Real life Django deployment

by **Jochem Oosterveen**

Brussels - 8th May 2015

IAmA

- **Information Science student**
  from Utrecht University

- **Python programmer**
  at Conceptive Engineering in Herentals

- emeritus **Linux sysadmin**
  at NextGear in Amsterdam

Will discuss

- My favourite stack (I am opinionated)
- Ansible and Fabric (which one to use for what?)
- Security, monitoring and #struggle

The stack

- Ubuntu (+ postfix)
- PostgreSQL
- nginx
- supervisor
- gunicorn
- Django, WSGI

Back in the days

- Early 2010
- FastCGI
- lighttpd
- daemontools
- no virtualenv

```
[jochem@snake ~]$ cat /usr/local/etc/lighttpd/vhosts.d/django-maastricht.conf
$HTTP["host"] =~ "^(www\.|)maastricht(\.djangohost|)\.nl" {
    accesslog.filename = "/var/log/lighttpd/maastricht.djangohost.nl-access.log"
    fastcgi.server = (
        "/maastricht.fcgi" => (
            "main" => (
                "host" => "127.0.0.1",
                "port" => 8008,
                "check-local" => "disable",
            )
        ),
    )
    alias.url = (
        "/afbeeldingen/" => "/home/maastricht/maastricht/media/images/",
        "/css/" => "/home/maastricht/maastricht/media/css/",
        "/media/" => "/home/maastricht/maastricht/media/",
    )

    url.rewrite-once = (
        "^(/afbeeldingen.*)$" => "$1",
        "^(/css.*)$" => "$1",
        "^(/media.*)$" => "$1",
        "^/favicon\.ico$" => "/media/favicon.ico",
        "^(/.*)$" => "/maastricht.fcgi$1",
    )
}
[jochem@snake ~]$
```

```
[root@snake /var/service]# cat django-
acme/run
#!/bin/sh

exec setuidgid acme envdir ./env
   /usr/local/bin/python
  /home/acme/django/manage.py runfcgi
        protocol=fcgi host=127.0.0.1
         port=8004
        –settings=“acme/settings.py"
         method=prefork daemonize=false
```

WSGI

- **WSGI -** Web Server Gateway Interface (PEP 333, PEP 3333)
- Interface between application and server
- e.g. Django, Flask, Tornado

The host(s)

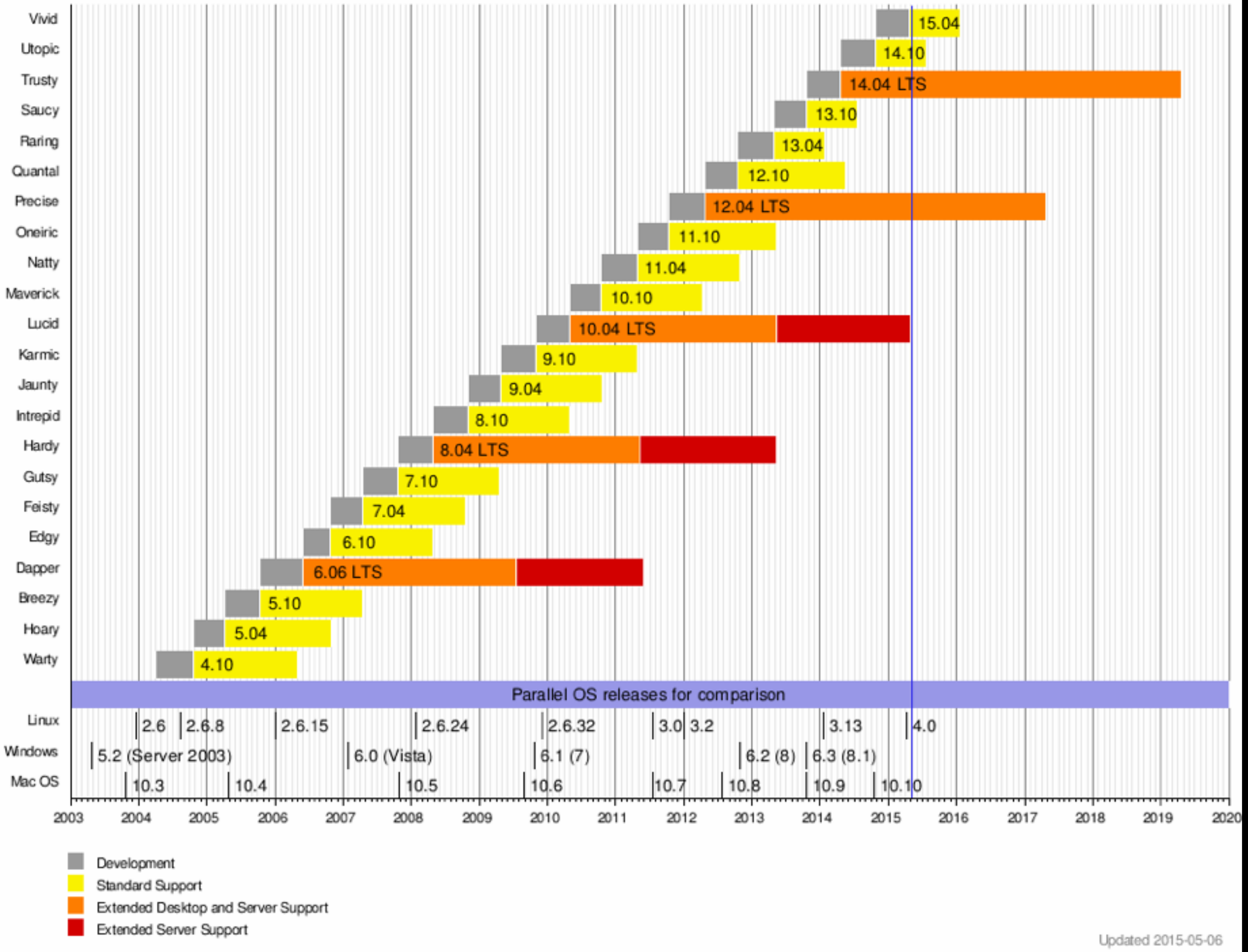
- Bare metal
- VPS
- Amazon EC2
- …doesn’t really matter


