Automation Testing

Postman with Jenkins

A Day in the Life of an Automation Testing Engineer

John has got the task of integrating Postman with Jenkins from his manager. This integration helps in establishing CI or CD in a fully automated system. He explores the Postman tool's integration capability with Jenkins for this task.

After this session, John can integrate Postman with Jenkins and create a Jenkins job that triggers all the API tests.

Learning Objectives

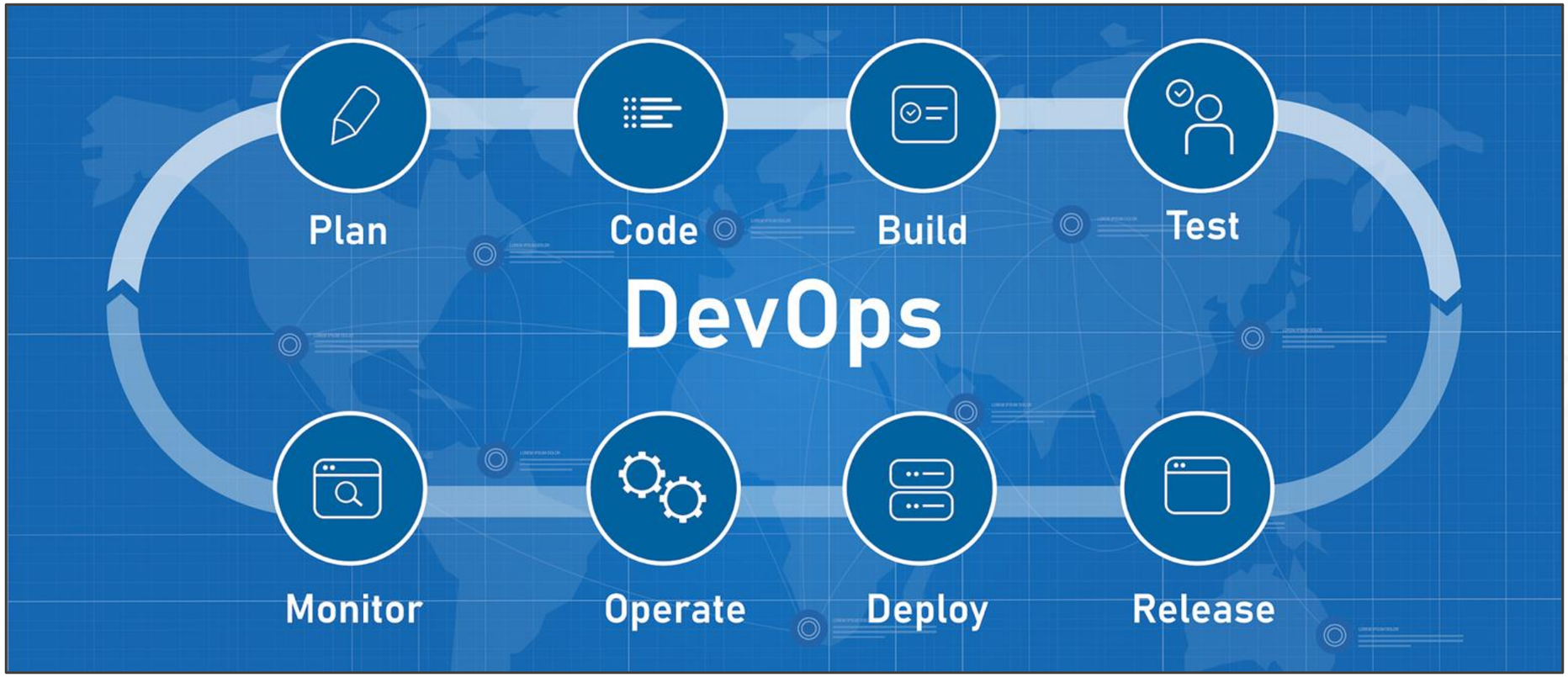
By the end of this lesson, you will be able to:

- Identify Jenkins, its importance, and its advantages
- Explain how to install Jenkins
- State Jenkins plugins
- Explain how to integrate Postman and Jenkins
- List the steps to execute Postman scripts in Jenkins

Continuous Integration Tools

Continuous Integration

Continuous integration refers to integrating a user's program and modifications continually, so they appear in the web repository (or cloud).



