**SELENIUM TEST AUTOMATION**

**What is selenium?**

* Selenium is Automated Testing suite for web applications
* Selenium is open source tool.(No Costs is involved)
* Selenium supports languages like JAVA, Ruby, PHP, Perl, Python, C#
* Selenium supports browsers like IE,Mozilla Firefox , Google Chrome,Safari, Opera.
* Selenium is very flexible when compared to other testing tools like QTP and it supports multiple languages



**Components of selenium**

* **Selenium IDE**

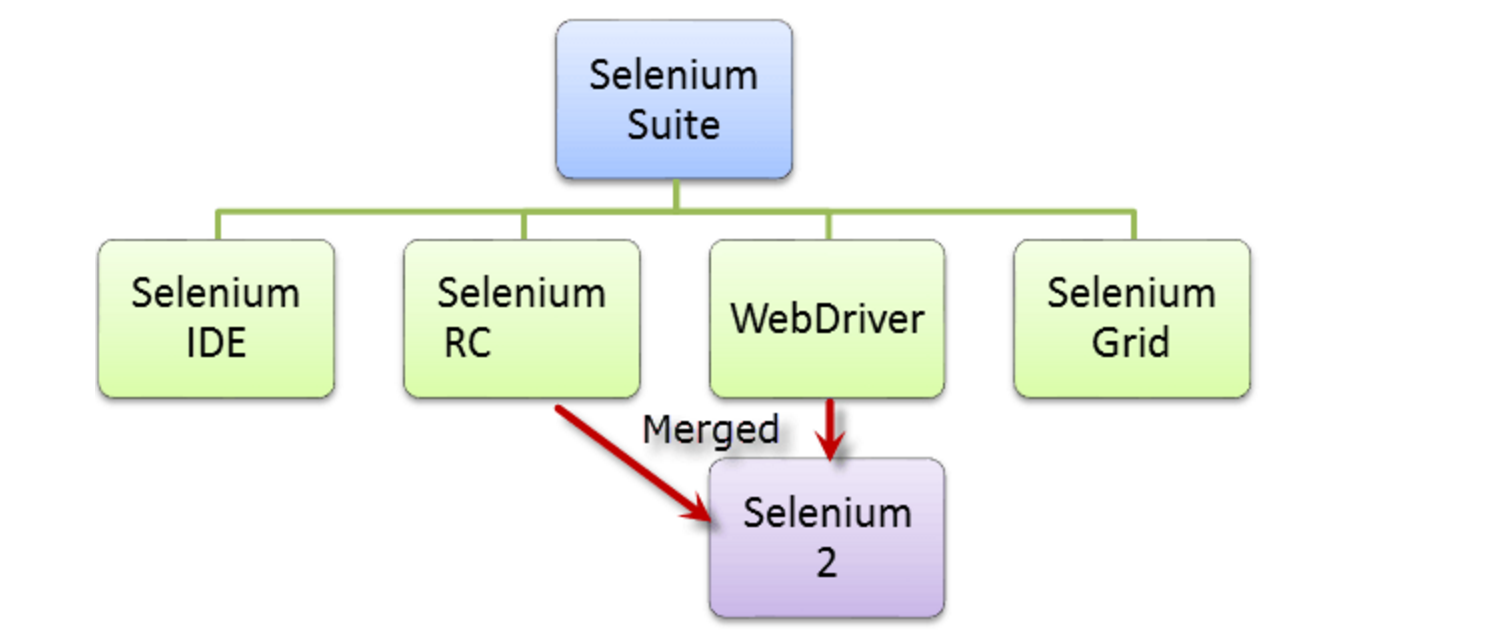
Integrated Development Environment - Record and PlayBack scripts

