



Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Installazione e Configurazione di un NMS di classe Enterprise

A. Messina, P. Storniolo

Rapporto Tecnico N.:
RT-ICAR-PA-12-03

Luglio 2012



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: www.icar.cnr.it
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.na.icar.cnr.it
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: www.pa.icar.cnr.it



Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Installazione e Configurazione di un NMS di classe Enterprise

A. Messina¹, P. Storniolo¹

Rapporto Tecnico N.:
RT-ICAR-PA-12-03

Luglio 2012

¹ Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo, Viale delle Scienze edificio 11, 90128 Palermo.

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

Sommario

1	PREFAZIONE	6
1.1	Introduzione	6
1.2	I sistemi di management Open Source	6
1.3	Scelta del sistema di management	8
2	PANORAMICA SU ZABBIX	9
2.1	Introduzione	9
2.2	Funzionamento	9
2.3	Vantaggi	10
2.4	Funzionalità	11
3	SETUP DEL SISTEMA OPERATIVO	12
3.1	Requisiti hardware	12
3.2	Requisiti software	12
3.3	Scelta del server	13
3.4	Installazione di CentOS 6.3	14
3.5	Installazione dei requisiti software	17
3.5.1	MySQL Server	18
3.5.2	Apache e PHP	18
3.5.3	Ambiente di sviluppo	19
3.5.4	Librerie per controlli IPMI, SSH e HTTP	19
3.5.5	Network tools	20
3.5.6	Supporto Jabber	20
3.5.7	Supporto Java	21
3.6	Configurazione	21

3.6.1	Creazione utente non privilegiato	21
3.6.2	Attivazione di MySQL	22
3.6.3	Configurazione di PHP	22
3.6.4	Configurazione di Apache	22
3.6.5	Configurazione del firewall locale	23
3.6.6	Disattivazione servizi non necessari	23
4	INSTALLAZIONE DI ZABBIX	24
4.1	Introduzione	24
4.2	Compilazione dei sorgenti	24
4.3	Predisposizione del database	26
4.3.1	Dimensioni del Database	27
4.4	Configurazione degli script di avvio automatico	29
4.5	Installazione dell'interfaccia web	29
4.6	Avvio dei servizi	30
4.7	Configurazione del frontend	30
4.7.1	Verifica dei prerequisiti	31
4.7.2	Configurazione e test della connessione al database	31
4.7.3	Generazione del file di configurazione del frontend	32
4.8	Termine dell'installazione	33
5	GUIDA RAPIDA ALL'USO DI ZABBIX	34
5.1	Introduzione	34
5.2	Creazione di un utente	34
5.2.1	Associazione di un <i>media</i>	35
5.2.2	Assegnazione dei privilegi	36
5.3	Creazione di un host	36
5.4	Creazione di un item	37

5.4.1	Visualizzazione dei dati	38
5.4.2	Grafici	39
5.5	Creazione di un trigger	39
5.5.1	Visualizzare lo stato di un trigger	41
5.6	Ricezione delle notifiche	41
5.6.1	Impostazioni e-mail	41
5.6.2	Creazione di una nuova azione	42
5.7	Creazione di un template	43
5.7.1	Aggiunta di un template	43
5.7.2	Aggiunta di un item ad un template	44
5.7.3	Associazione di un template ad un host	45

1 Prefazione

1.1 Introduzione

Negli ultimi anni si è assistito ad una notevole evoluzione dei servizi offerti dalle reti aziendali. Oggi è comune trovare, anche in piccole aziende, server per la gestione della posta elettronica, del sito web aziendale, del sito internet oltre ai tradizionali file server. Quindi si può affermare che ci sono servizi di cui un'azienda moderna non può fare a meno.

Il compito di un buon sistema di monitoraggio, denominato anche NMS (Network Management System), è controllare che questi servizi siano sempre attivi e raggiungibili. Un sito internet momentaneamente non raggiungibile può provocare danni a livello d'immagine, e notevoli perdite economiche per l'azienda. Perciò tutti i servizi che si affacciano su internet e che sono accessibili dall'esterno devono essere continuamente monitorati a intervalli prestabiliti.

Inoltre un sistema di monitoraggio deve controllare le risorse dei terminali di maggior importanza all'interno della rete aziendale. Se il terminale adibito a web-server ha troppi processi attivi questo potrebbe rallentare la macchina e quindi non fornire in modo adeguato il proprio servizio.

Naturalmente è di primaria importanza l'implementazione del sistema di monitoraggio in una situazione di piena sicurezza; di conseguenza la macchina di management sarà posta dietro ad un firewall, i dati che fluiscono attraverso la rete devono essere crittografati e chi ha accesso ai dati del monitoraggio ci può arrivare attraverso le opportune autenticazioni.

Un sistema di monitoraggio efficiente e completo è solo uno specchio della situazione del cliente: informa chi è adibito all'assistenza di quali servizi o terminali sono attivi o disattivi e non prende nessuna decisione attiva su come risolvere i problemi.

Sarà compito di chi è adibito all'assistenza risolvere il problema proposto dal sistema di monitoraggio.

1.2 I sistemi di management Open Source

Molti sono i sistemi di monitoraggio di reti presenti sul mercato e hanno caratteristiche in linea di massima pressoché similari.

Tra le soluzioni di management Open Source disponibili, alcune sono sicuramente delle ottime soluzioni per specifici requisiti di nicchia.

MRTG (Multi Router Traffic Grapher), scritto da Toby Oetiker, è un esempio eccellente di applicazione compatta che utilizza il protocollo SNMP per collezionare e memorizzare informazioni di performance e per visualizzarle graficamente.

Se già questo soddisfa le proprie necessità, non è necessario considerare altre eventuali alternative, ma si tenga presente che lo strumento non sarà molto di aiuto nella considerazione di problemi provenienti da dispositivi diversi e nella gestione di tali problemi fino alla loro risoluzione.

Un'estensione di MRTG è RRDTool (Round Robin Database Tool), sempre di Tobi Oetiker. E' sempre fondamentalmente uno strumento orientato alla raccolta periodica ed alla visualizzazione dei dati numerici di performance, ma si basa su database. La dimensione del database è comunque predeterminata al momento della creazione e i nuovi dati vanno a sovrascrivere i vecchi dopo un intervallo di tempo prefissato.

Una ulteriore estensione di RRDTool è Cacti, che fornisce un frontend completo alle funzionalità di RRDTool. Con Cacti è possibile usare un database relazionale MySQL come back end a monte dei database Round Robin. Le sorgenti dati, inoltre, possono essere costituite anche da script, oltre che ovviamente SNMP, e sono incluse funzionalità di gestione utenti. Cacti rimane comunque uno strumento per il collezionamento e la visualizzazione di dati di performance, e non una soluzione completa, basata su frame work, per il system management.

Salendo la scala delle funzionalità e complessità, troviamo alcune soluzioni più orientate al network management (netdisco, The Dude), ed altre invece più orientate al system management (Nagios, Zabbix).

Le soluzioni più complete e di più alto livello presentano un'architettura basata su database centralizzato (Nagios, Zenoss, OpenNMS, Zabbix).

Infine, alcuni sono progetti totalmente Open Source, tipicamente rilasciati sotto la licenza Gnu GPL (MRTG, RRDTool, Cacti) o la licenza BSD (netdisco), mentre alcuni sono disponibili gratuitamente (tipicamente in GPL) ma presentano estensioni con licenza commerciale (Zenoss).

In aggiunta alle licenze free, per diversi prodotti sono anche disponibili contratti di supporto a pagamento (Zenoss, Nagios, OpenNMS, Zabbix).

1.3 Scelta del sistema di management

In definitiva, la scelta del sistema di management più appropriato deve essere unicamente guidata dai requisiti desiderati.

Per piccoli ambienti di management, potrebbe essere preso in considerazione Nagios, ben testato ed affidabile e con alle spalle il supporto di una grossa comunità. Si tenga presente però che, per qualunque altro controllo che non sia dei semplici ping check o SNMP check, potrebbe esservi la necessità di installare plugin anche sugli host da monitorare. Le notifiche sono abbastanza semplici da impostare, ma se si ha bisogno anche di effettuare un minimo di analisi sul log degli eventi, allora Nagios non è la scelta migliore.

OpenNMS e Zenoss sono entrambi dei prodotti estremamente competitivi, dotati di funzionalità di auto-discovery, monitoraggio della disponibilità, problem management, performance management e reporting.

Zenoss ha una qualche forma di mapping di topologie di rete ed una migliore documentazione, ma il codice sembra essere meno affidabile.

OpenNMS soffre d'altra parte di un approccio un pò confusionario riguardo alla gestione di eventi, allarmi e notifiche. Inoltre, la modifica di un file di configurazione richiede il riavvio dell'intero ambiente.

Zabbix, infine, nonostante abbia una curva di apprendimento ripida, è dotato di quasi tutte le funzionalità normalmente richieste, supporta diversi tipologie di database di backend, ed offre delle peculiarità non riscontrabili in altri sistemi, quali il Browser SNMP e la gestione dei Service Level Agreement (SLA) per i servizi.

Data la sua architettura ben definita, la possibilità documentata di estenderne le funzionalità utilizzando apposite API, il supporto quasi completo ad SNMP (manca solo un MIB compiler), per gli scopi di questo lavoro la scelta è ricaduta sul sistema Zabbix.

2 Panoramica su Zabbix

2.1 Introduzione

Zabbix è una soluzione di monitoraggio distribuito open source e di classe enterprise creato nel 1998 come progetto interno in una banca a cura dello sviluppatore Alexei Vladishev e adesso attivamente sviluppato e supportato da Zabbix SIA.

Il software è scritto e distribuito sotto licenza GPL General Public License versione 2, quindi il suo codice sorgente è distribuito gratuitamente e reso disponibile a chiunque.

Il presente lavoro è incentrato sull'ultima versione disponibile, la 2.0.1.

2.2 Funzionamento

Zabbix consente il monitoraggio di praticamente tutti i parametri di un componente di rete e dello stato di salute e di integrità dei server e di qualunque host in generale.

Zabbix supporta due tipologie di interrogazione: *polling* e *trapping*.

In modalità *polling* interroga costantemente il database chiedendo quali siano gli item da controllare, passa alla lista di host e agent cui appartengono tali item, richiede i dati, li scrive nel database, li elabora e infine processa tutti i trigger.

Un *item* è un singolo controllo o un singolo dato di performance che viene monitorato dal server.

Un *trigger* è un'espressione logica che rappresenta lo stato del sistema. Lo stato del trigger è ricalcolato ogni volta che il server riceve un nuovo valore e quest'ultimo fa parte dell'espressione del trigger.

