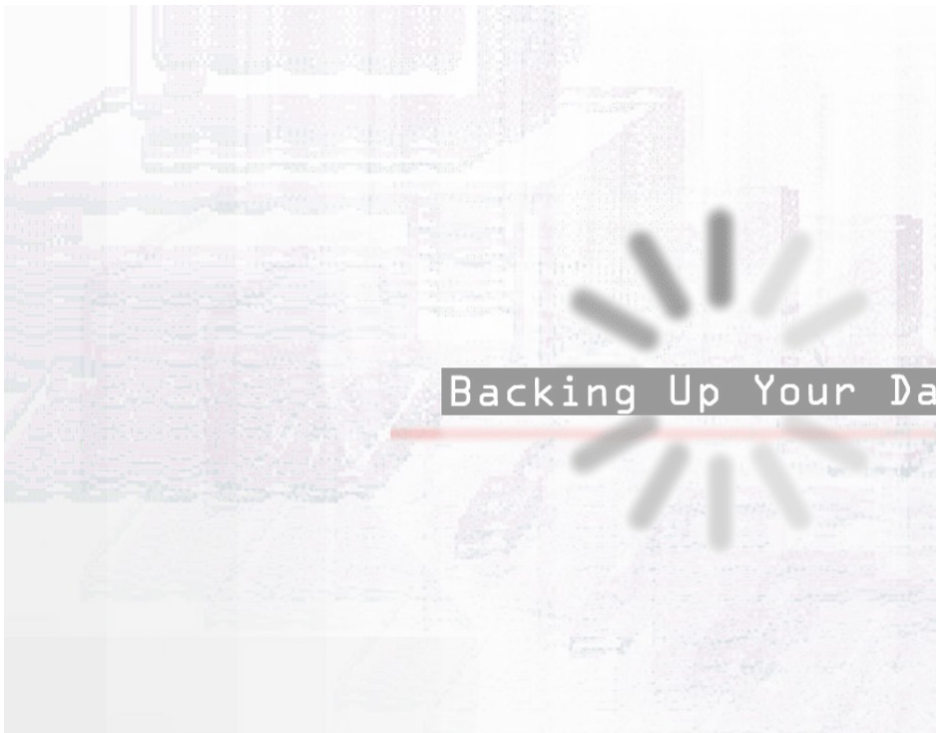
We’re integrating Linode and Akamai.

Set Up Automatic Backups to a Lin…

Backing Up Your Data

Updated Thursday, March 9, 2023, by Linode



If you store any customer or personal data on a Linode, it's important to make re
corrupted or inaccessible for any number of reasons - accidental deletions, misc
updates that don't play nicely with the rest of your configuration. Having a recen
easier to recover from these mishaps.

Assess Your Needs

Backups are not one-size-fits-all. Before you make your first backup (or create a r
what you need to back up and what tool is best for your situation.

What to Back Up

Think about the things on your Linode that would be difficult or impossible to re
common examples of data that should be backed up:

CMS websites: Database-driven websites, such as websites made with WordP
database to store content. Make sure you include a database dump in your ba

HTML websites: If you have standard HTML websites, you can probably just b

Email: If you use your Linode as a mail server, you should back up your raw e

Media: Make sure you back up your images, videos, and audio files.

Customer data: Customer data from sales and financial transactions are ofte to include a database dump in your backup.

Custom backend: If your Linode is highly customized (or took a long time to s entire Linode from the root level, or at least your software configuration setti content.

Once you have a list of items to be backed up, find where those files are on your s paths and databases for each item.

The type of backup that you make is important, too, because its format affects w think about the circumstances under which you will be making the restoration, s backup. There are two basic types of backups:

File-system backup: Copying all or part of your file system, along with its stru HTML files, software configuration files, email (in most cases), and media. If y to a Linode, it should work the way it did before. A full-server snapshot backu that preserves your entire server from a specific point in time. If you back up permissions, you'll have the content, but it may take a lot longer to get the re

Database dump: File-system backups are not always the best choice for data will of course also restore your databases, but raw database files are fairly us Running a SQL dump or something similar is better: you will get a human-rea can be imported to any other server running the same database type.

Decide whether you need a file-system backup, a database dump, or both. If you made first, and then the dump file can be saved as part of a file-system backup.

