GENROCKET CASE STUDY

CI/CD Pipeline Integration for Insurance Applications

BACKGROUND

A major insurance company was using a traditional Test Data Management (TDM) platform to provision test data for testing a variety of insurance applications. The business is diversified into a number of insurance categories including life, health, accident, critical illness, dental, vision, disability and related products such as annuities and retirement planning. Their insurance products are intended for both individuals and families as well as for businesses and organizations. They have more than 100 applications with complex data structures that all require continuous testing to maximize quality and minimize time to market.

Their TDM solution was proving to be costly and complicated. At the same time, manually provisioned test data had become a bottleneck as they streamlined their software development process into a ***Continuous Integration*** and ***Continuous Delivery*** (CI/CD) pipeline. They chose the Jenkins framework as the platform for automating all aspects of the build/test/release function.

Jenkins is the leading platform for CI/CD pipelines. It is open source, scalable, intuitive and supports a wide array of plugins to extend its functionality. Jenkins allows developers to compile code using a shared version control repository to accelerate the development process without introducing unnecessary errors. It allows automated testing that facilitates unit and integration testing into the CI/CD pipeline.

IT leaders at this insurance company soon found their TDM solution was too cumbersome to keep pace with the accelerated speed of development. They wanted a more nimble process allowing testers to generate any kind of test data on-demand with a simple self-service provisioning model. They began working with GenRocket and its Test Data Generation (TDG) approach for creating any kind of test data for any application requirement on-demand and in real-time using secure, synthetic test data.

THE TECHNICAL CHALLENGE

To fully address the technical and operational requirements of this insurance company, GenRocket needed to integrate its TDG engine with the Jenkins CI/CD server and its automated test environment. The test data solution needed to be fast, economical, seamless, and easy to use. The goal was to replace the Test Data Management system and its centralized provisioning process with a more automated, cost-effective and decentralized approach.

Equally important was the requirement to provision test data representing any of the complex data structures associated with any of the insurance applications with full referential integrity preserved.

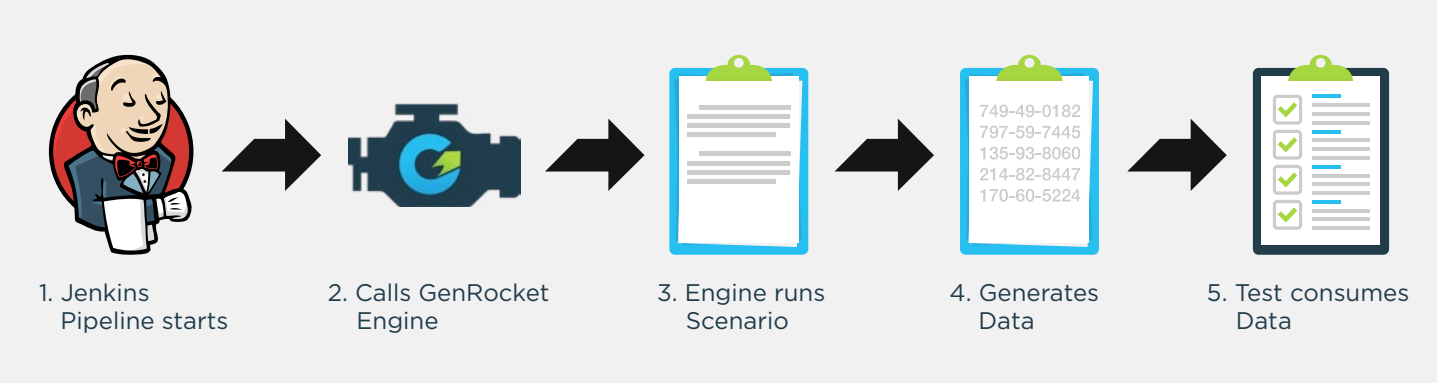
To that end, GenRocket developed an integration with Jenkins to incorporate real-time synthetic test data generation into a CI/CD pipeline to enable continuous end-to-end testing that ensures quality while keeping pace with the speed of development

GENROCKET'S JENKINS SOLUTION

Jenkins can easily run the GenRocket Engine to generate synthetic test data on-demand. Here's how it works. The Jenkins pipeline starts and calls the GenRocket Engine using the GenRocket API. Then the GenRocket Engine runs a test data Scenario to generate the data. The GenRocket Scenario specifies the type of patterned and conditioned test data needed and is based on a data model that represents the production data. Jenkins then consumes the test data and executes the appropriate test scripts. The process is seamless, easy to configure and completely automated.