In modalità *trapping* il server si trova in gran parte inattivo, aspettando che gli host e gli agent inviino i loro dati. Insieme ai dati riceverà anche la chiave item che il dato rappresenta. A quel punto il server scrive i dati nel database, li elabora e processa anche eventuali trigger.

Zabbix supporta anche un altro tipo di dato, denominato *internal data*, suddiviso a sua volta in due categorie: le voci *history* e le *funzioni aggregate*.

Le voci *history* contengono la storia di ogni item e di ogni trigger e vengono mantenute per un determinato periodo di tempo configurabile in un'apposita tabella del database.

Le *funzioni aggregate* sono da considerare come pseudo-item, che uniscono o aggregano valori provenienti da ulteriori item.

Ad esempio, si può ottenere il carico medio di CPU per un intero gruppo di server o il numero totale di utenti loggati in un gruppo di server.

Gli item calcolati vengono aggiornati con il meccanismo con cui sono stati aggregati, quindi in modalità polling o trapping.

Tutti i report, le statistiche, i file di configurazione sono accessibili mediante un frontend basato sul web. Ciò garantisce che lo stato della rete e dei server possa essere valutato virtualmente da qualsiasi luogo.

2.3 Vantaggi

Zabbix offre i seguenti vantaggi:

- Auto-discovery di server e dispositivi di rete
- Monitoraggio distribuito con amministrazione Web centralizzata
- Supporto per meccanismi di polling e trapping
- Software server per Linux, Solaris, Windows, OS X, AIX, FreeBSD, OpenBSD
- Agenti nativi ad alte prestazioni (software client per Linux, Solaris, Windows NT4.0, Windows 2000, Windows XP, Windows Vista, OS X, AIX, FreeBSD, OpenBSD)
- Autenticazione dell'utente sicura
- Autorizzazioni utente flessibili
- Interfaccia basata sul web
- Comunicazione di eventi predefiniti via e-mail
- Vista ad alto livello delle risorse monitorate

2.4 Funzionalità

Zabbix fornisce le seguenti funzionalità:

- Alert e notifiche ripetute. Un alert può essere configurato in modo molto flessibile e può comprendere non solo le notifiche, ma anche l'esecuzione di comandi remoti e IPMI.
- Supporto IPv6. Tutti i moduli di Zabbix supportano sia IPv4 che IPv6. Zabbix può essere usato in ambiente misto o IPv6 soltanto. IPv6 introduce alcuni nuovi servizi e semplifica molto la configurazione e la gestione delle reti IP. La sua caratteristica più appariscente è il più ampio spazio di indirizzamento: poiché riserva 128 bit per gli indirizzi, con IPv6 si hanno complessivamente 2^{128} indirizzi disponibili.
- Supporto IPMI. Zabbix supporta il monitoraggio di parametri IPMI e l'esecuzione manuale di comandi IPMI e remoti direttamente dal front-end.
- Monitoraggio distribuito.
- Processo Zabbix Proxy. Zabbix proxy è un processo leggero, che raccoglie i dati per conto di Zabbix Server. I proxy possono essere usati allo scopo di centralizzare il controllo di postazioni remote riportando al server uno dei nodi dell'ambiente distribuito. Zabbix Proxy semplifica significativamente l'implementazione e la manutenzione della centralizzazione del monitoraggio distribuito
- Dashboard. La Dashboard di Zabbix offre dettagli di alto livello circa l'ambiente monitorato. Questa è una delle parti centrali del front-end Zabbix.
- Modulo Cache Database. Quando questo modulo è abilitato viene aumentata la velocità di Zabbix di 4-8 volte a seconda della configurazione.
- Schermate Dinamiche. Un elemento dello schermo può essere reso dinamico e le relative informazioni visualizzate dipenderanno dal particolare host selezionato dall'utente.
- Zoom per grafici.
- Grafici a Torta.
- Comunicazioni efficienti con gli Agent. Gli Agent Zabbix supportano la memorizzazione di dati, regolata dai parametri BufferSize e BufferSend. Il protocollo di comunicazione supporta l'invio di più valori su una singola transazione su connessione TCP.
- Elaborazione speciale degli OID SNMP noti. Gli OID SNMP semplici come iDescr, ifInDescr e altri possono essere utilizzati in Zabbix e saranno tradotti automaticamente in una corretta rappresentazione numerica da Zabbix stesso.

3 Setup del Sistema Operativo

3.1 Requisiti hardware

I requisiti hardware minimi richiesti da Zabbix sono relativamente modesti:

- Processore Pentium IV
- 128 MByte di memoria RAM
- 256 MByte di spazio su disco

Comunque, lo spazio su disco richiesto effettivo dipenderà principalmente dal numero di host e di parametri che si intende monitorare, nonché dai dati di history che si vogliono mantenere.

Si tenga anche presente che ogni processo server di Zabbix richiede diverse connessioni al database. L'ammontare di memoria allocata per ciascuna connessione dipenderà dalla configurazione del particolare motore di database scelto.

Le configurazioni hardware tipiche in funzione del numero di host da monitorare sono le seguenti:

Host	Piattaforma	CPU/Memoria	Database
20	Ubuntu Linux	PII 350MHz 256 MB	SQLite
500	Ubuntu Linux 64 bit	AMD Athlon 3200+ 2 GB	MySQL InnoDB
>1000	Ubuntu Linux 64 bit	Intel Dual Core 6400 4 GB	RAID10 MySQL InnoDB o PostgreSQL
>10000	Red Hat Enterprise	Intel Xeon 2xCPU 8 GB	Fast RAID10 MySQL InnoDB o PostgreSQL

3.2 Requisiti software

Dal punto di vista software, invece, Zabbix supporta i seguenti tipi di DBMS:

- MySQL, versione 5.0 e successive
- Oracle, versione 10g e successive
- PostgreSQL, versione 8.1 e successive
- SQLite, versione 3.3.5 e successive (sperimentale)
- IBM DB2, versione 9.7 e successive (sperimentale)

Il frontend web-based richiede invece:

- Apache, versione 1.3.12 e successive
- PHP, versione 5.1.6 e successive.
- Estensioni PHP:
 - gd
 - bcmath
 - ctype
 - libXML
 - xmlreader
 - xmlwriter
 - session
 - sockets
 - mbstring
 - gettext
 - ibm_db2 (richiesta se IBM DB2 è il database di backend)
 - mysql (richiesta se MySQL è il database di backend)
 - oci8 (richiesta se Oracle è il database di backend)
 - pgsql (richiesta se PostgreSQL è il database di backend)
 - sqlite3 (richiesta se SQLite è il database di backend)

Per quanto riguarda il web browser, infine, sono supportate le ultime versioni di Google Chrome, Mozilla Firefox, Microsoft Internet Explorer e Opera. Cookies e JavaScript devono essere abilitati.

3.3 Scelta del server

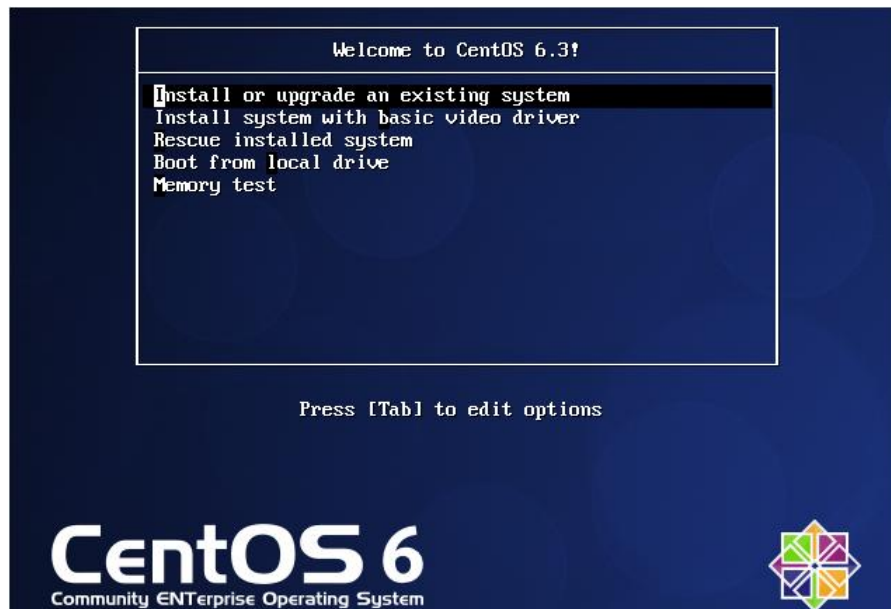
Considerati gli scopi del presente lavoro, si è deciso di utilizzare l'ambiente di virtualizzazione Citrix Xen Server 6.3 nel quale è stata creata una macchina virtuale con le seguenti risorse:

- 1 CPU
- 512 MB di RAM
- Hard Disk da 30 GB
- Scheda di rete 100 Mbit
- Lettore di CD-ROM

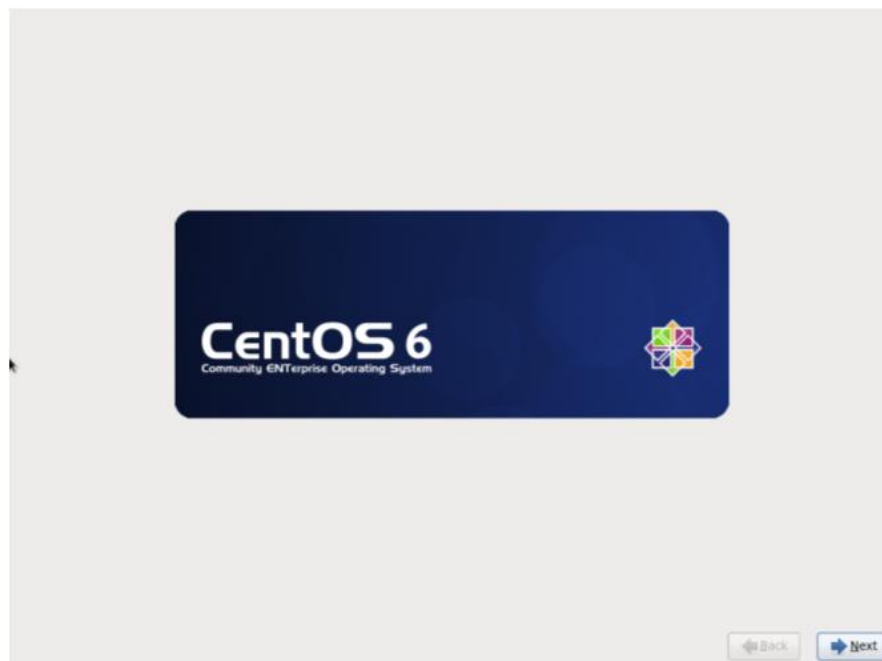
Il sistema operativo scelto è stato CentOS 6.3 64 bit, funzionalmente equivalente alla distribuzione Red Hat Enterprise Linux 6.3 64 bit.

