🚑 Selenium Tests: WTF Even? 🚧

A technical talk // from michael

🚑 Selenium Tests: WTF Even? 🚧

- ~~who even~~
- what even
- when even
- where even
- why even
- (w)how even

I can't even

- what even is Selenium?
- when stuff is broken, how even do I find out?
- where even does it run?
- why even do we have these tests?
- how even does it all work?

🚧

> The safe upper limit for selenium is 400 micrograms a day in adults. Anything above that is considered an overdose.
>
> – *WebMD*

what even is Selenium?

what even: Selenium itself

- Selenium is a browser automation tool
- used for automated testing
- simulates a real user clicking around and doing stuff

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

what even: Selenium itself

- an extremely powerful tool
- an extremely fiddly tool

what even: Selenium at Fastmail

how even: a history

- we used to have a full-time QA engineer
- started doing manual testing
- but then got into automated testing with Selenium

how even: a history

- Chris set up all the infrastructure, frameworks, etc.
- and consequently, it's all a bit weird
- but mostly, just kinda works!

how even: a history

- 06/2018: Topicbox tests rewritten from scratch
- 09/2018: Fastmail tests rewritten from scratch
- 12/2018: Chris leaves Fastmail
- 12/2018–09/2020: Michael tends the Selenium garden
- 10/2020–??: everyone tends the Selenium garden

where even

where even: the map

- testnet: [git host]/fastmail/testnet
- Jenkins: [internal url]
- #qa-department
- @trob

where even: the tests

- selenium tests run against three environments:
  - beta
  - qa
  - www
- ask trob in slack: "test $env"

where even: infrastructure

- ¯_(ツ)_/¯
- mostly, in Docker containers running on a bladecenter
- Marc knows *slightly* more about this than I do

[internal network architecture diagram redacted]

when even: test runs

- on demand, via trob
- via cron

how even: "@trob test QA"

- trob kicks off a Jenkins job
- which you can see at [internal url]
- which pulls latest testnet code
- and runs the tests against the specified environment
- and reports the results into #qa-department
  (new failures also to #fastmail)

why even have these tests?

why even: *these* tests?

- Cassandane (Cyrus)
- cunit (Cyrus)
- newt (Perl)
- JS tests (Squire)
- Swift/Kotlin tests (apps)

why even: Selenium tests

- end-to-end, real-world testing
- helps notice breakages before our users do
- the *only* tests we run regularly of new Fastmail-flavo[u]red Cyrus

why even: questions and answers

- Q: Aren't Selenium tests flaky?
- A: No.*

why even: questions and answers

- Q: Aren't Selenium tests flaky?
- A: No.*

Selenium itself can be fragile, but the tests are not.
Failures always indicate real breakages (but sometimes, the breakage is trivial).

why even: questions and answers

- Q: Do our Selenium tests provide full coverage of our UI?
- A: No.

why even: questions and answers

- Q: Do our Selenium tests provide full coverage of our UI?
- A: No.

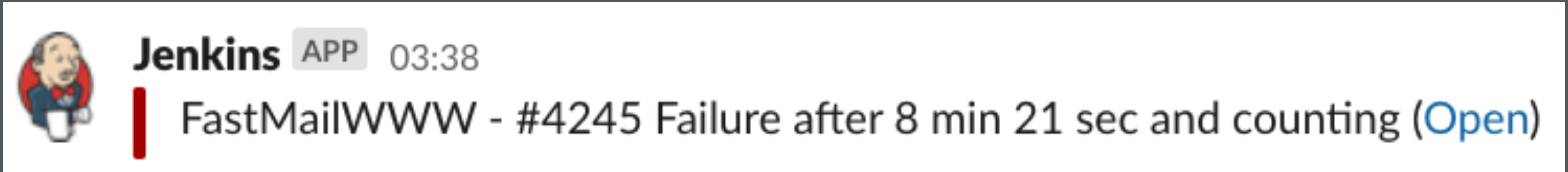
There's always room for more work.
But they are a useful last resort: "is this change I want to deploy to production totally broken?"

why even: questions and answers

- Q: You really couldn't think of another question to put here?
- A: Nope.

how even: diagnosing problems

how even: diagnosing problems



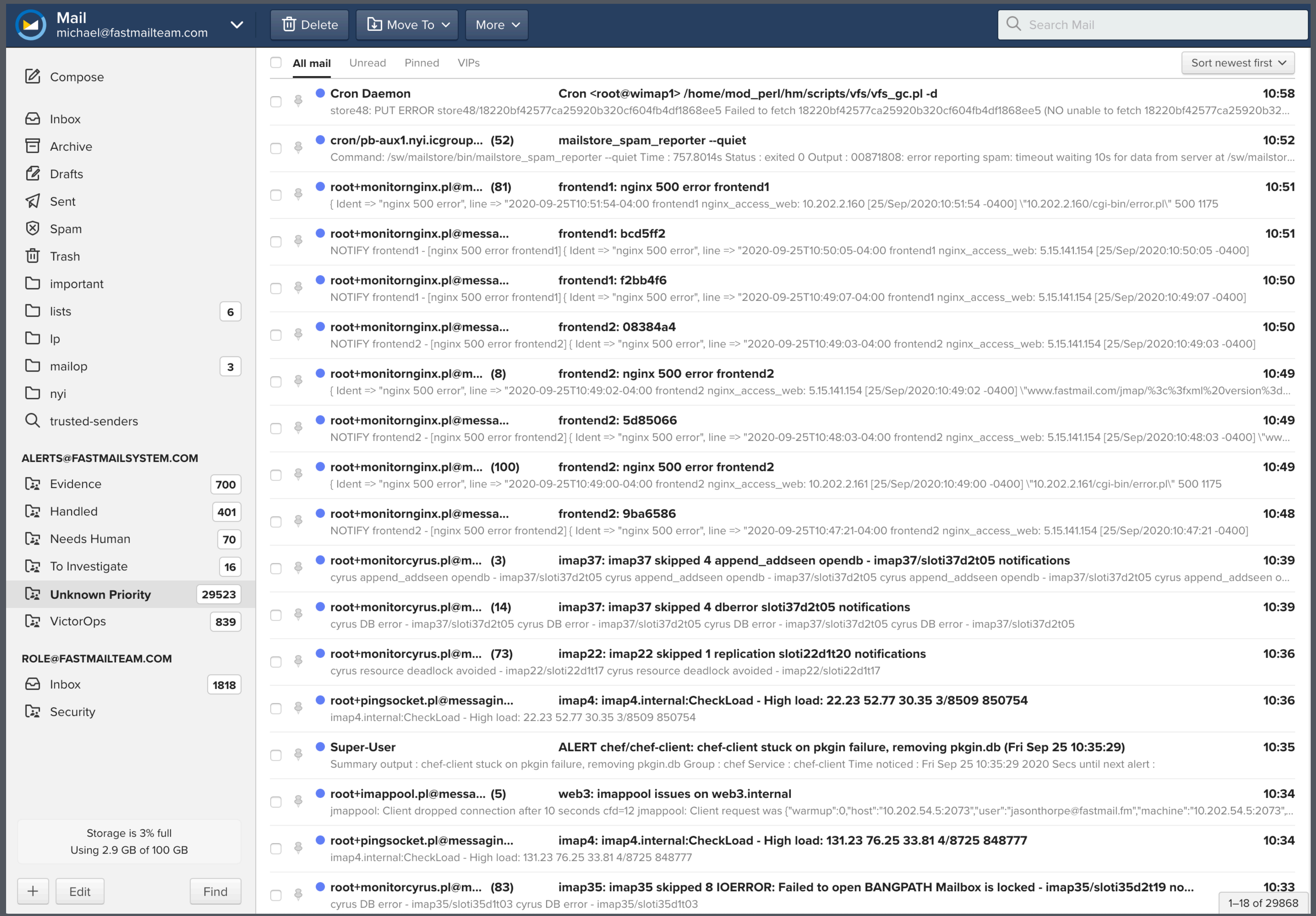
Jenkins APP 03:38

FastMailWWW - #4245 Failure after 8 min 21 sec and counting (Open)

