

Zero to Server

in 20 minutes + \$5/month



Assaf Gordon
<https://github.com/agordon>
assafgordon@gmail.com

Goal

- A Linux Server (w/ usable distro)
- Shell Access (w/ Root Access)
- Public IP + DNS record
- Apache Server (or NGINX)
- Developer-friendly (Perl/Python/PHP/Ruby/etc.)
- Infrastructure-friendly
(Postgres/MySQL/Redis/MongoDB/Memcache/etc.)
- And also:
 - Cheap
 - Fast
 - Reliable
 - Easy to use

Common Solutions

- Personal Computer at home + Dynamic DNS
- Work/School servers (BEST. EVER.)
- “Cloud” (??) services
 - Amazon (1-year Free-tier – highly recommended)
 - Linnode
 - Digital-Ocean
 - Many many many others, YMMV

Digital Ocean

- <https://www.digitalocean.com>
- “For Developers, By Developers”
 - (Lots of marketing fluff, but they certainly deliver)
 - Large community, many tutorials: <https://www.digitalocean.com/community>
- Cheap
 - \$5/month for 1CPU / 512MB RAM / 20GB DISK / 1TB transfer
 - Charged by the hour, for active instances (\$0.007/hour for above setup)
- Fast
 - New image start in 60 seconds. Really.
 - Support tickets answered within minutes. Really.
- Other niceties for developers
 - Snapshots + Periodic backups
 - Root password sent by email or pre-configured SSH key
 - Web-based VNC console access when sh*t goes down
 - Start server from pre-made distros, or from your own snapshot
- Did not try, and cannot attest to
 - “Web Scale”, API, Built-in DNS



<https://twitter.com/digitalocean>

The Plan

- Create new Droplet (DigitalOcean's lingo for a running instance)
- Setup
 - Create non-root user
 - Install Apache
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

The Plan

- **Create new Droplet** (DigitalOcean's lingo for a running instance)
- Setup
 - Create non-root user
 - Install Apache
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Starting a new Droplet (1 of 5)



Create Droplet



Droplet hostname

Hostname:



Select Size

512MB / 1 CPU
20GB SSD DISK
1TB TRANSFER

MONTHLY HOURLY
\$5.00 \$0.007

1GB / 1 CPU
30GB SSD DISK
2TB TRANSFER

MONTHLY HOURLY
\$10.00 \$0.015

2GB / 2 CPUS
40GB SSD DISK
3TB TRANSFER

MONTHLY HOURLY
\$20.00 \$0.030

4GB / 2 CPUS
60GB SSD DISK
4TB TRANSFER

MONTHLY HOURLY
\$40.00 \$0.060

8GB / 4 CPUS
80GB SSD DISK
5TB TRANSFER

MONTHLY HOURLY
\$80.00 \$0.119

16GB / 8 CPUS
160GB SSD DISK
6TB TRANSFER

MONTHLY HOURLY
\$160.00 \$0.238



Starting a new Droplet (2 of 5)



Select Region



New York 1



San Francisco 1



New York 2



Select Image

Linux Distributions

Applications

My Images

My Backups

Ubuntu



CentOS



Debian



Arch Linux



Fedora



Debian 7.0 x64

Debian 7.0 x32

Debian 6.0 x64

Debian 6.0 x32

Starting a new Droplet (3 of 5)



Add optional SSH Keys

gordon@rakis



Settings



Enable VirtIO

Create Droplet

Your root password will be emailed to you

Starting a new Droplet (4 of 5)



We are now creating your droplet

Your Droplet is being created!



Approximately 58 seconds remaining

Starting a new Droplet (4 of 5)



We are now creating your droplet

Your Droplet is being created!



