

AUTOMATIZZARE I BACKUP DI WORDPRESS CON **LINUX** E **AWS**.



Introduzione

Automatizzare in modo sicuro e prevedibile i backup di WordPress è un problema attuale e necessario da affrontare per la sicurezza dei siti nostri e dei nostri clienti. Le soluzioni per affrontare il problema sono molteplici, e la maggior parte delle volte si fa riferimento a piani premium di plugin come UpDraft, Duplicator o altri.

Noi vogliamo proporre una soluzione più snella, che non faccia riferimento a codice di terze parti e che sia attuabile in modo - quasi del tutto - gratuito, a parte pochi centesimi al mese per Amazon S3.

Tutto ciò di cui avete bisogno è:

- Un accesso via SSH al server dove è installato WordPress
- Un account Amazon AWS
- Un Terminale

Questa soluzione ci consentirà di effettuare dei backup giornalieri e settimanali dei file e del database su Amazon S3, che possono poi essere richiesti e ripristinati nello spazio di produzione in qualsiasi momento.

1. Preparazione

1.1 Creazione del bucket

Dal pannello di controllo di Amazon S3 creiamo un nuovo bucket con un nome univoco, questo sarà lo spazio dove verranno caricati automaticamente i file.

1.2 Accedere al server

Per accedere al server via SSH bisogna recuperare le credenziali dal provider dell'hosting, una volta recuperate possiamo accedere tramite terminale.

1.3 Individuare la dir di wordpress

Entrati nel server individuamo la dir contenente l'installazione di WordPress e trascriviamola, per farlo possiamo usare i comandi cd per spostarci tra le dir e ls per visualizzarne il contenuto.

1.4 Installare s3cmd

Se non è installato sul server sul quale lavoriamo, installiamo e configuriamo s3cmd, software Open Source che sarà necessario per effettuare l'upload dei file in cloud tramite comando.

```
sudo apt-get install s3cmd
```

Per avviare il wizard di s3cmd dopo averlo installato lanciamo il comando `s3cmd --configure` e seguiamo il processo guidato che viene così avviato. Ci chiederà dei dati che possiamo reperire all'interno dell'area riservata di AWS.

Per altri approfondimenti sulla configurazione possiamo seguire la documentazione ufficiale del progetto a <https://s3tools.org/>.

2. Creazione dello script

Creiamo un nuovo script e rendiamolo eseguibile, per farlo possiamo usare il comando:

```
touch backup_wp.sh && chmod +x backup_wp.sh && nano backup_wp.sh
```

Modifichiamo il file dall'editor che si è aperto incollando lo script Bash a seguire, avendo premura di **cambiare la dir di riferimento dell'installazione WordPress con quella del nostro ambiente e il nome del bucket Amazon S3 adattandolo a quello creato sul nostro cloud.**

```
#!/bin/bash
echo ""
### VARS - EDIT HERE: PUT YOUR WORDPRESS DIRECTORY HERE
WP_DIR=/path/www/nomesitoweb.com
BUCKET_S3=nome-tuo-bucket
# CHECK FOR WP-CONFIG.PHP
if [ ! -d ${WP_DIR} ]; then
    echo "[+] ERROR: Directory ${WP_DIR} does not exists"
    exit
fi
if [ ! -f ${WP_DIR}/wp-config.php ]; then
    echo "[+] ERROR: No wp-config.php in ${WP_DIR}"
    exit
fi
# GREP WHAT WE NEED, PRINT ONLY THE VALUE
DB_HOST=$(cat ${WP_DIR}/wp-config.php | grep DB_HOST | awk '{ print $3 }')
DB_USER=$(cat ${WP_DIR}/wp-config.php | grep DB_USER | awk '{ print $3 }')
DB_PASS=$(cat ${WP_DIR}/wp-config.php | grep DB_PASSWORD | awk '{ print $3 }')
DB_NAME=$(cat ${WP_DIR}/wp-config.php | grep DB_NAME | awk '{ print $3 }')
# REMOVING QUOTES FROM VARIABLES
DB_HOST=${DB_HOST:1:-1}
DB_USER=${DB_USER:1:-1}
DB_PASS=${DB_PASS:1:-1}
DB_NAME=${DB_NAME:1:-1}
# DUMP OF WP DATABASE
echo "[+] Creating Database dump..."
mysqldump --user=${DB_USER} --password=${DB_PASS} --host=${DB_HOST} --databases ${DB_NAME} > wp_db_backup.sql
# CREATE TAR WITH THE TWO BZIPPED FILES
echo "[+] bzip2 running on wordpress directory and database file..."
cp -R ${WP_DIR} wp_html_files
tar cf wp_backup.tgz -j wp_html_files wp_db_backup.sql
# RENAMING TAR FILE
CURRENT_DATE=$(date +%Y%m%d)
NEW_TAR_NAME=wp_backup_${CURRENT_DATE}.tgz
mv wp_backup.tgz ${NEW_TAR_NAME}
# UPLOADING TAR TO AWS S3
echo "[+] Uploading tar to S3..."
s3cmd --storage-class=STANDARD_IA put ${NEW_TAR_NAME} s3://${BUCKET_S3}/
# REMOVING LOCAL FILES
echo "[+] Removing local files..."
rm -rf wp_db_backup.sql wp_html_files ${NEW_TAR_NAME}
echo "[+] Finish!"
```