When to Back Up

The next consideration is how often you need to back up your data. This decision content on your server changes, and how critical it is that you capture those cha intervals:

Online store: At least daily

Blog or news site: As often as you update it

Development server: As often as you make changes

Game server: At least daily

Static site: Every six months, or before and after making significant changes

Email server: At least daily

Your requirements should help you decide whether you can use a manual backu

Where to Store Backups

Where to Store Backups

Next, think about where you want to store your backups. Here are some of the m

- **Same server**: This is the easiest place to store your backups, but if your serve accidentally erased, your backups will disappear too.
- **Different server**: You can store your backups on a different Linode, or a non-l option.
- **Personal device**: You can back up to your desktop computer or a portable ha probably not as secure as a professional data center, and your hardware is pr

You should also think about how many backups your storage platform can hold. two backups (an older, reliable one and a recent one), and possibly every backup out of disk space.

Backup Rotation

Finally, you should decide how long to keep your old backups and how many to s better than none, most people will want *at least two.* For example, if you replace keep any of your old ones, you would be out of luck if you discovered that your w The safest option is to store backups as frequently as possible without overwritir run out of space on your backup machine! Backup types that include compressic to store multiple backups.

Choose the Right Backup Utility

Once you know what your backup needs are, you can choose an appropriate utili good idea of the following:

- What files and databases you want to back up
- When you need to make new backups
- Where you want to store your backups
- How many old backups you want to keep on file

This guide will evaluate six different backup utilities to see how they meet these

Linode's Backup Service

Linode's Backup Service is a hassle-free backup service for your Linode. You can Linode Manager. This is a safe and easy-to-use option.

- **What**: Full-server file system backup.
- **When**: Snapshots are automatically created daily.
- **Where**: The files are stored in our secure data centers.
- **Rotation**: Backups are rotated automatically so you'll always have a daily, we

also store one snapshot of your choice indefinitely.

To configure Linode's Backup Service for your Linode, follow these instructions.

Linode's Disks

You can use the Cloud Manager to duplicate/clone your Linode's disk. This is not easy way to create a full snapshot of your Linode. Once you've duplicated the dis different Linode.

What: Full-server file system backup.

When: Duplicate disks are created manually. You have to shut down your ser

Where: The disk is stored on your Linode.

Rotation: Manual. The number of backups you can store at once depends on

See Managing Disks and Storage on a Linode to learn more about disks.

Rsync

Rsync is a free file copying utility that we highly recommend. It's a great backup t

Simple configuration. Many advanced options are also available.

Easy automation. Rsync commands can be set as cron jobs.

Efficient. Rsync only updates the files that have changed, which saves time a

You need a basic level of comfort with the command line to make the initial back

What: You set the file path for this file-system backup.

When: The basic command is manual, but you can set it to run automatically set up automatic daily backups with rsync.

Where: You set the destination. You can back up to a different folder on your home computer. As long as you can establish an SSH connection between th capable of running rsync, you can store your backups anywhere.

Rotation: Basic rotation is manual. However, with the right options, you can minimal amount of space. This will be covered later.

Rsync will be covered in more detail later in this guide. You can also read our rsyn

MySQL Backups

The data stored in your database can change quickly. Running a MySQL dump is you create a snapshot or perform another type of backup that just copies your fil preserved and should be restored properly in the context of a full-server snapsh you need.

What: MySQL databases and tables.

When: The basic command is manual, but you can set it to run automatically

Where: The backup file is saved on your server or downloaded to your home the file somewhere else if you want it stored in a different location.

Rotation: Basic rotation is manual.

To make human-readable backups of your databases that can be imported to a n instructions.

Tar

Tar can copy and compress the files on your Linode into a small backup file. This

Saves space on your backup machine

Reduces the amount of transfer used if you're saving your backup to a remot

Makes it easier to manipulate the backup since you're dealing with just one fi

On the other hand, you'll have to uncompress your backup file to make it usable through it looking for one folder.

What: You set the file path for this file-system backup.

When: The basic command is manual, but you can set it to run automatically

Where: By default, the archive file is created on the server itself. If you want t to set that up manually.

Rotation: Basic rotation is manual. The compressed nature of the backup ma

Here is a basic tar command:

```
tar pczvf my_backup_file.tar.gz /path/to/source/content
```

Explanation of flags:

p or –preserve-permissions: Preserves permissions

c or –create: Creates a new archive

z or –gzip: Compresses the archive with gzip

v or –verbose: Shows which files were processed

f or –file=ARCHIVE: Tells us that the next argument is the name for the new ar

For a more detailed discussion of tar and more examples, see Archiving and Com Zip.

