

HostinGuard

Dashboard per monitoraggio prestazioni VPS/server

Mirko Musumeci @Linux Day Bari 2018

HostinGuard

- Giugno 2016: Inizio dei lavori
- Sito web in WordPress con milioni di visitatori/mese
 - deployato su VPS Linux CentOS
 - improvviso degrado delle prestazioni e delle visite
 - **panico!**

Analisi puntuale delle problematiche

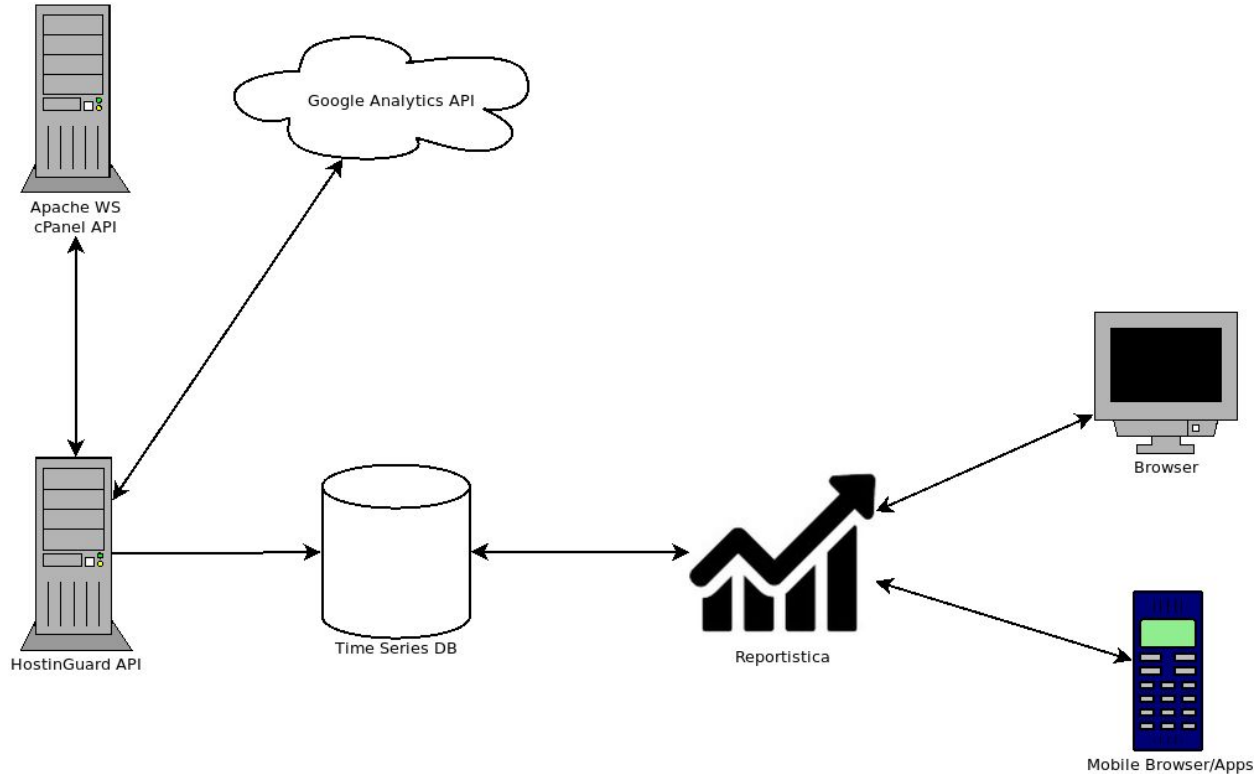
- Plugin cache WP, configurazione non ottimale
 - elevato workload CPU del VPS
 - elevato tempo di risposta del sito
 - perdita di impression

- Uptime non garantito dall'hosting
 - errori 50x
 - negazioni di servizio
 - perdita di impression e danno d'immagine