How Jenkins can easily run the GenRocket Engine to generate data:

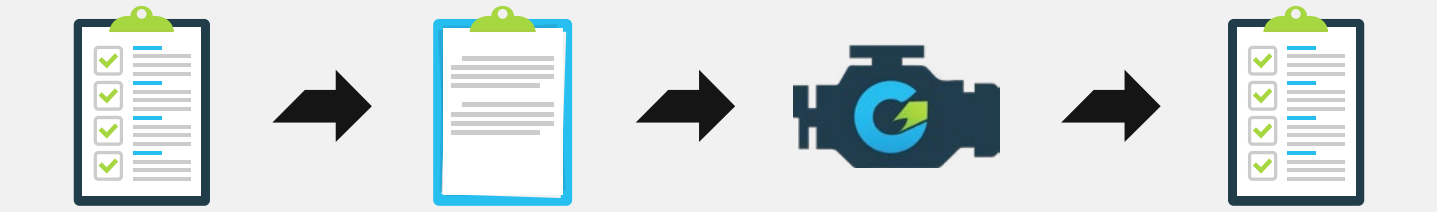
- Jenkins can call GenRocket Engine
- The GenRocket Engine runs the Scenario to generate the data
- Jenkins can consume the test data



GENROCKET SUPPORT FOR A JENKINS AUTOMATION FRAMEWORK

To support automated testing, GenRocket Scenarios can be called via an automation framework (e.g. Selenium). The GenRocket API can then be used to modify GenRocket Scenarios in real time for each test case. Based on the data requirements (i.e. CSV files, databases, etc.) GenRocket can populate the required data sources in the format required by the test cases.

- GenRocket Scenarios can be called via an automation framework.
- The GenRocket API can be used to modify GenRocket Scenarios in real time.
- GenRocket can populate required sources with data so they can be used by the test cases.