Rdiff-backup

Rdiff-backup is a utility designed to make incremental backups. As their website features of a mirror and an incremental backup." You end up with a replicated ve

features of a mirror and an incremental backup. You end up with a replicated ve
ability to go back to older files as well.

What: You set the file path for this file-system backup.

When: The basic command is manual, but you can set it up to run automatica

Where : You set the destination. You can back up to a different folder on your home computer.

Rotation: Both old and new files are kept.

For information, see Using Rdiff-backup with SSHFS.

Manual Backup via Rsync

The remainder of this guide will use rsync as an example; similar steps can be us above. This section explains how to perform a one-time manual backup.

What: You set the file path for this file-system backup.

When: This is a one-time backup.

Where: The files will get stored on the machine from which you are running t logged into the server or computer where you want to store your backups.

Rotation: This tutorial does not include any automatic rotation.

Throughout this guide, the Linode you want to back up will be referred to as you computer where you are storing your backups will be referred to as the *backup_s* examples given are for a *production_server* running Ubuntu 12.04 LTS and severa *personal_computers*.

Follow these steps to make a manual backup of your Linode:

1. Install rsync on your Linode and *backup_server* by entering the following cor

```
sudo apt-get install rsync
```

2. Run the rsync command from your *backup_server* or *personal_computer*:

```
rsync -ahvz user@production_server:/path/to/source/conten
```

> **Note**
>
> For a deeper explanation of the rsync command's options and arguments, command, please see the Understanding the Rsync Command section of t

3. Type your SSH password for the *production_server* when prompted. You will are copied. At the end, you should see a confirmation message like this:

```
sent 100 bytes  received 2.76K bytes  1.00K bytes/sec
```

```
sent 100 bytes  received 2.78K bytes  1.90K bytes/sec
total size is 20.73K  speedup is 7.26
```

That's it! You can double-check the folder you designated as your local storage fo over correctly. The next sections will cover how to automate your backups.

Set Up Automatic Backups to a Linux Server

In this section, you'll use rsync to automate daily snapshot-style backups and sto folders. You will need only slightly more disk space for the backups than you use you'll be storing identical files as hard links rather than separate files. (If you hav you will need more space.)

This process is ideal for individuals who want to automatically store backups of t This is the easiest and most secure option. It also works for backing up to a Linux computer is turned on when the backup is initiated.

What: You set the file path for this file-system backup.

When: This is an automatic daily backup.

Where: The files will get stored on the machine from which you are running t logged into the server or computer where you want to store your backups. Th backups on a remote Linux server.

Rotation: All your old backups are saved. Disk space is economized by using l

Follow these steps to set up automatic backups of your Linode to a Linux server:

1. Install rsync on both servers by entering the following command:

   ```
   sudo apt install rsync
   ```

2. On your *backup_server*, generate a passwordless SSH key by entering the fol prompted to enter a password - *do not enter a password*.

   ```
   ssh-keygen
   ```

3. From your *backup_server*, copy the public key to your *production_server* by by one:

   ```
   scp ~/.ssh/id_rsa.pub user@production_server:~/.ssh/uploa
   ssh user@production_server 'echo `cat ~/.ssh/uploaded_key
   ```

4. Try connecting to your *production_server* from your *backup_server* by enteri

   ```
   ssh user@production_server 'ls -al'
   ```

5. Create a directory to store your backups on your *backup_server* by entering t

5. Create a directory to store your backups on your *backup_server* by entering t

```
mkdir ~/backups/
```

6. Try creating a manual backup and storing it in `~/backups/public_orig/` . Thi future backups will be checked. From your *backup_server*, enter the followin

```
rsync -ahvz user@production_server:~/public ~/backups/pub
```

You should see a bunch of folders whizzing by and a confirmation message si

```
sent 100 bytes  received 2.76K bytes  1.90K bytes/sec
total size is 20.73K  speedup is 7.26
```

7. Now you need to build the command for automatic scheduled backups. We'v below, but you can modify it for your needs. Run the following command ma make sure you don't get any errors:

```
rsync -ahvz --delete --link-dest=~/backups/public_orig us
```

> **Note**
>
> For an explanation of the rsync command's options and arguments, and to command, please see the Understanding the Rsync Command section of t

8. The output should be similar to the output that was generated in Step 6. Feel make sure everything was created.
9. Add the command to cron so it gets executed automatically every day. Open editing by entering the following command:

```
crontab -e
```

> **Note**
>
> If this is your first time running the command, select your favorite text edit

10. Copy and paste the following line to the bottom of the file. This is the same li frequency information added at the beginning. Use this and cron will automa server every day at 3 AM.

```
0   3   *   *   *   rsync -ahvz --delete --link-dest=~/ba
```

> **Note**
>
> For more information about cron, and to learn how to create a custom sch

> For more information about cron, and to learn how to create a custom sch
> Schedule Tasks with Cron.