Approximately 19 seconds remaining

Starting a new Droplet (4 of 5)



We are now creating your droplet

Your Droplet is being created! ×



Approximately 6 seconds remaining

Starting a new Droplet (5 of 5)



 **Droplets** Create Droplet

Image	Name	IP Address	Status	Memory	Disk	Region
	pyyc	192.241.238.89	Active	512MB	20GB	sfo1

Droplet Control Panel



pyyyc

Console Access

192.241.238.89 Active 512MB Ram 20GB SSD Disk San Francisco 1 Debian 7.0 x64



Power



Access



Resize



Snapshots



Backups



Settings



Graphs



Destroy

Power Cycle

Power Off

This will reboot your droplet.

We recommend rebooting your droplet through the command line, as this action is the same as hard resetting the server and may cause data corruption.

Do you wish to proceed?

Power Cycle

Root Password by Email

Your new Droplet is created! (simrock) - DigitalOcean - Icedove Mail/News

From Digital Ocean Support <support@digitalocean.com> ☆

Subject **Your new Droplet is created! (simrock) - DigitalOcean**

09:01 PM

Yay! Your new Droplet has been created!

You can access it using the following credentials:

IP Address: 192.241.238.89

Username: root

Password: xsytjbxcdadf

INSTRUCTIONS

To login to your droplet, you will need to open a terminal window and copy and paste the following string:

```
ssh root@192.241.238.89
```

Once you've entered the string, copy and paste your password:

```
xsytjbxcdadf
```

If you're having trouble, please visit the following tutorial:

<https://www.digitalocean.com/community/articles/how-to-create-your-first-digitalocean-droplet-virtual-server>

Happy coding,
DigitalOcean

Connect to Droplet

```
$ ssh root@192.241.238.89
```

```
The authenticity of host '192.241.238.89 (192.241.238.89)' can't be established.
```

```
key fingerprint is fc:45:20:9e:21:c3:d2:fa:b5:a0:29:fa:74:34:4f:4a.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '192.241.238.89' (ECDSA) to the list of known hosts.
```

```
root@192.241.238.89's password: [ROOT PASSWORD FROM EMAIL]
```

```
Linux pyyyc 3.2.0-4-amd64 #1 SMP Debian 3.2.41-2+deb7u2 x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
```

```
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

```
root@pyyyc:~#
```


The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Create non-root user

```
root@pyyyc:~# adduser gordon
Adding user `gordon' ...
Adding new group `gordon' (1000) ...
Adding new user `gordon' (1000) with group `gordon' ...
Creating home directory `/home/gordon' ...
Enter new UNIX password: [ENTER PASSWORD]
Retype new UNIX password: [REPEAT PASSWORD]
passwd: password updated successfully
Changing the user information for gordon
[...]
Is the information correct? [Y/n] y
```

```
root@pyyyc:~# adduser gordon sudo
Adding user `gordon' to group `sudo' ...
Adding user gordon to group sudo
Done.
```

```
root@pyyyc:~# exit
Connection to 192.241.238.89 closed.
```

Connect as non-root user

```
$ ssh gordon@192.241.238.89
```

```
gordon@192.241.238.89's password: [ENTER PASSWORD]
```

```
gordon@pyyyc:~$ sudo date
```

```
We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:
```

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for gordon: [ENTER PASSWORD]
```

```
Fri Aug 23 03:52:41 UTC 2013
```

The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - **Install Apache**
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Install Apache Web server

```
gordon@pyyyc: ~$ sudo apt-get update
```

```
Get:1 http://security.debian.org wheezy/updates Release.gpg [836 B]
```

```
[...]
```

```
Fetches 10.1 MB in 18s (550 kB/s)
```

```
Reading package lists... Done
```

```
gordon@pyyyc: ~$ sudo apt-get install apache2 apache2-utils \  
libapache2-mod-python libapache2-mod-wsgi
```

```
Reading package lists... Done
```

```
The following extra packages will be installed:
```

```
apache2-mpm-prefork apache2.2-bin apache2.2-common
```

```
[...]
```

```
Do you want to continue [Y/n]? Y
```

```
[...]
```

```
To activate the new configuration, you need to run:
```