how even: diagnosing problems

1. find the build in Jenkins
2. look for the test logs
3. find the failure
4. read the test files

what even: the Jenkins UI



Cron Daemon	Cron <root@wimap1> /home/mod_perl/hm/scripts/vfs/vfs_gc.pl -d	10:58	store48: PUT ERROR store48/18220bf42577ca25920b320cf604fb4df1868ee5 Failed to fetch 18220bf42577ca25920b320cf604fb4df1868ee5 (NO unable to fetch 18220bf42577ca25920b32...
cron/pb-aux1.nyi.icgroup... (52)	mailstore_spam_reporter --quiet	10:52	Command: /sw/mailstore/bin/mailstore_spam_reporter --quiet Time : 757.8014s Status : exited 0 Output : 00871808: error reporting spam: timeout waiting 10s for data from server at /sw/mailstor...
root+monitornginx.pl@m... (81)	frontend1: nginx 500 error frontend1	10:51	{ Ident => "nginx 500 error", line => "2020-09-25T10:51:54-04:00 frontend1 nginx_access_web: 10.202.2.160 [25/Sep/2020:10:51:54 -0400] \"10.202.2.160/cgi-bin/error.pl\" 500 1175
root+monitornginx.pl@messa...	frontend1: bcd5ff2	10:51	NOTIFY frontend1 - [nginx 500 error frontend1] { Ident => "nginx 500 error", line => "2020-09-25T10:50:05-04:00 frontend1 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:50:05 -0400]
root+monitornginx.pl@messa...	frontend1: f2bb4f6	10:50	NOTIFY frontend1 - [nginx 500 error frontend1] { Ident => "nginx 500 error", line => "2020-09-25T10:49:07-04:00 frontend1 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:49:07 -0400]
root+monitornginx.pl@messa...	frontend2: 08384a4	10:50	NOTIFY frontend2 - [nginx 500 error frontend2] { Ident => "nginx 500 error", line => "2020-09-25T10:49:03-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:49:03 -0400]
root+monitornginx.pl@m... (8)	frontend2: nginx 500 error frontend2	10:49	{ Ident => "nginx 500 error", line => "2020-09-25T10:49:02-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:49:02 -0400] \"www.fastmail.com/jmap/%3c%3fxml%20version%3d...
root+monitornginx.pl@messa...	frontend2: 5d85066	10:49	NOTIFY frontend2 - [nginx 500 error frontend2] { Ident => "nginx 500 error", line => "2020-09-25T10:48:03-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:48:03 -0400] \"ww...
root+monitornginx.pl@m... (100)	frontend2: nginx 500 error frontend2	10:49	{ Ident => "nginx 500 error", line => "2020-09-25T10:49:00-04:00 frontend2 nginx_access_web: 10.202.2.161 [25/Sep/2020:10:49:00 -0400] \"10.202.2.161/cgi-bin/error.pl\" 500 1175
root+monitornginx.pl@messa...	frontend2: 9ba6586	10:48	NOTIFY frontend2 - [nginx 500 error frontend2] { Ident => "nginx 500 error", line => "2020-09-25T10:47:21-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:47:21 -0400]
root+monitorcyrus.pl@m... (3)	imap37: imap37 skipped 4 append_addseen opendb - imap37/sloti37d2t05 notifications	10:39	cyrus append_addseen opendb - imap37/sloti37d2t05 cyrus append_addseen opendb - imap37/sloti37d2t05 cyrus append_addseen opendb - imap37/sloti37d2t05 cyrus append_addseen o...
root+monitorcyrus.pl@m... (14)	imap37: imap37 skipped 4 dberror sloti37d2t05 notifications	10:39	cyrus DB error - imap37/sloti37d2t05 cyrus DB error - imap37/sloti37d2t05 cyrus DB error - imap37/sloti37d2t05 cyrus DB error - imap37/sloti37d2t05
root+monitorcyrus.pl@m... (73)	imap22: imap22 skipped 1 replication sloti22d1t20 notifications	10:36	cyrus resource deadlock avoided - imap22/sloti22d1t17 cyrus resource deadlock avoided - imap22/sloti22d1t17
root+pingsocket.pl@messagin...	imap4: imap4.internal:CheckLoad - High load: 22.23 52.77 30.35 3/8509 850754	10:36	imap4.internal:CheckLoad - High load: 22.23 52.77 30.35 3/8509 850754
Super-User	ALERT chef/chef-client: chef-client stuck on pkgin failure, removing pkgin.db (Fri Sep 25 10:35:29)	10:35	Summary output : chef-client stuck on pkgin failure, removing pkgin.db Group : chef Service : chef-client Time noticed : Fri Sep 25 10:35:29 2020 Secs until next alert :
root+imappool.pl@messa... (5)	web3: imappool issues on web3.internal	10:34	jmappool: Client dropped connection after 10 seconds cfd=12 jmappool: Client request was {"warmup":0,"host":"10.202.54.5:2073","user":"jasonthorpe@fastmail.fm","machine":"10.202.54.5:2073",...
root+pingsocket.pl@messagin...	imap4: imap4.internal:CheckLoad - High load: 131.23 76.25 33.81 4/8725 848777	10:34	imap4.internal:CheckLoad - High load: 131.23 76.25 33.81 4/8725 848777
root+monitorcyrus.pl@m... (83)	imap35: imap35 skipped 8 IOERROR: Failed to open BANGPATH Mailbox is locked - imap35/sloti35d2t19 no...	10:33	cyrus DB error - imap35/sloti35d1t03 cyrus DB error - imap35/sloti35d1t03

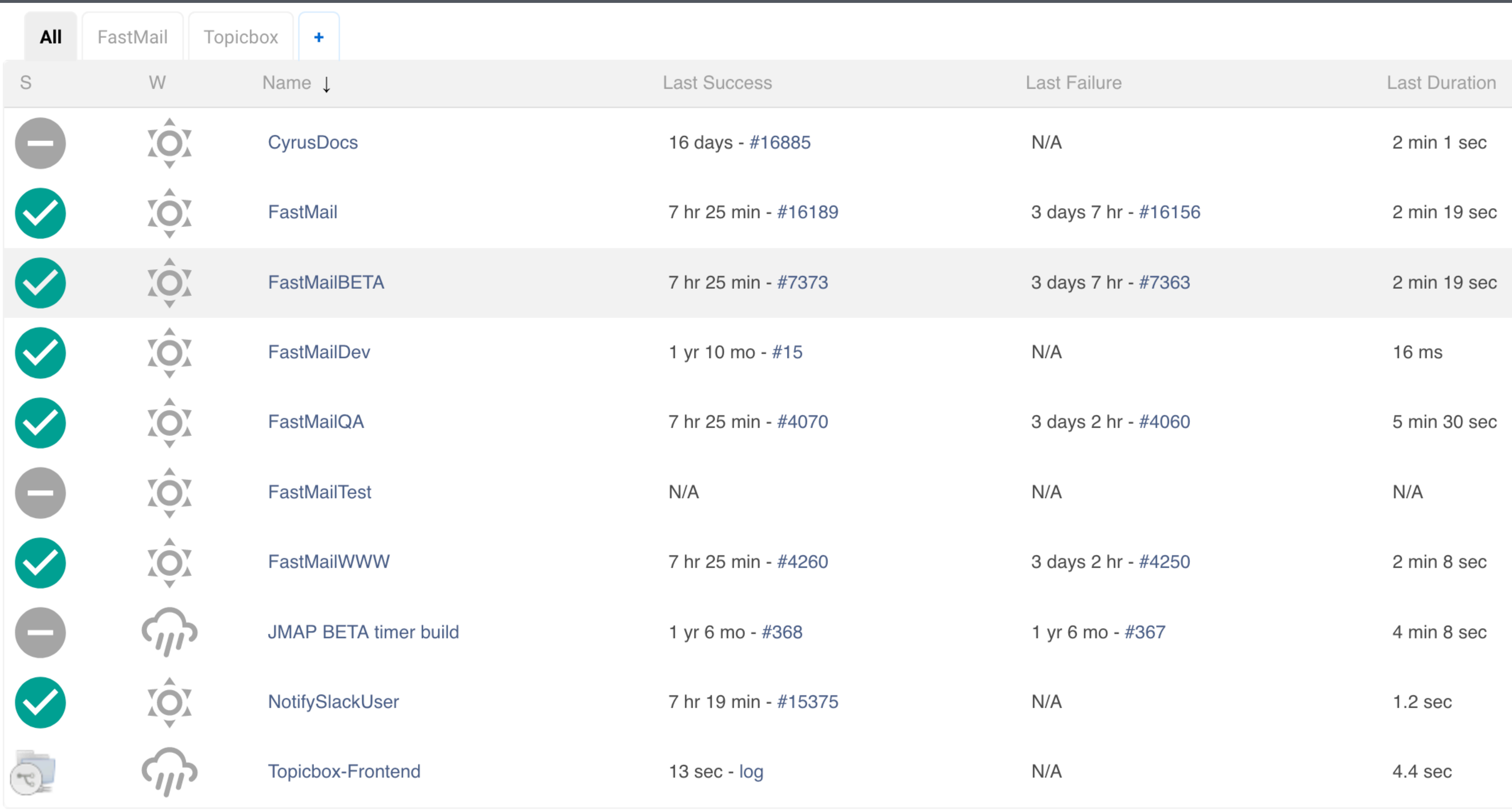
[this is the Fastmail UI; this is a hacky joke because the UI was designed by someone whose last name is Jenkins]