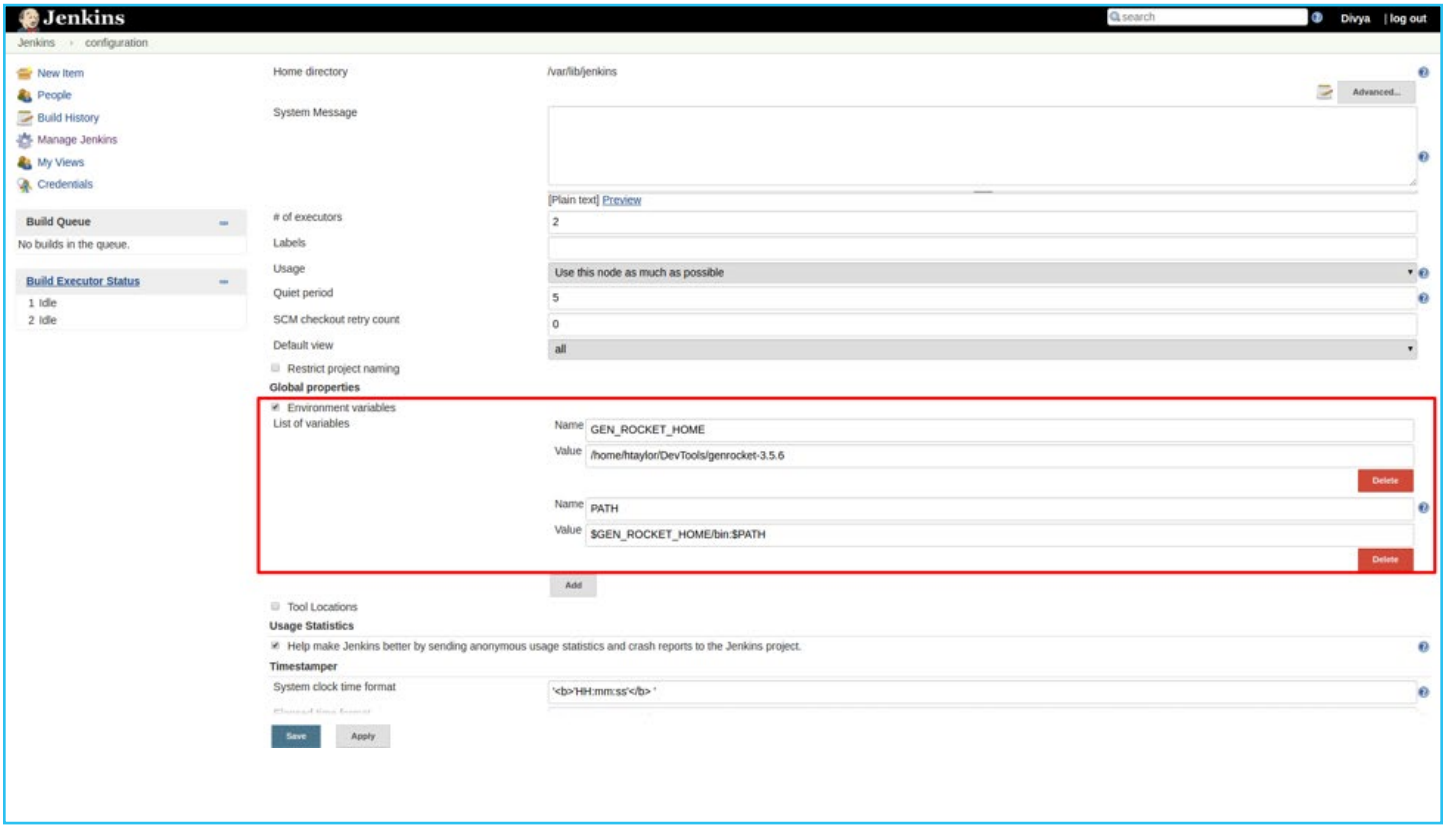


1. Test Case starts
2. Test Case calls GenRocket Scenarios
3. Test Data is generated and inserted into Test Case
4. Test Case is executed and returns Results

