

D.D. prot.n. 391/Ric del 5 luglio 2012



Ministero dell'Istruzione, dell'Università e della Ricerca

Guida Alla Installazione e Configurazione Di:

Zabbix Server

Zabbix Proxy

Monitoring Pillar



Abstract

In questo documento viene presentata la guida di installazione per l'infrastruttura di monitoraggio che comprende l'installazione di Zabbix Server, Zabbix Proxy e dell'applicazione (Pillar) per gestire e wrappare le API degli Zabbix Server.

Glossario, Acronimi e Abbreviazioni

1. Introduzione

2. Installazione e Configurazione Zabbix Server

2.1 Package Zabbix Server 2.4.x (Ubuntu 14.04.x LTS)

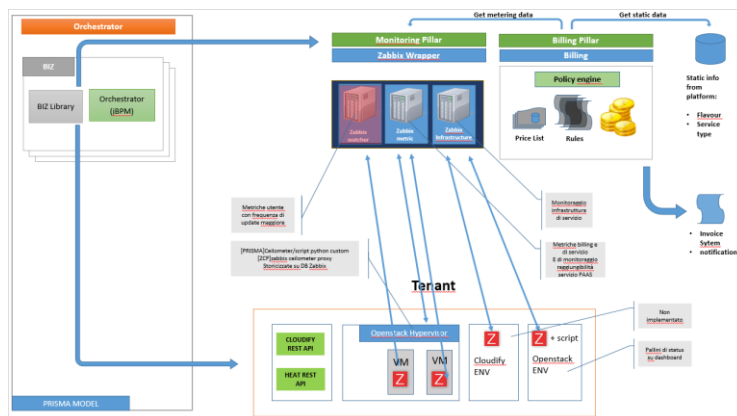
- [2.2 Servizio e Frontend](#)
- [2.3 Accesso Al Frontend](#)
- [2.4 Cambio Password Per L'utente Admin](#)
- [2.5 Configurazione Del Firewall](#)
- [2.6 Zabbix-IaaS](#)
 - [2.6.1 Zabbix Agent](#)
- [2.7 Zabbix-Metrics e Zabbix-Watcher](#)
 - [2.7.1 Zabbix Agent](#)
- [2.8 Comunicazione Server Zabbix-Business Layer](#)
- [3. Installazione e Configurazione Zabbix Proxy](#)
 - [3.1 Zabbix Proxy](#)
 - [3.2 Script Esterno](#)
- [4. Installazione Pillar e Test Della API](#)
 - [4.1 Ambiente Java, JBoss e Monitoring Pillar](#)
 - [4.2 Modifica Delle Proprietà](#)
 - [4.3 API](#)
- [5. Installazione Zabbix Agent](#)
 - [5.1 Tramite Template Heat](#)
 - [5.2 Tramite Ricetta Cloudify](#)

Glossario, Acronimi e Abbreviazioni

Item	Descrizione
Business Layer (BL)	Strato software della PaaS che orchestra tutte le componenti della piattaforma stessa e che espone web services alla web UI e a Zabbix stesso che può tramite call back notificare un evento
Template Zabbix	Gruppo logico di metriche (XML) linkabili agli host di Zabbix in modo tale da avere metriche configurabili globalmente nella piattaforma
Template Heat	Script in linguaggio YAML per orchestrare e automatizzare OpenStack
Cloudify	Deployer Cloud di macchine virtuali e processi a bordo delle stesse
Host Zabbix	Macchina monitorata dal Server
User Interface (UI)	Interfaccia grafica di Zabbix (in questo contesto)
Ceilometer	Software di monitoraggio per OpenStack
Nova	Software di monitoraggio per OpenStack
Simple Check	Categoria di items (metriche in Zabbix) di tipo Agentless
Baltig	Repository GitLab disponibile al seguente link

1. Introduzione

L'architettura dei tre Zabbix servers si inserisce nell'architettura PaaS della piattaforma Prisma/OCF più generale di seguito mostrata:



Come si nota dalla figura, i tre Zabbix server devono comunicare con il resto dei moduli della PaaS esponendo dei servizi che wrappano le API del Zabbix server stesso.

L'installazione all in one dei tre server Zabbix prevede le seguenti specifiche hardware consigliate:

- 8GB RAM
- 4 VCPU
- 50.0GB Disk

L'architettura implementata è composta da tre tipi di Zabbix server diversi a seconda del tipo di macchine monitorate. Le tre macchine sono macchine virtuali con SO Ubuntu 14.04. I database usati sono quelli relativi ad ogni Zabbix quindi uno per ogni macchina. I tre server Zabbix sono stati differenziati per le seguenti ragioni:

- **Zabbix-IaaS:** Con esso vengono monitorate le macchine di infrastruttura intese come bare-metal (cioè i nodi di OpenStack compute, network, controller e le macchine virtuali nel tenant di OpenStack contenenti gli Application Servers per far girare le applicazioni che implementano lo strato PaaS, repositories, etc..).

- **Zabbix-Metrics:** In questo server vengono raggruppate le macchine virtuali deployate e quindi i servizi PaaS. In particolare, ad ogni macchina vengono associate le metriche utili per il billing e ottenute in modo agentless per impedire all'utente (che deploya il servizio VMaaS) di cancellare l'agent a bordo.
- **Zabbix-Watcher:** Zabbix Watcher è il server in cui vengono installate diverse macchine deployate con i servizi PaaS a bordo. E' lo Zabbix utile per l'utente le cui metriche saranno disabilitate o meno a sua discrezione. Sarà costituito di tutte le metriche relative ai singoli servizi come Mysql, Postgresql, JBoss, apache in funzione del servizio che viene installato insieme alla macchina.

Per l'installazione, si possono seguire gli step come da documentazione ufficiale ([link](#)).

Si noti che gli indirizzi IP degli Zabbix server sono stati configurati per essere pubblici perchè le policies del testbed INFN non permettono che tra gruppi di reti private gli Zabbix Agent possano comunicare con i server.

Per ulteriore approfondimento della architettura di monitoraggio, si veda il "documento monitoring working group" al paragrafo 2.4 e 4. disponibile al [link](#).

Per il primo prototipo l'architettura di monitoraggio vede tutti i server e proxies zabbix con IP pubblico tutti a bordo del tenant CORE. I servizi deployati sui tenant utente saranno quindi in grado di comunicare con i proxies e servers.

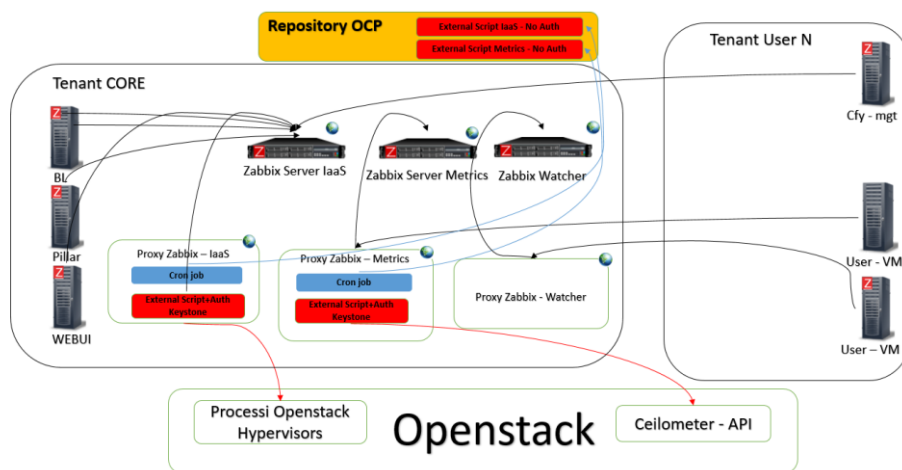


Fig. Architettura I prototipo

Per il secondo prototipo di OCP si implementerà l'architettura di seguito mostrata in cui gli zabbix server (metrics e watcher) più i proxy zabbix, a bordo dei quali ci sono gli external script che servono per comunicare con i processi e hypervisor di OpenStack, oltre che con Ceilometer,

saranno distribuiti nei tenant utente. Il proxy che deve passare i dati al server zabbix-iaas, invece, risiederà nel tenant CORE e sarà quindi installato una volta sola.

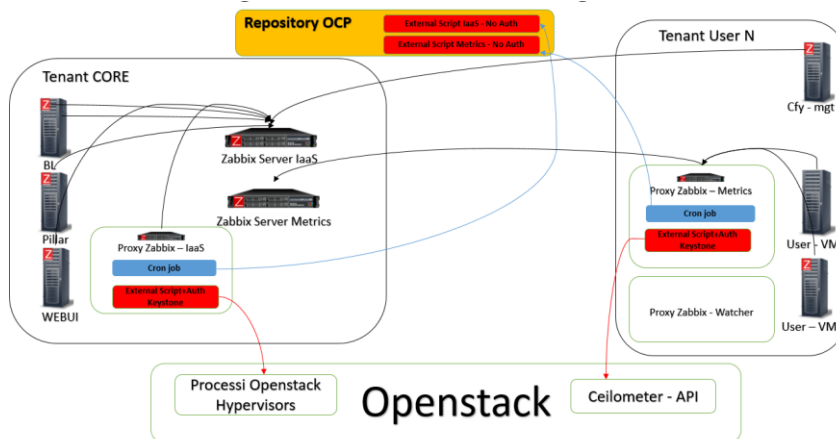


Fig. Architettura II prototipo

In virtù di un'architettura distribuita multi-site lo schema più generale è:

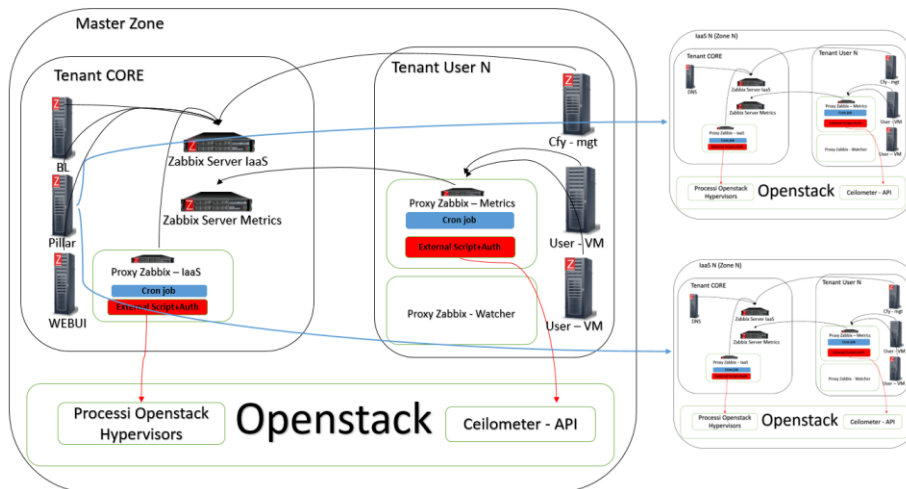


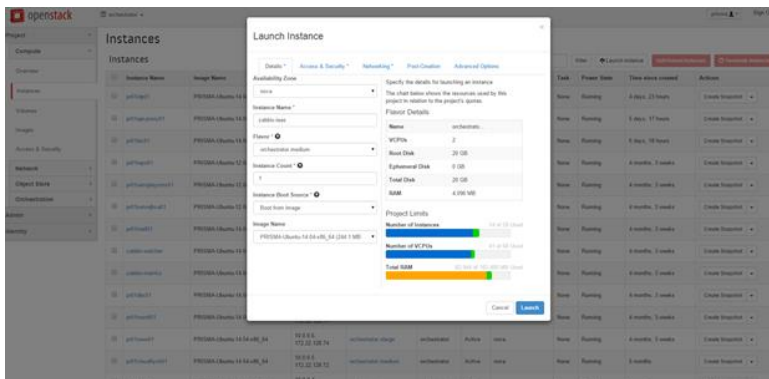
Fig. Architettura II prototipo multi-site

In questo scenario i server zabbix-metrics, iaas, watcher saranno installati in ogni site in modo tale da monitorare in modo granulare i dati di ogni sito. Alla master zone (o master site) arriveranno dati aggregati di ogni zona. I proxies avranno IP privato nel secondo prototipo per rispettare le policies di sicurezza di ogni Iaas. Il monitoring Pillar, di cui si parlerà più avanti, conosce le zone con cui comunicare e viene istruito dal Business Layer/Orchestrator a farlo.

2. Installazione e Configurazione Zabbix Server

Lanciare una istanza di macchina virtuale nel proprio tenant dal tab in OpenStack “Launch Instance” e seguire i wizard per:

- Creare l’istanza con dimensioni e sistema operativo opportuno:



- Assegnare gli opportuni firewall:

The screenshot shows the 'Launch Instance' dialog box with the 'Access & Security' tab selected. The 'Key Pair' dropdown is set to 'orchestrator'. The 'Security Groups' list includes 'smtp-mail-secgroup', 'ejbca-secgroup', 'tomcat-secgroup', 'DNS', 'zabbix_agent', 'zabbix_server', 'default', 'http-secgroup', 'JMX-monitoring', 'JVM-development', and 'mysql-secgroup'. The 'Launch' button is highlighted in blue.

Launch Instance

Details * Access & Security * Networking * Post-Creation Advanced Options

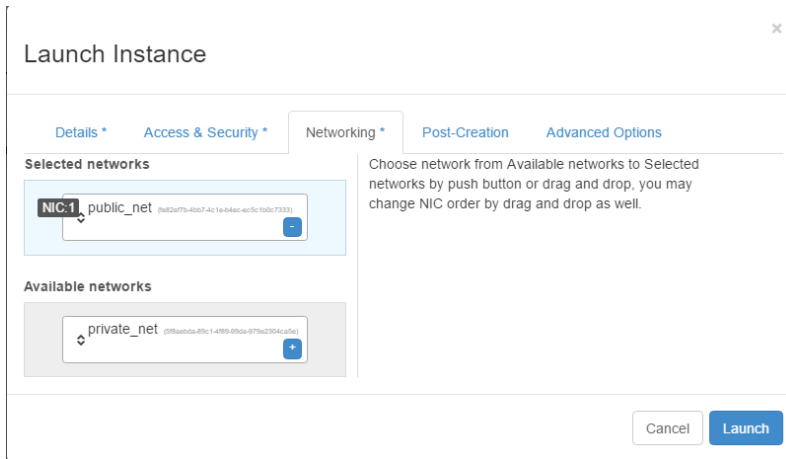
Key Pair ⓘ
orchestrator +

Control access to your instance via key pairs, security groups, and other mechanisms.

Security Groups ⓘ

- ☐ smtp-mail-secgroup
- ☐ ejbca-secgroup
- ☐ tomcat-secgroup
- ☐ DNS
- ☒ zabbix_agent
- ☒ zabbix_server
- ☒ default
- ☒ http-secgroup
- ☒ JMX-monitoring
- ☐ JVM-development
- ☒ mysql-secgroup

Cancel Launch



Assegnare l'opportuna configurazione di rete:

Le configurazioni che seguono valgono per ognuno dei tre server Zabbix.
Modificare il file `/etc/hosts` inserendo la riga:

```
127.0.0.1 <nome_vm>
```

2.1 Package Zabbix Server 2.4.x (Ubuntu 14.04.x LTS)

Loggarsi come root sulla macchina:

```
# sudo su
```

2.1.1 Attachment e mounting del volume per far funzionare il server zabbix con volume esterno

Si è deciso di fare una attachment del volume sulle macchine degli zabbix server perchè soggette a carico maggiore ma anche perchè l'installazione a partire dallo snapshot sarebbe improponibile in termini di risorse da spostare.

Dopo aver creato l'istanza con ubuntu a bordo quindi, si crei il volume dal tab di openstack volumes → create volumes e si completi il seguente wizard:

Create Volume

Volume Name *

test-volume

Description

Description:

Volumes are block devices that can be attached to instances.

Volume Limits

Total Gigabytes (150 GB)

500 GB Available

Number of Volumes (3)

25 Available

Volume Source

No source, empty volume

Type

No volume type

Size (GB) *

50

Availability Zone

nova

Cancel

Create Volume

A questo punto si può procedere a fare l'attachment del volume alla istanza desiderata (nel caso specifico quella di zabbix server) come mostrato nel seguente wizard che appare dopo aver cliccato nel menu a tendina in corrispondenza del volume creato sulla voce Edit Attachments:

Manage Volume Attachments

Attachments

Instance	Device	Actions
No items to display.		
Displaying 0 items		

Attach To Instance

Attach to Instance *

Select an instance

Device Name ?

/dev/vdc

Cancel

Attach Volume

Prima ancora di effettuare installazione di zabbix e Mysql dunque, si entri via ssh nella istanza di zabbix server e si proceda a partizionare il disco con i seguenti comandi:

```
~# fdisk /dev/vdb
```

Apparirà una console interattiva e output come di seguito:

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
Using default value 1
First sector (2048-104857599, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-104857599, default 104857599):
Using default value 104857599

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

root@test-zabbix-metrics-volume:~# mkfs.ext4 /dev/vdb1
mke2fs 1.42.9 (4-Feb-2014)
Etichetta del filesystem=
OS type: Linux
Dimensione blocco=4096 (log=2)
Dimensione frammento=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13106944 blocks
655347 blocks (5.00%) reserved for the super user
Primo blocco dati=0
Maximum filesystem blocks=4294967296
400 gruppi di blocchi
32768 blocchi per gruppo, 32768 frammenti per gruppo
8192 inode per gruppo
Backup del superblocco salvati nei blocchi:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424

Allocating group tables: fatto
Scrittura delle tavole degli inode: fatto
Creating journal (32768 blocks): fatto
Scrittura delle informazioni dei superblocchi e dell'accounting del
filesystem: fatto
```

A questo punto è possibile montare vdb1 su `/var/lib/mysql` che sarà la cartella nella quale mysql di zabbix andrà a scrivere i db:

```
# mkdir /var/lib/mysql
# mount /dev/vdb1 /var/lib/mysql
```

In appendice: un approfondimento su come gestire lo snapshot di macchine con volume attached.

E' finalmente possibile eseguire il download del pacchetto zabbix e configurare il repository:

```
# wget http://repo.zabbix.com/zabbix/2.4/ubuntu/pool/main/z/zabbix-
release/zabbix-release_2.4-1+trusty_all.deb
# dpkg -i zabbix-release_2.4-1+trusty_all.deb; apt-get update
```

(Per Zabbix 3.0 si seguano gli step indicati sotto:

```
# wget http://repo.zabbix.com/zabbix/3.0/ubuntu/pool/main/z/zabbix-
release/zabbix-release_3.0-1+trusty_all.deb
```

```
# dpkg -i zabbix-release_3.0-1+trusty_all.deb
```

```
# apt-get update
```

```
# apt-get install zabbix-server-mysql zabbix-frontend-php
```

```
# mysql -uroot -p<password>
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by
'<password>';
mysql> quit;
```

```
# cd /usr/share/doc/zabbix-server-mysql
# zcat create.sql.gz | mysql -uroot -D zabbix -p
```

n.b. Accertarsi che la password inserita nel file di configurazione del server appena installato coincida con quella inserita durante l'installazione di mysql:

`DBPassword=<pswd>`

Per aggiornare zabbix-~~proxy~~ 2.4 alla 3.0, si applichi:

```
# wget http://repo.zabbix.com/zabbix/3.0/ubuntu/pool/main/z/zabbix-
release/zabbix-release_3.0-1+trusty_all.deb
```

```
# dpkg -i zabbix-release_3.0-1+trusty_all.deb
```

```
# apt-get update
```

```
# apt-get install zabbix-proxy-mysql
```

```
# service zabbix-proxy restart
```

) Se si è installato zabbix 3.0, si riprenda l'installazione dal paragrafo 2.2. al punto *.

2.2 Servizio e Frontend

Per l'installazione del servizio e del frontend lanciare il seguente comando:

```
# apt-get install -y zabbix-server-mysql zabbix-frontend-php snmp
```

Verrà richiesta una password per il database MySQL (utente root). Un esempio potrebbe essere:

