

Continuous Performance Evaluation using Open Source Tools

Dr. Andreas Brunnert, RETIT GmbH



Motivation

- Continuous Integration (CI) has reached widespread adoption
- Functional requirements are tested with automated tests
- Non-functional requirements such as performance are often neglected

How can we track performance if the application changes every day?

Every hour?

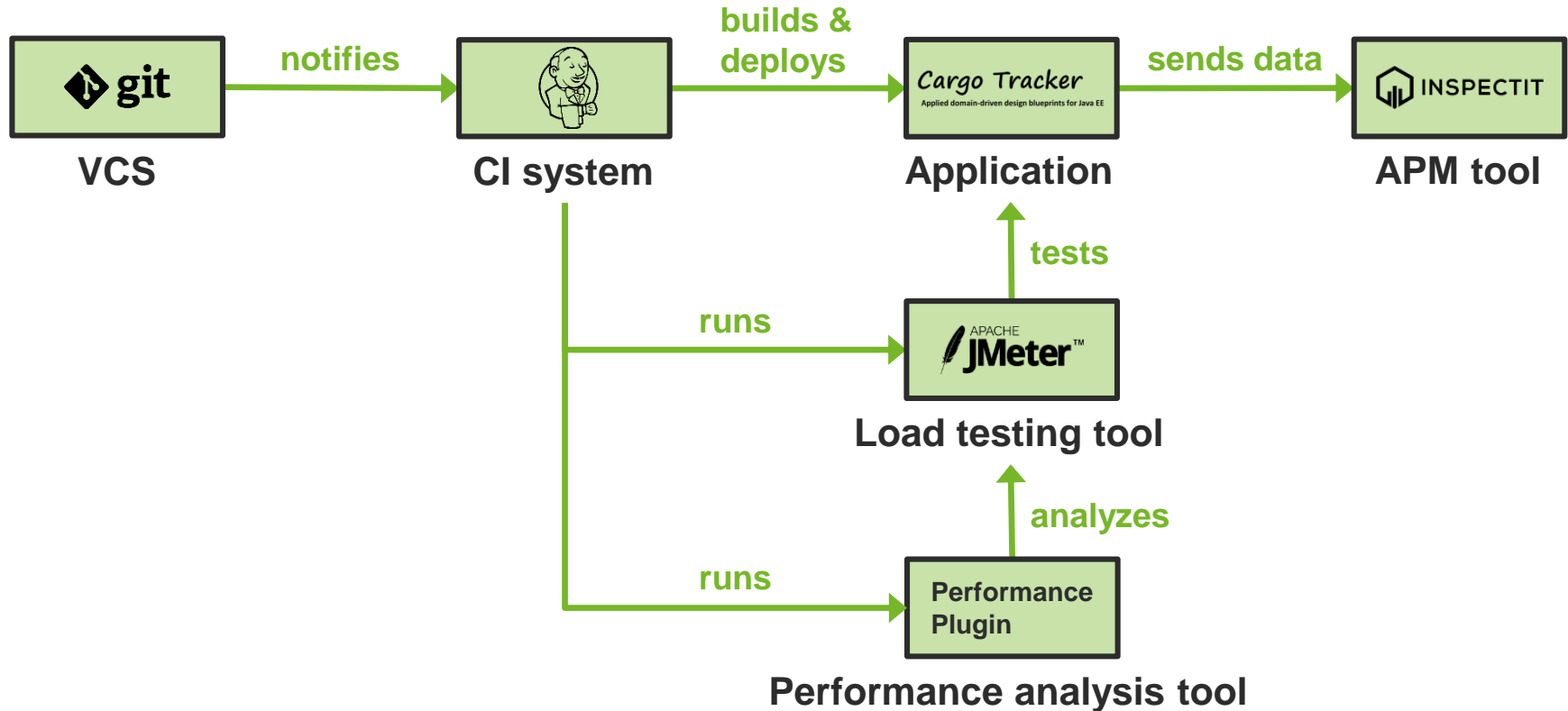
Every minute?

Motivation

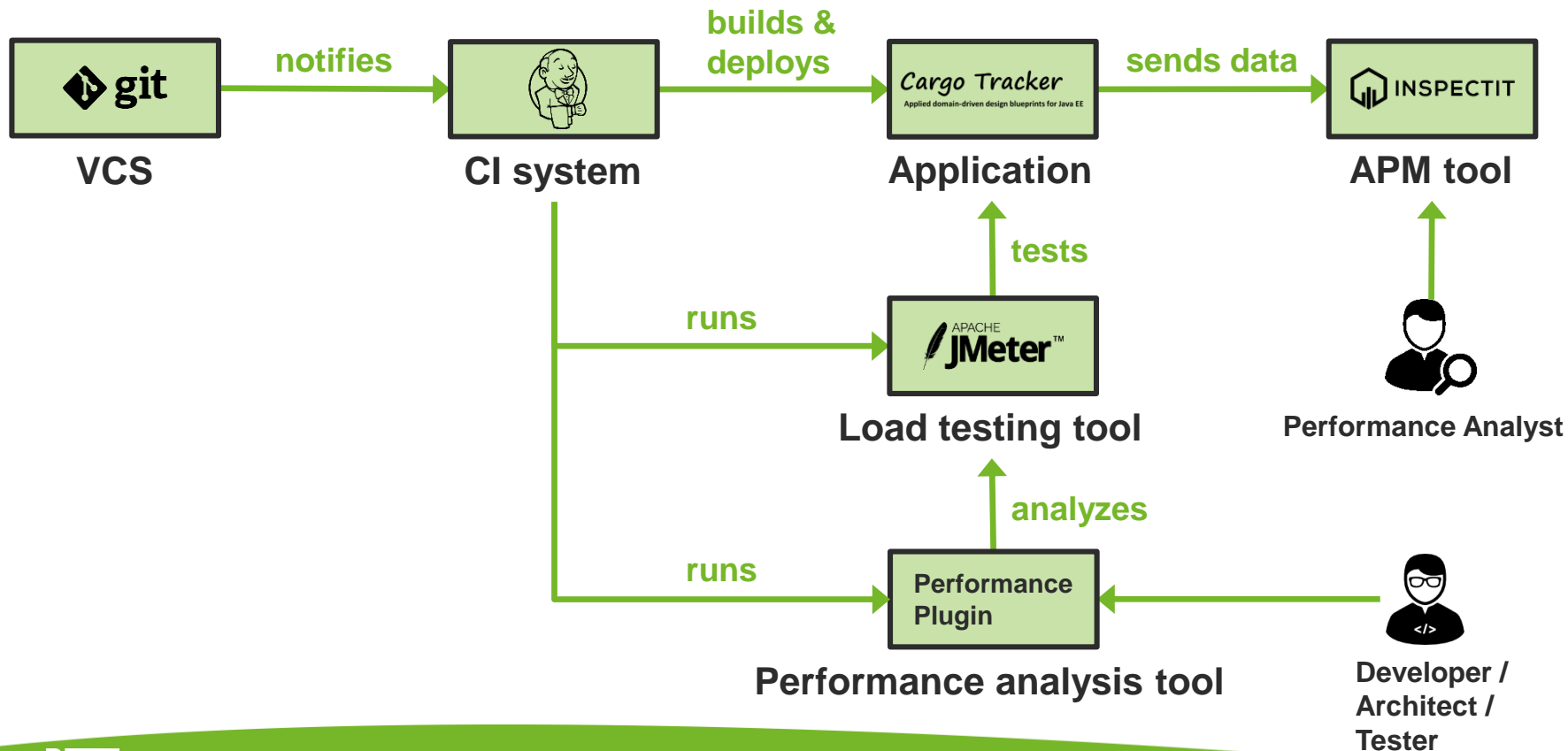
- Modern APM tools often have features to monitor the performance over multiple application versions
- However, they generally come with a high price tag

Can we do it with Open Source Software?

