Linux, Docker & GitLab CI

Łukasz Hryniuk

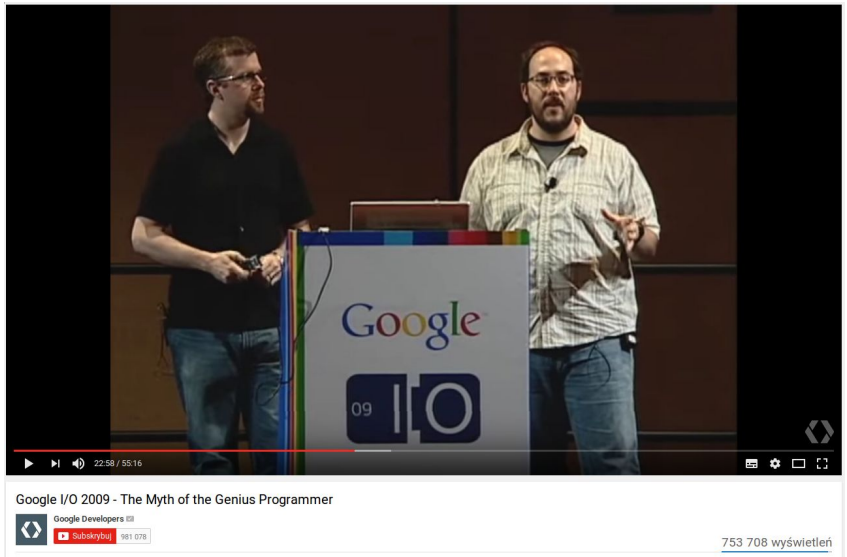
RST
SOFTWARE MASTERS

Agenda

1. Bit about trust.
2. Overview of our project.
3. Tools: fabric, ccache.
4. Jenkins.
5. GitLabCI configuration.
6. GitLabCI++.
7. Q&A.

RST
SOFTWARE MASTERS

Trust vs ~~control~~ constraints



The Myth of the Genius Programmer

One Hacker Way

RST
SOFTWARE MASTERS

Server

Problems we had

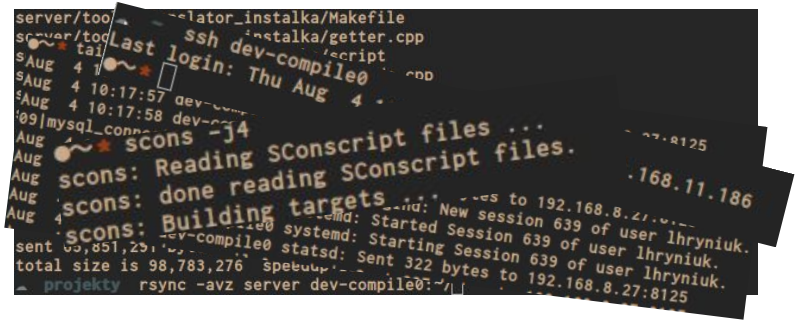
- manual building and testing
- manual environment configuration
- manual deploy
- one DEV environment

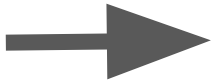
Step-by-step solution

1. FreeBSD + Jenkins + shell
2. FreeBSD/Linux + Jenkins/GitLabCI + shell
3. FreeBSD/Linux + Jenkins/GitLabCI + Puppet
4. Linux + GitLabCI + Puppet (**today**!, 25.10)

RST
SOFTWARE MASTERS

Fabric - execute remote tasks automatically





```
Note:
fab branch:develop sync build cpbin deploy syslog:grepfor='lhryniuk'
fab branch:develop sync build deploy syslog:grepfor='lhryniuk'
fab cpbin
fab deploy
fab names
fab sync
fab sync build
fab sync build:master_ut
fab sync build:master_ut test:master_ut
fab sync build deploy syslog
fab syslog
fab syslog:grepfor='lhryniuk'
fab syslog:lhryniuk
ssh-add ~/.ssh/id_rsa
```

