1: What is the test automation framework? What is selenium?  how does it work? and why do you need it?

**Test automation framework**

**“Test automation frameworks are a set of rules and corresponding tools that are used for** **building test cases. It is designed to help engineering functions work more efficiently.”**

The general rules for automation frameworks include coding standards that you can avoid manually entering, test data handling techniques and benefits, accessible storage for the derived test data results, object repositories, and additional information that might be utilized to run the tests in a suitable manner.

**Selenium:**

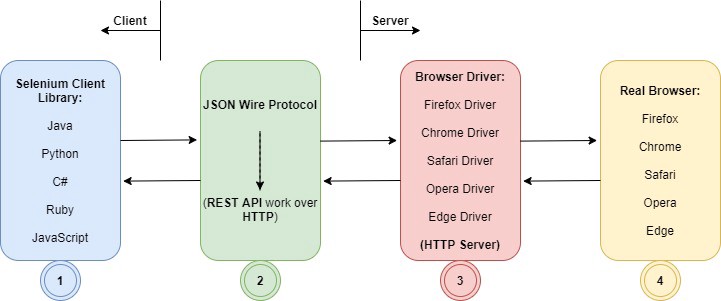
The [Selenium is a framework](https://www.browserstack.com/guide/selenium-framework/) that most widely used automation framework structure that increases code readability and hence improves test productivity. The ‘test case’ and ‘data’ are separately kept from each other to test how efficiently a web page can run. [Selenium](https://www.browserstack.com/selenium) automation frameworks can also be utilized by executing test cases from an external source.

This eliminates repetitive [manual testing](https://www.browserstack.com/guide/manual-vs-automated-testing-differences) that consumes lots of time and effort. Selenium conforms with the idea of Agile, and DevOps, which endorse the continuous delivery workflow. Thus, Selenium has been one of the favorite tools for testing as it meets the requirement of quick and reliable testing, which helps enterprises save time and money on testing.

**How does it work?**

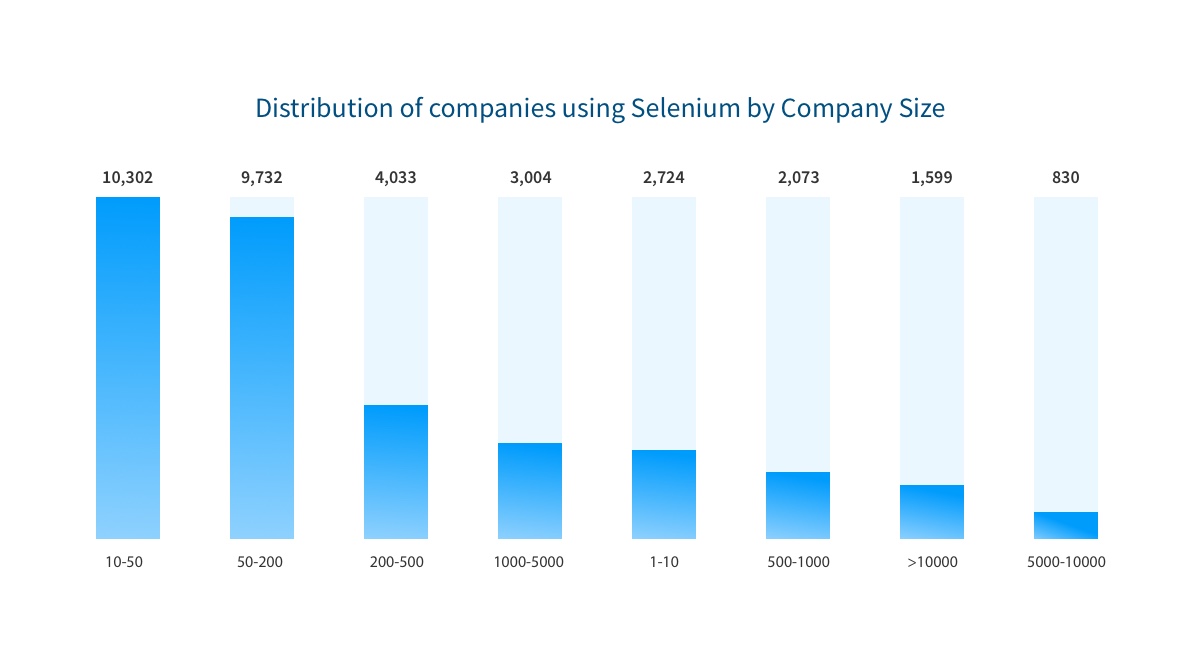
get started with the Selenium automation testing, you can go through [Browser Stack’s Automate documentation](https://www.browserstack.com/docs?product=automate). Decide which scripting language and framework you are planning to use for testing. And, then you can install the prerequisite components.

Selenium works with the client-server design. Client-server design is a software architecture model that consists of two parts namely a client system and a server system that communicates either through a computer network or on the same computer. For more details, let’s look at the following Selenium architecture

****

**Why do you need it?**

 When there are a large number of data sets to be tested for the web application. Then you must opt for a data-driven framework, which separates them from the actual code.It is widely used for testing among the bigwigs like Google, Netflix, Fitbit, etc. to the emerging startups such as likewise. Here’s a graph that shows how Selenium framework is extensively used among various companies of different scale and sizes.

****

1. <https://www.browserstack.com/guide/best-test-automation-frameworks#:~:text=Test%20automation%20frameworks%20are%20a,engineering%20functions%20work%20more%20efficiently>.
2. [**https://medium.com/easyread/how-selenium-works-1414c3ac5d3f**](https://medium.com/easyread/how-selenium-works-1414c3ac5d3f)
3. <https://www.browserstack.com/guide/selenium-framework>

2 : The most common tools that are used for configuration management are packer and ansible. You need to concisely compare both of them.

**What is Ansible?** Radically simple configuration-management, application deployment, task-execution, and multi-node orchestration engine*.* Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. Ansible’s goals are foremost those of simplicity and maximum ease of use.

**What is Packer*?***Create identical machine images for multiple platforms from a single source configuration*.* Packer automates the creation of any type of machine image. It embraces modern configuration management by encouraging you to use automated scripts to install and configure the software within your Packer-made images.

|  |  |
| --- | --- |
| **Ansible** | **Packer** |
| Ansible's natural automation language allows sysadmins, developers, and IT managers to complete automation projects in hours, not weeks. | Super fast infrastructure deployment. Packer images allow you to launch completely provisioned and configured machines in seconds, rather than several minutes or hours. |
| Ansible uses SSH by default instead of requiring agents everywhere. Avoid extra open ports, improve security, eliminate "managing the management", and reclaim CPU cycles. | Multi-provider portability. Because Packer creates identical images for multiple platforms, you can run production in AWS, staging/QA in a private cloud like OpenStack, and development in desktop virtualization solutions such as VMware or VirtualBox |
| Ansible automates app deployment, configuration management, workflow orchestration, and even cloud provisioning all from one system. | improved stability. Packer installs and configures all the software for a machine at the time the image is built. If there are bugs in these scripts, they'll be caught early, rather than several minutes after a machine is launched. |
| Ansible is agentless, so your build toolchain has fewer moving parts. | The ansible Packer provisioner runs Ansible playbooks  Various plate form – Azure, GCE, JSON config |
| Only dependencies – SSH , python,  No need for centeral server | Multiple provisioners –Ansible, sait, bash |

<https://stackshare.io/stackups/ansible-vs-packer>

<https://cloudblogs.microsoft.com/opensource/2018/05/23/immutable-infrastructure-azure-vsts-terraform-packer-ansible/>