

StorageVPS: Quick Start

Delimiter's StorageVPS product is based on our high-availability cloud service which makes the initial configuration a little daunting if you are not used to the terminology.

Unlike a regular VPS, Delimiter's StorageVPS comes bundled with a separate firewall which requires adds an additional configuration step which some people may not be accustomed to.

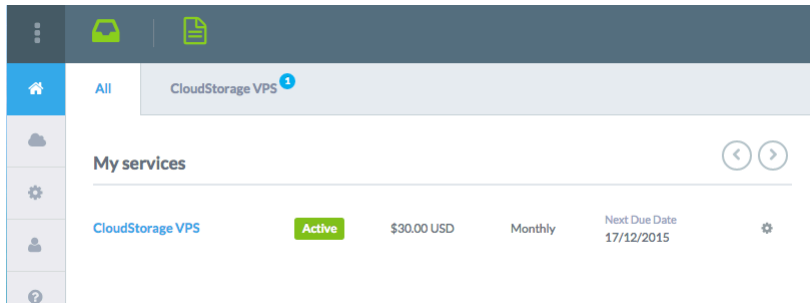
In this Quick Start we are going to install Ubuntu 14.04 and configure the VPS for incoming ssh and http (web) requests.

- [Step-by-step guide](#)
 - [Setting up the VPS](#)
 - [Setting up the Firewall](#)
 - [Test connecting to the VPS](#)
- [Related articles](#)

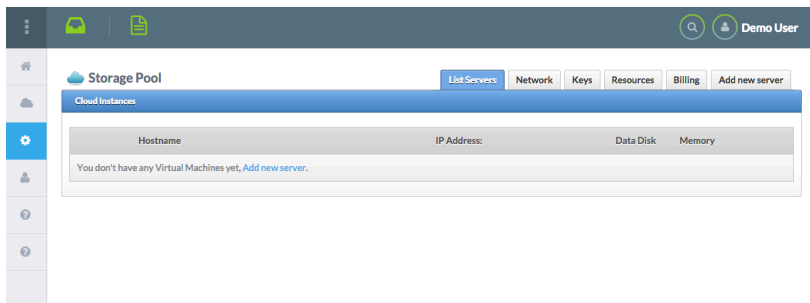
Step-by-step guide

Setting up the VPS

1. Login to the client portal and select the CloudStorage VPS:



2. As your VPS has not been setup yet, there will be no VPS's running. Click on 'Add new server':



3. You'll now create your new VPS:

Storage Pool

Create new server

Hostname:

OS Template: Centos-6-x86_64-20151026 (0.26)

RAM: 1024 MB

CPU Cores: 1

Data Disk: 1500 GB

HA-Enabled: ☐

User Data:

Hourly On Rate: \$ 0 USD

Hourly Off Rate: \$ 0 USD

Create new Virtual Machine

Available resources:

- Memory: 1024 MB
- CPU Cores: 1
- Storage: 1500 GB

Increase limits

The sliders will be automatically set to 100% of your ordered resources, if you want to reserve some disk space for snapshots or for another disk then adjust the Data Disk accordingly. (This is covered in another article). For the sake of this article, we're assuming 100% of RAM, CPU and Disk will be used>

- Set the name, OS and resources:

Storage Pool

Create new server

Hostname: test01

OS Template: Ubuntu-14.04-x86_64-20151030

RAM: 1024 MB

CPU Cores: 1

Data Disk: 1500 GB

HA-Enabled: ☒

User Data:

Hourly On Rate: \$ 0 USD

Hourly Off Rate: \$ 0 USD

Create new Virtual Machine

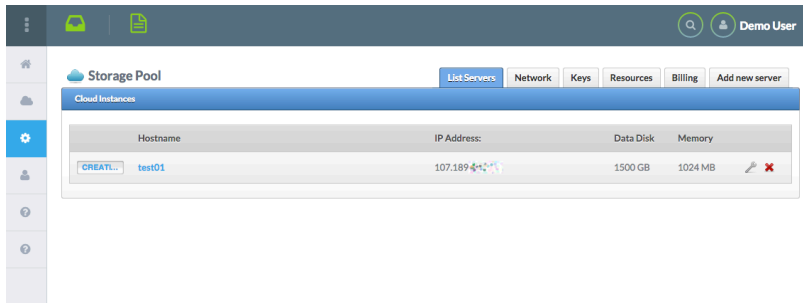
Available resources:

- Memory: 1024 MB
- CPU Cores: 1
- Storage: 1500 GB

Increase limits

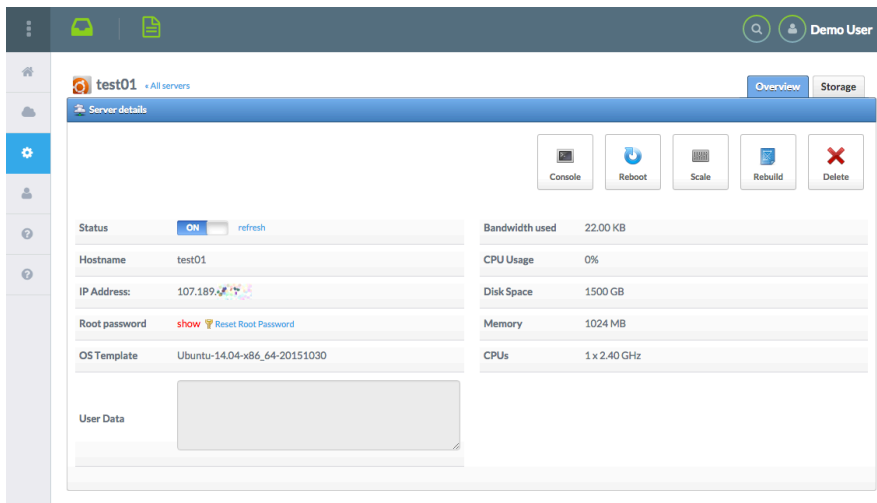
Then click the 'Create new Virtual Machine' button at the bottom

- The VPS will begin building and once the initial build is completed, you'll be redirected to the summary page:



Note that on the VPS will remain in 'Creating' state as shown on the left of the table. Your public IP is will be shown in the IP Address column.