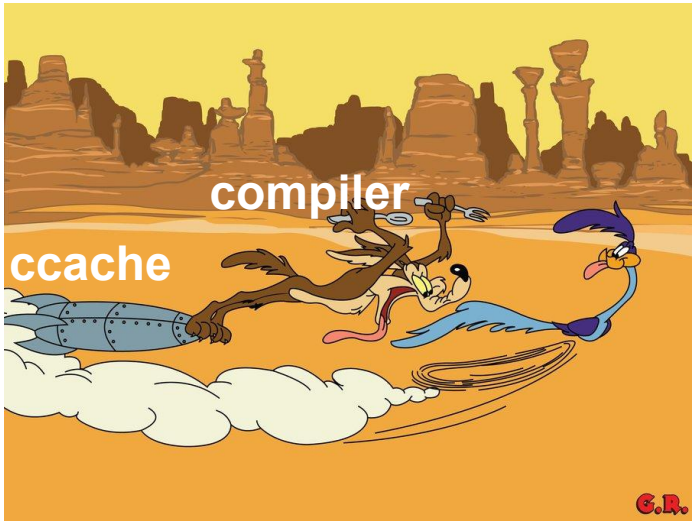
RST
SOFTWARE MASTERS

Compilation

RST
SOFTWARE MASTERS

ccache

- compiler cache for C family of languages
- keep object files between compilations
  - even if you run ‘clean’ compilation
- speed up dependent on number of changes
  - but for unchanged files stored obj’s are used



RST
SOFTWARE MASTERS

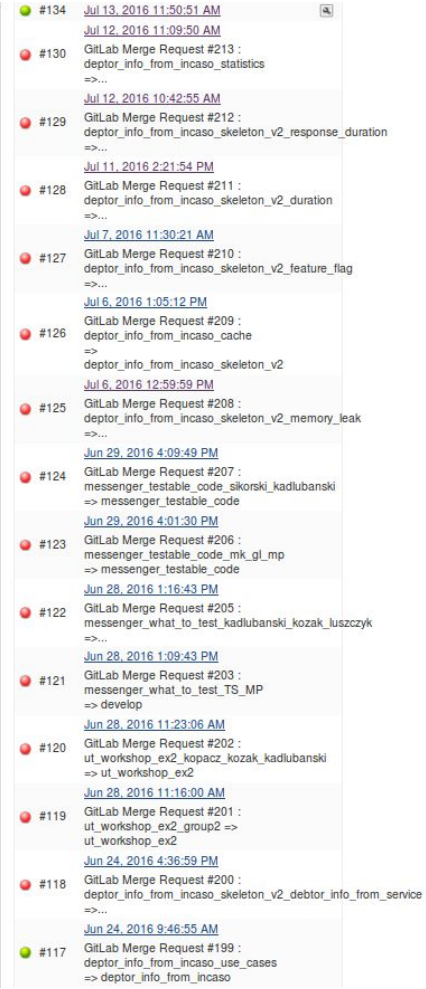
ccache

compilation times			
project	without ccache	with ccache	speed up
server	11m15.851s	0m45.086s	~1500%
master_ut	01m27.746s	0m08.707s	~1000%
segment_ut	10m09.432s	0m33.681s	~1800%

RST
SOFTWARE MASTERS

So where is this CI?

RST
SOFTWARE MASTERS



Jenkins...

- 24.06 - 12.07 (**18 days**), 13 builds, **NO successes**
- compilation started only when MR is opened

RST
SOFTWARE MASTERS

...resurrected!

Fixes

- **set correct version of dependencies**
- new configuration for GitLab build triggers

Results

- 13.07 - 01.08 (**18 days**) 70 builds, **63 successes**

RST
SOFTWARE MASTERS

Jenkins :(

- difficult to configure - everything requires a plugin
- UT coverage report only in XML
- only one job running at any point
- binaries are hard to find

