

# Performance Testing in Continuous Delivery Pipelines



**Andrey Pokhilko**  
Chief Scientist, BlazeMeter



# Importance of CI and CD

- Machine time costs nothing, human time priceless
- De-facto winning practice
- Most advanced teams go with CD
- A lot of teams are still in process of adopting it



# Agenda


1. Things we put into CI
2. Best practice (for now)
3. Challenges of testing in CI
4. Jenkins for Performance Testing

**Things we put into CI**

# Triggering and Preparations

- When to do the job
- VCS checkout + dependencies checkout
- Building project (compiling etc.)
- Put resulting packages into repos

# Quality Control

1. Static code analysis
  2. Unit tests
  3. Functional tests
  4. Performance tests
- 
- needs deployment

# Deployment

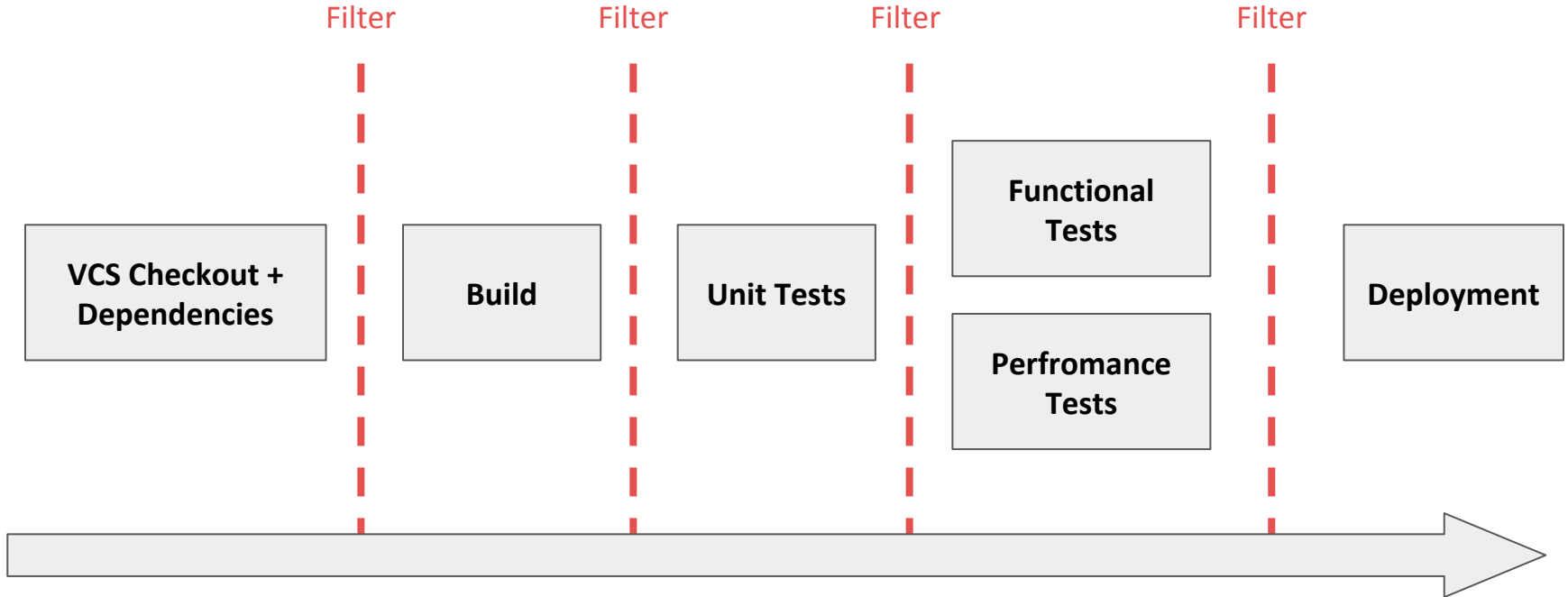
- To QA environment (for further manual tests)
- To staging environment
- To production environment



