Automation Testing

Selenium Grid

A Day in the Life of an Automation Test Engineer

Joel is an Automation Engineer who wants to test multiple projects.

As an Automation Test Engineer, he wants to run tests regularly and reduce the execution time of test cases.

One of the ways to reduce the time of execution of test cases is to run the tests in parallel on multiple machines.

To know more about it, let us go through the lesson.

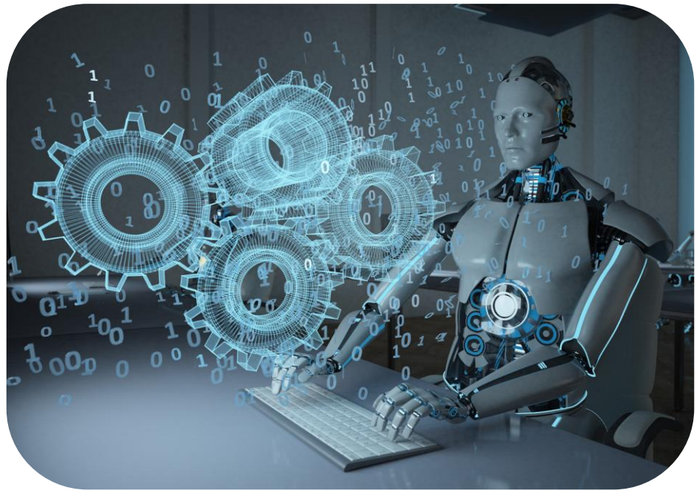
Learning Objectives

By the end of this lesson, you will be able to :

- Define Selenium Grid
- Implement the hub
- Configure the nodes

Introduction to Selenium Grid

Introduction to Selenium Grid



The Selenium Grid allows the execution of WebDriver scripts on remote machines (virtual or real) by routing commands from the client to remote browser instances.

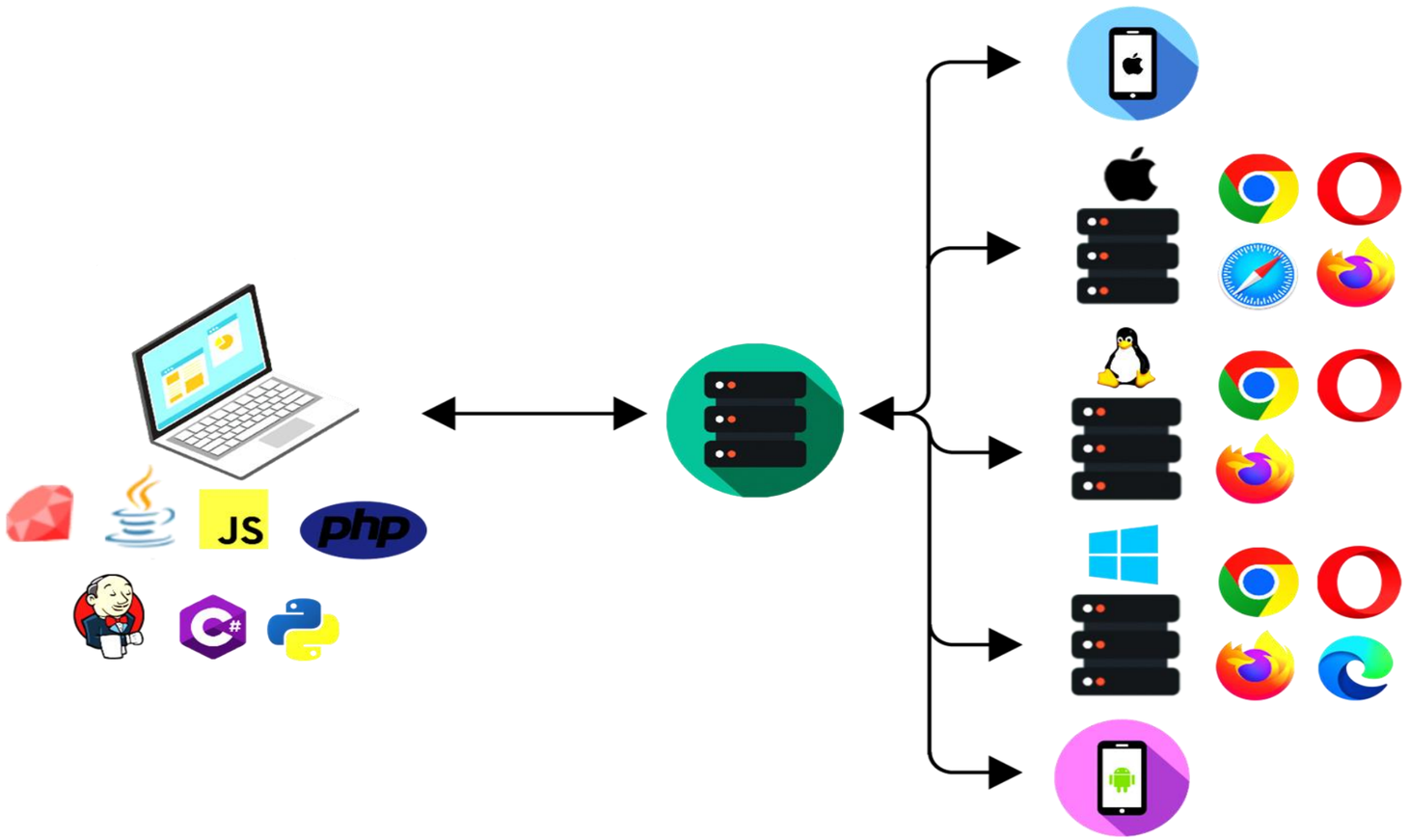
The goal is to offer an easy method for running tests on multiple machines simultaneously.