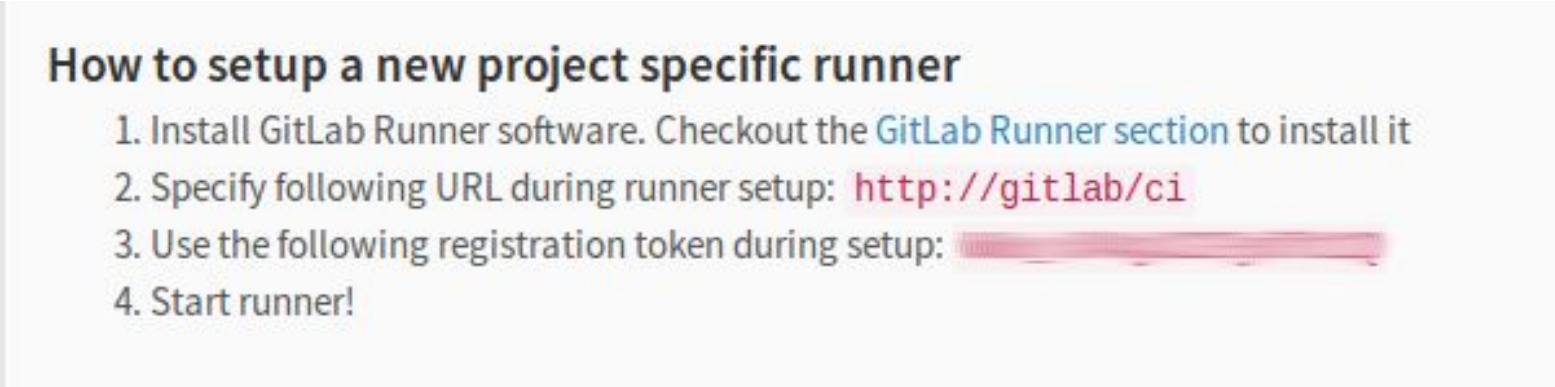
New sprint, Linux, “accidental” root access

- server must be migrated from FreeBSD to Linux
- all tests and regressions must run on new platform
- better compilation result notifications were needed (**the reason for checking GitLab CI**)
- new host to test server in Linux env (and subsequent **root access**)

RST
SOFTWARE MASTERS

GitLab CI - installation

- `curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-ci-multi-runner/script.rpm.sh | sudo bash`
- `sudo yum install gitlab-ci-multi-runner`
- `sudo gitlab-ci-multi-runner register`