Congratulations! You have now configured daily automatic snapshot-style backu
server, you'll be able to restore from a backup at any time.

Set Up Automatic Backups to a Desktop Computer

Now that you've learned how to back up your Linode to another Linux server, it's
desktop computer. There are several reasons why you may want to do this. It's a
than pay for two virtual servers, you can keep everything on your home compute
need to set up development environments on their desktop computers.

What: You set the file path for this file-system backup.

When: This is an automatic daily backup.

Where: The files are stored on the machine running the command, so make s
you want to use to store your backups. This section is designed to make back

Rotation: All of the old backups are saved. Disk space is saved by using hard l

Verify that rsync is installed on your desktop computer. Linux users can execute
`yum install rsync` to install rsync. Mac OS X already has rsync installed by defa
article for more details .

Linux

Linux users should follow the instructions presented in the previous Set Up Auto
section of this guide.

Mac OS X

OS X users can also follow the instructions presented in the previous Set Up Auto
section of this guide. The only difference is that you do not have to install rsync, a
`date` variable slightly. Your final rsync command in Step 7 should look like this:

```
rsync -ahvz --delete --link-dest=~/backups/public_orig user@
```

Your final crontab entry in Step 9 should look like this:

```
0      3       *       *       *       rsync -ahvz --delete
```

> **Note**
>
> If you run into a permissions error with cron but not when you run the comma
> password on your SSH key which doesn't normally pop up because you have i
> You might want to set up a new OS X user with a passwordless key for the purp

Windows

Windows is a bit different. You'll need to install a lot of tools that are available by
mind that Windows doesn't have the same type of file ownership and permissior
extra work to restore permissions and ownership when you restore one of your b
walkthrough that shows you how to install cwRsync for Windows, and explains h
automatic backups.

Follow these steps to set up automatic backups of your Linode to a Windows des

1. Install cwRsync. You can get the latest free version here (grab the top one, no
2. It's important that the SSH key runs as the same user as cwRsync, so first nav
command prompt, navigate to the folder where you installed cwRsync. For e

```
cd C:\Program Files (x86)\cwRsync\bin
```

3. Generate an SSH key for your computer.

```
ssh-keygen
```

4. You will have to specify a valid file path to where you want to save the key. Th
something like this for the path (make sure all of the directories already exist

```
C:\Users\user\.ssh\id_rsa
```

5. When prompted for a passphrase, just press Return. You should see the priva
directory you specified.
6. Now you need to upload your public key to the server. You can use your prefe
example will use PSCP, which is another program in the PuTTY family that let
7. Next you'll add both PSCP and cwRsync to your Path environment variable, s
command prompt from any location. These instructions are for Windows 7 a

    1. From your Start menu, open the **Control Panel**.
    2. Choose **System and Security**.
    3. Choose **System**.
    4. Choose **Advanced system settings** from the left sidebar.
    5. Go to the **Advanced** tab.
    6. Click the **Environment Variables…** button.
    7. Under **System variables**, scroll down until you find the **Path** variable. Hig

8. Do NOT delete what is currently there. You just want to add to it.
9. Add the paths to pscp.exe and cwRsync's bin directory. Separate paths wi

```
C:\Program Files (x86)\PuTTY;C:\Program Files (x86)\cw
```

10. Click **OK** until you're back to the Control Panel.
11. Restart your command prompt if you have it open.

8. Use PSCP to upload the key. In your Windows command prompt, run the follo

```
pscp -scp C:\Users\user\.ssh\id_rsa.pub user@production_s
```

9. On your *production_server*, run this command to append your new key to th

```
echo `cat ~/.ssh/uploaded_key.pub` >> ~/.ssh/authorized_k
```

10. Create a directory on your Windows machine where you will store your backu

```
mkdir %HOMEPATH%\backups
```

11. Create one backup manually, stored in `C:\Users\user\backups\public_orig` future backups will be checked. From your Windows machine, run:

```
rsync -hrtvz --chmod u+rwx user@production_server:~/publi
```

Note that these commands use Linux-style paths even for Windows: `C:\User` becomes `/cygdrive/c/Users/user/backups/public_orig/`.