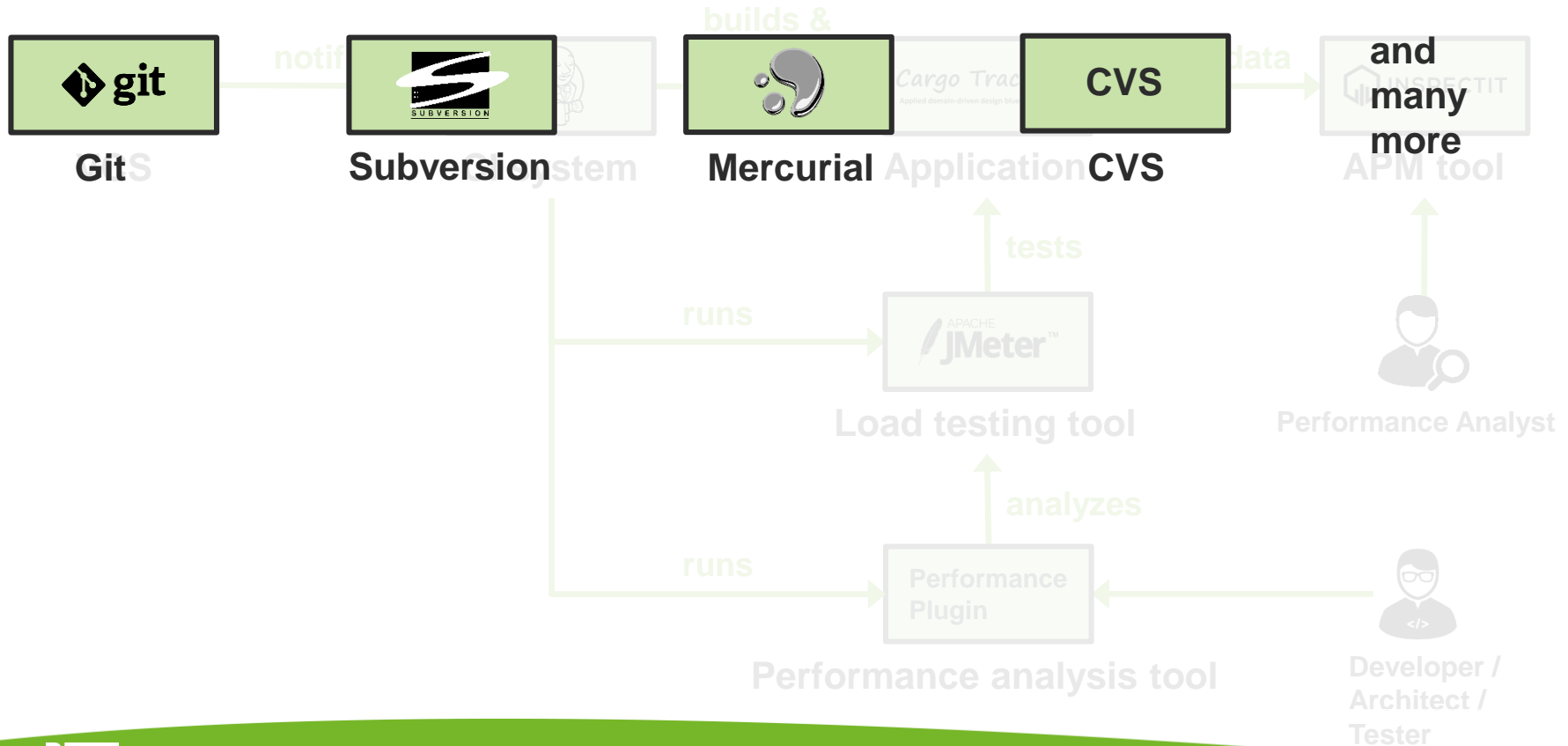
General workflow



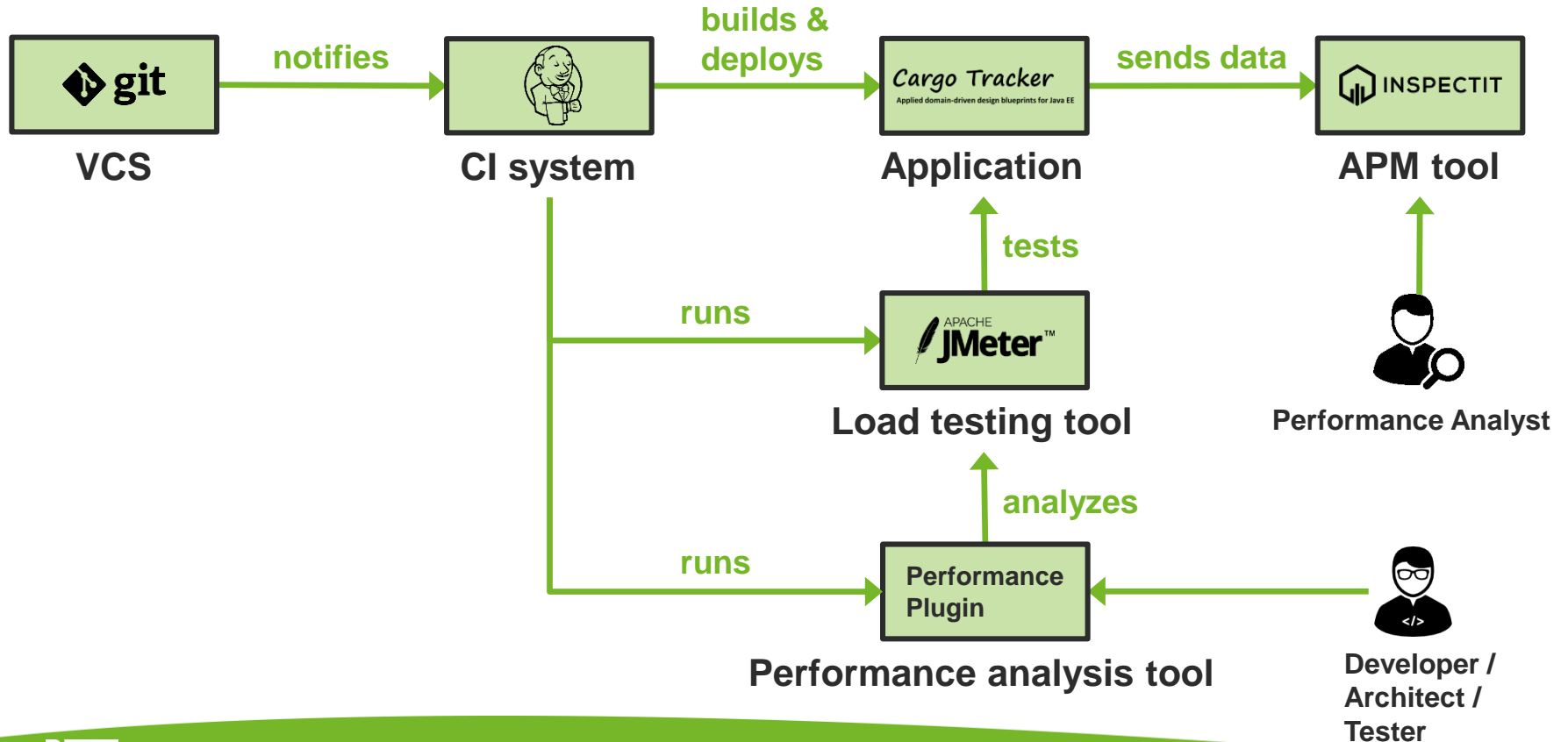
General workflow



Alternatives: Version Control System