3.4 Installazione di CentOS 6.3

Assegnata all'unità CD-ROM l'immagine ISO della distribuzione CentOS 6.3 64 bit *minimal*, in fase di avvio la macchina virtuale carica immediatamente il boot loader in essa contenuto:

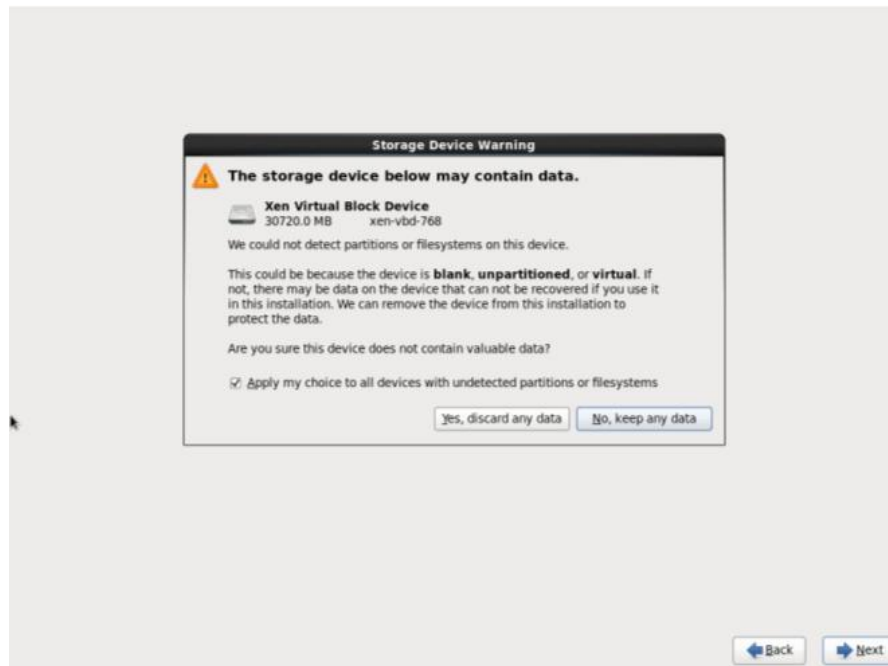


La selezione della prima voce fa partire immediatamente la procedura di installazione:

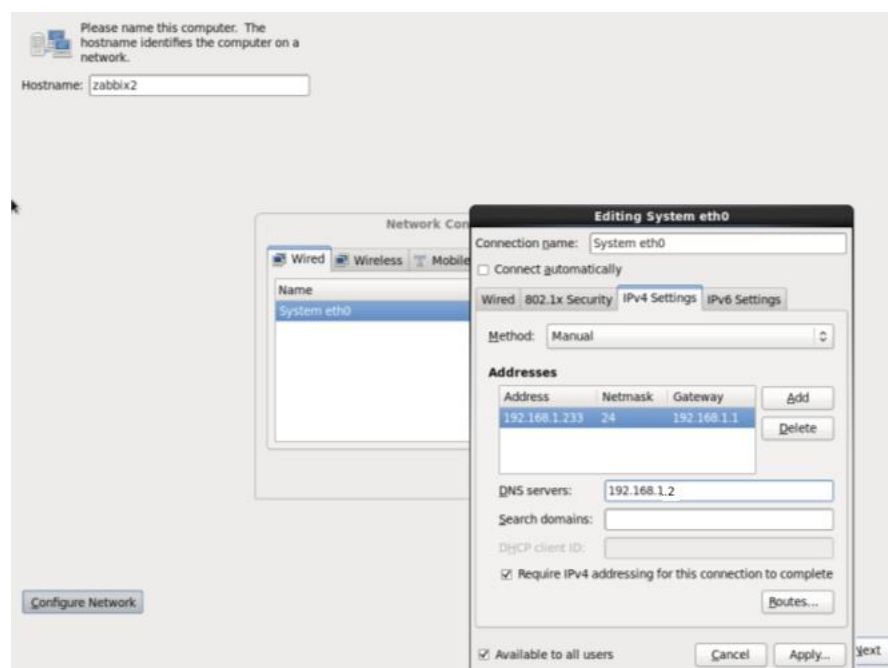


Poiché l'unità disco assegnata non contiene né dati né partizioni, ne viene richiesta l'inizializzazione.

In questa fase è sufficiente confermare di volerla utilizzare scegliendo le impostazioni di default, che implicano l'utilizzo dell'intero disco e di uno schema di partizionamento predefinito.



Successivamente viene richiesto di assegnare un nome al sistema che si sta andando ad installare ed è possibile anche indicarne la configurazione di rete.

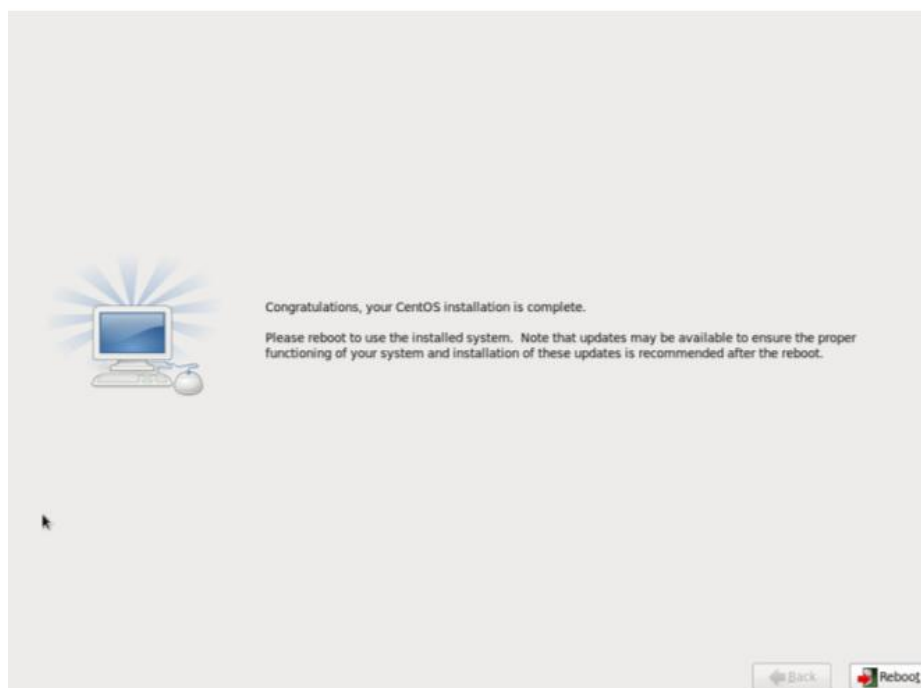


A questo punto il disco viene partizionato e formattato ed inizia la procedura di copia dei pacchetti.

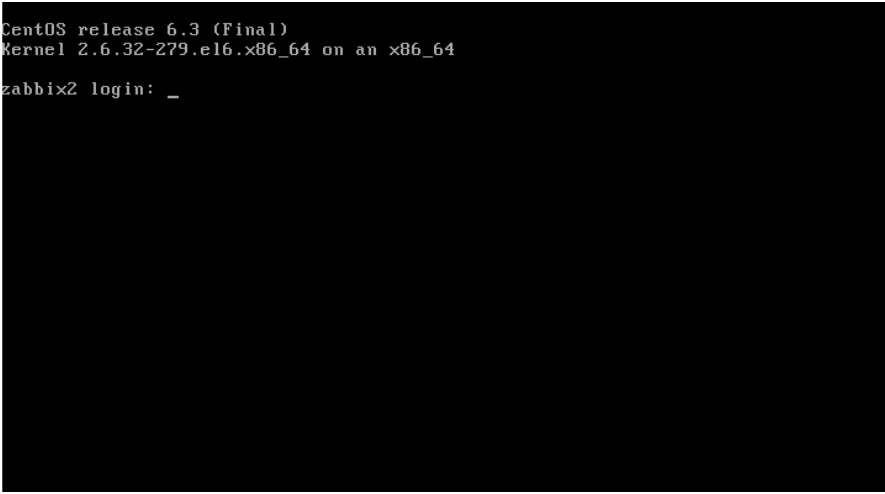
Avendo scelto di installare la versione minimal della distribuzione questa fase dura solo pochi secondi.



Terminata la copia dei pacchetti il sistema è installato e viene richiesto di effettuare un reboot della macchina.



Dopo il riavvio il sistema carica il nuovo sistema operativo e si è subito pronti per procedere con le successive fasi di installazione e configurazione dei software necessari.

A terminal window showing the boot process of CentOS 6.3. The text displayed is: "CentOS release 6.3 (Final)", "Kernel 2.6.32-279.el6.x86_64 on an x86_64", and "zabbix2 login: _". The rest of the terminal is blacked out.

```
CentOS release 6.3 (Final)
Kernel 2.6.32-279.el6.x86_64 on an x86_64
zabbix2 login: _
```

Per iniziare a lavorare con un sistema sicuramente aggiornato, al primo login è consigliabile eseguire immediatamente il comando:

```
yum -y update
```

3.5 Installazione dei requisiti software

Come già anticipato, Zabbix richiede la presenza di alcuni requisiti software per funzionare.

I più importanti consistono sicuramente in uno dei motori di database supportati, nel server http Apache, e nel linguaggio di scripting server-side PHP con relative estensioni.

Il motore di database scelto è MySQL, nella sua ultima versione disponibile nei repository standard della distribuzione CentOS.

Inoltre, vengono anche installati tutti quegli altri software e librerie necessarie per attivare le funzionalità più avanzate di Zabbix, quali il supporto a IPMI, le notifiche Jabber, il monitoraggio di applicazioni web e Java, l'esecuzione remota di comandi tramite SSH.

Di seguito viene mostrato quali comandi eseguire per installare le singole componenti.

