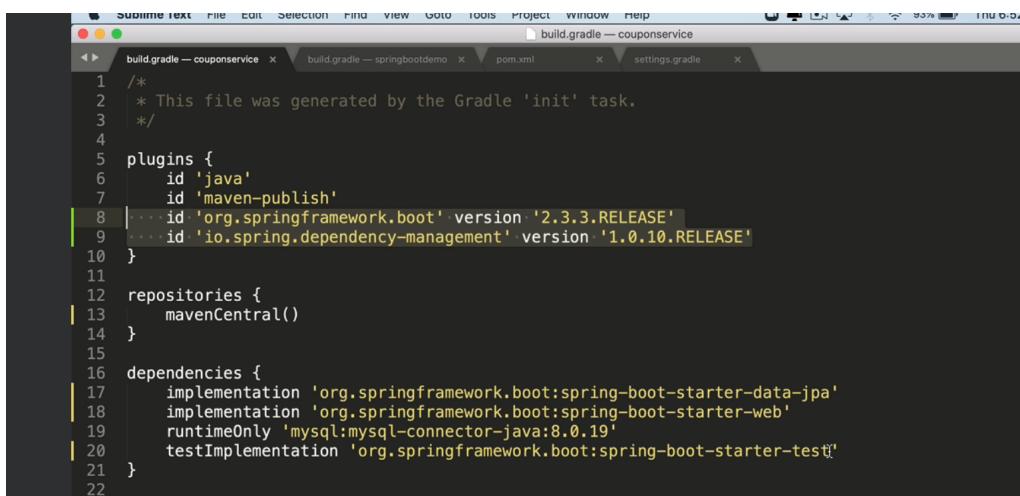
It generates build file like this



```
build.gradle settings.gradle build.gradle
1 /*
2  * This file was generated by the Gradle 'init' task.
3  */
4
5 plugins {
6     id 'java'
7     id 'maven-publish'
8 }
9
10 repositories {
11     mavenLocal()
12     maven {
13         url = uri('http://repo.maven.apache.org/maven2')
14     }
15 }
16
17 dependencies {
18     implementation 'org.springframework.boot:spring-boot-starter-data-jpa:2.2.6.RELEASE'
19     implementation 'org.springframework.boot:spring-boot-starter-web:2.2.6.RELEASE'
20     runtimeOnly 'mysql:mysql-connector-java:8.0.19'
21     testImplementation 'org.springframework.boot:spring-boot-starter-test:2.2.6.RELEASE'
22 }
23
24 group = 'com.bharath.springcloud'
25 version = '0.0.1-SNAPSHOT'
26 description = 'couponservice'
27 sourceCompatibility = '1.8'
```

3 lines, 14 characters selected master (5936)

Add plugins and remove versions. Then you are ready to go.



```
build.gradle -- couponservice build.gradle -- springbootdemo pom.xml settings.gradle
1 /*
2  * This file was generated by the Gradle 'init' task.
3  */
4
5 plugins {
6     id 'java'
7     id 'maven-publish'
8     id 'org.springframework.boot' version '2.3.3.RELEASE'
9     id 'io.spring.dependency-management' version '1.0.10.RELEASE'
10 }
11
12 repositories {
13     mavenCentral()
14 }
15
16 dependencies {
17     implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
18     implementation 'org.springframework.boot:spring-boot-starter-web'
19     runtimeOnly 'mysql:mysql-connector-java:8.0.19'
20     testImplementation 'org.springframework.boot:spring-boot-starter-test'
```