3. Configurazione Cronjob

Lanciamo il comando

```
crontab -e
```

Se è la prima volta che lanciamo il comando ci verrà chiesto di scegliere un editor, una volta scelto questo si aprirà con i Cronjob attualmente attivi.

Incolliamo la seguente stringa avendo la premura di modificare il percorso in relazione alla dir nella quale abbiamo creato lo script bash backup_wp.sh.

```
0 1 * * * /path/to/backup_wp.sh >/dev/null 2>&1
```

Usciamo dall'editor salvando le modifiche.

In questo momento abbiamo un nuovo cronjob installato nel sistema, che va a creare un backup del sito e del database direttamente su S3 su base giornaliera. Ogni Backup del sito consisterà in un file compresso che comprende l'intera struttura del sito web e il database MySQL.

Se vogliamo modificare la frequenza dei backup possiamo andare a modificare le cifre che aprono l'ultima stringa che abbiamo visto, considerando che i primi cinque campi su ogni riga specificano con che frequenza e quando eseguire il comando che segue.

```
# ----- [m]inute: minuto (0 - 59)
# | .----- [h]our: ora (0 - 23)
# | | .----- [d]ay of month: giorno del mese (1 - 31)
# | | | .----- [mon]th: mese (1 - 12) OPPURE jan,feb,mar,apr...
# | | | | .---- [w]eek day: giorno della settimana (0 - 6) (domenica=0 o 7) OPPURE sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * comando da eseguire
```

4. Ripristino del backup

Per ripristinare un backup abbiamo preparato uno script apposito che può essere lanciato in qualsiasi momento. Creiamolo e rendiamolo eseguibile tramite:

```
touch restore_wordpress.sh && chmod +x restore_wordpress.sh && nano restore_wordpress.sh
```