```
<mysql_passwd> = mypasszab
```

Alla richiesta di utilizzo di dbconfig-common, rispondere Yes.

Verrà richiesta la password per administrative account di Zabbix con cui creare user e database nel DB MySQL. Inserire la **mysql_passwd**.

Verrà richiesta la scelta di una password per zabbix-server-mysql e di confermarla.

Un esempio potrebbe essere:

```
<zabbix-server-mysql_passwd> = zabservpass
```

*

Modificare il file di configurazione php:

```
# nano /etc/apache2/conf-available/zabbix.conf
```

decommentando la seguente stringa:

```
php_value date.timezone Europe/Rome
```

```
# service apache2 restart
```

Opzionale (solo nel caso in cui si stia monitorando ambienti di tipo VMware):

Cambiare il file di configurazione del Server affinché sia abilitato a monitorare hyperVisor:

```
nano /etc/zabbix/zabbix_server.conf
```

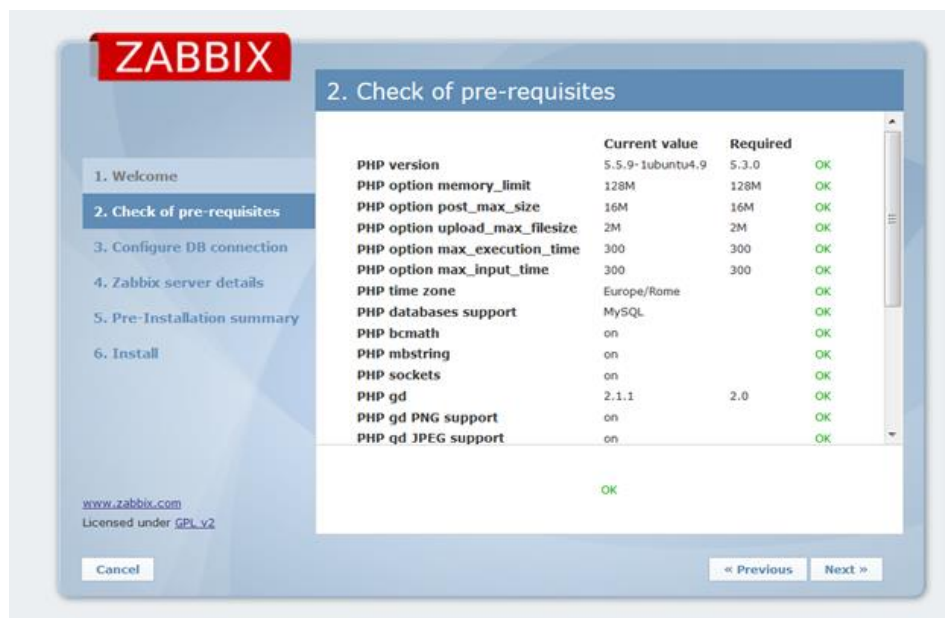
```
StartVMwareCollectors=1
VMwareCacheSize=8M
VMwareFrequency=60
```

2.3 Accesso Al Frontend

Collegarsi all'URL: **http://<IP_server>:80/zabbix/**

Si verrà rediretti all'URL **http://<IP_server>:80/zabbix/setup.php**

Dopo aver cliccato su Next, verrà mostrato un check dei prerequisiti:



Cliccando su Next, verranno richiesti i dati per la connessione al DB popolando i campi con i seguenti valori:

Database type: MySQL
Database host: localhost
Database port: 3306
Database name: zabbix
Database user: zabbix
Password: *****

Inserire la **mysql_passwd** e cliccare su **“Test connection”**.
Se l’esito sarà **“ok”**, si potrà cliccare su **“Next”**.

Verrà mostrata la seguente pagina:

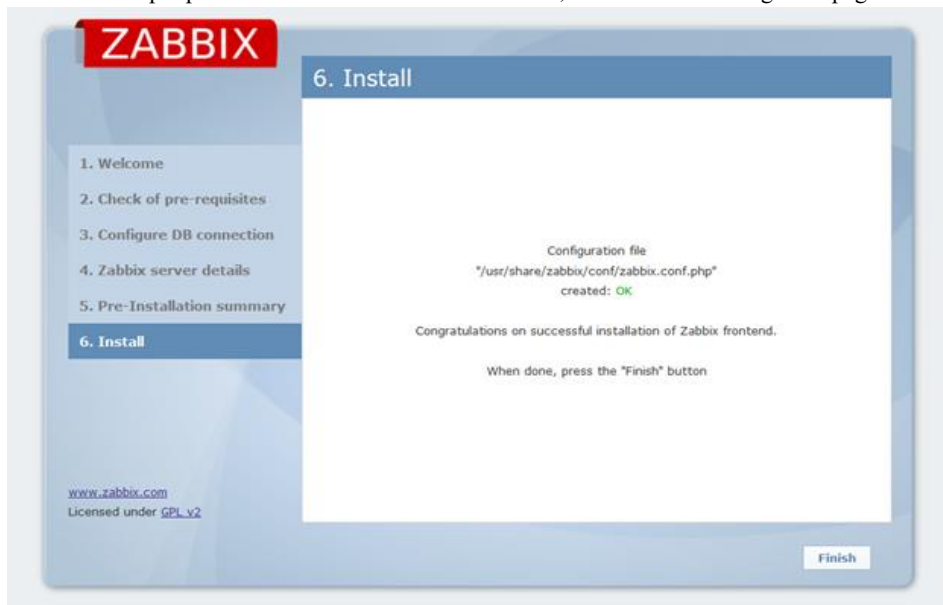


The screenshot shows the Zabbix installation wizard interface. At the top left is the ZABBIX logo. Below it is a sidebar with a list of steps: 1. Welcome, 2. Check of pre-requisites, 3. Configure DB connection, 4. Zabbix server details (highlighted), 5. Pre-Installation summary, and 6. Install. The main area is titled '4. Zabbix server details' and contains the following text: 'Please enter host name or host IP address and port number of Zabbix server, as well as the name of the installation (optional)'. Below this text are three input fields: 'Host' with 'localhost', 'Port' with '10051', and 'Name' which is empty. At the bottom left is a 'Cancel' button, and at the bottom right are '« Previous' and 'Next »' buttons. In the bottom left corner of the window, there is a small text area with the website 'www.zabbix.com' and the license 'Licensed under GPL v2'.

Procedere inserendo il nome per l’installazione (es. Zabbix-Watcher) e cliccare **Next**.
Verrà mostrata la seguente pagina nella quale inserire in corrispondenza del campo:
“Database User”: **“zabbix”**
come la guida ufficiale di Zabbix suggerisce e come è opportuno si faccia.



Cliccare **Next** per procedere all'installazione. Al termine, verrà mostrata la seguente pagina:



Cliccare su **Finish**.
Verrà mostrata la pagina di login:



Inserire:

username = admin

password = zabbix

Procedura Alternativa - Creazione Automatica Del File Di Configurazione

In alternativa all'uso della procedura web guidata descritta in questi paragrafi, è anche possibile generare il file di configurazione automaticamente, o crearlo a mano.

Un esempio di file generato automaticamente:

```
cat > /usr/share/zabbix/conf/zabbix.conf.php << EOL
<?php
// Zabbix GUI configuration file.
global \ $DB;

\ $DB["TYPE"]                = 'MYSQL';
\ $DB["SERVER"]              = 'localhost';
\ $DB["PORT"]                = '0';
\ $DB["DATABASE"]           = '$ZABBIXDB';
\ $DB["USER"]                = 'zabbix';
\ $DB["PASSWORD"]           = '$ZABBIXPWD';
// Schema name. Used for IBM DB2 and PostgreSQL.
\ $DB["SCHEMA"]              = '';

\ $ZBX_SERVER                = 'localhost';
\ $ZBX_SERVER_PORT           = '10051';
\ $ZBX_SERVER_NAME           = '';

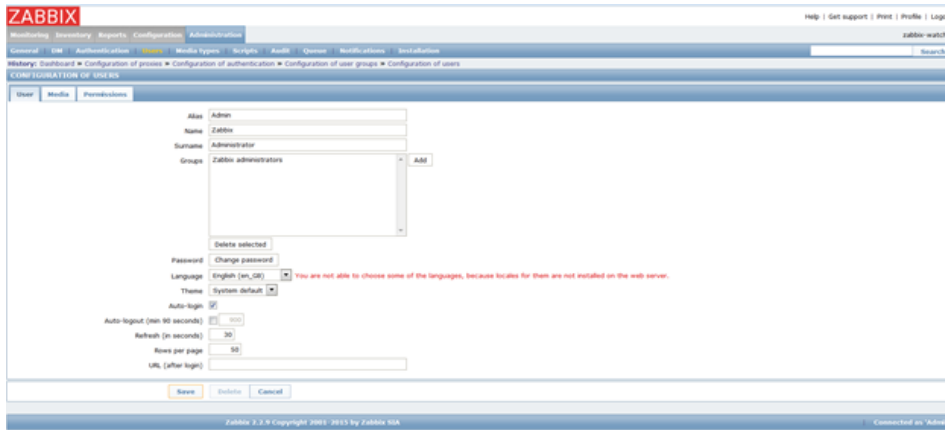
\ $IMAGE_FORMAT_DEFAULT     = IMAGE_FORMAT_PNG;
?>
EOL
```

Avendo cura di impostare i valori delle variabili **\$ZABBIXDB** e **\$ZABBIXPWD** con i valori precedentemente scelti durante l'installazione e configurazione del database.

2.4 Cambio Password Per L'utente Admin

Una volta loggati con le credenziali di default, cliccare su **Administration** → **Users**.
In corrispondenza del nome “**Zabbix administrators**”, cliccare su **Users** e poi **Admin**.

Verrà mostrata la seguente pagina:

The screenshot shows the Zabbix Administration web interface. The top navigation bar includes 'Administration', 'Users', and 'Groups'. The breadcrumb trail is 'Dashboard > Configuration of process > Configuration of authentication > Configuration of user groups > Configuration of users'. The main content area is titled 'CONFIGURATION OF USERS' and shows the 'Admin' user profile. Fields include 'Alias' (Admin), 'Name' (Zabbix), 'Surname' (Administrator), 'Groups' (Zabbix administrators), 'Password' (Change password), 'Language' (English [en_GB]), 'Theme' (System default), 'Auto-login' (checked), 'Auto-logout (min 60 seconds)' (60), 'Refresh (in seconds)' (30), 'Rows per page' (50), and 'URL (after login)'. A 'Save' button is at the bottom.

Cliccare su **change password**, ed inserire la nuova password. Ad esempio:

```
<admin_zabbix_frontend_passwd> = webpasszab
```

Quindi cliccare su **Save**.

Successivamente, eseguire un logout ed il nuovo login inserendo la password appena modificata.

2.5 Configurazione Del Firewall

Affinchè il server possa essere contattato dagli agent installati sui client da monitorare è importante che la porta **10051/TCP** risulti aperta.

2.6 Zabbix-IaaS

Per questo server il file del template da Importare nel server appena installato è **TemplateIaaS** disponibile nella cartella in cui è contenuto questo documento. Per esso, una volta importato si avranno a disposizione i seguenti template (o gruppo di metriche).

L'import si effettua cliccando sul tab **Templates** e poi su **Import Templates**, flaggare anche la voce groups.

- **Template Infrastructure:** gruppo di metriche di sistema operativo
- **Template Prisma Compute:** gruppo di metriche per il monitoraggio del nodo compute
- **Template Prisma Controller:** gruppo di metriche per il monitoraggio del nodo controller
- **Template Prisma Network:** gruppo di metriche per il monitoraggio del nodo network
- **Template OS Linux Prisma 1:** gruppo di metriche di sistema operativo per le macchine sul tenant
- **Template Jboss:** gruppo di metriche per il monitoraggio del processo jboss

Nota: si osservi che solo per questo server, preposto a monitorare una infrastruttura che cambia da ambiente ad ambiente, si è deciso di installare gli agent sulle singole macchine, perchè queste a loro volta vengono installate manualmente. Si è deciso inoltre di fare in modo che ogni macchina (host in zabbix) si autoregistri sul server.

Seguire quindi solo per Zabbix-IaaS il paragrafo 2.6.1 per effettuare l'installazione dell'agent a bordo di ogni macchina facente parte della infrastruttura virtuale (nel tenant CORE del sito IaaS).

2.6.1 Zabbix Agent

Lo Zabbix Agent viene installato su una macchina da monitorare. Gli agent possono fare check passivi e attivi. Nel caso di quelli passivi, l'agent risponde ad una richiesta di dati inviata dal server, mentre nel caso in cui si abbia modalità attiva, è l'agent a contattare il server in push, quest'ultimo invia la configurazione delle metriche all'agent. L'agent deve fare retrieving di una serie di items dallo Zabbix Server per un processing indipendente (si condulti la [guida](#) ufficiale per ulteriori dettagli).

L'aggiunta delle macchine da monitorare avviene in due fasi:

1. Installazione dell'agent nella macchina da monitorare mediante i seguenti comandi:
 - a. `# apt-get install -y zabbix-agent snmp`
 - b. `# nano /etc/zabbix/zabbix_agentd.conf`

- c. Impostare il nome dell'host solo se si vuole che non venga assegnato quello di default che poi corrisponde all'hostname di sistema operativo. Questo nome è lo stesso che il server riconoscerà e che monitorerà.

```
Server=<IP-server-proxy-da-cui-verrà-monitorato>
ServerActive=<IP-server-proxy-da-cui-verrà-monitorato>
Hostname=<hostname>
HostMetadata=<alias_utile_per_autoregistrarsi>
```

d. # service zabbix-agent restart

A scopo informativo (qualora si vogliano aggiornare le versioni degli agent non possibile tramite un semplice comando) vengono listati i comandi per la cancellazione dell'agent:

- ❖ disinstallazione dell'agent e delle sue dipendenze:
`sudo apt-get remove --auto-remove zabbix-agent`
- ❖ purge della configurazione e dati:
`sudo apt-get purge --auto-remove zabbix-agent`

2. L'aggiunta da UI di Zabbix dell'host a cui l'agent fa riferimento è stata eseguita mediante **autoregistrazione**. L'installazione dell'agent tramite autoregistrazione è stata effettuata solo in Zabbix-IaaS, soprattutto utile, ma non solo, per aggiungere un host che corrisponde alla macchina di cloudify-manager.

Per effettuare l'autoregistrazione si eseguono i seguenti passi da UI di Zabbix:

- a. Se la configurazione via ssh è stata eseguita correttamente, un host che ha un nome uguale a quello impostato nel parametro "Hostname" del file di configurazione dell'agent comparirà nel frontend Zabbix e apparterrà ad un gruppo con link al template settato.
- b. Per configurare l'autoregistrazione da front end di zabbix si selezioni dal menu a tendina "Auto Registration" raggiungibile dai tab: Configuration → Actions. Quindi si clicchi su "Create Action" e si inserisca il nome della Action, la condition e le operation secondo i seguenti screen shot (i quali mostrano la creazione della action di tipo auto registration). Se nel file di configurazione dell'agent si è inserito il parametro "HostMetadata=oslinux" esso verrà aggiunto al gruppo infrastructure (precedentemente importato) e al template di nome Template-OSLinux-Prisma-OCP.
 - i.

Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Screens Slide shows Maps Discovery

History: Configuration of media types » Configuration of GUI » Configuration of media types » Configuration of hosts » Configuration of actions

CONFIGURATION OF ACTIONS

Action Conditions Operations

Name: Autoregistr-OSLinux

Default subject: Auto registration: {HOST.HOST}

Default message: Host name: {HOST.HOST}
Host IP: {HOST.IP}
Agent port: {HOST.PORT}

Enabled ☒

Update Clone Delete Cancel

ii.

Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Screens Slide shows Maps Discovery IT services

History: Configuration of media types » Configuration of GUI » Configuration of media types » Configuration of hosts » Configuration of actions

CONFIGURATION OF ACTIONS

Action Conditions Operations

Conditions

Label	Name	Action
A	Host metadata like <i>oslinux</i>	Remove

New condition

Host metadata ▼ like ▼ *oslinux*

[Add](#)

Update Clone Delete Cancel

iii.

Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Screens Slide shows Maps Discovery IT services

History: Configuration of media types » Configuration of GUI » Configuration of media types » Configuration of hosts » Configuration of actions

CONFIGURATION OF ACTIONS

Action Conditions Operations

Action operations

Details	Action
Add to host groups: Infrastructure	Edit Remove
Link to templates: Template-OSLinux-Prisma-OCF	Edit Remove
New	

Update Clone Delete Cancel

- iv. Alla fine l'action di tipo autoregistration deve essere costituita dalle seguenti voci per poter configurare zabbix-iaas con successo:

Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Screens Slide shows Maps Discovery IT services

History: Configuration of media types » Configuration of GUI » Configuration of media types » Configuration of hosts » Configuration of actions

CONFIGURATION OF ACTIONS

Actions

Displaying 1 to 7 of 7 found

<input type="checkbox"/>	Name	Conditions	Operations
<input type="checkbox"/>	Autoregistr-CFY-MGT	Host metadata like cfy-mgt	Add to host groups: Infrastructure Link to templates: TemplateCloudifyManager
<input type="checkbox"/>	Autoregistr-Idp	Host metadata like idp	Add to host groups: Infrastructure Link to templates: Template-Prisma-OCF-IdP
<input type="checkbox"/>	Autoregistr-JbossAS	Host metadata like jbossas	Add to host groups: Infrastructure Link to templates: TemplateJBoss
<input type="checkbox"/>	Autoregistr-MailServer	Host metadata like mail	Add to host groups: Infrastructure Link to templates: TemplateMailServer
<input type="checkbox"/>	Autoregistr-OSLinux	Host metadata like oslinux	Add to host groups: Infrastructure Link to templates: Template-OSLinux-Prisma-OCF
<input type="checkbox"/>	Autoregistr-WebUI	Host metadata like webui	Add to host groups: Infrastructure Link to templates: Template-Prisma-OCF-WebUI
<input type="checkbox"/>	AutoRegistr-Repo	Host metadata like repo	Add to host groups: Infrastructure Link to templates: Template-Prisma-OCF-Repository

Per ulteriori dettagli sulla autoregistrazione, seguire la guida ufficiale di Zabbix ([link](#)).

2.6.2 Creazione Alertscript per mandare email (opzionale)

Verrà fornito il caso in cui zabbix manda mails via alert script e installando il programma ssmtp che non richiede l'installazione di demoni come postfix:

```
# apt-get install ssmtp mailutils
# nano /etc/ssmtp/ssmtp.conf
```

Modificare il file di proprietà come di seguito:

```
root=gmail-username@gmail.com
mailhub=smtp.gmail.com:587
rewriteDomain=your_local_domain
hostname=your_local_FQDN
UseTLS=Yes
UseSTARTTLS=Yes
AuthUser=Gmail_username
AuthPass=Gmail_password
FromLineOverride=YES
```

Testare che lo script funzioni con il seguente comando.

