

SURVEY Share your thoughts on the state of open-source in our latest Currents survey.



Community



TUTORIAL

How To Set Up Apache Virtual Hosts on CentOS 7

CentOS Apache

By [Josh Barnett](#)

Updated March 7, 2019 • 4 versions • 1.1m

This tutorial is out of date and no longer maintained.

Not using CentOS 7?

Choose a different version or distribution.

CentOS 7



Status: Deprecated

Reason: This article is not actively maintained and has a new version available.

See Instead: [How To Install the Apache Web Server on CentOS 7](#)

Introduction

The Apache web server is the most popular way of serving web content on the Internet. It serves more than half of all of the Internet's active websites, and is extremely powerful and flexible.

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics.



Enter your email address

Sign Up

SCROLL TO TOP

Each domain that is configured will direct the visitor to a specific directory holding that site's information, without ever indicating that the same server is also responsible for other sites. This scheme is expandable without any software limit, as long as your server can handle the traffic that all of the sites attract.

In this guide, we will walk through how to set up Apache virtual hosts on a CentOS 7 VPS. During this process, you'll learn how to serve different content to different visitors depending on which domains they are requesting.

Prerequisites

Before you begin with this guide, there are a few steps that need to be completed first.

You will need access to a CentOS 7 server with a non-root user that has `sudo` privileges. If you haven't configured this yet, you can run through the [CentOS 7 initial server setup guide](#) to create this account.


You will also need to have Apache installed in order to configure virtual hosts for it. If you haven't already done so, you can use `yum` to install Apache through CentOS's default software repositories:

```
sudo yum -y install httpd
```

Next, enable Apache as a CentOS service so that it will automatically start after a reboot:

```
sudo systemctl enable httpd.service
```

After these steps are complete, log in as your non-root user account through SSH and continue with the tutorial.

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

virtual host configuration with dummy values near the end of the tutorial.

Step One – Create the Directory Structure

First, we need to make a directory structure that will hold the site data to serve to visitors.

Our **document root** (the top-level directory that Apache looks at to find content to serve) will be set to individual directories in the `/var/www` directory. We will create a directory here for each of the virtual hosts that we plan on making.

Within each of these directories, we will create a `public_html` directory that will hold our actual files. This gives us some flexibility in our hosting.

We can make these directories using the `mkdir` command (with a `-p` flag that allows us to create a folder with a nested folder inside of it):


```
sudo mkdir -p /var/www/example.com/public_html
sudo mkdir -p /var/www/example2.com/public_html
```

Remember that the portions in red represent the domain names that we want to serve from our VPS.

Step Two – Grant Permissions

We now have the directory structure for our files, but they are owned by our `root` user. If we want our regular user to be able to modify files in our web directories, we can change the ownership with `chown`:

```
sudo chown -R $USER:$USER /var/www/example.com/public_html
```

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

We should also modify our permissions a little bit to ensure that read access is permitted to the general web directory, and all of the files and folders inside, so that pages can be served correctly:

```
sudo chmod -R 755 /var/www
```

Your web server should now have the permissions it needs to serve content, and your user should be able to create content within the appropriate folders.

Step Three – Create Demo Pages for Each Virtual Host

Now that we have our directory structure in place, let's create some content to serve.


Because this is just for demonstration and testing, our pages will be very simple. We are just going to make an `index.html` page for each site that identifies that specific domain.

Let's start with `example.com`. We can open up an `index.html` file in our editor by typing:

```
nano /var/www/example.com/public_html/index.html
```

In this file, create a simple HTML document that indicates the site that the page is connected to. For this guide, the file for our first domain will look like this:

```
<html>
  <head>
    <title>Welcome to Example.com!</title>
  </head>
  <body>
    <h1>Success! The example.com virtual host is working!</h1>
```

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

typing:

```
cp /var/www/example.com/public_html/index.html /var/www/example2.com/public_html/index.
```

Now let's open that file and modify the relevant pieces of information:

```
nano /var/www/example2.com/public_html/index.html
```


```
<html>
  <head>
    <title>Welcome to Example2.com!</title>
  </head>
  <body>
    <h1>Success! The example2.com virtual host is working!</h1>
  </body>
</html>
```

Save and close this file as well. You now have the pages necessary to test the virtual host configuration.