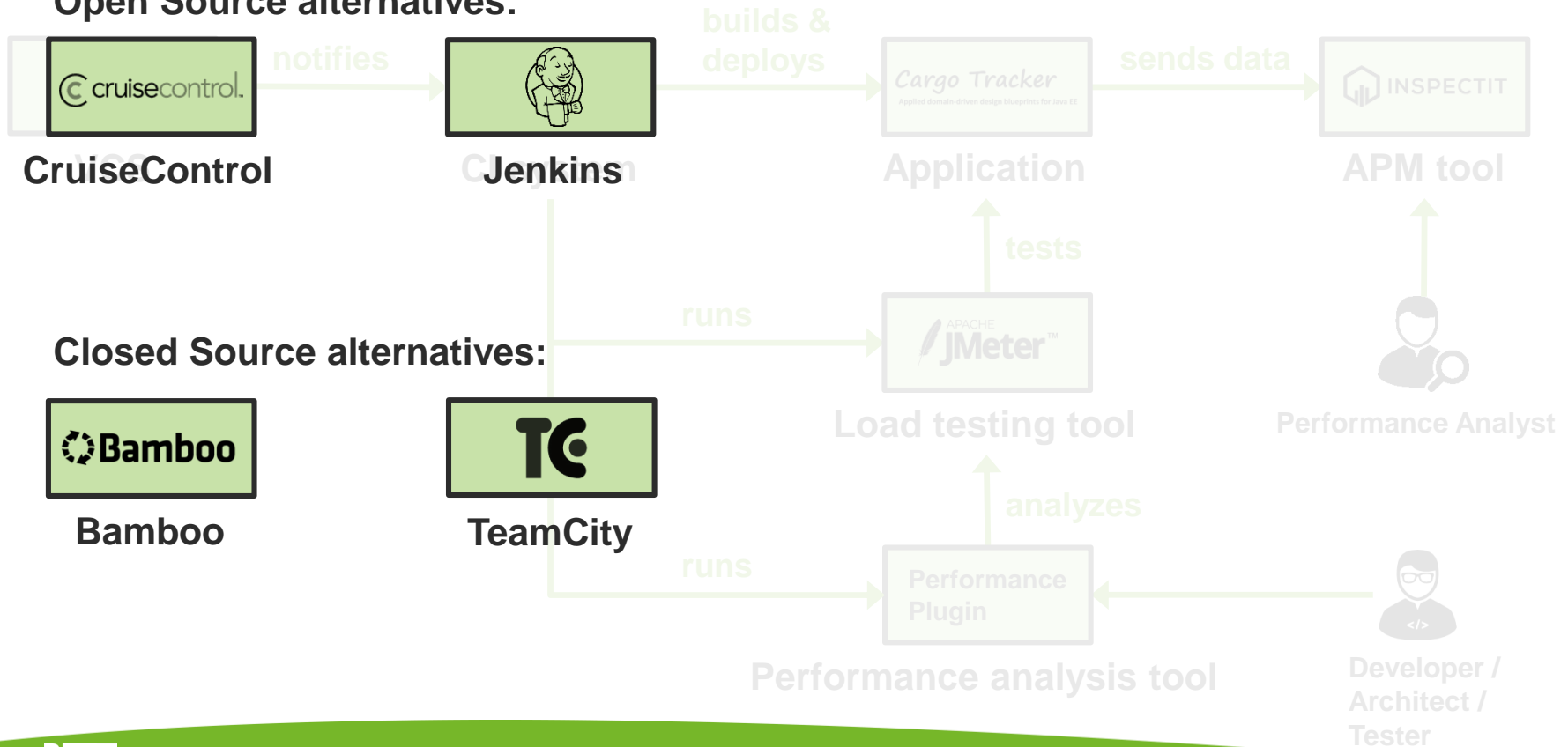


Alternatives: Version Control System

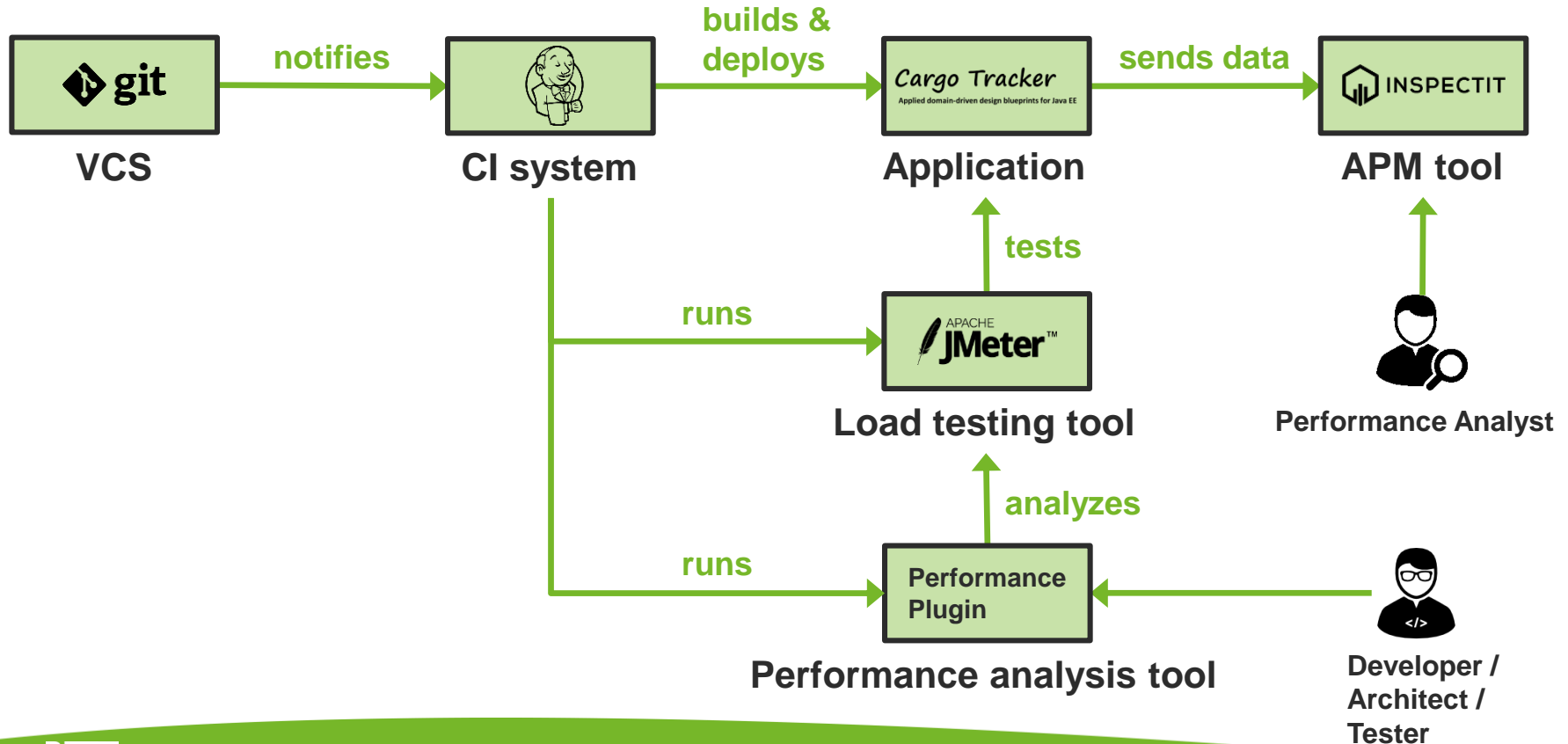


Alternatives: Continuous Integration Systems