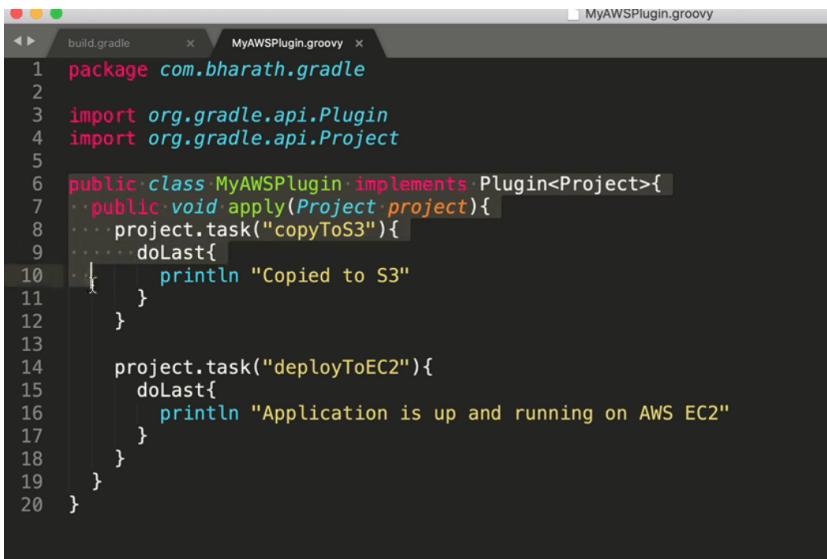
Create plugins : Plugin is nothing but group of task

```

8
9   import com.bharath.gradle.MyAWSPlugin
10
11  plugins {
12      // Apply the java plugin to add support for Java
13      id 'war'
14      id 'org.gretty' version '3.0.3'
15      id 'jacoco'
16      id 'org.sonarqube' version '3.0'
17 }

```

Every user defined plugin class must implements the plugin interface of type project as shown below.



```

1 package com.bharath.gradle
2
3 import org.gradle.api.Plugin
4 import org.gradle.api.Project
5
6 public class MyAWSPlugin implements Plugin<Project>{
7     public void apply(Project project){
8         project.task("copyToS3"){
9             doLast{
10                 println "Copied to S3"
11             }
12         }
13
14         project.task("deployToEC2"){
15             doLast{
16                 println "Application is up and running on AWS EC2"
17             }
18         }
19     }
20 }

```

How to apply in build.gradle and how use provided below

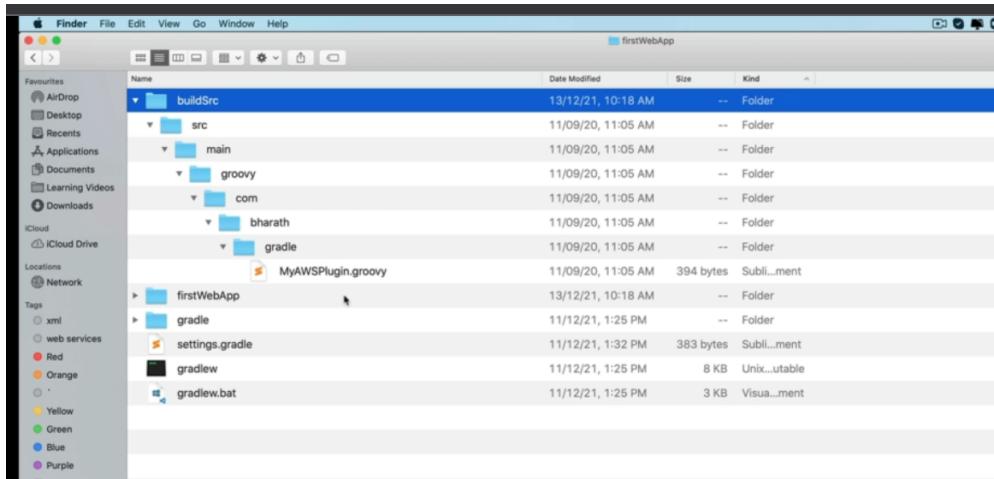
```

29     testCompile group: 'org.mockito', name:
30
31
32 }
33
34 gretty.httpPort=8090
35
36
37
38
39 apply plugin: MyAWSPlugin
40
41
42 test.finalizedBy jacocoTestReport
43 jacocoTestReport.dependsOn test
44
45 deployToEC2.dependsOn copyToS3

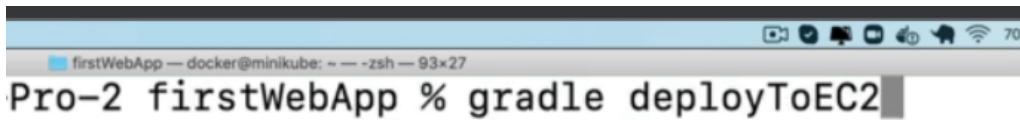
```