what even: the Jenkins UI

- but seriously, the Jenkins UI is a bit of a nightmare
- we can click through it live a bit later
- but for posterity, screenshotted here

how even: diagnosing problems

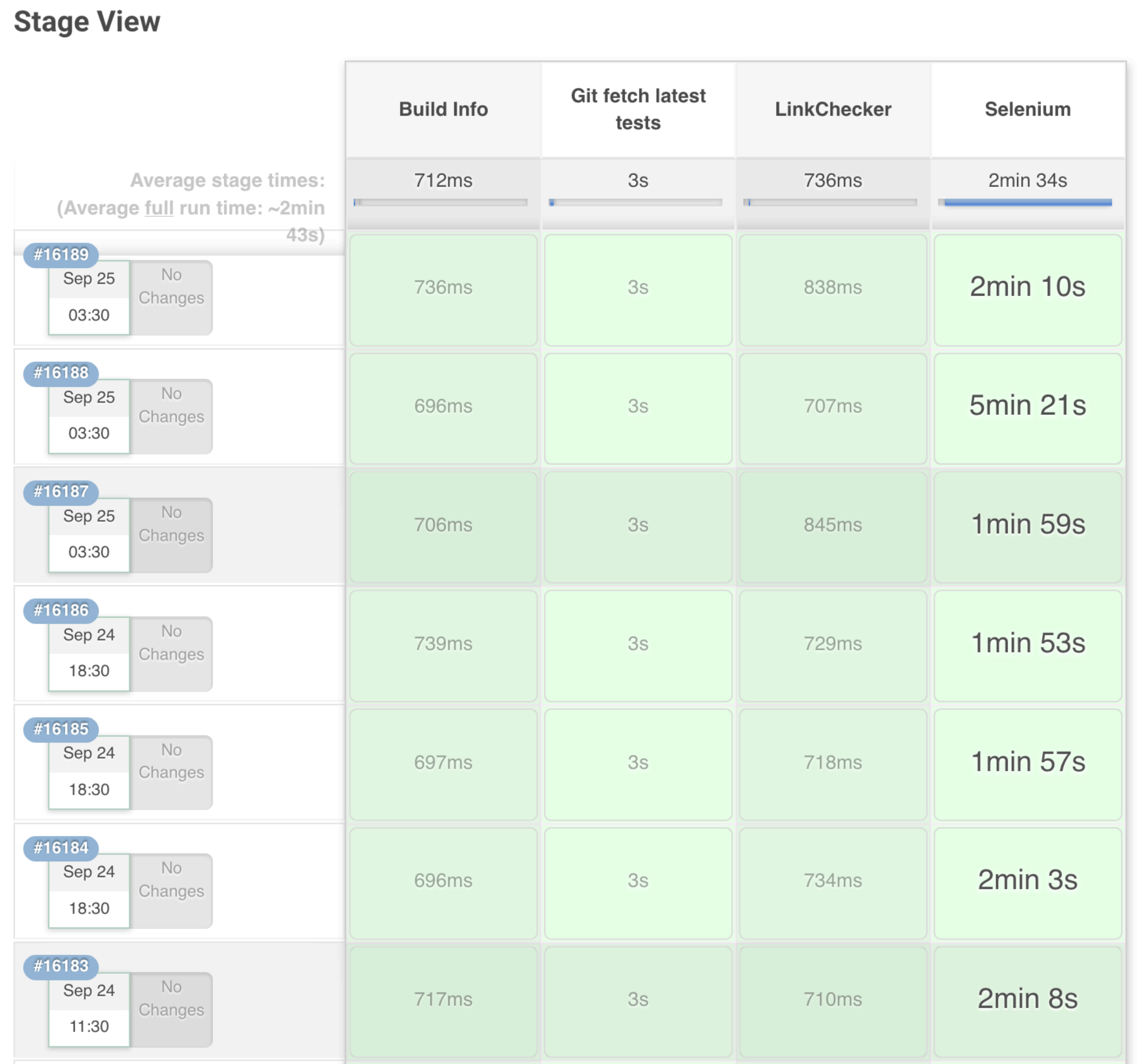
1. **find the build in Jenkins**
2. look for the test logs
3. find the failure
4. read the test files



S	W	Name ↓	Last Success	Last Failure	Last Duration
—	☀	CyrusDocs	16 days - #16885	N/A	2 min 1 sec
✓	☀	FastMail	7 hr 25 min - #16189	3 days 7 hr - #16156	2 min 19 sec
✓	☀	FastMailBETA	7 hr 25 min - #7373	3 days 7 hr - #7363	2 min 19 sec
✓	☀	FastMailDev	1 yr 10 mo - #15	N/A	16 ms
✓	☀	FastMailQA	7 hr 25 min - #4070	3 days 2 hr - #4060	5 min 30 sec
—	☀	FastMailTest	N/A	N/A	N/A
✓	☀	FastMailWWW	7 hr 25 min - #4260	3 days 2 hr - #4250	2 min 8 sec
—	🌧	JMAP BETA timer build	1 yr 6 mo - #368	1 yr 6 mo - #367	4 min 8 sec
✓	☀	NotifySlackUser	7 hr 19 min - #15375	N/A	1.2 sec
	🌧	Topicbox-Frontend	13 sec - log	N/A	4.4 sec