EASE OF DEPLOYMENT

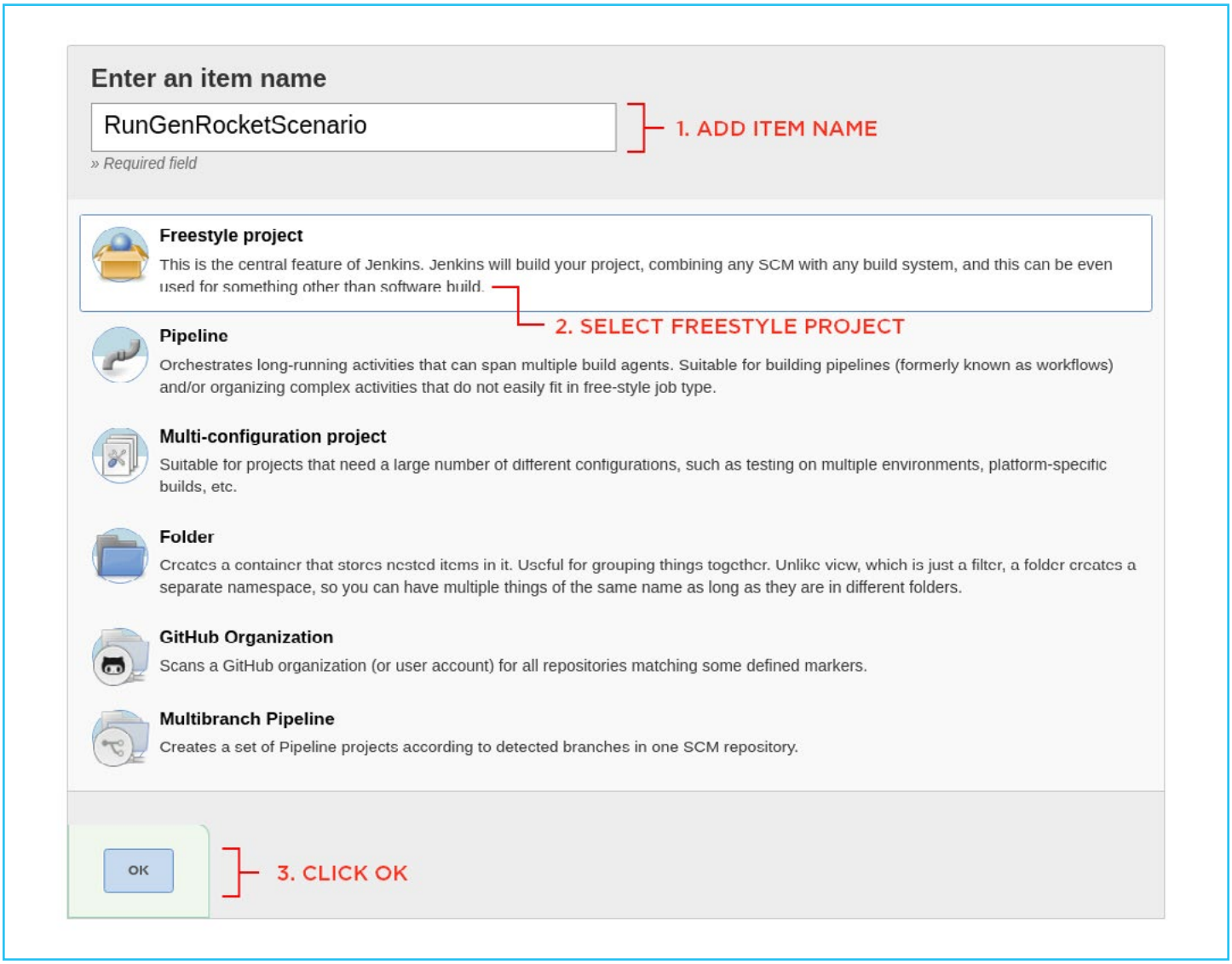
Deploying GenRocket into your CI/CD pipeline is easy. There are just a few simple steps to take and they are described below to provide a basic understanding of the process. A more detailed explanation is available in the resource library on the GenRocket website.

Simply install the GenRocket runtime on the Jenkins Server. Downloading and installing the GenRocket runtime on the Jenkins server is similar to downloading and installing GenRocket on any local computer. Now configure the Jenkins server. Once installed, set the *Environment* variable on the Jenkins server via the Jenkins GUI (see screenshot below).