Step Four – Create New Virtual Host Files

Virtual host files are what specify the configuration of our separate sites and dictate how the Apache web server will respond to various domain requests.

To begin, we will need to set up the directory that our virtual hosts will be stored in, as well as the directory that tells Apache that a virtual host is ready to serve to visitors. The `sites-available` directory will keep all of our virtual host files, while the `sites-enabled` directory will hold symbolic links to virtual hosts that we want to publish. We can make both directories by typing:

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

temporarily enable and disable virtual hosts this way).

Next, we should tell Apache to look for virtual hosts in the `sites-enabled` directory. To accomplish this, we will edit Apache's main configuration file and add a line declaring an optional directory for additional configuration files:

```
sudo nano /etc/httpd/conf/httpd.conf
```

Add this line to the end of the file:

```
IncludeOptional sites-enabled/*.conf
```

Save and close the file when you are done adding that line. We are now ready to create our first virtual host file.

Create the First Virtual Host File

Start by opening the new file in your editor with root privileges:

```
sudo nano /etc/httpd/sites-available/example.com.conf
```


Note: Due to the configurations that we have outlined, all virtual host files *must* end in `.conf`.

First, start by making a pair of tags designating the content as a virtual host that is listening on port 80 (the default HTTP port):

```
<VirtualHost *:80>
```

```
</VirtualHost>
```

Next we'll declare the main server name `www.example.com`. We'll also make a server

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

```
    ServerName www.example.com
    ServerAlias example.com
</VirtualHost>
```

Note: In order for the `www` version of the domain to work correctly, the domain's DNS configuration will need an A record or CNAME that points `www` requests to the server's IP. A wildcard (*) record will also work. To learn more about DNS records, check out our [host name setup guide](#).

Finally, we'll finish up by pointing to the root directory of our publicly accessible web documents. We will also tell Apache where to store error and request logs for this particular site:

```
<VirtualHost *:80>

    ServerName www.example.com
    ServerAlias example.com
    DocumentRoot /var/www/example.com/public_html
    ErrorLog /var/www/example.com/error.log
    CustomLog /var/www/example.com/requests.log combined
</VirtualHost>
```


When you are finished writing out these items, you can save and close the file.

Copy First Virtual Host and Customize for Additional Domains

Now that we have our first virtual host file established, we can create our second one by copying that file and adjusting it as needed.

Start by copying it with `cp`:

```
sudo cp /etc/httpd/sites-available/example.com.conf /etc/httpd/sites-available/example2
```

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

domain. When you are finished, your second virtual host file may look something like this:

```
<VirtualHost *:80>
    ServerName www.example2.com
    DocumentRoot /var/www/example2.com/public_html
    ServerAlias example2.com
    ErrorLog /var/www/example2.com/error.log
    CustomLog /var/www/example2.com/requests.log combined
</VirtualHost>
```

When you are finished making these changes, you can save and close the file.

Step Five — Enable the New Virtual Host Files

Now that we have created our virtual host files, we need to enable them so that Apache knows to serve them to visitors. To do this, we can create a symbolic link for each virtual host in the `sites-enabled` directory:


```
sudo ln -s /etc/httpd/sites-available/example.com.conf /etc/httpd/sites-enabled/example
sudo ln -s /etc/httpd/sites-available/example2.com.conf /etc/httpd/sites-enabled/example2
```

When you are finished, restart Apache to make these changes take effect:

```
sudo apachectl restart
```

Step Six — Set Up Local Hosts File (Optional)

If you have been using example domains instead of actual domains to test this procedure, you can still test the functionality of your virtual hosts by temporarily

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

Note: Make sure that you are operating on your local computer for these steps and not your VPS server. You will need access to the administrative credentials for that computer.

If you are on a Mac or Linux computer, edit your local `hosts` file with administrative privileges by typing:

```
sudo nano /etc/hosts
```

If you are on a Windows machine, you can find instructions on altering your hosts file [here](#).

The details that you need to add are the public IP address of your VPS followed by the domain that you want to use to reach that VPS:

```
127.0.0.1    localhost
127.0.1.1    guest-desktop
server_ip_address example.com
server_ip_address example2.com
```

This will direct any requests for `example.com` and `example2.com` on our local computer and send them to our server at `server_ip_address`.

