**overview of build automation tools,key differences between maven and Gradle**

Build tools are one of the essential components of the software development process. Their core function is to transform source code into executable applications. They also help build automation tasks, dependency management, debugging, test automation, code packaging & deployment.

**[Maven](https://maven.apache.org/" \t "https://www.srmtech.com/knowledge-base/blogs/gradle-vs-maven/_blank)** and **[Gradle](https://gradle.org/" \t "https://www.srmtech.com/knowledge-base/blogs/gradle-vs-maven/_blank)** are the most popular build tools available in the market and are widely used by most developer communities.

## ****What is Gradle?****

A flourishing build automation tool launched in 2008, combining apache’s maven and ant concepts. Gradle is mainly preferred for its multi-language project support, flexibility, & customization.

## ****What is Maven?****

A widely used open-source software build & project management tool that the Apache Foundation improved from Apache Ant in 2004. Maven is based on **[Project Object Model (POM)](https://maven.apache.org/guides/introduction/introduction-to-the-pom.html" \t "https://www.srmtech.com/knowledge-base/blogs/gradle-vs-maven/_blank)** & focuses on simplifying and standardizing the building process.

## **Difference between Maven and Gradle**

Considering the functionalities & capabilities of a build tool, Both Maven and Gradle has some similarities and distinctive characteristics, making them to top-line of well-rounded build management tools available in the market.

| ****Aspect**** | ****Gradle**** | ****Maven**** |
| --- | --- | --- |
| ****Key Focus**** | Gradle is meant to add additional functionalities to the application. | Maven is meant to develop applications in a given timeline. |
| ****Language Support**** | Gradle supports languages like Java, C, C++, Groovy, Scala & Android. | Maven supports languages like Java, Scala, C#, and Ruby. |
| ****Configuration**** | Gradle uses DSL(Domain Specific Language) based on Groovy to configure projects. | Maven uses XML to configure projects. |
| ****Performance**** | Gradle allows multi-module builds to run in parallel. It performs better than maven due to features like an incremental build, build cache, API usage & compiler daemon that makes compilation faster. | Maven also allows multi-module builds to run in parallel, but its build time is longer as it lacks a build cache to create local temporary files. It does not support incremental compilations. |
| ****Customization**** | Gradle provides extensive customization to cover different ecosystems with its Groovy-based build script. | Maven offers limited customizations due to its project structure & focuses on pre-defined goals. |
| ****Dependency Management**** | Gradle uses a dependency tree approach, and IVY Metadata compatibility helps specify a version for a dynamic dependency & resolves the highest version dependency found in the tree. It allows for creating new custom dependency scopes, which leads to better modeled & faster builds. | Maven uses a declaration order approach and resolves dependency conflict based on the shortest path. It has a central JAR repository & facilitates the usage of JARs across the projects. Maven has a built-in dependent scope, and additional scopes cannot be added. |
| ****Debugging Tools**** | Gradle provides a web-based debugging tool called Build Scan for debugging & build optimization. It also allows for trend analysis, build comparison & history collection. | Maven has debug mode to identify the cause of errors. It also provides a Surefire plug-in for project debugging & Eclipse to debug test runs. |

## ****Why Gradle is used?****

* Highly Customizable
* Multi-language projects
* Better IDE support
* Better Performance
* Incremental Builds

## ****Downsides of Gradle****

* Expertise scarcity
* Extensive Documentation
* Plug-ins Availability

## ****Why Maven is used?****

* Highly Standardized
* Streamlined Build Process
* Wider Component Builds
* Better Collaboration
* Reduce Duplication

## ****Downsides of Maven****

* No Incremental Compilations
* Lacks Build Cache
* Limited IDE support

## ****What is Gradle used for?****

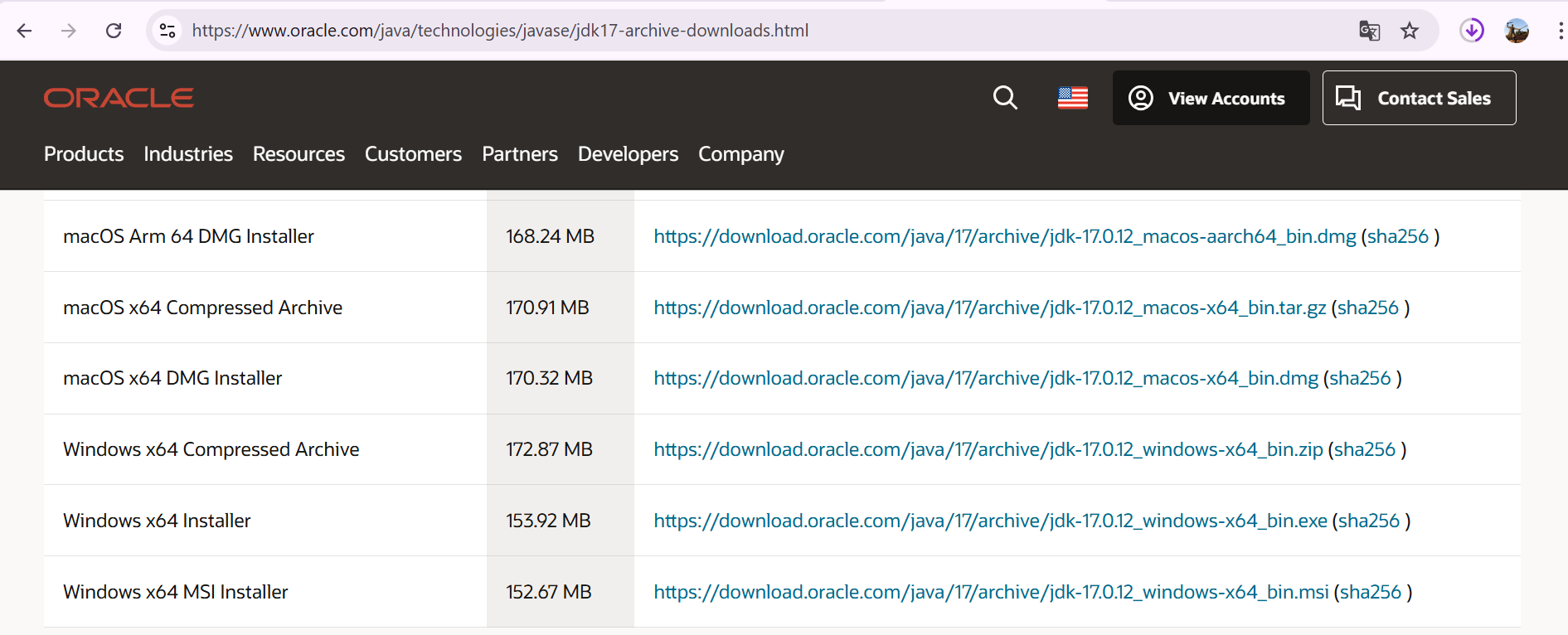
Gradle is very flexible & ideal for complex projects, where priorities are versatility, performance & incremental builds.