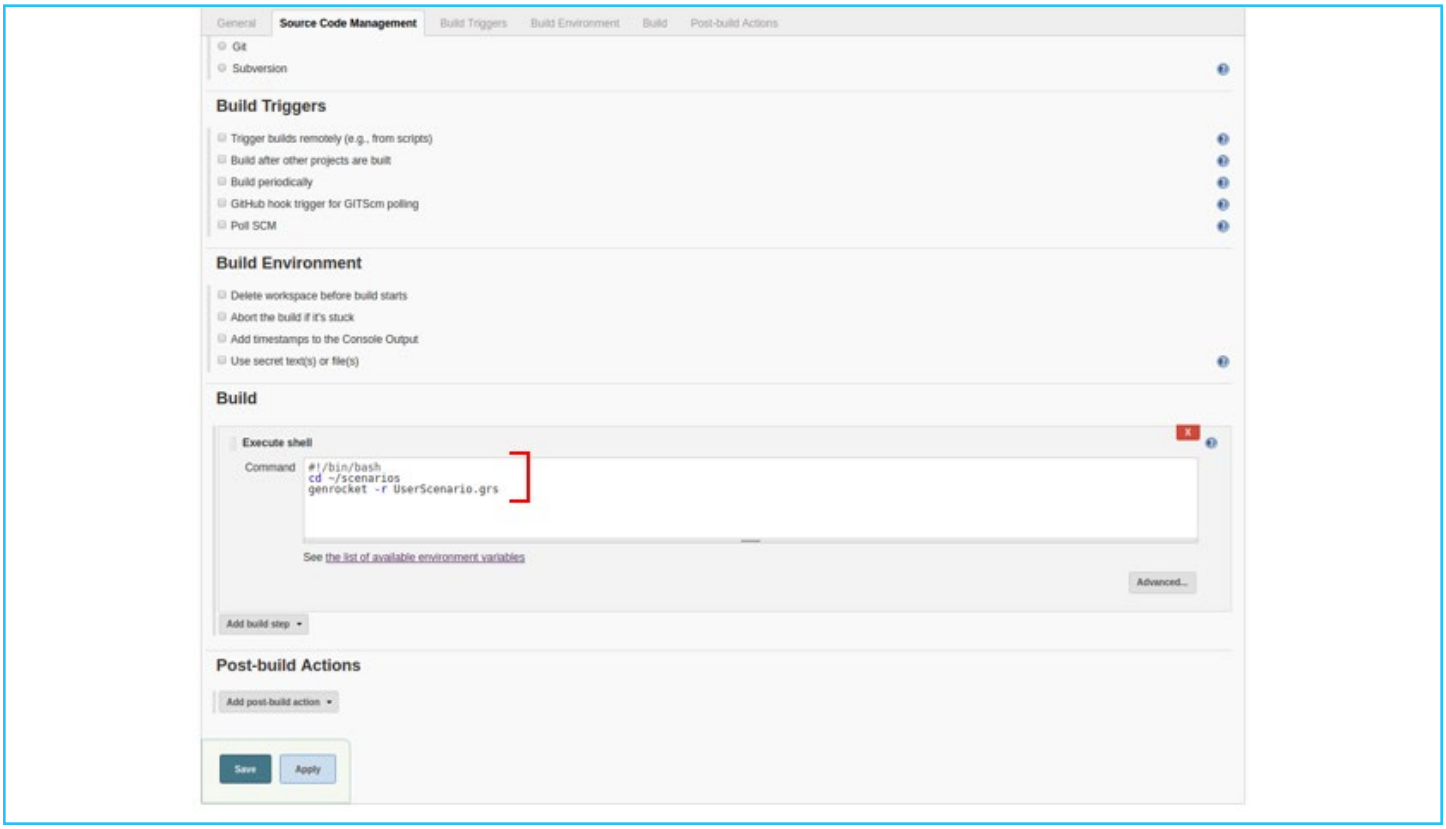
Add the Jenkins server to the GenRocket platform and update the resource variables with the appropriate values with respect to Jenkins Server. Once configured the *"ServerProfile.grp"* can be placed in *.genrocket* folder of Jenkins server (see screenshot below).

Create a Jenkins *Job* to run GenRocket *Scenarios*. In Jenkins, create a new *Freestyle* project and add the shell script to execute the GenRocket *Scenarios* (see screenshot below).



Here's an example of the Shell script which will execute the Scenarios in $HOME/scenarios folder. You can easily modify this script to run your *Scenarios.*

```
#!/bin/bash
cd ~/scenarios
genrocket -r UserScenario.grs
```