Step Seven — Test Your Results

Now that you have your virtual hosts configured, you can test your setup easily by going to the domains that you configured in your web browser:

```
http://example.com
```

You should see a page that looks like this:

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

If all of the sites that you configured work well, then you have successfully configured your new Apache virtual hosts on the same CentOS server.

If you adjusted your home computer's `hosts` file, you may want to delete the lines that you added now that you've verified that your configuration works. This will prevent your hosts file from being filled with entries that are not actually necessary.

Conclusion

At this point, you should now have a single CentOS 7 server handling multiple sites with separate domains. You can expand this process by following the steps we outlined above to make additional virtual hosts later. There is no software limit on the number of domain names Apache can handle, so feel free to make as many as your server is capable of handling.

About the authors



Josh Barnett

has authored 12 tutorials.

Still looking for an answer?



Ask a question



Search for more help

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

MiteshGanatra November 9, 2014

- 1 Here is the similar guide on How To Set Up Apache Virtual Hosts on CentOS:
<http://blog.miteshganatra.com/how-to-set-up-apache-virtual-hosts-on-centos>

[Reply](#) [Report](#)

ecehilalyigit November 16, 2014

- 1 hello ,
i took this fault when I wrote : “sudo apachectl restart”

Job for httpd.service failed. See ‘systemctl status httpd.service’ and ‘journalctl -xn’ for details.

I can not start apache server
could you help please ?
thank you .

[Reply](#) [Report](#)

J05huaMClure January 17, 2015

- 1 I had this same error, but I found that my vhost conf file had the wrong path in it.
Once the path was corrected everything started working.

[Reply](#) [Report](#)

ravichopu January 27, 2015

- 0 plx tell me how you resolve this problem. i am getting some problem.

[Reply](#) [Report](#)

J05huaMClure September 6, 2016

- 0 It depends on what your configuration is... I have multiple websites running on a single droplet, so I use the apache2 vhost configuration to direct traffic to the appropriate app.

Whenever I create a new app on the server, I run the following commands:

Step 1: Create the application root.

You may need to run a `chmod` and `chown` on this directory to ensure that the server has permissions. Follow [this link](#) for more info.

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics.



Sign Up

```
sudo nano /etc/apache2/sites-available/mysite.com.conf
```

Step 3: Define the app configuration. Use this sample as a template.

```
<VirtualHost *:80>
    ServerName mysite.com
    ServerAlias www.mysite.com
    #AllowOverride All
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/mysite.com
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # This is for Drupal8 clean URLs.
    <Directory /var/www/html/mysite.com>
        RewriteEngine On
        RewriteBase /
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
    </Directory>
</VirtualHost>
```

Step 4: Enable the site and restart the server.


```
sudo a2ensite mysite.com.conf
```

```
sudo systemctl restart apache2
```

[Reply](#) [Report](#)

[alhelal](#) April 11, 2018

, a2ensite is not available in CentOS

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

[Reply](#) [Report](#)[heyokdangce0349c35752d302d](#) November 29, 2018

0 I have disabled SELinux and restart centos7 and fixed problems
<https://linuxize.com/post/how-to-disable-selinux-on-centos-7/>

now i can put 2 lines below in virtualhost and no problems with restart httpd
ErrorLog /var/www/example2.com/error.log
CustomLog /var/www/example2.com/requests.log combined

thank you.

[Reply](#) [Report](#)[heyokdangce0349c35752d302d](#) December 7, 2018

0 you shouldn't disable SELinux, this is good solution to fix problem.

<https://muchbits.com/apache-selinux-vhosts.html>

[Reply](#) [Report](#)[heyokdangce0349c35752d302d](#) December 7, 2018


0 chcon -R -t httpdsyscontentt /var/www/lichsu.org/publichtml/
chcon -R -u systemu /var/www/lichsu.org/publichtml/

[Reply](#) [Report](#)[heyokdangce0349c35752d302d](#) December 24, 2018

0 How to Linux re apply or restore SELinux security labels context
Q. How do I set back the file security contexts to specific files or directories under SELinux enabled kernel? I have restored the /wwwdata /domain1, domain2 directories but my security labels are incorrect.