L'obiettivo e la sfida

- Obiettivo: realizzazione di un ambiente “controllato”
 - pannello di controllo in sala macchine
 - monitoraggio dei KPI cardine (Workload CPU, RAM, TTFB, LOGs, ecc.)
 - interventi tecnici tempestivi => arginamento dei danni economici e d'immagine
- La sfida: superare i limiti imposti dagli hosting provider (es. chroot su VPS)
 - permessi limitati: limiti su installazione di ulteriore software
 - punto di forza: comandi e servizi messi a disposizione da Linux
 - free
 - cat, grep..
 - crontab

L'infrastruttura



Lo stack tecnologico

- Backend
 - Python 3.6
 - Django 2.0.4
 - Django REST Framework 3.7.3
- Time Series Database
 - Elasticsearch 6.3.2
- Frontend Reportistica
 - Grafana 5.2.4

Django & Django REST Framework

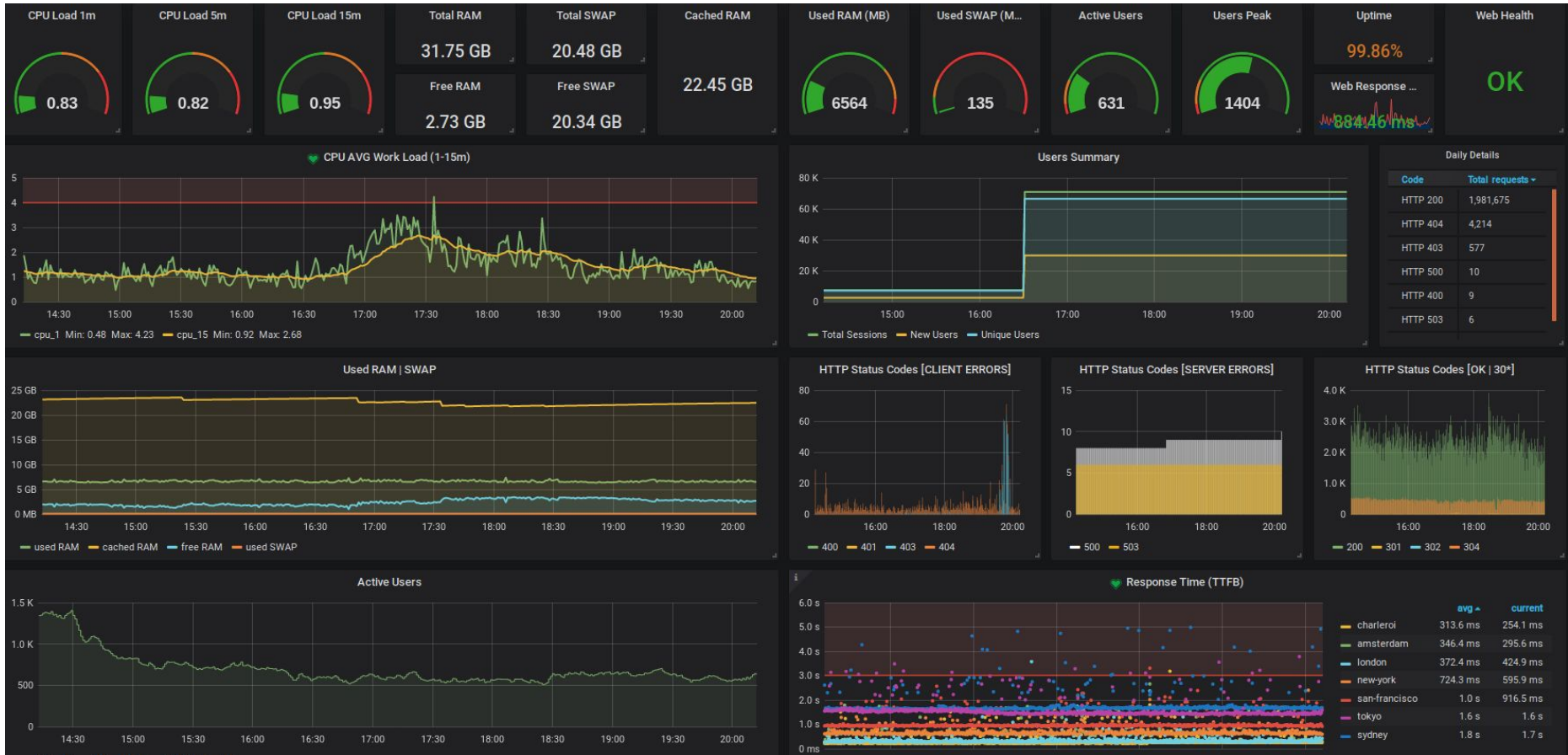
- 2005, Framework per lo sviluppo di applicazioni web (MVC)
- Scritto in linguaggio Python
- Aree di impiego
 - Creazione di blog, webmagazine, ..
 - Toolkit per la realizzazione di REST API (Django REST Framework)
- Licenza Open Source

