Sistemi di *monitoring*, *logging* e *alerting* moderni

Ovvero di come non uscirne pazzi

M. Finelli
BioDec

BIODEC
Evolving ICT Infrastructures

Indice

Indice

BIODEC
Evolving ICT Infrastructures

Cloud + Virtualizzazione = Infrastruttura distribuita

GRAZIE AI SISTEMI CLOUD E ALLA VIRTUALIZZAZIONE, nei prossimi anni sarà necessario dotarsi di strumenti efficaci per gestire la complessità di un'**infrastruttura distribuita**.

O **rinunciare** alla gestione della medesima.

BIODEC
Evolving ICT Infrastructures

Cloud + Virtualizzazione = Infrastruttura distribuita

GRAZIE AI SISTEMI CLOUD E ALLA VIRTUALIZZAZIONE, nei prossimi anni sarà necessario dotarsi di strumenti efficaci per gestire la complessità di un'**infrastruttura distribuita**.
O **rinunciare** alla gestione della medesima.

BIODEC
Evolving ICT Infrastructures

Non solo distribuita, ma anche numerosa

Un secondo effetto è che l'infrastruttura sarà **molto maggiore** di quella a cui è abituato l'IT tradizionale, e certi problemi di gestione si presentano solo su grande scala.

O per meglio dire: sapere quello che accade nel **piccolo** non è un buon metro per capire cosa accade nel **grande**.

BIODEC
Evolving ICT Infrastructures

Non solo distribuita, ma anche numerosa

Un secondo effetto è che l'infrastruttura sarà **molto maggiore** di quella a cui è abituato l'IT tradizionale, e certi problemi di gestione si presentano solo su grande scala.
O per meglio dire: sapere quello che accade nel **piccolo** non è un buon metro per capire cosa accade nel **grande**.

Indice

L'inizio

Tutto incomincia con . . . Patrick Debois che nel 2007 si trova a svolgere un lavoro "ibrido" di sviluppo e di sistemi, e non è contento di come sta procedendo.

Agile 2008 Andrew Shafer parla di *"Agile Infrastructure"*, o per meglio dire "monologa".

23 giugno 2009 John Allspaw presenta il talk *"10+ deploys per day: Dev & Ops cooperation at Flickr"*.

30-31 ottobre 2009 Il primo *DevOps Days* a Ghent, in Belgio. Grande successo di pubblico e di critica.

*da una presentazione di Damon Edwards pubblicata su IT Revolutions.

L'inizio

Tutto incomincia con . . . Patrick Debois che nel 2007 si trova a svolgere un lavoro "ibrido" di sviluppo e di sistemi, e non è contento di come sta procedendo.

Agile 2008 Andrew Shafer parla di *"Agile Infrastructure"*, o per meglio dire "monologa".

23 giugno 2009 John Allspaw presenta il talk *"10+ deploys per day: Dev & Ops cooperation at Flickr"*.

30-31 ottobre 2009 Il primo *DevOps Days* a Ghent, in Belgio. Grande successo di pubblico e di critica.

*da una presentazione di Damon Edwards pubblicata su IT Revolutions.

BIODEC Evolving ICT Infrastructures

L'inizio

Tutto incomincia con . . . Patrick Debois che nel 2007 si trova a svolgere un lavoro "ibrido" di sviluppo e di sistemi, e non è contento di come sta procedendo.

Agile 2008 Andrew Shafer parla di *"Agile Infrastructure"*, o per meglio dire "monologa".

23 giugno 2009 John Allspaw presenta il talk *"10+ deploys per day: Dev & Ops cooperation at Flickr"*.

30-31 ottobre 2009 Il primo *DevOps Days* a Ghent, in Belgio. Grande successo di pubblico e di critica.

*da una presentazione di Damon Edwards pubblicata su IT Revolutions.

L'inizio

Tutto incomincia con . . . Patrick Debois che nel 2007 si trova a svolgere un lavoro "ibrido" di sviluppo e di sistemi, e non è contento di come sta procedendo.

Agile 2008 Andrew Shafer parla di *"Agile Infrastructure"*, o per meglio dire "monologa".

23 giugno 2009 John Allspaw presenta il talk *"10+ deploys per day: Dev & Ops cooperation at Flickr"*.

30-31 ottobre 2009 Il primo *DevOps Days* a Ghent, in Belgio. Grande successo di pubblico e di critica.

*da una presentazione di Damon Edwards pubblicata su IT Revolutions.

L'inizio

#DevOps diventa un tema caldo in numerose conferenze: viene creato un formato, i DevOps Days, che in pochi anni si replicano per decine di volte in tutto il mondo.
Si enfatizza il tema di come funziona l'IT e di come dovrebbe invece funzionare. Il discorso verte sugli strumenti, su quali funzionano e su quali no, sulle *best practices* e sulle tecniche.

Il movimento oggi

NEL MARZO DEL 2011 anche Gartner si accorge del movimento e pubblica il suo oroscopo: *"The Rise of a New IT Operations Support Model"* che prevede che per il **2015** il movimento sarà passato, da una nicchia nell'ambito *cloud*, all'**adozione nel 20% delle imprese *Global 2000***.

A parte le chiacchiere: si afferma a livello globale l'esistenza di un movimento *from practitioners, to practitioners*.

Il movimento in Italia

Nell'**ottobre 2012**, c'è stata la prima edizione dei DevOps Days italiana, a Roma, con quasi duecento partecipanti da tutto il mondo.
Nel **febbraio 2013**, a Firenze, si è tenuto il primo "Incontro DevOps Italia", con 80+ persone presenti.
Nel **febbraio 2014**, a Bologna, si è tenuto il secondo "Incontro DevOps Italia", con 120+ persone presenti.