* **Selenium Remote Control**

Server and Launches Browser . Library of selenium

* **Selenium GRID**

Run tests on different machines against different browsers in parallel



**SELENIUM GRID**

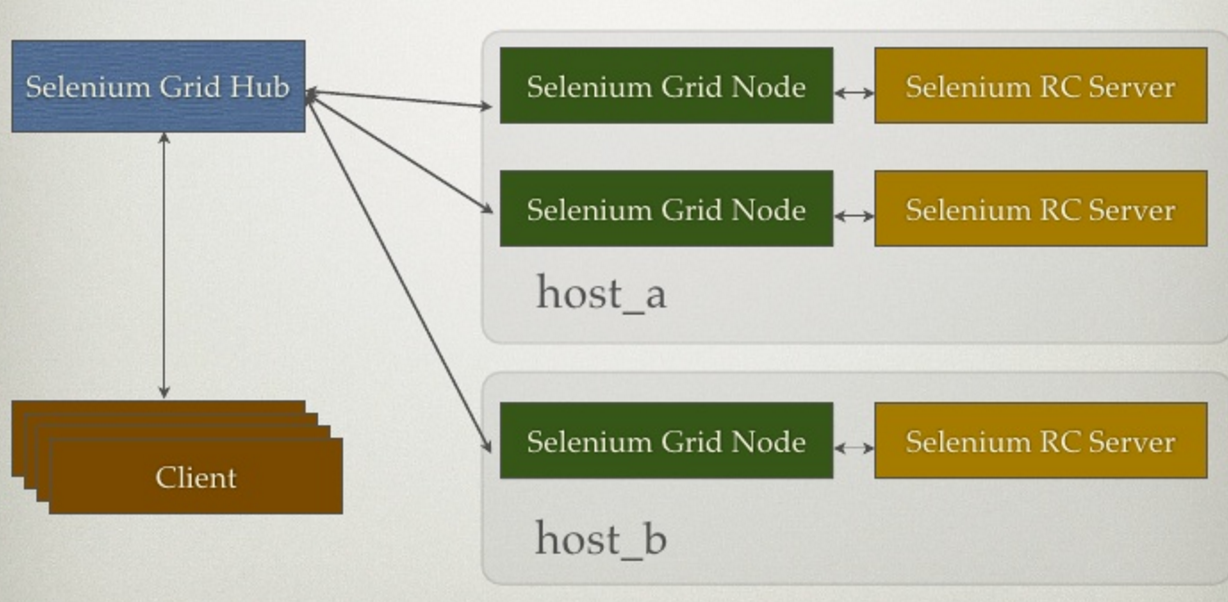
* To run tests against multiple browsers, multiple versions of browser, and browsers running on different operating systems.
* To reduce the time it takes for the test suite to complete a test pass.
* Selenium Grid uses a hub-node concept where you only run the test on a single machine called a **hub**, but the execution will be done by different machines called **nodes**.

**HUB**

* Hub is the central point where you load your tests into.
* The machine containing the hub is where the tests will be run, but you will see the browser being automated on the node.

**NODES**

* Nodes are the Selenium instances that will execute the tests that you loaded on the hub.
* Nodes can be launched on multiple machines with different platforms and browsers.



**Selenium Grid With Jenkins**

* On master, Selenium Grid Hub is started on port 4444, unless configured otherwise in Jenkins global configurations. This is where all tests should connect to.
* For each slave, necessary binaries are copied and Selenium RCs are started.
* RCs and the Selenium Grid Hub are hooked up together automatically.

**Installing Selenium Grid**

* Search for "Selenium Grid" on Jenkins' "Manage Plugins" screen.
* Plugin adds a “Selenium Grid” option to the sidebar of the Jenkins dashboard.
* Screen gives an overview of the status of the Hub, such as console output, and will list any connected Nodes.
* It also lists the URL needed in our Selenium test code to utilize the Grid

**Creating a Node Configuration**

* Navigate to “Configurations” in the Selenium Grid Page.
* Configurations are used to define options for each Node, such as what browsers are present, and where their binaries are stored.
* Select “New Configuration”. Select “Match by a label” option to specify which Jenkins Slave this configuration is for.