A. If files or directories restored from backup or compied from other source over network/medium you need to restore back SELinux security labels.

Use restorecon command to set file security contexts. This command is primarily used to set the security context (extended attributes) on one or more files. It can be run at any time to correct errors, to add support for

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

restorecon -Rv /wwwdata

Where,

-R: Change files and directories file labels recursively

-v: Show changes in file labels.

Read man page of restorecon for more information.

[Reply](#) [Report](#)

[lelinh](#) July 22, 2017

0 Hi, Have you resolved it yet? I have the same error and can not fix it.

[Reply](#) [Report](#)

[eklect](#) November 27, 2014

0 I had the same errors as everyone else, this is what I did to fix it →

<http://www.iameklect.com/2014/11/27/creating-virtual-hosts-on-centos-7/>

[Reply](#) [Report](#)

[br4v37](#) December 25, 2014

0 Thank you, can u poste a tutorial about set vhost with diffrent users . in ubuntu i used mpm-itk, but when i tried httpd-itk on centos 7 nothing works with me.

[Reply](#) [Report](#)

[smartdesigner](#) January 26, 2015

1 There are only old httpd-itk in centos 7 repo (old for apache 2.2, but in centos 7 using new apache 2.4 by default)

[Reply](#) [Report](#)

[br4v37](#) March 10, 2015

0 can u explain more plz !!

[Reply](#) [Report](#)

[thebadsysadmin](#) January 3, 2015

0 [REQUEST] can you place write a tutorial on how to write a puppet module to do this? So lets say i want to have this done to multiple server nodes, how will you write a puppet module to do this?

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics.



Sign Up

First off thanks for that post, it's real useful, but.

- 0 Is a machine placed in the same network as the server is, supposed to be able to get those virtual-hosts content? (ofc the machine has correct name-ip matching configuration.

I am deploying a server in a shell-only CentOS and want to develop a web in an other machine.

(Now I can access the server, but it always returns the default "index.html" no matter the virtual-host I am requesting to)

[Reply](#) [Report](#)

[thmsdwld](#) January 20, 2015

- 1 Great tutorial!!

However, I have one remark.

When changing the httpd.conf in step four you need to add

```
IncludeOptional sites-enabled/*.conf
```

but what you should add is

```
NameVirtualHost *:80
```

```
IncludeOptional sites-enabled/*.conf
```

This is just a small correction, and it won't give you this

[warn] _default_ VirtualHost overlap on port 80, the first has precedence
when restarting **apachectl**.

[Reply](#) [Report](#)

[leeDavies](#) February 27, 2015

- 0 Adding NameVirtualHost *:80 worked for me, thanks!

[Reply](#) [Report](#)

[khalilamar](#) May 23, 2018

- 0 **[deleted]**

[Reply](#) [Report](#)

[khalilamar](#) May 23, 2018

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics.



Sign Up

course includes SELinux making it good for the DMZ in which it lives. Without pre-creating the log files and assigning the correct security context Apache fails to restart.

Insert Step 4a:

```
touch /var/www/example.com/error.log
touch /var/www/example.com/requests.log
chcon --reference /var/log/httpd/error_log /var/www/example.com/error.log
chcon --reference /var/log/httpd/access_log /var/www/example.com/requests.log

touch /var/www/example2.com/error.log
touch /var/www/example2.com/requests.log
chcon --reference /var/log/httpd/error_log /var/www/example2.com/error.log
chcon --reference /var/log/httpd/access_log /var/www/example2.com/requests.log
```

Since enabling virtual hosts disables the default web server, if you have DNS configured for an available site but it is not enabled, then it will arbitrarily go to the alphabetically first enabled site, which is probably not what you want! The way round this would be to reconfigure your default web server as a virtual server and make sure it comes first, giving the config file a name like 0_default.conf

[@thmsdwld](#) NameVirtualHost is deprecated and due for removal. It's not needed unless you have elected to install an older Apache on CentOS 7!