3.5.1 MySQL Server

```
[root@zabbix2 ~]# yum -y install mysql.x86_64 mysql-devel.x86_64 \
mysql-embedded-devel.x86_64 mysql-libs.x86_64 mysql-server.x86_64

Installed:
mysql.x86_64 0:5.1.61-4.el6
mysql-devel.x86_64 0:5.1.61-4.el6
mysql-embedded-devel.x86_64 0:5.1.61-4.el6
mysql-libs.x86_64 0:5.1.61-4.el6
mysql-server.x86_64 0:5.1.61-4.el6

Dependency Installed:
keyutils-libs-devel.x86_64 0:1.4-4.el6
krb5-devel.x86_64 0:1.9-33.el6
libcom_err-devel.x86_64 0:1.41.12-12.el6
libselinux-devel.x86_64 0:2.0.94-5.3.el6
libsepol-devel.x86_64 0:2.0.41-4.el6
mysql-embedded.x86_64 0:5.1.61-4.el6
openssl-devel.x86_64 0:1.0.0-20.el6_2.5
perl.x86_64 4:5.10.1-127.el6
perl-DBD-MySQL.x86_64 0:4.013-3.el6
perl-DBI.x86_64 0:1.609-4.el6
perl-Module-Pluggable.x86_64 1:3.90-127.el6
perl-Pod-Escapes.x86_64 1:1.04-127.el6
perl-Pod-Simple.x86_64 1:3.13-127.el6
perl-libs.x86_64 4:5.10.1-127.el6
perl-version.x86_64 3:0.77-127.el6
pkgconfig.x86_64 1:0.23-9.1.el6
zlib-devel.x86_64 0:1.2.3-27.el6
```

3.5.2 Apache e PHP

```
[root@zabbix2 ~]# yum -y install httpd.x86_64 httpd-tools.x86_64 \
php.x86_64 php-bcmath.x86_64 php-common.x86_64 php-gd.x86_64 \
php-xml.x86_64 php-mbstring.x86_64 php-mysql.x86_64 php-cli.x86_64

Installed:
httpd.x86_64 0:2.2.15-15.el6.centos.1
httpd-tools.x86_64 0:2.2.15-15.el6.centos.1
php.x86_64 0:5.3.3-14.el6_3
php-bcmath.x86_64 0:5.3.3-14.el6_3
php-cli.x86_64 0:5.3.3-14.el6_3
php-common.x86_64 0:5.3.3-14.el6_3
php-gd.x86_64 0:5.3.3-14.el6_3
php-mbstring.x86_64 0:5.3.3-14.el6_3
php-mysql.x86_64 0:5.3.3-14.el6_3
php-xml.x86_64 0:5.3.3-14.el6_3

Dependency Installed:
apr.x86_64 0:1.3.9-5.el6_2
apr-util.x86_64 0:1.3.9-3.el6_0.1
apr-util-ldap.x86_64 0:1.3.9-3.el6_0.1
freetype.x86_64 0:2.3.11-6.el6_2.9
libX11.x86_64 0:1.3-2.el6
libX11-common.noarch 0:1.3-2.el6
libXau.x86_64 0:1.0.5-1.el6
libXpm.x86_64 0:3.5.8-2.el6
libjpeg.x86_64 0:6b-46.el6
```

```
libpng.x86_64 2:1.2.49-1.el6_2
libxcb.x86_64 0:1.5-1.el6
libxslt.x86_64 0:1.1.26-2.el6
mailcap.noarch 0:2.1.31-2.el6
php-pdo.x86_64 0:5.3.3-14.el6_3
```

3.5.3 Ambiente di sviluppo

Poiché si andrà ad installare Zabbix a partire direttamente dai sorgenti, è necessario installare anche un set minimo di tool di sviluppo per la compilazione dei sorgenti scritti in linguaggio C.

```
[root@zabbix2 ~]# yum -y install gcc make autoconf

Installed:
  autoconf.noarch 0:2.63-5.1.el6
  gcc.x86_64 0:4.4.6-4.el6
  make.x86_64 1:3.81-20.el6

Dependency Installed:
  cloog-ppl.x86_64 0:0.15.7-1.2.el6
  cpp.x86_64 0:4.4.6-4.el6
  glibc-devel.x86_64 0:2.12-1.80.el6_3.3
  glibc-headers.x86_64 0:2.12-1.80.el6_3.3
  kernel-headers.x86_64 0:2.6.32-279.2.1.el6
  libgomp.x86_64 0:4.4.6-4.el6
  mpfr.x86_64 0:2.4.1-6.el6
  ppl.x86_64 0:0.10.2-11.el6
```

3.5.4 Librerie per controlli IPMI, SSH e HTTP

```
[root@zabbix2 ~]# yum -y install OpenIPMI-devel.x86_64 \
  OpenIPMI-libs.x86_64 OpenIPMI.x86_64 libssh2.x86_64 \
  libssh2-devel.x86_64 libcurl.x86_64 libcurl-devel.x86_64

Installed:
  OpenIPMI.x86_64 0:2.0.16-12.el6
  OpenIPMI-devel.x86_64 0:2.0.16-12.el6
  OpenIPMI-libs.x86_64 0:2.0.16-12.el6
  libcurl-devel.x86_64 0:7.19.7-26.el6_2.4
  libssh2-devel.x86_64 0:1.2.2-11.el6_3

Dependency Installed:
  automake.noarch 0:1.11.1-1.2.el6
  libidn-devel.x86_64 0:1.18-2.el6
  lm_sensors-libs.x86_64 0:3.1.1-10.el6
  ncurses-devel.x86_64 0:5.7-3.20090208.el6
  net-snmp-libs.x86_64 1:5.5-41.el6_3.1
```

3.5.5 Network tools

```
[root@zabbix2 zabbix-2.0.1]# yum -y install traceroute nmap

Installed:
  nmap.x86_64 2:5.51-2.el6
  traceroute.x86_64 3:2.0.14-2.el6

Dependency Installed:
  libpcap.x86_64 14:1.0.0-6.20091201git117cb5.el6

[root@zabbix2 ~]# rpm -Uvh \
http://dl.fedoraproject.org/pub/epel/6/x86_64/fping-2.4b2-10.el6.x86_64.rpm
```

3.5.6 Supporto Jabber

```
[root@zabbix2 ~]# yum -y install gnutls-devel.x86_64 gnutls.x86_64 \
gnutls-utils.x86_64

Installed:
  gnutls.x86_64 0:2.8.5-4.el6_2.2
  gnutls-devel.x86_64 0:2.8.5-4.el6_2.2
  gnutls-utils.x86_64 0:2.8.5-4.el6_2.2

Dependency Installed:
  libgcrypt-devel.x86_64 0:1.4.5-9.el6_2.2
  libgpg-error-devel.x86_64 0:1.7-4.el6
  libtasn1.x86_64 0:2.3-3.el6_2.1

[root@zabbix2 ~]# rpm -Uvh \
http://dl.atrpms.net/el6-x86_64/atrpms/stable/libiksemel3-1.4-
2_2.el6.x86_64.rpm

[root@zabbix2 ~]# rpm -Uvh \
http://dl.atrpms.net/el6-x86_64/atrpms/stable/iksemel-devel-1.4-
2_2.el6.x86_64.rpm

[root@zabbix2 ~]# rpm -Uvh \
http://dl.atrpms.net/el6-x86_64/atrpms/stable/iksemel-1.4-
2_2.el6.x86_64.rpm

[root@zabbix2 ~]# rpm -Uvh \
http://dl.atrpms.net/el6-x86_64/atrpms/stable/iksemel-utils-1.4-
2_2.el6.x86_64.rpm
```

3.5.7 Supporto Java

```
[root@zabbix2 ~]# yum -y install java-1.7.0-openjdk.x86_64 \
  java-1.7.0-openjdk-devel.x86_64

Installed:
  java-1.7.0-openjdk.x86_64 1:1.7.0.5-2.2.1.el6_3
  java-1.7.0-openjdk-devel.x86_64 1:1.7.0.5-2.2.1.el6_3

Dependency Installed:
  alsa-lib.x86_64 0:1.0.22-3.el6
  flac.x86_64 0:1.2.1-6.1.el6
  giflib.x86_64 0:4.1.6-3.1.el6
  jline.noarch 0:0.9.94-0.8.el6
  jpackage-utils.noarch 0:1.7.5-3.12.el6
  libICE.x86_64 0:1.0.6-1.el6
  libSM.x86_64 0:1.1.0-7.1.el6
  libXext.x86_64 0:1.1-3.el6
  libXi.x86_64 0:1.3-3.el6
  libXrender.x86_64 0:0.9.5-1.el6
  libXtst.x86_64 0:1.0.99.2-3.el6
  libasyncns.x86_64 0:0.8-1.1.el6
  libogg.x86_64 2:1.1.4-2.1.el6
  libsndfile.x86_64 0:1.0.20-5.el6
  libvorbis.x86_64 1:1.2.3-4.el6_2.1
  pulseaudio-libs.x86_64 0:0.9.21-14.el6_3
  rhino.noarch 0:1.7-0.7.r2.2.el6
  tzdata-java.noarch 0:2012c-3.el6
```

3.6 Configurazione

Installate le dipendenze software, prima di installare Zabbix è necessario predisporre il sistema affinché venga garantita la corretta esecuzione dei vari moduli di Zabbix.

3.6.1 Creazione utente non privilegiato

Anche per ragioni di sicurezza, è altamente sconsigliato eseguire i processi di Zabbix con privilegi amministrativi (utente *root*) e quindi viene utilizzato un utente creato ad-hoc:

```
[root@zabbix2 ~]# groupadd zabbix
[root@zabbix2 ~]# useradd -g zabbix zabbix
```

3.6.2 Attivazione di MySQL

Il motore di database MySQL va attivato sin dalla fase di avvio del sistema e bisogna impostare una password per l'utente *root*, privilegiato anche in ambito database:

```
[root@zabbix2 ~]# chkconfig mysqld on
[root@zabbix2 ~]# service mysqld start
[root@zabbix2 ~]# /usr/bin/mysqladmin -u root password 'new-password'
[root@zabbix2 ~]# /usr/bin/mysqladmin -u root -h zabbix2 password \
'new-password'
```

3.6.3 Configurazione di PHP

Le impostazioni predefinite riguardanti PHP possono andare bene per un utilizzo generale, ma non per le necessità del frontend di Zabbix.

```
date.timezone = Europe/Rome
post_max_size = 32M
memory_limit = 256M
upload_max_filesize = 16M
max_execution_time = 600
max_input_time = 600
```

3.6.4 Configurazione di Apache

Apache non richiede una configurazione particolare se non la sua attivazione al boot del sistema.

```
[root@zabbix2 ~]# chkconfig httpd on
[root@zabbix2 ~]# service httpd start
```

Può essere utile aggiungere semplicemente *index.php* alla direttiva *DirectoryIndex* presente già nel file di configurazione *httpd.conf*:

```
DirectoryIndex index.html index.html.var index.php
```

3.6.5 Configurazione del firewall locale

Il firewall locale iptables viene attivato di default all'avvio del sistema, ma le sue impostazioni predefinite non consentono connessioni TCP in entrata diverse da quelle dirette sulla porta 22, relative al servizio SSH.