Continuous Integration

Development means making changes to the code. While deploying entails integrating to the primary build, testing entails API testing on the new code.

Back-end APIs are crucial for the system as the front end of a project relies on the same API services.

A lot of logic is written for the functionality of API Services.

Continuous Integration

The code is constantly changing, which may impact how well the API functions and how accurate the outcomes of the tests are.

Users must continuously test to gain confidence in the flawless operation of APIs.

The unit tests in the Postman tool validate the code errors.

Test engineers code the Postman tests, while developers code the unit tests.

Continuous integration is carried on to ensure that the above aspects are followed and maintained.

Continuous Integration

Consider a scenario to understand why continuous integration is needed.

Background:

A developer frequently works in a team or group, and the team may be dispersed worldwide or within a single office. Even though they may not have ever met, they all collaborate on the same project, creating unique modules or changing the same code base.

Continuous Integration

The Issue

Now, suppose the users modify an existing code or add a new one.

The Solution

They must review how this update reflects the changes in the program they have made.

Jenkins

What Is Jenkins?

- Jenkins is a Java-based open-source automation server.
- It continually creates and tests projects, making it easier for a developer and a tester to work on the product.
- It employs continuous integration and development for the program's development and deployment, which simplifies a developer's work.

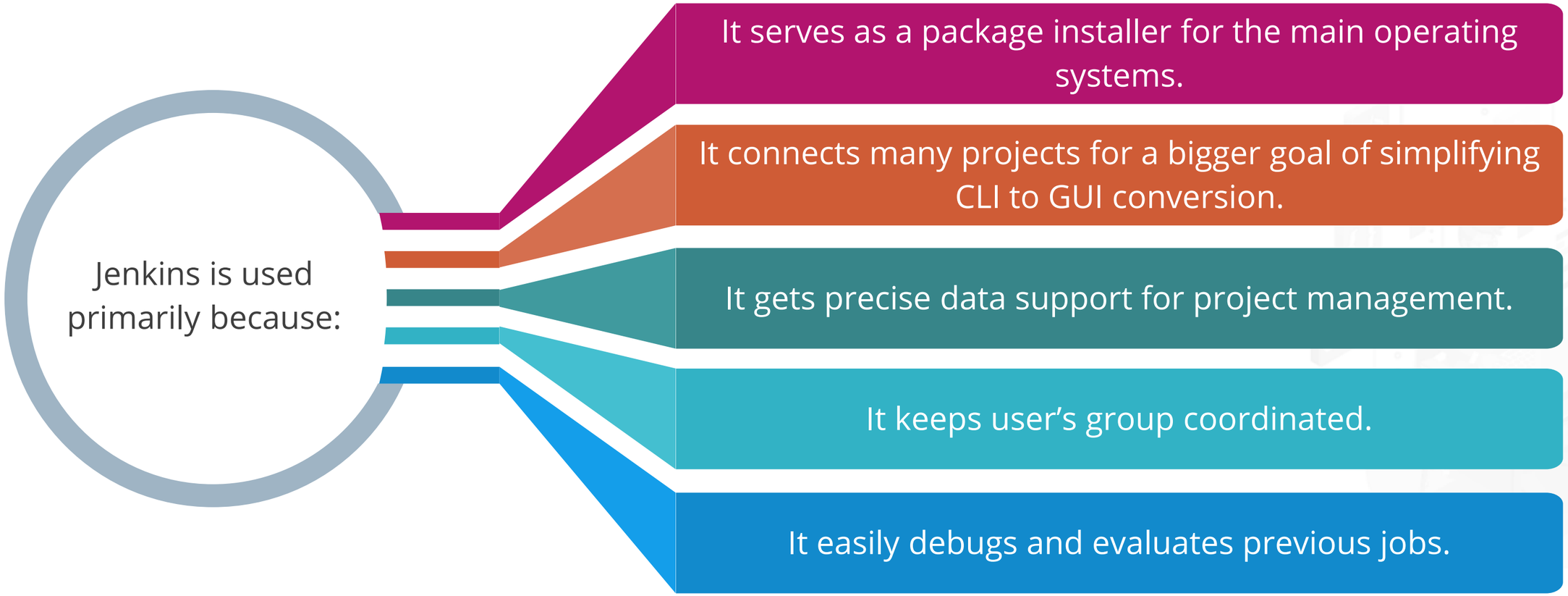
Jenkins in Organizations

Here are some reasons why Jenkins is used in organizations:

Organizations can use Jenkins to automate and speed up the software development process.