Cosa c'è dentro . . .

LA MIA PERSONALE impressione è che sia ancora un *pot-pourri* di teorie, tecniche e pratiche proveniente da ambiti differenti:

1. il movimento *agile*,
2. le *lean methodologies*,
3. le caratteristiche delle comunità *free software* (apertura, condivisione, codice aperto, standard).

Cosa c'è dentro . . .

LA MIA PERSONALE impressione è che sia ancora un *pot-pourri* di teorie, tecniche e pratiche proveniente da ambiti differenti:

1. il movimento *agile*,
2. le *lean methodologies*,
3. le caratteristiche delle comunità *free software* (apertura, condivisione, codice aperto, standard).

Cosa c'è dentro . . .

LA MIA PERSONALE impressione è che sia ancora un *pot-pourri* di teorie, tecniche e pratiche proveniente da ambiti differenti:

1. il movimento *agile*,
2. le *lean methodologies*,
3. le caratteristiche delle comunità *free software* (apertura, condivisione, codice aperto, standard).

. . . e cosa rimane fuori

Si fa prima a dire che cosa non sia DevOps:

- non è una certificazione,
- non è un titolo,
- non è strumento specifico o un software particolare.

Indice

Uno slogan: CAMS

C *culture*

A *automate*

M *measure*

S *share*

Uno slogan: CAMS

C *culture*

A *automate*

M *measure*

S *share*

Uno slogan: CAMS

C *culture*

A *automate*

M *measure*

S *share*

Uno slogan: CAMS

C *culture*

A *automate*

M *measure*

S *share*

Uno slogan: CAMS

C *culture*

A *automate*

M *measure*

S *share*

Culture

1 CREARE UNA CULTURA della collaborazione. È il primo dettame, ma è sovente il più negletto — anche perché è il più difficile da mettere in pratica.

> *People and process first. If you don't have culture, all automation attempts will be fruitless. (John Willis)*

Automate

2 AUTOMATIZZARE ogni azione. Se un'azione manuale può essere svolta da un programma, che lo si scriva. E lo si scriva secondo i crismi con cui si scrivono i programmi: il fatto che sia un programma per i sistemi (o per i *server*) non è un'offesa. "Sistemista" non è un'offesa.

Automate

2 AUTOMATIZZARE ogni azione. Se un'azione manuale può essere svolta da un programma, che lo si scriva. E lo si scriva secondo i crismi con cui si scrivono i programmi: il fatto che sia un programma per i sistemi (o per i *server*) non è un'offesa.

"Sistemista" non è un'offesa.

Automate

2 AUTOMATIZZARE ogni azione. Se un'azione manuale può essere svolta da un programma, che lo si scriva. E lo si scriva secondo i crismi con cui si scrivono i programmi: il fatto che sia un programma per i sistemi (o per i *server*) non è un'offesa. "Sistemista" non è un'offesa.

Corollario: *Infrastructure as code*

Se solo il codice definisce i componenti dell'infrastruttura significa che questa **non puó** essere determinata da:

1. configurazioni manuali,
2. cose che si *cliccano* di qua e di là,
3. persone (*a.k.a.* consulenti) che arrivano e fanno cose.

Corollario: *Infrastructure as code*

Se solo il codice definisce i componenti dell'infrastruttura significa che questa **non puó** essere determinata da:

1. configurazioni manuali,
2. cose che si *cliccano* di qua e di là,
3. persone (*a.k.a.* consulenti) che arrivano e fanno cose.

Corollario: *Infrastructure as code*

Se solo il codice definisce i componenti dell'infrastruttura significa che questa **non puó** essere determinata da:

1. configurazioni manuali,
2. cose che si *cliccano* di qua e di là,
3. persone (*a.k.a.* consulenti) che arrivano e fanno cose.

Corollario: *Infrastructure as code*

Se solo il codice definisce i componenti dell'infrastruttura significa che questa **non puó** essere determinata da:

1. configurazioni manuali,
2. cose che si *cliccano* di qua e di là,
3. persone (*a.k.a.* consulenti) che arrivano e fanno cose.

Measure everything

3 MISURARE ogni componente dell'infrastruttura. Il concetto di *monitoring* non è affatto nuovo, l'innovazione è nell'avere degli strumenti che permettano di controllare **tutte le parti**.

Nell'approccio tradizionale si controlla **solo la parte sistemistica** mentre la parte applicativa ha — nella migliore delle ipotesi — al più una soluzione *ad hoc*.

Measure everything

3 MISURARE ogni componente dell'infrastruttura. Il concetto di *monitoring* non è affatto nuovo, l'innovazione è nell'avere degli strumenti che permettano di controllare **tutte le parti**.
Nell'approccio tradizionale si controlla **solo la parte sistemistica** mentre la parte applicativa ha — nella migliore delle ipotesi — al più una soluzione *ad hoc*.

Share

4 CONDIVIDERE un progetto comune, un obiettivo, delle pratiche, delle tecniche, degli strumenti, fra gruppi eterogenei, e che hanno obiettivi differenti (complementari).

> *Sharing is the loopback in the CAMS cycle. Creating a culture where people share ideas and problems is critical. (John Willis)*

BIODEC
Evolving ICT Infrastructures

Forse non si è capito ma . . .

. . . dire che solo il codice definisce l'infrastruttura, e che ogni azione deve essere automatizzata . . . ovvero trasformata in *software* . . . implica che chiunque adotti queste pratiche, e indipendentemente dal nome con cui si fa chiamare, è anche un . . .

PROGRAMMATORE !

Forse non si è capito ma . . .

. . . dire che solo il codice definisce l'infrastruttura, e che ogni azione deve essere automatizzata . . . ovvero trasformata in *software* . . .
implica che chiunque adotti queste pratiche, e indipendentemente dal nome con cui si fa chiamare, è anche un . . .