Elasticsearch

- 2010, Base di dati orientata allo stoccaggio di serie temporali
- $t_0=\{‘a’: 0.1, ‘b’: 0.3\}$; $t_1=\{‘a’: 0.5, ‘b’: 0.2\}$; ... $t_n=\{‘a’: 0.3, ‘b’: 0.6\}$.
- Sviluppato in Java
- Documenti serializzati in JSON
- Interazione via REST API
- Licenza Open Source

Grafana

- Applicazione web per la creazione di dashboard
- Sviluppato in Go language
- Supporta svariati datasource
 - Elasticsearch
 - InfluxDB
 - Graphite
 - DB relazionali
- Ideale per analisi e monitoraggio metriche
- Licenza Open Source



Carico CPU: cPanel/WHM API - *loadavg*

JSON API

```
https://hostname.example.com:2087/cpsess#####/json-api/loadavg?api.version=1
```

Parameters

This function does not accept parameters.

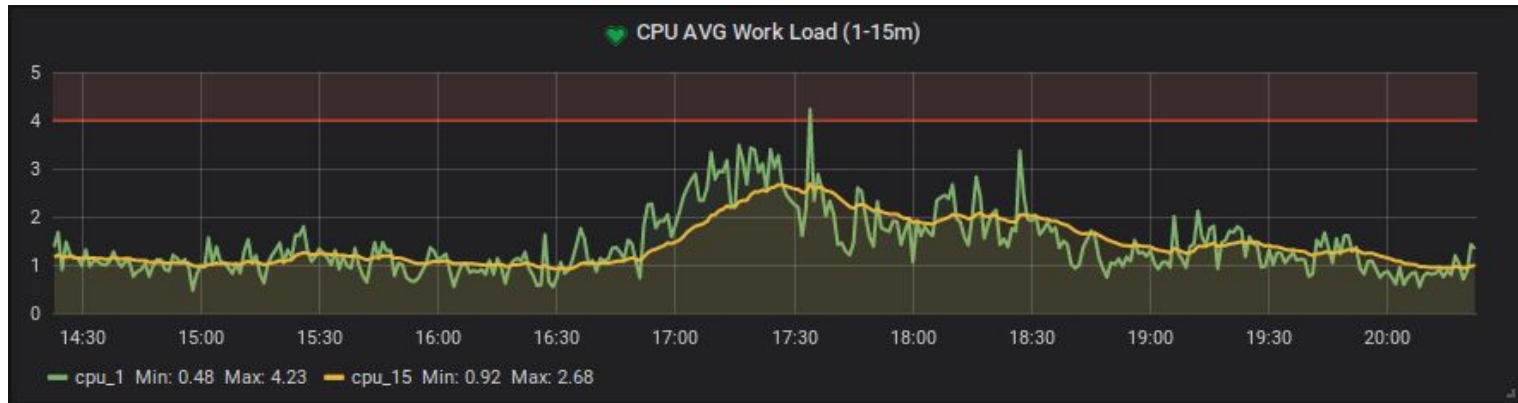
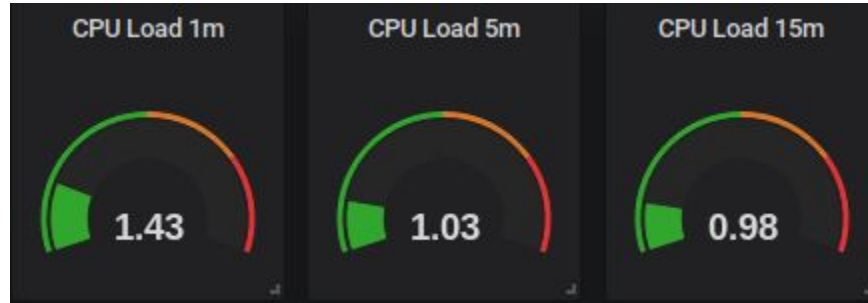
Returns