Jenkins combines several stages of the development life cycle, such as build, document, test, package, deploy, and static analysis.

Why Jenkins?



Advantages of Jenkins

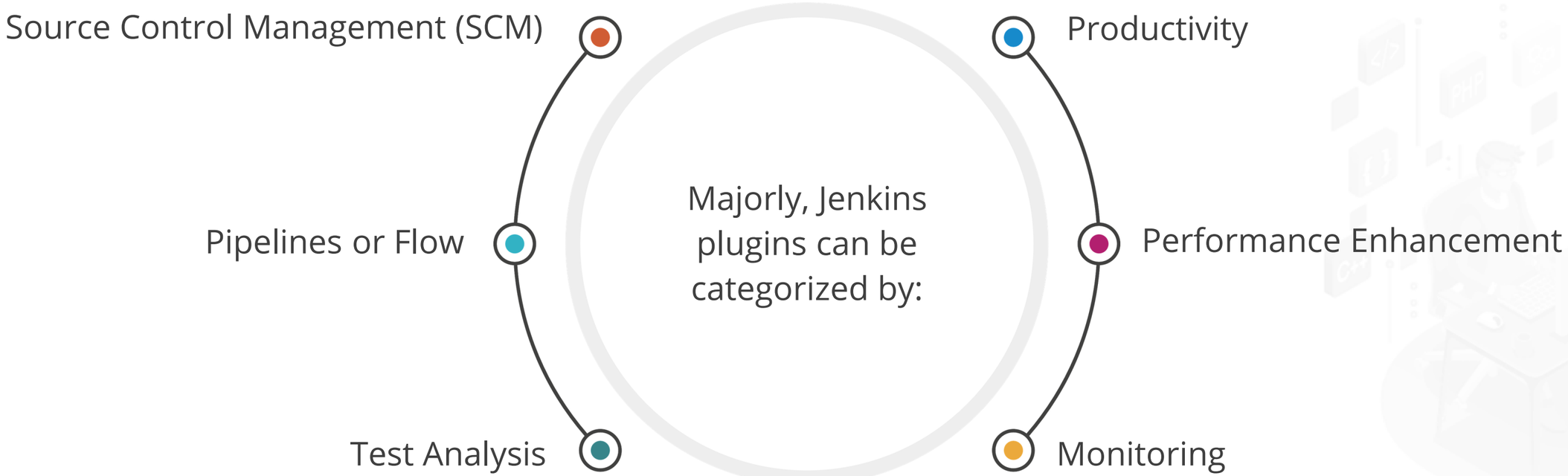
- It is a community-supported open-source tool.
- It can be easily installed.
- It is available as freeware.
- It is available with numerous plugins in Jenkins.
- Jenkins has a plugin for any task; if there is not one, it can be easily created.
- It can run on all the major platforms because it is coded using Java.

Jenkins Plugins

Jenkins Plugins

The 1,700+ plugins offered by Jenkins cover source code management, administration, platforms, UI or UX, building management, and so on. One might use any number of plugins to increase productivity.

Jenkins Plugins



Jenkins Plugins: Productivity

Dashboard View Plugin:

It checks the status of each task used. In an ideal world, this would also have time tracking capabilities for the length of time spent on each job as well as an execution time graph.

View Job Filters Plugin:

It is a pre-built technique to create several views for each of the Jenkins jobs. Users can access various filtering options, including build status, trends, and triggers.

Folders Plugin:

Jenkins jobs may be organized into groups quite effectively with this. The nestable folder this plugin offers make it possible to organize the CI server appropriately.

Jenkins Plugins: Monitoring

Monitoring Plugin:

Jenkins is monitored through the Monitoring Plugin using JavaMelody. This plugin allots charts for CPU, HTTP response time, RAM, and many other metrics.

Metrics Plugin:

It performs health checks and keeps the users updated on the project's status in real-time by making the Dropwizard Metrics API available to Jenkins for the application-level metrics.