```
# echo "Body test email from 'hostname -f' " | mail -s "subject here"
gmail user@gmail.com
```

Se ritorna il messaggio di errore del tipo: “mail: cannot send message: Process exited with non-zero status”, loggarsi al proprio gmail account e cliccare su “turn on” al seguente link: <https://www.google.com/settings/security/lesssecureapps>

Nel server zabbix si configuri l'alert script:

```
# nano /usr/lib/zabbix/alertscripts/send-email-ssmtp
```

Incollare il seguente script nel file appena aperto, salvare e uscire:

```
#!/bin/bash
echo "$3" | /usr/bin/mail -s "$2" $1
```

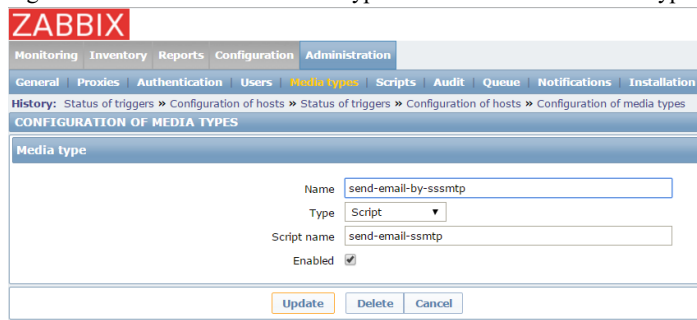
Assegnare gli opportuni permessi per eseguirlo:

```
# chmod +x /usr/lib/zabbix/alertscripts/send-email-ssmtp
```

Testarne il funzionamento con il seguente:

```
#!/usr/lib/zabbix/alertscripts/send-email-ssmtp
gmail_username@gmail.com "Subject here" "Body of the message here"
```

Da interfaccia grafica di zabbix si crei il media type: Administration → Media types → Create:



ZABBIX

Monitoring Inventory Reports Configuration Administration

General Proxies Authentication Users Media types Scripts Audit Queue Notifications Installation

History: Status of triggers » Configuration of hosts » Status of triggers » Configuration of hosts » Configuration of media types

CONFIGURATION OF MEDIA TYPES

Media type

Name: send-email-by-ssmtp

Type: Script

Script name: send-email-ssmtp

Enabled: ☒

Update Delete Cancel

Creare lo script da Configuration → Actions → Create Action → Event Source Trigger e compilare come di seguito:

ZABBIX

Monitoring
Inventory
Reports
Configuration
Administration

Host groups
Templates
Hosts
Maintenance
Actions
Screens
Slide shows
Maps
Discovery
IT services

History: Configuration of hosts » Status of triggers » Configuration of hosts » Configuration of media types » Configuration of actions

CONFIGURATION OF ACTIONS

Action

Conditions

Operations

Name

SendEmailByAlertScript

Default subject

EmailAlertFromZabbixRec

Default message

Trigger: {TRIGGER.NAME}
Trigger status: {TRIGGER.STATUS}
Trigger severity: {TRIGGER.SEVERITY}
Trigger URL: {TRIGGER.URL}
Group: {TRIGGER.HOSTGROUP.NAME}
Host: {HOST.NAME}

Item values:
1. {ITEM.NAME1} ({HOST.NAME1}-{ITEM.KEY1}): {ITEM.VALUE1}
2. {ITEM.NAME2} ({HOST.NAME2}-{ITEM.KEY2}): {ITEM.VALUE2}
3. {ITEM.NAME3} ({HOST.NAME3}-{ITEM.KEY3}): {ITEM.VALUE3}

Original event ID: {EVENT.ID}

Recovery message

☐

Enabled ☒

Update

Clone

Delete

Cancel

ZABBIX

Monitoring
Inventory
Reports
Configuration
Administration

Host groups
Templates
Hosts
Maintenance
Actions
Screens
Slide shows
Maps
Discovery
IT services

History: Configuration of hosts » Status of triggers » Configuration of hosts » Configuration of media types » Configuration of actions

CONFIGURATION OF ACTIONS

Action

Conditions

Operations

Type of calculation

And/Or

A and B and C

Conditions

Label	Name	Action
A	Maintenance status not in maintenance	Remove
B	Trigger value = PROBLEM	Remove
C	Host group = Prisma - SO4 - INFW	Remove

New condition

Trigger name

like

Add

Update

Clone

Delete

Cancel

ZABBIX

Monitoring
Inventory
Reports
Configuration
Administration

Host groups
Templates
Hosts
Maintenance
Actions
Screens
Slide shows
Maps
Discovery
IT services

History: Configuration of hosts » Status of triggers » Configuration of hosts » Configuration of media types » Configuration of actions

CONFIGURATION OF ACTIONS

Action

Conditions

Operations

Default operation step duration

3600 (minimum 60 seconds)

Action operations

Steps	Details	Start in	Duration (sec)	Action
1	Send message to user groups: OCP Devs via send-email-by-ssmtp	Immediately	Default	Edit Remove

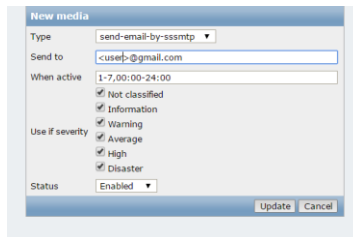
Update

Clone

Delete

Cancel

Aggiungere l'utente via Administration→ Users → TAB:Media:



A questo punto sarà possibile ricevere le mail generate dalla action opportunamente configurata.

2.7 Zabbix-Metrics e Zabbix-Watcher

Per Zabbix-Metrics e Watcher si importino i seguenti template prendendo dalla cartella in cui è contenuto questo documento il file XML “**TemplatePaaS**” e importandolo come fatto per Zabbix-IaaS:

- **TemplateJboss**: gruppo di metriche per il monitoraggio del processo jboss
- **TemplateApache**: gruppo di metriche per il monitoraggio del processo apache
- **TemplateGlassfish**: gruppo di metriche per il monitoraggio del processo glassfish
- **TemplateHAProxy**: gruppo di metriche per il monitoraggio del processo haproxy
- **TemplateIaaS**: gruppo di metriche per il monitoraggio di so della VM; sono le metriche per che restituiscono i valori forniti da ceilometer
- **TemplateMySQL**: gruppo di metriche per il monitoraggio del processo mysql
- **TemplatePentaho**: gruppo di metriche per il monitoraggio del processo pentaho
- **TemplatePostgreSQL**: gruppo di metriche per il monitoraggio del processo postgresql
- **TemplateRabbitMQ**: gruppo di metriche per il monitoraggio del processo rabbitmq
- **TemplateTomcat**: gruppo di metriche per il monitoraggio del processo tomcat
- **TemplateVMaaS**: contiene solo la metrica per pingare in modo agentless una macchina (item di tipo simplecheck in Zabbix)
- **TemplateX2go**: gruppo di metriche per il monitoraggio del processo x2go server

2.7.1 Zabbix Agent

L’installazione degli agent dei server Zabbix-Metrics e Watcher sono stati automatizzati nel template di heat (nel caso di deploy di una VMaaS o di DBaaS) e nella ricetta di cloudify (nel caso di AppaaS e altri servizi). L’automatizzazione dell’installazione degli agent Zabbix in heat e cloudify è dunque possibile soltanto installando il Business Layer di Prisma. Per completezza, nei successivi paragrafi sono state descritte le ricette e template rispettivamente di cloudify e heat ma che non sono necessarie per il funzionamento del monitoring pillar come layer software

standalone. E' tuttavia assolutamente necessario che gli agent vengano installati nelle macchine deployate dall'utente che usa la piattaforma Prisma perchè quest'ultima provvede a chiamare la API di creazione dell'host in Zabbix le cui caratteristiche (hostname, UUID e IP) devono coincidere con quelle della macchina che l'utente ha istanziato, senza questi parametri la comunicazione tra Zabbix Server ed Agent (utile per monitorare i servizi PaaS non può avvenire con successo). Si noti che solo nel caso di VMaaS non viene installato alcun agent perchè l'utente potrebbe eliminarlo avendone l'accesso via ssh. Per esso si monitoreranno le metriche di sistema operativo tramite Ceilometer e metriche di tipo simplechecks (metriche Zabbix agentless) configurate automaticamente mediante import dei template e assegnate agli host in fase di creazione dell'host stesso mediante API del pillar.

L'aggiunta da UI di Zabbix dell'host a cui l'agent fa riferimento viene quindi eseguita automaticamente mediante opportuno workflow che parte dalla WebUI di Prisma, passa per il client dell'orchestrator prima di chiamare il pillar preposto a wrappare le API di Zabbix tra cui, in questo caso specifico: host.create. Questa API viene chiamata durante il deploy del servizio PaaS e IaaS da Business Layer per la creazione di un host e associazione del template (gruppo di metriche e triggers) allo stesso.

Si osservi anche che i gruppi vengono creati automaticamente ogni volta che viene creato un workgroup o project o tenant in OpenStack.

2.8 Comunicazione Server Zabbix-Business Layer

Tale funzionalità è stata descritta in questa guida perchè fa parte delle configurazioni adottate in in Prisma/OCF per permettere al server Zabbix-**metrics** di comunicare direttamente con il Monitoring pillar che a sua volta espone una API utile a scrivere su un file di log, il cui contenuto sarà opportunamente inviato a Elastic Search prima di essere mostrato su Kibana. Per automatizzare dunque le callback si eseguano gli stessi passi seguiti nel paragrafo precedente.:

- Creare una action come di seguito mostrato:
 - Dare il nome della action nel campo Name: StatusMachAction
 - Impostare il seguente Default subject: token_monitoring
 - Impostare:

```
{
  "paaSServiceId": {INVENTORY.TAG},
  "triggerStatus" : "PROBLEM",
  "hostId" : "{HOST.NAME1}"
}
```

- Ripetere l'operazione per il recovery message come di seguito inserendo:

```
{
  "paaSServiceId": {INVENTORY.TAG},
  "triggerStatus" : "RESOLVED",
  "hostId" : "{HOST.NAME1}"
}
```

- Con la seguente condizione:

- Il tab operations deve essere modificato con Edit e settato come mostrato di seguito:

- Per configurare l'endpoint del servizio si deve aggiungere un Mediatype da Interfaccia di Zabbix:
 - Creazione di un Mediatype: cliccare sul tab **Administration** → **Media Types** → **Create media type** → Riempire i campi come illustrato di seguito:
 Name : StatusServiceMedia
 Type : Script
 Script name: StatusMachUpdate.sh
 NB Se si sta usando zabbix 3.0 aggiungere anche gli script parameters come illustrato di seguito:

PARAMETER	ACTION
(ALERTSENDTO)	Remove
(ALERTSUBJECT)	Remove
(ALERTMESSAGE)	Remove
-group	Remove
-debug	Remove

[Add](#)

- Aggiunta dell'endpoint: andare in **Administration Users → Zabbix Administrators (Members) → Media** aggiungere l'endpoint per la mediaType create:
`<url>/biz/rest/monitoring/updateServiceStatus`

Inserire lo script .sh nel server zabbix-metrics :

```
# cd /root; git clone https://baltig.infn.it/OCP-MONITORING/zabbix-external-scripts.git;  
# mv zabbix-external-scripts/alert-scripts-metrics/statusMachUpdate.sh  
/usr/lib/zabbix/alertscripts/
```

Il contenuto dello script è:

```
#!/bin/bash  
curl -X POST -H "Content-Type: application/json" -d $3 $1
```

Dare gli opportuni permessi allo script:

```
# chmod 755 /usr/lib/zabbix/alertscript/statusMachUpdate.sh
```

Per ulteriore approfondimento sugli alert script, si veda la [guida](#) ufficiale.

2.9 Comunicazione zabbix-monitoring-pillar-logstash

L'integrazione monitoring IaaS/PaaS è stata ottenuta mediante l'installazione di logstash e sua configurazione all'interno della macchina che ospita l'applicazione di monitoraggio IaaS/PaaS (monitoring pillar).

Partendo dal presupposto di aver installato logstash da packages seguendo la guida per l'installazione dei componenti dello stack ELK, si apportino le seguenti modifiche presenti nel file di configurazione di logstash, chiamato.

logstash_paas_callbacks.conf, nel path # /etc/logstash/conf.d/:

Guardare la guida per l'installazione del pillar per un approfondimento su come installare logstash e quale configurazione settare.

Le callback da zabbix alla applicazione dopo aver configurato propriamente zabbix.

Tale funzionalità permette al server Zabbix-metrics di comunicare direttamente con il Monitoring

pillar che a sua volta espone una API utile a scrivere su un file di log il cui contenuto sarà opportunamente inviato a Elastic Search prima di essere mostrato su Kibana. Per automatizzare dunque le callback si eseguano gli stessi passi seguiti nel paragrafo precedente.:

Creare una action come di seguito mostrato:

- Dare il nome della action nel campo Name: eventCallbackAction
- Impostare il seguente Default subject: application/json
- Riempire il body della action come di seguito:

```
{
  "paaSServiceId": "{INVENTORY.TAG}",
  "triggerStatus" : "PROBLEM",
  "hostId" : "{$UUID}",
  "hostName": "{HOST.NAME}",
  "group": "{TRIGGER.HOSTGROUP.NAME}",
  "metric": "{ITEM.NAME}",
  "threshold": "{TRIGGER.NAME}",
  "description": "{TRIGGER.DESCRPTION}",
  "ip": "{HOST.IP}"
}
```
- Ripetere l'operazione inserendo il recovery message:

```
{
  "paaSServiceId": "{INVENTORY.TAG}",
  "triggerStatus" : "RESOLVED",
  "hostId" : "{$UUID}",
  "hostName": "{HOST.NAME}",
  "group": "{TRIGGER.HOSTGROUP.NAME}",
  "metric": "{ITEM.NAME}",
  "threshold": "{TRIGGER.NAME}",
```

```
"description": "{TRIGGER.DESCRPTION}",
"ip": "{HOST.IP}"
}
```

- Lasciare la condizione come di default e settare “Operations” per inviare i messaggi agli user groups via eventCallback.
- Aggiungere il tipo di media dai tab: **Administration** → **Media Types**:
 Name : eventCallbackMedia
 Type: Script
 Script name : eventCallback.sh
 NB se si sta usando zabbix 3.0 aggiungere i parametri come di seguito:

PARAMETER	ACTION
(ALERT.SENDTO)	Remove
(ALERT.SUBJECT)	Remove
(ALERT.MESSAGE)	Remove
--group	Remove
--debug	Remove

- Aggiunta dell’endpoint: andare in **Administration Users** → **Zabbix Administrators (Members)** → **Media** aggiungere l’endpoint per la mediaType create:
 <url>/monitoring/adapters/zabbix/zones/{zone}/types/service/event-callbacks
- Aggiungere lo script nel server zabbix-metrics:
 # cd /root; git clone <https://baltig.infn.it/OCP-MONITORING/zabbix-external-scripts.git>;
 # mv zabbix-external-scripts/alert-scripts-metrics/eventCallback.sh
 /usr/lib/zabbix/alertscripts/
- Dare gli opportuni permessi allo script:
 # chmod 755 /usr/lib/zabbix/alertscript/eventCallback.sh

3. Installazione e Configurazione Zabbix Proxy

L'installazione degli Zabbix Proxies è necessaria in quanto si è voluto fare evolvere l'architettura del Prototipo 1 per evitare che le credenziali di autenticazione di Keystone risiedessero all'interno dei servers che sono pubblici.

In questo modo gli script esterni che renderanno possibile l'autenticazione con Keystone per parlare con la API di Ceilometer, saranno all'interno degli Zabbix Proxies che saranno all'interno del tenant Core con IP privato.

Inoltre, questo ci dà il vantaggio di poter pensare ad un'architettura di monitoring distribuita geograficamente (i servers non necessariamente devono risiedere nel tenant Core della cloud monitorata) ed in più i proxies possono snellire il carico dei servers.

Per installare le varie componenti, seguire i successivi passaggi.

Andare sulle 3 macchine adibite per ospitare gli **Zabbix Proxy** ed eseguire i seguenti comandi:

3.1 Zabbix Proxy

```
# wget http://repo.zabbix.com/zabbix/2.4/ubuntu/pool/main/z/zabbix-  
release/zabbix-release_2.4-1+trusty_all.deb  
# dpkg -i zabbix-release_2.4-1+trusty_all.deb  
# apt-get update  
# apt-get install -y zabbix-proxy-mysql snmp git  
# service zabbix-proxy stop  
  
# nano /etc/zabbix/zabbix_proxy.conf
```

```
...  
ProxyMode=0  
...  
Server=<zabbix-server-public-ip>  
...  
Hostname=<ProxyName>  
...  
#SourceIP=  
...
```

```
ExternalScripts=/usr/lib/zabbix/externalscripts/zabbix-external-  
scripts  
...  
#DebugLevel=4  
...  
ProxyOfflineBuffer=720  
...  
Timeout=30  
...  
ConfigFrequency=30
```

Nota: per gli hostname potremmo proporre questo formalismo (per esempio per Bari):

- *ProxyBalaas*
- *ProxyBaMetrics*
- *ProxyBaWatcher*

Questo formalismo nel secondo prototipo lascerà il posto ad un altro tipo di formalismo: i proxies metrics e watcher a bordo di ogni tenant utente avranno il nome proxy-*<id-tenant>*. Si è scelto questo tipo di formalismo per mantenere coerenza con i dati presenti nel DB della PaaS ma anche per facilitare il mapping nel pillar tra hostgroup (tenant utente) e proxy (installato in ogni tenant).

```
# service zabbix-proxy start
```

3.2 Script Esterno nel Proxy IaaS e Metrics

```
# cd /root  
# git clone https://baltig.infn.it/OCP-MONITORING/zabbix-external-  
scripts.git
```

- Se si è all'interno del **proxy-metrics** si eseguano i comandi:
mkdir /usr/lib/zabbix/externalscripts/zabbix-external-scripts
cp /root/zabbix-external-scripts/external-script-metrics/*
/usr/lib/zabbix/externalscripts/zabbix-external-scripts/

```
# cp /usr/lib/zabbix/externalscripts/zabbix-external-
scripts/credentials_template.conf
/usr/lib/zabbix/externalscripts/zabbix-external-
scripts/credentials.conf
# cp /usr/lib/zabbix/externalscripts/zabbix-external-
scripts/token_backup_template
/usr/lib/zabbix/externalscripts/zabbix-external-
scripts/token_backup
```

- Mentre per il **proxy-iaas** si lanci il comando:

```
# cd /root; git clone https://baltig.infn.it/OCP-
MONITORING/zabbix-external-scripts.git;
mkdir /usr/lib/zabbix/externalscripts/zabbix-external-scripts
# cp /root/zabbix-external-scripts/external-script-iaas/*
/usr/lib/zabbix/externalscripts/zabbix-external-scripts/
# cp /usr/lib/zabbix/externalscripts/zabbix-external-
scripts/credentials_template.conf
/usr/lib/zabbix/externalscripts/zabbix-external-
scripts/credentials.conf
# cp /usr/lib/zabbix/externalscripts/zabbix-external-
scripts/token_backup_template
/usr/lib/zabbix/externalscripts/zabbix-external-
scripts/token_backup
```

In entrambi i casi si esegua

```
# chown -R zabbix:zabbix /usr/lib/zabbix
```

```
# nano credentials.conf
```

```
[ceilometer]
```

```
protocol = <ceilometer API protocol, http or https>
```

```
server = <ceilometer API ip or hostname>
```

```
port = <ceilometer API port>
```

```
[keystone]
```

```
protocol = <keystone API protocol, http or https>
```

```
server = <keystone API ip or hostname>
```

```
port = <keystone API port>
```

```
tenant = <keystone API tenant>
```

```
username = <keystone API username>
```

```
password = <keystone API password>
```

Note:

Si raccomanda di utilizzare il tenant "service" con lo username "ceilometer".

Altrimenti se si intende utilizzare il tenant "admin", lo username sarà "admin".

Per effettuare un test dello script per verificare se riesce ad autenticarsi con Keystone e se colloquia con la API di Ceilometer provare ad eseguire:

```
# sudo -u zabbix ./ceilometer.sh <meter_name> <uuid_istanza>
```

Ad esempio:

```
# sudo -u zabbix ./ceilometer.sh cpu_util 1d61306b-1818-4ba0-b471-399e7c88f6f4
```

Nota:

Se l'esecuzione dello script dovesse andare in timeout dare questo comando:

```
# rm -f /usr/lib/zabbix/externalscripts/zabbix-external-scripts/*.lock; service zabbix-proxy restart
```

Se si vuole che gli script esterni siano sempre alla versione più aggiornata presente nel repository git si può creare un cron job che faccia il pull ogni cinque minuti con questo comando, per il proxy metrics:

```
# echo "*/5 * * * * root git -C zabbix-external-scripts/ pull && cp zabbix-external-scripts/external-script-metrics/* /usr/lib/zabbix/externalscripts/zabbix-external-scripts/" >> /etc/cron.d/zabbix-update
```

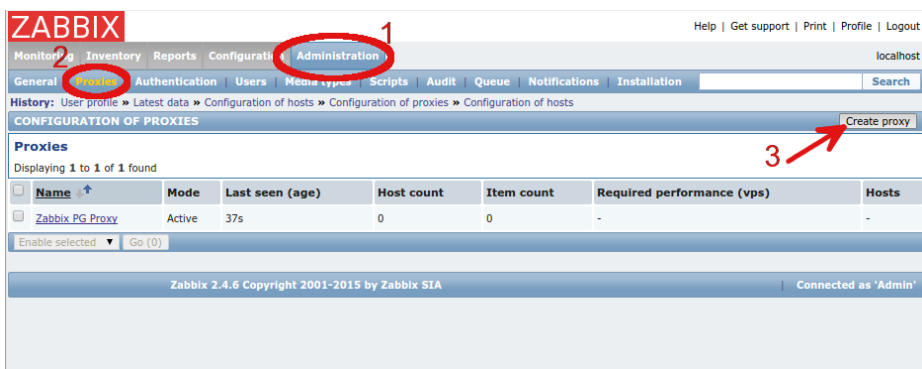
o questo comando, per il proxy laaS:

```
# echo "*/5 * * * * root git -C zabbix-external-scripts/ pull && cp
zabbix-external-scripts/external-script-iaas/*
/usr/lib/zabbix/externalscripts/zabbix-external-scripts/" >>
/etc/cron.d/zabbix-update
```

3.3 Configurazione Zabbix Server Dashboard

Fare login nella dashboard dello Zabbix Server in cui si vuole configurare il proxy.

Andare su **Administration** → **Proxies** → **Create proxy**.



Dare un nome al proxy:

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring Inventory Reports Configuration **Administration** localhost

General **Proxies** Authentication Users Media types Scripts Audit Queue Notifications Installation Search

History: Latest data » Configuration of hosts » Configuration of proxies » Configuration of hosts » Configuration of proxies

CONFIGURATION OF PROXIES

Proxy

Proxy name: Zabbix Ba Proxy 1

Proxy mode: Active

Hosts: Proxy hosts

Other hosts: zabbix-agent-01, Zabbix server

Description:

Add Cancel 2

Zabbix 2.4.6 Copyright 2001-2015 by Zabbix SIA | Connected as 'Admin'

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring Inventory Reports Configuration **Administration** localhost

General **Proxies** Authentication Users Media types Scripts Audit Queue Notifications Installation Search

History: Latest data » Configuration of hosts » Configuration of proxies » Configuration of hosts » Configuration of proxies

Proxy added

CONFIGURATION OF PROXIES Create proxy

Proxies

Displaying 1 to 2 of 2 found

<input type="checkbox"/>	Name	Mode	Last seen (age)	Host count	Item count	Required performance (vps)	Hosts
<input checked="" type="checkbox"/>	Zabbix Ba Proxy	Active	-	0	0	-	-
<input type="checkbox"/>	Zabbix PG Proxy	Active	40s	0	0	-	-

Enable selected Go (0)

Zabbix 2.4.6 Copyright 2001-2015 by Zabbix SIA | Connected as 'Admin'

3.4 Monitorare Un Host Tramite Il Proxy

Questo è solo un esempio se si vuole testare manualmente il funzionamento del proxy.

Creare un nuovo host:

ZABBIX

Help | Get support | Print | Profile | Logout

Monitoring | Inventory | Reports | **Configuration** | Administration

Host groups | Templates | **Hosts** | Maintenance | Actions | Screens | Slide shows | Maps | Discovery

IT services

History: Latest data » Configuration of hosts » Configuration of proxies » Configuration of hosts » Configuration of proxies

CONFIGURATION OF HOSTS

Hosts

Displaying 1 to 2 of 2 found

Create host | Import

Group: all

Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability
zabbix-agent-01	Applications (0)	Items (0)	Triggers (0)	Graphs (0)	Discovery (0)	Web (0)	127.0.0.1: 10050	-	Enabled	
Zabbix server	Applications (11)	Items (69)	Triggers (43)	Graphs (12)	Discovery (2)	Web (0)	127.0.0.1: 10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	Enabled	

Export selected | Go (0)

Zabbix 2.4.6 Copyright 2001-2015 by Zabbix SIA | Connected as 'Admin'

Host | Templates | IPMI | Macros | Host inventory

Host name: **zabbix-agent**

Visible name:

Groups: In groups: Virtual machines

Other groups: CMSaaS, Discovered hosts, Hypervisors, Linux servers, MQaaS, OCP services, Templates, VMaaS, VMPaaS

New group:

Agent interfaces:

IP address	DNS name	Connect to	Port	Default
127.0.0.1		IP DNS	10050	<input checked="" type="radio"/> Remove

SNMP interfaces: Add

JMX interfaces: Add

IPMI interfaces: Add

Description:

Monitored by proxy: **Zabbix Ba Proxy**

Enabled: ☒

Add Cancel

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring Inventory Reports Configuration Administration localhost

Host groups Templates **Hosts** Maintenance Actions Screens Slide shows Maps Discovery

IT services Search

History: Configuration of hosts » Configuration of proxies » Configuration of hosts » Configuration of proxies » Configuration of hosts

CONFIGURATION OF HOSTS

Host Templates IPMI Macros Host inventory

Linked templates

Name	Action
No templates linked.	

Link new templates

1 Template OS Linux Active X Select

2 Add

Add Cancel

Zabbix 2.4.6 Copyright 2001-2015 by Zabbix SIA Connected as 'Admin'

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring Inventory Reports Configuration Administration localhost

Host groups Templates **Hosts** Maintenance Actions Screens Slide shows Maps Discovery

IT services Search

History: Configuration of hosts » Configuration of proxies » Configuration of hosts » Configuration of proxies » Configuration of hosts

CONFIGURATION OF HOSTS

Host Templates IPMI Macros Host inventory

Macro Value

1 (\$UUID) → b052fb50-9af3-4685-91c1-d9bf7df9952d Remove

Add

2 Add Cancel

Zabbix 2.4.6 Copyright 2001-2015 by Zabbix SIA Connected as 'Admin'

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring Inventory Reports Configuration Administration localhost

Host groups Templates **Hosts** Maintenance Actions Screens Slide shows Maps Discovery

IT services Search

History: Configuration of hosts » Configuration of proxies » Configuration of hosts » Configuration of proxies » Configuration of hosts

Details Host added

CONFIGURATION OF HOSTS Create host Import

Hosts Group all

Displaying 1 to 3 of 3 found

Show filter

Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability
Zabbix Ba Proxy: zabbix-agent	Applications (10)	Items (38)	Triggers (12)	Graphs (5)	Discovery (2)	Web (0)	127.0.0.1: 10050	Template OS Linux Active	Enabled	📶 📶 📶 📶

3.5 Altre configurazioni utili

Per tutti i server zabbix, si apporti la configurazione da front-end come mostrato di seguito. Questa modifica è necessaria per vedere la notifica del trigger per tre minuti. Quando il trigger ritorna in stato “resolved” scomparirà dopo tre minuti.

The screenshot shows the Zabbix Configuration page for 'Trigger displaying options'. The page has a top navigation bar with tabs: Monitoring, Inventory, Reports, Configuration, Administration, and a user profile 'PetrMazak'. Below the navigation bar is a breadcrumb trail: Home > Configuration > Hosts > Hosts Type > All hosts > General > Notifications > Installation. The main content area is titled 'CONFIGURATION OF ZABBIX' and 'Trigger displaying options'. It contains a table with columns 'Unacknowledged PROBLEM events', 'Acknowledged PROBLEM events', 'Unacknowledged OK events', and 'Acknowledged OK events'. Each row has a 'Colour' column with a color swatch and a 'Blinking' column with a checkbox. The 'Display OK triggers for' and 'On status change triggers blink for' fields are both set to '180' seconds. At the bottom are 'Update' and 'Reset defaults' buttons. A dropdown menu is open on the right, showing options: GUT, Housekeeping, Images, Icon mapping, Regular expressions, Macros, Value mapping, Working time, Trigger severities, **Trigger displaying options**, and Other.

Unacknowledged PROBLEM events	Acknowledged PROBLEM events	Unacknowledged OK events	Acknowledged OK events
Colour	Colour	Colour	Colour
DC0000	DC0000	00A000	00A000
Blinking	Blinking	Blinking	Blinking
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Display OK triggers for seconds
On status change triggers blink for seconds

[Update](#) [Reset defaults](#)

- Trigger displaying options
- GUT
- Housekeeping
- Images
- Icon mapping
- Regular expressions
- Macros
- Value mapping
- Working time
- Trigger severities
- Trigger displaying options**
- Other

4. Installazione Pillar e Test Della API

Accertarsi di aver seguito i passi della guida per avere i prerequisiti utili all'installazione e funzionamento della applicazione del monitoring pillar.

Per maggiore chiarezza, ai fini dei test sul funzionamento della piattaforma di monitoring IaaS/PaaS, si riepiloga la procedura per richiamare con successo le API che wrappano le funzionalità di Zabbix:

1. Installazione degli Zabbix-IaaS, Metrics e Watcher come indicato nei paragrafi 2-3 per l'installazione e configurazione
2. Importare i template.xml disponibili al repository: seguire il paragrafo 3 per l'export/import degli stessi
3. Importare nei server i file template.xml e host.xml basterebbe in questa prima fase a fare funzionare le API del pillar. Gli host di IaaS ovviamente non verranno monitorati se, non prima configurati per essere monitorati dal server Zabbix-IaaS (mettere l'IP opportuno nel file di configurazione dell'agent). Il template host.xml non è necessario importarlo se sta monitorando una infrastruttura diversa da OS INFN Bari
4. Test delle API

Nota: questi non saranno agganciati a nessun host.

Per far sì che un host possa essere monitorato con successo deve essere:

- Installato in una VM (host da monitorare) e configurato da file di configurazione dell'agent Zabbix stesso come indicato nel paragrafo 4. Questo step, ai fini dei test, conviene farlo manualmente se non si dispone di tutta l'applicazione Prisma e suoi componenti software infrastrutturali (cloudify e heat).
- Aggiunto alla UI di Zabbix server specificando l'opportuno gruppo di appartenenza, ip della VM in cui è stato installato e link al template (gruppo di metriche).
- E' possibile effettuare la creazione di un host tramite chiamata alle API esposte dal pillar. In Prisma la creazione dell'host viene eseguita solo in Zabbix-Metrics e Watcher. Usato in modo standalone, il pillar permette la creazione di un host anche in Zabbix-IaaS a patto di specificarlo nell'url (si veda paragrafo 6).

Per quanto riguarda Zabbix-Metrics, ci sono alcune metriche (quelle per il billing) che non richiedono l'installazione dell'agent sulla macchina virtuale (questo allo scopo di non permettere all'utente la cancellazione dell'agent stesso se deploya una servizio VMaaS). Queste metriche

sono quelle che fanno parte del “TemplateIaaS” da importare in Metrics e Watcher e aver importato anche lo script Python nel server disponibile nel file “monitoring installation pack” da repository baltig. Ovviamente si sta supponendo che l’host monitorato in questo caso sia stato istanziato in OpenStack e che ceilometer sia disponibile nell’infrastruttura.

In Zabbix-IaaS, gli host devo essere aggiunti in modo manuale, o tramite autoregistrazione oppure ancora in modalità automatica tramite template heat o ricette cludify per Zabbix-IaaS (quest’ultima possibilità non applicabile se non si dispone di uno strato software utile per richiamare le API di heat e cludify rispettivamente). La natura degli host in Zabbix-IaaS, in realtà, dipende dalla natura della infrastruttura che si sta monitorando. Per Prisma, per esempio, si sta monitorando il testbed di INFN Bari. Nel caso di OS Torino, gli agent dovranno essere installati sui nodi utili per il compute, network, controller. Viene monitorato anche un gruppo di macchine virtuali che ospitano gli AS e applicazioni in Prisma e che sono state raggruppate in un tenant di OS.

4.1 Ambiente Java, JBoss e Monitoring Pillar

La documentazione relativa all’installazione dell’ambiente e application server con applicazione è raggiungibile al [link](#) sotto il path “documentation/guide/”.

4.2 Modifica Delle Proprietà

Seguire lo stesso link indicato nel paragrafo 4.1.

4.3 API

Il documento delle API è al [link](#) sotto la cartella “documentation”.

5. Installazione Zabbix Agent

5.1 Tramite Template Heat

Opzionale per uso del pillar in modalità standalone

In prisma l'utente non può accedere via ssh alle macchine dotate di servizi paas (tranne che per il servizio VMaaS o servizio IaaS). Siccome sia Cloudify sia Heat sono i software di riferimento in Prisma per deployare le macchine, si è reso necessario automatizzare l'installazione dell'agent a bordo delle macchine e sua configurazione. Questo paragrafo è stato aggiunto per dare un quadro completo del ruolo e configurazione di Zabbix automatica in una piattaforma PaaS ma non è assolutamente necessario per testare il pillar e sue API in modo standalone.

Un esempio di script heat è esposto come ws al seguente endpoint:

`http://90.147.102.53/templates/iaas/vm/vm.template`

Commented [1]: Da togliere?

Questo template è utile per istanziare una vm a cui viene agganciata l'immagine (sistema operativo) opportuna. Lo strato software Business Layer si occupa di chiamare le API di heat per lanciare tale deploy.

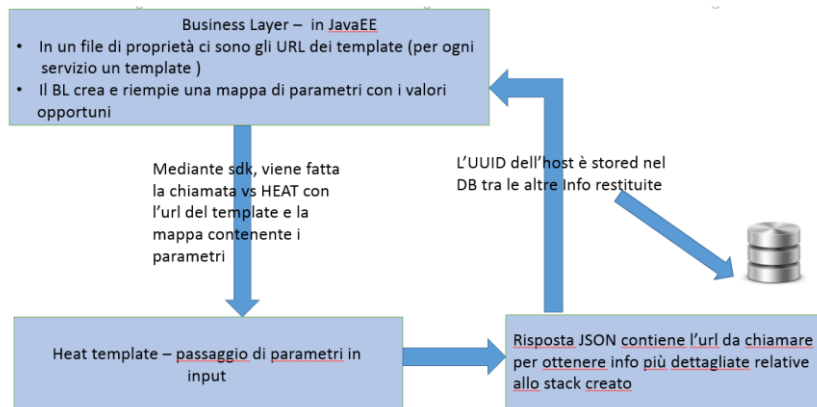
Un esempio di script da inserire nel template per installare e configurare l'host è di seguito mostrato.