PROGRAMMATORE !

Forse non si è capito ma . . .

. . . dire che solo il codice definisce l'infrastruttura, e che ogni azione deve essere automatizzata . . . ovvero trasformata in *software* . . . implica che chiunque adotti queste pratiche, e indipendentemente dal nome con cui si fa chiamare, è anche un . . .

PROGRAMMATORE !

Forse non si è capito ma . . .

. . . dire che solo il codice definisce l'infrastruttura, e che ogni azione deve essere automatizzata . . . ovvero trasformata in *software* . . . implica che chiunque adotti queste pratiche, e indipendentemente dal nome con cui si fa chiamare, è anche un . . .

PROGRAMMATORE !

Indice

BIODEC
Evolving ICT Infrastructures

Istruzioni per l'uso

Una nota sull'uso dei font:

- il testo normale indica i programmi utilizzati in produzione, in BioDec,
- il *testo slanted* indica programmi valutati ma non in produzione (per ragioni diverse, non necessariamente perché inadeguati),
- il ~~testo cancellato~~ indica programmi che si possono sostituire con alternative più moderne.

Se l'aspetto teorico è ancora *in fieri* . . .

RIMANIAMO ANCORATI alle poche certezze che abbiamo,

. . . ovvero ai (nuovi) strumenti che sono stati creati in questi anni, per:

- il logging,
- il monitoring,
- la gestione degli allarmi (alerting).

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Logging la gestione degli **eventi**.

Monitoring la gestione delle **misure**.

Alerting la gestione delle **notifiche**.

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Logging la gestione degli **eventi**.

Monitoring la gestione delle **misure**.

Alerting la gestione delle **notifiche**.

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Logging la gestione degli **eventi**.

Monitoring la gestione delle **misure**.

Alerting la gestione delle **notifiche**.

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Logging la gestione degli **eventi**.

Monitoring la gestione delle **misure**.

Alerting la gestione delle **notifiche**.

Indice

BIODEC
Evolving ICT Infrastructures

Come si incastrano i vari sistemi fra loro

UNA COSA IMPORTANTE, che a volte genera confusione, è che i software di cui parleremo spesso svolgono diverse funzioni **tutte insieme**.

I.e. raccolgono log e ci fanno sopra un analisi volta a generare un alert, oppure riportano una misura relativa a *quando* si è rilevato un certo alert, eccetera.

Come si incastrano i vari sistemi fra loro

UNA COSA IMPORTANTE, che a volte genera confusione, è che i software di cui parleremo spesso svolgono diverse funzioni **tutte insieme**.

I.e. raccolgono log e ci fanno sopra un analisi volta a generare un alert, oppure riportano una misura relativa a *quando* si è rilevato un certo alert, eccetera.

Modularità

Come principio generale sarebbe bene avere un sistema di **alert indipendente**, che faccia (bene) solo quello, e **che usi come componenti** i dati dei sistemi di **logging e di monitoring**.

I sistemi commerciali falliscono quasi tutti su questo punto: il *feature creep* — derivante dal fatto che sono comparati sul numero di funzionalità e non sulla qualità — è responsabile di creare software *moloch* che fanno tutto, ma **male**.

Modularità

Come principio generale sarebbe bene avere un sistema di **alert indipendente**, che faccia (bene) solo quello, e **che usi come componenti** i dati dei sistemi di **logging e di monitoring**.
I sistemi commerciali falliscono quasi tutti su questo punto: il *feature creep* — derivante dal fatto che sono comparati sul numero di funzionalità e non sulla qualità — è responsabile di creare software *moloch* che fanno tutto, ma **male**.

Indice

BIODEC
Evolving ICT Infrastructures

Andare oltre `tail -f /var/log/syslog`

IL CONCETTO DI LOG è a volte sovrapposto o confuso con il concetto di misurare il funzionamento di un sistema.

Un log è diverso da un sistema di misura, perché, sebbene abbia la medesima connotazione di serie temporale, quanto tracciato sono **eventi** e non dati numerici.

BIODEC
Evolving ICT Infrastructures

Andare oltre `tail -f /var/log/syslog`

IL CONCETTO DI LOG è a volte sovrapposto o confuso con il concetto di misurare il funzionamento di un sistema.

Un log è diverso da un sistema di misura, perché, sebbene abbia la medesima connotazione di serie temporale, quanto tracciato sono **eventi** e non dati numerici.

Andare oltre `tail -f /var/log/syslog`

I componenti di un sistema di logging:

Route	syslog-ng, rsyslog, *logstash*, *heka*,
Store	elasticsearch (mongodb),
Aggregate	graylog2,
Visualize	graylog2, *kibana3*,
Analyze	graylog2, *kibana3*,
Alert	un sistema di alerting.

Per intenderci, un sistema “classico” ha tutti i componenti svolti da syslog, con programmi come logwatch o simili per farne l’analisi. Oppure soluzioni proprietarie, tipicamente molto costose.