[Reply](#) [Report](#)

[vkr](#) December 9, 2016

1 Your step 4a did a magic for me. As others said article needs an update as it is hitting and helping many in google.

[@Sandvika](#) , I didn't quite get your point about DNS & 0_default.conf though

[Reply](#) [Report](#)

[jstok](#) July 18, 2018

0 +1 this should be in the main article body

[Reply](#) [Report](#)

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics.



Sign Up

[Reply](#) [Report](#)**ravichopu** January 25, 2015

- 0 when i adding *IncludeOptional sites-enabled/.conf* line in **httpd.conf* then in restarting httpd i got error *Invalid command 'IncludeOptional', perhaps misspelled or defined by a module not included in the server configuration*

[Reply](#) [Report](#)**djain123** February 26, 2015

- 0 Hi Ravi, I get the same misspelled error. Were you able to resolve this?

[Reply](#) [Report](#)**owen744626** April 2, 2015

- 0 IncludeOptional is an option for later versions of Apache (v2.4.x). You must be running an older version; use this instead: `Include sites-enabled/*.conf`

[Reply](#) [Report](#)**javiersalazar** March 5, 2015

- 0 Help me , i need change the directoryroot `/var/www/html` to `/a/Pro/Www` , i change this part on `httpd.conf` >

DocumentRoot: The directory out of which you will serve your

documents. By default, all requests are taken from this directory, but symbolic links and aliases may be used to point to other locations.

MC DocumentRoot “/var/www/html”

DocumentRoot “/a/Pro/Www”

but dont work, where ist wrong?

[Reply](#) [Report](#)

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

[Reply](#) [Report](#)[email1028841](#) May 1, 2015

0 If you were getting an error on Step #5 about the service failing try the following....

1. Go back to your httpd.conf file

```
sudo vi /etc/httpd/conf/httpd.conf
```

2. Go back to the bottom where you added

```
IncludeOptional sites-enabled/*.conf . Change it to the following  
IncludeOptional sites-available/*.conf . Save and exit.
```

3. Now try restarting.

```
sudo apachectl restart
```

4. You should now be able to browse to example.com(or your domain) and see the example page or your content.

[Reply](#) [Report](#)[MrMarcie](#) July 10, 2015

0

```
]$ systemctl status httpd.service
```

```
httpd.service - The Apache HTTP Server
```

```
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
```

```
Active: failed (Result: exit-code) since Fri 2015-07-10 15:13:13 CEST; 31s
```

```
Process: 2870 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=1/F
```

```
Process: 2869 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited,
```

```
Main PID: 2869 (code=exited, status=1/FAILURE)
```

and

```
<^>$ journalctl -xn
```

```
No journal files were found.<^>
```

Now what? If i go to http://myip I get no connection

[Reply](#) [Report](#)

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

[Reply](#) [Report](#)**sethubv** June 16, 2015

0 Hi Josh Barnett,

Very good steps. It is perfect and it worked for me. Could you please help me to create the same for SSL also.

Regards,

Sethu

[Reply](#) [Report](#)**tiberiu** June 20, 2015

0 Works like a charm! Thank you, but, my primary IP from VPS always show the latest virtual host added.

How I can set up a default page for VPS IP?

[Reply](#) [Report](#)**cansurmeli** July 9, 20150 Really nice tutorial! Although I've followed it step by step and checked multiple times, my configuration does not work. When I visit any of my sites I get the error
cannot open page because it could not connect to the server. Is it a firewall issue?[Reply](#) [Report](#)**MrMarcie** July 9, 2015

0 After :

```
<html>
<head>
<title>Welcome to Example.com!</title>
</head>
<body>
<h1>Success! The example.com virtual host is working!</h1>
</body>
</html>
```

Save and close the file when you are finished.

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

EDIT:

Waht I did is with cd i go to var/www/domainname.com/public_html/ and than it is possible to make the index.html. So it seems a rights issue, what have I done wrong??

[Reply](#) [Report](#)

[satadru](#) August 23, 2015

- 0 Okay, How to disable vHost from sites-enabled ??