**Launching a Node**

* Once we have a configuration, navigate to “Nodes Matching Configuration”.
* This screen lists any available Jenkins Slaves that match your configurations; currently, it should list any Jenkins slaves that match the configuration we just made.
* Now we can connect a Node to the Selenium Hub using the configuration.
* Click on the name of the slave you’d like to run Selenium Tests on, and click the “Start” button to enable it as a Selenium Node

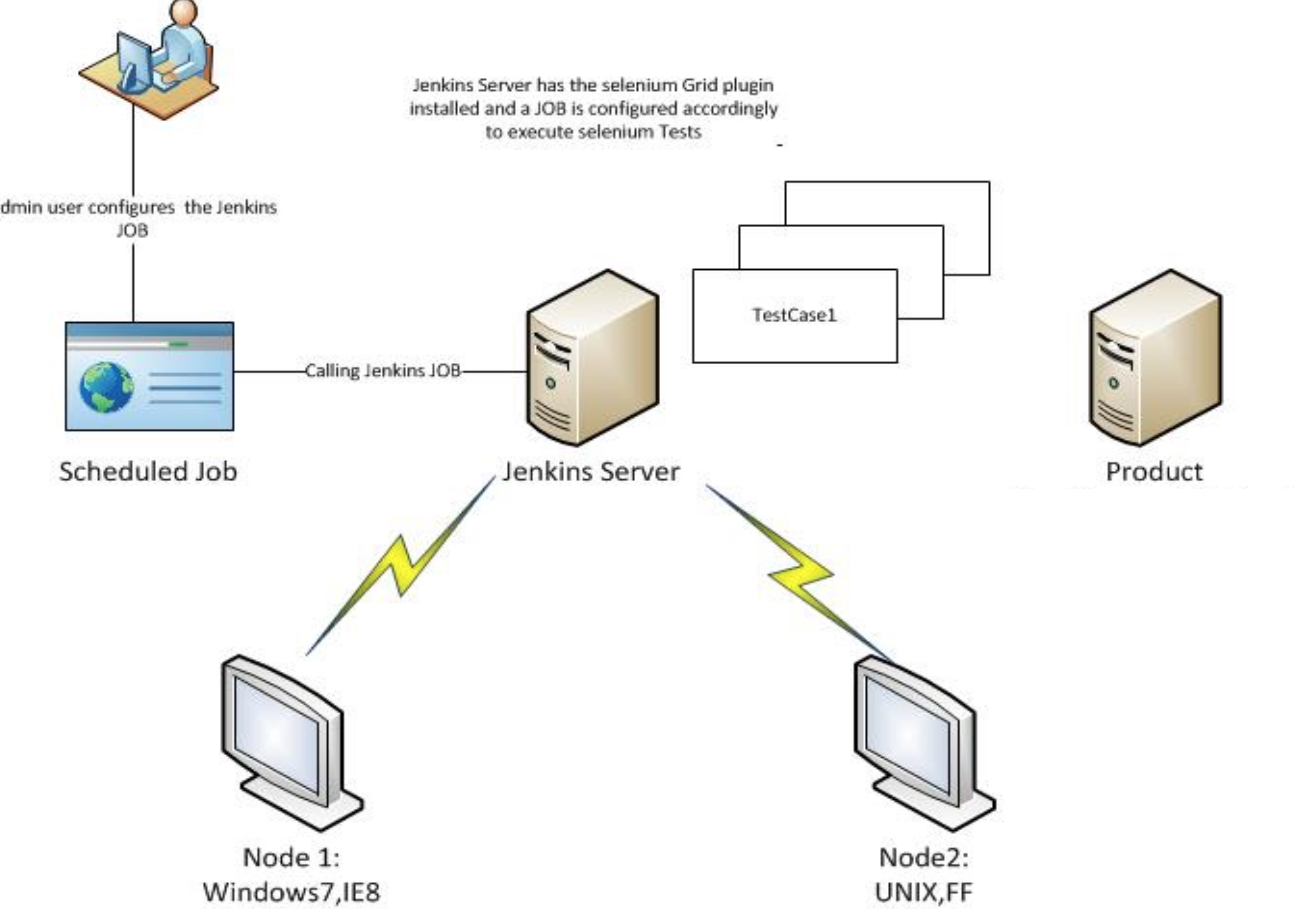
## **Connecting to Selenium Grid**

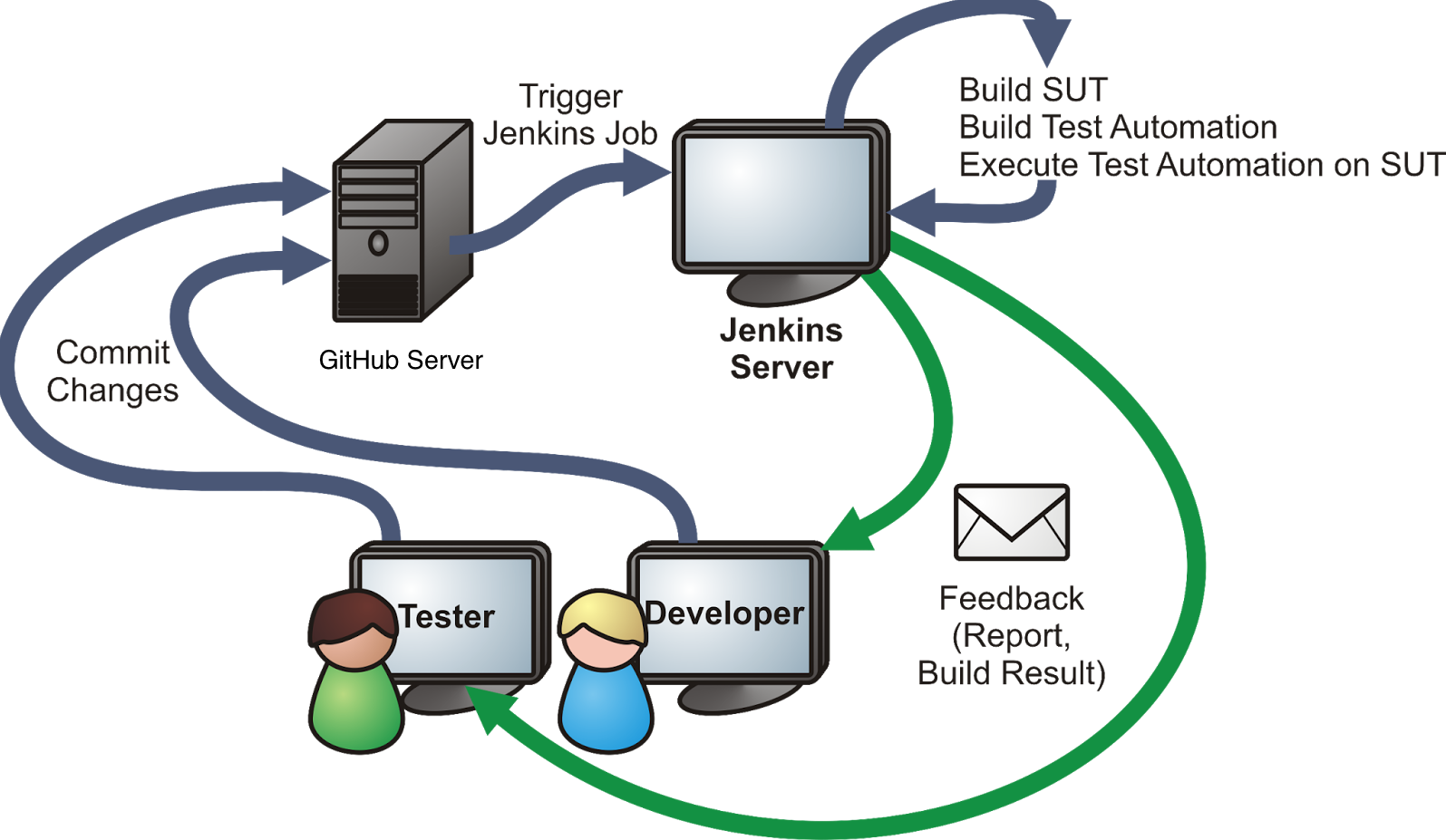
When we run selenium tests in stand-alone Selenium, you must specify the type of the browser in the constructor.

WebDriver driver = new RemoteWebDriver(new URL("http://jenkins.mydomain:4444/wd/hub"), capability);

In addition to standard platform matching capability offered out-of-the-box by Selenium Grid, Jenkins allows you to specify "jenkins.label" as a capability, whose value is an expression of label names to narrow down where to run the tests.

DesiredCapabilities capability = DesiredCapabilities.firefox();  
// say you use the redhat5 label to indicate RHEL5 and the amd64 label to specify the architecture  
capability.setCapability("jenkins.label","redhat5 && amd64");  
// Say you want a specific node to thread your request, just specify the node name (it must be running a selenium configuration though)  
capability.setCapability("jenkins.nodeName","(master)");





.