6. In a few seconds the page will refresh (or you can manually refresh) and the VPS will be shown as running. Click on the VPS hostname, in our case test01 and the VPS status page will appear:



From this page you can see the status, the resources allocated as well as the root password that has been assigned to the system.

7. Click on 'show' to the right of Root password and take note of this. We will require it when logging into the VPS.

At this point the VPS is running, but now we need to configure the firewall to allow SSH traffic to pass to the VPS. This firewall is independent of the VPS, its a separate dedicated firewall/router which allows you to offload firewalling and port forwarding.

Setting up the Firewall

1. From the VPS status page, click on 'All servers' at the top of the page next to the host name, you'll be returned to the summary page:



2. Click on the Network tab and you'll see the private and public IPs assigned to your service:

IP Address	Netmask	Gateway	VM	Interface
10.0.0.112	255.255.0.0	10.0.0.1	test01	NIC 1

Public IP	Source NAT	Static NAT	VM	Interface
107.189	Yes	No		

The top IPs start 10.0.0 are your internal IPs, these are automatically assigned to your VPS and if you had more than one in the same pool, for failover or storage clustering, then you would see them there.

At the bottom you see your Public address, this is the IP address you'll connect to from the Internet to access your VPS.

3. We need to start by telling the firewall what traffic is permitted to go out of the VPS towards the Internet. Click on 'Outbound Firewall' under the tab bar:

Protocol	Start Port	End Port	CIDR
Nothing to display			

Add new rule:

Network: client_471861_13_cloud CIDR: 0.0.0.0/0 Protocol: All

As we assume in this example that whatever you are sending from your VPS to the Internet is good traffic, we'll just allow everything. At the bottom enter 0.0.0.0/0 meaning every IP and in protocol select All. Click submit.

If you want to limit which traffic may go your VPS towards the Internet then you can fine tune this by port and protocol. (This is covered in another article).

4. Now we want to connect the Public IP address back to the VPS, we've decided we want to allow SSH (port 22/TCP) and HTTP (port 80/TCP) to pass from the Internet to the VPS. We will start by passing SSH (port 22/TCP):

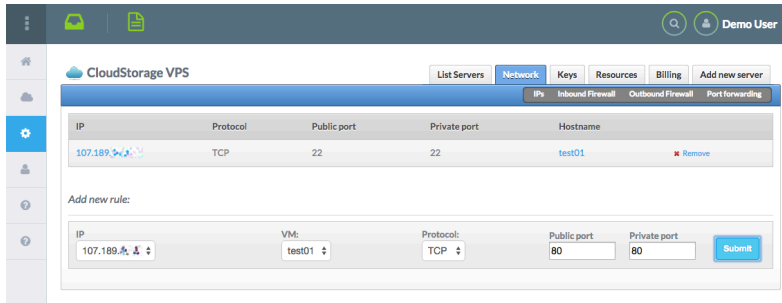
IP	Protocol	Public port	Private port	Hostname
Nothing to display				

Add new rule:

IP: 107.189 VM: test01 Protocol: TCP Public port: 22 Private port: 22

Select Protocol TCP, Public port 22 and Private port 22 then click Submit. This will forward all traffic coming to the public IP address port 22/TCP to the VPS's private IP address port 22.

5. Repeat the process for HTTP (port 80/TCP):



Select Protocol TCP, Public port 80 and Private port 80 then click Submit.

The firewall is now configured, now its time to test.

Test connecting to the VPS

1. Using SSH, we need to connect to the VPS:

```
ssh ubuntu@1.2.3.4
```

Substituting 1.2.3.4 for your Public IP address. When prompted from the password this will be your root password shown in the VPS Status page.

```
~ $ ssh ubuntu@107.189.219.219
Warning: Permanently added '107.189.219.219' (RSA) to the list of known hosts.
ubuntu@107.189.219.219's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-66-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Tue Nov 17 01:19:45 UTC 2015

System load: 0.16           Memory usage: 5%        Processes:   50
Usage of /:  57.0% of 1.32GB Swap usage:   0%        Users logged in: 0

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@test01:~$
```

Ubuntu 14.04 does not permit root login via SSH, so you have to use the user 'ubuntu' and the root password. Most other OS will accept root as the login user.

2. Now that you are connect to the server, we'll update everything and activate the webserver. We'll start by updating the packages on the server, enter:

```
sudo apt-get update
```

and once that has completing updating the package list, we'll do the actual upgrades:

```
sudo apt-get -y upgrade
```



This will take a moment to download and install anything thats new since the template was made.

3. We need to install Apache2 (web server) so we can test that port 80 is properly forwarded to the VPS:

```
sudo apt-get -y install apache2
```

Once its completed, then head to your web browser and enter <http://1.2.3.4/> Substituting 1.2.3.4 for your Public IP address. You should

see the default Apache/Ubuntu page:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

Congratulations your StorageVPS is configured for use. If you have any difficulties or need assistance then please open a support ticket and our engineers will be happy to assist.

Related articles

- [StorageVPS: Quick Start](#)