


---

# Devops

# Nuovi modi di

# gestire i server



The poster features the Linux penguin mascot in the center, surrounded by various icons representing technology and development: a CD, a lightbulb, gears, a microchip, a cloud with Wi-Fi signal, a document, and a laptop. In the top left corner is the BGLUG logo (Bergamo Linux User Group). In the top right corner is the logo of the Comune di Bergamo, with the text 'Con il patrocinio di' above it.

**LINUX DAY**

Giornata nazionale a favore della diffusione del software libero e del sistema operativo GNU/Linux

**SABATO 24 OTTOBRE 2015**

**VIA DEL POLARESCO, 15, 24129 BERGAMO**

INGRESSO LIBERO. Per info e prenotazioni visita <http://bglug.it/linuxday>

---

~~Mattia Rossi~~ Francesco Vigotti • 24.10.2015 - !! linuxday !! - bergamo

---

# Devops



## Dev + IT Operations

Sviluppo & tecnologia

....

Software automation

- Automated configuration
- Automated deployment

Agile principles

Tools ( docker, jenkins, vagrant, **ansible** )

---

---

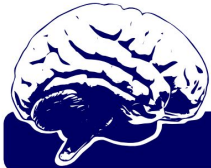
# Qualche domanda...



Raise your hand if you  
didn't get enough  
sleep last night



SpamSpamSpamSpamSpam



**FOSDEM**

Free and Open Source Software  
Developers' European Meeting



*Codelovers*

Bergamo Developers User Group



ANSIBLE

---

# Ansible

## Provisioning

- Server Setup
- Install software

## App Deployment

- The delivery part

## Configuration Management

- Configure everything (app/cluster)

## Continuous Delivery

- When jenkins trigger ansible / ansible-pull

---

SPOILER --->  
(don't watch that logo)



---

# Prima di ansible

- Manual updates + logbook + manual deploy (scp)
- logbook + bash + manual deploy (scp)
- logbook + bash + bash (rsync lsyncd )
- logbook + bash + bash (rsync lsyncd ) + python ( fabric )
- Ansible
- ..... the future



---

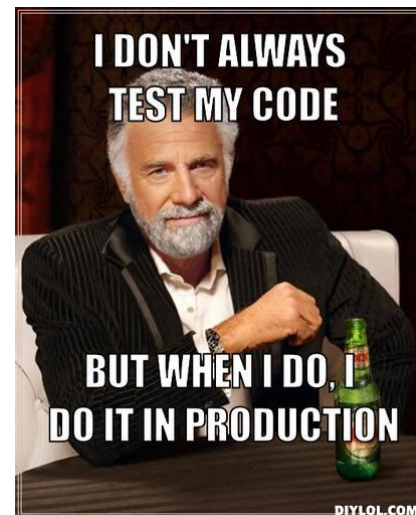
# Perchè automatizzare

## Velocizzare il workflow

- Inserire piu controlli
- Iterazioni piu brevi
  - Diversi stadi di testing , piu o meno isolati