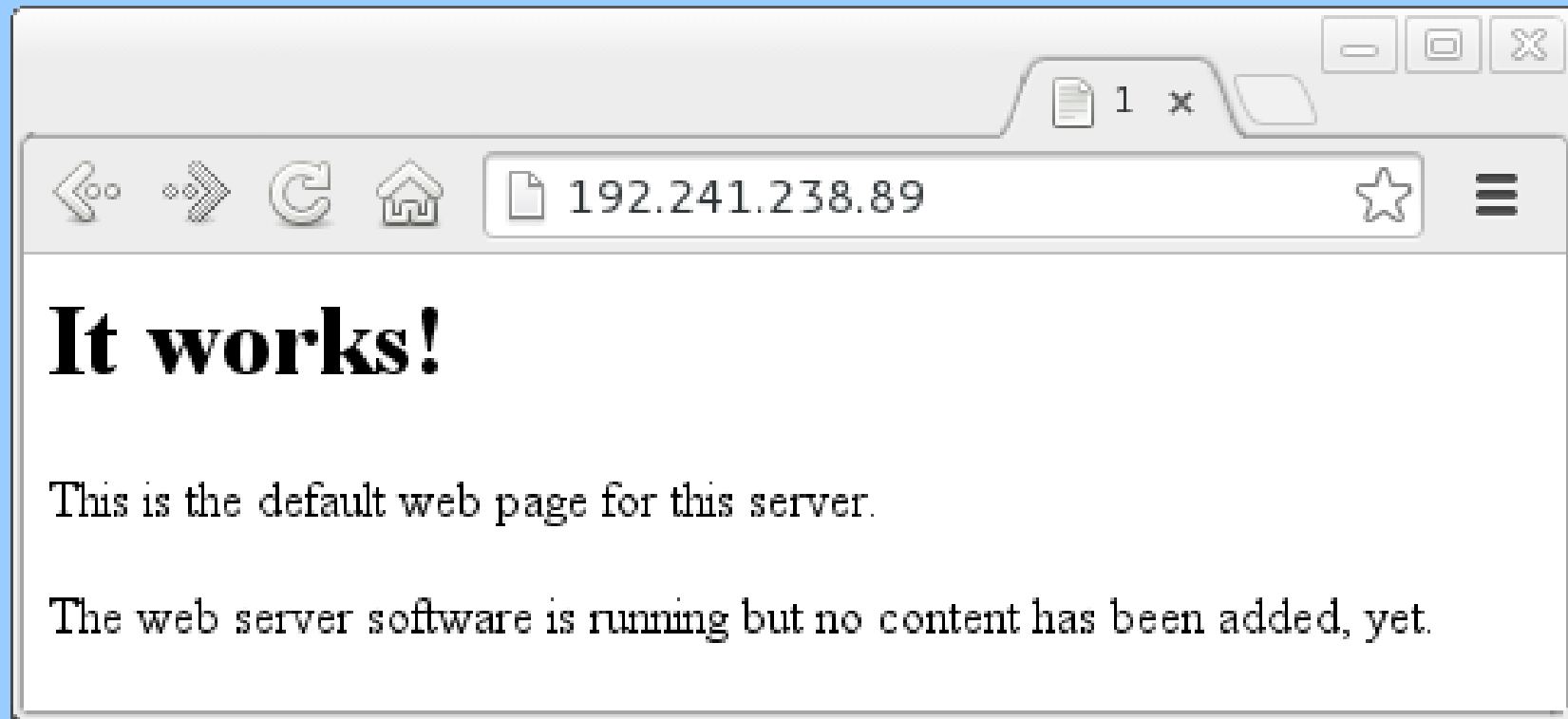
```
service apache2 restart
```

```
gordon@pyyyc: ~$ sudo service apache2 restart
```

```
... waiting apache2: Could not reliably determine the server's fully  
qualified domain name, using 127.0.0.1 for ServerName
```

```
. ok
```