## ****What is Maven used for?****

Maven is simple to use and ideal for standard projects, where priorities are modularization, dependency management & consistency

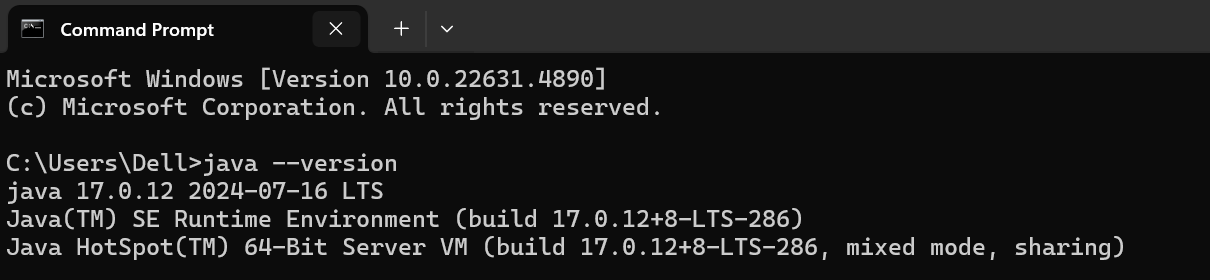
MAVEN INSTALLATION

In windows, open the command prompt, and check for the java version. If the java is not installed install it with