This time, you will be prompted to enter your *production_server's* password. message like this:

```
sent 100 bytes  received 2.76K bytes  1.90K bytes/sec
total size is 20.73K  speedup is 7.26
```

You can `dir` the contents of `%HOMEPATH\backups\public_orig\` to verify tha

12. Add the final version of the command to your `cwrsync.cmd` file and run it on working before adding the automation.

1. From the Start menu, under All Programs, open the cwRsync folder.
2. Right-click **1.Batch example** and choose **Run as administrator**.
3. This will open up the `cwrsync.cmd` file for editing.
4. Do NOT delete any of the default contents.

4. ~~Do NOT delete any of the default contents.~~

5. At the bottom, add this line:

```
rsync -hrtvz --chmod u+rwx --delete --link-dest=/cygdr
```

> **Note**
>
> For a deeper explanation of the rsync command's options and argume
> please see the Understand the Rsync Command section of this guide.

6. Save the file.

7. Run the file with the following line for your command prompt:

```
"C:\Program Files (x86)\cwRsync\cwrsync.cmd"
```

This will create today's backup and create the correct environment for a p

You should see output similar to the output from Step 11.

13. Finally, add `cwrsync.cmd` as a daily task in Task Scheduler.

1. From the Start menu, go to > All Programs > Accessories > System Tools >

2. Click **Create Basic Task…**. The Task Wizard will pop up.

3. Fill in a name and description; "rsync backups," for example.

4. Choose **Daily** from the radio button list.

5. Set a start date (today) and time (when your server won't be busy, like 3 a
be on). It should recur every day.

6. Choose **Start a program**.

7. In the **Program/script** field, enter:

```
"C:\Program Files (x86)\cwRsync\cwrsync.cmd"
```

8. Click **Finish**.

Keep in mind that these backups use your allotted transfer, so running them very
charges.

You have now configured daily automatic snapshot-style backups. If something g
able to choose a restoration point from any day from here on forward.

Restore Your Rsync Backup

Restore Your Rsync Backup

If you followed the instructions listed in one of the sections above, your Linode is another server or a desktop computer. But what if something happens to your Li backup files to another computer? This section will show how to use rsync to res

1. Navigate to your backups directory on your *backup_server* or desktop.
2. Locate the folder with the right date.
3. Choose whether you want to restore the entire backup (`public/` in our exam
4. Upload your chosen files to the *production_server* with scp, SFTP, rsync, etc.
5. *Windows only:* Restore the correct Linux ownership and file permissions.

Maintain Your Backups

Even with automatic backups successfully configured, it is important to monitor surprises and keep the backup process efficient.

Backups to a remote server or desktop (via rsync or some other tool) count a an eye on your usage to avoid overage charges.

To set a different email notification address for a cron job, add this to your cr

```
MAILTO="user@example.com"
```

To disable email notifications for your cron jobs, add this instead:

```
MAILTO=""
```

Make sure your backup server doesn't run out of disk space. You may need to you're using the rsync backup presented in this article, the server will fill up f frequently. You can automate backup deletion if desired.

If you're using the automatic rsync backup presented here, you may want to cron command to a newer backup folder if you've made a lot of changes sinc time and disk space.

Understand the Rsync Command

Rsync is a powerful tool, but the half-dozen options in the example commands u need to customize the command, or encounter errors, it helps to have a deeper u This section will walk through the options and arguments used in the basic comr

```
rsync -ahvz user@production_server:/path/to/source/content /
```

rsync

> **Note**
>
> For a basic overview of rsync, check out the manual page.

A basic rsync command takes this form:

```
rsync copyfrom copyto
```

The file or directory you want to back up is `copyfrom`, and `copyto` is the place y
`Copyfrom` and `copyto` are arguments of the rsync command and are required. A
`copyfrom` and `copyto` arguments would look like this:

```
rsync user@production_server:~/public ~/backups/mybackup

|---| |----------------------------| |----------------|
  ^                ^                          ^
  |                |                          |
rsync           copyfrom                   copyto
```

Rsync can also run with additional options, which are included in the command

```
rsync --options copyfrom copyto
```

-ahvz

Here are some standard options for rsync:

```
-ahvz
```

These are four rsync options that have been combined into a single directive. You
this:

```
-a -h -v -z
```

These options have the following effects:

- -a or –archive: Preserves our file permissions and ownership, copies recursiv
- -h or –human-readable: Number outputs are human readable
- -v or –verbose: Displays more output
- -z or –compress: Compress file data during transfer

You can add or remove any of the rsync options. For example, if you don't need t
run:

```
-ahz
```

When creating backups, the essential option is `-a` or `--archive`.