Users can simultaneously run tests on multiple machines and manage different browser versions and browser configurations.

Selenium Grid Architecture and Components

Architecture of Selenium Grid

A Selenium Grid is composed of a hub and nodes.



Hub and Node

A Selenium Grid is composed of a hub and nodes.

Hub	The hub is the central point in the Selenium Grid that routes test commands to the nodes.
Nodes	The nodes represent the physical or virtual machines on which tests are executed.

Hub and Nodes Configuration

Hub and Node Configuration

Following are the steps to configure the hub and the Node:

01 Download the **Selenium Server**

02 Start the **Hub**

03 Open the **Command Prompt**

04 Type the command: **java -jar selenium-server-4.3.0.jar hub**

Hub and Node Port Configuration

By default, the hub will be started automatically using port 4444.

Example:

```
http://localhost:4444 port number
```


The user can run the program on another port by changing the "--port *port_number>".

Hub and Node Port Configuration

The following command can be used to start the node:

Example:

```
Java –jar selenium-server-standalone-
,version>.jar node
```


A Selenium server, by default, runs on port 5555.

Execute Selenium WebDriver Code via Grid

The following code is an example of how to execute a test using the Selenium Grid setup:

Example:

```
import java.net.URL;

Run All
public class TestSeleniumGrid {

    @Test
    Run | Debug
    public void verifySeleniumGridExecution() throws Exception{

        ChromeOptions chromeOptions = new ChromeOptions();
        chromeOptions.setCapability("browserVersion", "102");
        chromeOptions.setCapability("platformName", "MAC");

        // Showing a test name instead of the session id in the Grid UI
        chromeOptions.setCapability("se:name", "My simple test");


        WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444"), chromeOptions);

        driver.get("http://test.qatechhub.com");

        String actualTitle = driver.getTitle();

        String expectedTitle = "Test QA Tech Hub – Learning By Doing is the best way to learn!";

        Assert.assertEquals(actualTitle, expectedTitle);

        driver.quit();

    }

}
```

Key Takeaways

- In a grid, there is one hub that acts as the control unit and multiple nodes where tests are executed.
- The Selenium Grid reduces the execution time drastically as it executes tests in parallel.
- The user must set up the environment using the Jar file to use the Selenium Grid.

Thank You