XS4ALL

Getting the system running

- PXE image
- Clone of existing machine
- vanilla install + Ansible

Ubuntu Release Timeline



You had to ship it yesterday

- So you…
  - …hope it's still working
  - …manually configured 'stuff'
  - …manually did the deployment
  - …run the application in a screen/tmux session

Let’s assume…

- You’re sensible person and you are using virtualenv.
- We’re using the green unicorn: gunicorn (WSGI HTTP Server)
    - Alternative: uWSGI

Fixing it

- Running the application:
  use a process manager
  - init script (what if it crashes?)
  - DJB's daemontools (yeah nope)
  - runit (meh)
  - supervisor (yay!)

Supervisor config

- apt-get install supervisor
- Create a file per project in /etc/supervisor/conf.d/
- Define programs and groups

eggs.conf

[group:eggs-production]
programs=eggs-production-gunicorn, eggs-production-celery

[program:eggs-production-gunicorn]
command=/home/eggs/pr/.venv/bin/gunicorn -b 127.0.0.1:1201 eggs.wsgi:application -w4
directory=/home/eggs/production/
user=eggs
autostart=true
autorestart=true
redirect_stderr=true

[program:eggs-production-celery]
command=/home/eggs/production/.virtualenv/bin/python manage.py celery worker
directory=/home/eggs/production/
user=eggs
autostart=true
autorestart=true
redirect_stderr=true

supervisorctl

```
# supervisorctl
eggs-production:eggs-production-celery                RUNNING    pid 18301, uptime 0:00:03
eggs-production:eggs-production-gunicorn              RUNNING    pid 18317, uptime 0:00:02
eggs-production:eggs-production-sockjs                RUNNING    pid 3155, uptime 41 days,
14:28:10
supervisor>

supervisor> restart eggs-production:*

supervisor> reload
```

Scripting the deployment

- Fabric, http://www.fabfile.org/
  - “tool for streamlining the use of SSH for application deployment or systems administration tasks”
- Write Python, store it with the code

