

# Come scaricare Ubuntu 16.04 Repo to Local Repo Appliance con crittografia SHA256 cloudrepo.key

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Problema](#)

[Sincronizzazione di Ubuntu 16.04 Repo Download to Local Repo Appliance](#)

[Soluzione](#)

## Introduzione

In questo documento viene descritto come sincronizzare un'appliance Repo locale con il sito repo.cliqrtech.com per scaricare Ubuntu 16.04 Repo sull'appliance Repo con la crittografia SHA256 cloudrepo.key.

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Interfaccia Linux
- Ambiente macchina virtuale
- Appliance Repo

### Componenti usati

Le informazioni di questo documento si basano sulle seguenti versioni software:

- CloudCenter versione 4.8.1.1
- Appliance Repo

## Premesse

CloudCenter ha iniziato a supportare Ubuntu 16.04 con la release 4.8.1.1. Tuttavia, il repository Ubuntu 16.04 non è disponibile nell'appliance CloudCenter Repo.

Pertanto, se si desidera scaricare il repository Ubuntu 16.04, sincronizzare l'accessorio Repo locale con `repo.ciqrtech.com`.

## Problema

### Sincronizzazione di Ubuntu 16.04 Repo Download to Local Repo Appliance

Se l'accessorio Repo locale si sincronizza con il file `repo.ciqrtech.com`, non sarà in grado di ottenere correttamente il repo Ubuntu 16.04 perché la crittografia `cloudrepo.key` è SHA1 e Ubuntu 16.04 richiede almeno un SHA256.

## Soluzione

Per poter utilizzare `cloudrepo.key`, è necessario creare una nuova chiave GnuPrivacyGuard (GPG) utilizzando SHA256, che scarica il repository Ubuntu 16.04 sull'appliance Repo locale.

Passaggio 1. Verificare che `repo.ciqrtech.com` sia il repository principale.

```
/usr/bin/repo_config_wizard.sh
```

Passaggio 2. Nella directory `/tmp` dell'accessorio Repo locale, copiare e incollare questo script e denominarlo `recreate_gpg_key.sh` (lo script è allegato a questo articolo di Techzone).

```
#!/bin/bash

REPO_DIR='/repo'

# Move gnupg folder to recreate keys
if [[ -d '/home/repo/.gnupg' ]]; then
    su repo -c "mv -f /home/repo/.gnupg /home/repo/gnupg_bkp"
fi

# Create gpg.conf file and add sha256 algorithm to it
gpg_conf_file='/home/repo/.gnupg/gpg.conf'
su repo -c "gpg --list-keys"
if [[ ! -f ${gpg_conf_file} ]]; then
    echo "Gpg.conf file not created. Failing in gpg install/configure"
    exit 1
fi
echo cert-digest-algo SHA256 >> ${gpg_conf_file}
echo digest-algo SHA256 >> ${gpg_conf_file}

gpg_txt='/tmp/gpg.txt'

cat << EOF > ${gpg_txt}
%echo Generating a basic OpenPGP key
Key-Type: RSA
Key-Length: 4096
Name-Real: CloudRepo
Name-Comment: GPG key for Cloud Repo
Name-Email: foo@foo.bar
Expire-Date: 0
Passphrase: cloudrepo
# Do a commit here, so that we can later print "done" :-)
```

```

%commit
%echo done
EOF

su repo -c "gpg --batch --gen-key ${gpg_txt}"
rm -f ${gpg_txt}

# Remove cloudrepo key before recreating it
cloud_repo_key=${REPO_DIR}/cloudrepo.key'
if [[ -f ${cloud_repo_key} ]]; then
    rm -f ${cloud_repo_key}
fi
gpg_key_val=$(su repo -c "gpg --list-keys" | grep "^pub" | tail -n 1 | awk '{print $2}' |
cut -d '/' -f 2 )

gpg_tmp_file='/tmp/gpg.tmp'
echo cloudrepo > ${gpg_tmp_file}
su repo -c "cat ${gpg_tmp_file} | gpg --no-tty --batch --passphrase-fd 0 --output
${cloud_repo_key} --armor --export ${gpg_key_val}"
rm -f ${gpg_tmp_file}

# Run rebuild metadata script to recreate ubuntu1604 metadata with the new key
touch /repo/debRepo/ubuntu1604/amd64/binary/tmp_file
touch /repo/debRepo/ubuntu1404/amd64/binary/tmp_file
touch /repo/debRepo/ubuntu1204/amd64/binary/tmp_file
su repo -c "/repo/scripts/rebuild_repo_metadata.sh"
rm -f /repo/debRepo/ubuntu1604/amd64/binary/tmp_file
rm -f /repo/debRepo/ubuntu1404/amd64/binary/tmp_file
rm -f /repo/debRepo/ubuntu1204/amd64/binary/tmp_file

exit 0

```

### Passaggio 3. Modificare l'autorizzazione `recreate_gpg_key.sh`.

```
chmod 755 recreate_gpg_key.sh
```

### Passaggio 4. Eseguire `recreate_gpg_key.sh` come root.

```
./recreate_gpg_key.sh
```

### Passaggio 5. Confermare che il repository di Ubuntu 16.04 sia stato aggiunto all'accessorio locale.

```
ls /repo/debRepo/
```