Experiment 3 - Working with Gradle: Setting Up a Gradle Project, Understanding Build Scripts (Groovy and Kotlin DSL), Dependency Management and Task Automation

Working with Gradle: Setting Up a Gradle Project, Understanding Build Scripts (Groovy and Kotlin DSL), Dependency Management and Task Automation

Introduction to Gradle

Gradle is a powerful build automation tool used for Java, Kotlin, and Android projects. It provides fast builds, dependency management, and flexibility using Groovy/Kotlin-based scripts.

Objective:

Gradle is a powerful build automation tool used for Java, Kotlin, and Android projects. It simplifies the build process, manages project dependencies, and supports plugins for different tasks. It uses XML files (called pom.xml) for project configuration and dependency management.

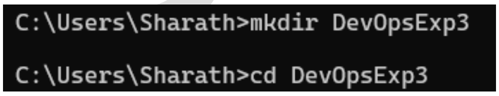
Why Use Gradle?

- **Incremental builds (faster execution)**
- **Dependency management (supports Maven/Ivy)**
- **Customizable tasks**
- **Multi-project support Key Features of Maven:**
- **Project Management**: Handles project dependencies, configurations, and builds.
- **Standard Directory Structure**: Encourages a consistent project layout across Java projects.
- **Build Automation**: Automates tasks such as compilation, testing, packaging, and deployment.
- **Dependency Management**: Downloads and manages libraries and dependencies from repositories (e.g., Maven Central).
- **Plugins**: Supports many plugins for various tasks like code analysis, packaging, and deploying.

To understand build automation tools, compare **Maven** and **Gradle**, and set up both tools for software development.

Using Command Line:

- Open command prompt.
- **mkdir DevOpsExp3** – this will create **DevOpsExp3** folder.
- **cd DevOpsExp3** – navigate **DevOpsExp3** folder.
- After that, follow the below steps to work with Maven project.



```
C:\Users\Sharath>mkdir DevOpsExp3

C:\Users\Sharath>cd DevOpsExp3
```

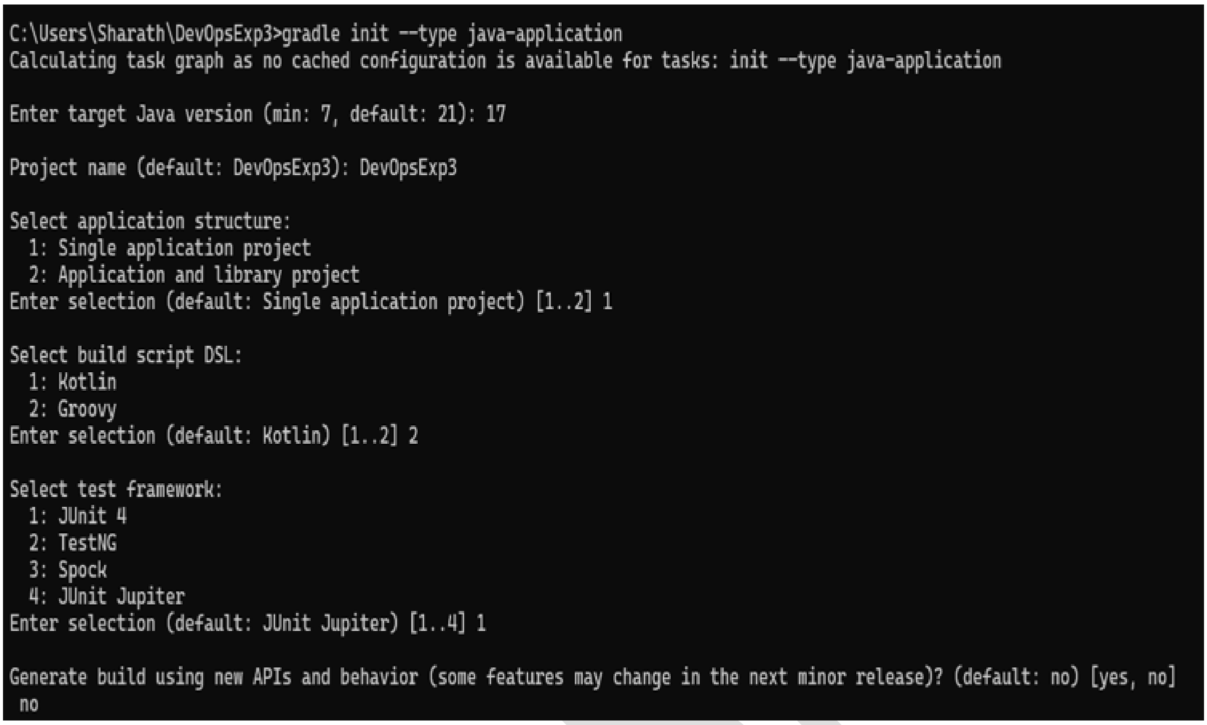
Setting Up a Gradle Project

- **Install Gradle** (If you haven't already):
  - Follow Gradle installation Program 1