## Riduzione errori

- Errore umano
- Errore di automation





---

# Ansible vs alternatives



---

# Ok, but why Ansible?

## Semplicità

- YAML
- Python

## Requisiti

- Agent-less
- SSH

## Open source

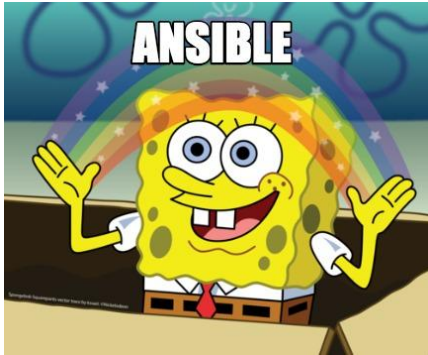
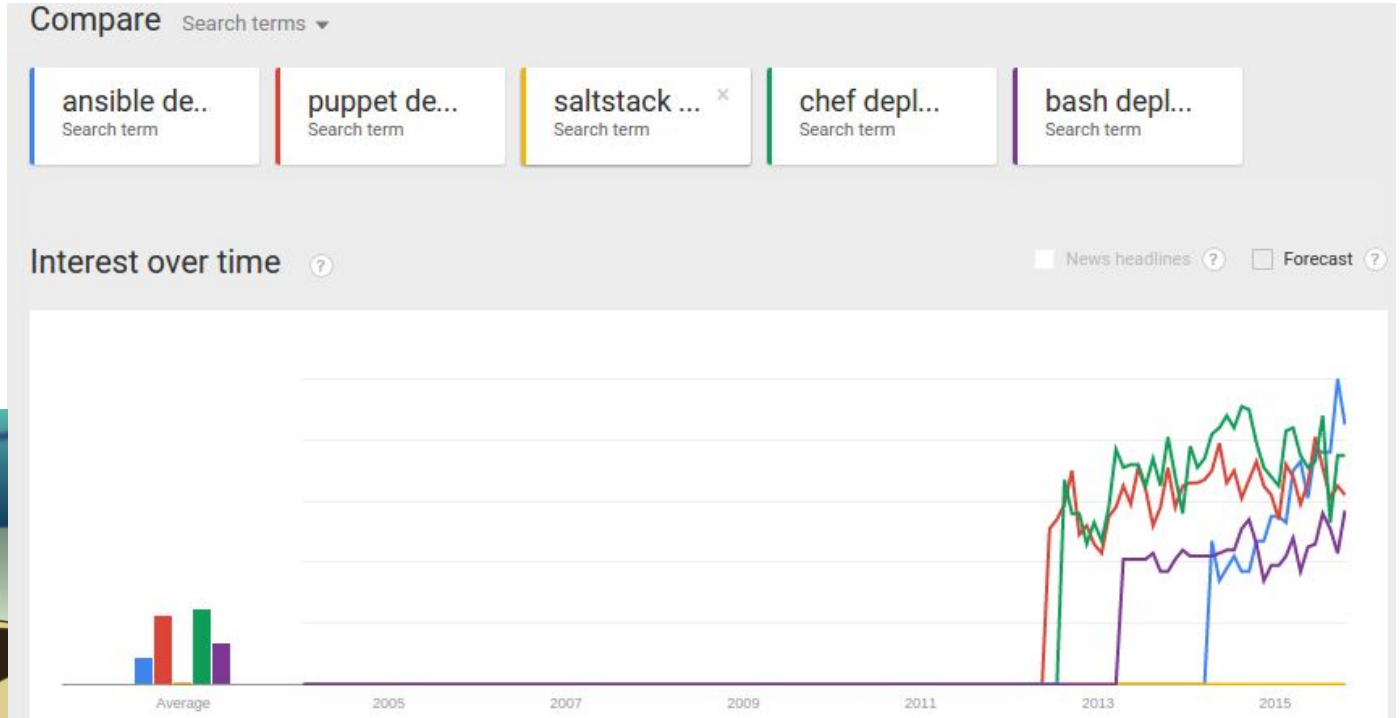
- Free & Open-source
- [github.com/ansible/ansible](https://github.com/ansible/ansible)

## Giuste convenzioni

- Idempotente
- Autodocumentato
- Gerarchie e dinamismo (quanto basta)



# Il piu recente e gia popolare



---

# Logica di funzionamento

## Cosa

- Applicazione python

## Dove

- Dal tuo computer
- Dalla destinazione (pull)
- Dal server di CI

## Come

- Direttamente da linea di comando (semplici task)
- Files / Repository di configurazione (configurazioni piu complesse)

## Quando

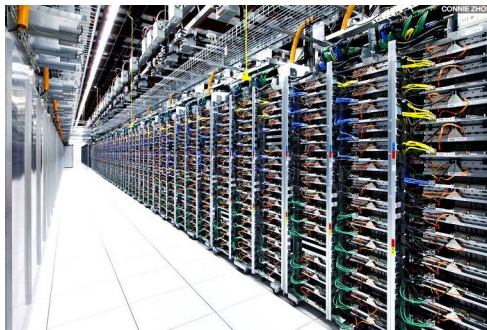
- Quando vuoi
  - Dopo i test
  - Cron
-