Return	Type	Description	Possible values	Example
one	<i>string</i>	The server's load average over the previous minute.	A decimal number. For example, a value of .18 represents 18% of a single CPU's processor capacity.	0.17
five	<i>string</i>	The server's load average over the previous five minutes.	A decimal number. For example, a value of .18 represents 18% of a single CPU's processor capacity.	0.18
fifteen	<i>string</i>	The server's load average over the previous fifteen minutes.	A decimal number. For example, a value of .18 represents 18% of a single CPU's processor capacity.	0.19

Carico CPU: cPanel/WHM API - *loadavg* (2)

```
def get_cpanel_data(self) -> dict:
    cpanel_data = {}
    whm = cpanel_client.Client(
        username=self.username,
        host=self.host,
        password=self.password,
        ssl=self.use_ssl
    )
    loadavg = whm.call('loadavg')
    cpanel_data['cpu_1'] = float(loadavg['one'])
    cpanel_data['cpu_5'] = float(loadavg['five'])
    cpanel_data['cpu_15'] = float(loadavg['fifteen'])
    return cpanel_data
```

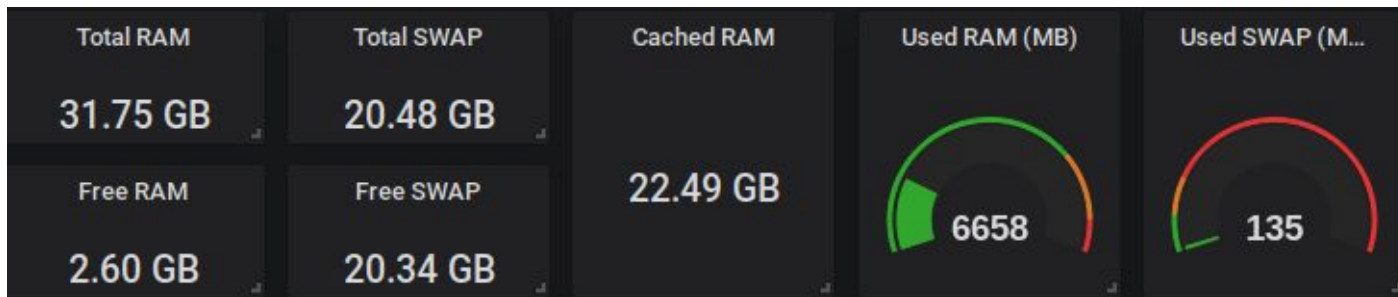
Carico CPU: cPanel/WHM API - *loadavg* (3)



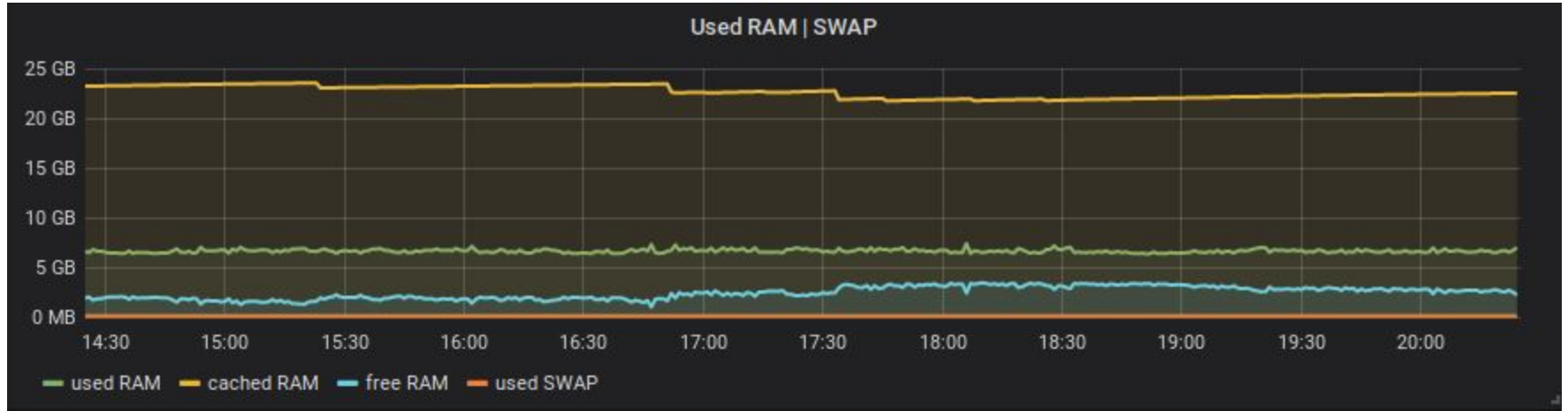
Utilizzo RAM

```
free -m > /home/myUser/kpi/free.out
```

```
Mem: total used free shared buff/cache available
Swap: 20475 146 20329
```