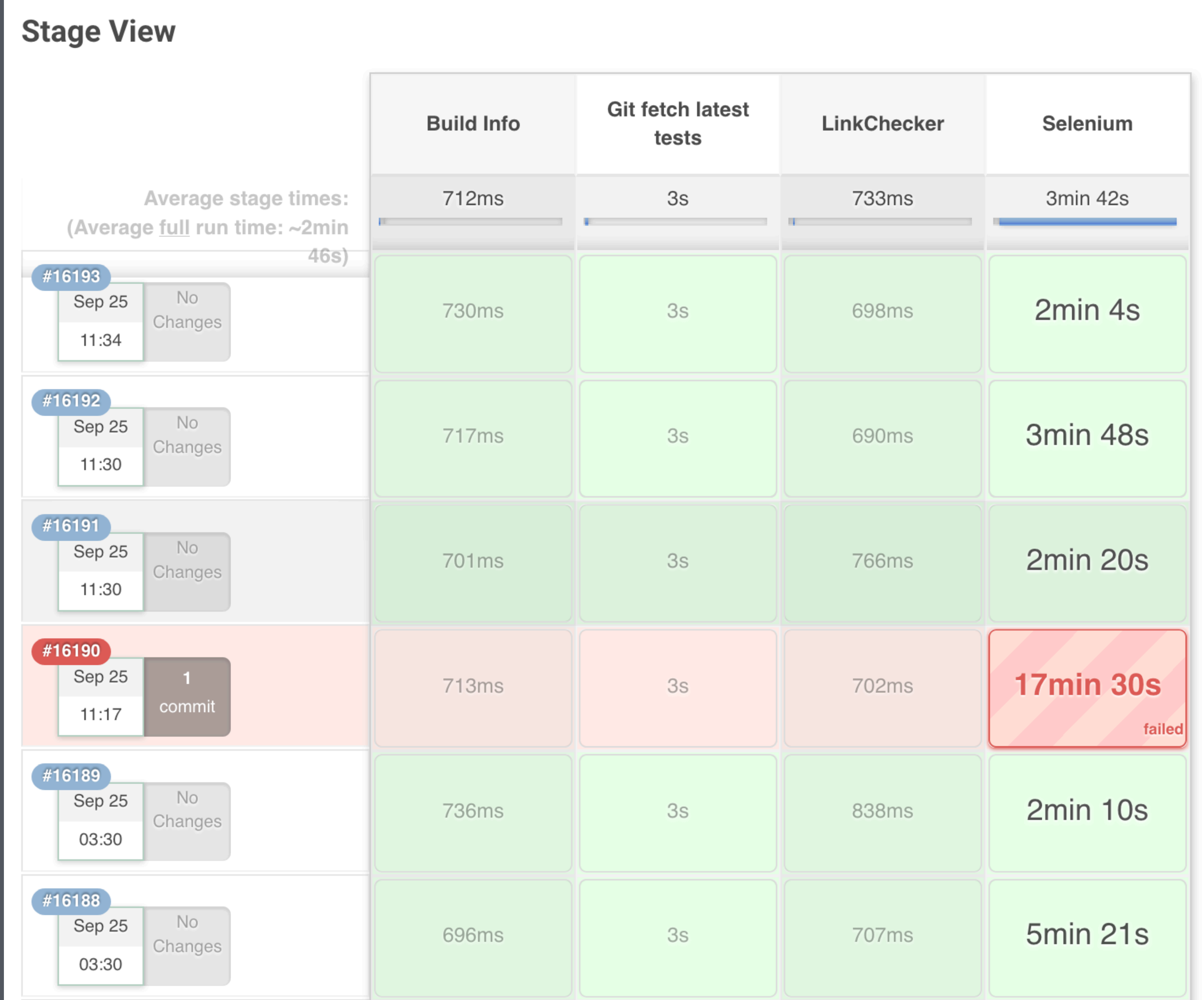
what even: the Jenkins UI

- there are jobs defined for every environment
- but those jobs have no useful information
- so everything you want is in the job labeled "Fastmail"



Stage View

	Build Info	Git fetch latest tests	LinkChecker	Selenium
Average stage times: (Average full run time: ~2min 43s)	712ms	3s	736ms	2min 34s
#16189 Sep 25 03:30 No Changes	736ms	3s	838ms	2min 10s
#16188 Sep 25 03:30 No Changes	696ms	3s	707ms	5min 21s
#16187 Sep 25 03:30 No Changes	706ms	3s	845ms	1min 59s
#16186 Sep 24 18:30 No Changes	739ms	3s	729ms	1min 53s
#16185 Sep 24 18:30 No Changes	697ms	3s	718ms	1min 57s
#16184 Sep 24 18:30 No Changes	696ms	3s	734ms	2min 3s
#16183 Sep 24 11:30 No Changes	717ms	3s	710ms	2min 8s

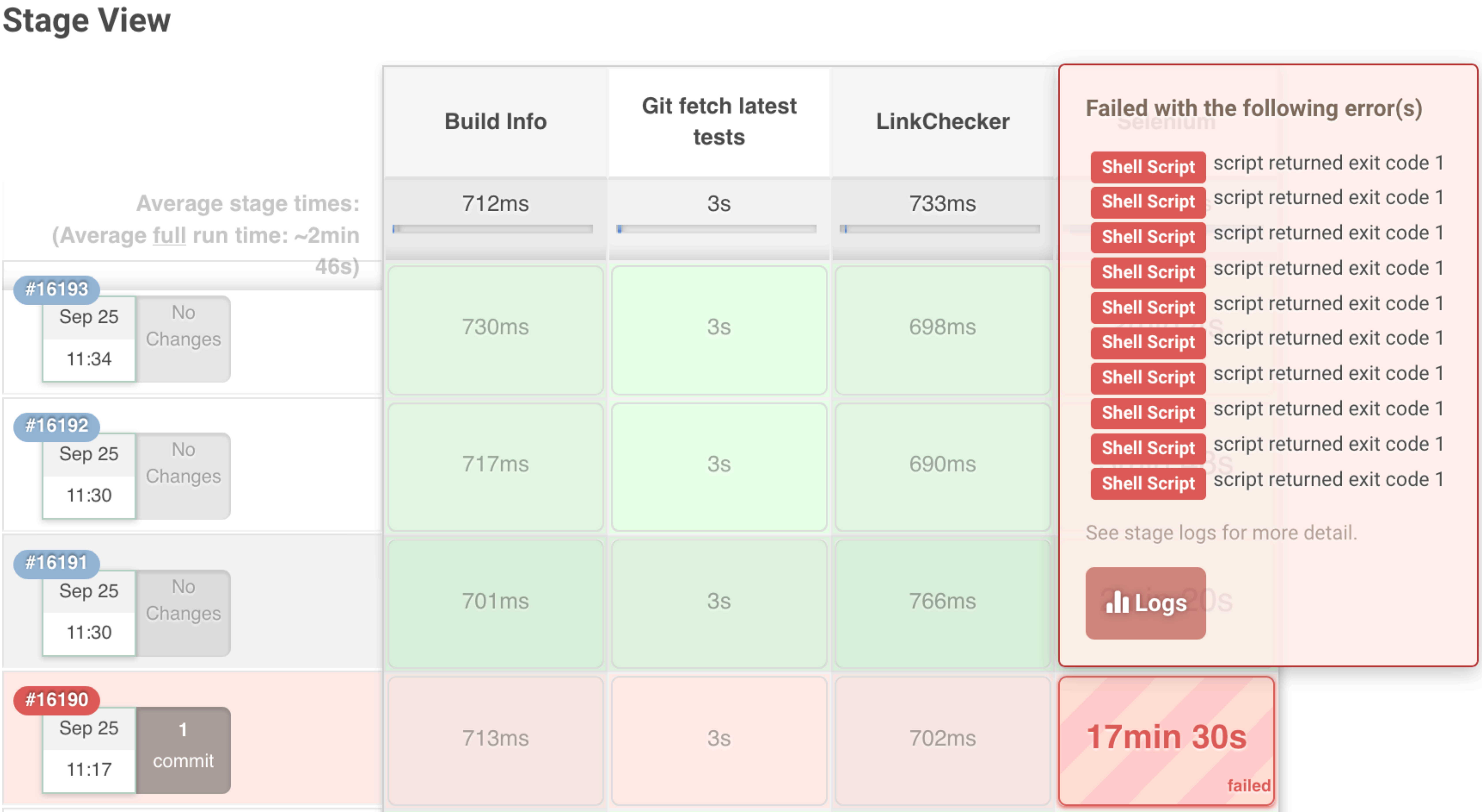


Stage View

	Build Info	Git fetch latest tests	LinkChecker	Selenium
Average stage times: (Average full run time: ~2min 46s)	712ms	3s	733ms	3min 42s
#16193 Sep 25 11:34 No Changes	730ms	3s	698ms	2min 4s
#16192 Sep 25 11:30 No Changes	717ms	3s	690ms	3min 48s
#16191 Sep 25 11:30 No Changes	701ms	3s	766ms	2min 20s
#16190 Sep 25 11:17 1 commit	713ms	3s	702ms	17min 30s failed
#16189 Sep 25 03:30 No Changes	736ms	3s	838ms	2min 10s
#16188 Sep 25 03:30 No Changes	696ms	3s	707ms	5min 21s

how even: diagnosing problems

1. find the build in Jenkins
2. **look for the test logs**
3. find the failure
4. read the test files