---

# Componenti : Inventario

## Ini file

- default: /etc/ansible/hosts
- env: ANSIBLE\_HOSTS
- inline: with “-i”



## Syntax

[backend]

host0.example.org ansible\_ssh\_host=192.168.33.10 ansible\_ssh\_user=root

host1.example.org ansible\_ssh\_host=192.168.33.11 ansible\_ssh\_user=root

[database]

host2.example.org

[revproxies]

www[01:50].example.com

[us:children]

backend

database

[us]

www[01:40].example.com

[eu]

www[41:50].example.com

---

# Componenti : Moduli

## Task predefiniti

- 300+ moduli [ [docs.ansible.com/ansible/modules\\_by\\_category.html](https://docs.ansible.com/ansible/modules_by_category.html) ]
- Package manager, repositories management, download , git , ..
- Os configuration , hostname, dns , accounts, ssh keys, iptables/ufw , ...
- App management ( helpers for : mysql, postgres, apache , nginx , zookeeper , ...)
- Ping, wait\_for, fail, debug, assert, pause



## Custom

High power tasks : command, raw, script, shell

Custom modules ( interfacce gia predisposte :  
key=value IN , json OUT )

[http://docs.ansible.com/ansible/developing\\_modules.html](http://docs.ansible.com/ansible/developing_modules.html) )

```
#!/usr/bin/python
```

```
import datetime  
import json
```

```
date = str(datetime.datetime.now())  
print json.dumps({  
    "time" : date  
})
```

---

# Componenti : Moduli > esempi

1. - copy: src=/srv/myfiles/foo.conf dest=/etc/foo.conf owner=foo group=foo mode=0644
  2. - apt: name=openjdk-6-jdk state=latest install\_recommends=no
  3. - git: repo=git://foosball.example.org/path/to/repo.git dest=/srv/checkout update=no
  4. - htpasswd: path=/etc/mail/passwords name=alex password=oedu2eGh crypt\_scheme=md5\_crypt
  5. - mysql\_user: name=root password={{ mysqlpass }} login\_unix\_socket=/var/run/mysqld/mysqld.sock
  6. - mysql\_user: name=bob password=12345 priv=\*.\*:USAGE state=present
  7. - get\_url: url=http://example.com/path/file.conf dest=/etc/foo.conf headers='key:value,key:value'
-

---

# Componenti : Moduli > esempi



- command: `/sbin/shutdown -t now`

---



---

# Componenti : Variabili

## Scope

- Global
- Play
- Host

## WARNING

!! Precedence caos !!  
( use conventions )

## Declaration entry points

- extra vars ( -e in command line )
- playbook
- external files
- group\_vars / hosts\_vars dirs
- roles defaults
- runtime defined
- ..... ( specialmente in v2 )



---

# Componenti : Variabili > esempi

## YAML syntax

```
person: Jane Jolie
```

```
filename: v1.j2
```

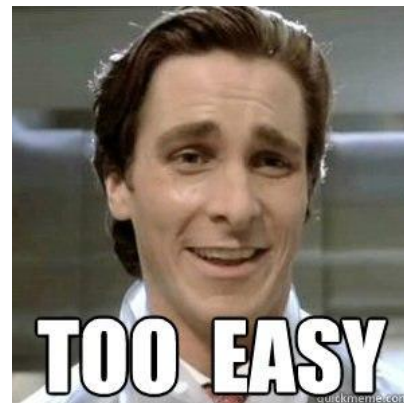
```
roles:
```

```
- { role: app_user, name: Ian }
```

```
- { role: app_user, name: Terry }
```

```
- { role: app_user, name: Graham }
```

```
- { role: app_user, name: John }
```



---

# &YAML > | \*YAML

---  
receipt: Oz-Ware Purchase Invoice

date: 2012-08-06

customer:

given: Dorothy

family: Gale

items:

- part\_no: A4786

descrip: Water Bucket (Filled)

price: 1.47

quantity: 4

- part\_no: E1628

descrip: High Heeled "Ruby" Slippers

size: 8

price: 100.27

quantity: 1

bill-to: &id001

street: |

123 Tornado Alley

Suite 16

city: East Centerville

state: KS

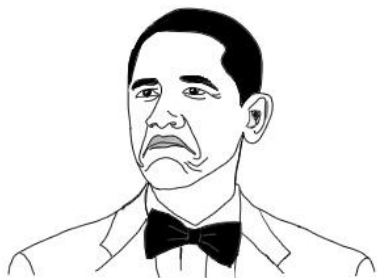
ship-to: \*id001

specialDelivery: >

Follow the Yellow Brick  
Road to the Emerald City.

Pay no attention to the  
man behind the curtain.

...



**NOT BAD**

---

---

# Componenti : Variabili > Filtri

JINJA2 syntax ( [jinja.pocoo.org/docs/dev/](http://jinja.pocoo.org/docs/dev/) )

- `foo: {{ some_variable | default(5) }}`
- `{{ list1 | min }}`
- `{{ list1 | union(list2) }}`
- `{{ 59 | random }} * * * * root /script/from/cron`
- `{{ 100 | random(step=10) }} => 70`
- `{{ ['a','b','c']|shuffle }} => ['c','a','b']`
- `{{ myvar | pow(2) }}`
- `{{ '192.0.2.1/24' | ipaddr('address') }}`
- `{{ 'test1'|hash('sha1') }}`
- `{{ {'a':1, 'b':2}|combine({'b':3}) }}`
- `{{ string_value | quote }}`
- `{{ path | basename }}`
- `{{ ansible_distribution_version | version_compare('12.04', '>=') }}`
- `{{ encoded | b64decode }}`
- `{{ hostname | to_uuid }}`
- `{{ 'ansible' | regex_replace('^a.*(.)$', 'a\\1') }}`
- `local_home : {{ lookup('env','HOME') | default(lookup('env','SECOND_HOME')| default('home')) }}`



---

# Componenti : Variabili > Interpolazione

JINJA2 syntax ([jinja.pocoo.org/docs/dev/](http://jinja.pocoo.org/docs/dev/))

## Template module:

- template: src=/mytemplates/foo.j2 dest=/etc/file.conf owner=bin group=wheel mode=0644



---

# Componenti : Variabili > Interpolazione

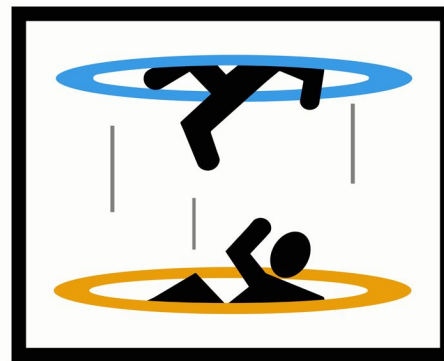
JINJA2 syntax ([jinja.pocoo.org/docs/dev/](http://jinja.pocoo.org/docs/dev/))

```
{{ ansible_eth0["ipv4"]["address"] }}
```

```
{% if 'webserver' in group_names %}
  # some part of a configuration file that only applies
  # to webserver
{% endif %}
```

```
{% for host in groups['app_servers'] %}
  {{ hostvars[host]['ansible_eth0']['ipv4']['address'] }}
{% endfor %}
```

```
{% if loop.index is divisibleby 3 %}
{% endif %}
```