Graylog2

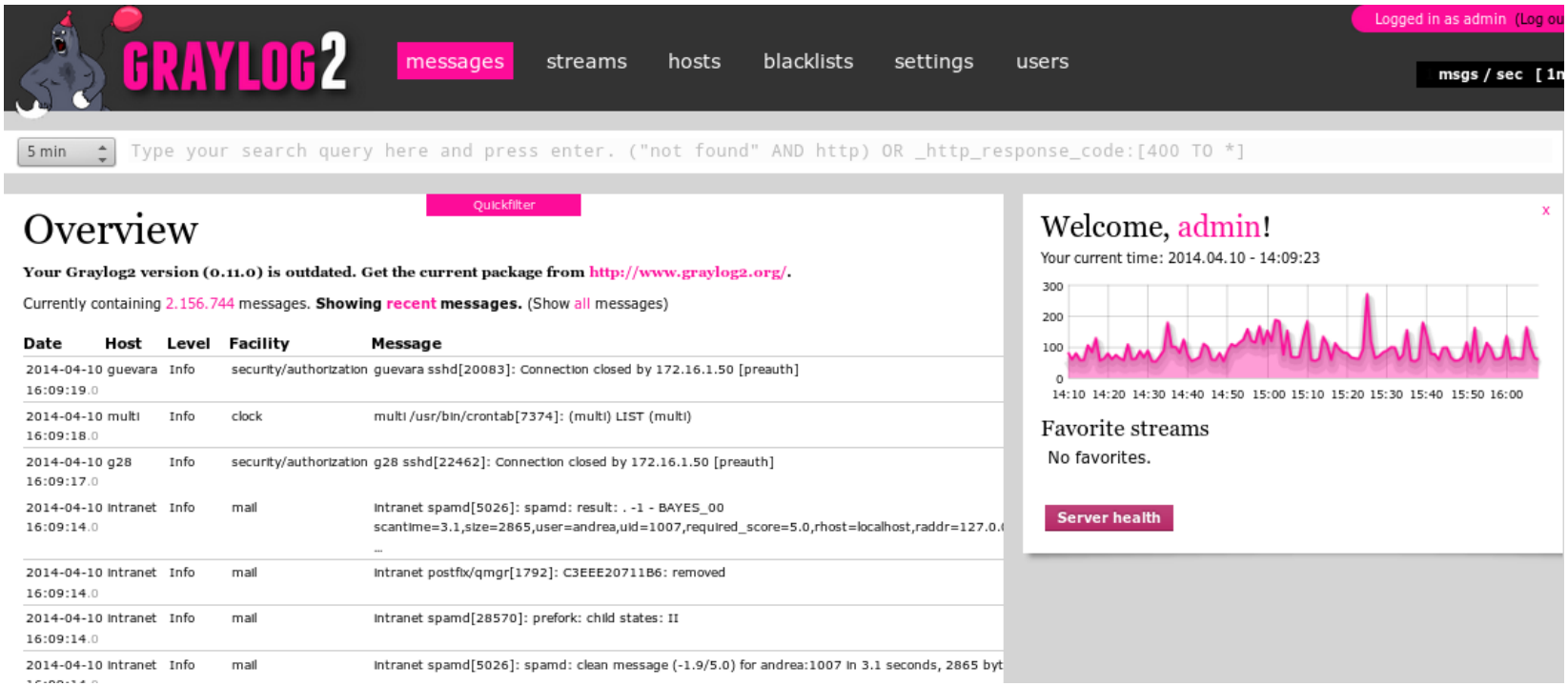
Overview

Your Graylog2 version (0.11.0) is outdated. Get the current package from http://www.graylog2.org/.

Currently containing 2.156.744 messages. **Showing recent messages.** (Show all messages)

Date	Host	Level	Facility	Message
2014-04-10 16:09:19.0	guevara	Info	security/authorization	guevara sshd[20083]: Connection closed by 172.16.1.50 [preauth]
2014-04-10 16:09:18.0	multi	Info	clock	multi /usr/bin/crontab[7374]: (multi) LIST (multi)
2014-04-10 16:09:17.0	g28	Info	security/authorization	g28 sshd[22462]: Connection closed by 172.16.1.50 [preauth]

Graylog2



BIODEC
Evolving ICT Infrastructures

Indice

BIODEC
Evolving ICT Infrastructures

Measure ! Measure ! Measure everywhere !

PER DEFINIRE l'atto del controllare, dobbiamo definire *cosa* intendiamo controllare, ovvero *cosa intendiamo misurare*.

Una misura è un **valore numerico con un nome** e il **momento** in cui essa è stata effettuata. Una successione di misure è pertanto una serie temporale di valore numerici associati ad un'etichetta (o nome).

Measure ! Measure ! Measure everywhere !

PER DEFINIRE l'atto del controllare, dobbiamo definire *cosa* intendiamo controllare, ovvero *cosa intendiamo misurare*. Una misura è un **valore numerico con un nome** e il **momento** in cui essa è stata effettuata. Una successione di misure è pertanto una serie temporale di valore numerici associati ad un'etichetta (o nome).

Measure ! Measure ! Measure everywhere !

I componenti di un sistema di misura:

Route	collectd, statsd, *metricsd*,
Store	graphite (whisper), *blueflood*
Aggregate	graphite (carbon), *blueflood*
Visualize	graphite-web, graphana, graph-explorer,
Analyze	*sensu*,
Alert	un sistema di alerting.

Per intenderci, un sistema “classico” ha tutti i componenti svolti da Nagios, con Cacti / Pnp4Nagios o ~~Munin~~ come sistema di visualizzazione.

