

# Migrare Zabbix 5.0 da CentOS 7 LAMP a Debian 10 LEMP

**Migrazione Zabbix 5.0 da un server CentOS 7 a Debian 10 Buster con Nginx e MySQL.**

**Iniziamo con una premessa: seguire questa procedura in produzione senza creare disservizio è possibile solo nel caso in cui si abbia un database esterno al server che ospita Zabbix. In caso contrario, il disservizio si presenterà nel momento in cui verrà spostato il database.**

Per ambienti non troppo grandi in cui il server Zabbix è anche server SQL, è bene calcolare i tempi per eseguire il DUMP completo del DB e il relativo import nel nuovo server. In tal modo sarà possibile indicare nella dichiarazione di disservizio il tempo corretto.

Iniziamo subito con la preparazione del server Debian 10 come server **stand alone**. Durante l'installazione di Debian viene richiesta la scelta dell'ambiente, a quel punto dobbiamo togliere la spunta su "Ambiente Grafico" e metterla su "server ssh" per poter prendere accesso da remoto (se necessario). Dobbiamo selezionare solo "Sistema base" senza altri pacchetti, prestando attenzione a non selezionare "Server web"

poiché di default viene installato Apache2.

Primo passo, quindi, installare WGET e il repository ZABBIX 5.0:

```
# apt update && apt install -y wget vim
#
#                                     wget
https://repo.zabbix.com/zabbix/5.0/debian/pool/main/z/zabbix-r
elease/zabbix-release_5.0-1+buster_all.deb
# dpkg -i zabbix-release_5.0-1+buster_all.deb
```

Passiamo subito all'installazione di Zabbix 5.0:

```
# apt-get update && apt install zabbix-server-mysql zabbix-
nginx-conf
```

Ora che abbiamo installato Zabbix 5.0, passiamo a NGINX, PHP 7.3 e MySQL (quest'ultimo se non abbiamo già un server dedicato per i database):

```
# apt install nginx php-fpm php-mysql
```

Ora creiamo il classico file info.php:

```
# vim /var/www/html/info.php
```

Inseriamo il codice seguente:

```
<?php phpinfo(); ?>
```

Ora apriamo il file /etc/nginx/sites-available/default e aggiungiamo le direttive per poter interpretare le pagine scritte in php:

```
location ~ /\.php$ {
include snippets/fastcgi-php.conf;
fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
}
```

Inoltre, cambiamo la porta di ascolto di NGINX così quando

metteremo il virtualhost per Zabbix non andrà in conflitto con il default.

Sostituiamo queste direttive:

```
listen 80 default_server;  
listen [::]:80 default_server;
```

con le seguenti:

```
listen 82 default_server;  
listen [::]:82 default_server;
```

ed ora riavviamo NGINX:

```
# systemctl restart nginx
```

Verifichiamo il corretto funzionamento aprendo al url `http://<P_SERVER>:82/info.php`

Passiamo all'installazione del RDBMS MySQL:

```
# apt-get install gnupg  
#  
# wget  
https://dev.mysql.com/get/mysql-apt-config_0.8.16-1_all.deb  
# dpkg -i mysql-apt-config_0.8.16-1_all.deb  
# apt-get update  
# apt-get install mysql-community-server  
# systemctl status mysql.service
```

Procediamo con la sicurezza di base del db server con il seguente comando:

```
# mysql_secure_installation
```

Ora creiamo database ed utente per Zabbix:

```
# mysql -u root -p -e "CREATE DATABASE zabbix character set utf8 collate utf8_bin;"  
# mysql -u root -p -e "CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'password';"
```

```
# mysql -u root -p -e "GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';"
```

Non ci resta che verificare la connessione al DB tramite il PHP, apriamo il nostro file info.php

```
# vim /var/www/html/info.php
```

ed inseriamo queste righe di php prestando attenzione nella valorizzazione delle variabile alle prime righe:

```
<?php
$user = "zabbix";
$password = "password";
$database = "zabbix";
$table = "todo_list";
try {
$db = new PDO("mysql:host=localhost;dbname=$database", $user,
$password);
echo "<h2>CONNESSIONE AL DATABASE AVVENUTA CON
SUCCESSO</h2><ol>";
foreach($db->query("SELECT content FROM $table") as $row) {
echo "<li>" . $row['content'] . "</li>";
}
echo "</ol>";
} catch (PDOException $e) {
print "Error!: " . $e->getMessage() . "<br/>";
die();
}
```