# **The Best Practice**



# One Way Road (Driverless)



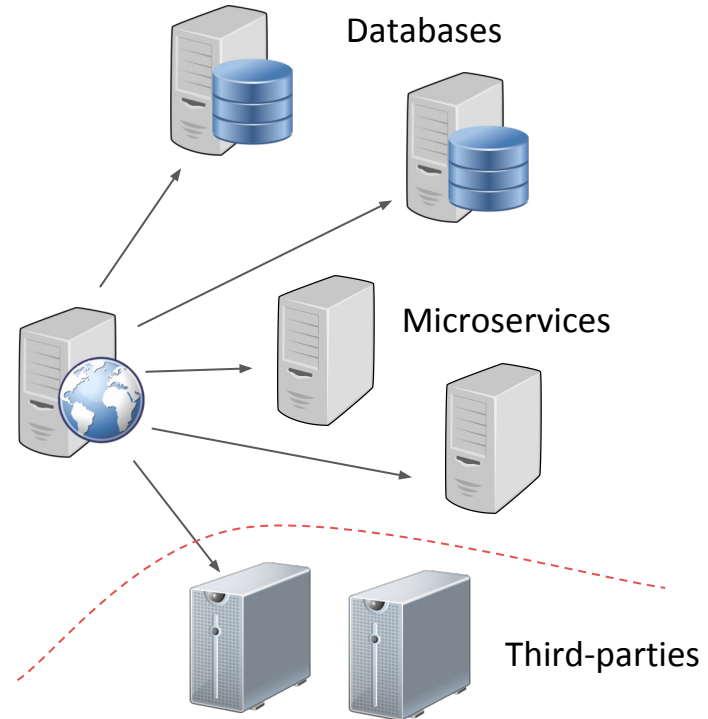
# VCS-Driven Pipeline

- Natural evolution of CI systems
- Branching and pull requests
- Jenkins 2.0 pipelines
- Taurus Tool as part of this approach

# Why testing is challenging in CI

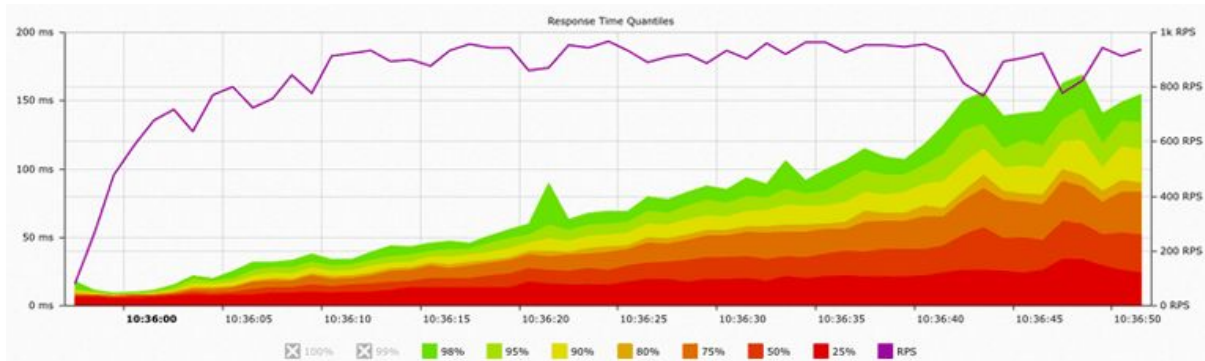
# Challenge #1: Test Environment

- Applications are complex
- Lots of dependencies
- Third-party systems



## Challenge #2: Time Consuming

- Preparations
- A lot of functional tests
- Performance tests are naturally long



# Challenge #3: Debugging CI Jobs

- Evolving test complexity
- Debugging and troubleshooting
- Build history is a value

 <a href="#"><u>#32</u></a>	01.09.2016 16:15
 <a href="#"><u>#31</u></a>	23.08.2016 20:05
 <a href="#"><u>#30</u></a>	18.08.2016 12:25
 <a href="#"><u>#29</u></a>	18.08.2016 11:56
 <a href="#"><u>#28</u></a>	08.08.2016 17:05
 <a href="#"><u>#27</u></a>	07.08.2016 11:35
 <a href="#"><u>#26</u></a>	12.07.2016 12:05
 <a href="#"><u>#25</u></a>	05.07.2016 22:40
 <a href="#"><u>#24</u></a>	05.07.2016 22:25
 <a href="#"><u>#23</u></a>	17.06.2016 16:25
 <a href="#"><u>#22</u></a>	08.06.2016 14:35
 <a href="#"><u>#21</u></a>	01.06.2016 15:55
 <a href="#"><u>#20</u></a>	31.05.2016 17:35
 <a href="#"><u>#19</u></a>	04.05.2016 13:35
 <a href="#"><u>#18</u></a>	25.04.2016 14:35
 <a href="#"><u>#17</u></a>	14.04.2016 15:15
 <a href="#"><u>#16</u></a>	11.04.2016 16:35

