Test Automation Frameworks

* 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, test data handling techniques and accessible storage for the derived test data results.[1]

##### **Selenium Framework And Work**

* The Selenium framework is the 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.
* It can also be utilized by executing test cases from an external source.
* 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.
* For instance, when there are more functionalities to be tested for the web application, it is suggested that the operations are stored in a separate table.
* These keywords are called to use a specific functional operation.
* On occasions when data sets and functionalities both are high in number, use a hybrid driven framework to avoid complexity.

**Ansible:**

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or rolling updates. Ansible's goals are foremost those of simplicity and maximum ease of use.

**Packer:**

Packer automates the creation of any type of machine image from a single source configuration. It encourages you to use automated scripts to install and configure the software within your Packer-made images.

**Ansible vs Packer**

|  |  |
| --- | --- |
| * Ansible's natural automation language allows sysadmins, developers, and IT managers to complete automation projects in hours, not weeks. * Ansible uses SSH by default instead of requiring agents everywhere. Avoid extra open ports, improve security, eliminate "managing the management", and reclaim CPU cycles. * Ansible automates app deployment, configuration management, workflow orchestration, and even cloud provisioning all from one system. | * Super fast infrastructure deployment. Packer images allow you to launch completely provisioned and configured machines in seconds, rather than several minutes or hours. * 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. * 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. |

Reference:

[1] <https://www.browserstack.com/guide/selenium-framework>

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