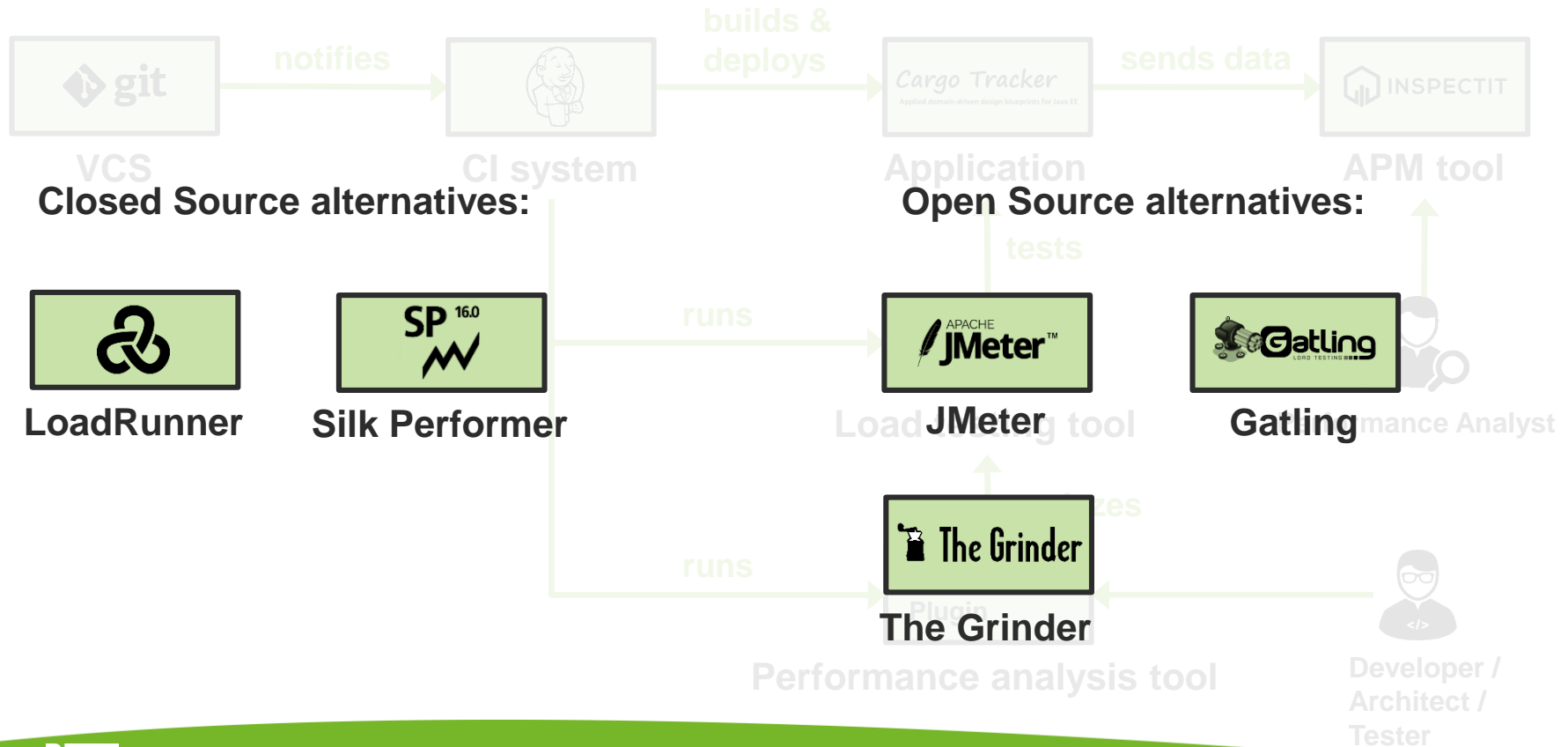
Open Source alternatives:



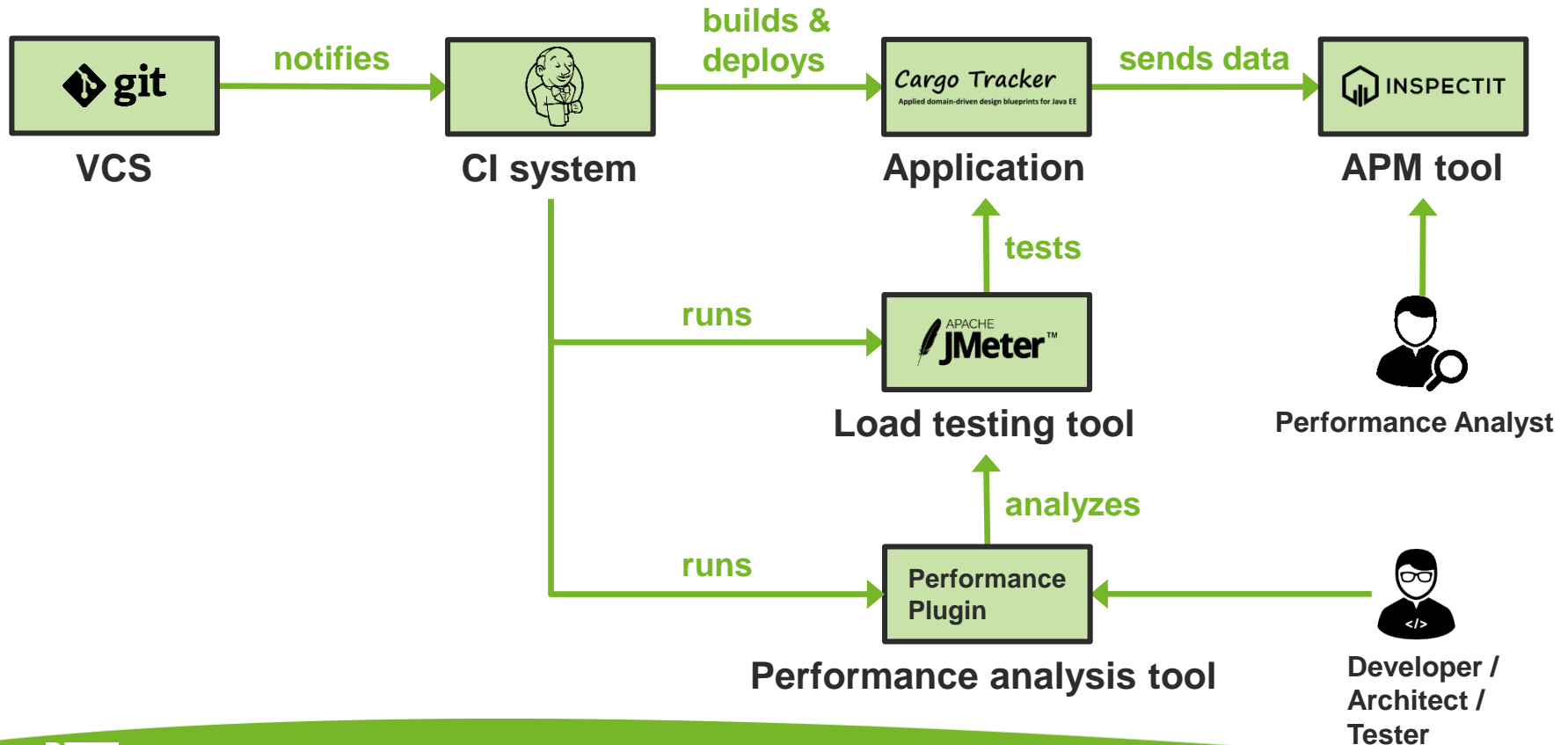
Alternatives: Continuous Integration Systems