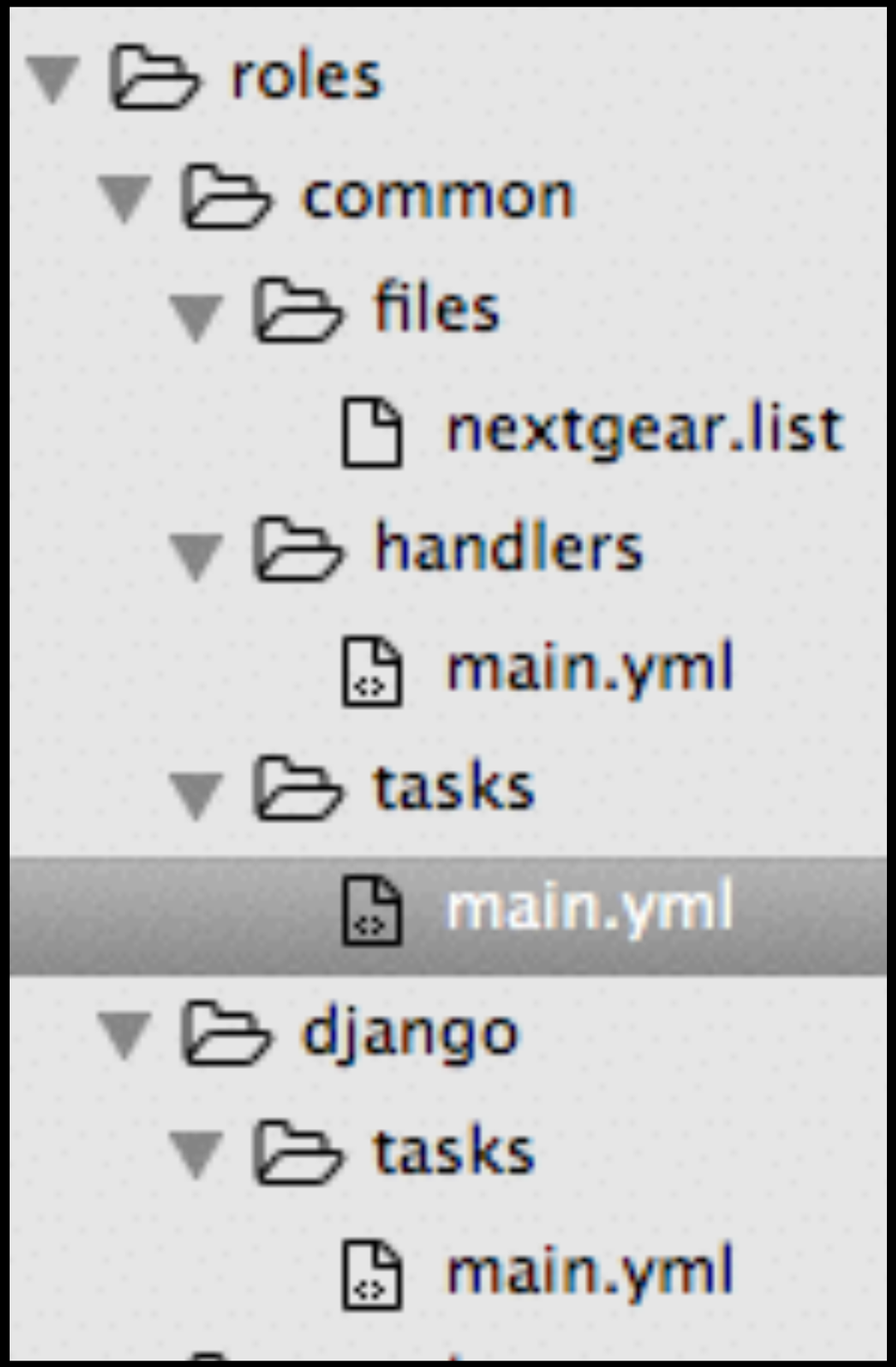
```
83  @runs_once
84  def deploy():
85      require('name', provided_by=('staging', 'production'))
86      execute(check_environment)
87      execute(pull)
88      with cd(env.root):
89          with cd('bluesuit'):
90              if not files.exists('local_settings.py'):
91                  run('ln -s %s_settings.py local_settings.py' % env.name)
92          with prefix('source .virtualenv/bin/activate'):
93              execute(update_requirements)
94              execute(migrate)
95              collect_static()
96      with settings(user='jochem'):
97          execute(restart)
98
```

```
57
58  @parallel
59  def check_environment():
60      if not files.exists(env.root):
61          run('mkdir %s' % env.root)
62      with cd(env.root):
63          if not files.exists('.git'):
64              print("Creating repository")
65              run('git init')
66              run('git remote add origin git@bitbucket.org:nextgear/bluesuit.git')
67          if not files.exists('.virtualenv'):
68              print("Creating virtualenv")
69              run('virtualenv .virtualenv')
70
```

Manually configured ‘stuff’