## Challenge #4: Results Analysis

- Rich reporting needed
- Jenkins UI has its limits
- Automated decision making for CD
- Collaboration for non-tech people

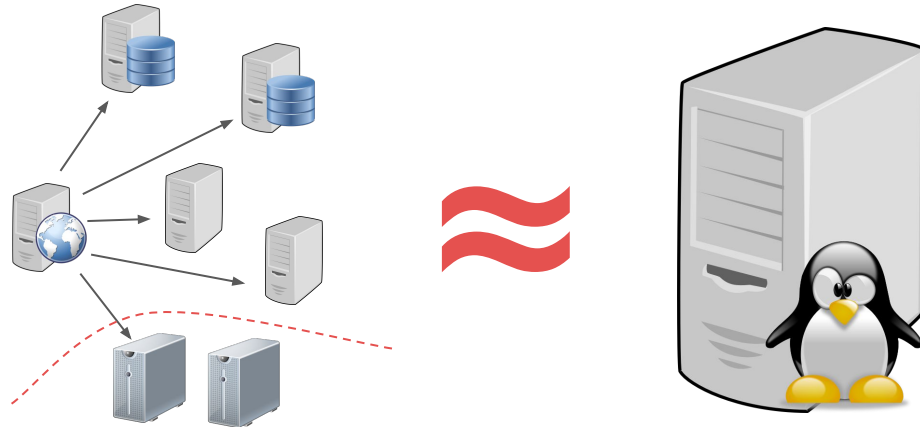


# Overcoming Challenges



# Overcome: Test Environment

- Containers help (you have to use containers :)
- You don't need to be as realistic