Stage View

	Build Info	Git fetch latest tests	LinkChecker	Selenium
Average stage times: (Average full run time: ~2min 46s)	712ms	3s	733ms	
#16193 Sep 25 11:34 No Changes	730ms	3s	698ms	
#16192 Sep 25 11:30 No Changes	717ms	3s	690ms	
#16191 Sep 25 11:30 No Changes	701ms	3s	766ms	
#16190 Sep 25 11:17 1 commit	713ms	3s	702ms	17min 30s failed

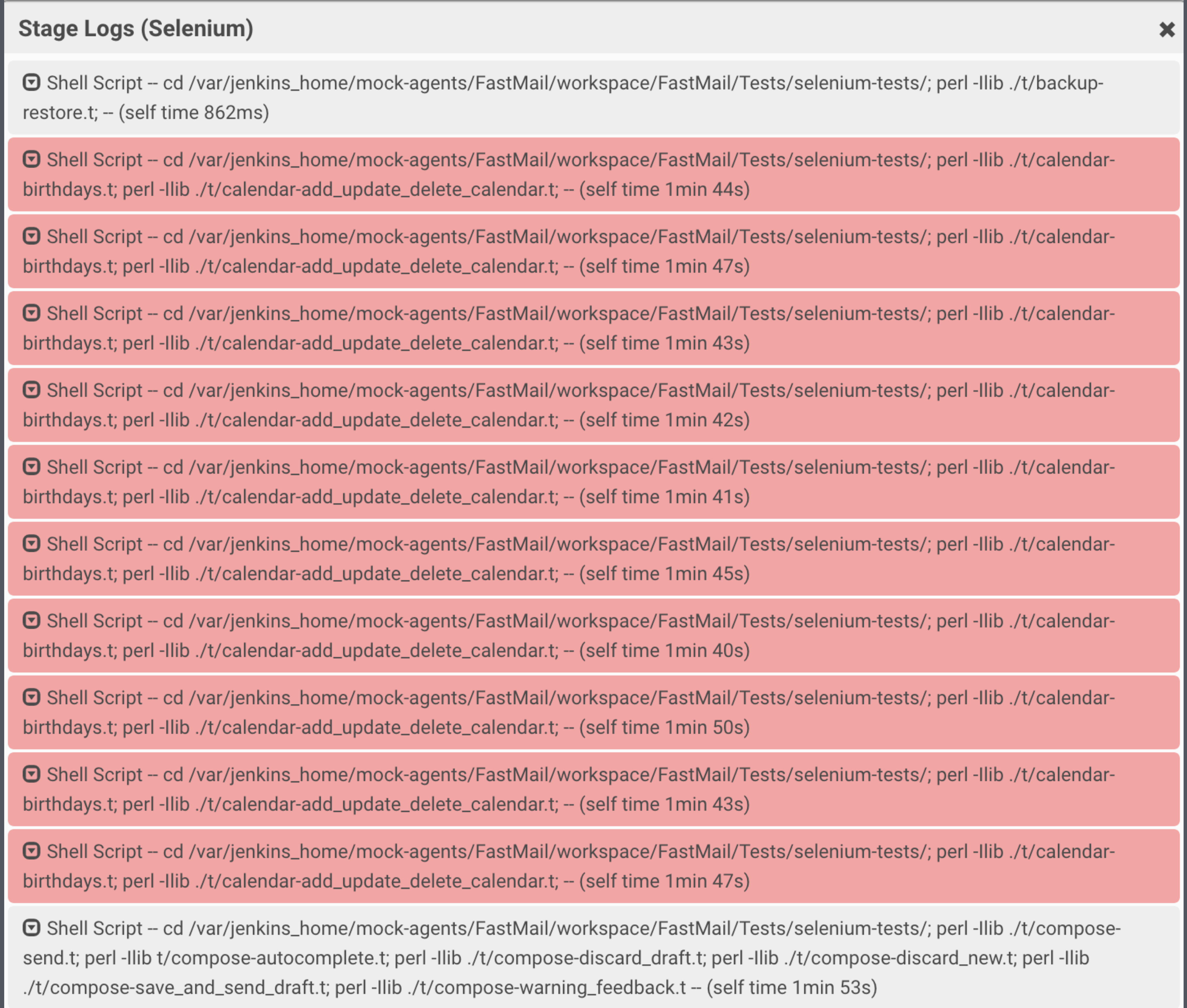
Failed with the following error(s)

Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1
Shell Script script returned exit code 1

See stage logs for more detail.

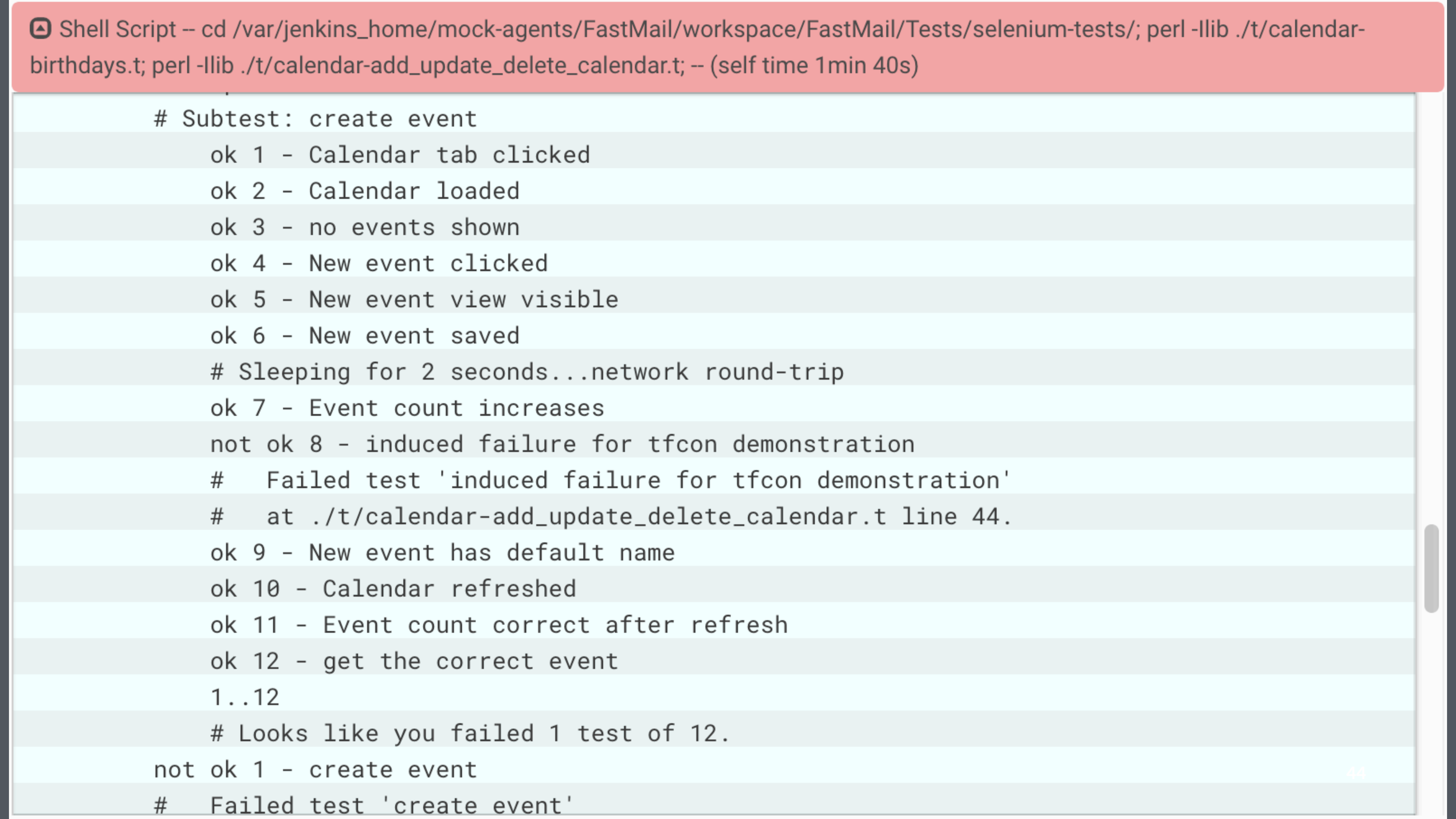
how even: diagnosing problems

1. find the build in Jenkins
2. look for the test logs
3. **find the failure**
4. read the test files



