

Backup server

This guide walks you through configuring BorgBackup for automated backups on your Ubuntu server.

SSH Key pair

- The Backup Server uses an SSH Key pair to perform its backups and restoration jobs.
-

1. Overview

BorgBackup is a program that performs and manages backups, and is usable between the local device and a remote one.

In our case, the **Backup Server** backs up files from a remote device over SSH and manages it locally. That use-case is not common, because normally the contrary is done, and is supported natively. To do what we wanted here, we needed to use `sshfs` to *mount* the remote directory locally through **SSH**, and then perform the backup or restoration, before *unmounting* the remote directory.

The backups are automatically done every day at midnight, and then removed periodically (leaving only a few).

The restoration feature is manual, and a **Bash script** is to use in this case. The user needs to type the date in format `YYYY-MM-DD` to restore that specific snapshot into the **Webserver**.

2. Backup script

: `~/backup.sh` :

```
#!/bin/bash
```

```
# Configuration
```

```
REMOTE_USER="webserver"
REMOTE_HOST="10.0.1.2"
REMOTE_PATH="/var/www/infrasi"
MOUNT_POINT="/mnt/backup_source"
REPOSITORY="/home/backup-server/backups"
ARCHIVE_NAME="$(hostname)-$(date +%Y-%m-%d)"
BACKUP_DEST="$REPOSITORY::$ARCHIVE_NAME"

# Debugging output
echo "Backup Source: $REMOTE_USER@$REMOTE_HOST:$REMOTE_PATH"
echo "Backup Destination: $BACKUP_DEST"

# Create a mount point directory if it does not exist
if [ ! -d "$MOUNT_POINT" ]; then
    echo "Creating mount point directory..."
    mkdir -p "$MOUNT_POINT"
fi

# Mount the remote directory using sshfs
echo "Mounting the remote directory..."
if ! mountpoint -q "$MOUNT_POINT"; then
    sshfs "$REMOTE_USER@$REMOTE_HOST:$REMOTE_PATH" "$MOUNT_POINT"
    if [ $? -ne 0 ]; then
        echo "Failed to mount the remote directory. Aborting backup."
        exit 1 # Exit with a non-zero status to indicate failure
    fi
fi

# Check destination path
echo "Checking destination path..."
if [ ! -d "$REPOSITORY" ]; then
    echo "Destination directory does not exist. Creating it now."
    mkdir -p "$REPOSITORY"
    if [ $? -ne 0 ]; then
        echo "Failed to create destination directory. Aborting backup."
        exit 1 # Exit with a non-zero status to indicate failure
    fi
fi

# Initialize the Borg repository if it does not exist
```

```
if ! borg info $REPOSITORY &>/dev/null; then
    echo "Initializing the Borg repository..."
    borg init --encryption=repokey $REPOSITORY
    if [ $? -ne 0 ]; then
        echo "Borg repository initialization failed. Check the logs for details."
        exit 1 # Exit with a non-zero status to indicate failure
    fi
fi

# Run BorgBackup with options
echo "Starting Borg Backup..."
borg create -v --stats "$BACKUP_DEST" "$MOUNT_POINT"

# Check the status of the backup command
if [ $? -ne 0 ]; then
    echo "Borg Backup failed. Check the above logs for details."
    exit 1 # Exit with a non-zero status to indicate failure
else
    echo "Borg Backup completed successfully."
    # Verify the backup
    echo "Verifying the backup..."
    borg check "$BACKUP_DEST"
    if [ $? -ne 0 ]; then
        echo "Backup verification failed. Check the logs for details."
        exit 1 # Exit with a non-zero status to indicate verification failure
    else
        echo "Backup verification completed successfully."
    fi
fi

# Unmount the remote directory
echo "Unmounting the remote directory..."
fusermount -u "$MOUNT_POINT"

# Check if unmounting was successful
if [ $? -ne 0 ]; then
    echo "Failed to unmount the remote directory. Manual cleanup might be required."
    exit 1 # Exit with a non-zero status to indicate failure
fi
```

```
echo "Script completed successfully."
```

The automation of the backup is done this way:

```
crontab -e
```

```
0 0 * * * /home/user/backup_scripts/backup.sh >> /var/log/backup.log 2>&1
```

3. Restoration script

: ~/restore.sh :

```
#!/bin/bash

# Configuration
REMOTE_USER="webserver"
REMOTE_HOST="10.0.1.2"
REMOTE_PATH="/var/www/infrasi"
MOUNT_POINT="/mnt/backup_source"
REPOSITORY="/home/backup-server/backups"
ARCHIVE_NAME="$(hostname)-$(date +%Y-%m-%d)"
BACKUP_DEST="$REPOSITORY::$ARCHIVE_NAME"

# Retrieving the date of the snapshot to restore:
echo "Type the date of the snapshot you want to restore: [YYYY-MM-DD]"
read -p ">> " date

# Mount the remote directory using sshfs
echo "Mounting the remote directory..."
if ! mountpoint -q "$MOUNT_POINT"; then
    sshfs "$REMOTE_USER@$REMOTE_HOST:$REMOTE_PATH" "$MOUNT_POINT"
    if [ $? -ne 0 ]; then
        echo "Failed to mount the remote directory. Aborting restoration."
        exit 1 # Exit with a non-zero status to indicate failure
    fi
fi

# Restore the requested version into the webserver
borg extract $REPOSITORY::$(hostname)-$date $MOUNT_POINT
```

```
# Unmount the remote directory
echo "Unmounting the remote directory..."
fusermount -u "$MOUNT_POINT"

# Check if unmounting was successful
if [ $? -ne 0 ]; then
    echo "Failed to unmount the remote directory. Manual cleanup might be required."
    exit 1 # Exit with a non-zero status to indicate failure
fi

echo "Script completed successfully."
```

Revision #1

Created 9 August 2024 09:10:30 by Admin

Updated 9 August 2024 09:11:22 by Admin