[Reply](#) [Report](#)

[neutronite](#) September 3, 2015

- 0 Hello Josh Barnett

i know this is coming rather too late, please i have a situation. i have created two vhost on my centos (e.g www.example1.com and www.example2.com). Example1.com was configured to listen on port 80 and 8080 example2.com was configured to listen on port 80, but whenever i try www.example2.com:8080 it display the content of www.example1.com please how do i make only example1.com bind to port 8080 such that no other domain can use it. i hope you understand my question. thanks

[Reply](#) [Report](#)

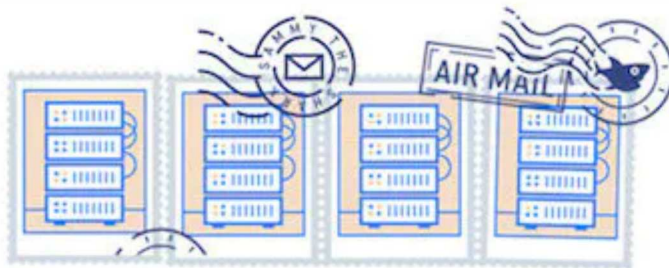
Load More Comments



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

**GET OUR BIWEEKLY NEWSLETTER**

Sign up for Infrastructure as a
Newsletter.

**HUB FOR GOOD**

Working on improving health
and education, reducing
inequality, and spurring
economic growth? We'd like to
help.



Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

Sign Up

nonprofits.

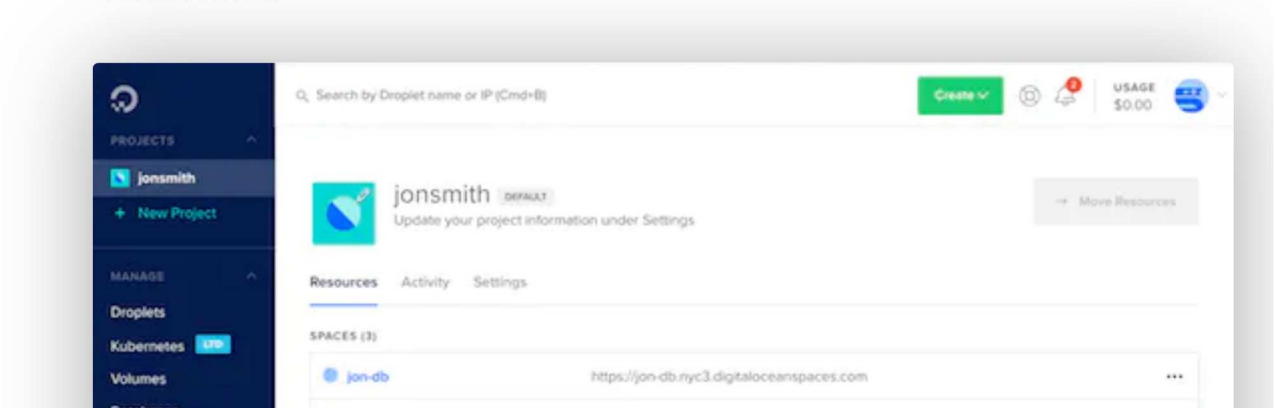
Featured on [Community](#) [Kubernetes Course](#) [Learn Python 3](#) [Machine Learning in Python](#)
[Getting started with Go](#) [Intro to Kubernetes](#)

DigitalOcean Products [Virtual Machines](#) [Managed Databases](#) [Managed Kubernetes](#)
[Block Storage](#) [Object Storage](#) [Marketplace](#) [VPC](#) [Load Balancers](#)

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you’re running one virtual machine or ten thousand.

[Learn More](#)



© 2020 DigitalOcean, LLC. All rights reserved.

[Company](#)

[About](#)

[Leadership](#)

[Blog](#)

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. ×

[Sign Up](#)

Products


Pricing
Products Overview
Droplets
Kubernetes
Managed Databases
Spaces
Marketplace
Load Balancers
Block Storage
API Documentation
Documentation
Release Notes

Community

Tutorials
Q&A
Tools and Integrations
Tags
Product Ideas
Write for DigitalOcean
Presentation Grants
Hatch Startup Program
Shop Swag
Research Program
Open Source
Code of Conduct

Contact

Get Support
Trouble Signing In?
Sales
Report Abuse
System Status

Sign up for our newsletter Get the latest tutorials on SysAdmin and open source topics. 

Sign Up