Qualora ci fossero problemi di connessione apriamo il file my.cnf:

```
# vim /etc/mysql/mysql.conf.d/mysql.cnf
```

aggiungiamo la entry **“default\_authentication\_plugin=mysql\_native\_password”** subito sotto **“[mysqld]”**:

```
[mysqld]
```

```
pid-file = /var/run/mysqld/mysqld.pid
socket = /var/run/mysqld/mysqld.sock
datadir = /var/lib/mysql
log-error = /var/log/mysql/error.log
default_authentication_plugin= mysql_native_password
```

Verifichiamo il corretto funzionamento aprendo al url `http://<P_SERVER>:82/info.php`, a questo punto dovremmo vedere il messaggio **“CONNESSIONE AL DATABASE AVVENUTA CON SUCCESSO”**.

A questo punto abbiamo davanti due possibilità: provare Zabbix 5.0 prima di importare il DB dal server CentOS oppure iniziare la migrazione dei dati. In questo secondo caso, passare direttamente alla connessione al DB (punto successivo).

Scegliendo la prima opzione, invece, importiamo i dati di default di Zabbix:

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz |
mysql -uzabbix -p zabbix
```

## Connessione al DB

A questo punto abbiamo NGINX e PHP correttamente funzionanti, non resta che verificare l'installazione di ZABBIX 5.0: il primo passaggio è quello di configurare il file `/etc/zabbix/zabbix.conf` con gli stessi parametri che avevamo sul server CentOS.

Arrivati al momento dalla configurazione delle credenziali di connessione al DB possiamo scegliere tra le seguenti opzioni:

- inserire i parametri del DB locale,
- inserire i dati del DB del server CentOS,
- inserire i parametri del DB centralizzato.

Dobbiamo inserire i dati di connessione qui:

```
vim /etc/zabbix/zabbix_server.conf
```

e sostituire le seguenti entry:

```
DBHost=localhost  
DBName=zabbix  
DBUser=zabbix  
DBPassword=password
```

Nel caso in cui avessimo scelto la prima opzione, cioè qualora avessimo inserito i parametri del DB locale, potremmo provare la nostra installazione di Zabbix accedendo con nome utente e password di default alla seguente url `http://<P_SERVER>`. Il nome utente è "Admin" e la password è "zabbix".

Qualora avessimo scelto la seconda o la terza opzione, cioè nel caso in cui avessimo inserito i dati del DB del server CentOS o del DB centralizzato, vedremo la GUI lavorare con i dati di esercizio pertanto l'accesso alla url `http://<P_SERVER>` avverrebbe con le utenze di produzione. Prestare attenzione: si tratta di dati di esercizio.

## **SCENARIO RDBMS ESTERNO**

La procedura è praticamente terminata, non ci resta che iniziare ad acquisire i dati invertendo gli IP tra Debian 10 e CentOS 7, in questo modo non dovremo modificare nessun proxy o agent.

## **SCENARIO MYSQL INTERNO**

Andiamo sul server CentOS e fermiamo Zabbix in modo da interrompere la scrittura dei dati sul DB:

```
# systemctl stop zabbix-server
```

Lanciamo il comando `mysqldump` per esportare tutto il DB su di un file:

```
# mysqldump -u user_zabbix -ppassword_zabbix db_zabbix >
```

```
zabbix_server.sql
```

Portiamo il file zabbix\_server.sql sul nostro server Debianb 10:

```
# scp zabbix_server.sql utente@IP_Debian:/tmp
```

Torniamo su Debian 10, fermiamo Zabbix ed importiamo il DB:

```
# systemctl stop zabbix-server
```

```
# mysql -u user_zabbix -ppassword_zabbix db_zabbix <  
zabbix_server.sql
```

Attendiamo il tempo di import del DB, una volta terminato avremo spostato ZABBIX 5.0 da CentOS a Debian 10. Non ci resta ora che iniziare ad acquisire i dati invertendo gli IP tra Debian 10 e CentOS 7 ed eseguire lo start di zabbix-server:

```
# systemctl start zabbix-server
```

Finito!

Articolo correlato: [Upgrade Zabbix 4.4 to Zabbix 5.0 Cenots 7.X](#)