



The enSite™ Backup Utility

for the enSite™ Management System

The enSite™ backup utility provides an easy way to back up contents of the enSite™ server to a local drive. This utility allows for the complete backup of all of the device data, hardware profiles, configurations, firmware, events, alerts and other data generated by the enSite™ server. The resulting backup file can be stored securely through existing backup procedures.

See the following:

- [Section 2.1, *Preparation*](#), on page 1
- [Section 2.2, *Running the Script*](#), on page 2
- [Section 2.3, *Scheduling the Backup Script*](#), on page 2
- [Section 2.4, *Benchmarks*](#), on page 4
- [Section 2.5, *Restoring a Backup*](#), on page 4
- [Section 2.6, *Script Parameters*](#), on page 7
- [Section 2.7, *Content of the Backup Script*](#), on page 7

2.1 Preparation

The enSite™ backup utility is run from the command line within the Ubuntu interface. It can be run only by the built-in root user. That user must have a partition labeled **enbackup** and formatted as **ext4** to store the resulting backup file.

By default, there will be a second physical disk for backup purposes. In the initial setup, Encore Networks, Inc., will set the backups to write to that second disk.

On a drive other than the included backup drive (that is, other than the second disk), use the following procedure to create a partition labeled **enbackup**.

- 1 Log in as **root**.
- 2 Create a partition on an attached drive.

- 3 Format that partition as **ext4**, and name it **enbackup**, as shown in the following example.

```
mkfs.ext4 -L enbackup /dev/sdc2
```

2.2 Running the Script

The backup script, **backup.sh**, is pre-installed in the root folder.

- 1 To run the script, execute **bash /backup.sh** in the command line.
 - » As the script executes, it indicates the number of blocks that have written to the disk.
 - » The script will create an output file called **backup.[timestamp].tar.gz**. That output file can be stored on another physical disk or on a backup network.
 - » After the backup completes, the disk is automatically unmounted, to preserve integrity in case of a power loss.

Note: You can also mount the disk manually to check empty space, the number of backups, or backup integrity.

- 2 To check backup integrity, use the command **tar tvf {filename}**. For example, to check for errors, use:

```
tar tvf /mnt/backups/backup.2018-06-07.12_06_42.tar.gz
```

- 3 After the backup check, use the following command to unmount the backup drive (to keep the backup drive safe):

```
umount /dev/disk/by-label/enbackup
```

Note: The command is **umount**, not **unmount**.

2.3 Scheduling the Backup Script

During initial configuration, Encore sets the backup script to run once a week, and creates a backup file on the **enbackup** partition.

You can use the script-scheduling command **crontab -e** to set the script to run on a schedule of your choosing. The crontab program is a scheduling program built into Ubuntu and should be run from the command line.

Note: For a full explanation of crontab, execute **crontab -h**. Or consult <https://help.ubuntu.com/community/CronHowto>.

To execute the backup script using the crontab scheduler, perform the following steps.

- 1 As root, execute the command **crontab -e** to edit the options and schedule the **backup.sh** script.

Note: Below are some examples of using **crontab** to execute the **backup.sh** script for various backup schedules.

- Example 1: Daily midnight backup:

```
0 0 * * * bash /backup.sh 1000
```

» This will run the backup script daily at 0 minutes after 0 hours (that is, at 12:00 a.m., midnight), and will store a backup.

After 1000 executions of the command, 1000 backups will have been stored. The next time the command is executed, that 1001st backup will be stored, and the oldest backup in the set will be deleted (so that the 1000 most recent backups are retained).

- Example 2: Weekly backup at midnight on Sunday:

```
0 0 * * sun bash /backup.sh 999
```

» This will run the backup script every Sunday at 12:00 a.m. (midnight), and will retain the 999 most recent backups.

Note: For details of syntax for the crontab command, see the literature for Ubuntu; also see https://www.tutorialspoint.com/unix_commands/crontab.htm.

Table 2-1 lists some common settings for scheduling backups using **crontab -e**.

Table 2-1. Some crontab Scheduling Commands¹

Minute	Hour	Day of Month ²	Month	Day of Week ³	Command	Comment
0	0	*	*	*	bash /backup.sh 999	Executes script daily, at 0 minutes of 0 hour (midnight) every day of the month. Keeps the 999 most recent backups.
0	*	*	*	*	bash /backup.sh 999	Executes script at 0 minutes of every hour. Keeps the 999 most recent backups.
0	0	*	*	sun	bash /backup.sh 100	Executes script at 0:00 (midnight) every Sunday. Keeps the 100 most recent backups.
0	0	*/2	*	*	bash /backup.sh 10	Executes script at 0:00 (midnight) every 2 days. Keeps the 10 most recent backups.

1. See <https://crontab-generator.org/> for help with preparing crontab commands.

2. The combination of an asterisk and a slash indicates intervals. For example, */15 as an argument for **Minutes** means “every 15 minutes.” The interval starts with the lowest possible value for the argument. The lowest possible value for minute is 0, so */15 is the equivalent of 0,15,30,45. The lowest possible value for the day of the month is 1. For more information, see <https://askubuntu.com/questions/2368/how-do-i-set-up-a-cron-job#2369>.