Stage Logs (Selenium)

```
Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/backup-restore.t; -- (self time 862ms)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 44s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 47s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 43s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 42s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 41s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 45s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 40s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 50s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 43s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 47s)

Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/compose-send.t; perl -Ilib t/compose-autocomplete.t; perl -Ilib ./t/compose-discard_draft.t; perl -Ilib ./t/compose-discard_new.t; perl -Ilib ./t/compose-save_and_send_draft.t; perl -Ilib ./t/compose-warning_feedback.t -- (self time 1min 53s)
```



Shell Script -- cd /var/jenkins_home/mock-agents/FastMail/workspace/FastMail/Tests/selenium-tests/; perl -Ilib ./t/calendar-birthdays.t; perl -Ilib ./t/calendar-add_update_delete_calendar.t; -- (self time 1min 40s)

```
    # Subtest: create event
        ok 1 - Calendar tab clicked
        ok 2 - Calendar loaded
        ok 3 - no events shown
        ok 4 - New event clicked
        ok 5 - New event view visible
        ok 6 - New event saved
        # Sleeping for 2 seconds...network round-trip
        ok 7 - Event count increases
        not ok 8 - induced failure for tfcon demonstration
        #   Failed test 'induced failure for tfcon demonstration'
        #   at ./t/calendar-add_update_delete_calendar.t line 44.
        ok 9 - New event has default name
        ok 10 - Calendar refreshed
        ok 11 - Event count correct after refresh
        ok 12 - get the correct event
        1..12
        # Looks like you failed 1 test of 12.
    not ok 1 - create event
    #   Failed test 'create event'
```

how even: diagnosing problems

```
ok 6 - New event saved
# Sleeping for 2 seconds...network round-trip
ok 7 - Event count increases
not ok 8 - induced failure for tfcon demonstration
#   Failed test 'induced failure for tfcon demonstration'
#   at ./t/calendar-add_update_delete_calendar.t line 44.
ok 9 - New event has default name`
```

how even: diagnosing problems

1. find the build in Jenkins
2. look for the test logs
3. find the failure
4. **read the test files**

how even does it all work?

how even: the framework

- I rewrote from scratch in 2018 (twice)
- so I know a lot about it!
- and soon, you will too!

how even: the framework

- [git host]/fastmail/testnet
- this repository is weird (has infra stuff too)
- `cd Tests/selenium-tests`, normal perl dir layout

```
[..s/selenium-tests on master*] $ tree
.
├── config
│   ├── ajax-254-fm-beta.conf
│   ├── ajax-254-fm-live.conf
│   └── ajax-254-fm-qa.confy
├── lib
│   └── FastMail
│       └── Selenium
│           ├── APIHelper.pm
│           ├── BrowserMap.pm
│           ├── Config.pm
│           ├── Driver.pm
│           ├── ElementFinder.pm
│           ├── JMAPTester.pm
│           ├── Tester
│           │   └── Abort.pm
│           ├── Tester.pm
│           └── WebElement.pm
└── t
    ├── backup-restore.t
    ├── ...
```

how even: the framework

```
use Test::Routine;
with 'FastMail::Selenium::Tester';

sub test_realm { 'calendar' }

test 'calendar CRUD' => sub ($self) {
  my $driver = $self->driver;

  $self->reset_account;
  $self->login or die "couldn't log in";
```

🚧 `reset_account` 🚧

why even: `reset_account`

- Selenium tests do not (and must not) create users
- But we still want to start fresh every time
- But these are just ordinary Fastmail users
- And if we want to test the UI, and the UI is broken, we can't use the UI to reset them

ok but seriously I don't really even want to tell you about this because it's scary and horrible but also you probably need to know it if you want to debug anything that's going on with these tests

how even: `reset_account`

- makes an LWP request to [internal url]/action/resetaccount/
- winds up in ME::ActionURL (in hm)
- which does IP checks
- and then runs utils/ResetAccount.pl (via Runner)
- which is *horrifying*

```
$ grep -i delete utils/ResetAccount.pl

DELETE FROM Personalities
DELETE FROM PopLinks
DELETE FROM CalendarDAVLinks
DELETE FROM SecondFactors
DELETE FROM ApplicationPasswords
DELETE FROM VerificationCodes
$Mailbox->delete_messages("1:*", 1);
$Folder->DeleteFolder($Mailbox);
DELETE FROM Rules
$Dbh->do("DELETE FROM ExternalCredentials WHERE UserId = ?", {}, $UserId);
$Dbh->do("DELETE FROM SavedSearches WHERE UserId = ?", {}, $UserId);
$User->VFS()->DeleteEntireVFS();
$User->CalDAV()->DeleteEntireCalDAV();
```

how even: `reset_account`

- different users per test realm
  (same user for all calendar tests)
- different users per environment
  (diff. users for calendar tests on beta/QA)
- defined in config files (in the repo)

how even: test framework

```
[..s/selenium-tests on master*] $ tree lib
lib
└── FastMail
    └── Selenium
        ├── APIHelper.pm
        ├── BrowserMap.pm
        ├── Config.pm
        ├── Driver.pm
        ├── ElementFinder.pm
        ├── JMAPTester.pm
        ├── Tester
        │   └── Abort.pm
        ├── Tester.pm
        └── WebElement.pm
```

what even: the boring bits

- FM::S::BrowserMap - literally says "use chrome on linux"
- FM::S::Tester::Abort - standard Test::Abortable copypasta
- FM::S::WebElement - subclass to add `->blur` to element objects

what even: the boring bits

- FM::S::BrowserMap - literally says "use chrome on linux"
- FM::S::Tester::Abort - standard Test::Abortable copypasta
- FM::S::WebElement - subclass to add `->blur` to element objects
- FM::S::JMAPTester -

```
package FastMail::Selenium::JMAPTester;
# This is shamefully copied directly from ME::Test::JMAPTester in hm.git, but
# with some code removed. -- michael, 2019-05-10
```

```
[..s/selenium-tests on master*] $ tree lib
lib
└── FastMail
    └── Selenium
        ├── APIHelper.pm
        ├── Config.pm
        ├── Driver.pm
        ├── ElementFinder.pm
        └── Tester.pm
```

what even: FastMail::Selenium::APIHelper

- handles `->login`
- talks to internal FM API
- which is *much* faster and less fragile than waiting for Selenium to click through the UI

what even: FastMail::Selenium::Config

- reads a config file, provides methods to get at it
- also not super interesting, just an excuse
- I *love* config objects (see also: Topicbox, mint-tag, synergy)

```
# FastMail::Selenium::Config

sub username_for ($self, $what) { $self->_config->{ $what }->{username} }
sub password_for ($self, $what) { $self->_config->{ $what }->{password} }
sub remote_server_addr   { $_[0]->_config->{selenium}->{remote_server_addr}  }
sub user_domain          { $_[0]->_config->{mail}->{user_domain}             }
sub ui                   { $_[0]->_config->{mail}->{ui}                      }
sub service              { $_[0]->_config->{mail}->{url}                     }
sub business_url         { $_[0]->_config->{mail}->{business_url}            }
sub password_reset       { $_[0]->_config->{mail}->{password_reset}          }
sub cc_address           { $_[0]->_config->{compose}->{cc_address}           }
sub bcc_address          { $_[0]->_config->{compose}->{bcc_address}          }
sub compose_account_name { $_[0]->_config->{compose}->{compose_account_name} }
```

what even: the interesting bits

- FastMail::Selenium::Tester
- FastMail::Selenium::Driver
- FastMail::Selenium::ElementFinder