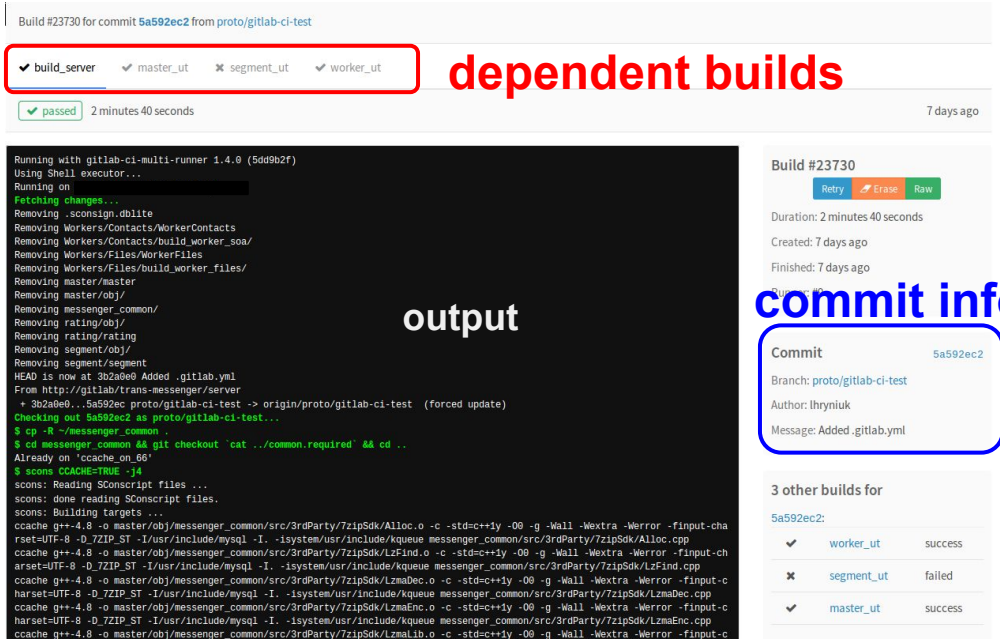
RST SOFTWARE MASTERS

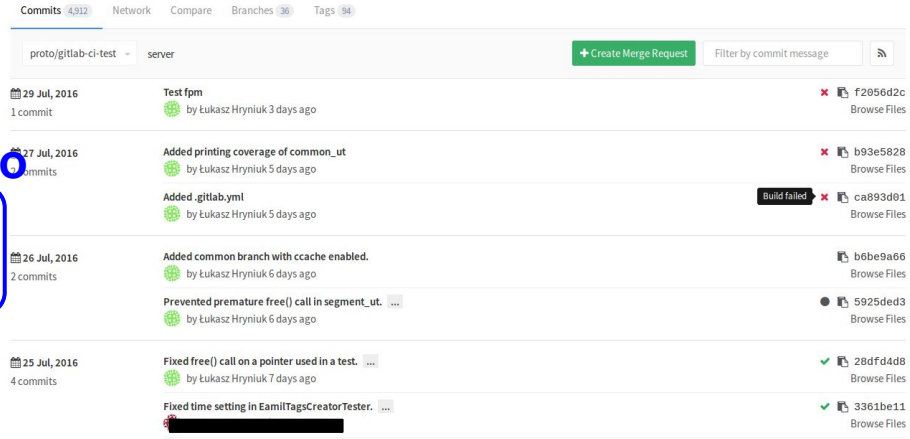
...and configuration: **.gitlab-ci.yml**

```
segment_ut:
  stage: test
  script:
    - scons TEST_COVERAGE=TRUE CCACHE=TRUE -j4 segment_ut
    - ./tests/Segment/segment_ut --gtest_shuffle --gtest_output='xml:segment_ut.junit.xml'
    - chmod a+x ./tools/gcovr/gcovr
    - ./tools/gcovr/gcovr -p -r . -e tests -e messenger_common -e common --html -o segment_ut_cov.html
    - ./tools/gcovr/gcovr -p -r . -e tests -e messenger_common -e common
  artifacts:
    paths:
      - segment_ut_cov.html
      - segment_ut.junit.xml
    name: "${CI_BUILD_NAME}_${CI_BUILD_REF}"
    expire_in: 1 week
```

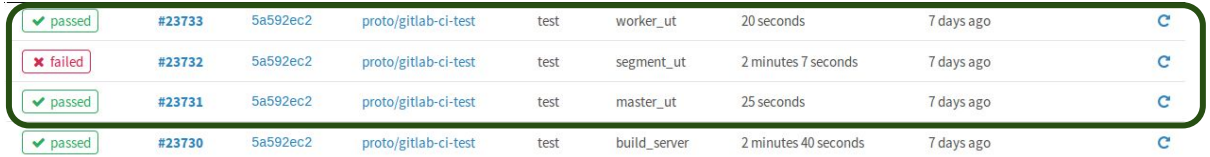
RST
SOFTWARE MASTERS

First build





tests



RST
SOFTWARE MASTERS

Select executor

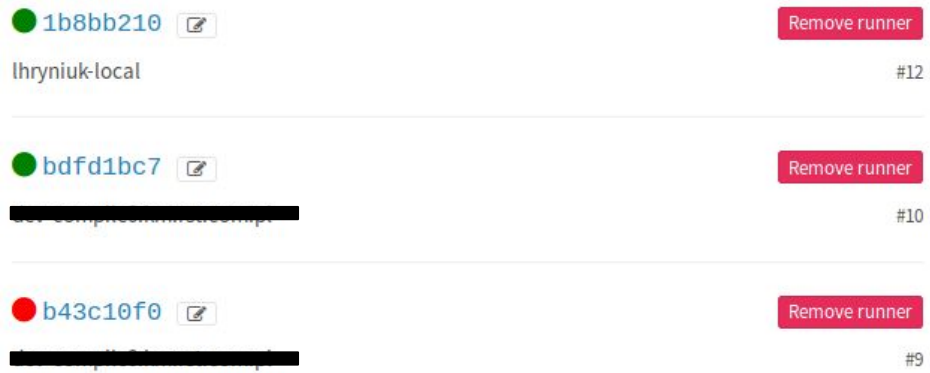
- Help me select executor
- Shell
- Docker and Docker-SSH
- Parallels
- VirtualBox
- SSH