11 characters selected

Where to create that plugin file



Use complete name of task in recent version of gradle short names are not working.



We can also create plugin build.gradle and apply it as shown below.

```

33
34 public class MyAWSPlugin implements Plugin<Project>{
35     public void apply(Project project){
36         project.task("copyToS3"){
37             doLast{
38                 println "Copied to S3"
39             }
40         }
41
42         project.task("deployToEC2"){
43             doLast{
44                 println "Application is up and running on AWS EC2"
45             }
46         }
47     }
48 }
49
50
51
52
53 apply plugin: MyAWSPlugin
54
55

```

```

App bharaththippireddy$ pwd
/Users/bharaththippireddy/Documents/gradle/firstWebApp
App bharaththippireddy$ gradle tasks --all

```

```
Other tasks
-----
compileJava - Compiles main Java source.
compileTestJava - Compiles test Java source.
copyToS3
deployToEC2
prepareKotlinBuildScriptModel
```

```
and_line_warnings

BUILD SUCCESSFUL in 734ms
1 actionable task: 1 executed
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$ gradle -q cTS
Copied to S3
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$ gradle -q dTE
Application is up and running on AWS EC2
```

Introduce line number 59

```
49
50
51
52
53 apply plugin: MyAWSPlugin
54
55
56 test.finalizedBy jacocoTestReport
57 jacocoTestReport.dependsOn test
58
59 deployToEC2.dependsOn copyToS3
60
61
```

```
Application is up and running on AWS EC2
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$ gradle -q dTE
Copied to S3
Application is up and running on AWS EC2
bharaths-MacBook-Pro:firstWebApp bharaththippireddy$
```

In case if you want maintain separate file

You need to keep import statements and the class name should match with the file name. These are not required incase of build.gradle.

The screenshot shows the Sublime Text editor with two tabs open: 'build.gradle' and 'MyAWSPlugin.groovy'. The 'MyAWSPlugin.groovy' tab contains the following Groovy code:

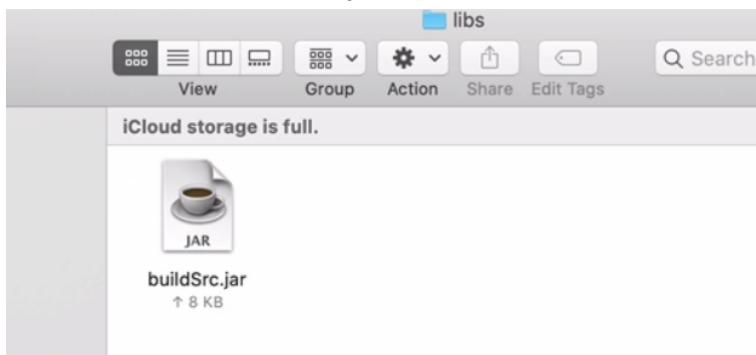
```
1 package com.bharath.gradle
2
3 import org.gradle.api.Plugin
4 import org.gradle.api.Project
5
6 public class MyAWSPlugin implements Plugin<Project>{
7     public void apply(Project project){
8         project.task("copyToS3"){
9             doLast{
10                 println "Copied to S3"
11             }
12         }
13
14         project.task("deployToEC2"){
15             doLast{
16                 println "Application is up and running on AWS EC2"
17             }
18         }
19     }
20 }
```

And we have to import this class in build.gradle

The screenshot shows the 'build.gradle' file with the following content:

```
7 */
8
9 import com.bharath.gradle.MyAWSPlugin
10
11 plugins {
12     // Apply the java plugin to add support for Java
13     id 'war'
14     id 'org.gretty' version '3.0.3'
```

In libs folder we can find a jar file for this.