Jenkins Plugins: Performance

Performance Plugin:

Through a wide range of popular testing tools such as JMeter, JUnit, or Taurus, the tool offers report capturing capabilities.

Performance Publisher Plugin:

It has the ability to generate trend and global reports for test result analysis, while integrating with all testing tools is a standout feature.

Jenkins Plugins: Jenkins Server Scalability

Kubernetes Plugin: Users can use this plugin to install and remove Jenkins agents if they want to use Kubernetes for their infrastructure. Although switching to Kubernetes is not simple, the benefits are worthwhile.

Self-Organizing Swarm Modules Plugin: Whether users want to use this plugin or the one above depends on whether they are using Kubernetes or Docker Swarm. This plugin provides another way to start up and remove Jenkins slaves.

Amazon ECS Container Service: This plugin manages Jenkins cloud slaves and distributes Docker-based apps on a cluster.

Azure Container Service: The AKS cluster orchestration plugin allows users to use Jenkins to execute a container as an agent.

Jenkins Plugins: Tests Analysis

Test Results Plugin:

It makes test results and execution trends more visible and installation simple. Users may find the results of each test for all the builds by using the Tests Result Analyzer, which provides a variety of graphical representations and a very detailed matrix table.

Bootstrapped-Multi-Test-Result-Report Plugin:

Users can create HTML reports based on test results with this plugin. One of its main advantages is that it can create interactive reports that allow users to see the general picture of all outcomes and the specific results of step statuses.

Jenkins Plugins: Pipeline or Flow

Job DSL Plugin:

It allows users to manage scripts, describe tasks, and update Jenkins jobs. Users can define jobs in their most basic programmatic form using templates and Job DSL. Users can also access many job views, including BuildMonitorView, Build Pipeline View, and CategorizedJobsView. These views provide high visibility into the status of chosen Jenkins jobs.

Build Pipeline Plugin:

It allows users to view upstream and downstream jobs in their build pipeline. It also enables them to establish manual triggers for certain actions that may require supervision before execution. This Plugin is a game-changer for Jenkins as it makes pipelines scriptable and gives Jerkin users a remarkably potent means of creating complex DevOps.

Jenkins Plugins: Source Control Management

SCM API:

It offers a cutting-edge API for communicating with SCM systems and a fully functional event system that enables developers to give users precisely targeted alerts.

Git Plugin:

It gives users access to GitHub, an SCM that many other services use as a repository browser.

GitHub Integration Plugin:

It is the essential plugin for connecting Jenkins with GitHub projects. Users may schedule their build, download code and data files from GitHub repositories to Jenkins, and automatically start each build as needed with the help of this plugin.

Running Scripts from Jenkins

How to Run Scripts from Jenkins?

Postman offers built-in tools that help integrate API with some of the most widely-used Continuous Integration tools, like Jenkins.

Postman can be integrated with Jenkins using Newman, the command-line collection runner for Postman.

Jenkins Alternatives

Alternatives to Jenkins include Thought Works Snap, CircleCI, JetBrains TeamCity, and Atlassian Bamboo.

Why Postman with Newman and Jenkins?

- Jenkins has a much wider selection of plugins than any competing software. Postman also recommends it on its official website.
- Continuous integration, carried out in Postman by Newman and Jenkins, successfully combines tests and codes.
- Jenkins is readily integrated with Postman Newman. This functionality allows developers to assess how APIs are performing following code changes quickly.
- As Postman is integrated with Jenkins with the aid of Newman, once a change is pushed, Jenkins will then use Newman to perform the Postman Collections.

How to Run Scripts from Jenkins?

Prerequisites

To integrate the API Test in a CI pipeline, users essentially need three files:

Collection:
The Collection of requests and tests of the users are defined by Postman Collection.

Environment:
Target Environment variables are defined by Postman Environment.

Jenkinsfile:
It defines our pipeline.

Key Takeaways

- Users will integrate Postman with CI/CD build system using Newman.
- They can set up Jenkins build that uses Newman to run a Collection.
- Jenkins can be installed locally as well on http://localhost:8080.
- The console output can be used to check the results in Jenkins.
- The green check mark indicates the successful execution of Jenkins.