- Config management
- So why Ansible?
  - Get to the point where we could run existing fab file
  - Documentation by configuration
  - Requires only SSH, runs anywhere
  - Python package, Jinja2 templates

```yaml
---
- hosts: flyingcircus
  sudo: yes
  roles:
    - common
    - django

  tasks:
  - name: create system user
    user: name=flyingcircus group=django

  - name: create .ssh dir
    file: path=/home/flyingcircus/.ssh state=directory owner=flyingcircus mode=0700

  - name: copy ssh-key
    copy: src=flyingcircus/deploy.key dest=/home/flyingcircus/.ssh/id_rsa owner=flyingcircus mode=0600

  - rabbitmq_vhost: name=flyingcircus state=present
    delegate_to: "{{ rabbitmq_host }}"

  - rabbitmq_user: user=flyingcircus
                   password=******
                   vhost=flyingcircus
                   configure_priv=.*
                   read_priv=.*
                   write_priv=.*
                   state=present
    delegate_to: "{{ rabbitmq_host }}"

  # https://github.com/Supervisor/supervisor/issues/121
  - template: src=flyingcircus/supervisor.conf.j2 dest=/etc/supervisor/conf.d/flyingcircus.conf mode=0644
    notify: restart supervisor

  - copy: src=flyingcircus/{{ item }} dest=/etc/nginx/ssl/
    with_items:
      - flyingcircus_nl.key
      - flyingcircus_nl.pem
    notify: reload nginx

  - template: src=flyingcircus/nginx.conf.j2 dest=/etc/nginx/sites-available/flyingcircus.conf mode=0644
    when: loadbalancer is not defined
    notify: reload nginx

  - template: src=flyingcircus/nginx.conf.j2 dest=/etc/nginx/sites-available/flyingcircus.conf mode=0644
    when: loadbalancer is defined
    delegate_to: "{{ loadbalancer }}"
    notify: reload nginx

  - file: src=/etc/nginx/sites-available/flyingcircus.conf dest=/etc/nginx/sites-enabled/flyingcircus.conf state=link
    when: loadbalancer is not defined
    notify: reload nginx

  - file: src=/etc/nginx/sites-available/flyingcircus.conf dest=/etc/nginx/sites-enabled/flyingcircus.conf state=link
    when: loadbalancer is defined
    delegate_to: "{{ loadbalancer }}"
    notify: reload nginx
```




```yaml
---
- name: add NextGear apt repository
  synchronize: src=nextgear.list dest=/etc/apt/sourc

- name: update apt cache
  apt: update_cache=yes

- name: install/upgrade default packages
  action: apt pkg={{ item }} state=latest
  tags: common
  with_items:
    - openntpd

- name: install/upgrade NextGear packages
  action: apt pkg={{ item }} state=latest force=yes
  tags: common
  with_items:
    - ng-monitor
```

```yaml
---
- name: restart apache
  service: name=apache2 state=restarted

- name: restart mysql
  service: name=mysql state=restarted

- name: restart nagios
  command: service nagios3 restart
  delegate_to: "{{ nagios_host }}"

- name: restart supervisor
  service: name=supervisor state=restarted

- name: reload nginx
  service: name=nginx state=reloaded
```

```
(ansible)verre:ansible jochem$ ansible-playbook -i hosts bluesuit.yml

PLAY [bluesuit] ***************************************************************

GATHERING FACTS ***************************************************************
ok: [moves-priv.nextgear.nl]

TASK: [common | add NextGear apt repository] **********************************
ok: [moves-priv.nextgear.nl -> 127.0.0.1]

TASK: [common | update apt cache] *********************************************
ok: [moves-priv.nextgear.nl]

TASK: [common | install/upgrade default packages] *****************************
ok: [moves-priv.nextgear.nl] => (item=openntpd)

TASK: [common | install/upgrade NextGear packages] ****************************
ok: [moves-priv.nextgear.nl] => (item=ng-monitor)

TASK: [django | install/upgrade packages] **************************************
ok: [moves-priv.nextgear.nl] => (item=git,memcached,python-virtualenv)

TASK: [create system user] ****************************************************
ok: [moves-priv.nextgear.nl]

TASK: [create .ssh dir] *******************************************************
ok: [moves-priv.nextgear.nl]

TASK: [copy ssh-key] **********************************************************
ok: [moves-priv.nextgear.nl]

TASK: [rabbitmq_vhost name=bluesuit state=present] ****************************
fatal: [moves-priv.nextgear.nl -> media01.priv.nextgear.net] => Missing sudo password

FATAL: all hosts have already failed -- aborting

PLAY RECAP ********************************************************************
           to retry, use: --limit @/Users/jochem/bluesuit.retry

moves-priv.nextgear.nl     : ok=9    changed=0    unreachable=1    failed=0

(ansible)verre:ansible jochem$
```

nginx

```
upstream pythonic-production {
    server 192.168.88.21:2201;
    server 192.168.88.22:2201;
}

server {
    listen 172.30.20.10;
    server_name pythonic.be;
    access_log /var/log/nginx/pythonic.be-access.log;
    error_log /var/log/nginx/pythonic.be-error.log;

    location / {
        proxy_pass    http://pythonic-production;
}
```

Security

- Erik Romijn's "pony check" is a good start: http://ponycheckup.com/
- SSL, sslcertifcaten.nl (any Belgian company?)
  - Doing it right is hard
  - Mixed content, resources from other domains

Privacy error

Jochem

https://cipherli.st

Your connection is not private

Attackers might be trying to steal your information from **cipherli.st** (for example, passwords, messages, or credit cards). NET::ERR_SSL_PINNED_KEY_NOT_IN_CERT_CHAIN