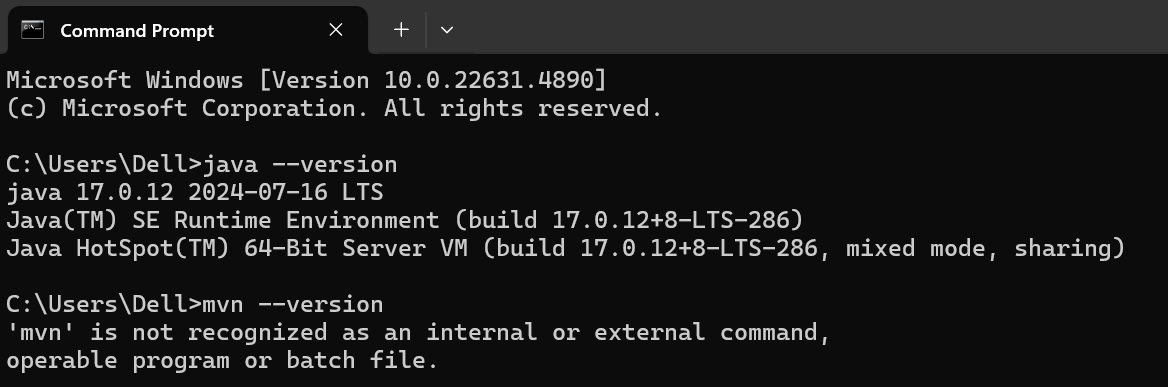


https://download.oracle.com/java/17/archive/jdk-17.0.12\_windows-x64\_bin.exe

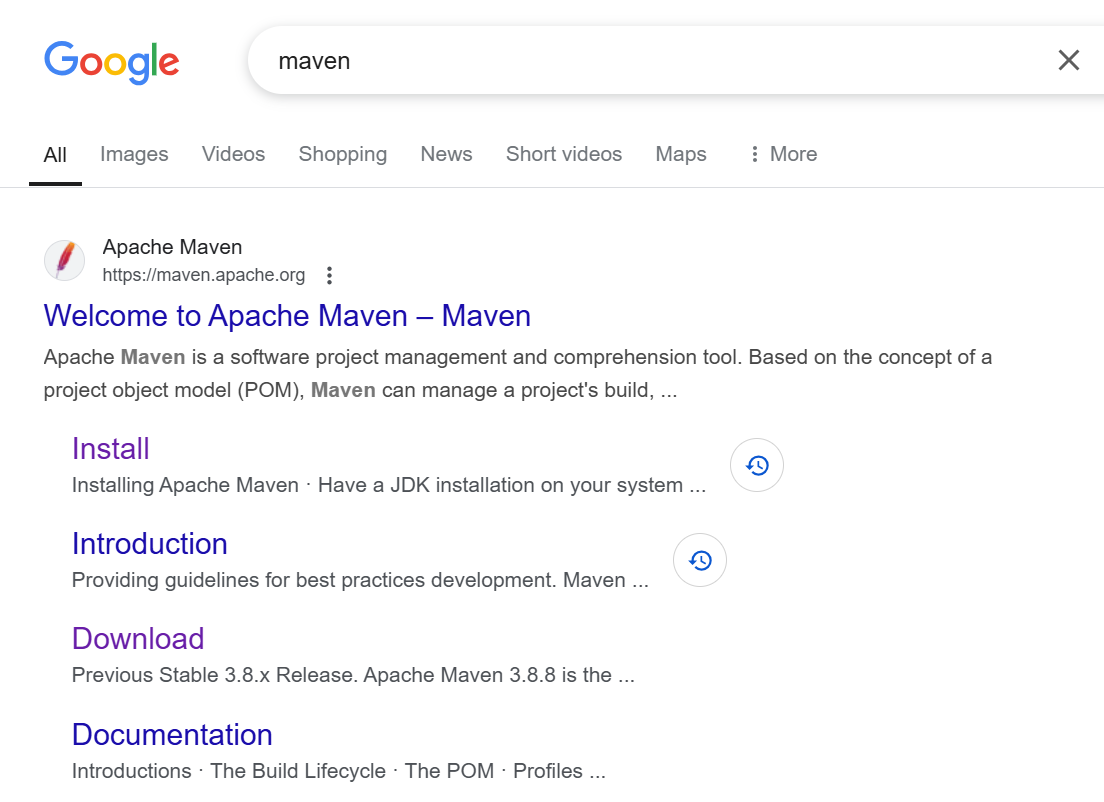
Check for the java using java --version



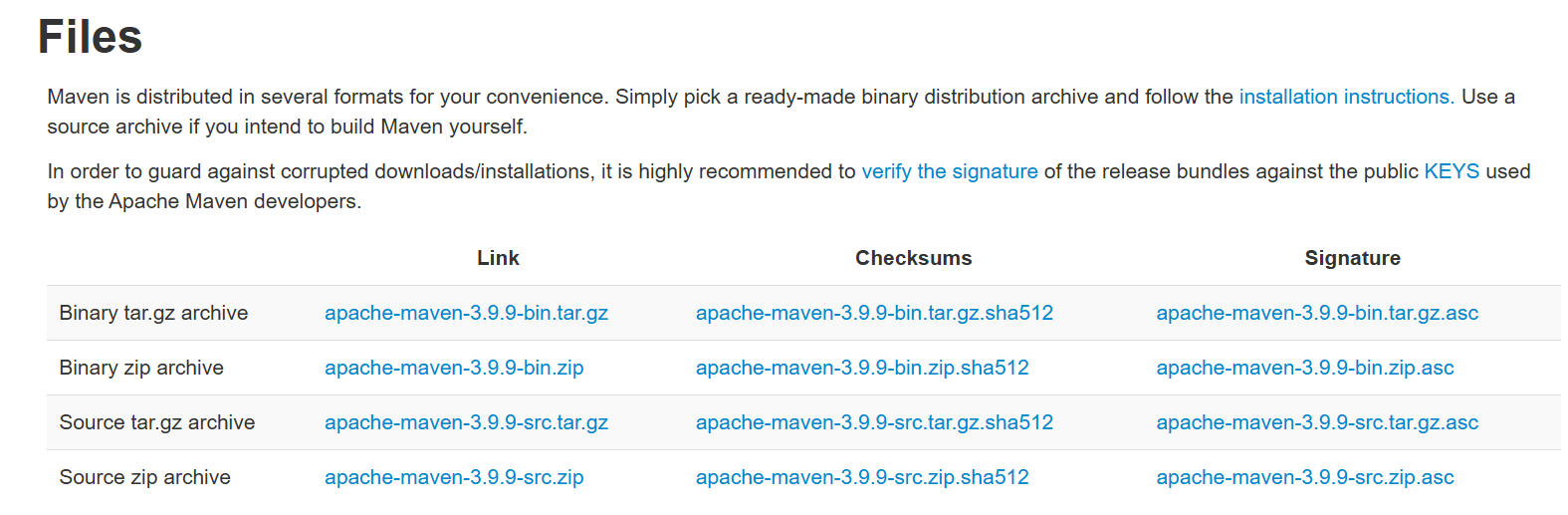
Search for the maven using mvn --version



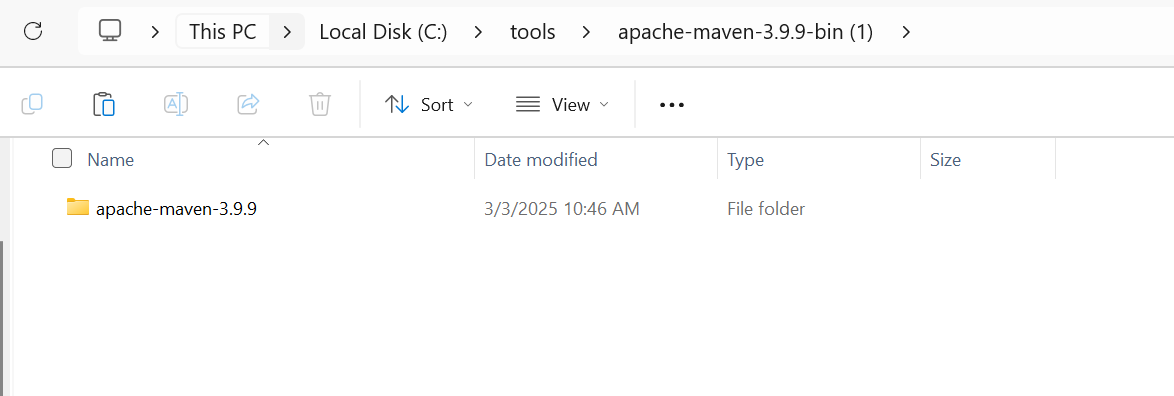
In Google Search for the maven, and then click on the Downloads