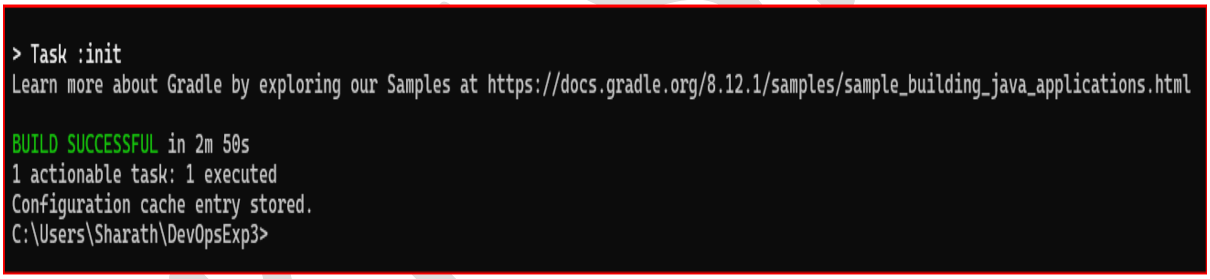
- **Create a new Gradle project**: You can create a new Gradle project using the Gradle Wrapper or manually. Using the Gradle Wrapper is the preferred approach as it ensures your project will use the correct version of Gradle.

- **To create a new Gradle project using the command line:**
  In command prompt type: gradle init --type java-application

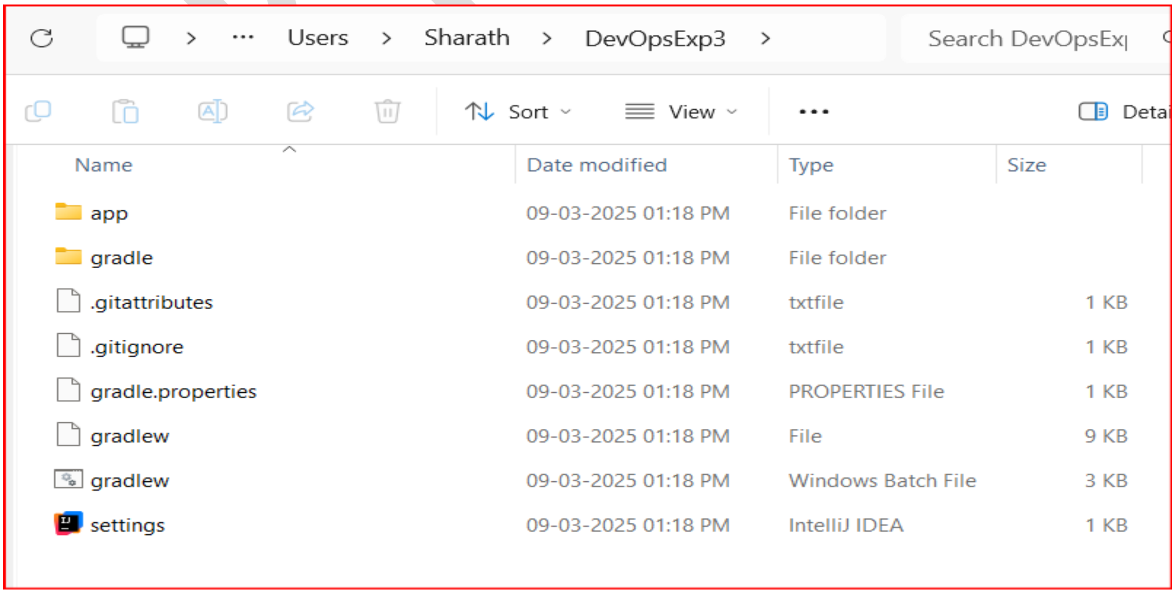
This command creates a new Java application project with a sample **build.gradle** file.



```
C:\Users\Sharath\DevOpsExp3>gradle init --type java-application
Calculating task graph as no cached configuration is available for tasks: init --type java-application

Enter target Java version (min: 7, default: 21): 17

Project name (default: DevOpsExp3): DevOpsExp3

Select application structure:
  1: Single application project
  2: Application and library project
Enter selection (default: Single application project) [1..2] 1

Select build script DSL:
  1: Kotlin
  2: Groovy
Enter selection (default: Kotlin) [1..2] 2

Select test framework:
  1: JUnit 4
  2: TestNG
  3: Spock
  4: JUnit Jupiter
Enter selection (default: JUnit Jupiter) [1..4] 1

Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no]
 no
```



```
> Task :init
Learn more about Gradle by exploring our Samples at https://docs.gradle.org/8.12.1/samples/sample_building_java_applications.html

BUILD SUCCESSFUL in 2m 50s
1 actionable task: 1 executed
Configuration cache entry stored.
C:\Users\Sharath\DevOpsExp3>
```



Name	Date modified	Type	Size
app	09-03-2025 01:18 PM	File folder	
gradle	09-03-2025 01:18 PM	File folder	
.gitattributes	09-03-2025 01:18 PM	txtfile	1 KB
.gitignore	09-03-2025 01:18 PM	txtfile	1 KB
gradle.properties	09-03-2025 01:18 PM	PROPERTIES File	1 KB
gradlew	09-03-2025 01:18 PM	File	9 KB
gradlew	09-03-2025 01:18 PM	Windows Batch File	3 KB
settings	09-03-2025 01:18 PM	IntelliJ IDEA	1 KB