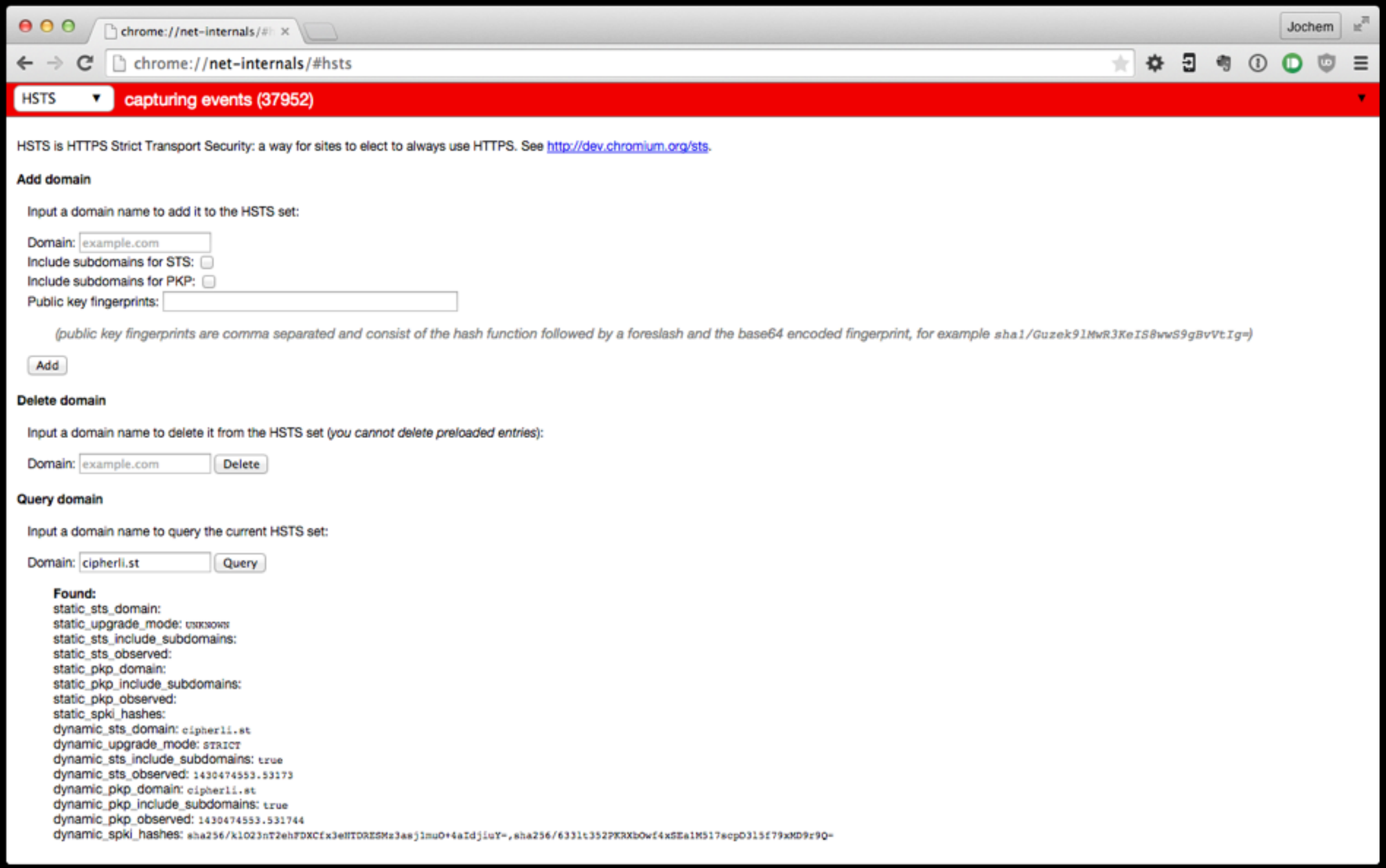
Hide advanced

Reload

cipherli.st normally uses encryption to protect your information. When Chrome tried to connect to cipherli.st this time, the website sent back unusual and incorrect credentials. Either an attacker is trying to pretend to be cipherli.st, or a Wi-Fi sign-in screen has interrupted the connection. Your information is still secure because Chrome stopped the connection before any data was exchanged.

You cannot visit cipherli.st right now because the website uses certificate pinning. Network errors and attacks are usually temporary, so this page will probably work later.

```
verre:~ jochem$ curl -v cipherli.st
* Rebuilt URL to: cipherli.st/
*   Trying 178.62.214.186...
* Connected to cipherli.st (178.62.214.186) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.41.0
> Host: cipherli.st
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Server: nginx
< Date: Thu, 07 May 2015 20:06:38 GMT
< Content-Type: text/html
< Content-Length: 178
< Connection: keep-alive
< Location: https://cipherli.st/
< Public-Key-Pins: pin-sha256="AWFBG0p7yngPzJBvz37wsEko4HY1X0TS24C5X0s91nw="; pin-sha256="EohwrK1N7rr
< Strict-Transport-Security: max-age=63072000; includeSubDomains;
< Coffee: Black
< Tea: Earl-Gray
< X-Frame-Options: DENY
< X-Content-Type-Options: nosniff
<
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
* Connection #0 to host cipherli.st left intact
verre:~ jochem$
```


chrome://net-internals/#hsts
Jochem
chrome://net-internals/#hsts
HSTS
capturing events (37952)
HSTS is HTTPS Strict Transport Security: a way for sites to elect to always use HTTPS. See http://dev.chromium.org/sts.
Add domain
Input a domain name to add it to the HSTS set:
Domain: example.com
Include subdomains for STS:
Include subdomains for PKP:
Public key fingerprints:
(public key fingerprints are comma separated and consist of the hash function followed by a foreslash and the base64 encoded fingerprint, for example sha1/Guzek9lMwR3KeIS8wwS9gBvVtIg=)
Add
Delete domain
Input a domain name to delete it from the HSTS set (you cannot delete preloaded entries):
Domain: example.com
Delete
Query domain
Input a domain name to query the current HSTS set:
Domain: cipherli.st
Query
Found:
static_sts_domain:
static_upgrade_mode: UNKNOWN
static_sts_include_subdomains:
static_sts_observed:
static_pkp_domain:
static_pkp_include_subdomains:
static_pkp_observed:
static_spki_hashes:
dynamic_sts_domain: cipherli.st
dynamic_upgrade_mode: STRICT
dynamic_sts_include_subdomains: true
dynamic_sts_observed: 1430474553.53173
dynamic_pkp_domain: cipherli.st
dynamic_pkp_include_subdomains: true
dynamic_pkp_observed: 1430474553.531744
dynamic_spki_hashes: sha256/klO23nT2ehFDXCfx3eHTDRESMz3asj1muO+4aIdjiuY=,sha256/633lt352PKRXbOwf4xSEa1M517scpD3l5f79xMD9r9Q=