Per fruire del frontend web based bisogna abilitare almeno la porta 80, standard per il servizio HTTP erogato dal server web Apache.

Inoltre, bisogna consentire che gli agent installati sugli host eventualmente da monitorare possano colloquiare direttamente con il server Zabbix. La porta TCP utilizzata allo scopo è la 10051.

Per rendere permanenti tali impostazioni bisogna intervenire sul file di configurazione */etc/syslinux/iptables*, aggiungendo le seguenti due righe:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 10051 -j ACCEPT
```

Per attivarle immediatamente sarà sufficiente eseguire il comando:

```
service iptables restart
```

3.6.6 Disattivazione servizi non necessari

Nella distribuzione linux CentOS utilizzata, così come d'altra parte anche nella Red Hat Enterprise Linux dalla quale deriva, come impostazione predefinita sono attivati alcuni processi relativi ai servizi RPC e NFS.

In questo ambito non sono necessari e quindi si può procedere tranquillamente alla loro disattivazione.

```
[root@zabbix2 ~]# service rpcbind stop
Stopping rpcbind: [ OK ]
[root@zabbix2 ~]# chkconfig rpcbind off
[root@zabbix2 ~]# chkconfig rpcgssd off
[root@zabbix2 ~]# service rpcidmapd stop
Stopping RPC idmapd: [ OK ]
[root@zabbix2 ~]# chkconfig rpcidmapd off
[root@zabbix2 ~]# service nfslock stop
Stopping NFS statd: [ OK ]
[root@zabbix2 ~]# chkconfig nfslock off
```

4 Installazione di Zabbix

4.1 Introduzione

Installato il sistema operativo, soddisfatte le dipendenze software ed effettuate le necessarie configurazioni preliminari, si può procedere alla installazione di Zabbix.

Pur essendo spesso disponibile come pacchetto binario scaricabile all'interno dei repository di riferimento della distribuzione linux scelta, normalmente è preferibile compilare direttamente i sorgenti prelevabili dal sito <http://www.zabbix.com>.

In tal modo si è certi di lavorare con l'ultima versione rilasciata ed è possibile effettuarne le eventuali dovute personalizzazioni in base alle particolari esigenze, cosa non sempre possibile se piuttosto si opta per la pacchettizzazione reperibile nei repository della distribuzione assolutamente generica in termini di funzionalità.

Quindi, scaricato il pacchetto in formato `.tar.gz`, si può direttamente procedere alla estrazione dei suoi contenuti ed a impostare la nuova directory corrente di lavoro:

```
[root@zabbix2 ~]# tar xzf zabbix-2.0.1.tar.gz
[root@zabbix2 ~]# cd zabbix-2.0.1
```

4.2 Compilazione dei sorgenti

La fase di compilazione, come da consuetudine consolidata per la stragrande maggioranza dei moderni software unix/linux open source distribuiti in formato sorgente, deve essere preceduta da una preliminare procedura di *configurazione* automatica.

Tramite tale attività viene generato un apposito Makefile contenente le necessarie dipendenze in termini di file include e librerie da utilizzare per la generazione dei binari, e ciò in funzione delle funzionalità che si intende attivare.

La configurazione va effettuata eseguendo il comando `configure`, facendolo seguire da una serie di *switch* di attivazione, ognuno corrispondente ad una specifica funzionalità altrimenti non disponibile in configurazione standard.

Ad esempio, per attivare la generazione dei binari del server e dell'agent, l'utilizzo di MySQL come database di backend, il supporto a IPv6, SNMP, HTTP, SSH e IPMI e alle notifiche tramite Jabber, avremo:

```
[root@zabbix2 zabbix-2.0.1]# ./configure --enable-server --enable-agent \
--with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --with-ssh2 \
--enable-java --with-jabber --with-openipmi

Configuration:

Detected OS:          linux-gnu
Install path:        /usr/local
Compilation arch:    linux

Compiler:            gcc
Compiler flags:      -g -O2 -I/usr/include/mysql -g -pipe
                    -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector
                    --param=ssp-buffer-size=4 -m64 -D_GNU_SOURCE -D_FILE_OFFSET_BITS=64
                    -D_LARGEFILE_SOURCE -fno-strict-aliasing -fwrapv -fPIC -DUNIV_LINUX
                    -DUNIV_LINUX -I/usr/include/rpm -I/usr/local/include
                    -I/usr/lib64/perl5/CORE -I. -I/usr/include -I/usr/include
                    -I/usr/include

Enable server:       yes
Server details:
  With database:     MySQL
  WEB Monitoring via:  cURL
  Native Jabber:     yes
  SNMP:              net-snmp
  IPMI:              openipmi
  SSH:               yes
  Linker flags:      -rdynamic -L/usr/lib64/mysql
                    -L/usr/lib64 -L/usr/lib64 -L/usr/lib -L/usr/lib
  Libraries:         -lm -lrt -lresolv -lmysqlclient -liksemel
                    -lcurl -lnetsnmp -lcrypto -lnetsnmp -lcrypto -lssh2 -lOpenIPMI
                    -lOpenIPMIposix

Enable proxy:       no

Enable agent:       yes
Agent details:
  Linker flags:      -rdynamic
  Libraries:         -lm -lrt -lresolv -lcurl

Enable Java gateway:  yes
Java gateway details:
  Java compiler:     javac
  Java archiver:     jar

LDAP support:       no
IPv6 support:       yes

*****
*               Now run 'make install'               *
*                                                       *
*               Thank you for using Zabbix!           *
*               <http://www.zabbix.com>              *
*****
```

Infine con `make install` si otterrà la compilazione e l'installazione di Zabbix.

4.3 Predisposizione del database

Avendo scelto MySQL come dbms di backend per Zabbix, è necessario innanzitutto creare un database ed un utente che abbia i privilegi per accedervi e effettuare le modifiche:

```
[root@zabbix2 zabbix-2.0.1]# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'zabbix';
Query OK, 0 rows affected (0.00 sec)

mysql> create database zabbix character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Successivamente si può procedere con la creazione dello schema del database, con la definizione delle tabelle e il caricamento di alcuni dati iniziali necessari.

A tale scopo all'interno della directory *database* si possono trovare gli script sql da utilizzare in relazione al motore di database scelto.

Per MySQL avremo:

```
[root@zabbix2 zabbix-2.0.1]# mysql -uzabbix -pzabbix zabbix \  
< database/mysql/schema.sql  
[root@zabbix2 zabbix-2.0.1]# mysql -uzabbix -pzabbix zabbix \  
< database/mysql/images.sql  
[root@zabbix2 zabbix-2.0.1]# mysql -uzabbix -pzabbix zabbix \  
< database/mysql/data.sql
```

4.3.1 Dimensioni del Database

I dati di configurazione di Zabbix richiedono una quantità prefissata di spazio su disco e normalmente non variano molto.

La dimensione del database dipende principalmente dalle seguenti variabili, che definiscono l'ammontare di dati storici da memorizzare:

- **Numero di valori elaborati al secondo.**

Questo è il numero medio di valori nuovi che Zabbix riceve ogni secondo.

Ad esempio, se si hanno 3000 item da monitorare con una frequenza di refresh pari a 60 secondi, il numero di valori al secondo sarà pari a $3000/60 = 50$.

Ciò significa che 50 nuovi valori vengono aggiunti al database di Zabbix ogni secondo.

- **Impostazioni storico history.**

Zabbix mantiene memorizzati i valori per un periodo di tempo prefissato, normalmente diverse settimane o mesi. Ogni nuovo valore richiede una certa quantità di spazio disco per il dato stesso e l'indice.

Quindi, se volessimo mantenere 30 giorni di history e si ricevono 50 valori al secondo, il numero totale di valori sarà $(30*24*3600)*50 = 129.600.000$, cioè circa 130 milioni di valori.

A seconda del motore di database utilizzato e il tipo di valori ricevuti (float, interi, stringhe, file di log, ecc), lo spazio su disco per mantenere un singolo valore può variare da 40 a diverse centinaia di bytes. Normalmente è di circa 50 bytes per valore.

Nel nostro caso di esempio, significa che 130 milioni di valori richiederanno $130M*50 \text{ bytes} = 6.5 \text{ GBytes}$ di spazio su disco.

- **Impostazioni storico trend.**

Zabbix mantiene nella tabella *trends* per ogni item i valori massimi, minimi, medi e contatore relativamente all'ultima ora. I dati vengono utilizzati per le analisi di trend ed i grafici di lungo periodo. Tale periodo pari a 1 ora non può essere personalizzato. Il database di Zabbix, a seconda del tipo di database, richiede circa 128 byte per ogni n-pla.

Se supponiamo di voler mantenere i dati di trend per 5 anni, i valori per 3000 item richiederanno $3000*24*365*128 = 3.4 \text{ GBytes}$ per anno, quindi complessivamente 16.8 GBytes per 5 anni.

- **Impostazioni storico eventi.**

Ogni evento richiede approssimativamente circa 130 bytes di spazio disco, ma è molto difficile poter stimare il numero di eventi generati da Zabbix giornalmente. Nel caso peggiore, si può assumere che Zabbix generi un evento al secondo. Questo significa che se vogliamo mantenere 3 anni di eventi, lo spazio richiesto sarà pari a $3*365*24*3600*130 = 12.3$ GBytes.

La tabella seguente contiene le formule che possono essere utilizzate per calcolato lo spazio disco complessivo richiesto da un sistema Zabbix.

Parametro	Formula per lo spazio richiesto (in bytes)
Configurazione	Dimensione fissa. Normalmente 10 MBytes o meno.
History	$days * (items / refresh\ rate) * 24 * 3600 * bytes$ items: numero di item days: numero di giorni da mantenere nell'history refresh rate: frequenza di aggiornamento media bytes: numero di bytes richiesti per memorizzare un singolo valore, dipende dal motore database, normalmente pari a 50 bytes.
Trends	$days * (items / 3600) * 24 * 3600 * bytes$ items: numero di item days: numero di giorni da mantenere nell'history bytes: numero di bytes richiesti per memorizzare un singolo valore di trend, dipende dal motore database, normalmente pari a 128 bytes.
Eventi	$days * events * 24 * 3600 * bytes$ events: numero di eventi al secondo. Uno (1) nel caso di scenario peggiore. days: numero di giorni da mantenere nell'history bytes: numero di bytes richiesti per memorizzare un singolo evento, dipende dal motore database, normalmente pari a 130 bytes.

In definitiva quindi, lo spazio totale richiesto sarà pari a:

configurazione + history + trends + eventi

4.4 Configurazione degli script di avvio automatico

Poiché è necessario che il processo server e l'agent locale di Zabbix siano attivi all'avvio del sistema, se ne deve abilitare l'attivazione automatica.