BIODEC
Evolving ICT Infrastructures

Punto chiave: la visualizzazione

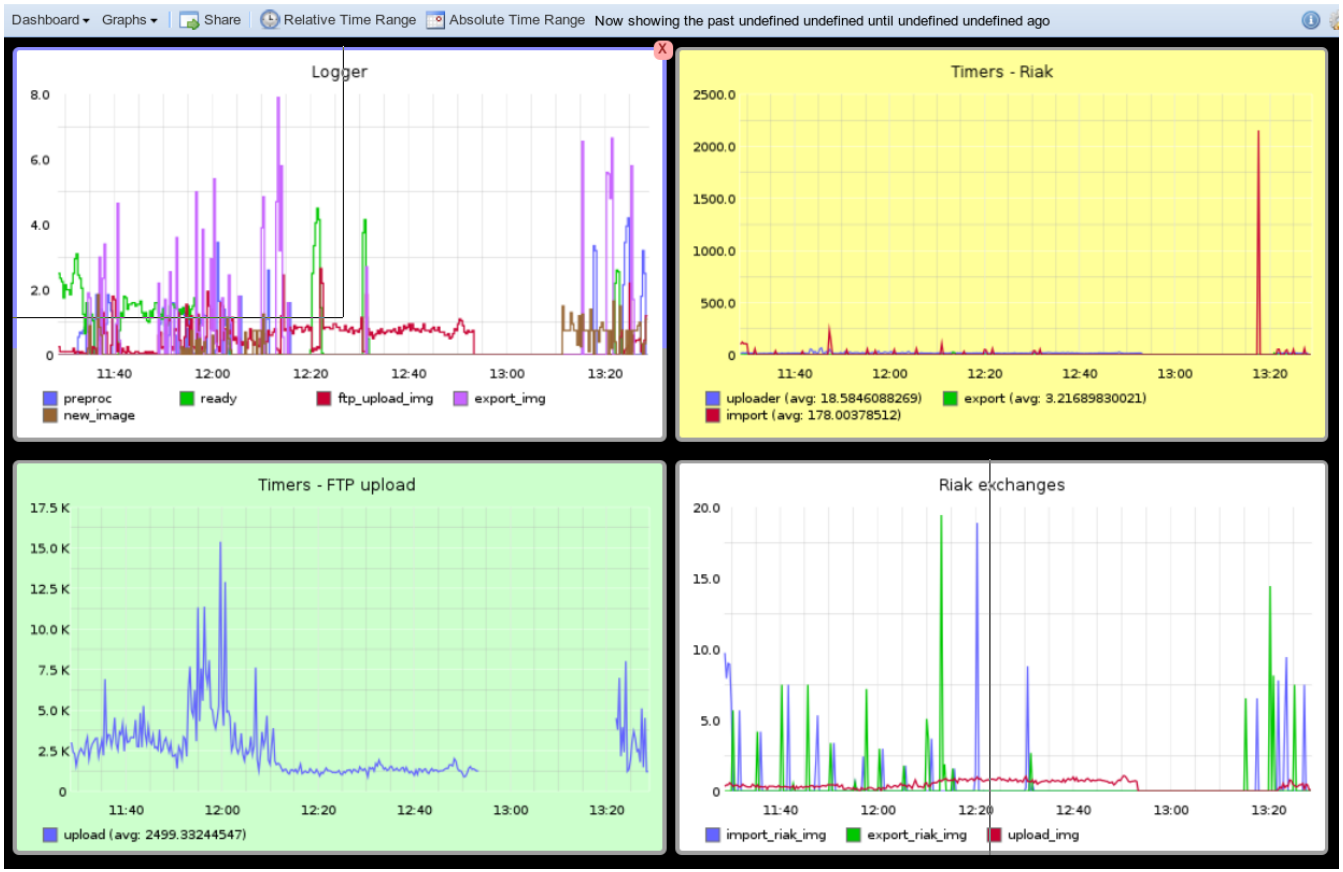
SEBBENE TUTTE le componenti siano necessarie, ce n'è una che è più critica delle altre. La caratteristica chiave di un sistema di monitoring è la **visualizzazione**.

Ovvero rendere immediatamente **esplicite** le informazioni.

