



Creating backups on DigitalOcean using Spaces

Creating a cron job to backup files

A cron job is a popular way to create a backup archive of important files and store your backup in remote (cloud) storage such as S3, Digital Ocean Spaces, a remote FTP server, or simply just copying it using tools such scp.

1. Navigate to <https://cloud.digitalocean.com/spaces> and click **Create a Spaces Bucket**.
2. Select a region closer to the Droplet created earlier, enter a unique name for the bucket, and click **Create a Spaces Bucket**.

In the given example, **Singapore** has been selected as the region and the bucket name is **wanderlust-backups**.



Take note:

DigitalOcean Spaces bucket names are globally unique. Do not use the same name as mentioned in the example.

Create a Spaces Bucket

Choose a datacenter region



Singapore • Datacenter 1 • SGP1



Content Delivery Network (CDN)

Deliver web assets up to 70% faster with global edge caching technology. No additional cost, standard bandwidth fees apply



Enable CDN

Finalize and create

Choose a unique Spaces Bucket name*

Names must be in lowercase. They can be between 3 and 63 characters long and may contain dashes.

wanderlust-backups

Your Spaces Bucket's origin URL: <https://wanderlust-backups.sgp1.digitaloceanspaces.com>

Select a project



Demos



Total monthly cost



Spaces subscription

\$5.00

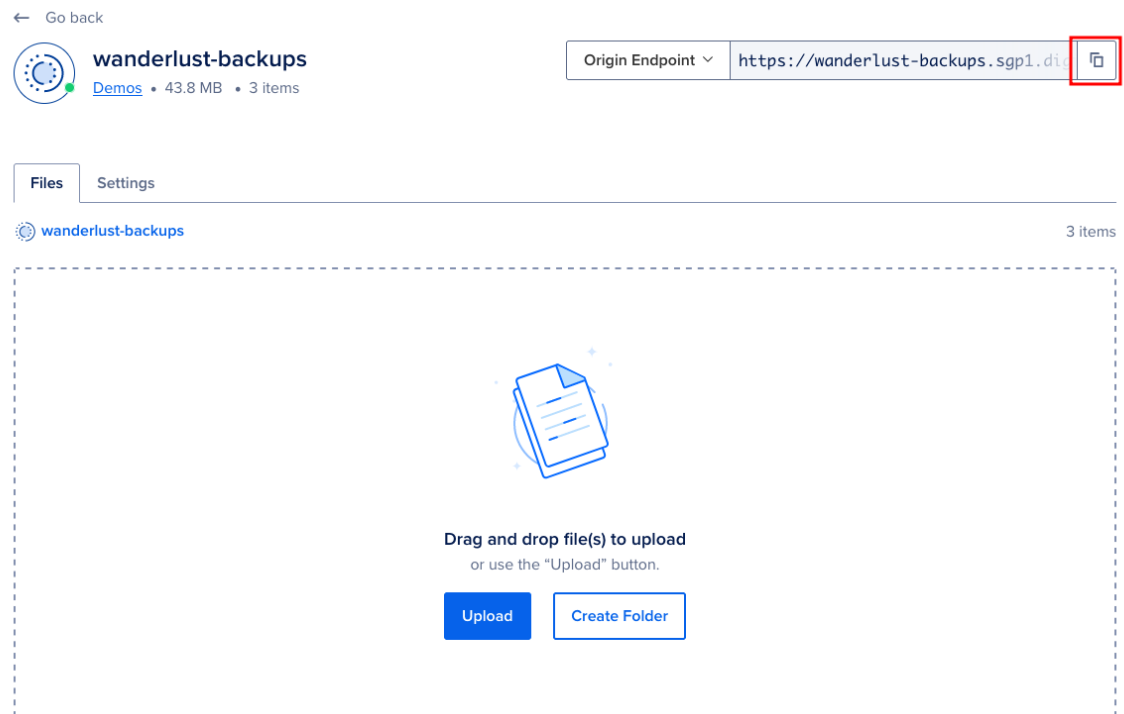


Total cost

\$5.00/month

\$0.007/hour

- Once the bucket is successfully created, copy the **Origin Endpoint** which consists of the bucket name and the region endpoint.

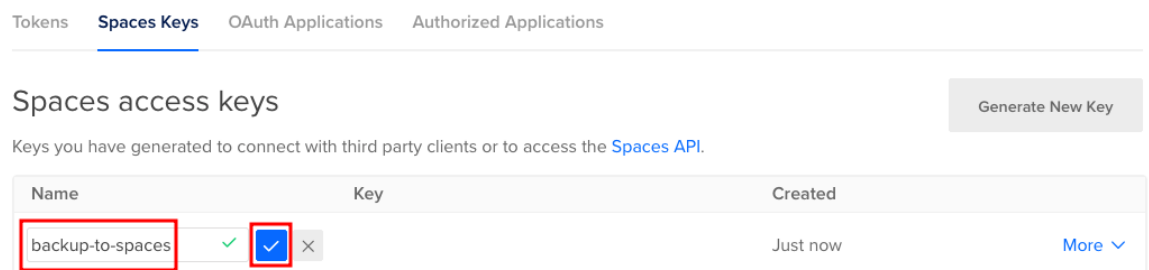


- Separate the bucket name and the region endpoint for later use.