**CAUTION**  
**INFINITE LOOP**

---

---

# Componenti : Variabili > Interpolazione

JINJA2 syntax ( [jinja.pocoo.org/docs/dev/](http://jinja.pocoo.org/docs/dev/) )

```
{% for host in groups['zk_clients']|sort %}
# host {{ host }}
{% set default_ip = "127.0.0.1" -%}
{% if hostvars[host]['ansible_default_ipv4'] is defined %}
    {% set default_ip = hostvars[host]['ansible_default_ipv4']['address'] %}
{% else %}
    {% set default_ip = hostvars[host]['public_ip']|default("127.0.0.1") %}
{% endif %}
```

```
-p tcp -s {{ hostvars[host].zkserver_connection_ip | default(hostvars[host].public_ip) | default(hostvars[host]['ansible_ssh_host']) | default( default_ip ) }} -
m multiport --dports {{ zkserver_hostbind_client_PORT }},{{ zkserver_hostbind_peer_PORT }},{{ zkserver_hostbind_election_PORT }} -j ACCEPT
{% endfor %}
```

```
mqa.host.list={% for node in groups['nodes'] %}{{ node }}:5672{% if not loop.last %},{% endif %}{% endfor %}
```



# Strategies

- Playbook
- Roles
- Variables

---





You are doing it right.

# Roles: organize , reuse , share

## Directory structure:

```
rolename/
  files/
  templates/
  tasks/
    main.yml
  handlers/
  vars/
    main.yml
  defaults/
    main.yml
  meta/
    main.yml
```

- rolename/tasks/main.yml

---

- name: Download Packer

get\_url: >

```
url=https://dl.bintray.com/mitchellh/packer/packer_{{ packer_version }}_linux_amd64.zip
dest=/usr/local/src/packer_{{ packer_version }}_linux_amd64.zip
```

- name: Extract and install Packer

```
unarchive: src=/usr/local/src/packer_{{ packer_version }}_linux_amd64.zip
           dest=/usr/local/bin
           copy=no
```

---

# Playbook:



```
---
- hosts: all
  gather_facts: yes
  remote_user: craun
  serial: "25%"
  sudo: yes
  vars_files:
    - "{{ PRIVATE_FILE_LOCATION }}/vars.yml"
  pre_tasks:
    - name: upgrade packages
      apt: upgrade=yes update_cache=yes
      cache_valid_time=300
      sudo: yes
  roles:
    - { role: brightbox_ruby, ruby_version: 2.2.2 }
    - { role: deploy_code,
        git_branch: master,
        repo: 'https://github.com/calebwoods/myapp',
        ruby_env_template: 'templates/production_env.rb' , tags: ["deploy", "
backend"] }
    - { role: unicorn }
    - { role: nginx }
    - { role: sidekiq, apps_dir: '/var/www' , tags: ["deploy", "frontend"] }
  post_tasks:
    - name: setup log rotation
      sudo: yes
      template: src=templates/logrotate.conf dest=/etc/logrotate.d/myapp.conf
```

---

---

## Run:

```
ansible-playbook -i production site.yml --tags ntp --limit boston[0-10]
```

## More tools:

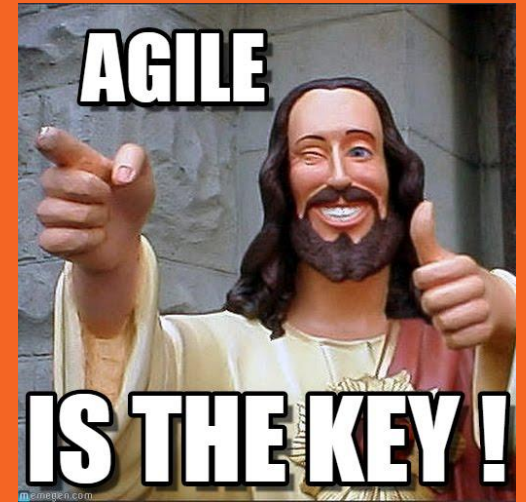
- Vault
- ansible-pull
- Dynamic inventory
- Ansible tower