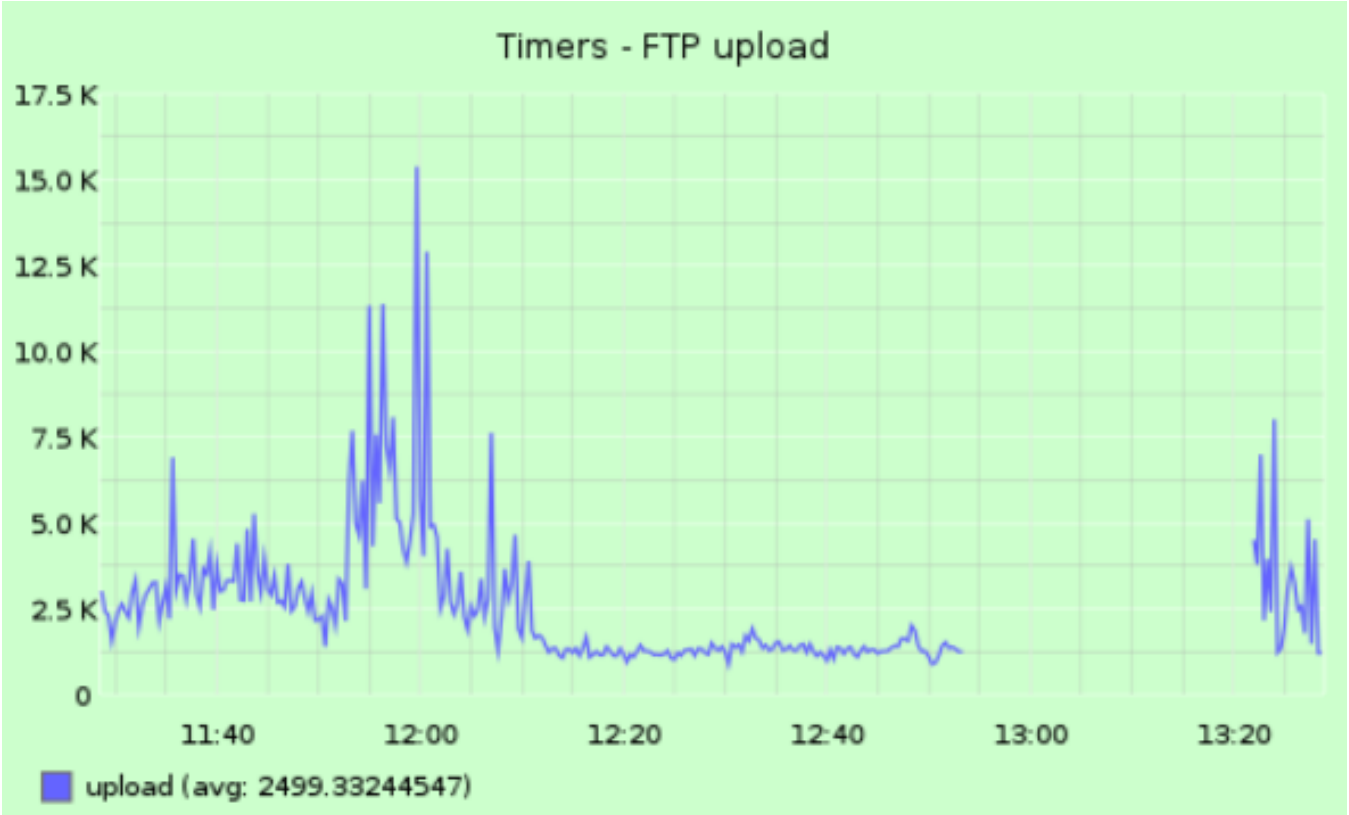
Punto chiave: la visualizzazione

SEBBENE TUTTE le componenti siano necessarie, ce n'è una che è più critica delle altre. La caratteristica chiave di un sistema di monitoring è la **visualizzazione**.
Ovvero rendere immediatamente **esplicite** le informazioni.