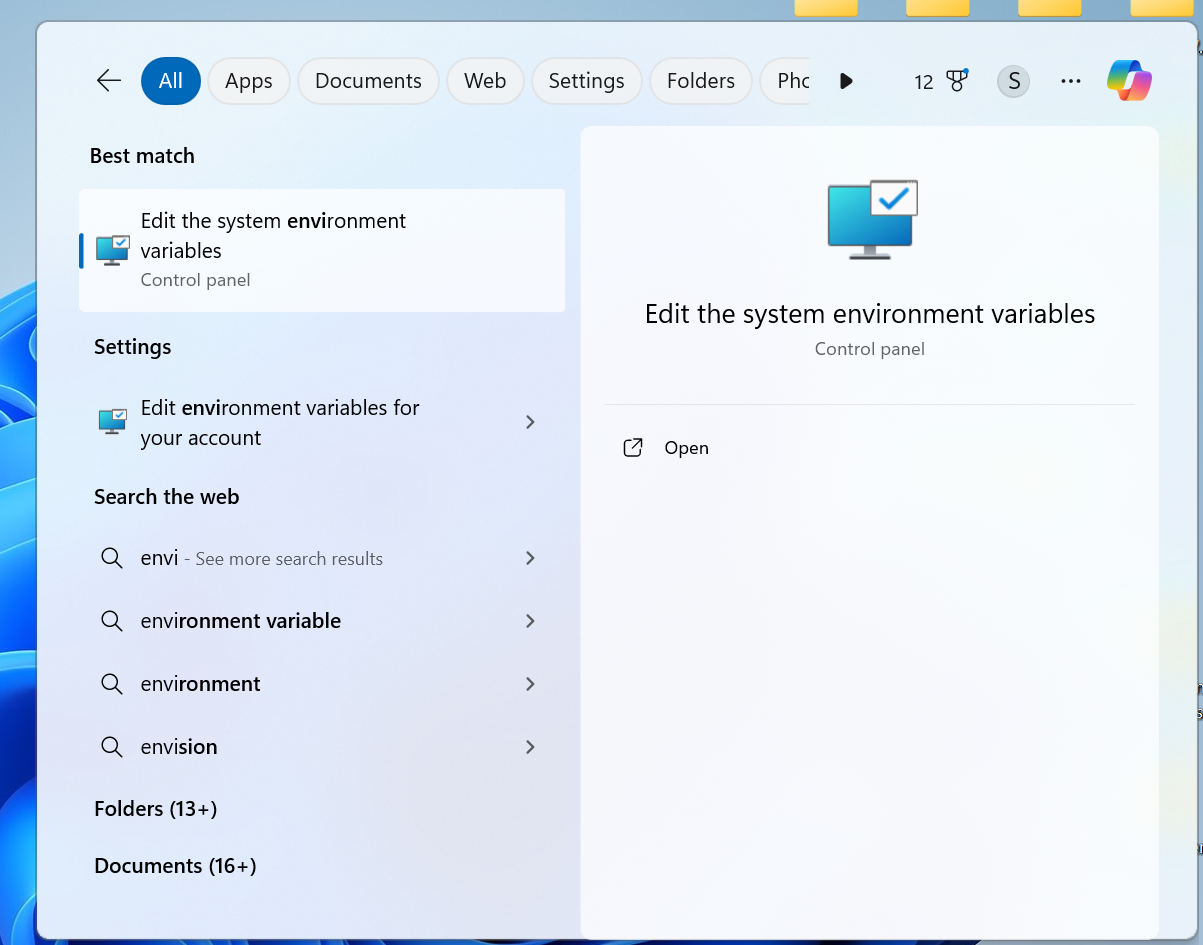
Then you will be redirected to download files, then download for the binary zip file



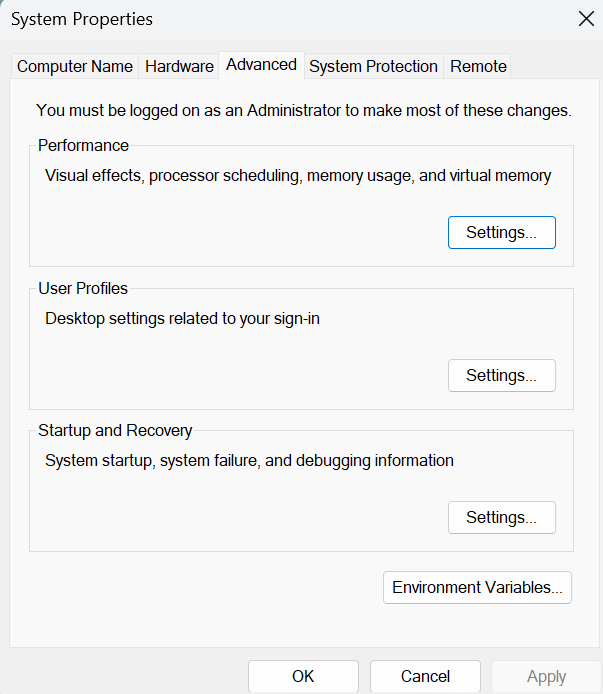
Then unzip the folder downloaded



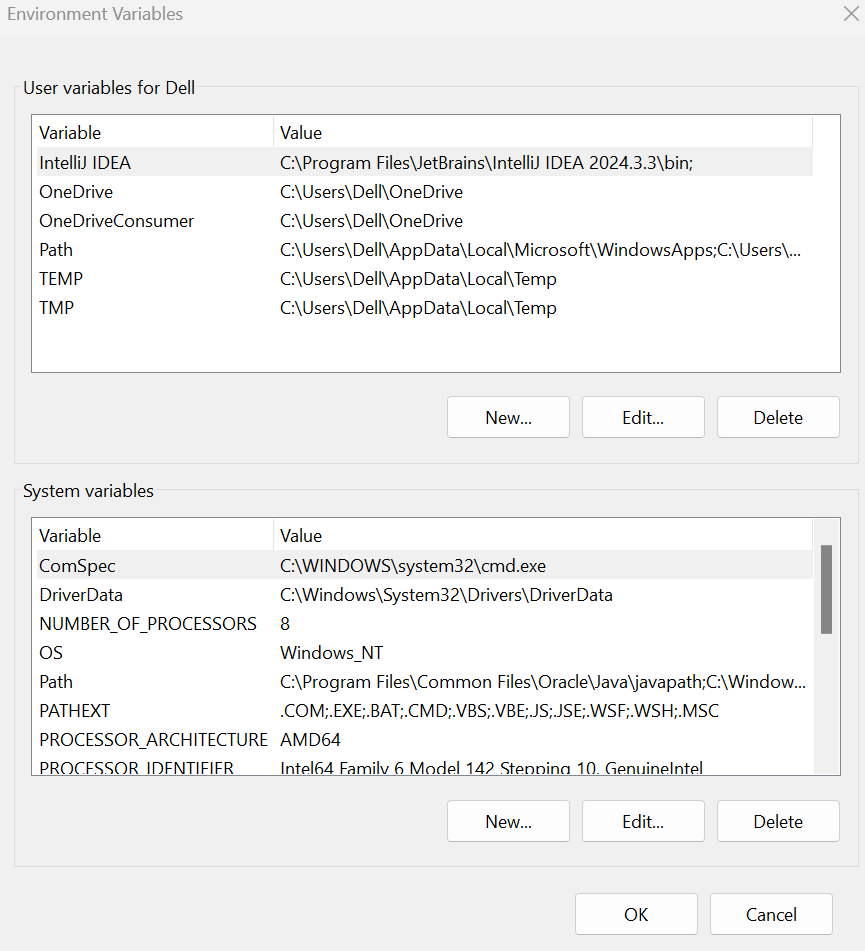
After unzipping, at start check for the edit the system environment variables.