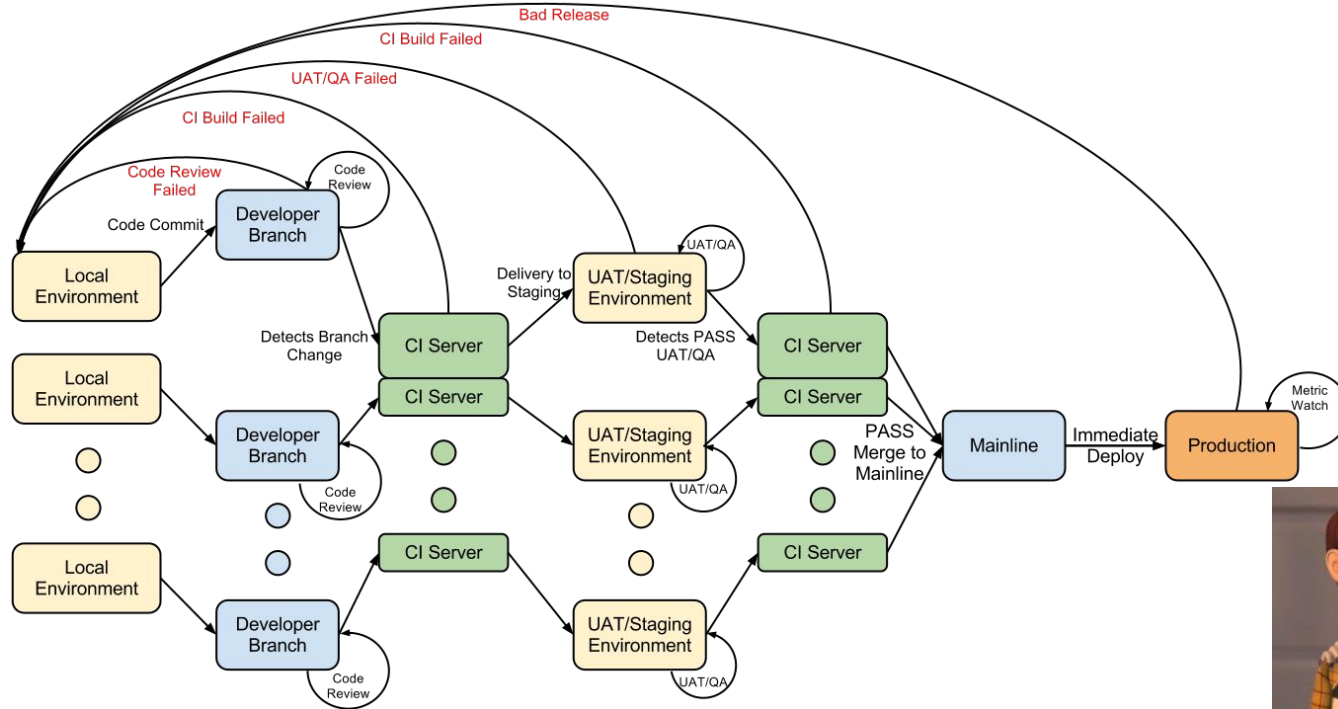
---

# The future

( IS NOW )