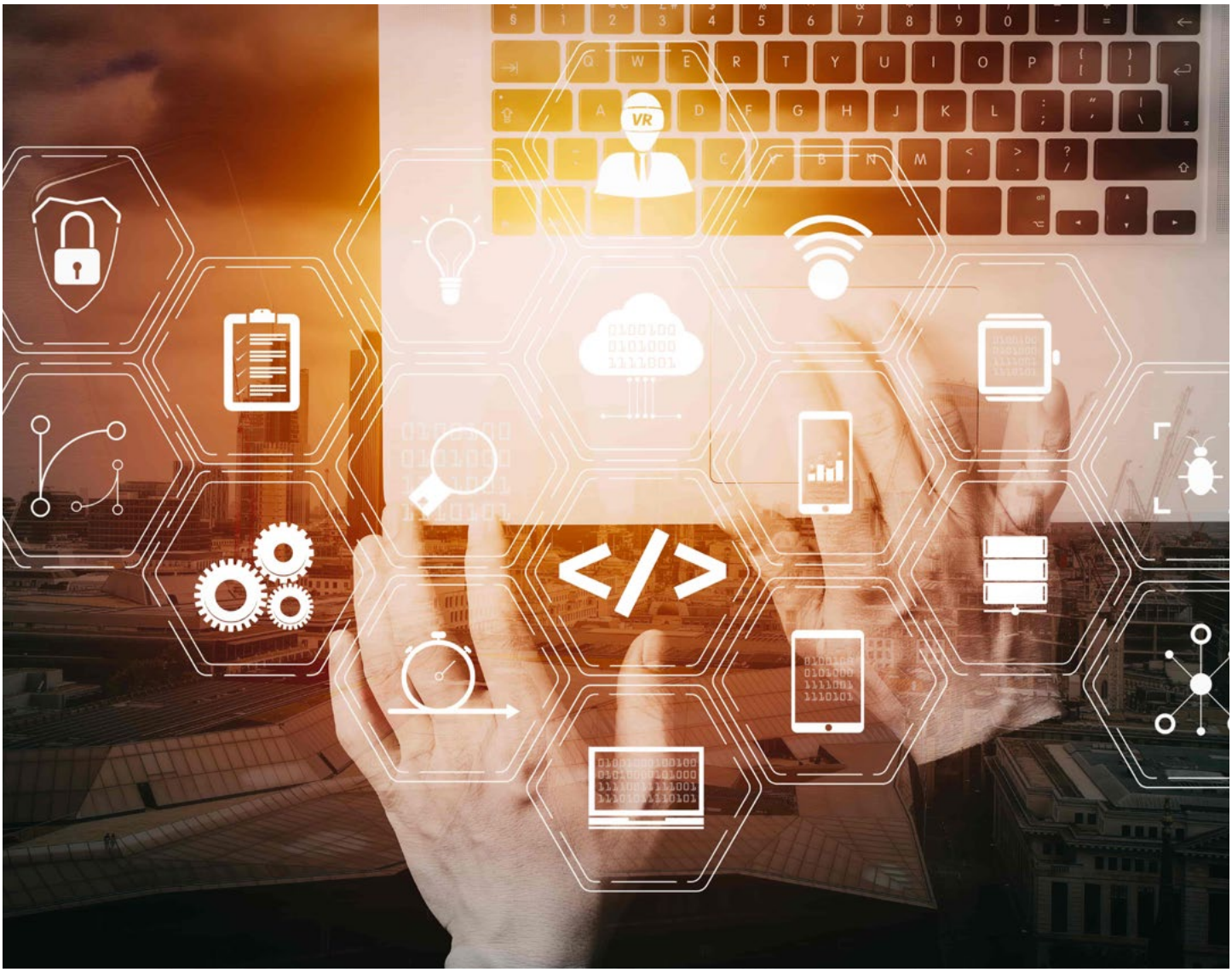
Now run the Jenkins job and view the output (see screenshot below). When the Jenkins pipeline starts, Jenkins will call the GenRocket Engine. The GenRocket Engine runs the Scenario to generate the data and Jenkins can consume the test data via its automation framework.

This easy deployment process allows anyone to provision complex test data for any test case on-demand to perform end-to-end testing in a CI/CD environment.

THE OUTCOME

Using GenRocket's Test Data Generation platform, this insurance company was able to meet all of their test data challenges and realize the following important benefits:

- Generate test data for any insurance application on-demand and in real-time
- Integrate with Jenkins to streamline the development process in a CI/CD pipeline
- Automate the testing process using GenRocket's API to generate and consume test data
- Replace a costly and cumbersome TDM process with an easy and cost-effective solution



GenRocket

If you would like to know more about GenRocket's Test Data Generation platform and our industry solutions, please visit our website at www.genrocket.com.