Alternatives: Load Testing Tools

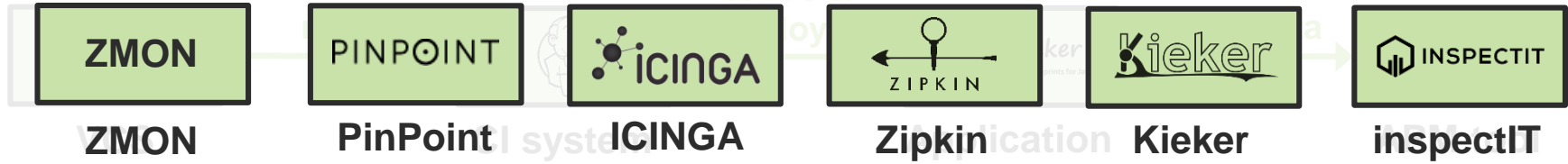


Alternatives: Load Testing Tools

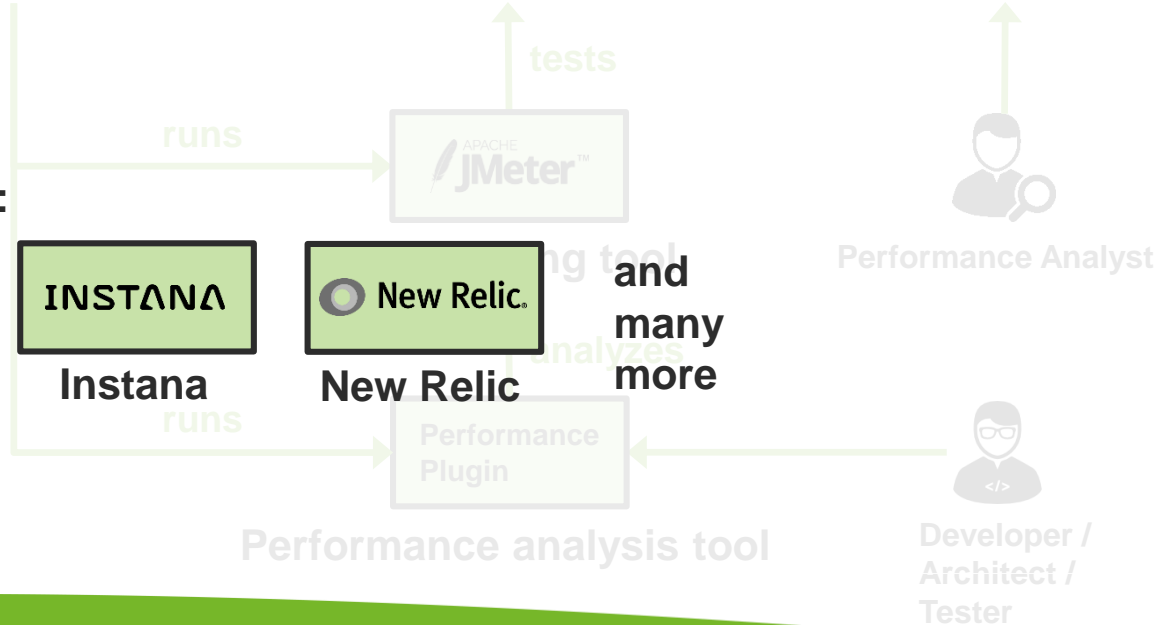


Alternatives: APM Tools

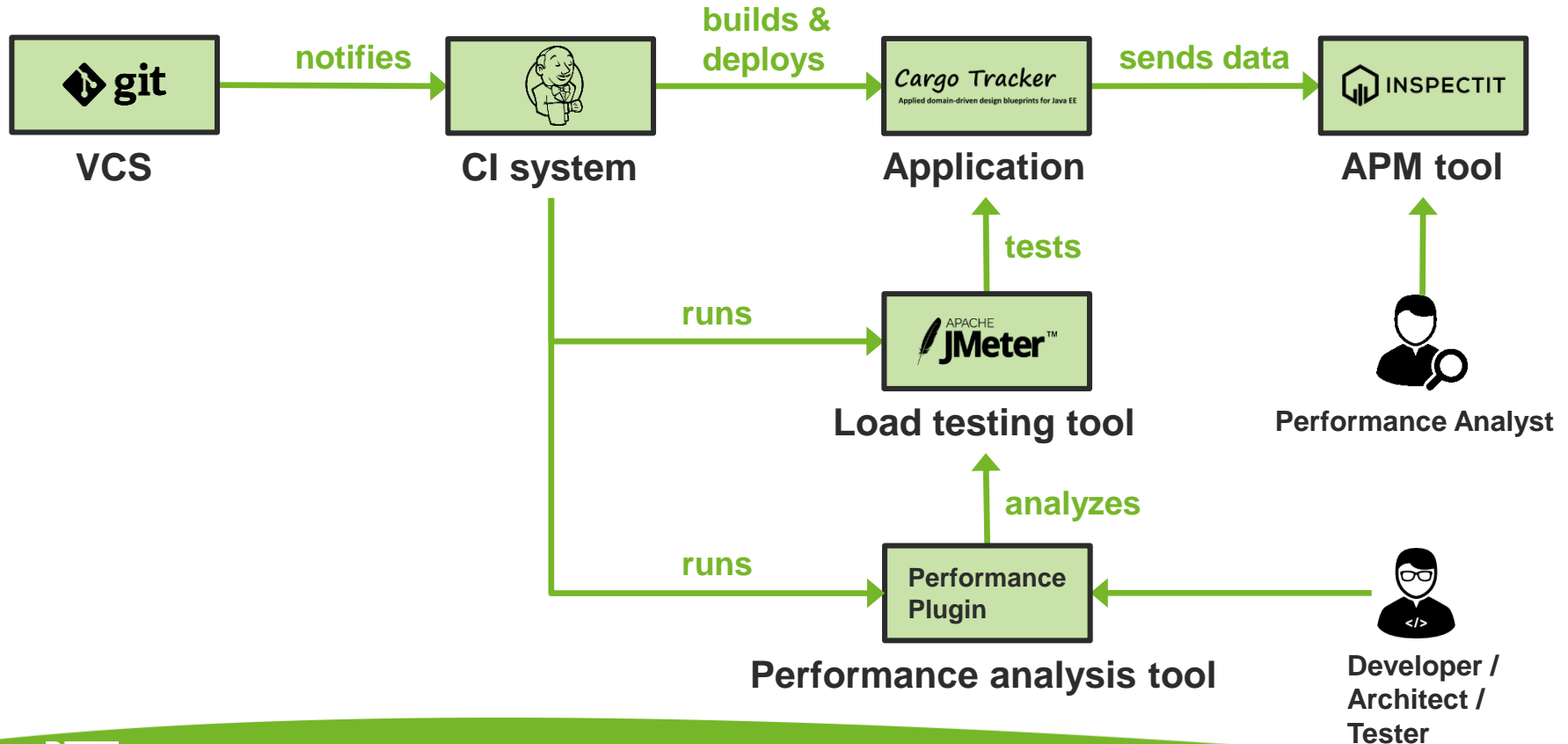
Open Source alternatives:



Closed Source alternatives:



Alternatives: APM Tools



Demo

<https://github.com/RETIT/continuous-performance-evaluation>

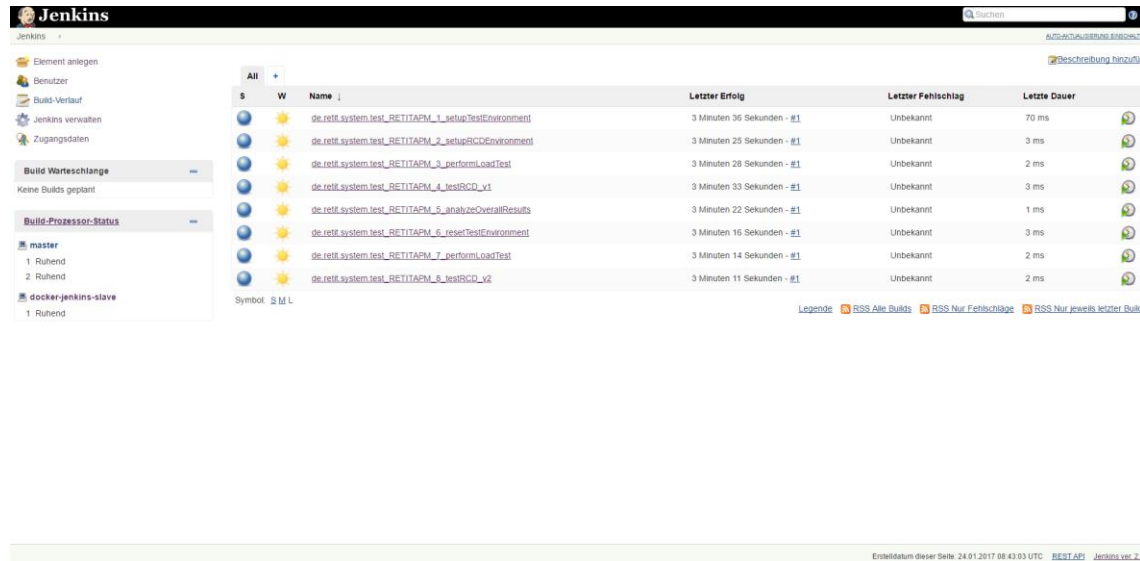


Tip #1: Use Pipelines

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
 - Avoid manual configuration
 - Treat pipeline as code

Tip #1: Use Pipelines

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
 - ➔ Avoid manual configuration
 - ➔ Treat pipeline as code



The screenshot shows the Jenkins web interface. On the left, there is a sidebar with navigation links: 'Element anlegen', 'Benutzer', 'Builds-Verlauf', 'Jenkins verwalten', and 'Zugangsdaten'. Below these are sections for 'Build Warteschlange' (no builds planned) and 'Build-Processor-Status' (showing 'master' with 1 idle and 2 running, and 'docker-jenkins-slave' with 1 idle). The main area displays a table of builds with columns: 's', 'W', 'Name', 'Letzter Erfolg', 'Letzter Fehlschlag', and 'Letzte Dauer'. The table lists 8 builds, all of which are successful. The build names are prefixed with 'de.retit.system.test_RETITAPM_'. At the bottom right, there is a legend for RSS feeds: 'RSS Alle Builds', 'RSS Nur Fehlschläge', and 'RSS Nur jeweils letzter Build'.

s	W	Name	Letzter Erfolg	Letzter Fehlschlag	Letzte Dauer
●	☀	de.retit.system.test_RETITAPM_1_setupTestEnvironment	3 Minuten 36 Sekunden - #1	Unbekannt	70 ms
●	☀	de.retit.system.test_RETITAPM_2_setupRCDEnvironment	3 Minuten 25 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retit.system.test_RETITAPM_3_performLoadTest	3 Minuten 28 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retit.system.test_RETITAPM_4_testRCDD_v1	3 Minuten 33 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retit.system.test_RETITAPM_5_analyzeOverallResults	3 Minuten 22 Sekunden - #1	Unbekannt	1 ms
●	☀	de.retit.system.test_RETITAPM_6_resetTestEnvironment	3 Minuten 16 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retit.system.test_RETITAPM_7_performLoadTest	3 Minuten 14 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retit.system.test_RETITAPM_8_testRCDD_v2	3 Minuten 11 Sekunden - #1	Unbekannt	2 ms

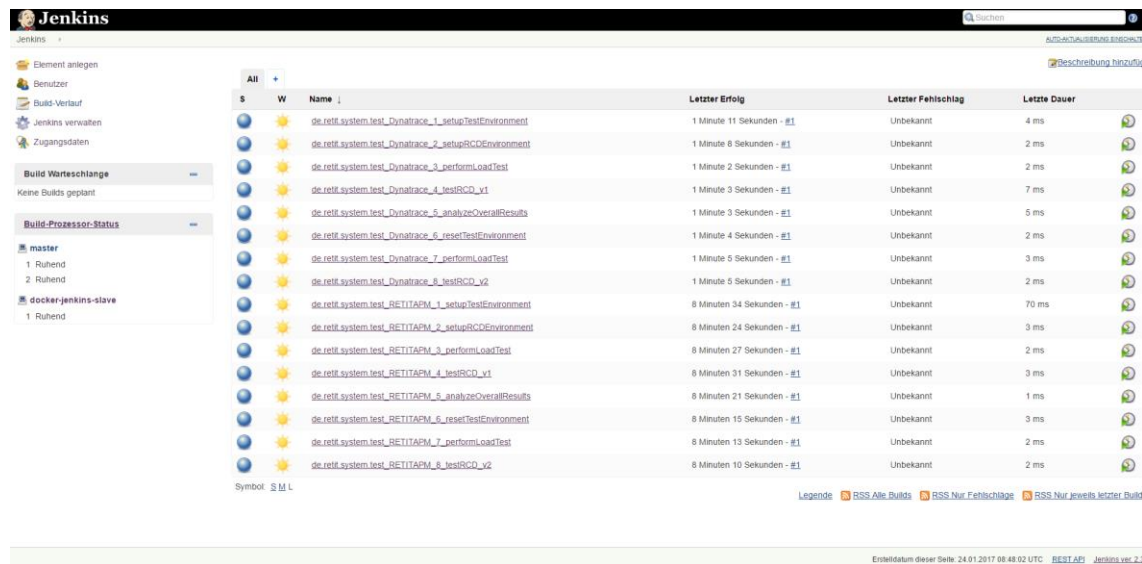
Symbol: S M L

Legende: RSS Alle Builds RSS Nur Fehlschläge RSS Nur jeweils letzter Build

Erstelltatum dieser Seite: 24.01.2017 08:43:03 UTC [BESTAF](#) [Jenkins ver. 2.3](#)

Tip #1: Use Pipelines

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
 - ➔ Avoid manual configuration
 - ➔ Treat pipeline as code



The screenshot shows the Jenkins web interface. On the left, there is a sidebar with navigation links: 'Element anlegen', 'Benutzer', 'Builds-Verlauf', 'Jenkins verwalten', and 'Zugangsdaten'. Below these are sections for 'Build Warteschlange' (no builds planned) and 'Build-Processor-Status' (showing 'master' with 2 idle nodes and 'docker-jenkins-slave' with 1 idle node). The main area displays a table of pipeline builds. The table has columns for status (S), build status (W), name, last success, last failure, and last duration. The builds are for a pipeline named 'de.retiti.system.test_Dynatrace'. The table shows 14 builds, all with a status of 'Success' (green sun icon) and a last success time of 'Unbekannt'. The last duration for each build is listed in milliseconds.