3. The crontab command also accepts a numeral for the day of the week: 0 = Sunday, 1 = Monday, and so forth.

2.4 Benchmarks

For reference, when backing up a 16 GB image to a USB 2.0 external drive, the backup script took about 7 minutes to complete, and the backup script produced a 1.3 GB tar.gz file (Table 2-2).

Table 2-2. Benchmarks for Results after Running Backup Script on Initial Image

Initial Image Size	Backup Drive	Time	File Size	File Type
16 GB	External USB 2.0 drive	~7 minutes	~1.3 GB	tar.gz

2.5 Restoring a Backup

Warning: On a live system, the database is constantly writing to its journal, and restoring a backup into a live database will cause serious synchronization errors.

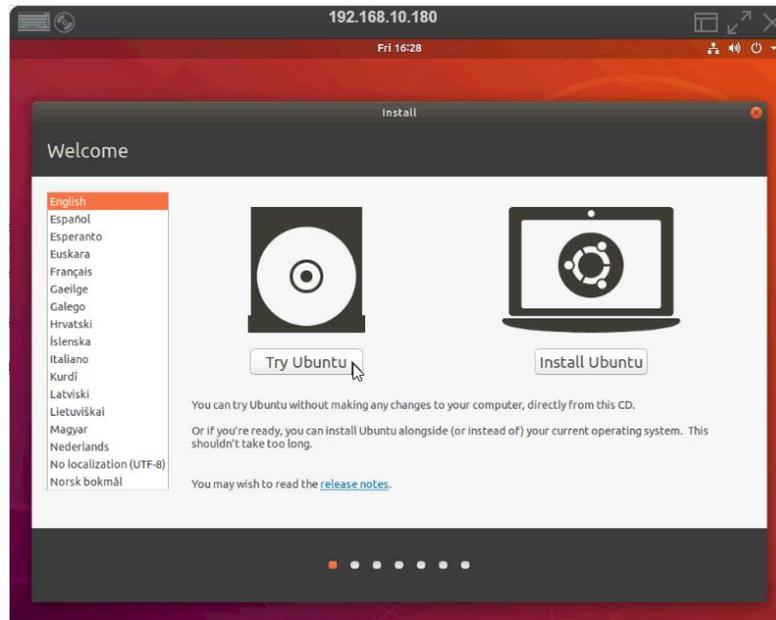
Instead of restoring an enSite™ backup to a live system, Encore Networks, Inc., recommends using one of the following practices to restore backups:

- Use the rescue environment,
or
- Use a bootable CD or USB drive to boot into a trial version of Ubuntu long enough to restore the backup.

The instructions for restoring a backup using a bootable CD/USB with a trial version of Ubuntu follow.

- 1 Place the USB or CD containing the trial version of Ubuntu into a USB drive or a CD drive, and boot the computer to that drive.
 - » The trial version of Ubuntu opens (Figure 2-1).

Figure 2-1. Ubuntu Welcome Screen



- 2 Mount the disk containing the backup files. Ubuntu Trial edition will generally recognize a USB drive and mount it for you. If it does not, see the example terminal commands below to mount the drive:

```
mount /dev/sdd1 /media/ubuntu/enbackup
```

- 3 Use the **cd** (change directory) command to go to the **root** directory. This is where you will restore the backed-up files. On the Ubuntu trial edition, this is generally the DOROOT drive. Using the terminal commands:

```
cd /
```

- 4 Then do one of the following to restore the backup:

- a To do a full restore of the device data, system files, and enSite™ code, execute the extract command with no postfixes from the **root** directory; for example:

```
sudo tar xvf /media/ubuntu/enbackup/backup.2018-06-07.12_06_42.tar.gz
```

» The back-up files will be written to the same directory in which the extract command is run. This should be the DOROOT or / directory in the Ubuntu Trial edition.

Figure 2-2. Executing the Extract Command for Full Restoration of Backup

```

192.168.10.180
ubuntu@ubuntu: /media/ubuntu/DOROOT
File Edit View Search Terminal Help
ubuntu@ubuntu: /media/ubuntu/DOROOT$ tar xvf /media/ubuntu/enbackup/backup.2018-10-05.02_00_01.tar.gz
ubuntu@ubuntu: /media/ubuntu/enbackup
File Edit View Search Terminal Help
ubuntu@ubuntu: /media/ubuntu/enbackup$ ls -al
total 7529344
drwxr-xr-x 3 root root 4096 Oct 5 16:00 .
drwxr-xr-x 4 root root 80 Oct 5 16:30 ..
-rw-r--r-- 1 root root 792092723 Oct 4 19:03 backup.2018-10-04.15_00_01.tar.gz
-rw-r--r-- 1 root root 8339456 Oct 4 22:37 backup.2018-10-04.18_37_09.tar.gz
-rw-r--r-- 1 root root 756624402 Oct 4 22:43 backup.2018-10-04.18_40_06.tar.gz
-rw-r--r-- 1 root root 757910860 Oct 4 23:03 backup.2018-10-04.19_00_01.tar.gz
-rw-r--r-- 1 root root 762065705 Oct 5 00:03 backup.2018-10-04.20_00_01.tar.gz
-rw-r--r-- 1 root root 762399477 Oct 5 01:03 backup.2018-10-04.21_00_01.tar.gz
-rw-r--r-- 1 root root 762863509 Oct 5 02:03 backup.2018-10-04.22_00_01.tar.gz
-rw-r--r-- 1 root root 763528716 Oct 5 03:03 backup.2018-10-04.23_00_01.tar.gz
-rw-r--r-- 1 root root 764346368 Oct 5 04:03 backup.2018-10-05.00_00_01.tar.gz
-rw-r--r-- 1 root root 765356087 Oct 5 05:03 backup.2018-10-05.01_00_01.tar.gz
-rw-r--r-- 1 root root 766451463 Oct 5 06:03 backup.2018-10-05.02_00_01.tar.gz