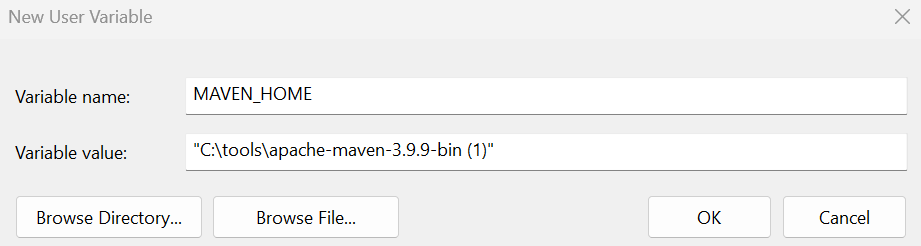
Click on the Environment Variables.



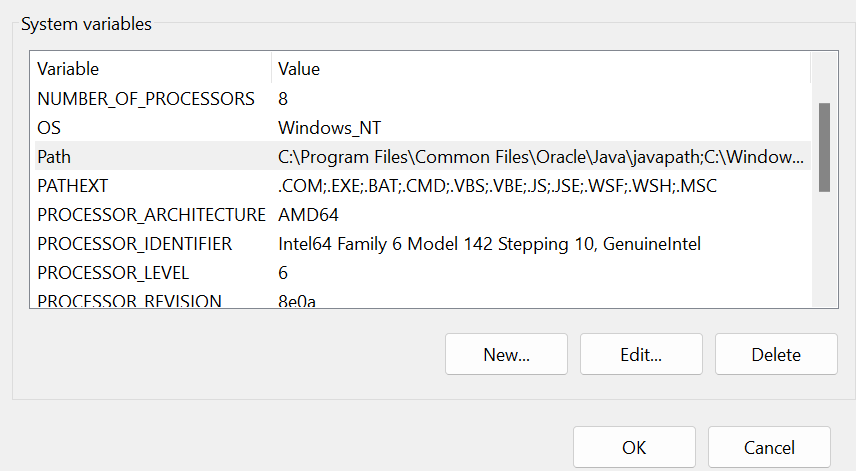
In User Variables click on New



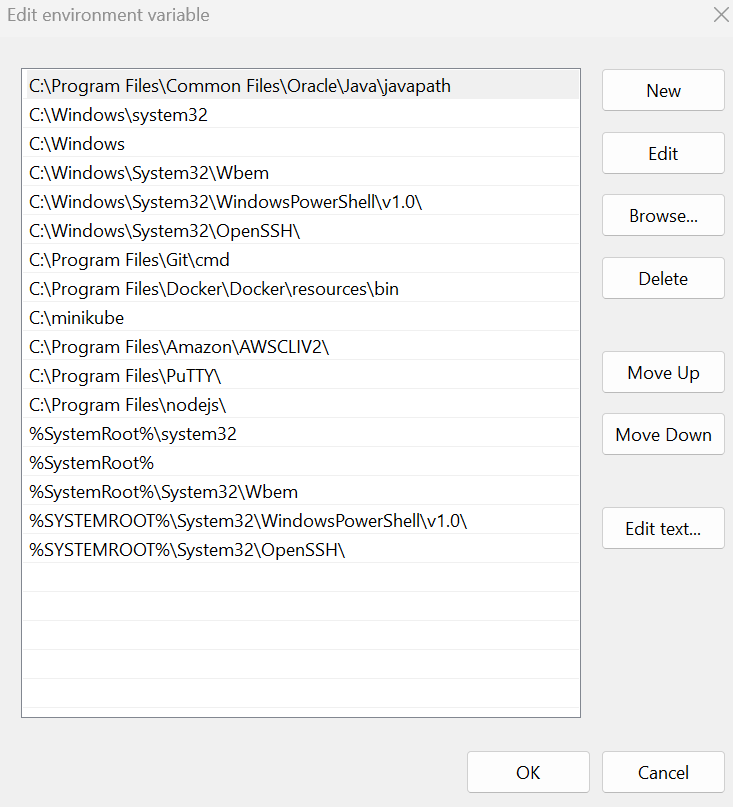
Add variable name as MAVEN\_HOME and variable value as the folder path of the extracted folder.



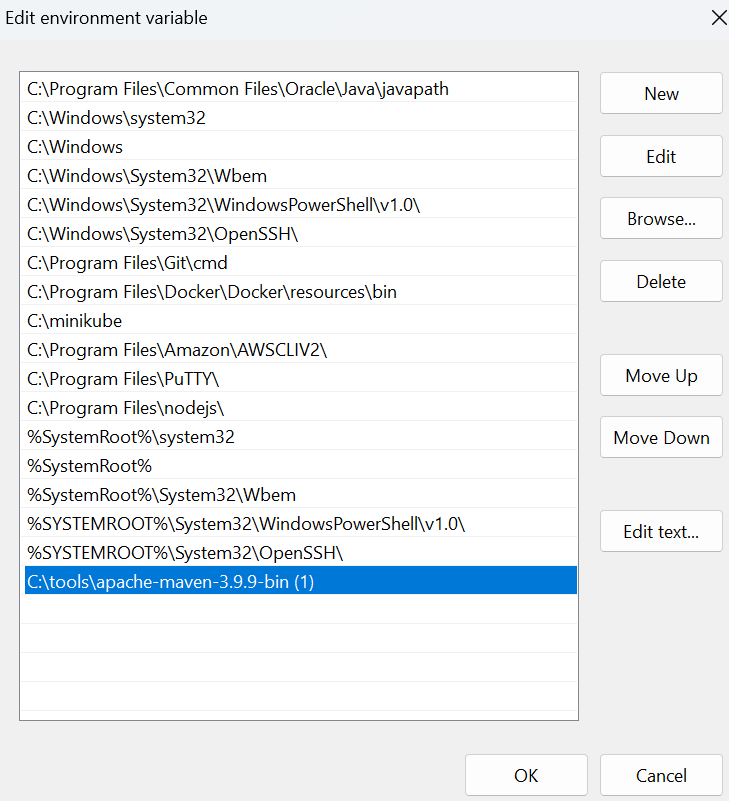
In the System Variable, click on the New