# Continuous delivery model

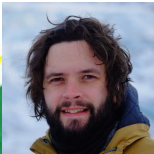


# Microservices

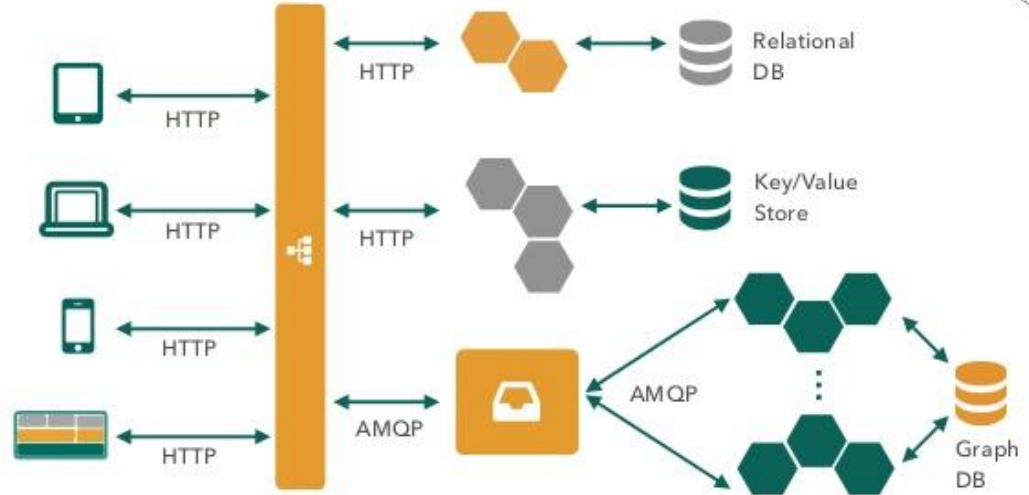


Italian AgileDays

6/7 Novembre  
2015



## Microservice Architecture



# New solutions require more solutions!

Infrastructure as service (IaaS)

Scalability

Stability

Continuous deployment

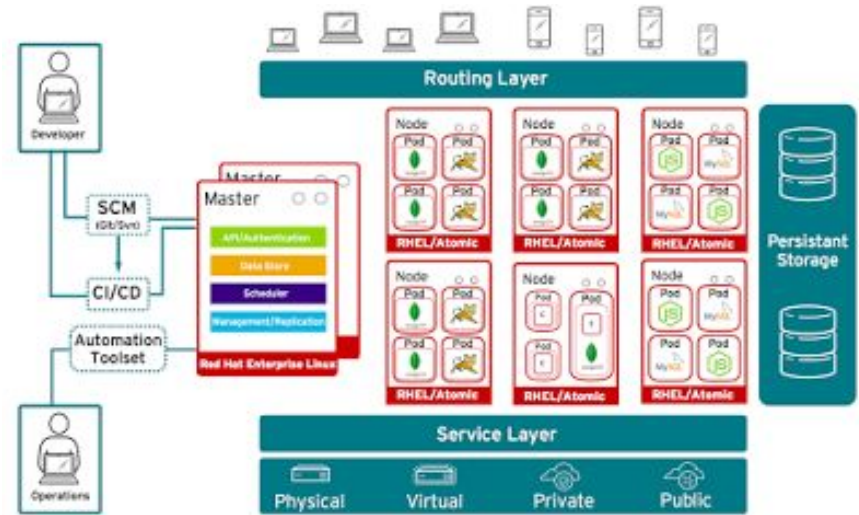
Better Tested Code ( Continuous integration / better defined services boundaries )

Incremental updates ( concurrent versioning )

Monitoring

Logging

Dynamic configuration





# Logging (logstash - elastic search - kibana)

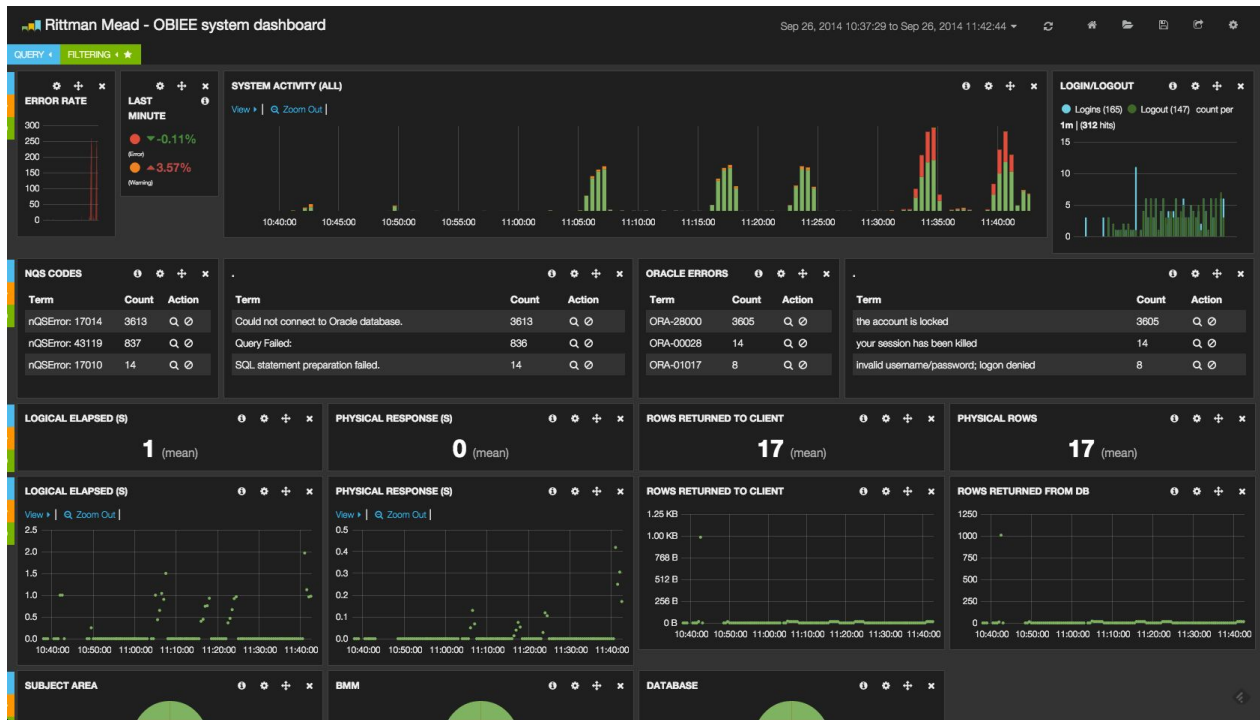
The screenshot displays the Kibana logging interface with the following components:

- Search Bar:** Shows a search query and 16 hits.
- Columns:** A list of fields including @fields.client, @fields.in-reply-to, @fields.retweeted, @fields.urls, @fields.user, @message, @source, @source\_host, @source\_path, @tags, @timestamp, and @type.
- EVENTS:** A bar chart showing event counts over time, with a query filter for 'count per 30s [348584]'. The chart shows a peak in activity around 14:00:00.
- PROCESS TIME (MIN):** Two bar charts showing process time in minutes. The left chart is for 'time\_processed min per 30s [290628]' and the right is for 'time\_processed min per 30s [290628]'. Both show activity over time.
- LOGS:** A table of log entries with columns for host, date\_time, request\_url, response\_code, time\_processed, and src\_ip. The table shows 0 to 100 of 10000 available for paging.
- Message List:** A list of log messages under the '@message' field, including tweets and announcements about CouchDB and PouchDB.



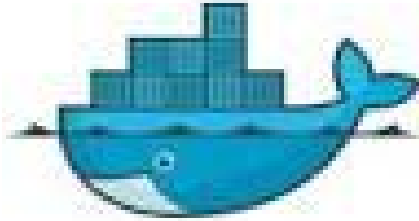


# Monitoring (Grafana - Graphite and InfluxDB)

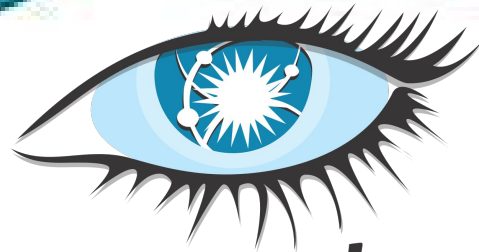
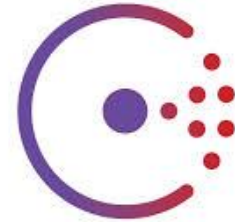


---

Dynamic Dynamic Dynamic, and flexible



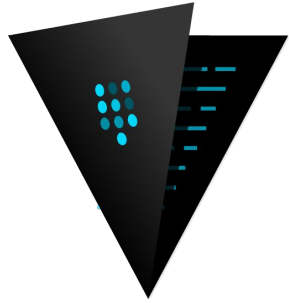
docker



*cassandra*



kafka



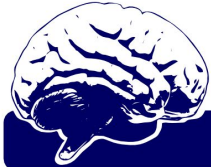
**amazon**  
web services™



MESOS

---

SpamSpamSpamSpamSpam



**FOSDEM**

Free and Open Source Software  
Developers' European Meeting



*Codelovers*

Bergamo Developers User Group