Per far fronte a questa esigenza, all'interno della directory *misc/init.d* sono presenti degli script da utilizzare con le varie distribuzioni di Linux. Nel nostro caso è possibile utilizzare direttamente e senza alcuna modifica gli script realizzati per la distribuzione Fedora e che vanno copiati nella directory */etc/init.d*.

Fatto ciò, è sufficiente aggiungere e attivare i due nuovi servizi *zabbix_server* e *zabbix_agentd*.

```
[root@zabbix2 zabbix-2.0.1]# cp misc/init.d/fedora/core5/zabbix_agentd \
/etc/init.d/
[root@zabbix2 zabbix-2.0.1]# cp misc/init.d/fedora/core5/zabbix_server \
/etc/init.d/
[root@zabbix2 zabbix-2.0.1]# chkconfig --add zabbix_server
[root@zabbix2 zabbix-2.0.1]# chkconfig --add zabbix_agentd
[root@zabbix2 zabbix-2.0.1]# chkconfig zabbix_server on
[root@zabbix2 zabbix-2.0.1]# chkconfig zabbix_agentd on
```

4.5 Installazione dell'interfaccia web

L'installazione del front end web-based consiste nella semplice copia del contenuto della directory *frontends/php* all'interno della directory *root* del web server.

L'unico accorgimento da seguire, eseguendo la suddetta operazione di copia come utente *root*, è quello di impostare l'utente non privilegiato con cui viene eseguito il server Apache come proprietario dei file e cartelle del front end, di modo che possa essere generato senza errori il file di configurazione per l'interfaccia PHP.

```
[root@zabbix2 zabbix-2.0.1]# mkdir /var/www/html/zabbix
[root@zabbix2 zabbix-2.0.1]# cp -a frontends/php/* /var/www/html/zabbix/
[root@zabbix2 zabbix-2.0.1]# chown -R apache:apache /var/www/html/zabbix
```

4.6 Avvio dei servizi

Avendo predisposto tutto quanto correttamente e senza errori, si possono mandare in esecuzione i servizi server ed agent utilizzando i comandi standard di sistema:

```
[root@zabbix2 zabbix-2.0.1]# service zabbix_server start
[root@zabbix2 zabbix-2.0.1]# service zabbix_agentd start
```

4.7 Configurazione del frontend

L'ultima operazione da compiere prima di poter fruire appieno delle potenzialità di Zabbix, ed in particolare della sua interfaccia basata su web, è la configurazione del frontend.

Tale operazione verrà avviata automaticamente non appena si richiamerà per la prima volta la url del frontend:

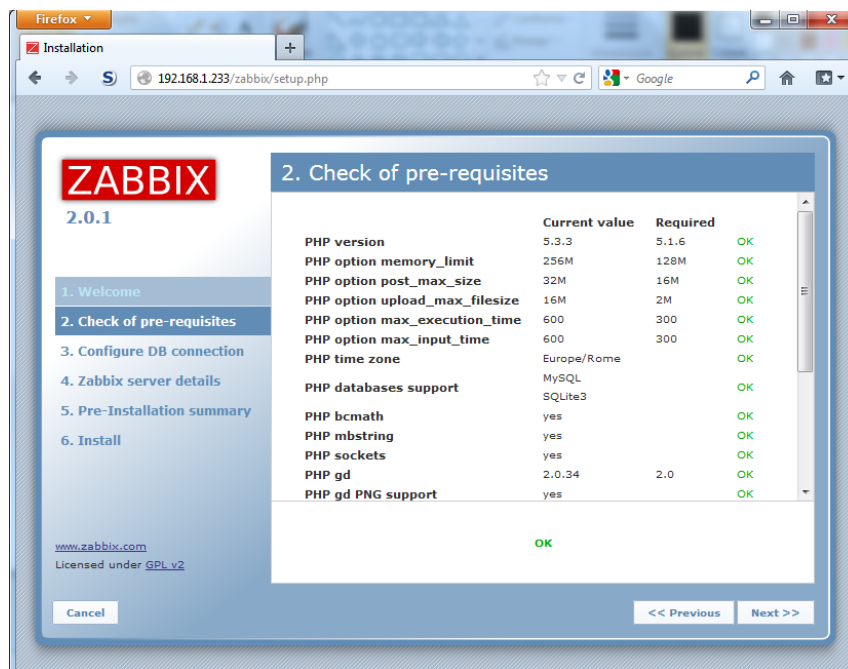
```
http://<nome server>/zabbix
```

ed avverrà mediante procedura guidata.



4.7.1 Verifica dei prerequisiti

Il primo step della procedura di configurazione del frontend è la verifica dei prerequisiti, che riguardano sostanzialmente le impostazioni di configurazione dell'ambiente di scripting server-side PHP, al fine di garantire una corretta esecuzione di tutte le componenti del frontend. Avendo già configurato opportunamente PHP in fase di setup del sistema e installato tutte le estensioni necessarie, la verifica dei prerequisiti dovrebbe immediatamente dare esito positivo.



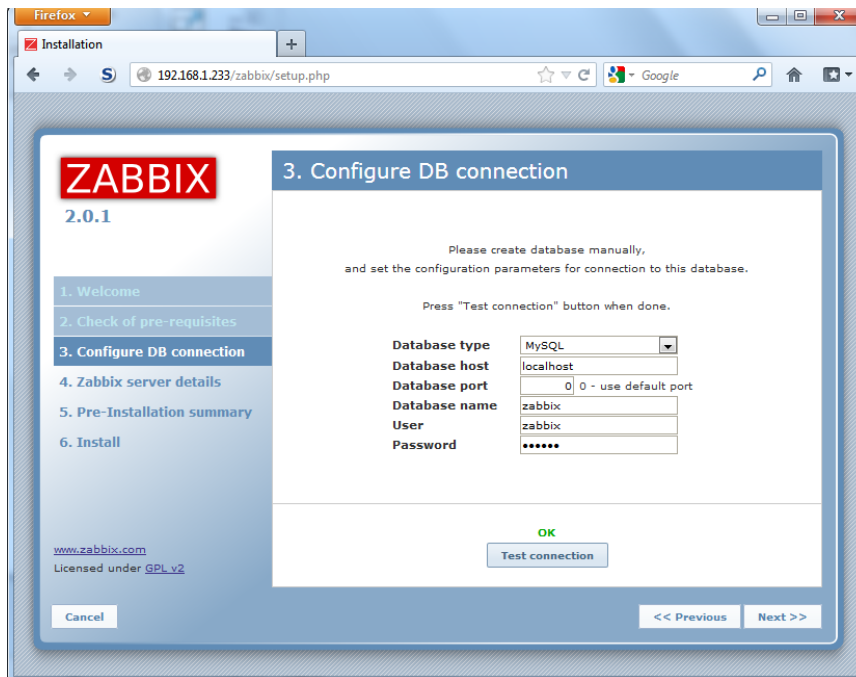
4.7.2 Configurazione e test della connessione al database

Il passo successivo consiste nella configurazione della connessione al database. In tal modo il frontend PHP accederà direttamente al database di backend utilizzato dal processo server e ne potrà estrarre o modificare i contenuti per effettuarne la visualizzazione, per la produzione dei report, per la configurazione degli oggetti da porre sotto monitoraggio, etc.

Le informazioni richieste in questo step riguardano:

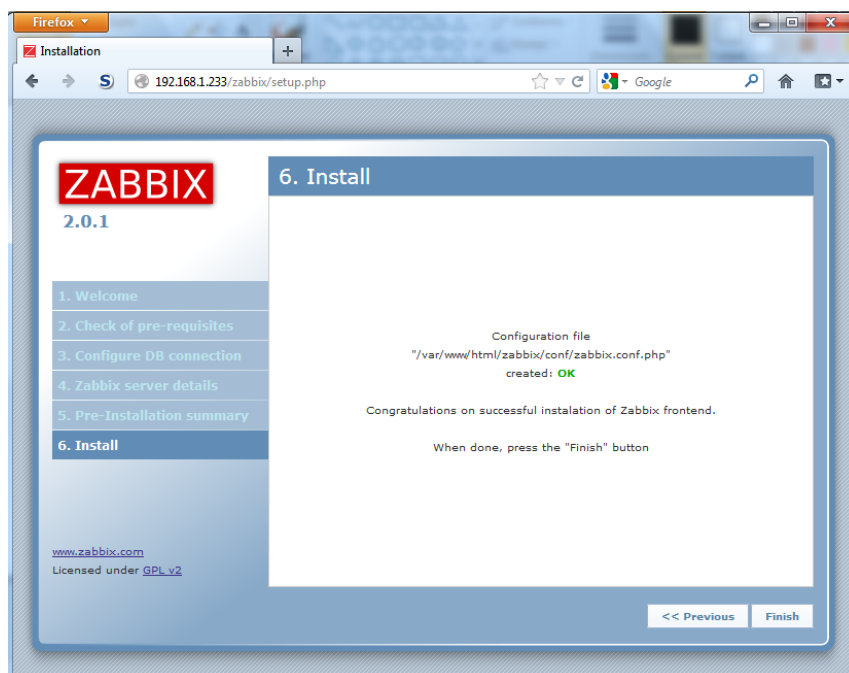
- il tipo di motore dbms;
- l'indirizzo o nome del server che lo ospita;
- la porta sulla quale il database è in attesa di connessioni;
- il nome del database utilizzato da zabbix;
- le credenziali dell'utente abilitato ad accedere al database.

Impostate tale informazioni, è possibile immediatamente effettuare un test della connessione al fine di verificarne la correttezza.



4.7.3 Generazione del file di configurazione del frontend

L'ultimo passo consiste nella generazione del file di configurazione del frontend, operazione anch'essa operazione del tutto automatica.



4.8 Termine dell'installazione

Generato il file di configurazione e cliccato sul bottone *Finish*, si viene automaticamente rediretti alla schermata di login del frontend.



Utilizzando come username **Admin** e password **zabbix** si ottiene l'accesso come *Zabbix superuser*. Infatti, nell'angolo in basso a destra della pagina verrà mostrato il messaggio '*Connected as Admin*' e viene consentito l'accesso alle voci di menu *Configuration* e *Administration*.

5 Guida rapida all'uso di Zabbix

5.1 Introduzione

Avendo adesso un sistema Zabbix installato e funzionante, saranno illustrate le operazioni fondamentali per poterlo iniziare ad utilizzare.

Verrà descritto come creare nuovi utenti, assegnando loro permessi e associandovi i meccanismi di notifica, come creare un nuovo host, come aggiungere un item ad un host, come creare un trigger, come avviene la ricezione delle notifiche di un problema e infine come creare un nuovo template.