```
# wget "http://repo.zabbix.com/zabbix/2.2/ubuntu/pool/main/z/zabbix-
release/zabbix-release_2.2-1+precise_all.deb"
# dpkg -i zabbix-release_2.2-1+precise_all.deb
# apt-get update
# apt-get install -y zabbix-agent
# zabbix_metrics="zabbix-metrics.infn.ponsmartcities-prisma.it"
# zabbix_watcher="zabbix-watcher.infn.ponsmartcities-prisma.it"
# sed -i -e "s/^Server=.*Server=${zabbix_metrics},${zabbix_watcher}/"
/etc/zabbix/zabbix_agentd.conf
# sed -i -e
"s/^ServerActive=.*ServerActive=${zabbix_metrics},${zabbix_watcher}/"
/etc/zabbix/zabbix_agentd.conf
```

Commented [2]: da modificare con l'indirizzo del proxy?

```
# sed -i -e "s/^Hostname=.* /Hostname=${HOSTNAME//./_}/"
/etc/zabbix/zabbix_agentd.conf
# service zabbix-agent restart
```

Per maggiore chiarezza il deploy di un servizio nella piattaforma è costituito dal seguente WF:



5.2 Tramite Ricetta Cloudify

Le ricette cloudify sono disponibili al seguente [link](#).

Di seguito è stata incollata solo la parte relativa alla installazione e configurazione degli agents nelle macchine virtuali su cui sono stati deployati i servizi PaaS.

```
#!/bin/bash

# script per installazione di zabbix agent nelle VM di Cloudify
# DNS zone
DNS_ZONE="infno.ponsmartcities-prisma.it"
zabbix_metrics="zabbix-metrics.$DNS_ZONE"
zabbix_watcher="zabbix-watcher.$DNS_ZONE"
zabbix_iaas="zabbix-iaas.$DNS_ZONE"

echo "--Configurazione Zabbix Agent INFN..."
wget http://repo.zabbix.com/zabbix/2.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_2.2-1+precise_all.deb
```

```

sudo dpkg -i zabbix-release_2.2-1+precise_all.deb
sudo apt-get -q update
sudo apt-get install -y zabbix-agent

# verifico se si tratta di una VM Management o Application
# esaminando se il nome dell'host contiene la parola manager
if [[ $HOSTNAME == *"manager"* ]]; then
    echo "$HOSTNAME is VM manager: Zabbix Agent autoconfiguring with
zabbix iaas..."
    sudo sed -i -e "s/^Server=.*Server=${zabbix_iaas}/"
/etc/zabbix/zabbix_agentd.conf
    sudo sed -i -e "s/^ServerActive=.*ServerActive=${zabbix_iaas}/"
/etc/zabbix/zabbix_agentd.conf

else
    echo "$HOSTNAME is VM Application: Zabbix Agent autoconfiguring
with zabbix metrics and watcher..."
    sudo sed -i -e
"s/^Server=.*Server=${zabbix_metrics},${zabbix_watcher}/"
/etc/zabbix/zabbix_agentd.conf
    sudo sed -i -e
"s/^ServerActive=.*ServerActive=${zabbix_metrics},${zabbix_watcher}/"
/etc/zabbix/zabbix_agentd.conf
fi

sudo sed -i -e "s/^Hostname=.*Hostname=${HOSTNAME//./_}/"
/etc/zabbix/zabbix_agentd.conf
sudo service zabbix-agent restart

exit 0

```

Commented [3]: da modificare con l'indirizzo del proxy?

Commented [4]: da modificare con l'indirizzo del proxy?

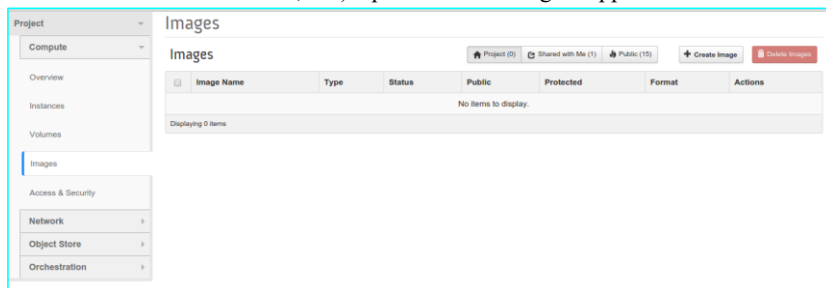
Appendice:

Installazione mediante snapshot di macchine con volume attached :

1. Nel tenant *tenant1* c'è la VM di nome *vm1* con attaccato il volume *volume1*
2. Per **clonarla** nel tenant *tenant2* bisogna seguire i passi descritti di seguito:

Per clonare la virtual machine si effettuino i seguenti passi:

3. Creare tramite interfaccia grafica uno snapshot di *vm1* (chiamato qui *snap-vm1*);
4. scaricare l'immagine *snap-vm1* digitando nel **client a linea di comando** (utilizzando come credenziali il proprio username e come tenant il *tenant1*):
\$ glance image-download --file <FILE_NAME> snap-vm1
dove **<FILE_NAME>** è il nome che si vuole dare al file dell'immagine che verrà scaricato, **snap-vm1** è il nome dello snapshot da scaricare;
5. caricare in *tenant2* il file scaricato:
 - a. accedere alla dashboard in *tenant2*
 - b. selezionare "Images", poi "Create Image" (come nella figura a pag. seguente)
 - c. nella finestra aperta, creare uno snapshot caricando il file scaricato al punto precedente;
6. istanziare in *tenant2* la VM (*vm2*) a partire dall'immagine appena creata.



Per clonare il volume:

- In *tenant1*, creare uno snapshot del volume *volume1*;
- a partire da questo snapshot, creare il *volume2*;
- da terminale, autenticandosi in *tenant1*, trovare l'ID di *volume2* digitando:
\$ cinder list
- richiedere l'autorizzazione a trasferire *volume2* su un altro tenant digitando:
\$ cinder transfer-create <ID_del_volume2>
questo comando genererà una tabella che contiene due campi da ricordare: **auth_key** e **id**;
- accedere a *tenant2* e accettare il trasferimento digitando:

```
$ cinder transfer-accept <transferID> <auth_key>
```

dove **<transferID>** e **<auth_key>** sono quelli riportati nella tabella citata. A questo punto *volume2* compare nel tenant *tenant2* e scompare dal tenant *tenant1*. *volume2* può ora essere attaccato a *vm2* e montato.