```

- b To restore only the application code of enSite, use the postfix **home**. This restores only the enSite code itself, and not the device data. For example:

```
sudo tar xvf /media/ubuntu/enbackup/backup.2018-06-07.12_06_42.tar.gz home
```

- c To restore only device database contents, use the postfix **var**. This restores only the device data, alerts, and notifications. For example:

```
sudo tar xvf /media/ubuntu/enbackup/backup.2018-06-07.12_06_42.tar.gz var
```

- d To restore only the system configuration, use the postfix **etc**. This restores only the Ubuntu system configuration, such as crontab and others. For example:

```
sudo tar xvf /media/ubuntu/enbackup/backup.2018-06-07.12_06_42.tar.gz etc
```

- 5 After you have restored the files, leave the bootable USB or CD of the Ubuntu trial edition in the drive, reboot the system, and remove the Ubuntu trial edition bootable USB or CD. The server should then boot into the restored server environment.

Note: In [step 4](#), if you performed a full restore or restored only the **/etc** directory, you may need to replace or restore changes that were made to your configuration after the backup was performed.

- 6 After the server has rebooted, log in as **root**. Then issue the following commands to restart the enSite database and restore user permissions, then to reboot the server:

```
sudo su
```

```
chmod +rx /etc/ssl/private
```

```
chmod ug+r /etc/ssl/private/ssl-cert-snakeoil.key
```

```
chown root /etc/ssl/private
```

```

sudo chown postgres /etc/ssl/private/ssl-cert-snakeoil.key

sudo chgrp postgres /etc/ssl/private/ssl-cert-snakeoil.key

service postgresql restart

reboot

```

2.6 Script Parameters

Table 2-3 summarizes script parameters used in the `backup.sh` script.

Table 2-3. Parameters in the Script

Parameter Name	Type	Usage	Description
PREFIX	string	<code>backup</code>	Prefix for backup filenames
SUFFIX	string	<code>tar.gz</code>	Defines the type of backup and the extension for backup files. The last part, <code>gz</code> , can be exchanged with <code>bz2</code> or <code>xz</code> , which also changes the algorithm to compress these files.
DELIMITER	single character	<code>.</code> [period]	Separator for file prefix, suffix, ID, and extension parts
COUNT	integer	<code>250</code>	The number of backups that will be kept after creating this backup. (That number includes the current backup.) Warning: A value of <code>0</code> means no backups will be kept! This parameter is an integer argument.
TARGET	directory name	<code>/mnt/backups</code>	The target directory for backups.
DISK	directory name	<code>by-label/enbackup</code>	The disk to use. The example uses a disk labeled <code>enbackup</code> .
DIST	string	A unique ID The enSite™ backup uses a timestamp.	Must be unique across backups; time is appropriate to use. The enSite™ backup uses a timestamp with the format <code>YYYY-MM-DD.HH_MM_SS</code> .

2.7 Content of the Backup Script

Figure 2-3 displays the content of the backup script (`backup.sh`).

Figure 2-3. Backup Script

```
[begin script]

#!/bin/bash

PREFIX="backup"

SUFFIX="tar.gz"

DELIMITER="."

DIST="$(date +%Y-%m-%d.%H_%M_%S)"

NAME="${PREFIX}${DELIMITER}${DIST}${DELIMITER}${SUFFIX}"

COUNT="${1:-10}"

TARGET="/mnt/backups"

DISK="by-label/enbackup"

test -e "/dev/disk/${DISK}" && {
    TARGET=$(mktemp -d)
    mount "/dev/disk/${DISK}" "${TARGET}"
}

grep -qi "${TARGET}" /etc/mtab || mount "${TARGET}" || {
    echo "disk required"
    exit 1
}

tar --checkpoint -pC / -zcaf "${TARGET}/${NAME}" etc home var

ls -1 "${TARGET}" | grep -oE "^${PREFIX}.*${SUFFIX}\$" | head
-n-"${COUNT}" | sed "s#^#${TARGET}/#g" | xargs rm -v

umount "${TARGET}"

rmdir "${TARGET}"

[end script]
```