For example, in the origin endpoint <https://wonderlust-backups.sgp1.digitaloceanspaces.com>, **wonderlust-backups** is the bucket name and **sgp1.digitaloceanspaces.com** is the region endpoint.

- As a prerequisite we should also create an access key and secret key pair for Spaces buckets. To do this, navigate to <https://cloud.digitalocean.com/account/api/spaces> and click **Generate New Key**. Next, give it a name; for this task use **backup-to-spaces** as the name and click the blue check box to create it.

Applications & API



- Copy the keys as they will be needed to configure the s3cmd utility.

Applications & API

Tokens **Spaces Keys** OAuth Applications Authorized Applications

Spaces access keys Generate New Key

Keys you have generated to connect with third party clients or to access the [Spaces API](#).

Name	Key	Created
backup-to-spaces	DO0 [REDACTED]	Just now
Secret	p0pFz7n [REDACTED]	

[More](#) ▾

- Navigate to the SSH terminal and run the following command to install **s3cmd**.

```
sudo apt install -y s3cmd
```

- Next, configure s3cmd for Spaces. Type the following command to begin the configuration process:

```
sudo s3cmd --configure
```

- Now, follow the prompts in the following steps carefully.
 - Enter the Access Key copied earlier (1 on the diagram)
 - Enter the Secret Key copied earlier (2)
 - Press enter for Default Region (3)
 - You are then prompted for the S3 Endpoint; enter the Spaces endpoint as extracted in step 4 which is of the format **<region>.digitaloceanspaces.com**. In this example it is **sgp1.digitaloceanspaces.com** (4)
 - The next prompt asks for the URL template. Enter it as **%(bucket)s.<region>.digitaloceanspaces.com** which in this example is **%(bucket)s.sgp1.digitaloceanspaces.com** (5)
 - If you wish to use encryption, enter an Encryption password (6). It is optional and can be skipped by pressing enter for this step.
 - Press enter for Path to GPG program (7)
 - Press enter again for Use HTTPS Protocol (8)
 - Press enter a third time for HTTP Proxy server name (9)

```
johndoe@ubuntu-app-server-01:~$ sudo s3cmd --configure

Enter new values or accept defaults in brackets with Enter.
Refer to user manual for detailed description of all options.

Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.
Access Key: D00[REDACTED]P 1
Secret Key: p0p[REDACTED]vfc[REDACTED]guH[REDACTED] 2
Default Region [US]: 3

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.
S3 Endpoint [s3.amazonaws.com]: sgp1.digitaloceanspaces.com 4

Use "%(bucket)s.s3.amazonaws.com" to the target Amazon S3. "%(bucket)s" and "%(location)s" vars can be used
if the target S3 system supports dns based buckets.
DNS-style bucket+hostname:port template for accessing a bucket [% (bucket)s.s3.amazonaws.com]:
%(bucket)s.sgp1.digitaloceanspaces.com 5

Encryption password is used to protect your files from reading
by unauthorized persons while in transfer to S3
Encryption password: 6
Path to GPG program [/usr/bin/gpg]: 7

When using secure HTTPS protocol all communication with Amazon S3
servers is protected from 3rd party eavesdropping. This method is
slower than plain HTTP, and can only be proxied with Python 2.7 or newer
Use HTTPS protocol [Yes]: 8

On some networks all internet access must go through a HTTP proxy.
Try setting it here if you can't connect to S3 directly
HTTP Proxy server name: 9
```

Finally, review the settings and then type **Y** to test access. Type **y** again to save the settings.

```
New settings:
Access Key: D00[REDACTED]
Secret Key: p0pFz7nW[REDACTED]
Default Region: US
S3 Endpoint: sgp1.digitaloceanspaces.com
DNS-style bucket+hostname:port template for accessing a bucket: %(bucket)s.sgp1.digitalocean
spaces.com
Encryption password:
Path to GPG program: /usr/bin/gpg
Use HTTPS protocol: True
HTTP Proxy server name:
HTTP Proxy server port: 0

Test access with supplied credentials? [Y/n] Y
Please wait, attempting to list all buckets...
Success. Your access key and secret key worked fine :-)
```

```
Now verifying that encryption works...
Success. Encryption and decryption worked fine :-)
```

```
Save settings? [y/N] y
Configuration saved to '/root/.s3cfg'
```