Andiamo ad inserire all'interno dell'editor il seguente script **modificando la directory di destinazione dell'installazione di WordPress e il nome del bucket S3**.

```
#!/bin/bash
echo ""
# INPUT
echo "Restore Wordpress backup"
echo -n "Insert backup's date you want to restore (format DD/MM/YYYY): "
read BACKUP_DATE
YEAR=$(echo ${BACKUP_DATE} | awk -F '/' '{ print $3 }')
MONTH=$(echo ${BACKUP_DATE} | awk -F '/' '{ print $2 }')
DAY=$(echo ${BACKUP_DATE} | awk -F '/' '{ print $1 }')
SELECTED_DATE=${YEAR}${MONTH}${DAY}
### VARS - EDIT HERE: PUT YOUR WORDPRESS DIRECTORY HERE
WP_DIR=/home/webadmin/infrastructure/www/dev.thei.it
BUCKET_S3=thei-it-wp-backup
# PREPARE ENVIRONMENT
mkdir $PWD/.restoring_wordpress_temp
cd $PWD/.restoring_wordpress_temp
# DOWNLOADING TAR FROM AWS S3
echo "[+] Downloading tar from S3..."
s3cmd get s3://${BUCKET_S3}/wp_backup_${SELECTED_DATE}.tgz > /dev/null
tar xf wp_backup_${SELECTED_DATE}.tgz
# GREP WHAT WE NEED, PRINT ONLY THE VALUE
BCK_WP_DIR=$PWD/wp_html_files
DB_HOST=$(cat ${BCK_WP_DIR}/wp-config.php | grep DB_HOST | awk '{ print $3 }')
DB_USER=$(cat ${BCK_WP_DIR}/wp-config.php | grep DB_USER | awk '{ print $3 }')
DB_PASS=$(cat ${BCK_WP_DIR}/wp-config.php | grep DB_PASSWORD | awk '{ print $3 }')
DB_NAME=$(cat ${BCK_WP_DIR}/wp-config.php | grep DB_NAME | awk '{ print $3 }')
# REMOVING QUOTES FROM VARIABLES
DB_HOST=${DB_HOST:1:-1}
DB_USER=${DB_USER:1:-1}
DB_PASS=${DB_PASS:1:-1}
DB_NAME=${DB_NAME:1:-1}
# RESTORE BACKUP
echo "[+] Restoring Database..."
mysql --user=${DB_USER} --password=${DB_PASS} --host=${DB_HOST} --database=${DB_NAME} < wp_db_backup.sql
rm -rf ${WP_DIR}/*
cp -R ${BCK_WP_DIR}/* ${WP_DIR}/
# REMOVING LOCAL FILES
echo "[+] Removing local files..."
cd ..
rm -rf $PWD/.restoring_wordpress_temp
echo "[+] Finish!"
echo ""
```

Salviamo il documento e usciamo dall'editor di testo

4. Ripristino del backup

NB. Questo script non effettua un controllo delle date, quindi prima di procedere assicuriamoci di avere il backup della data che vogliamo ripristinare.

Per ripristinare un backup precedente adesso possiamo eseguire lo script `restore_wordpress.sh` posizionandoci nella dir nella quale lo abbiamo creato e lanciare:

```
./restore_wordpress.sh
```

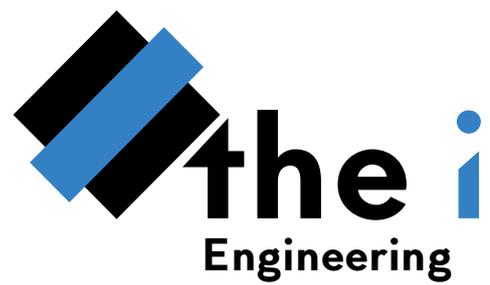
Inseriamo quindi la data del backup che vogliamo ripristinare in formato `dd/mm/YYYY` e lanciamo il restore. In questo momento lo script andrà a scaricare il file compresso da AWS, lo estrarrà, quindi eliminerà tutto il vecchio contenuto della dir del sito web, e andrà ad inserire quello nuovo direttamente dal pacchetto di Backup. Per concludere sovrascriverà anche tutto il contenuto del database completando il ripristino.

Conclusioni

Utilizzare degli script Bash per automatizzare e ripristinare i backup di WordPress è una soluzione rapida e poco dispendiosa rispetto all'utilizzo di plug-in, ma queste attività vanno svolte con cautela perché possono avere, se non fatte con cognizione di causa, effetti irreversibili sui file che elaborano.

Consigliamo pertanto di effettuare dei test su ambienti di sviluppo e non direttamente sulla produzione.

Gli script che utilizziamo e che abbiamo inserito in questa guida sono disponibili nel Repository github.com/THEI-it/wordpress-backup-script.



thei.it

facebook.com/thei.it

twitter.com/thei_it

linkedin.com/company/thei

