Full Disaster Recovery

Overview

> **Warning:**
> - **Only** experienced system administrators should use the procedures in this document.
> - cPanel, LLC **cannot** support failed hard drives or hard drive recovery. We are **not** responsible for any data loss.

This document describes a general strategy of how to completely restore your server from a crashed or failed hard drive.

Procedure

Perform the following steps to restore your server from a crashed or failed hard drive to a new hard drive with a clean installation of cPanel & WHM.

1

Do you possess recent remote backups?

- If you answered Yes, proceed to step 4.
- If you answered No, continue to step 2.

> **Important:**
> We **strongly** recommend that you rework your backup strategy to store regularly-scheduled backups in a remote location.

2

Can you boot the failed hard drive in your server?

- If you answered Yes, continue to step 3.
- If you answered No, you will need to mount the old filesystems in a `chroot` environment, so proceed to step 4.

3

Can you access WHM from the failed hard drive on your server?

- If you answered Yes, run backups with WHM's *Backup Configuration* interface (*WHM >> Home >> Backup >> Backup Configuration*).
- If you answered No, run the `pkgacct` script to back up the relevant files.

After you complete the backup procedure, transfer the backup files to a remote location.

4

Rebuild the server.

Some datacenters and hosting providers offer system restoration and reimages. Contact your hosting provider for more information.

1. Install a new hard drive as the primary and move the old hard drive to secondary.
2. Install the operating system. Read our Installation Guide documentation for more information.

> **Note:**
>
> We **strongly** recommend that you use the CentOS Minimal ISO available from CentOS's mirrors. Issues may arise if you use an operating system ISO or a third-party drive image

3. Install cPanel & WHM with the following command:

```
cd /home && curl -o latest -L
https://securedownloads.cpanel.net/latest && sh latest
```

Read our Installation Guide documentation for more information.

Make sure to request that the datacenter installs your old hard drive as a secondary drive.

After they complete the reimage, perform one of the following actions to update cPanel to the latest version:

- Run the `upcp` command from the command line.
- Use WHM's *Upgrade to Latest Version* interface *(WHM >> Home >> cPanel >> Upgrade to Latest Version).*

5

Recover your files from the failed hard drive.

- If you answered Yes in step 1 or 2, you do not need to worry about this procedure, so proceed to step 6.
- Otherwise, perform the following steps to back up the content from the failed hard drive so that you can restore it to the new hard drive.

The procedure

1. Mount the secondary drive in a `chroot` environment.
   - Click here to expand...
     You **must** mount the drive and partitions before you can bind mount the filesystems. Run the following commands as the `root` user, where `sdb1` represents the device name of your secondary drive:

     ```
     mkdir /mnt/chroot/
     mount /dev/sdb1 /mnt/chroot/
     ```

2. Mount the necessary filesystems in a `chroot` environment.
   - Click here to expand...
     Run the following commands as the `root` user to mount the `proc`, `dev`, and `sys` filesystems:

     ```
     mount --bind /proc /mnt/chroot/proc
     mount --rbind /dev /mnt/chroot/dev
     mount --bind /sys /mnt/chroot/sys
     ```

> **Note:**

We use the `rbind` command instead of the `bind` command for the `dev` directory in order to ensure that we mount the `/dev/pts` partition properly.

From the manual pages:

```
-B, --bind mount a subtree somewhere else (same as -o bind)
-R,
--rbind mount a subtree and all submounts somewhere else
```

Then, run the following command as the `root` user in order to verify that the system properly mounted each filesystem:

```
grep chroot /etc/mtab
```

Your results should resemble the following output:

```
/dev/sdb1 /mnt/chroot none rw,bind 0 0
/mnt/chroot/proc /proc none rw,bind 0 0
/mnt/chroot/dev /dev none rw,bind 0 0
/mnt/chroot/sys /sys none rw,bind 0 0
```

3. Start a screen session.
   ⌄ Click here to expand...

   The screen command allows you to use the `chroot` environment in a session that you can reconnect to if you lose your connection. Disconnections from sessions with `chroot` environments can cause problems for services such as MySQL, which may experience INNODB issues.

   Run the following commands as the `root` user in order to start a screen session for the `chroot` command:

   ```
   screen
   chroot /mnt/chroot /bin/bash -l
   export PS1="{chrooted}$PS1"
   ```

   Your results should resemble the following output:

   ```
   {chrooted}bash-4.1#
   ```

   The system now performs as if you booted into the crashed drive.