Test Apache Server



```
gordon@pyyyc: ~$ cat /var/www/index.html
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added,
yet.</p>
</body></html>
```

The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - **Install Python dev. Tools**
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Install Python dev. tools

```
gordon@pyyyc: ~$ sudo apt-get install python-dev \  
python-setuptools python-pip
```

```
[...]
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following extra packages will be installed:
```

```
[...]
```

```
Do you want to continue [Y/n]? Y
```

```
[...]
```

```
gordon@pyyyc: ~$ pip --help
```

```
Usage: pip COMMAND [OPTIONS]
```

```
Options:
```

```
--version          show program's version number and exit
```

```
-h, --help         Show help
```

```
[...]
```


The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - Install Python dev. Tools
 - **Install Python's Flask module**
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Install Python Flask Module

```
gordon@pyyyc: ~$ sudo pip install --upgrade flask
[...]
Downloading/unpacking flask
  Downloading Flask-0.10.1.tar.gz (544Kb): 544Kb downloaded
  Running setup.py egg_info for package flask
[...]
Successfully installed flask
Cleaning up...
```

Good

```
gordon@pyyyc: ~$ python -c 'import flask'
gordon@pyyyc: ~$
```

Bad

```
gordon@pyyyc: ~$ python -c 'import flask'
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named flask
```

The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - Install Python dev. Tools
 - Install Python's Flask module
- **Create Minimal Flask application**
 - **Test as Stand-alone**
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Minimal Flask Application

```
# See http://flask.pocoo.org/
```

```
gordon@pyyyc: ~$ cat /home/gordon/webapp1/myapp.py
```

```
from flask import Flask
app = Flask(__name__)
app.debug = True
```

```
@app.route('/')
def hello_world():
    return 'Hello World!'
```

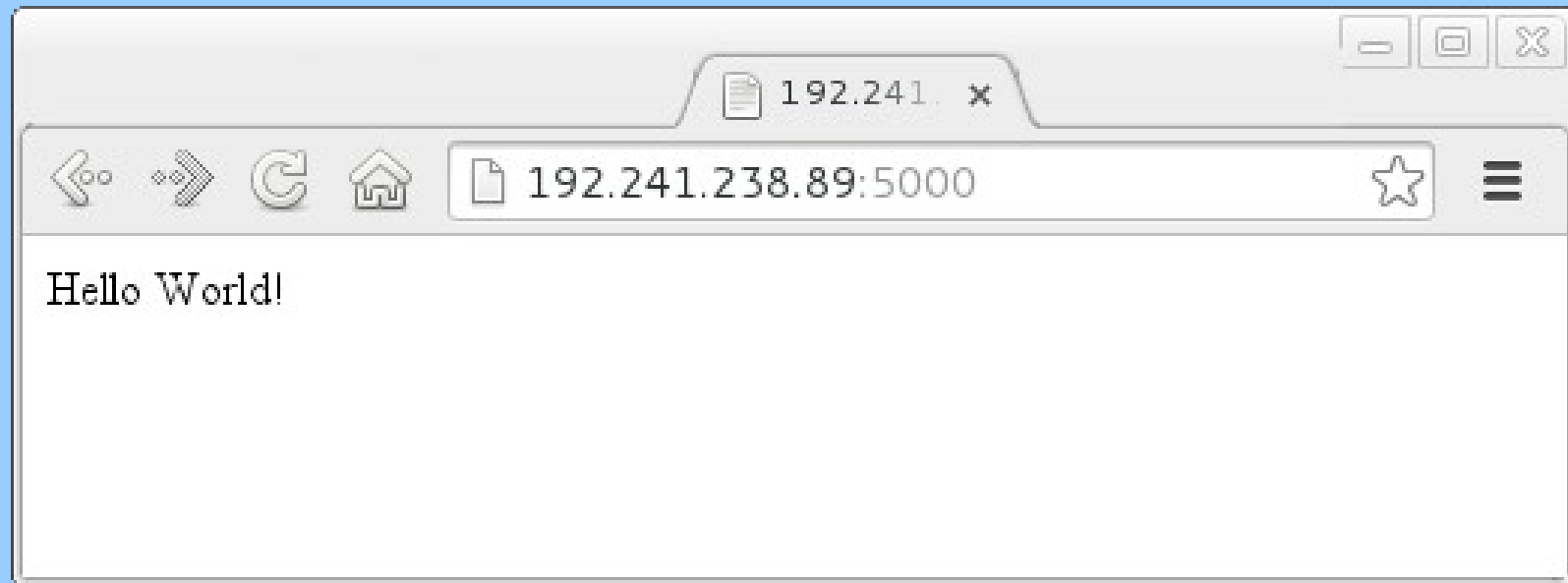
```
if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