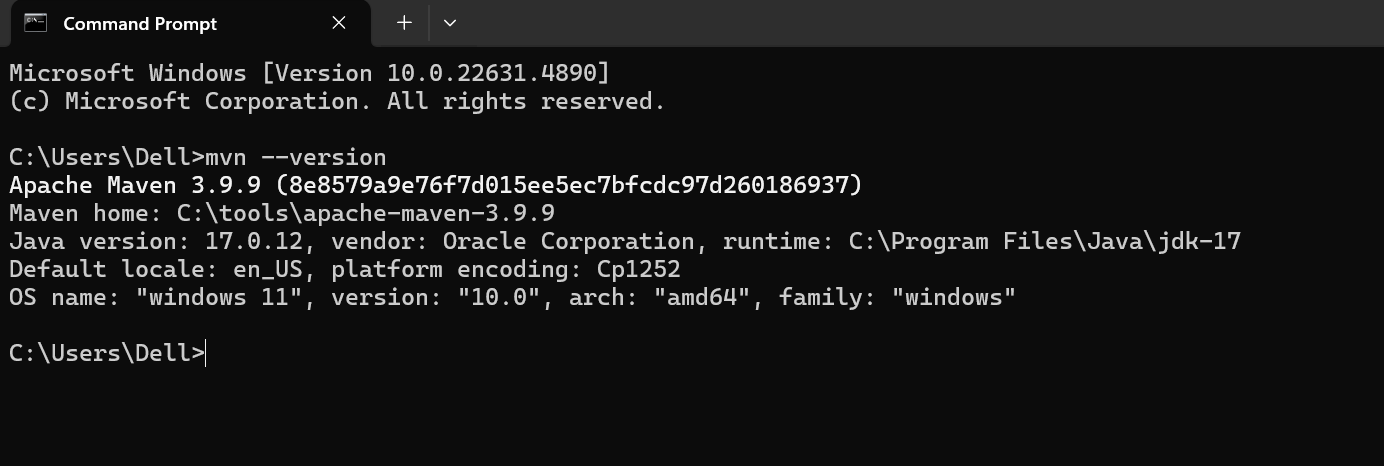
Click on the New



Then add the bin Folder path,

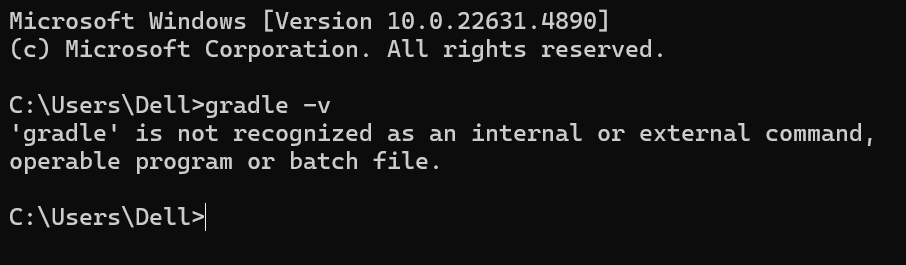


Then restart the command prompt and check for the mvn --version, the maven will be installed