Per quanto non descritto nel presente capitolo e per la descrizione delle funzionalità più avanzate si faccia riferimento alla documentazione ufficiale reperibile all'indirizzo <http://www.zabbix.com/documentation/2.0/manual/>.

5.2 Creazione di un utente

L'amministrazione degli utenti si effettua mediante la voce di menu *Administration* → *Users* e selezionando nella dropdown la voce *Users*.

<input type="checkbox"/>	Alias	Name	Surname	User type	Groups	Is online?	Login	Frontend access	Debug mode	Status
<input type="checkbox"/>	Admin	Zabbix	Administrator	Zabbix Super Admin	Zabbix administrators	Yes (Wed, 04 Jan 2012 15:39:51 +0200)	Ok	System default	Disabled	Enabled
<input type="checkbox"/>	guest	Default	User	Zabbix User	Guests	Yes (Wed, 04 Jan 2012 15:33:42 +0200)	Ok	System default	Disabled	Enabled

Inizialmente vi sono soltanto due utenti definiti in Zabbix:

- *Admin* è il superuser che ha pieni permessi su qualunque oggetto
- *guest* è un utente speciale che viene assegnato quando si accede al frontend senza autenticazione. Come impostazione predefinita guest non ha alcun permesso sugli oggetti di Zabbix.

Per aggiungere un nuovo utente bisogna cliccare sulla voce *Create user*.

Nella form che viene presentata bisogna accertarsi di associare l'utente che si sta creando ad almeno uno dei gruppi utente esistenti, ad esempio *Network administrator*.

The screenshot shows a user configuration window with three tabs: 'User', 'Media', and 'Permissions'. The 'User' tab is active. The fields are as follows:

- Alias: user
- Name: New
- Surname: User
- Password: [masked]
- Password (once again): [masked]
- Groups: Network administrators
- Language: English (en_GB)
- Theme: System default
- Auto-login:
- Auto-logout (min 90 seconds): 90
- Refresh (in seconds): 30
- Rows per page: 50
- URL (after login): [empty]

Buttons at the bottom: Save, Cancel.

5.2.1 Associazione di un *media*

Inoltre, per default, i nuovi utenti non hanno alcun *media* (metodo di invio delle notifiche) associato. Per crearne uno bisogna andare sul tab *Media* e cliccare su *Add*.

The 'New media' dialog box contains the following configuration options:

- Type: Email
- Send to: user@domain.tld
- When active: 1-7,00:00-24:00
- Use if severity:
 - Not classified
 - Information
 - Warning
 - Average
 - High
 - Disaster
- Status: Enabled

Buttons at the bottom: Add, Cancel.

Nella finestra pop-up che appare, selezionando ad esempio *Email* come tipo, è sufficiente inserire l'indirizzo e-mail dell'utente.

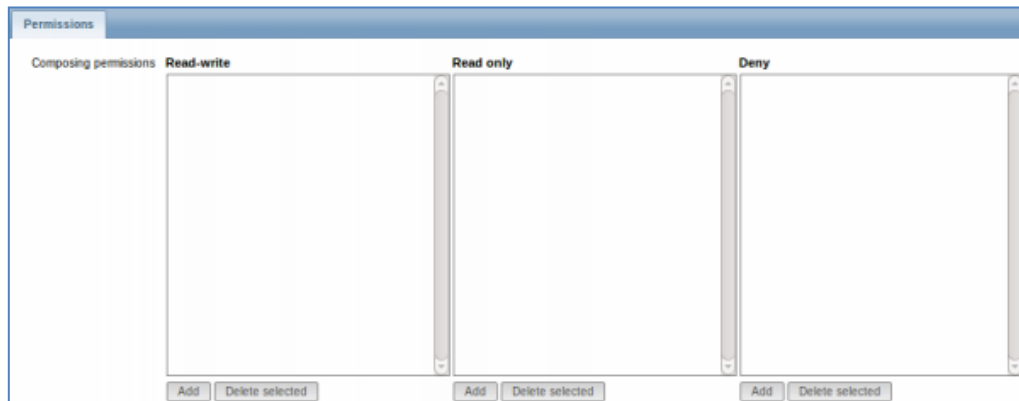
Inoltre è possibile specificare in quale periodo di tempo il *media* sarà attivo e associarvi particolari livelli di severità.

Cliccando quindi su *Add* e poi su *Save* nella maschera di creazione dell'utente, il nuovo utente apparirà nella lista utenti.

5.2.2 Assegnazione dei privilegi

Bisogna tenere presente che, per default, un nuovo utente non ha alcun permesso per accedere agli host e che i permessi di accesso agli host vengono assegnati ai gruppi di utenti, non ai singoli utenti.

Per dare quindi ad un utente i privilegi desiderati, dalla lista degli utenti si dovrà intanto cliccare sul nome del gruppo di appartenenza dell'utente (ad esempio: *Network administrators*) e dalla maschera di proprietà del gruppo scegliere il tab *Permissions*.



Volendo dare, ad esempio, un accesso in sola lettura al gruppo *Linux servers*, si dovrà cliccare su *Add* sotto la lista *Read only* e, nella finestra pop-up che apparirà, mettere un segno di spunta accanto a *Linux server* e premere *Select*.

A questo punto la voce *Linux server* apparirà nella lista degli oggetti a sola lettura e cliccando su *Save* nella form di proprietà del gruppo verranno memorizzate le modifiche effettuate.

5.3 Creazione di un host

Un host in Zabbix è un'entità di rete, fisica o virtuale, che si desidera monitorare. La definizione di cosa può essere un host in Zabbix è alquanto flessibile. Può essere, infatti, un server fisico, uno switch di rete, una macchina virtuale o una qualche applicazione.

Le informazioni sugli host configurati sono disponibili sotto la voce *Configuration* → *Hosts* e qui inizialmente vi è già presente un host predefinito, denominato *Zabbix server*.

Per aggiungerne uno nuovo, si deve cliccare su *Create* e verrà presentato un modulo dove inserire i dati di configurazione per il nuovo host.

Qui bisogna inserire almeno le seguenti informazioni:

- **Host name**, sarà il nome che identifica l'host e può contenere caratteri alfanumerici, spazi e underscore.
- **Groups**, uno o più gruppi cui apparterrà l'host. Poiché i permessi di accesso sono assegnati ai gruppi e non ai singoli host, qui deve essere selezionato almeno un gruppo.
- **IP address**, indirizzo IP dell'host.

Cliccando su *Save*, il nuovo host sarà visibile nella lista degli host.

Se l'icona con la lettera **Z** nella colonna *Availability* appare di colore rosso, vorrà dire che si è presentato un qualche errore nella comunicazione con l'host ed è possibile leggere il messaggio di errore spostandosi con il puntatore del mouse sopra l'icona stessa.

5.4 Creazione di un item

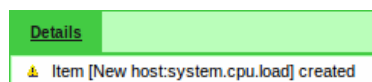
Gli item sono alla base della raccolta dei dati in Zabbix. Senza item non esistono dati, perché solo un item può definire una metrica o quale dato prelevare da un host.

Per aggiungere un item, si deve andare su *Configuration* → *Hosts*, cliccare sul link *Item* nella riga relativa all'host di interesse ed infine cliccare su *Create item*.

Ad esempio, volendo aggiungere un item che riguardi il carico della CPU, le informazioni essenziali da inserire sono:

- **Name**, sarà il nome dell'item visualizzato nelle liste e altrove.
- **Key**, è il nome tecnico di un item che identifica il tipo di informazione che verrà raccolta. *system.cpu.load* è solo una delle tante key predefinite nell'agent di Zabbix.
- **Type of information**, definisce il formato dei dati attesi.

Cliccando su *Save* il nuovo item comparirà nella lista degli item e la voce *Details* sopra la lista darà conferma di cosa è stato fatto esattamente.



5.4.1 Visualizzazione dei dati

Definito un item si può verificare che a questo corrispondano già dei dati raccolti andando su *Monitoring* → *Latest data*:

Name	Last check	Last value	Change	History
- other - (1 items)				
CPU Load	05 Jan 2012 14:48:38	0.47	+0.37	Graph

Prima che arrivi il primo dato potranno passare fino a 60 secondi: tale è infatti la frequenza con cui il server legge le modifiche alla configurazione e prende in considerazione gli eventuali nuovi item aggiunti.

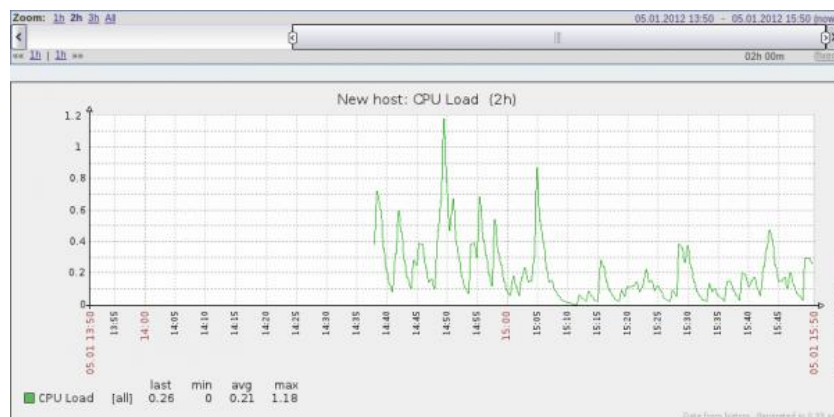
Se la colonna *Change* non riporta alcun valore significa che è stato ricevuto solo un valore e bisogna attendere 30 secondi per l'arrivo del successivo.

5.4.2 Grafici

Con un item definito già da un po' di tempo è possibile visualizzare una rappresentazione grafica della variazione dei suoi valori nel tempo.

A tale scopo si possono utilizzare anche dei semplici grafici predefiniti che sono in grado di visualizzare le informazioni relative a un qualunque item di tipo numerico senza ulteriori configurazioni.

Per visualizzarli è sufficiente aprire la schermata *Monitoring* → *Latest data* e cliccare sulla voce *Graph* accanto all'item di interesse.