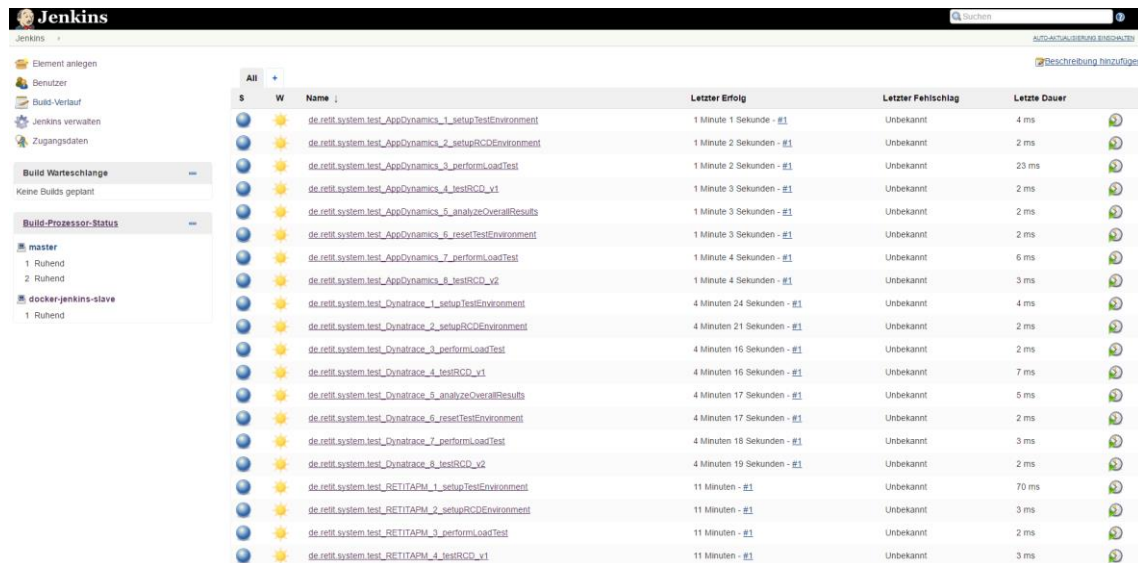
S	W	Name	Letzter Erfolg	Letzter Fehlschlag	Letzte Dauer
●	☀	de.retiti.system.test_Dynatrace_1_setupTestEnvironment	1 Minute 11 Sekunden - #1	Unbekannt	4 ms
●	☀	de.retiti.system.test_Dynatrace_2_setupRCDEnvironment	1 Minute 8 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retiti.system.test_Dynatrace_3_performLoadTest	1 Minute 2 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retiti.system.test_Dynatrace_4_testRCDD_v1	1 Minute 3 Sekunden - #1	Unbekannt	7 ms
●	☀	de.retiti.system.test_Dynatrace_5_analyzeOverallResults	1 Minute 3 Sekunden - #1	Unbekannt	5 ms
●	☀	de.retiti.system.test_Dynatrace_6_resetTestEnvironment	1 Minute 4 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retiti.system.test_Dynatrace_7_performLoadTest	1 Minute 5 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retiti.system.test_Dynatrace_8_testRCDD_v2	1 Minute 5 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retiti.system.test_RETITAFM_1_setupTestEnvironment	8 Minuten 34 Sekunden - #1	Unbekannt	70 ms
●	☀	de.retiti.system.test_RETITAFM_2_setupRCDEnvironment	8 Minuten 24 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retiti.system.test_RETITAFM_3_performLoadTest	8 Minuten 27 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retiti.system.test_RETITAFM_4_testRCDD_v1	8 Minuten 31 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retiti.system.test_RETITAFM_5_analyzeOverallResults	8 Minuten 21 Sekunden - #1	Unbekannt	1 ms
●	☀	de.retiti.system.test_RETITAFM_6_resetTestEnvironment	8 Minuten 15 Sekunden - #1	Unbekannt	3 ms
●	☀	de.retiti.system.test_RETITAFM_7_performLoadTest	8 Minuten 13 Sekunden - #1	Unbekannt	2 ms
●	☀	de.retiti.system.test_RETITAFM_8_testRCDD_v2	8 Minuten 10 Sekunden - #1	Unbekannt	2 ms

Legende: Alle Builds Nur Fehlschläge Nur jeweils letzter Build

Erstellt am: 24.01.2017 08:48:02 UTC [BEST AF](#) Jenkins ver. 2.3

Tip #1: Use Pipelines

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
 - ➔ Avoid manual configuration
 - ➔ Treat pipeline as code

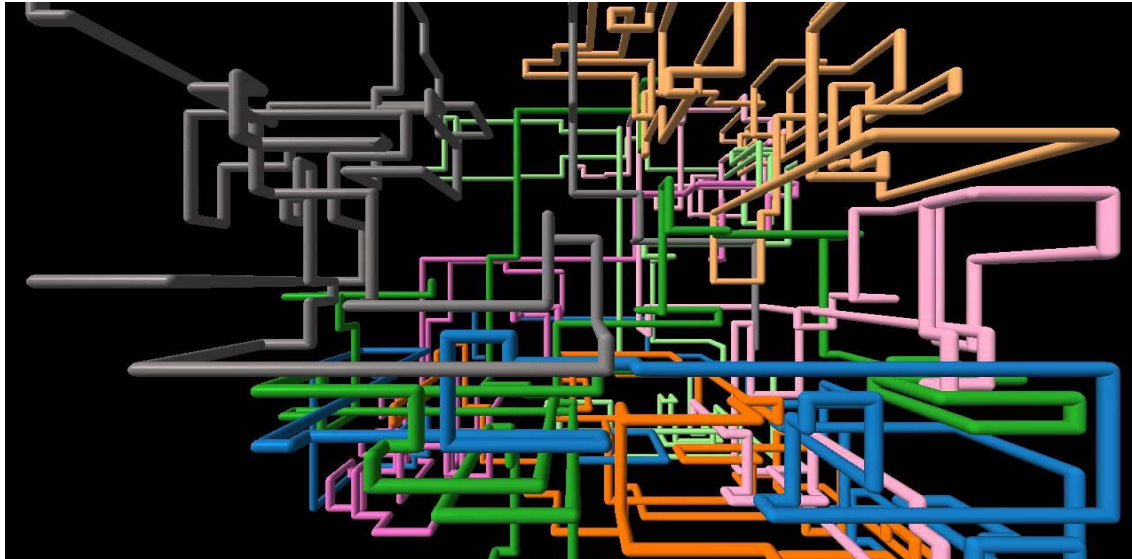


The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with navigation links like 'Element anlegen', 'Benutzer', 'Build-Verlauf', 'Jenkins verwalten', and 'Zugangsdaten'. Below these are sections for 'Build Warteschlange' (no builds planned) and 'Build-Processor-Status' (showing 'master' and 'dockerjenkins-slave' as idle). The main area displays a table of build jobs.