RST
SOFTWARE MASTERS

Dockerfile and more gitlab-runners

```
FROM centos:latest

RUN yum groupinstall -y "Development Tools" && \
    yum -y install scons \
        libstdc++-static \
        libkqueue-devel \
        mariadb-devel \
        libicu-devel \
        libmemcached-devel \
        boost-devel \
        python-devel \
        ccache \
        which \
        ruby-devel && gem install fpm
```



RST
SOFTWARE MASTERS

```
^TOTAL.*\s+(\d+\%)$
```
+ a bit of code =

passed	#24934	a74ff413	proto/gitlab-ci-test	test	worker_ut	49 seconds	4 days ago	71.0%	
failed	#24933	a74ff413	proto/gitlab-ci-test	test	segment_ut	3 minutes 13 seconds	4 days ago	19.0%	
passed	#24932	a74ff413	proto/gitlab-ci-test	test	master_ut	56 seconds	4 days ago	13.0%	
passed	#24922	a74ff413	proto/gitlab-ci-test	build	worker-files	42 seconds	4 days ago		
passed	#24921	a74ff413	proto/gitlab-ci-test	build	worker-contactlist	1 minute 9 seconds	4 days ago		
passed	#24920	a74ff413	proto/gitlab-ci-test	build	segment	3 minutes 7 seconds	4 days ago		
passed	#24919	a74ff413	proto/gitlab-ci-test	build	rating	37 seconds	4 days ago		
passed	#24918	a74ff413	proto/gitlab-ci-test	build	master	51 seconds	4 days ago		

coverage!

artifacts!

GitLab Messenger · GitLab · 11:42
[redacted] pushed to branch task/blocking_socket_function_test of trans-messenger/segment (Compare changes)
- made test more bdd style (7ef1c2)

GitLab Messenger · GitLab · 11:44
[redacted] pushed to branch task/blocking_socket_function_test of trans-messenger/segment (Compare changes)
- Made test more bdd style. (b1af00)
- Removed redundant space. (85212a)

GitLab Messenger · GitLab · 11:48
[redacted] pushed new branch task/performance-test-login to trans-messenger/segment

GitLab Messenger · GitLab · 11:49
[redacted] removed branch **task/performance-test-login** from trans-messenger/segment

GitLab Messenger · GitLab · 11:52
trans-messenger/segment: Commit 7ef1c206 of task/blocking_socket_function_test branch[redacted] in 596 second(s)

GitLab Messenger · GitLab · 11:56
trans-messenger/segment: Commit 85212a9d of task/blocking_socket_function_test branch[redacted] in 696 second(s)

GitLab Messenger · GitLab · 11:58
trans-messenger/segment: Commit 79fde76d of task/performance-test-login branch[redacted] in 623 second(s)

HipChat notifications

RST SOFTWARE MASTERS

What we had after first stage?

- several builds concurrently, inside Docker container
- build on push
- binaries are built separately
- tests are built and run only after success of the previous stage
- code coverage next to build status
- RPMs generation
- test artifacts in XML and HTML
- notifications about commits and builds
- **we can build binaries on Windows and run it on Linux**

RST
SOFTWARE MASTERS

Second stage

- Docker image in our own registry
- configs pull - Puppet
- RPM auto-push to the repo
- local DEV environment: Vagrant/Docker

More: Beamer slides generation and docs ;-)

Status	Build ID	Commit	Ref	Stage	Name	Duration	Finished at	
passed	#39749	e0d197ee	master	test	pdf	8 seconds	11 days ago	
failed	#39748	725aafd9	master	test	pdf	3 seconds	11 days ago	

Name	Size
presentation.pdf	11 KB

RST
SOFTWARE MASTERS

What we want to have

- SonarQube integration (test results/coverage/static code analysis)
- Coding Standards checks (git hooks maybe - **too much control(?)**)
- ...and autonotification about new package ready to deploy
- auto deploy and functional testing
- changelog generation
- more Docker (Docker swarm?)

RST
SOFTWARE MASTERS

Questions?

RST
SOFTWARE MASTERS

Thanks!

RST
SOFTWARE MASTERS