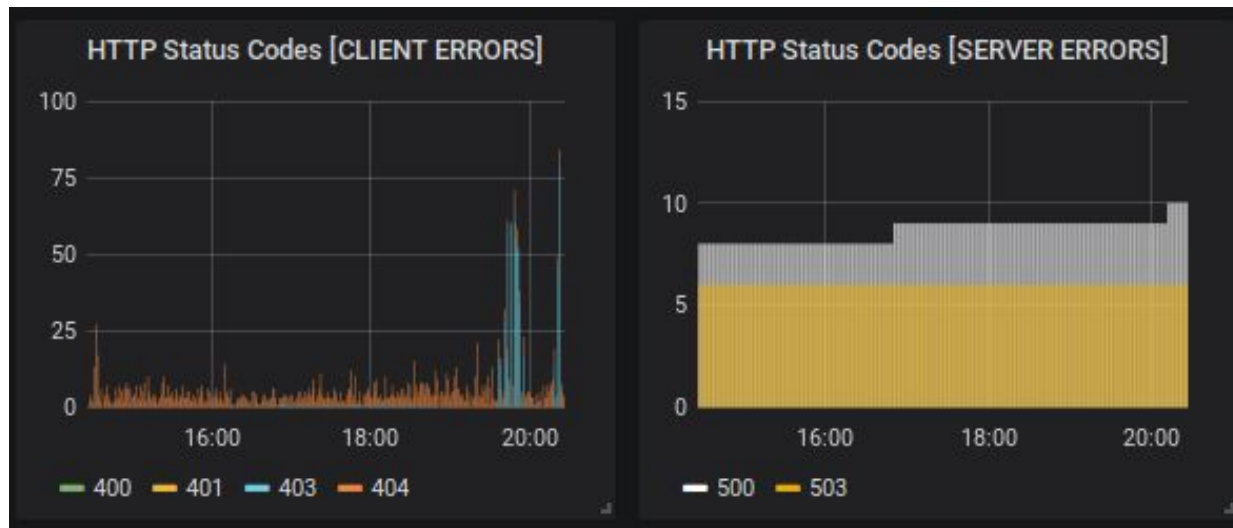
Utilizzo RAM (2)