10. Create a folder where you can store the backup script and the temporary archive files and navigate to the folder.

```
sudo mkdir /backup
cd /backup
```

11. Next, create a new backup script file.

```
sudo vi backup-to-spaces.sh
```

12. Copy and paste the following code replacing [SPACES_BUCKET_NAME] with your actual Spaces bucket name noted down earlier.

```
#!/bin/bash
tar -czvf backup-code-$(date +%y%m%d).tar.gz /var/www/html
tar -czvf backup-apache-config-$(date +%y%m%d).tar.gz
/etc/apache2/sites-available
tar -czvf backup-certificates-$(date +%y%m%d).tar.gz /etc/letsencrypt
s3cmd put backup-*.tar.gz s3://[SPACES_BUCKET_NAME]
rm *.tar.gz
```

The above script does the following:

- Create archives of the code, apache configuration, and certificates.
- Copy the archives to Spaces using the s3cmd command.
- Delete the generated archives from the web server.

13. Save and exit the file.

14. Make the file executable


```
sudo chmod +x backup-to-spaces.sh
```


15. Test the script by running the following command:

```
sudo sh backup-to-spaces.sh
```


16. Once the script has run, navigate to DigitalOcean Spaces dashboard and refresh the page to see your backup archives.

← Go back




 **wonderlust-backups**
[Demos](#) • 43.8 MB • 3 items

Origin Endpoint ▾ <https://wanderlust-backups.sgp1.dig> 

Files Settings

 **wonderlust-backups** 3 items

Create Folder Upload

<input type="checkbox"/>	Name	Size	Last modified	
<input type="checkbox"/>	 backup-apache-config-230228.tar.gz	3.19 KB	Just now	...
<input type="checkbox"/>	 backup-certificates-230228.tar.gz	7.28 KB	Just now	...
<input type="checkbox"/>	 backup-code-230228.tar.gz	43.83 MB	Just now	...

17. Finally you will automate this process using cron. Edit the crontab by using the command:

```
sudo crontab -e
```

18. Add the following line to the end, then save and exit

```
0 6 * * * /backup/backup-to-spaces.sh
```

Your crontab should look like this:

```
GNU nano 6.4 /tmp/crontab.M18iH6/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 6 * * * /backup/backup-to-spaces.sh
```

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^/ Go To Line

This cron entry will run the script at 6 AM everyday.

You have now successfully configured backups that will be stored in Spaces.



Extra resource

Read more about Cron and Crontab [here](#).


Deleting Spaces

If you performed the backup task and are not intending to pay for continued hosting, then you must clean up the Spaces as well as they may incur costs later.

Follow the instructions below to delete the Spaces bucket(s).

1. Start by navigating to Spaces by clicking **Spaces** under Manage on the left. Then click on the ellipsis next to the bucket name and select **Delete**.

The screenshot shows the 'Spaces Object Storage' management interface. On the left, a dark sidebar contains a 'MANAGE' section with various options; 'Spaces' is highlighted with a red box and a red circle containing the number '1'. The main content area displays a table of Spaces buckets. One bucket, 'wanderlust-backups', is listed with its size (220 KB) and creation time (1 day ago). To the right of this bucket is an ellipsis menu icon, which is also highlighted with a red box and a red circle containing the number '2'. A context menu is open over this icon, showing options: 'Settings', 'Move to...', and 'Delete'. The 'Delete' option is highlighted with a red box and a red circle containing the number '3'. Below the table, there are sections for 'Learn more about Spaces Object Storage', including 'PRODUCT DOCS', 'API', and 'TUTORIALS'.

Name	Size	Created
 wanderlust-backups <small>https://wanderlust-backups.sgp1.digitaloceanspaces.com</small>	220 KB 1 items	1 day ago

Learn more about Spaces Object Storage

PRODUCT DOCS

[Spaces Object Storage overview](#)

Discover features, tips, and tools that put Spaces to work for you and your data.

API

[Spaces Object Storage API Docs](#)

Use the Spaces API to create and manage Spaces programmatically.

TUTORIALS

[Spaces Object Storage community discussion](#)

Join the Spaces Object Storage community and engage with experts and peers.

2. On the pop-up screen, confirm by typing the bucket name and click **Delete** again.

Delete Spaces Bucket



Deleting a Spaces Bucket will schedule the Spaces Bucket and all the objects inside to be permanently deleted.



Deleting a Spaces Bucket can take several days depending on the amount of data. During this time, the name of the Spaces Bucket cannot be reused and you will NOT be billed for the Spaces Bucket.

Confirm you want to permanently delete this Spaces Bucket by entering its name below.

wanderlust-backups



wanderlust-backups

4

5

Cancel

Delete

- This will initiate the delete process. Although your bucket is not deleted immediately, you are not billed from this point onwards.