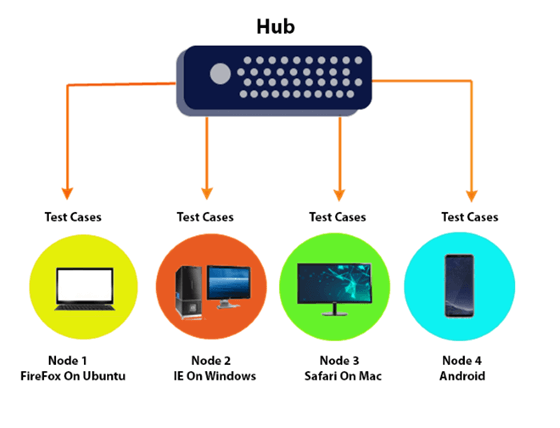
**Selenium Grid**

* Selenium Grid is a feature in Selenium that allows you to run test cases in different machines across different platforms. The control of triggering the test cases is on the local machine, and when the test cases are triggered, they are automatically executed by the remote machine. Suppose you have 5 test cases. Your local machine is running multiple applications, so you want to run your test cases in a remote machine. You need to configure the remote server so that the test cases can be executed there.
* It supports distributed test execution. Initially, you have a local machine where you write the test cases and executes on the same machine. However, in a big organization, you have multiple test cases, and it's not possible to run all the test cases in the same machine. In a large organization, you have multiple servers, so local machine distributes the test cases across different machines/servers.
* You can also run the test cases in parallel in multiple machines on Selenium Grid.

Architecture of Selenium Grid



Why Selenium Grid?

Selenium Grid is useful because of the following reasons:

* **Run on different platforms:** You can run the test cases in different platforms, so it is a platform independent. For example, you have a hub which has internet explorer 9. Hub has internet explorer 9 as many older websites support IE 9. Now we want to run the test cases in different platform such as Internet Explorer. As we know that only one version of a browser can be installed on a computer. You need to configure the node to which you want to send your test cases.
* **Parallel execution:** If you set up the Selenium Grid, you can run the multiple cases at the same time. This saves time in running the test suites.

**Key Components of Selenium Grid**

Hub: The hub is the central point where you load your tests. There is only one hub in a grid. The hub controls the test execution and serves as a load balancer. It distributes the tests to the nodes.

**Node:** Nodes are the machines (or instances) where your tests are actually executed. There can be multiple nodes in a grid, each running different browsers or versions of browsers. Nodes register themselves with the hub.

**How Selenium Grid Works**

**Setup Hub:** You set up a hub on a central machine. This hub receives the test requests from your automation scripts and distributes them to the appropriate nodes based on capabilities.

**Register Nodes:** You register nodes with the hub. These nodes can be on the same machine as the hub or on different machines. Each node can run a different browser or version of a browser.

**Execution:** When a test is executed, the hub receives the request and forwards it to the appropriate node that matches the test’s desired capabilities (such as browser type, version, operating system).

**Advantages of Using Selenium Grid**

**Parallel Execution:** Tests can be run in parallel, reducing the total execution time.

**Cross-Browser Testing:** Run tests on different browsers and browser versions simultaneously.

**Cross-Platform Testing:** Run tests on different operating systems.

**Scalability:** Easily add more nodes to scale up the testing efforts.

**Setting Up Selenium Grid**

Here's a basic guide on how to set up a Selenium Grid:

**Download Selenium Server:** Download the Selenium Server Standalone JAR file from the Selenium website.

**Start the Hub:**

* Open a command prompt or terminal window.
* Navigate to the directory where the Selenium Server JAR file is located.
* Run the command: java -jar selenium-server-standalone-<version>.jar -role hub
* The hub will start and by default will be available at http://localhost:4444/grid/console

**Start a Node:**

* Open another command prompt or terminal window.
* Navigate to the directory where the Selenium Server JAR file is located.
* Run the command: java -jar selenium-server-standalone-<version>.jar -role node -hub http://localhost:4444/grid/register
* This node will register with the hub and be available for running tests.

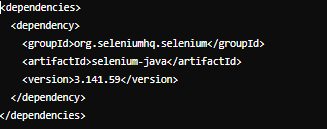
**Example: Running a Test on Selenium Grid**

**Create a Simple Selenium Test in Java**

**Set Up Your Project:**

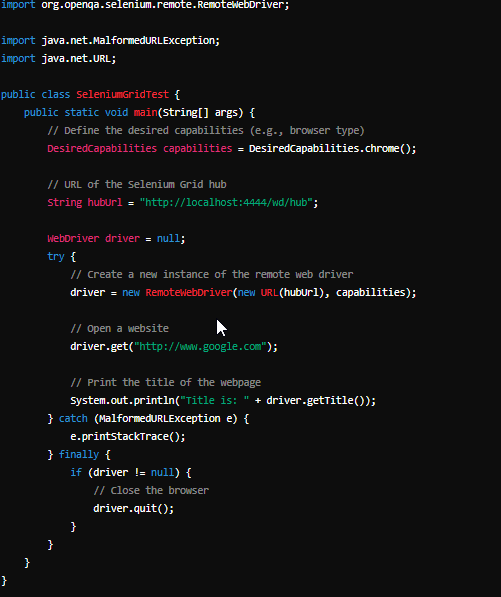
Create a new Maven project (or any Java project).

Add the Selenium dependency in your pom.xml file:



Write the Test Script:

Here's a simple example of a Selenium test script in Java that uses Selenium Grid:



**Step 3: Execute the Test**

Compile and Run the Test:

Make sure your Selenium Grid hub and nodes are running.

Run the Java test script.

**Explanation of the Test Script**

DesiredCapabilities: This class is used to set the desired capabilities for the browser. In this example, we're specifying that we want to use Chrome.

RemoteWebDriver: This class is used to send commands to a remote server (the Selenium Grid hub) where the actual browser sessions are managed.

Hub URL: The URL of the Selenium Grid hub (http://localhost:4444/wd/hub).

WebDriver: The WebDriver instance is used to interact with the web browser. The test script opens Google, prints the page title, and then closes the browser.

This setup allows you to run your tests on different machines and browsers, facilitating parallel and cross-browser testing.