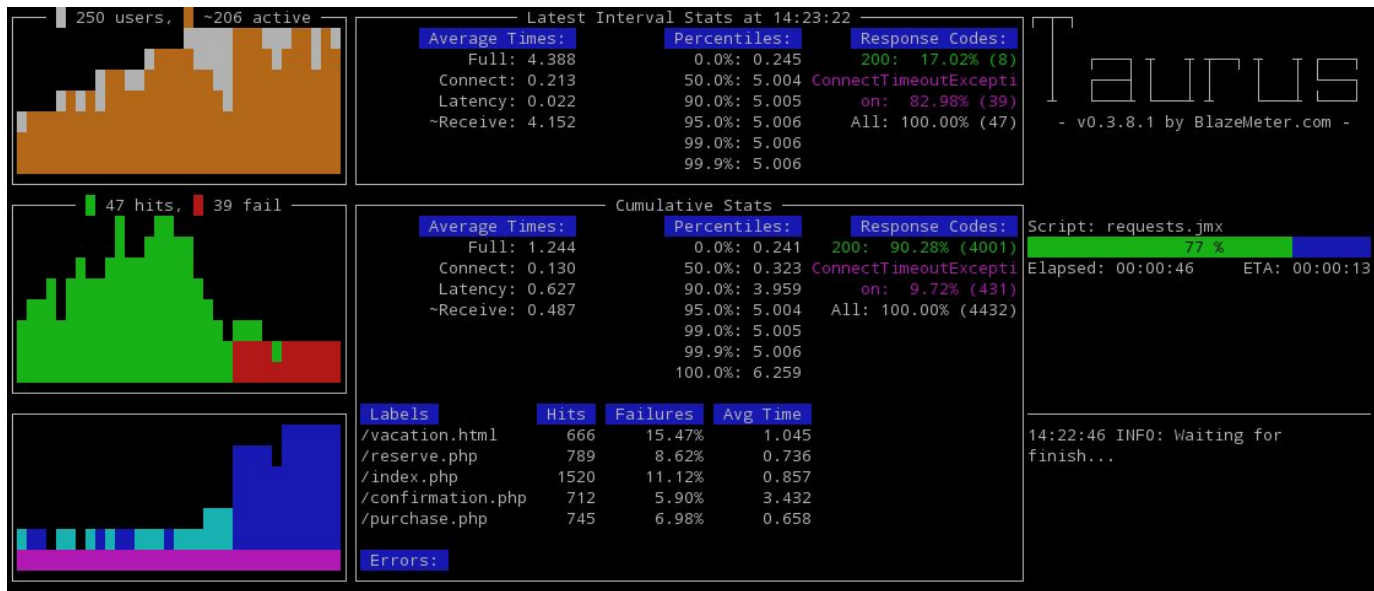


## Overcome: Time Consuming

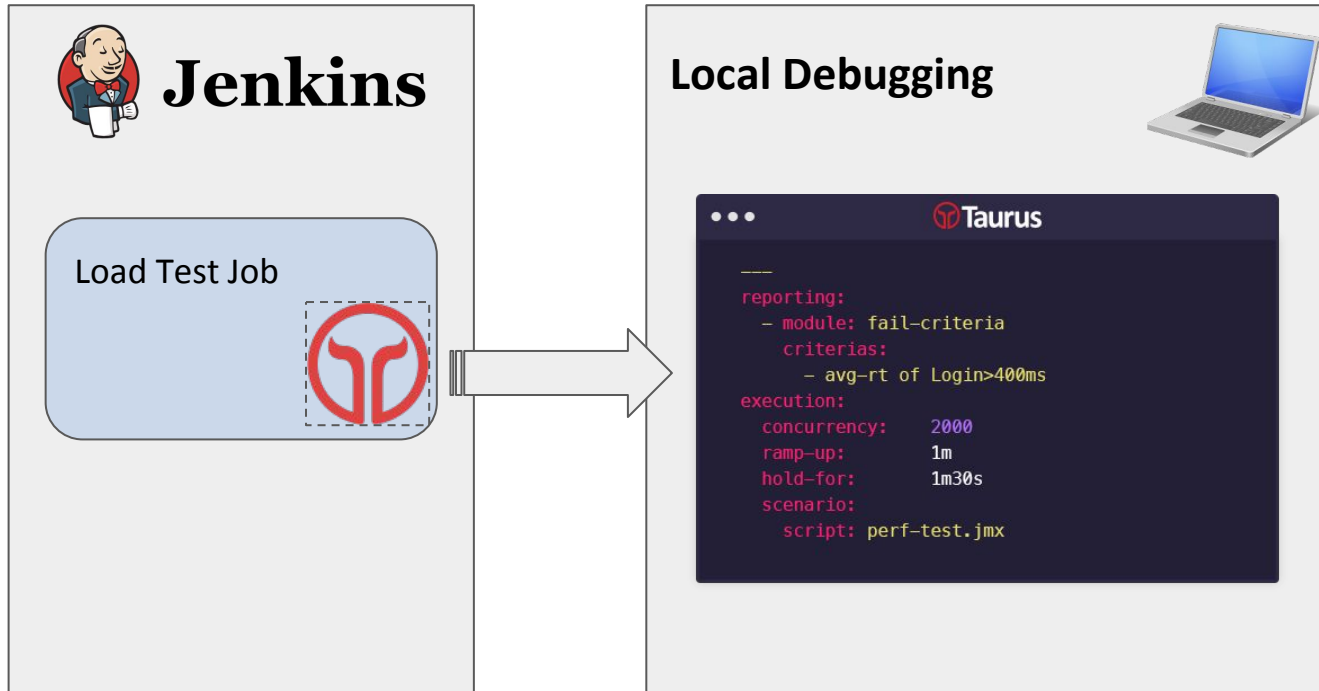
- Prepare what you can upfront - in the night
- Reuse what you can
- Short tests can reveal a lot of things
- Don't try to put spike and endurance tests into CI
- Parallelize tests, Jenkins 2.0 or Taurus helps

# Overcome: Debugging and Improving

- Taurus Tool, the handheld piece of CI



# Overcome: Debugging and Improving



# Overcome: Reporting & Result Consumption

- Purpose-built services with integrations
- Overview & Status in Jenkins



# Overcome: Decision Making

- Deploy to staging is great win
- Have pass/fail criteria, including APM info
- It's ok to have deploy as separate job  
(for UAT, to avoid disruptions)