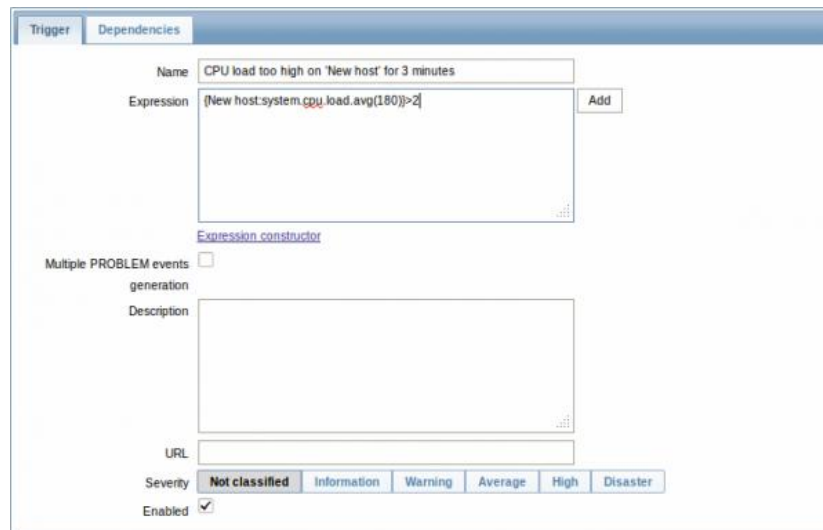
5.5 Creazione di un trigger

Gli item servono solamente a collezionare i dati, ma per effettuarne una valutazione è necessario definire dei trigger.

Un trigger contiene un'espressione per la definizione di un valore di soglia accettabile per un dato.

Se tale limite viene superato da un valore in arrivo, il trigger relativo si attiva e va in stato *Problem*, consentendoci di sapere che è successo qualcosa che merita attenzione. Quando il valore rientra nel livello accettabile, il trigger ritorna in stato *Ok*.

Ad esempio, per configurare un trigger collegato all'item definito precedentemente e contenente il carico della CPU, bisogna cliccare sulla voce *Triggers* presente nella riga relativa all'host di interesse dalla schermata *Configuration* → *Hosts* e poi scegliere la voce *Create trigger*.



Nella maschera di definizione di un nuovo trigger le informazioni basilari da inserire sono:

- **Name**, sarà il nome del trigger visualizzato nelle liste. Nel nostro caso potrà avere come valore *CPU load too high for 3 minutes*.
- **Expression**, è l'espressione del trigger vera e propria. Bisogna accertarsi che l'espressione sia corretta sino all'ultimo simbolo utilizzato.

Si consideri l'espressione $\{New\ host:system.cpu.load.avg(180)\}>2$. Per riferirsi all'item viene utilizzata la key *system.cpu.load* e l'espressione sostanzialmente dice che la soglia di allarme viene superata quando il carico medio della CPU per 3 minuti è superiore a 2.

Dopo aver cliccato su *Save* il nuovo trigger apparirà nella lista dei trigger.

Maggiori dettagli sulla sintassi delle espressioni dei trigger sono disponibili all'indirizzo

<http://www.zabbix.com/documentation/2.0/manual/config/triggers/expression>.

5.5.1 Visualizzare lo stato di un trigger

Lo stato di un trigger può essere visualizzato a partire dalla schermata *Monitoring* → *Triggers*.

Nel nostro caso di esempio, dopo 3 minuti o poco più (visto che l'intervallo inserito nell'espressione è appunto di 3), il nuovo trigger apparirà nella lista dei trigger, presumibilmente con un **OK** verde lampeggiante nella colonna *Status*.

	Severity	Status	Info	Last change ↓	Age	Duration	Acknowledged	Host	Name	Comments
	Not classified	OK		06 Jan 2012 14:06:38	9m 43s		Acknowledged	New host	CPU load too high on 'New host' for 3 minutes	Add

Il lampeggiamento indica un recente cambio di stato di un trigger, avvenuto negli ultimi 30 minuti.

5.6 Ricezione delle notifiche

Avendo già gli item che collezionano i dati ed i trigger impostati per evidenziare eventuali problemi, può essere utile avere un qualche meccanismo di avvertimento che ci possa portare a conoscenza del determinarsi di eventi importanti anche quando non si sta guardando direttamente il front-end di Zabbix

Questo è ciò che fanno le notifiche. Essendo l'e-mail il metodo più comunemente utilizzato per l'invio delle notifiche, si vedrà in questa sede come impostare una notifica via e-mail.

Si tenga anche presente che vi sono anche altri metodi di notifica predefiniti, come l'invio di messaggi tramite protocollo XMPP/Jabber o l'invio di SMS tramite modem GSM collegato ad una linea seriale.

5.6.1 Impostazioni e-mail

Per configurare le impostazioni e-mail bisogna andare nella schermata *Administration* → *Media types* e cliccare sulla voce Email nella lista dei media types predefiniti.

Media types				
Displaying 1 to 3 of 3 found				
<input type="checkbox"/>	Description	Type	Used in actions	Details
<input type="checkbox"/>	Email	Email	-	SMTP server: "mail.company.com", SMTP helo: "company.com", SMTP email: "zabbix@company.com"
<input type="checkbox"/>	Jabber	Jabber	-	Jabber identifier: "jabber@company.com"
<input type="checkbox"/>	SMS	SMS	-	GSM modem: "/dev/ttyS0"

Verrà visualizzata una maschera per la definizione delle impostazioni e-mail. In particolare il campo SMTP email verrà utilizzato come indirizzo mittente nelle notifiche inviate da Zabbix.

Media

Description

Type

SMTP server

SMTP helo

SMTP email

Una volta configurato correttamente il media type e-mail, si può associarlo agli utenti definendo gli specifici indirizzi di destinazione, altrimenti non verrà utilizzato.

5.6.2 Creazione di una nuova azione

L'invio delle notifiche è una delle possibili azioni che Zabbix può compiere al verificarsi di determinate condizioni. Quindi, per impostare una notifica, bisogna andare sulla schermata *Configuration* → *Actions* e cliccare su *Create action*.

Action

Name

Default escalation period (minimum 60 seconds) (seconds)

Default subject

Default message

Recovery message

Enabled

In questa form bisogna indicare un nome per l'azione che si sta definendo.

Le macro (o variabili) *{TRIGGER.STATUS}* e *{TRIGGER.NAME}*, visibili nei campi *Default subject* e *Default message* verranno sostituite con lo stato attuale e il nome del trigger.

Nel caso più semplice, se non si aggiungono altre particolari condizioni dal tab *Conditions*, l'azione verrà eseguita ogni qual volta il trigger cambia il suo stato da *Ok* a *Problem*. Cosa deve compiere l'azione viene definito nel tab *Operations*. Cliccando *New* si apre la form per l'aggiunta di una nuova operazione.

Qui si può cliccare su *Add* nel blocco *Send to Users* e aggiungere uno degli utenti definiti. Si seleziona *Email* come valore del campo *Send only to* e si termina l'aggiunta dell'operazione cliccando su *Add*.

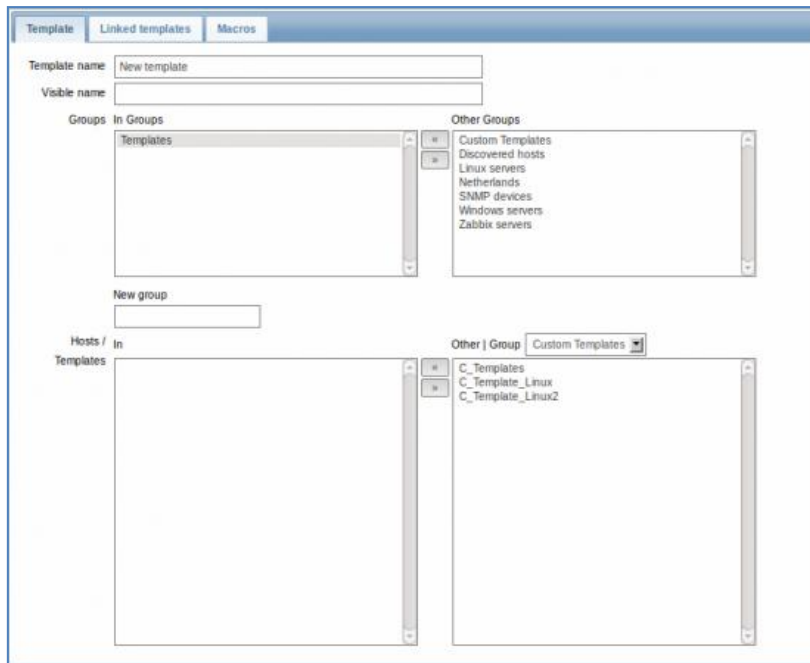
Infine, con *Save* si completa la definizione della nuova azione.

5.7 Creazione di un template

I template sono utili per raggruppare item, trigger ed altre entità in modo tale da poterli riutilizzare più volte applicandoli agli host effettuando un solo passaggio. Quando un template è collegato ad un host, l'host ne eredita tutte le entità definite. In tal modo si può applicare velocemente un set di controlli pre-configurato.

5.7.1 Aggiunta di un template

La creazione di un nuovo template si effettua aprendo la schermata *Configuration* → *Templates* e cliccando su *Create*. Verrà visualizzata la form di configurazione di un nuovo template.



I parametri richiesti da inserire sono:

- **Template name**, nome del template. Sono consentiti caratteri alfanumerici, spazi e underscore;
- **Groups**, uno o più gruppi di appartenenza. Un template deve appartenere ad almeno un gruppo.

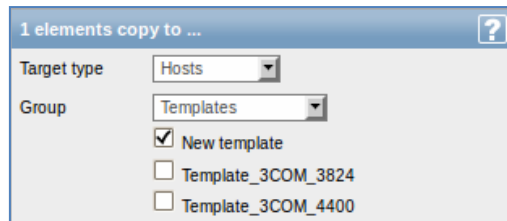
Cliccando su *Save* il nuovo template sarà visibile nella lista dei template e sarà vuoto, cioè senza item, trigger o altra entità definita.



5.7.2 Aggiunta di un item ad un template

Per aggiungere un item ad un template, si può andare sulla lista degli item relativi ad un host e da lì:

- mettere un segno di spunta accanto all'item di interesse
- selezionare *Copy selected to...* nel menu a tendina sotto la lista e premere *Go*
- selezionare il template dove va copiato l'item



- cliccare su *Copy*.

Se dopo si va a guardare il template, questi avrà un nuovo item.

5.7.3 Associazione di un template ad un host

Avendo un template pronto, si può procedere alla sua associazione ad un host. Per fare ciò, si apre la schermata *Configuration* → *Hosts*, si clicca sull'host di interesse e si seleziona il tab **Templates**.

Dopo aver cliccato su *Add*, si seleziona il template desiderato e poi si clicca su *Select*. Il template apparirà nella form.



Dopo aver salvato le modifiche, il template verrà aggiunto all'host, insieme a tutte le entità che contiene.

Ogni modifica agli item, trigger o altre entità effettuate sul template verranno automaticamente propagate agli host a cui il template è associato.