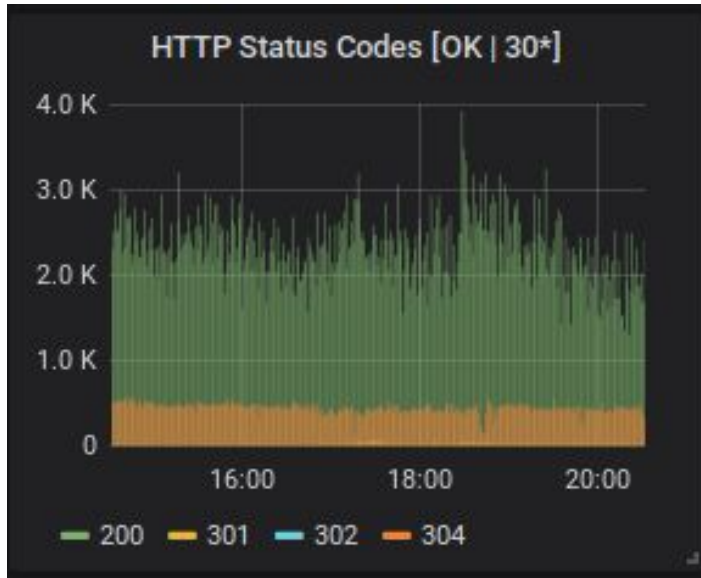
Log di Apache

```
cat /home/myUser/access-logs/mySite-ssl_log | grep "$(date +"%d/%b")" |  
cut -d ' ' -f 9 | sort | uniq -c | sort -nr > /home/myUser/kpi/logs.out
```

```
2123705 200  
19802 301  
4208 404  
281 302  
214 408  
23 403  
10 500  
7 400  
6 503
```



Log di Apache (2)



Daily Details

Code	Total requests ▾
HTTP 200	2,017,552
HTTP 404	4,291
HTTP 403	728
HTTP 500	10
HTTP 400	9
HTTP 503	6

Accessi: Google Analytics API

`ga:users`

UI Name: `Users`

The total number of users for the requested time period.

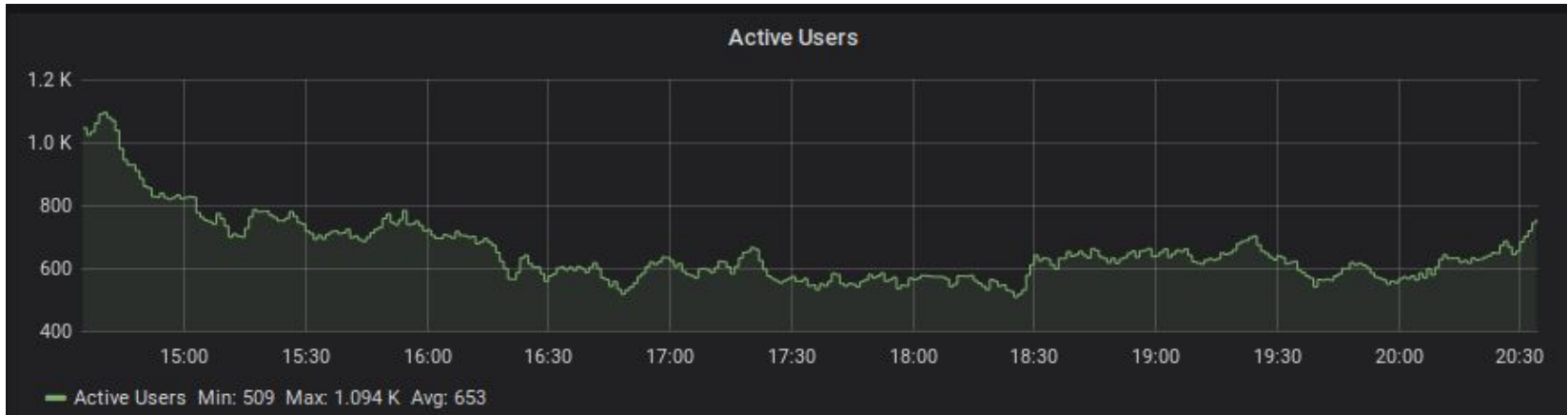
Attributes:

Data Type	Allowed In Segments	Added in API version
INTEGER	No	3

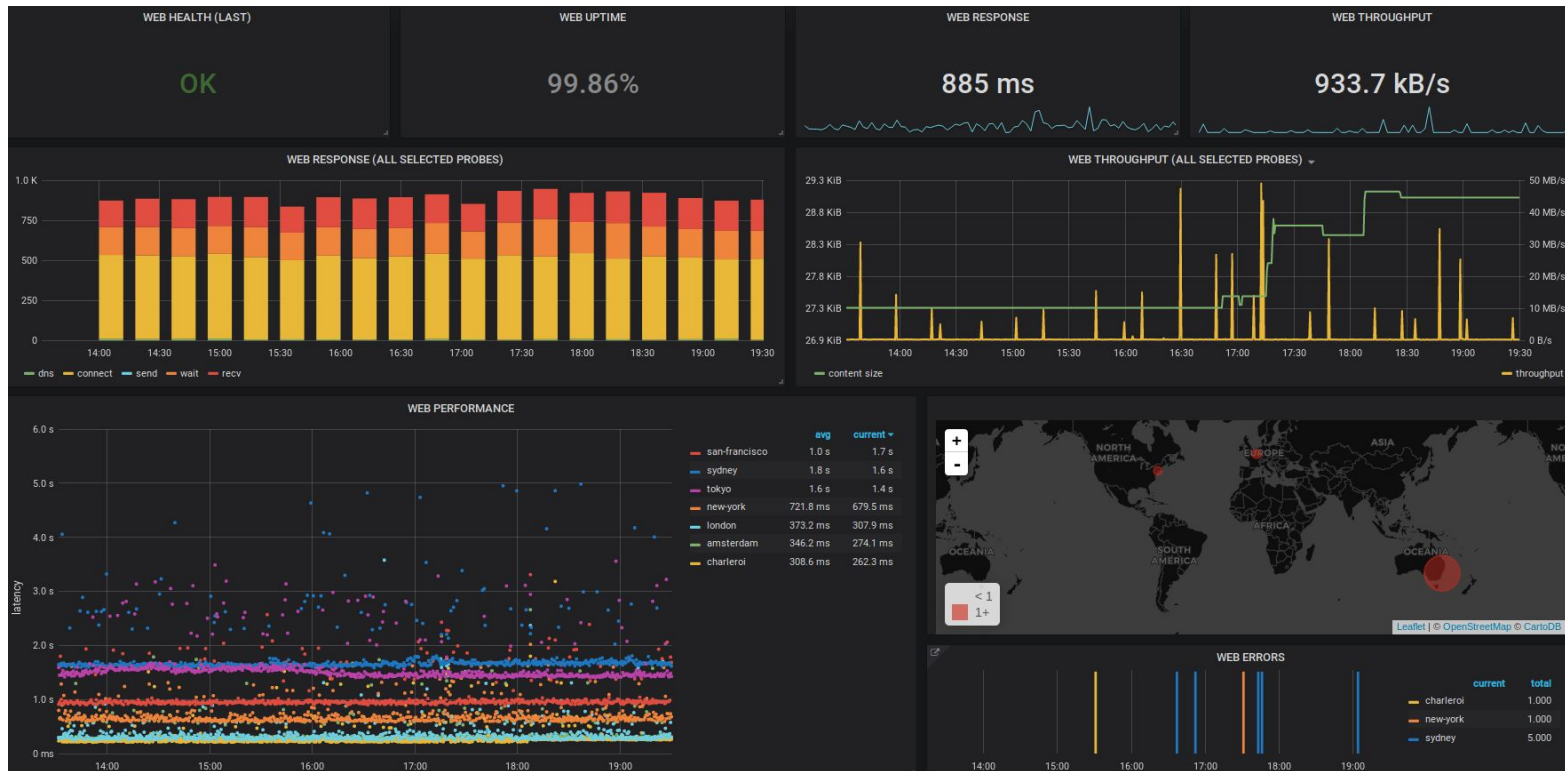
ed anche:

- `ga:sessions`
- `ga:newUsers`
- `rt:activeUsers`
- ...

Accessi: Google Analytics API (2)



Tempi di risposta: worldPing



Allarmi e notifiche

- Configurabili da Grafana
 - E-mail
 - Slack



Interventi di miglioramento

- (!) Layer di autenticazione
- Esternalizzazione della configurazione (es. ETCD)
- Introduzione di ulteriori KPI (es. uso disco)

Riferimenti

- Django Project: <https://www.djangoproject.com>
- Elasticsearch: <https://www.elastic.co/products/elasticsearch>
- Grafana: <https://grafana.com>

Mirko Musumeci

Software Engineer

<https://github.com/mirkochip>

<https://studiomusumeci.com>

@mirkochip84

Extras