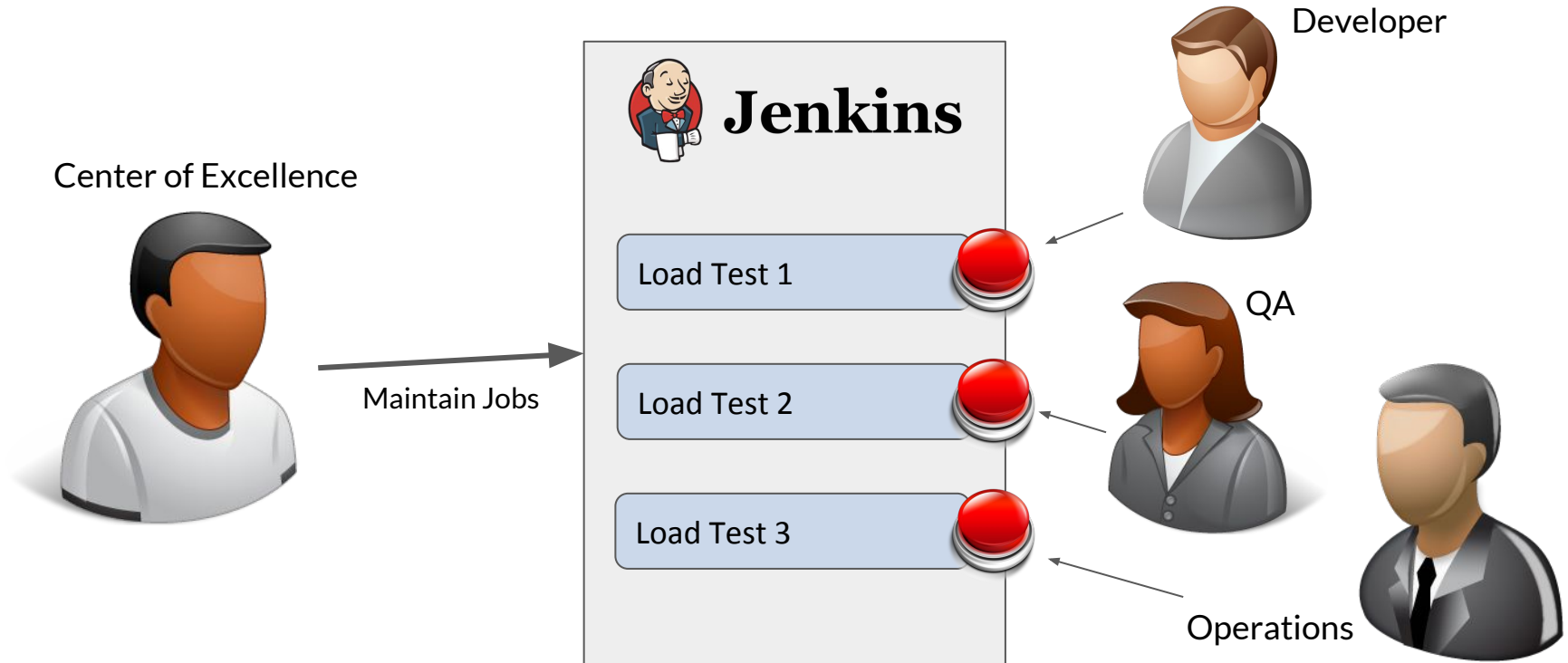
# Jenkins for Performance Testing

# Why Use Jenkins for Testing?

- Multi-step process
- Long process
- Repetitive process