   ### Disconnected session?

   If your session disconnects, reconnect and run the `screen -ls` command as the root user to list your active sessions:

   ```
   screen -ls
   ```

   Your results should resemble the following output:

```
There are screens on:
     12565.screen1    (Detached)
     12568.screen2    (Detached)
2 Sockets in /var/run/screen/S-root.
```

Run the `screen -x screenname` command as the `root` user to reconnect to the session where `screenname` represents the name of the session:

```
screen -x 12565.screen1
```

4. Recover the cPanel & WHM environment.
   ⌄ Click here to expand...
   Run the following command as the `root` user in order to recover the environment information from the server:

   ```
   source /etc/environment && source /etc/profile
   ```

5. Start required services from the old hard drive.
   ⌄ Click here to expand...
   Run the necessary commands as the `root` user to start any services that the pkgacct Script will require. For example, you will need the MySQL service for webmail databases, or PostgreSQL® if you use that database service.

   ```
   service mysql start
   ```

6. Back up accounts from the old hard drive.
   ⌄ Click here to expand...
   Run the following commands to back up the accounts from the old hard drive:

   ```
   cd /var/cpanel/users
   for i in `ls -1 *`; do /scripts/pkgacct $i; done
   ```

   This will store the backups in the `/mnt/chroot/home` directory. This will also store the feature lists and packages settings.

   > **Warning:**
   > We strongly recommend that you do **not** use the `rsync` command to back up accounts. This may cause issues with services such as MySQL®.

7. Back up service configurations from the old hard drive.
   ⌄ Click here to expand...
   Run the following commands to back up the service configuration:

   ```
   cd /usr/local/cpanel
   bin/cpconftool
   --modules=cpanel::smtp::exim,cpanel::system::backups,cpanel::system
   ::mysql,cpanel::system::whmconf,cpanel::easy::apache,cpanel::ui::th
   emes --backup
   ```

   The results will resemble the following output:

```
Backup Successful
/home/whm-config-backup-configuration__to__backup-10.550000-1452006
507.tar.gz
```

8. Transfer backup files to remote storage. Use the `scp` command or the `rsync` command to copy the files to a emote storage location.

> **Note:**
> Even though you could directly transfer the files from the old hard drive to the new hard drive, we **strongly** recommend that you transfer them to a remote location first. This will ensure that the backup files are safe from other hardware issues on your server that may have caused the hard drive to fail.

9. Clean up the `chroot` environment and mounts.
   - Click here to expand...

     Run the following commands to stop all of the services in the `chroot` environment, unmount the filesystems, and exit the `chroot` environment:

     ```
     service mysql stop
     umount {/proc,/dev,/sys}
     exit
     ```

     This will close the `chroot` environment. To close the screen session, run the `exit` command.

6

Restore files to the new hard drive.

Perform the following steps to restore the content from the backup files to your new hard drive in the server.

1. Copy the remote backup files to the server. Use the `scp` command or the `rsync` command to copy the files from the remote location to the new hard drive.
2. Restore feature lists to the server.
   - Click here to expand...

     Perform the following commands as the `root` user:

     ```
     cd /backups
     tar -xzvf _var_cpanel.tar.gz var/cpanel/features tar -xzvf
     _var_cpanel.tar.gz var/cpanel/features
     ```

3. Restore accounts to the server.
   - Click here to expand...

     Perform the following commands as the root user, where `BACKUPDIRECTORY/TYPE/DATE/` represents your backup directory:

     ```
     cd BACKUPDIRECTORY/TYPE/DATE/accounts/
     for archive in `ls`; do /scripts/restorepkg $archive; done
     ```

     For more information, read our The restorepkg Script documentation.

4. Restore configuration settings for Apache, Backups, cPanel themes, Exim, MySQL, and WHM to the server.
   - Click here to expand...

For more information, read our The cpconftool Script documentation.

5. If you run CloudLinux, restore CloudLinux settings to the server.
   ⌄ Click here to expand...

   For the procedure, read CloudLinux's Backup and Restore documentation.

7

Test the restored server

Test the websites, applications, and services on the new server to ensure that you have successfully restored the server's data and configuration.

Additional documentation

Suggested documentation   For cPanel users   For WHM users   For developers

- Remote Restoration
- How to Transfer Accounts and Configurations Between cPanel Servers
- How to Create a Custom Transport Script for Backups
- How to Run Backups on Locally-Mounted Remote Filesystems
- How to Manage Metadata Settings

- How to Manage Metadata Settings
- Backups

- Remote Restoration
- How to Transfer Accounts and Configurations Between cPanel Servers
- How to Create a Custom Transport Script for Backups
- How to Run Backups on Locally-Mounted Remote Filesystems
- How to Manage Metadata Settings

- WHM API 1 Functions - backup_set_list_combined
- UAPI Functions - Restore::directory_listing
- UAPI Functions - Restore::restore_file
- UAPI Functions - Restore::query_file_info
- UAPI Modules - Restore