S	W	Name	Letzter Erfolg	Letzter Fehlschlag	Letzte Dauer
		de.retit.system.test_AppDynamics_1_setupTestEnvironment	1 Minute 1 Sekunde - #1	Unbekannt	4 ms
		de.retit.system.test_AppDynamics_2_setupRCDEnvironment	1 Minute 2 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_AppDynamics_3_performLoadTest	1 Minute 2 Sekunden - #1	Unbekannt	23 ms
		de.retit.system.test_AppDynamics_4_testRCD_v1	1 Minute 3 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_AppDynamics_5_analyzeOverallResults	1 Minute 3 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_AppDynamics_6_resetTestEnvironment	1 Minute 3 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_AppDynamics_7_performLoadTest	1 Minute 4 Sekunden - #1	Unbekannt	6 ms
		de.retit.system.test_AppDynamics_8_testRCD_v2	1 Minute 4 Sekunden - #1	Unbekannt	3 ms
		de.retit.system.test_DynaTrace_1_setupTestEnvironment	4 Minuten 24 Sekunden - #1	Unbekannt	4 ms
		de.retit.system.test_DynaTrace_2_setupRCDEnvironment	4 Minuten 21 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_DynaTrace_3_performLoadTest	4 Minuten 16 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_DynaTrace_4_testRCD_v1	4 Minuten 16 Sekunden - #1	Unbekannt	7 ms
		de.retit.system.test_DynaTrace_5_analyzeOverallResults	4 Minuten 17 Sekunden - #1	Unbekannt	5 ms
		de.retit.system.test_DynaTrace_6_resetTestEnvironment	4 Minuten 17 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_DynaTrace_7_performLoadTest	4 Minuten 18 Sekunden - #1	Unbekannt	3 ms
		de.retit.system.test_DynaTrace_8_testRCD_v2	4 Minuten 19 Sekunden - #1	Unbekannt	2 ms
		de.retit.system.test_RETITAPM_1_setupTestEnvironment	11 Minuten - #1	Unbekannt	70 ms
		de.retit.system.test_RETITAPM_2_setupRCDEnvironment	11 Minuten - #1	Unbekannt	3 ms
		de.retit.system.test_RETITAPM_3_performLoadTest	11 Minuten - #1	Unbekannt	2 ms
		de.retit.system.test_RETITAPM_4_testRCD_v1	11 Minuten - #1	Unbekannt	3 ms

Tip #1: Use Pipelines

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
 - Avoid manual configuration
 - Treat pipeline as code



Tip #1: Use Pipelines

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
 - Avoid manual configuration
 - Treat pipeline as code

Pipeline Performance_Test_Job



[Last Successful Artifacts](#)



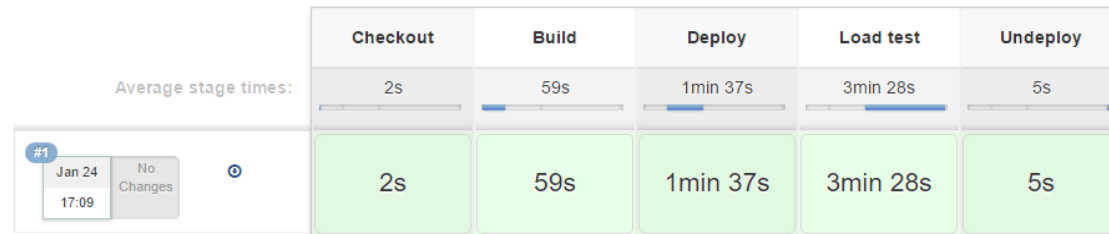
[dashBoard_cargotracker-jmeter.xml](#)

274 B [view](#)



[Recent Changes](#)

Stage View



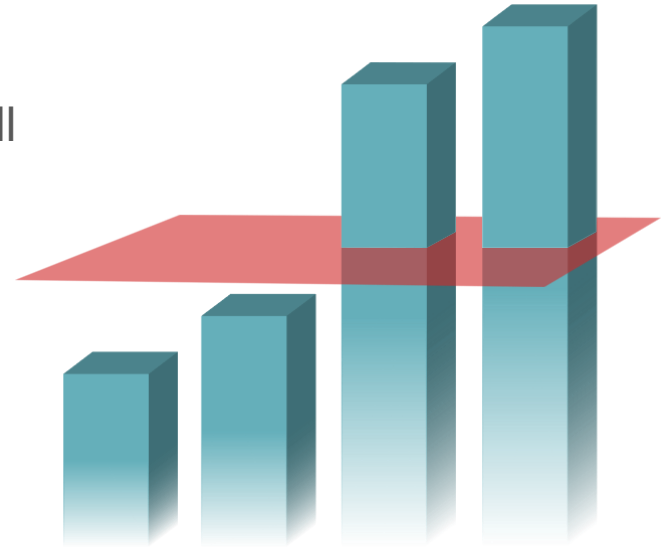
Tip #2: Use Load Tests correctly

- Load tests only yield meaningful results **if** used correctly
- When designing your tests, make sure that:
 - Your workload resembles how users would access the application
 - You achieve steady state performance (test runs long enough?)
 - Your test environment is comparable to your production environment
 - If not, be aware that relative comparisons are the best you can achieve
 - Your test environment has no external influences (e.g., from other load test environments)



Tip #3: Find good Thresholds

- Thresholds are a good way of notifying developers about performance problems
- However, good thresholds are hard to find
 - If the threshold is too low, developers will start ignoring the warnings
 - If the threshold is too high, potential problems might go unnoticed
- Each application and load test is different
 - Finding good thresholds takes time
 - Consider using absolute thresholds
 - You might also need to remove ramp-up using custom scripts



Tip #4: Run the Pipeline often

- Ideally, you would run your performance tests after every commit
- If you can't do that, minimum is once a day
- Try to minimize the pipeline run time so you can run it more often



Tip #5: Optimize your Pipeline

- Minimize data which needs to be processed sent or stored
 - Use JMeter's CSV output instead of XML
 - Don't store every Jenkins build, introduce log rotation
 - Use partial database resets (e.g., only delete data generated during the last test execution instead of full restores)
- Use caching effectively
 - Maven's local repository
 - Docker's image cache

More Performance? Visit our meetup!

<http://www.meetup.com/de-DE/Software-Performance-Meetup-Group/>



The screenshot shows the Meetup.com page for the 'Software Performance Meetup' group. The header features the group name and a navigation bar with links like 'Startseite', 'Mitglieder', 'Sponsoren', 'Fotos', 'Seiten', 'Diskussionen', 'Mehr', 'Gruppenverwaltung', and 'Mein Profil'. The main content area is divided into three columns. The left column contains a bar chart graphic, a 'Foto bearbeiten' button, location information 'München, Deutschland' (founded May 21, 2014), an 'Über uns...' button, a 'Freunde einladen' button, and member statistics 'Performance Management Worker' (758). The middle column has a 'Herzlich willkommen!' message, a '+ Neuen Meetup Termin festlegen' button, tabs for 'Bald (1)', 'Vergangene', and 'Kalender', a 'Featured Meetup' for '11th Software Performance Meetup' (Burda Bootcamp, May 9, 2017, 19:00), a row of member avatars, and a welcome message from the group. The right column shows 'Was gibt es Neues' with three photos of past events, a 'MEHR' link, and a 'NEUES MITGLIED' section featuring 'Kevin macht mit' (joined 3 hours ago).

Software Performance Meetup

Startseite Mitglieder Sponsoren Fotos Seiten Diskussionen Mehr Gruppenverwaltung Mein Profil

Herzlich willkommen!

+ Neuen Meetup Termin festlegen

Bald (1) Vergangene Kalender

Featured Meetup

11th Software Performance Meetup

Burda Bootcamp

Arabellastr 23, München (Karte)

Di, 9. Mai 19:00

Ich nehme teil

49 nehmen teil 0 Kommentare

Dear Performance Management Workers, we are happy to announce that our next meetup will take place on Tuesday, May 9th 2017 and will be hosted by the Burda Bootcamp! We have now confirmed two very interesting talks for our next meetup: Richard Vobl (RETIT) will talk about HTTP/2 performance improvements and Christian Vögele (Novatec) will...

München, Deutschland
Gegründet 21. Mai 2014

Über uns...

Freunde einladen

Performance Management Worker 758

Was gibt es Neues

MEHR

NEUES MITGLIED
Kevin macht mit
vor 3 Stunden

Thank you!

Dr. Andreas Brunnert
brunnert@retit.de



Resource Efficient Technologies & IT Systems