Quality & Testing SIG Talk: Automation with Jenkins and Continuous Integration
Questions and Answers
July 25, 2018

Q: Can Jenkins be integrated with LeanFT 14.0?

A: Yes, as long as you have the latest Jenkins plugin, LFT should work just fine.
You can get details of the latest plugin from the Jenkins Wiki and download the plugin from GitHub, via the plugin site or via the Jenkins plugin manager.

Q: Can you call only VUGen from Jenkins ?

A: No, you can call VUGEN if you want to run a simple test to check that a script works. You can also call LoadRunner Controller to run a performance test, Performance Center, UFT, Lean FT and a variety of other Micro Focus testing tools.

Q: If something in the test fails, does it stop the pipeline?

A: If you are running a series of tests in a pipeline, a failure in a test will stop subsequent tests. This is ideal if you want to double check that a site works functionally (perhaps using UFT) before running a performance test. It is also possible to run multiple tests in parallel via Jenkins, provided that you have enough slave nodes configured to run tests in parallel.

Q: Would it be over kill to use this plugin to only push test results in ALM? If yes, is there a more lightweight alternative?

A: This depends on your exact requirements. It is perfectly possible to run tests directly from ALM and store the results in ALM without using Jenkins. Jenkins can be useful if you want to incorporate other steps into your continuous development and testing processes. Nor example, you may want to build servers, then deploy software, then run tests before storing the results in ALM. It may be impractical to perform some of the earlier steps in this process without Jenkins or other orchestration tools.

Q: Does this work with HP's StormRunner Load tool as well as PC and Loadrunner? Do you retrieve the resource information from the work plan or the staffing profile?

A: Yes, Jenkins can integrate with StormRunner Load. The full list of integrations for v5.4 of the automation tools plugin is here:
ALM Lab Management 12.xx
ALM Octane 12.53.20 or later (12.55.4 or later required for direct UFT integration and for PC integration using pipelines)
Application Lifecycle Management 12.xx
LoadRunner 12.xx

Mobile Center 2.0-2.70
Performance Center 12.xx (12.53 or higher required for trend reports)
Service Virtualization 3.80 - 4.20
StormRunner Functional 1.40
StormRunner Load 2.8
UFT Pro (LeanFT) 14.03
Unified Functional Testing (UFT) 12.xx and 14.03

Q: How do you kick off the tests in ALM from Jenkins so that it executes the uft scripts? Would that be just a batch file point to your test set?

A: It's really cool. You setup a tie-in to ALM, and configure against a test set in ALM.

Q: What plugins required for performance center to integrate with jenkins?

A: Just the HPE Automation Tools plugin from the Jenkins plugin site. Please note that this plugin does have a number of dependencies.

Q: If the code is being deployed to 4 different test environments, will you have 4 pipelines or Jenkins can handle this in 1?

A: This depends on how complex you want to make your Jenkins pipelines. It may be that you deploy to all environments at the same time and then perform functional tests in one environment whilst simultaneously performing performance tests in another. It really depends on whet you're trying to achieve. There are a huge number of ways that you could configure Jenkins, we'd recommend thinking about what you want to achieve, proving a few tests or builds in isolation before stringing them together into a more complex automation process. Like most tools, when starting with something new it's good to start small and build complexity as you grow in confidence.

Q: How many test automation engineers do you have at your company? How many applications are tested?

A: I can't go into too much detail but at Lloyds Banking Group we have hundreds of testers and automation specialists testing thousands of applications. Having said that we are a very large company undertaking large-scale business transformation initiatives.

Q: Are your test automation engineers skilled to develop both UFT test scripts and also LoadRunner VUGen scripts/ Controller test scenarios?

A:Some of our engineers are, but it is more common to find functional testers specializing in UFT or other functional test tools such as Selenium and separate teams of performance testers using LoadRunner and Performance Center.

Q: You used the one pipeline job to kick-off the other 5 jobs. could you export the results from all five test sets back to your test management tool (ALM or Octane?)

A: Yes you could, you'd add a build step to transfer the results back to ALM after each test completed.

Q: Does Jenkins have its own built in reports or does it just report what other tools post back?

A: Jenkins doesn't have it's own reporting tools. It is basically a clever task scheduler which starts various tasks on a scheduled basis or based on event triggers.

Q: What are the minimum versions of ALM, PC,UFT that can support Jenkins plugin?

A: This is all documented on the Jenkins plugin page.   For v 5.4 the latest version so fMicro Focus software that are supported are:
ALM Lab Management 12.xx
ALM Octane 12.53.20 or later (12.55.4 or later required for direct UFT integration and for PC integration using pipelines)
Application Lifecycle Management 12.xx
LoadRunner 12.xx
Mobile Center 2.0-2.70
Performance Center 12.xx (12.53 or higher required for trend reports)
Service Virtualization 3.80 - 4.20
StormRunner Functional 1.40
StormRunner Load 2.8
UFT Pro (LeanFT) 14.03
Unified Functional Testing (UFT) 12.xx and 14.03

Q: Does the plugin support https communication to ALM server?

A: Yes it does.

Q: Is there a good reference document for application delivery teams to use to get started doing their own performance testing?

A: there is a free e-book going over this at a high level:
This book was co-authored by one of the Vivit board members, please get in touch if you have any questions or would like us to arrange other sessions after reading this.
http://www.effectiveperformanceengineering.com/get-the-book/

Q: Our teams consider themselves to be Agile, but most do not do their own performance testing, but, rather, turn to a central testing service to do the performance testing.... This, obviously, does not scale. But teams seem reluctant to take on the challenges of performance testing.

A: Performance testing is a skilled process and relies on many skills that aren't commonly found in functional test teams such as understanding of test data, test environments, performance monitoring as well as stakeholder management and reporting. For this reason it is common to maintain specialized performance test teams centrally in large organisations. This is what Micro Focus themselves do and it is a common approach.
If you'd like to do some performance testing earlier in the lficecycle, you could start small and consider, for example API testing or running simple comparative performance tests before handing over the performance test team. It may be useful to invite performance testers to help you run some preliminary early performance tests to avoid finding out about performance "too late" in the development cycle.

Q: What coverage level have you achieved and do you still have any manual test left which are performed in devops approach?

A: I am not able to answer this since in my role, I support other teams and help them to achieve the most from our investment in Micro Focus and other tools. Manual testing and exploratory testing does still take place and it probably always will. One of the benefits of increasing automation is that we can free up time to allow more exploratory testing, ultimately leading to more improvements and a better user experience.

Q: Having performance test run for short amount of time triggered by Jenkins at the end - is it the only one perf test you are running in the dev process? (e.g. such short test will most likely not capture memory leak - how do you handle that?) Maybe it is about defining goal of such short perf tests?

A: This demo only triggered a short performance test to show that it could be done. These short tests can be good to identify performance problems related to a poor build but a sensible test plan will also include some long running tests to ensure that problems such as memory leaks are captured during the QA process.