 Good job!

Question 1:

Which of the following is a correct Plugin implementation

public class MyAWSPlugin implements Project<Plugin>

public class MyAWSPlugin implements Plugin<Project>

public class MyAWSPlugin implements Plugin<Task>

public class MyAWSPlugin implements GradlePlugin<Project>

 Good job!

Question 2:

What is the name of the folder under our project where we put our plugin code

src

build

buildSrc

plugins

 Good job!

Question 1:

Gradle use convention over configuration

True

False

 Good job!

Question 2:

We use which of the following to write a gradle build file

xml

json

groovy or kotlin

 Good job!

Question 3:

Gradle is super fast because it does which of the following

Incremental Builds

Build Cache

Daemon Process

All the above

 Good job!

Question 4:

Which of the following plugin gives tasks to run the java application

Java

run

application

main

 Good job!

Question 5:

The java compile task is added by which of the following plugin

compile

jvm

application

java

 Good job!

Question 6:

What should be the dependency scope for junit

runtime

implementation

testcompile

testimplementation

 Good job!

Question 7:

Gradle will run the compile task every-time even if the code has not changed

True

False

 Good job!

Question 8:

GStrings can be initialized using which of the following

Single Quotes

//

triple single quotes

double quotes

 Good job!

Question 9:

Which of the following holds the value of the parameter passed to a closure

This

That

Var

it

 Good job!

Question 10:

Closures are similar to which of the following in java

classes

interfaces

lambdas

abstract classes

 Good job!

Question 11:

Which of the following is used to define a List type

()

[]

@()

<>

 Good job!

Question 12:

Which of the following should be used between each key and value in a map

,

:

;

/

 Good job!

Question 13:

Groovy supports named parameters using which of the following

lists

sets

tree

map

 Good job!

Question 14:

Every build.gradle has an object of which of the following Gradle api classes

Build

Task

Settings

Project

Good job!

Question 15:

Which of the following should be used to add custom property called abc on the gradle project object

- project.props.abc
- project.abc
- project.abc.ext
- project.ext.abc

Good job!

Question 16:

Which of the following method in a task will be run during the execution phase

- execute
- doFirst
- doLast
- last

GAP Utilities Good job!

Question 17:

Using which of the below can we always run a task after another task runs

- dependsOn
- finalizedBy
- after
- afterTask

Good job!

Question 18:

Which plugin can be used to run a web application

- War
- Tomcat
- Jetty
- gretty

Good job!

Question 19:

The war plugin also will bring in the java plugin tasks

- True
- False

Good job!

Question 20:

Which task can be used to run the web application when using gretty plugin

- runApp
- run
- app
- apprun

Good job!

Question 21:

Which of the following is the correct syntax to declare child projects in a parents settings.gradle

- include 'web';'services';'data';'integration'
- rootProject.name = 'couponapp'
- include 'web';'services';'data';'integration'

Good job!

Question 22:

Which of following methods will apply build logic across projects including the root project

- subprojects
- projects
- someprojects
- allprojects

Good job!

Question 23:

We can control the dependencies of a child project or module from the parent projects build.gradle

- True
- False



Good job!

Question 24:

The tests should always be run for Jacoco to generate the reports



True



False



Good job!

Question 25:

Which of the following task will generate a Jacoco Test Report



jTR



Good job!

Question 26:

Which tasks runs the sonar scanner to scan our code and create a report when we are using sonar qube plugin



sonarscanner



sonarscan



sonar



sonarqube



Question 27:

To use Gradle in jenkins we should install the gradle plugin



True



False



Question 28:

Which of the following is a correct Plugin implementation



public class MyAWSPlugin implements Project<Plugin>



public class MyAWSPlugin implements Plugin<Project>