(please forgive the big M; it was a long time ago)

what even: Fastmail::Selenium::Tester

what even: Tester.pm

- the Test::Routine class for these tests (like ME::Test::Routine)
- provides methods for common things
  (e.g., `driver`, `reset_account`, `login`, `populate_mailbox`)
- actually sets up Selenium to run

```
around run_test => sub ($orig, $self, $test){
  my $driver = eval { FastMail::Selenium::Driver->new($self->test_realm) };

  if ($@ && $@ =~ /did not return proper status/) {
    warn "Unable to run tests locally, will run on [internal url] instead: $@";
    local $ENV{USE_LOCAL_SELENIUM} = 0;
    $driver = FastMail::Selenium::Driver->new($test->name);
  } elsif ($@) {
    # We got some other error from Selenium: just give up.
    die "Driver instantiation failed: $@";
  }

  $self->driver($driver);
  $driver->set_implicit_wait_timeout(3000);

  $self->$orig($test);

  $self->driver->quit;
};
```

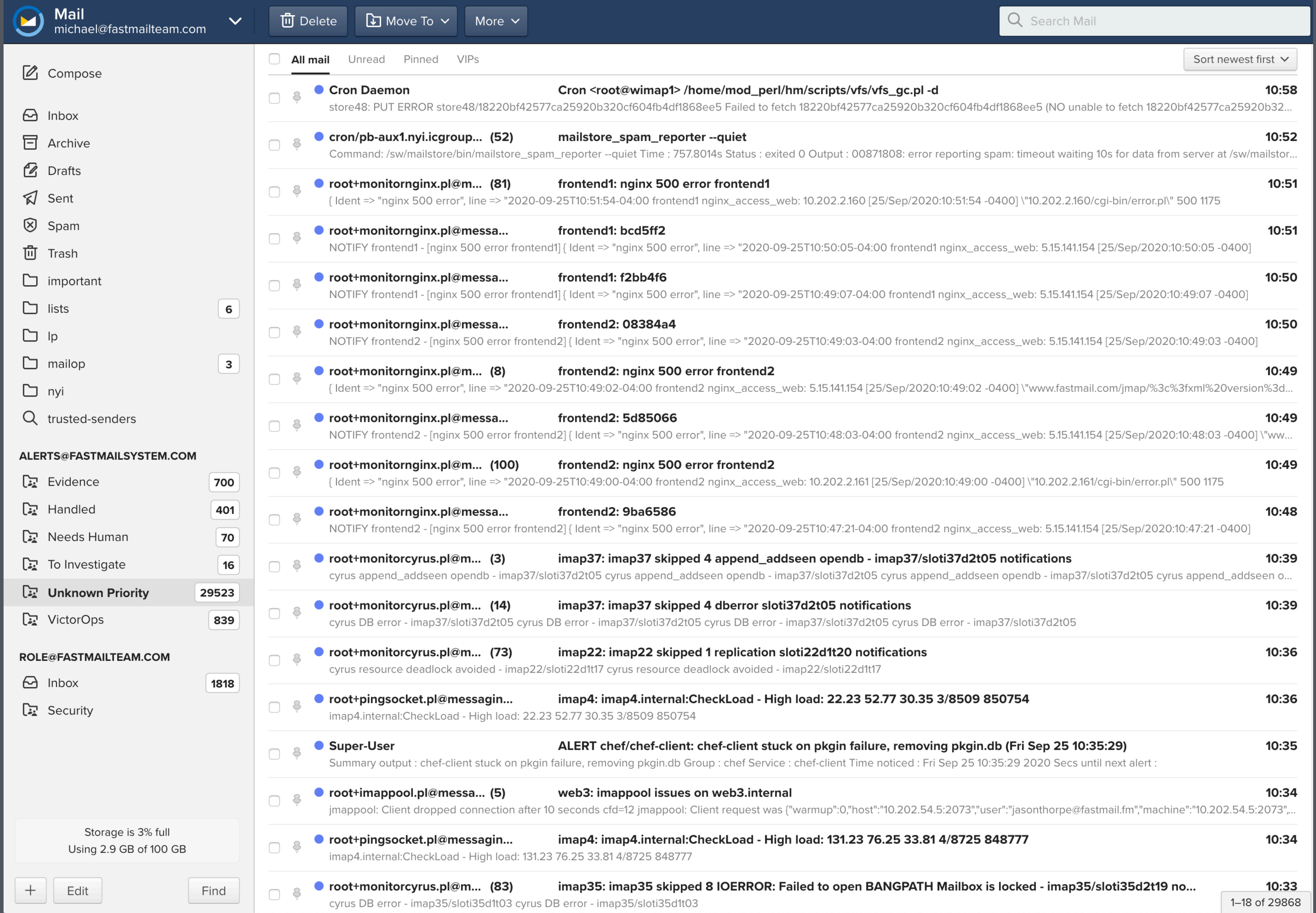
what even: Fastmail::Selenium::Driver

what even: Fastmail::Selenium::Driver

- a subclass of Selenium::Remote::Driver (on CPAN)
- with a bunch of additional methods for doing things in the UI (clicking buttons by name, navigating around)
- it's great, except:
  - Selenium::Remote::Driver kinda sucks.
  - because Selenium kinda sucks.

why even: does Selenium kinda suck

- we try to make it hard for robots to do things on the internet
- what is Selenium if not a robot on the internet?



Cron Daemon — **Cron <root@wimap1> /home/mod_perl/hm/scripts/vfs/vfs_gc.pl -d** — 10:58
store48: PUT ERROR store48/18220bf42577ca25920b320cf604fb4df1868ee5 Failed to fetch 18220bf42577ca25920b320cf604fb4df1868ee5 (NO unable to fetch 18220bf42577ca25920b32...

cron/pb-aux1.nyi.icgroup... (52) — **mailstore_spam_reporter --quiet** — 10:52
Command: /sw/mailstore/bin/mailstore_spam_reporter --quiet Time : 757.8014s Status : exited 0 Output : 00871808: error reporting spam: timeout waiting 10s for data from server at /sw/mailstor...

root+monitornginx.pl@m... (81) — **frontend1: nginx 500 error frontend1** — 10:51
{ Ident => "nginx 500 error", line => "2020-09-25T10:51:54-04:00 frontend1 nginx_access_web: 10.202.2.160 [25/Sep/2020:10:51:54 -0400] \"10.202.2.160/cgi-bin/error.pl\" 500 1175

root+monitornginx.pl@messa... — **frontend1: bcd5ff2** — 10:51
NOTIFY frontend1 - [nginx 500 error frontend1] { Ident => "nginx 500 error", line => "2020-09-25T10:50:05-04:00 frontend1 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:50:05 -0400]

root+monitornginx.pl@messa... — **frontend1: f2bb4f6** — 10:50
NOTIFY frontend1 - [nginx 500 error frontend1] { Ident => "nginx 500 error", line => "2020-09-25T10:49:07-04:00 frontend1 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:49:07 -0400]

root+monitornginx.pl@messa... — **frontend2: 08384a4** — 10:50
NOTIFY frontend2 - [nginx 500 error frontend2] { Ident => "nginx 500 error", line => "2020-09-25T10:49:03-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:49:03 -0400]

root+monitornginx.pl@m... (8) — **frontend2: nginx 500 error frontend2** — 10:49
{ Ident => "nginx 500 error", line => "2020-09-25T10:49:02-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:49:02 -0400] \"www.fastmail.com/jmap/%3c%3fxml%20version%3d...

root+monitornginx.pl@messa... — **frontend2: 5d85066** — 10:49
NOTIFY frontend2 - [nginx 500 error frontend2] { Ident => "nginx 500 error", line => "2020-09-25T10:48:03-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:48:03 -0400] \"ww...

root+monitornginx.pl@m... (100) — **frontend2: nginx 500 error frontend2** — 10:49
{ Ident => "nginx 500 error", line => "2020-09-25T10:49:00-04:00 frontend2 nginx_access_web: 10.202.2.161 [25/Sep/2020:10:49:00 -0400] \"10.202.2.161/cgi-bin/error.pl\" 500 1175

root+monitornginx.pl@messa... — **frontend2: 9ba6586** — 10:48
NOTIFY frontend2 - [nginx 500 error frontend2] { Ident => "nginx 500 error", line => "2020-09-25T10:47:21-04:00 frontend2 nginx_access_web: 5.15.141.154 [25/Sep/2020:10:47:21 -0400]

root+monitorcyrus.pl@m... (3) — **imap37: imap37 skipped 4 append_addseen opendb - imap37/sloti37d2t05 notifications** — 10:39
cyrus append_addseen opendb - imap37/sloti37d2t05 cyrus append_addseen opendb - imap37/sloti37d2t05 cyrus append_addseen opendb - imap37/sloti37d2t05 cyrus append_addseen o...