Source Location

The `copyfrom` location is the path to what you want to back up on your *producti*
put the file path to your content on the server.

```
user@production_server:~/public
|--------------------| |------|
          ^                ^
          |                |
     SSH login           path
```

Since you're trying to copy from a remote server (the *production_server*), you sh
first. Then use a colon (`:`), followed by an absolute file path to the folder you wa

In this example, you're backing up the `~/public` directory, which is where your
followed the Hosting a Website guide. `~` is a shortcut for `/home/user/`. The trail
directory because you want to include the `public` folder itself in the backup, no

If you want to do a full-server backup from root, you should use `/*` as your path
from the backup so you don't get a lot of warnings and errors every time you run
`/run` do not contain permanent data, and `/mnt` is the mount point for other file
the `--exclude` option at the very end of the rsync command, after everything el

```
--exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*}
```

You will also need to use either `root` or a sudo-capable user for the backup, if yo
high-level directory. If you use a sudo user, you will need to either disable the ne
used, or send the password to the server. The crashingdaily blog has a good disc

Target Location

The `copyto` location is the path to where you want to store your backup on your

```
~/backups/mybackup
```

In the command for automatic backups (see below), a date variable is appended

```
~/backups/public_$(date -I)
                 |--------| <- date variable
|-------------------------|
```

```
'                       '
            ^
            |
          path
```

This is the local file path on the *backup_server* where you want to store the back
built-in `date` function to add the current date to the end of the file path. This ma
each backup, and also makes individual backups easy to find.

Cron

The following command extends the previous example to enable automatic back

```
0   3   *   *   *   rsync -ahvz --delete --link-dest=~/backu
```

The series of numbers and asterisks specifies when the cron task should be run (
to the `crontab` file earlier in this guide).

```
0       3           *           *       *        Command

^       ^           ^           ^       ^          ^
|       |           |           |       |          |

Minute Hour     Day of Month  Month  Weekday   Shell command
```

For each of the five time categories you can specify either a specific number or `*`
hour clock for hours. The example above specifies that the command should run
every day (3 am). Anything you add after the fifth number or asterisk is considere
run just as if you had typed it in your shell. You can read more about cron here.

For testing purposes, you can set the task to run at `* * * * *`, which will create
backups like this may use your allotted transfer, so running a new one every min

–delete

`--delete` is another rsync option.

```
--delete
```

With `--delete`, if a file was removed from your `copyfrom` location, it will not be
backup, even if it was in earlier backups. It will NOT remove the file from earlier b
easy to navigate.

–link-dest

This is the option that makes our archive of old rsync backups so efficient:

This is the option that makes our archive of old rsync backups so efficient.

```
--link-dest=~/backups/public_orig
```

`--link-dest` is another rsync option and very important to our incremental bac different folder names for different backups. It also lets us store multiple full bac disk space.

`--link-dest` has its own required argument, `comparison_backup_folder`. In its like this:

```
--link-dest=comparison_backup_folder
```

You can change the `comparison_backup_folder` whenever you want. The more s environment, the more efficient rsync will be.

The trailing `/` is omitted to match the `copyto` path.

Different Server Locations

This guide specifies a remote *production_server* and a local *backup_server*. How local *production_server* and a remote *backup_server*, with local backups to a dif with two remote servers. Any remote server requires an SSH login before the file

Running the rsync command from the backup server is a “pulled” backup, while is a “pushed” backup.

Local folders don’t need an SSH login, while remote folders need the SSH login a More valid rsync command examples:

```
rsync   copyfrom                copyto

rsync   /local1                 /local2/
rsync   /local                  user@remote:/remote/
rsync   user@remote:/remote     /local/
rsync   user@remote1:/remote    user@remote2:/remote/
```

All servers involved must have rsync installed, and any remote server must be ru

More Information

You may wish to consult the following resources for additional information on th the hope that they will be useful, please note that we cannot vouch for the accur materials.

- rsync Man Page

WebGnuru's rsync Tutorial

This page was originally published on Thursday, April 4, 2013.

SECURITY

Your Feedback Is Important

Let us know if this guide was helpful to you

Provide Feedback

Join the conversation.

Read other comments or post your own below. Comments must be respectful, cc
the guide. Do not post external links or advertisements. Before posting, consider
addressed by contacting our Support team or asking on our Community Site.

Careers