Step 2: build.gradle (Groovy DSL)

```
plugins {
   id 'application'
}

application {
   mainClass = 'com.example.AdditionOperation'
}

repositories {
   mavenCentral()
}

dependencies {
   testImplementation 'junit:junit:4.13.2'
}

test {
   outputs.upToDateWhen { false }

   testLogging {
      events "passed", "failed", "skipped"
      exceptionFormat "full"
      showStandardStreams = true
   }
}
```

Step 3: AdditionOperation.java(Change file name and update below code)

- After creating the project, **change the file name**.
- Manually navigate the folder path like **src/main/java/org/example/**
- Change the file name **App.java** to **AdditionOperation.java**
- After that, open that file and copy the below code and paste it, save it.

```
package com.example;
public class AdditionOperation {
   public static void main(String[] args) {
      double num1 = 5;
      double num2 = 10;
      double sum = num1 + num2;
      System.out.printf("The sum of %.2f and %.2f is %.2f%n", num1, num2, sum);
   }
}
```

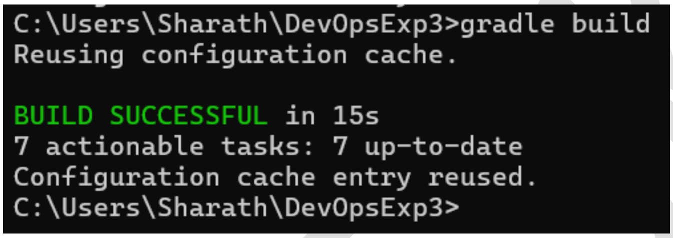
Step 4: AdditionOperationTest.java (JUnit Test) (Change file name and update below code)

- After creating project **change the file name**.
- Manually navigate the folder path like **src/test/java/org/example/**
- Change the file name **AppTest.java** to **AdditionOperationTest.java**
- After then open that file and copy the below code and past it, save it.

```
package com.example;
import org.junit.Test;
import static org.junit.Assert.*;
public class AdditionOperationTest {
   @Test
   public void testAddition() {
      double num1 = 5;
      double num2 = 10;
      double expectedSum = num1 + num2;

      double actualSum = num1 + num2;
      assertEquals(expectedSum, actualSum, 0.01);
   }
}
```

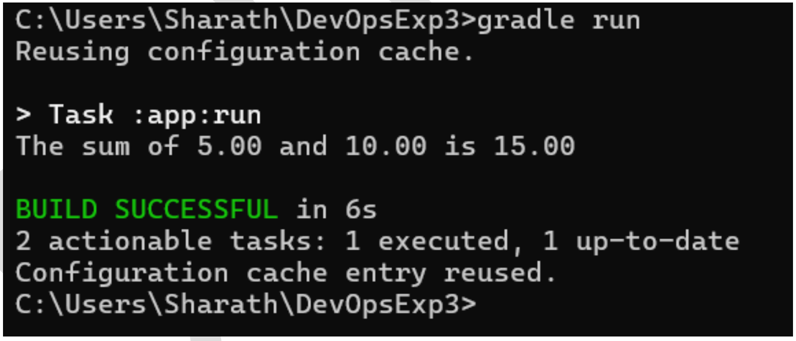
Step 5: Run Gradle Commands

- To **build** the project:
  gradle build



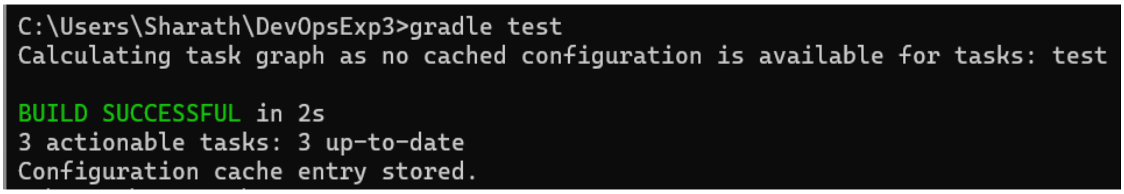
```
C:\Users\Sharath\DevOpsExp3>gradle build
Reusing configuration cache.

BUILD SUCCESSFUL in 15s
7 actionable tasks: 7 up-to-date
Configuration cache entry reused.
C:\Users\Sharath\DevOpsExp3>
```

- To **run** the project:
  In command prompt type: gradle run



```
C:\Users\Sharath\DevOpsExp3>gradle run
Reusing configuration cache.

> Task :app:run
The sum of 5.00 and 10.00 is 15.00

BUILD SUCCESSFUL in 6s
2 actionable tasks: 1 executed, 1 up-to-date
Configuration cache entry reused.
C:\Users\Sharath\DevOpsExp3>
```

- To **test** the project:
  In command prompt type: gradle test



```
C:\Users\Sharath\DevOpsExp3>gradle test
Calculating task graph as no cached configuration is available for tasks: test

BUILD SUCCESSFUL in 2s
3 actionable tasks: 3 up-to-date
Configuration cache entry stored.
```