

Cómo descargar el Repo de Ubuntu 16.04 al dispositivo local del Repo con el cifrado cloudrepo.key SHA256

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Problema](#)

[Sincronización de la descarga del Repo de Ubuntu 16.04 al dispositivo local del Repo](#)

[Solución](#)

Introducción

Este documento describe cómo sincronizar un dispositivo local del Repo con repo.ciqrtech.com para descargar el Repo de Ubuntu 16.04 al dispositivo del Repo con el uso del cifrado `cloudrepo.key` SHA256.

Prerequisites

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- Interfaz de Linux
- Entorno de la máquina virtual
- Dispositivo del Repo

Componentes Utilizados

La información que contiene este documento se basa en estas versiones de software:

- Versión 4.8.1.1 de CloudCenter
- Dispositivo del Repo

Antecedentes

CloudCenter comenzó a soportar Ubuntu 16.04 con la versión de 4.8.1.1. Sin embargo, el repo de Ubuntu 16.04 no está disponible en el dispositivo del Repo de CloudCenter.

Por lo tanto, si usted quiere descargar el repo de Ubuntu 16.04, sincronice el dispositivo local del Repo con repo.ciqrtech.com.

Problema

Sincronización de la descarga del Repo de Ubuntu 16.04 al dispositivo local del Repo

Si el dispositivo local del Repo sincroniza para arriba con repo.ciqrtech.com, no puede obtener correctamente el repo de Ubuntu 16.04 debido al cifrado `cloudrepo.key` es SHA1 y Ubuntu 16.04 requiere por lo menos un SHA256.

Solución

Para tener `cloudrepo.key`, una nueva clave de GnuPrivacyGuard (GPG) debe ser creada con el uso de SHA256, que descarga con eficacia el repo de Ubuntu 16.04 al dispositivo local del Repo.

Paso 1. Asegúrese de que repo.ciqrtech.com sea el repositorio principal.

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

Paso 2. En el directorio local de `/tmp` del dispositivo del Repo, la copia y pega este script y lo nombra `recreate_gpg_key.sh` (el script se asocia a este artículo de 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

```

Paso 3. Change el permiso de **recreate_gpg_key.sh**.

```
chmod 755 recreate_gpg_key.sh
```

Paso 4. Ejecute **recreate_gpg_key.sh** como raíz.

```
./recreate_gpg_key.sh
```

Paso 5. Confirme que el repo de Ubuntu 16.04 está agregado al dispositivo local del Repo.

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