# Start Simple

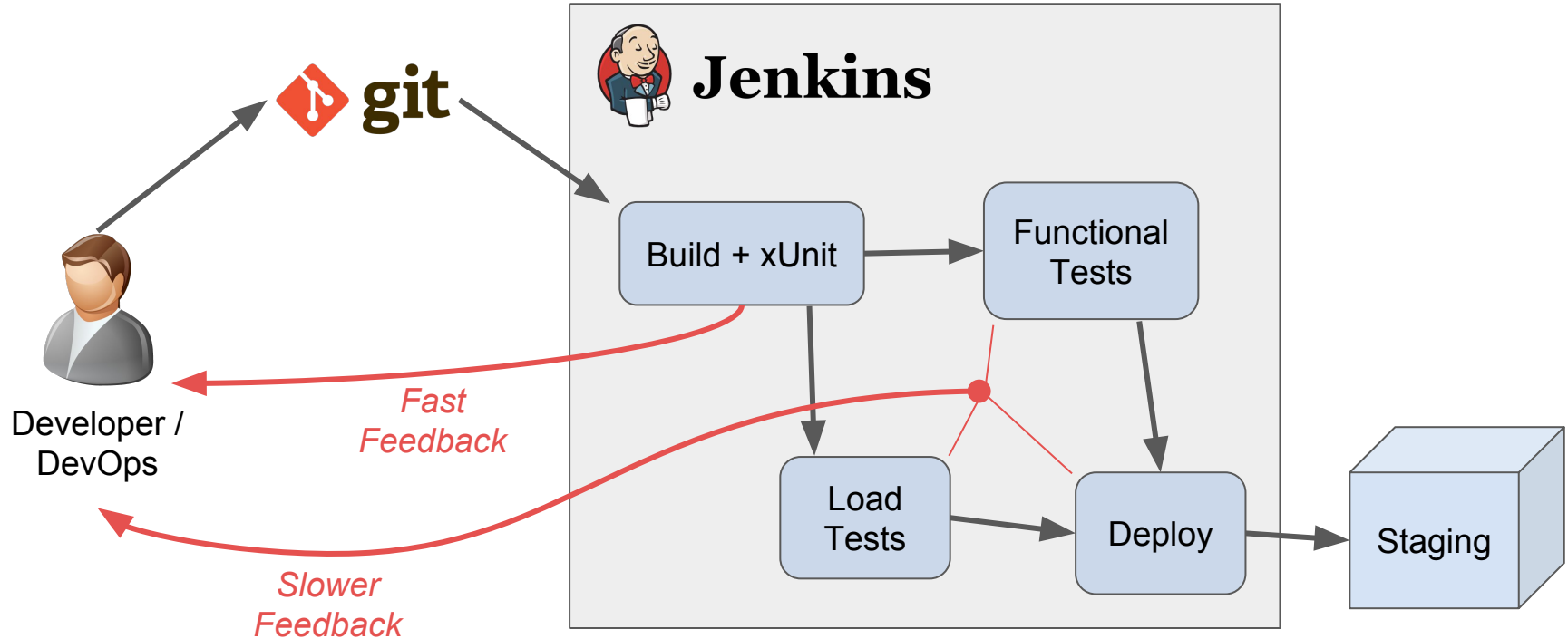


# Become Continuous

- Periodic jobs have many applications
- Have valid failure criteria (thresholds)
- Use non-blocking downstream jobs