root+monitorcyrus.pl@m... (14) — **imap37: imap37 skipped 4 dberror sloti37d2t05 notifications** — 10:39
cyrus DB error - imap37/sloti37d2t05 cyrus DB error - imap37/sloti37d2t05 cyrus DB error - imap37/sloti37d2t05 cyrus DB error - imap37/sloti37d2t05

root+monitorcyrus.pl@m... (73) — **imap22: imap22 skipped 1 replication sloti22d1t20 notifications** — 10:36
cyrus resource deadlock avoided - imap22/sloti22d1t17 cyrus resource deadlock avoided - imap22/sloti22d1t17

root+pingsocket.pl@messagin... — **imap4: imap4.internal:CheckLoad - High load: 22.23 52.77 30.35 3/8509 850754** — 10:36
imap4.internal:CheckLoad - High load: 22.23 52.77 30.35 3/8509 850754

Super-User — **ALERT chef/chef-client: chef-client stuck on pkgin failure, removing pkgin.db (Fri Sep 25 10:35:29)** — 10:35
Summary output : chef-client stuck on pkgin failure, removing pkgin.db Group : chef Service : chef-client Time noticed : Fri Sep 25 10:35:29 2020 Secs until next alert :

root+imappool.pl@messa... (5) — **web3: imappool issues on web3.internal** — 10:34
jmappool: Client dropped connection after 10 seconds cfd=12 jmappool: Client request was {"warmup":0,"host":"10.202.54.5:2073","user":"jasonthorpe@fastmail.fm","machine":"10.202.54.5:2073",...

root+pingsocket.pl@messagin... — **imap4: imap4.internal:CheckLoad - High load: 131.23 76.25 33.81 4/8725 848777** — 10:34
imap4.internal:CheckLoad - High load: 131.23 76.25 33.81 4/8725 848777

root+monitorcyrus.pl@m... (83) — **imap35: imap35 skipped 8 IOERROR: Failed to open BANGPATH Mailbox is locked - imap35/sloti35d2t19 no...** — 10:33
cyrus DB error - imap35/sloti35d1t03 cyrus DB error - imap35/sloti35d1t03

why even: does Selenium kinda suck

- we're talking to Selenium through Perl
- which is a crappy translation layer over Selenium's JSON API
- which is a crappy translation layer over Selenium's Java API
- and it's not well documented, to boot (and often breaks)

how even: Selenium from above

- navigate to a URL
- find elements on the page
- do stuff with them

how even: Selenium from above

- navigate to a URL
- **find elements on the page ⁉️**
- do stuff with them

what even: Fastmail::Selenium::Driver

what even: Fastmail::Selenium::Driver

- configured with a bunch of poorly-named environment variables:
  - `MAIL_SELENIUM_TESTS_CONFIG` or `FMQA_SERVICE_UNDER_TEST`
  - `USE_LOCAL_SELENIUM` (optional)
  - `JENKINS_HOME` (optional)
  - `FMQA_USE_BROWSER` (mostly unused)

what even: Fastmail::Selenium::Driver

- opens a connection to Selenium itself to control the browser
- and then drives it, mostly by clicking around through the UI

what even: Fastmail::Selenium::Driver

- whole bunch of convenience methods
- 63 methods in this file, but less than 1000 lines
- I won't tell you about all of them!
- but will tell you about *some* of them

what even: Driver test methods

- test utility methods
- `login`, `logout`, `login_url`
- `url_for`
- `kill_test`, `print_session_details`, `sort_items`

what even: Driver test methods

- waiting methods (avoid `sleep` littered about)
- `sleep`, `sleep_up_to`
- `do_when_possible`, `refresh_when_possible`
- `with_implicit_wait_timeout`
- `wait_for_X`:
  X ∈ { `contacts`, `conversations`, `notes`, `mailbox_load`, `message_sent`, `notification_text` }

what even: Driver test methods

- element retrieval convenience methods
- `get_element_named`, `get_child_element_named`
- `get_app_name`
- `get_calendar_events`, `get_contacts`, `get_messages`, `get_notification`

what even: Driver test methods

- click convenience methods
- `click_button`, `click_link_text`, `click_checkbox_in`
- FM UI affordances
  - `click_main_menu_item`
  - `click_sidebar_item`, `click_settings_sidebar_item`
  - `click_event_named`

what even: Driver test methods

- Driver exists mainly to find elements on the page (⁉)
- basically two methods to do this:
  - `find_element`
  - `get_element_named`

how even: finding elements

- `find_element` - by CSS selector, XPath, link text, [others]
- which is *fiiine* but:

  ```
  find_element(
   '.v-EditEvent-part:nth-of-type(1) .v-TimeText
  input'
  )
  ```
- lots of work into making sure you never have to type that
- enter: ElementFinder.pm

what even: ElementFinder

```
# write this
$driver->click_element_named('calendar_birthday_toggle');

# instead of
my $elem = eval {
  $driver->find_element('.v-CalendarVisibility--birthdays .v-Icon.i-present');
};
abort($@) if $@;
```

what even: ElementFinder

- very straightforward
- abstracts away all the selectors

```
sub beta_modal_close_button {
  my $self = shift;
  return with_abort { $self->find_element('button.v-BetaInfo-close') };
}

sub contact_list_items {
  my $self = shift;
  return with_abort { $self->find_elements('.v-ContactItem') };
}
```

```perl
sub with_abort (&) {
  my ($code) = @_;
  my @ret = eval { $code->() };
  abort($@) if $@;

  return @ret if wantarray;
  return $ret[0];
}
```

where even were we: an overview

- FM::S::Tester - Test::Routine class
- FM::S::Driver - Selenium::Remote::Driver subclass
- FM::S::ElementFinder - selector abstractions

what even does a test look like

```perl
subtest 'read message detail' => sub {
  my ($message) = $driver->get_messages;

  ok(
    $driver->find_child_element($message, 'Show details', 'link_text')->click,
    'Message details toggle clicked'
  );

  $self->sleep(0.5, 'detail open animation');

  my $expect_addr = sprintf('%s <%s>',
    $self->config->compose_account_name,
    $self->address_for_test
  );

  my @details = map {; $_->get_text } $driver->get_message_details($message);

  is($details[0], $expect_addr, 'From address matches current account address');
  is($details[1], $self->address_for_test, 'To address is the same address as used on compose');
  is($details[2], $self->config->cc_address, 'CC address is the same address as used on compose');

  my @cards = $driver->get_elements_named('message_items');
  my $text = $driver->get_message_plain_text($cards[0]);

  is($text, $body, 'Body text matches sent message text');
};
```

```
subtest 'read message detail' => sub {
  my ($message) = $driver->get_messages;

  ok(
    $driver->find_child_element($message, 'Show details', 'link_text')->click,
    'Message details toggle clicked'
  );

  $self->sleep(0.5, 'detail open animation');
  my @details = map {; $_->get_text } $driver->get_message_details($message);

  is($details[0], $expect_addr, 'From address ok');
  is($details[1], $self->address_for_test, 'To address is right');
  is($details[2], $self->config->cc_address, 'CC address is right');

  my @cards = $driver->get_elements_named('message_items');
  my $text = $driver->get_message_plain_text($cards[0]);
  is($text, $body, 'Body text matches sent message text');
};
```

how even could we improve these tests?

- unbreak currently broken (skipped) tests
- new tests for new features
- test on more browsers/platforms (especially: mobile)
- require less arcane knowledge (especially: infrastructure)

what even have we learned?

- what even Selenium is
- where even the test code all lives
- how even the tests are all put together
- who even is the chief perpetrator of Selenium (me)
- why even we bother having these tests
- when (in the) even(t) something is broken, how to fix it

ok 1 - 🚑 Selenium Tests: WTF Even? 🚧

the end