Cipherli.st – Strong Ciphers for Apache, nginx and Lighttpd

https cipherli.st

Reader

Cipherli.st Strong Ciphers for Apache, nginx and Lighttpd

Apache

```
SSLCipherSuite AES128+EECDH:AES128+EDH
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder On
SSLSessionTickets Off
Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains; preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
```

Rationale and tutorial on Strong SSL Security on Apache

nginx

```
ssl_ciphers "AES128+EECDH:AES128+EDH";
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_session_cache shared:SSL:10m;
add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
ssl_session_tickets off; # Requires nginx >= 1.5.9
ssl_stapling on; # Requires nginx >= 1.3.7
ssl_stapling_verify on; # Requires nginx => 1.3.7
resolver $DNS-IP-1 $DNS-IP-2 valid=300s;
resolver_timeout 5s;
```

Rationale and tutorial on Strong SSL Security on nginx

Lighttpd

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "AES128+EECDH:AES128+EDH"
ssl.use-compression = "disable"
setenv.add-response-header = (
    "Strict-Transport-Security" => "max-age=63072000; includeSubdomains; preload",
    "X-Frame-Options" => "DENY",
    "X-Content-Type-Options" => "nosniff"
)
ssl.use-sslv2 = "disable"
ssl.use-sslv3 = "disable"
```

Rationale and tutorial on Strong SSL Security on Lighttpd

Configs and testing

- https://cipherli.st
- https://mozilla.github.io/server-side-tls/ssl-config-generator/
- Qualys SSL Server Test: https://www.ssllabs.com/ssltest/

Monitoring

- nagios / icinga
- NewRelic - http://newrelic.com/
- App Enlight - https://appenlight.com
- Opbeat (for Django) - https://opbeat.com/

gay-production – New Relic

https://rpm.newrelic.com/accounts/427144/applications/3247098

New Relic. APM BROWSER SYNTHETICS MOBILE SERVERS PLUGINS INSIGHTS Alerts Beta Tools Help NextGear

Applications Service maps Key transactions Alerts