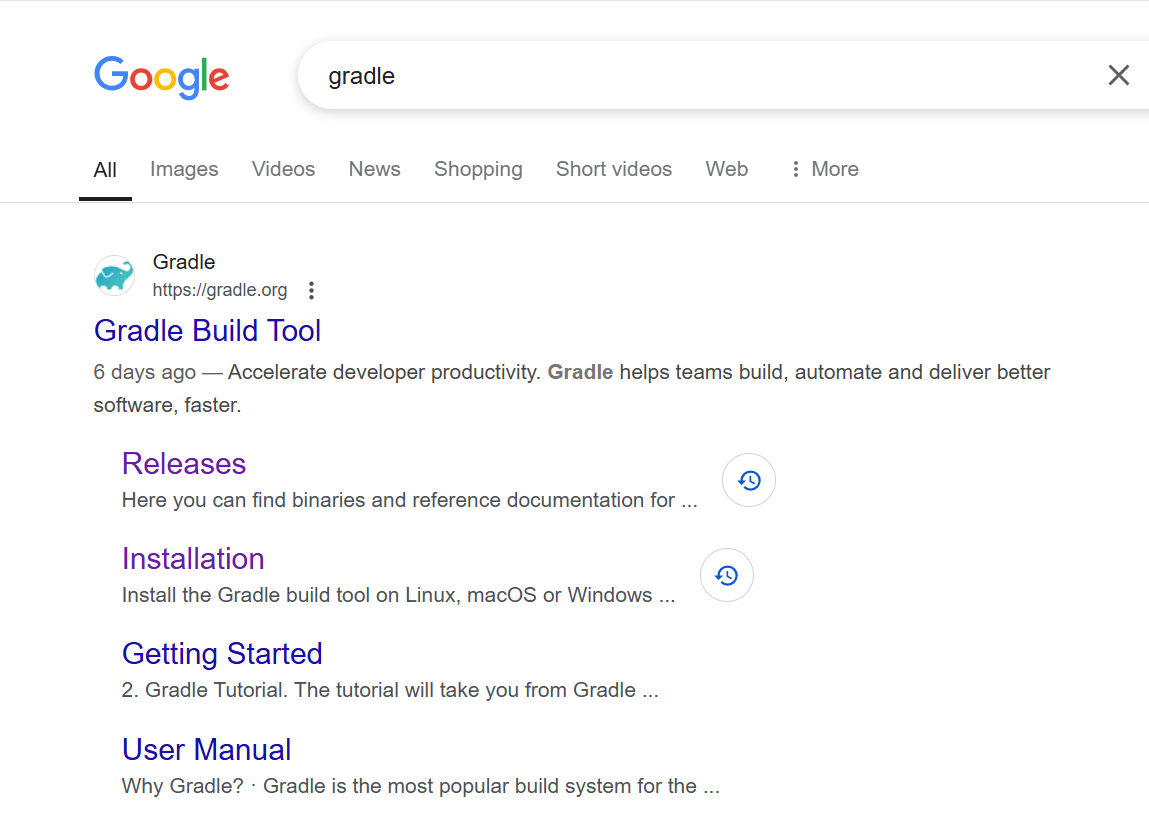


GRADLE INSTALLATION

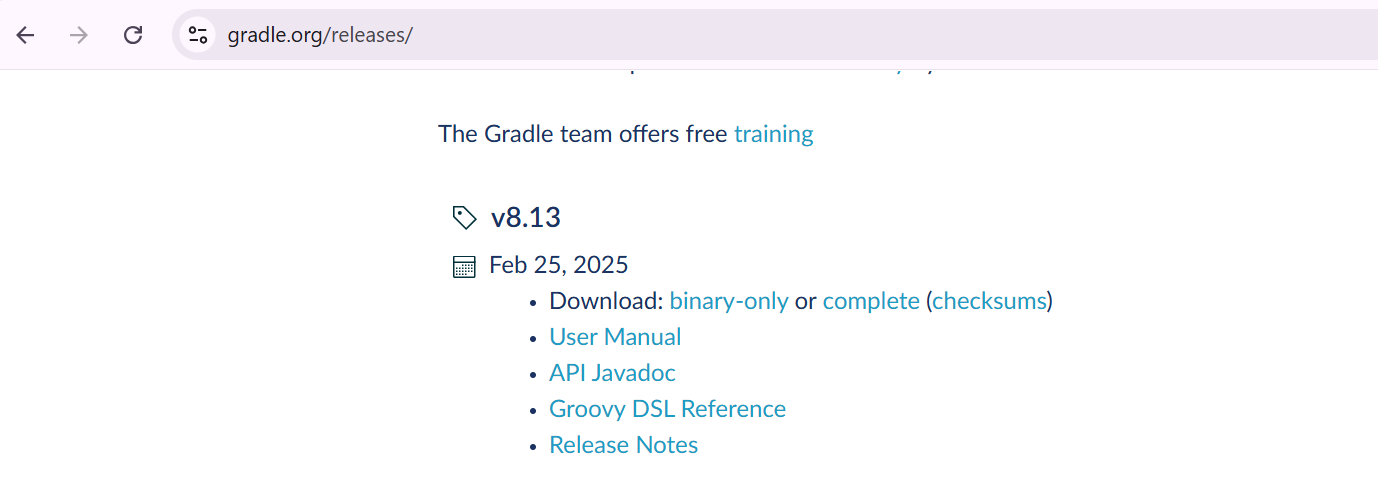
Check for the Gradle version in the Command Prompt