Graphite + Statsd

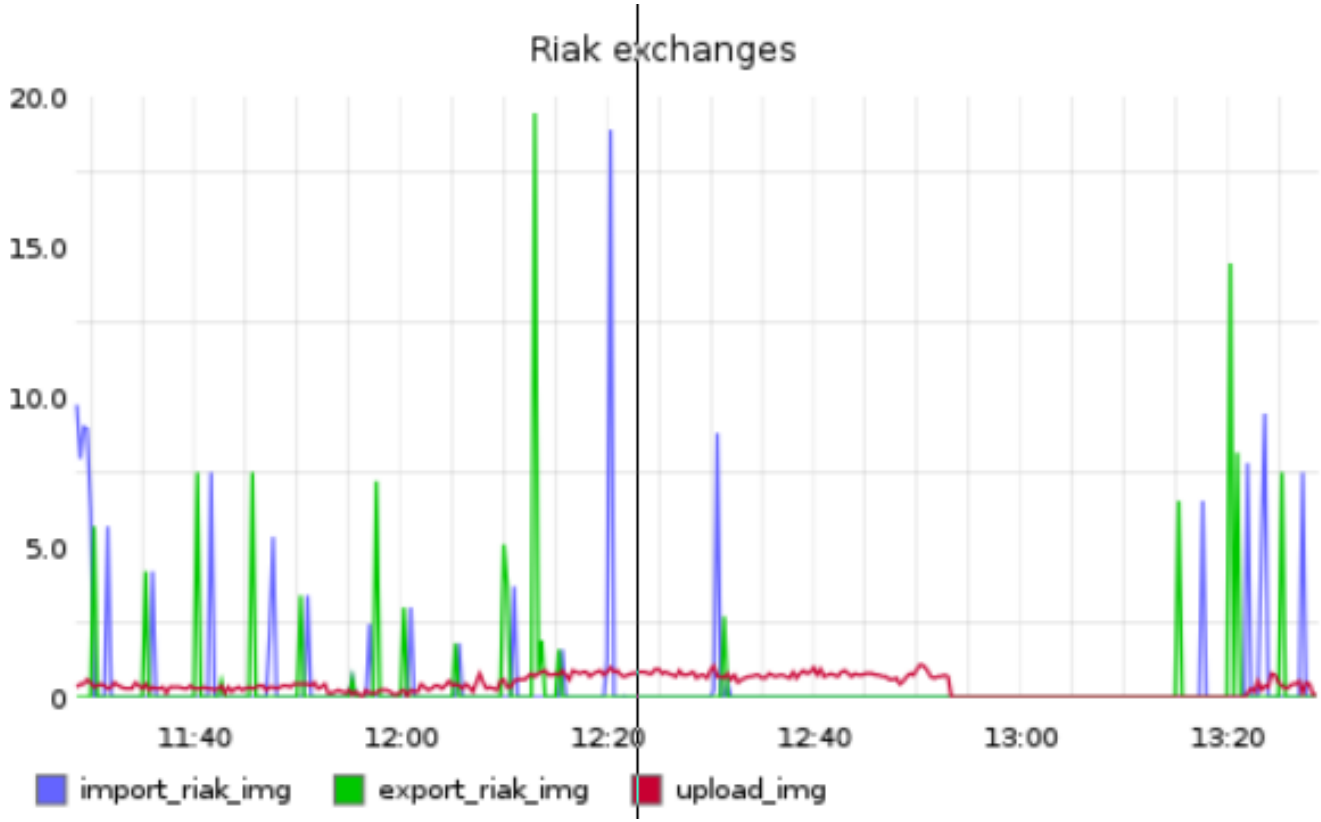


Graphite + Statsd



BIODEC
Evolving ICT Infrastructures

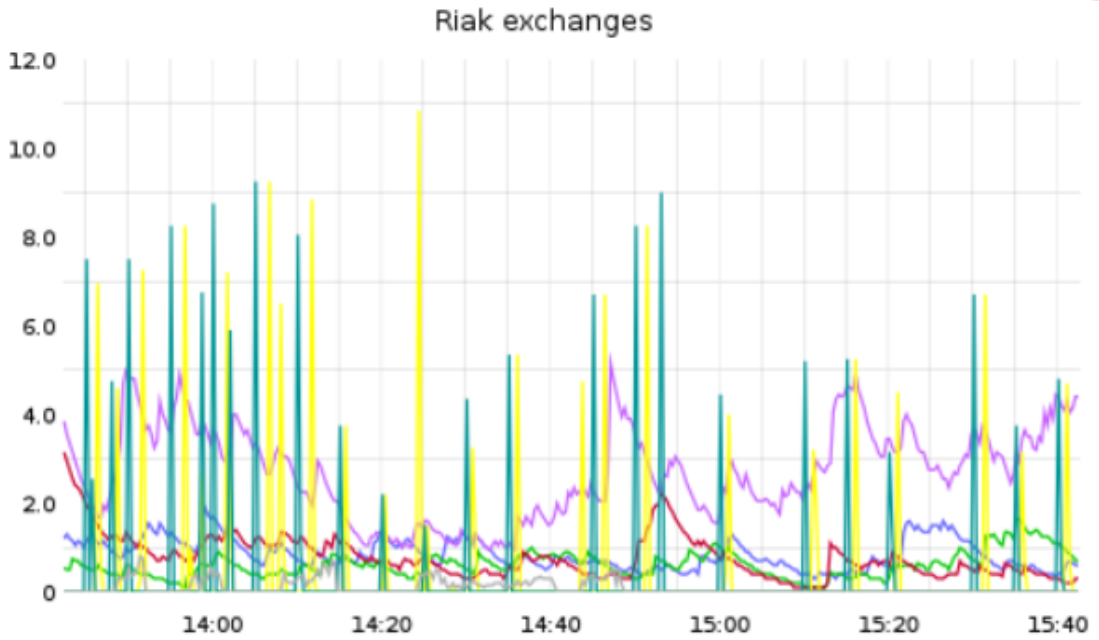
Graphite + Statsd



BIODEC
Evolving ICT Infrastructures

Integrazione di dati applicativi e di sistema

È lo stesso grafico di prima — un paio di ore dopo, per la precisione — ma col load di **ogni Riak server del cluster** (5 server virtuali **diversi**).



BIODEC
Evolving ICT Infrastructures

Indice

BIODEC
Evolving ICT Infrastructures

Una telefonata nel cuore della notte

```
Address: node022.example.com
Service: Memory used
State: WARNING -> OK (RECOVERY)
Command: check_mk-mem.used
Output: OK - 3.07 GB used (2.82 GB RAM + 0.24 GB
SWAP, this is 4.9% of 62.89 GB RAM)
```

Il sistema di allarme

SIA IL SISTEMA di logging che quello di monitoring avevano come ulteriore (sotto-)componente il sistema di alerting, che è pertanto **trasversale** e **comune** a diversi ambiti.

Un sistema d'allarme è un meccanismo che **genera messaggi** specifici ad uno **stato del sistema**, e li **recapita** ad un determinato **destinatario**.

Il sistema di allarme

SIA IL SISTEMA di logging che quello di monitoring avevano come ulteriore (sotto-)componente il sistema di alerting, che è pertanto **trasversale** e **comune** a diversi ambiti.
Un sistema d'allarme è un meccanismo che **genera messaggi** specifici ad uno **stato del sistema**, e li **recapita** ad un determinato **destinatario**.

Come fare un sistema di alerting ?

Nella sua versione minimale un sistema d'allarme è composto dai seguenti componenti:

1. un **generatore** di allarmi,
2. il **messaggio**, che descrive l'allarme,
3. il **destinatario** del messaggio,
4. il **sotto-sistema preposto alla consegna** del messaggio.

BIODEC
Evolving ICT Infrastructures

Come fare un sistema di alerting ?

Nella sua versione minimale un sistema d'allarme è composto dai seguenti componenti:

1. un **generatore** di allarmi,
2. il **messaggio**, che descrive l'allarme,
3. il **destinatario** del messaggio,
4. il **sotto-sistema preposto alla consegna** del messaggio.

BIODEC
Evolving ICT Infrastructures

Come fare un sistema di alerting ?

Nella sua versione minimale un sistema d'allarme è composto dai seguenti componenti:

1. un **generatore** di allarmi,
2. il **messaggio**, che descrive l'allarme,
3. il **destinatario** del messaggio,
4. il **sotto-sistema preposto alla consegna** del messaggio.

BIODEC
Evolving ICT Infrastructures

Come fare un sistema di alerting ?

Nella sua versione minimale un sistema d'allarme è composto dai seguenti componenti:

1. un **generatore** di allarmi,
2. il **messaggio**, che descrive l'allarme,
3. il **destinatario** del messaggio,
4. il **sotto-sistema preposto alla consegna** del messaggio.

Come fare un sistema di alerting ?

I componenti di un sistema di alerting:

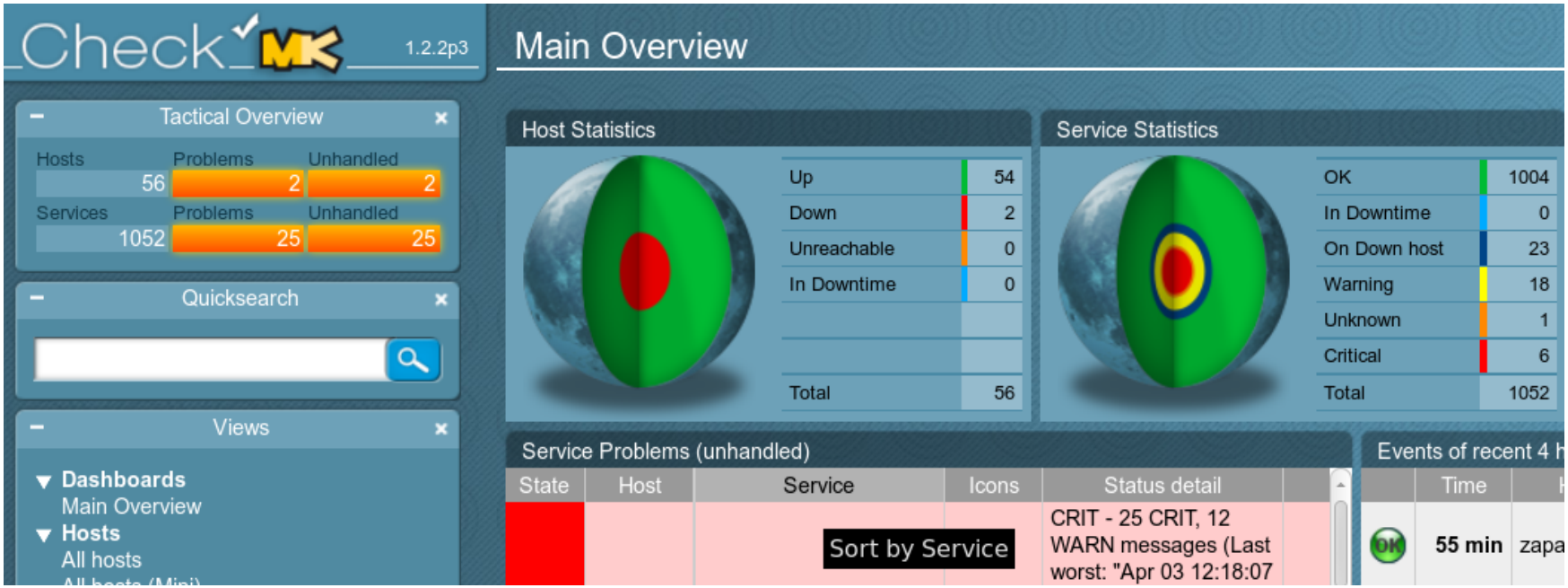
Generator nagios, icinga, *flapjack* (che chiama questa componente *event processing*), *sensu*,

Message email, SMS, sirene, . . .

Router nagios, icinga, *flapjack*, *sensu*,

Delivery determinata dal *message type*, quindi SMTP per l'email, eccetera.

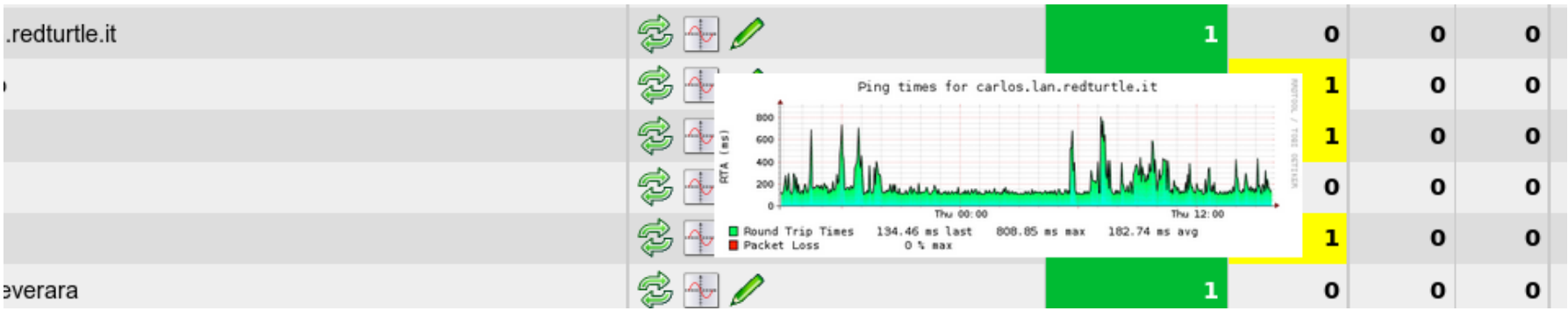
~~Good ol' Nagios~~ Check_MK Multisite



BIODEC
Evolving ICT Infrastructures

~~Good ol' Nagios~~ Check_MK Multisite

Dettaglio: sono presenti anche dati di monitoraggio. Gli strumenti di vecchia concezione mescolano diversi aspetti in un'unica interfaccia.



Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Concludendo e riassumendo

- I sistemi di logging, monitoring e alerting, **servono** e **serviranno** sempre di più.
- Sono sistemi **trasversali**, soprattutto per l'integrazione di dati applicativi (*Dev*) e di sistema (*Ops*). **Integrazione** senza la quale certe attività di debug possono essere **impossibili**.
- Esistono soluzioni moderne per costruire sistemi **modulari** ed **evolvibili**.
- Rifuggete la tentazione di adottare un sistema **monolitico**.

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2015 Incontro DevOps Italia 2015 ???

More news Seguite il blog BioDec
http://blog.biodec.com/

BIODEC
Evolving ICT Infrastructures

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2015 Incontro DevOps Italia 2015 ???

More news Seguite il blog BioDec
http://blog.biodec.com/

BIODEC
Evolving ICT Infrastructures

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2015 Incontro DevOps Italia 2015 ???

More news Seguite il blog BioDec
http://blog.biodec.com/

*licenza della presentazione:
http://creativecommons.org/licenses/by-sa/3.0/

BIODEC
Evolving ICT Infrastructures

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2015 Incontro DevOps Italia 2015 ???

More news Seguite il blog BioDec
`http://blog.biodec.com/`

*licenza della presentazione:
`http://creativecommons.org/licenses/by-sa/3.0/`

BIODEC
Evolving ICT Infrastructures