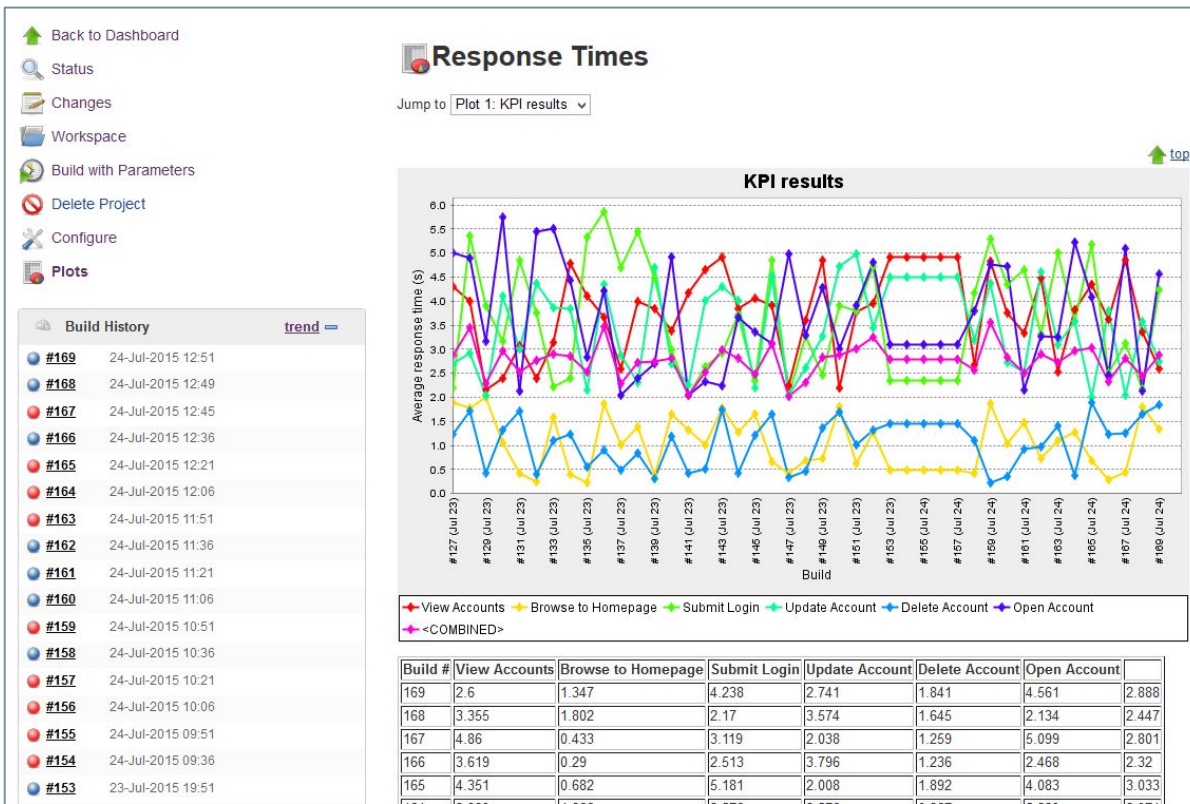


# Pursue Continuous Delivery

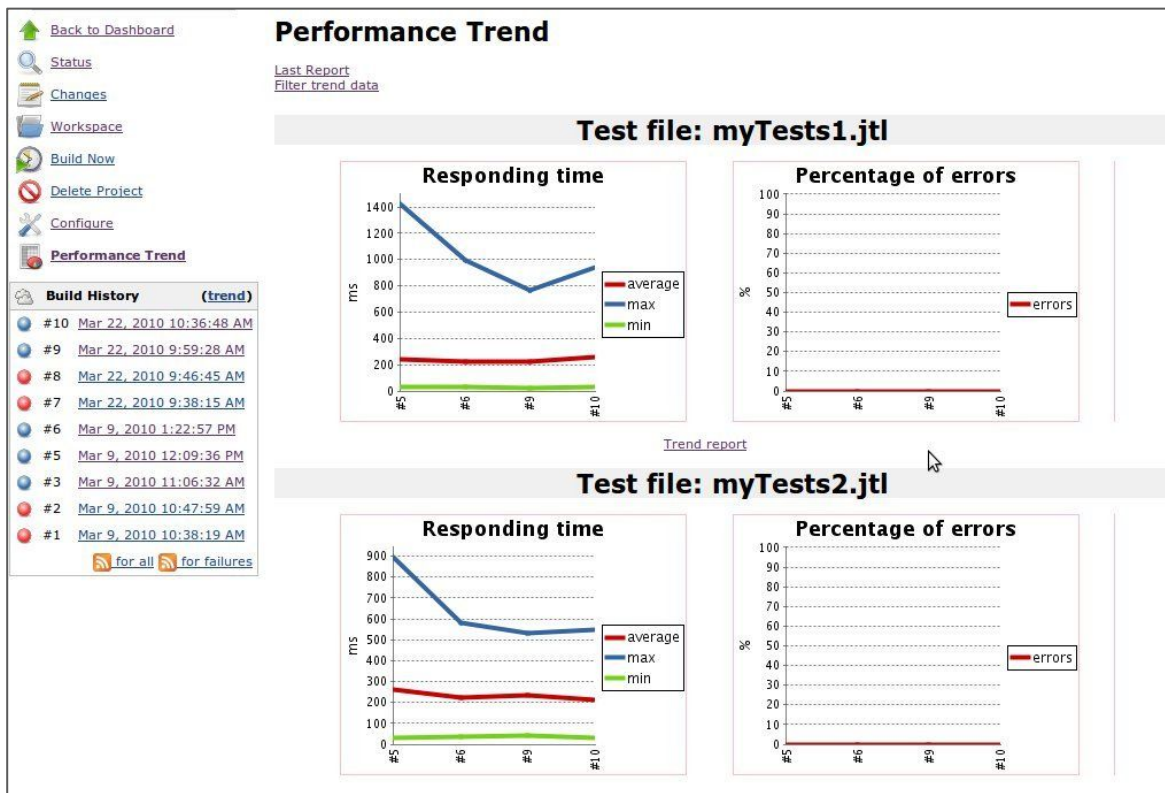


## Some Useful Plugins

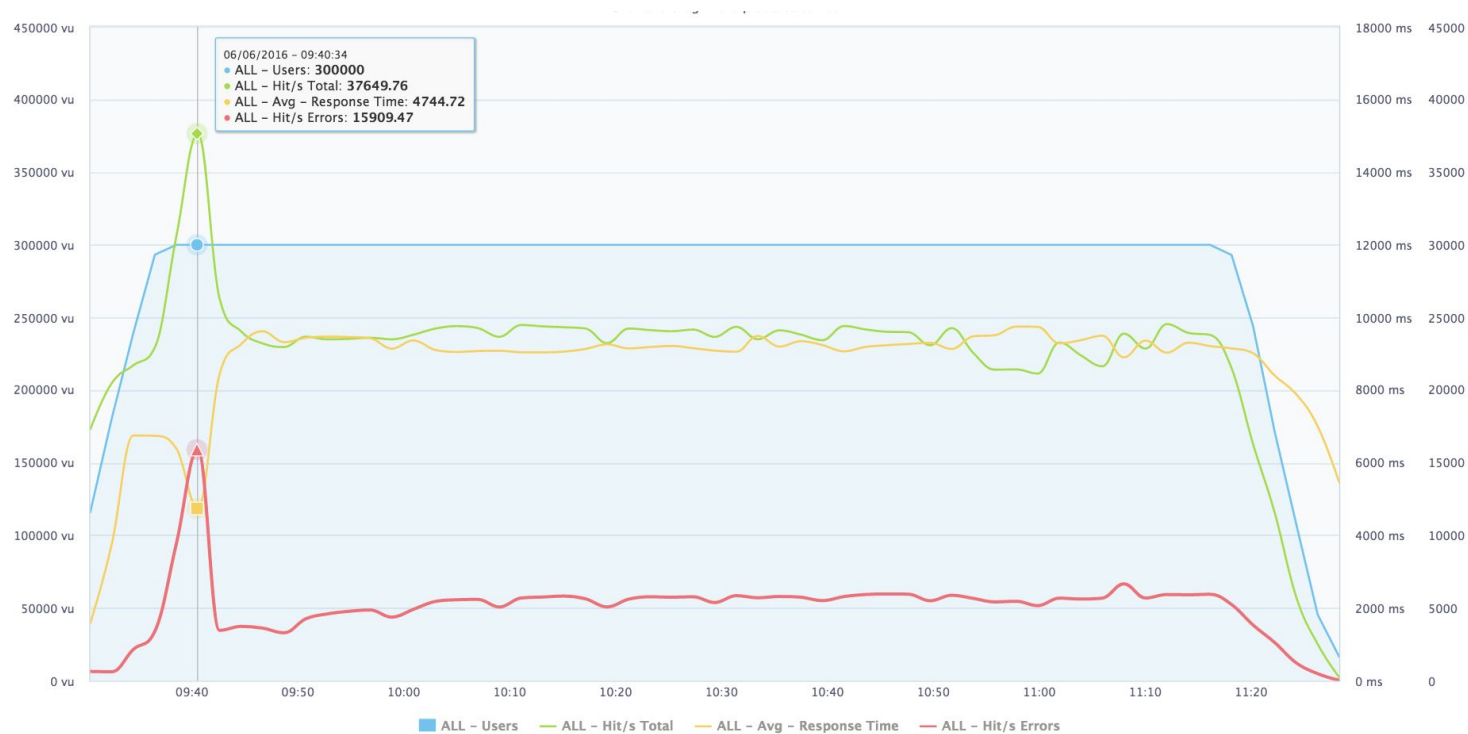
# Plot Plugin



# Performance Publisher Plugin



# BlazeMeter Plugin



# Taurus Tool

- Why not plugin, but command-line tool
- Universal plug for tests inside Jenkins
- Taurus designed to work with other Jenkins plugins







# Final Summary

1. We have to test inside CI
2. There are ways to start it simple
3. There are tools & plugins to help
4. Jenkins 2.0 pipelines **FTW!**

# Thank you!

Any questions?