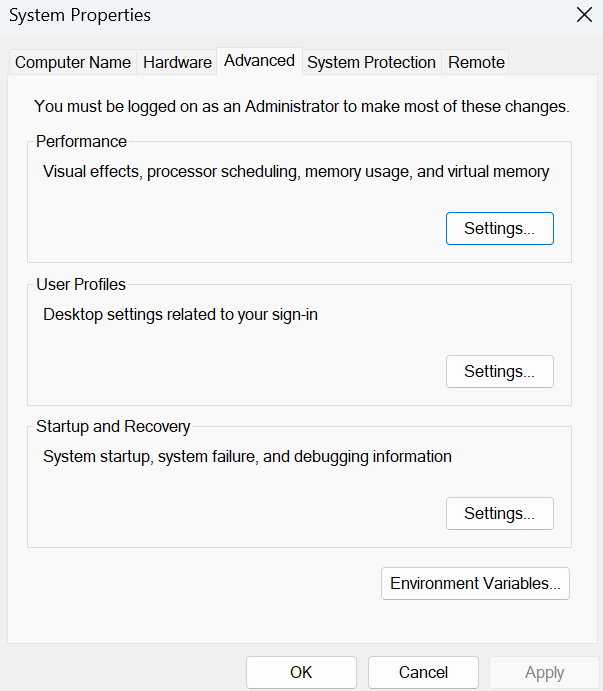


Search for the Gradle in Windows, and Click on the Releases

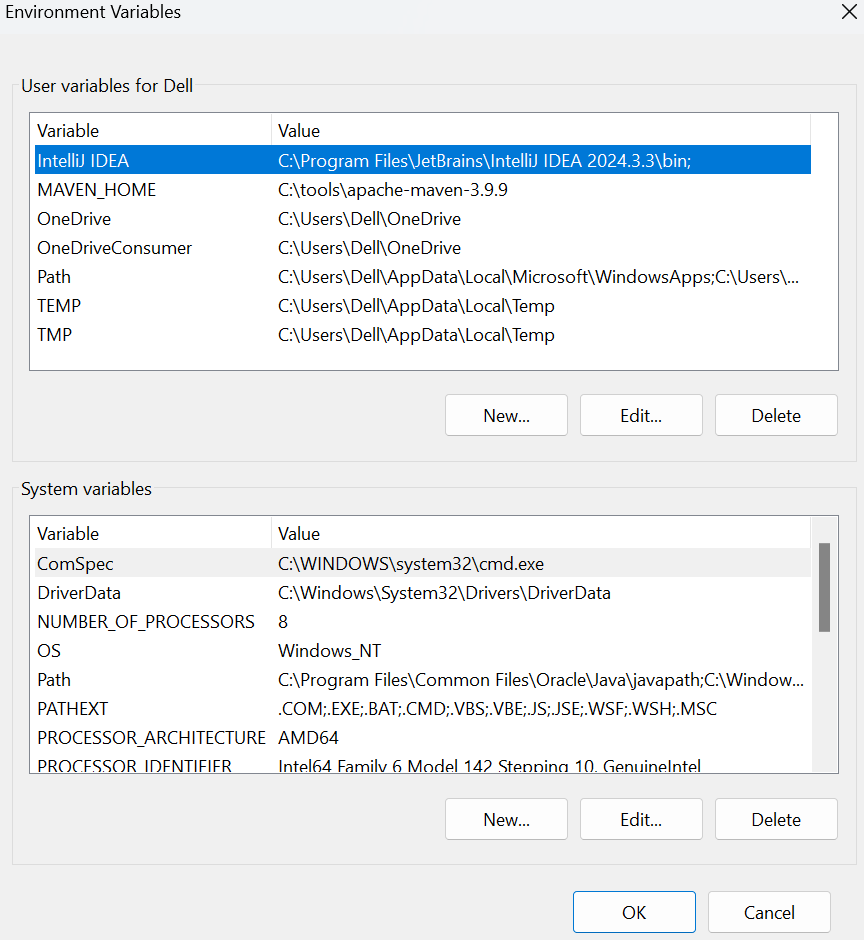


Click on the Binary Download and the binary zip file will be Downloaded

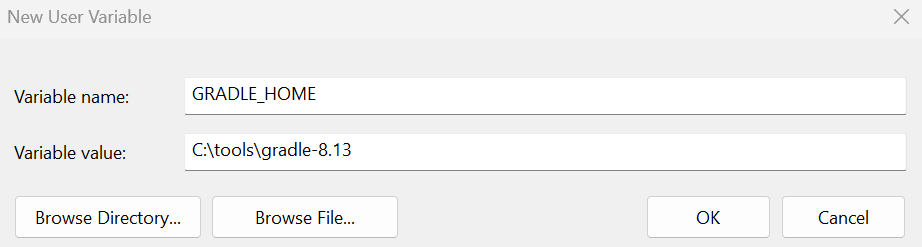


In start, check for the Edit the System Environment Variable , Click on environment Variable 

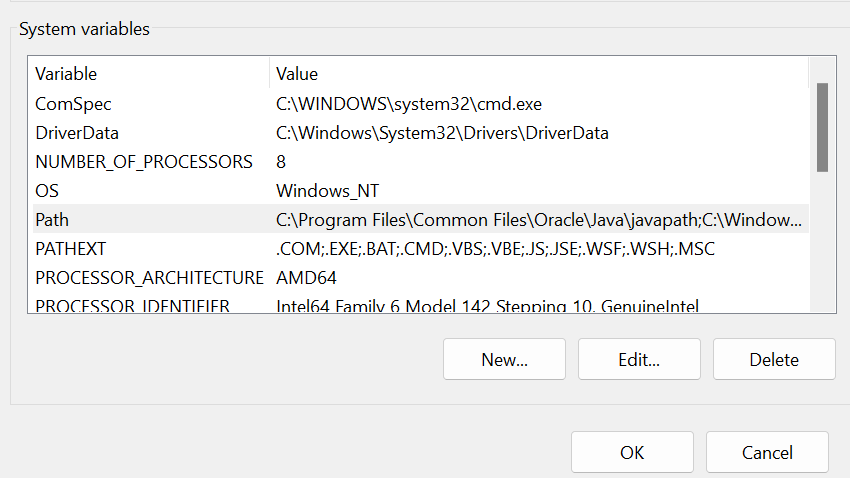
In User Variables for Dell Click on the New



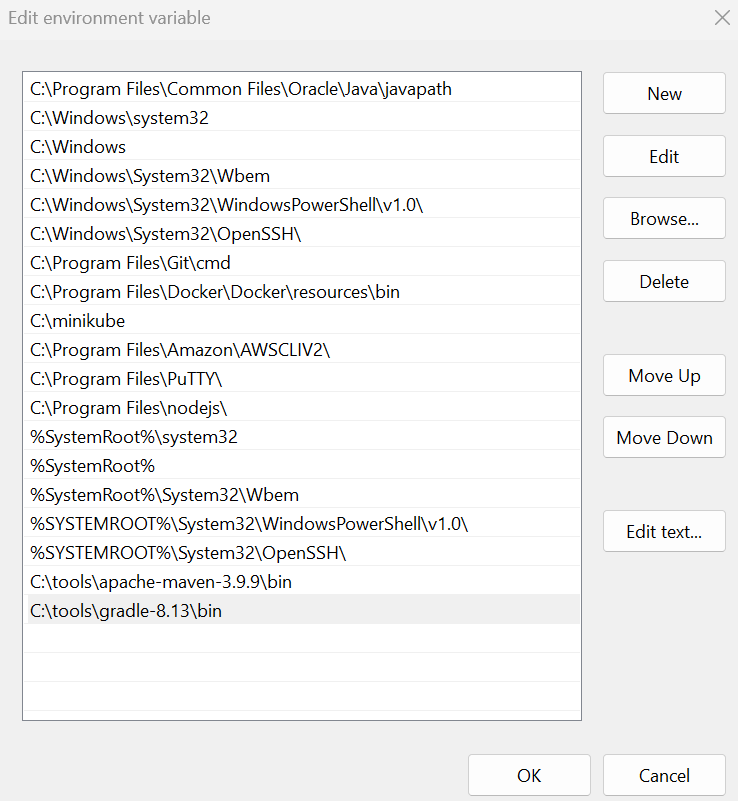
In Variable Name Add Variable name as GRADLE\_HOME. And in Variable value add the gradle folder path



And in the System Variables click on the New



Click on New, and then add the bin folder path



Now check for the gradle version in the command prompt