APPS

TIME PICKER Last 30 minutes ending now

SERVERS All servers (3)

MONITORING
- Overview
- Service maps
- Transactions
- Database
- External services

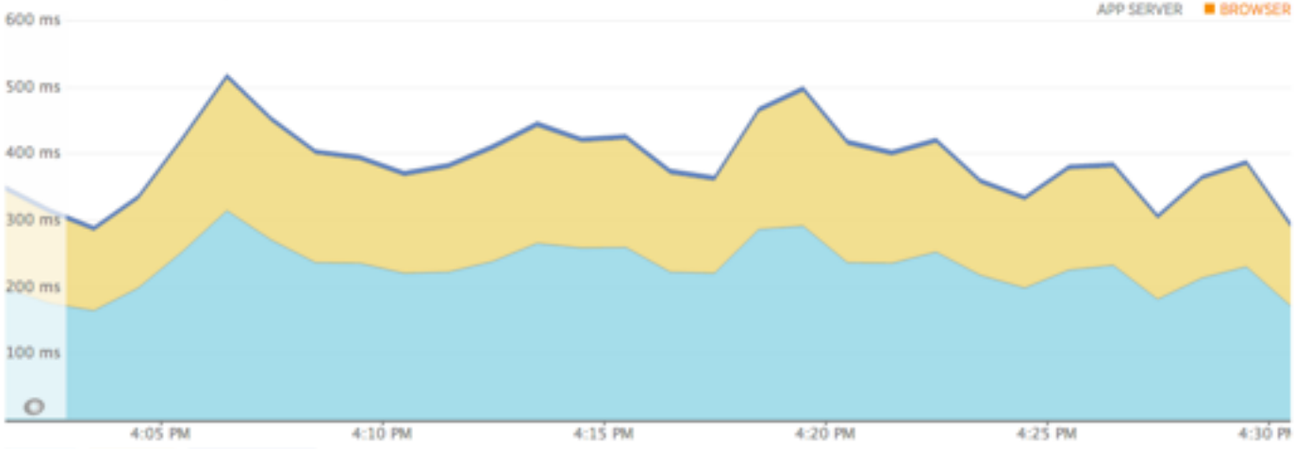
EVENTS
- Errors
- Alerts
- Deployments
- Thread profiler

REPORTS
- SLA
- Availability
- Capacity
- Scalability
- Web transactions
- Database
- Background jobs
- Speed index

SETTINGS
- Application
- Availability monitoring
- Environment

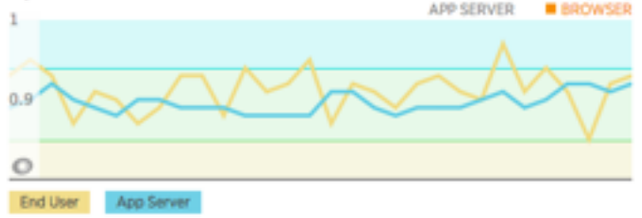
Web transactions response time

389 ms APP SERVER 4.98 s BROWSER



Apdex score

0.9 [0.5] APP SERVER 0.91 [7.0] BROWSER



Throughput

427 rpm AVERAGE

Transactions

	App server time
/hotnot.views:hotornot	6.18 sec
/hotnot.views:get	2,020 ms
/api.views:featured	1,130 ms
/users.views:profile	685 ms
/users.views:home	495 ms

Error rate

0.0234 %

Recent events

Member of policy Application policy

All

Today

End user Apdex score < 0.85
Non-critical problem 10:08

End user Apdex score < 0.7
Critical problem 8:38

End user Apdex score < 0.7
Critical problem 8:15

End user Apdex score < 0.7
Critical problem 4:02

GAY-PRODUCTION

3 servers

Server name	Apdex	Resp. time	Throughput	Error Rate	CPU usage	Memory
python01 8 instances	0.92 0.5	302 ms	256 rpm	0.0260 %	76 %	980 MB
python02 8 instances	0.86 0.5	550 ms	84 rpm	0.0392 %	66 %	750 MB

End user 5.09 s 63.3 rpm | App server 350 ms 411 rpm 0.0812 err%

Transactions – gay-produ

https://rpm.newrelic.com/accounts/427144/applications/3247098/transactions

Jochem

New Relic. APM BROWSER SYNTHETICS MOBILE SERVERS PLUGINS INSIGHTS Alerts Beta Tools Help NextGear

Applications Service maps Key transactions Alerts

APPS

TIME PICKER Last 30 minutes ending now

SERVERS All servers (3)

MONITORING
- Overview
- Service maps
- **Transactions**
- Database
- External services

EVENTS
- Errors
- Alerts
- Deployments
- Thread profiler

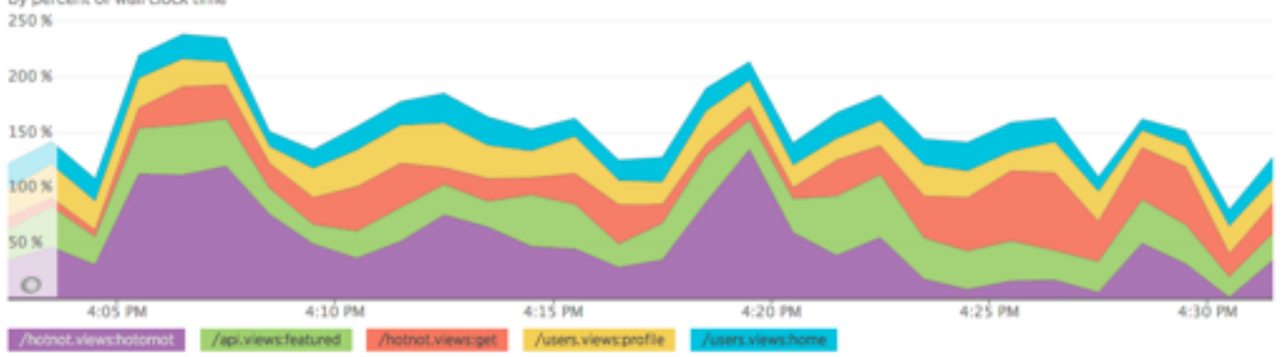
REPORTS
- SLA
- Availability
- Capacity
- Scalability
- Web transactions
- Database
- Background jobs
- Speed index

SETTINGS
- Application
- Availability monitoring
- Environment

Most time consuming

/hotnot.views:hotornot	18.1%
/api.views:featured	12.1%
/hotnot.views:get	10.4%
/users.views:profile	9.07%
/users.views:home	7.33%
/messenger.views:messages_quick_list	6.59%
/api.views:message_list_new	3.71%
/api.views:contacts	2.63%
/search.views:search	2.48%
/content.views:article	2.4%
/api.views:messenger	2.14%
/api.views:version	2.06%
/users.views:visitors	2.01%
/messenger.views:messages_history	1.68%
/users.views:meet_pagination	1.65%
/users.views:meet	1.38%
/users.views:friends	1.36%
/users.views:photos	1.23%
/api.views:nearby	1.19%
/search.views:new	1.1%

Top 5 web transactions
by percent of wall clock time

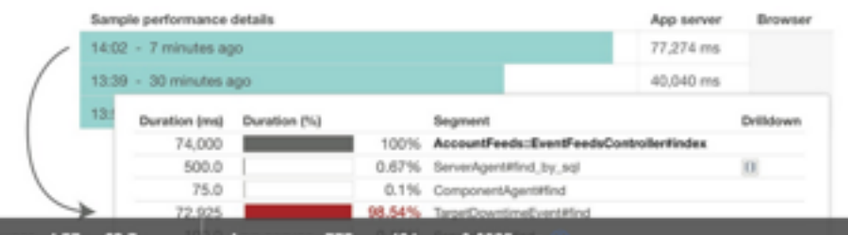


Throughput (rpm)

Recent Slow Transactions Resp. Time

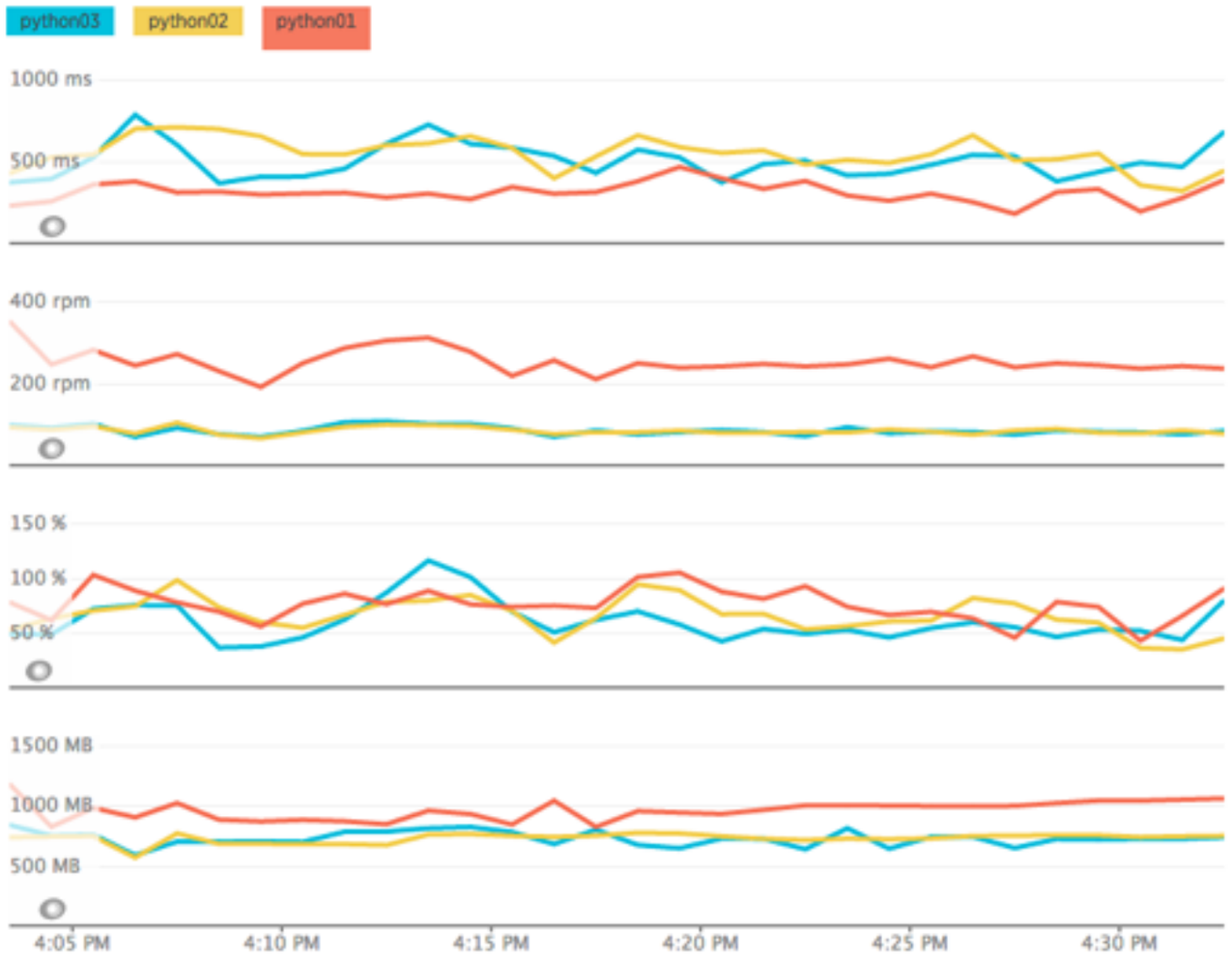
Transaction traces provide deeper information about where slowdowns are occurring in your code, right do wn to a specific method or database query. More about transaction traces »

Upgrade to New Relic Pro now »



End user 4.53 s 62.3 rpm App server 335 ms 404 rpm 0.0825 err%

3 servers



3 servers

Server name	Apdex	Resp. time	Throughput	Error Rate	CPU usage	Memory
python01 8 instances	$0.92_{0.5}$	311 ms	252 rpm	0.0265 %	77 %	990 MB
python02 8 instances	$0.86_{0.5}$	552 ms	83 rpm	0.0399 %	66 %	750 MB
python03 8 instances	$0.86_{0.5}$	510 ms	84 rpm	0.00 %	60 %	760 MB

Possible next steps

- Running a local caching proxy for pypi
- Creating more generic roles for Ansible
- Auto configure failover (VRRP)
- Setup database replication
- Automatically setup backups
- …?

Finale

- **Now is the time for questions!**
- But if you forgot any…

  twitter @jocmeh
  mail jochem@pythonic.be