```
gordon@pyyyc: ~$ python /home/gordon/webapp1/myapp.py
```

```
* Running on http://0.0.0.0:5000/
```

```
* Restarting with reloader
```

Test Minimal Flask Application



```
gordon@pyyc: ~$ python /home/gordon/webapp1/myapp.py
* Running on http://0.0.0.0:5000/
* Restarting with reloader
184.64.154.197 - - [23/Aug/2013 02:42:31] "GET / HTTP/1.1" 200 -
```

The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - **Configure Apache2 + WSGI**
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Apache + WSGI configuration

- Useful resources
 - WSGI
http://en.wikipedia.org/wiki/Web_Server_Gateway_Interface
 - Flask Deployment (general)
<http://flask.pocoo.org/docs/deploying/>
 - Flask + Apache + WSGI
http://flask.pocoo.org/docs/deploying/mod_wsgi/
 - Apache on Debian overview
<http://www.control-escape.com/web/configuring-apache2-debian.html>
- Relevant Files (in Debian)
 - Sites configuration: `/etc/apache2/sites-available/*`
 - Active sites (soft links): `/etc/apache2/sites-enabled/*`
 - Enable/disable sites (w/ sudo): `a2dissite` , `a2ensite`

Apache2 WSGI config. (1 of 4)

```
# WSGI wrapper script for Flask application
# (you'll need to create this file)

gordon@pyyyc: ~$ cat /home/gordon/webapp1/myapp.wsgi
import sys

sys.path.append('/home/gordon/webapp1')

from myapp import app as application
```


Apache2 WSGI config. (2 of 4)

```
# New Apache configuration file for our web site
# (you'll need to create this file, with sudo)

gordon@pyyyc: ~$ cat /etc/apache2/sites-available/webapp1
<VirtualHost *:80>
    WSGIDaemonProcess test1 user=gordon group=www-data threads=5 \
                                home=/home/gordon/webapp1
    WSGIScriptAlias / /home/gordon/webapp1/myapp.wsgi

    <Directory /home/gordon/webapp1/>
        WSGIProcessGroup test1
        WSGIApplicationGroup %{GLOBAL}
        WSGIScriptReloading On
        Order deny,allow
        Allow from all
    </Directory>
</VirtualHost>
```

Apache2 WSGI config. (3 of 4)

```
# Disable the default site
```

```
gordon@pyyyc: ~$ sudo a2dissite default
```

```
Site default disabled.
```

```
To activate the new configuration, you need to run:
```

```
service apache2 reload
```

```
# Enable our new site
```

```
gordon@pyyyc: ~$ sudo a2ensite webapp1
```

```
Enabling site webapp1.
```

```
To activate the new configuration, you need to run:
```

```
service apache2 reload
```

```
# Restart Apache
```

```
gordon@pyyyc: ~$ sudo service apache2 restart
```

```
[....] Restarting web server: apache2apache2: Could not reliably  
determine the server's fully qualified domain name, using 127.0.0.1  
for ServerName
```

```
... waiting apache2: Could not reliably determine the server's fully  
qualified domain name, using 127.0.0.1 for ServerName
```

```
. ok
```

Apache2 WSGI config. (4 of 4)

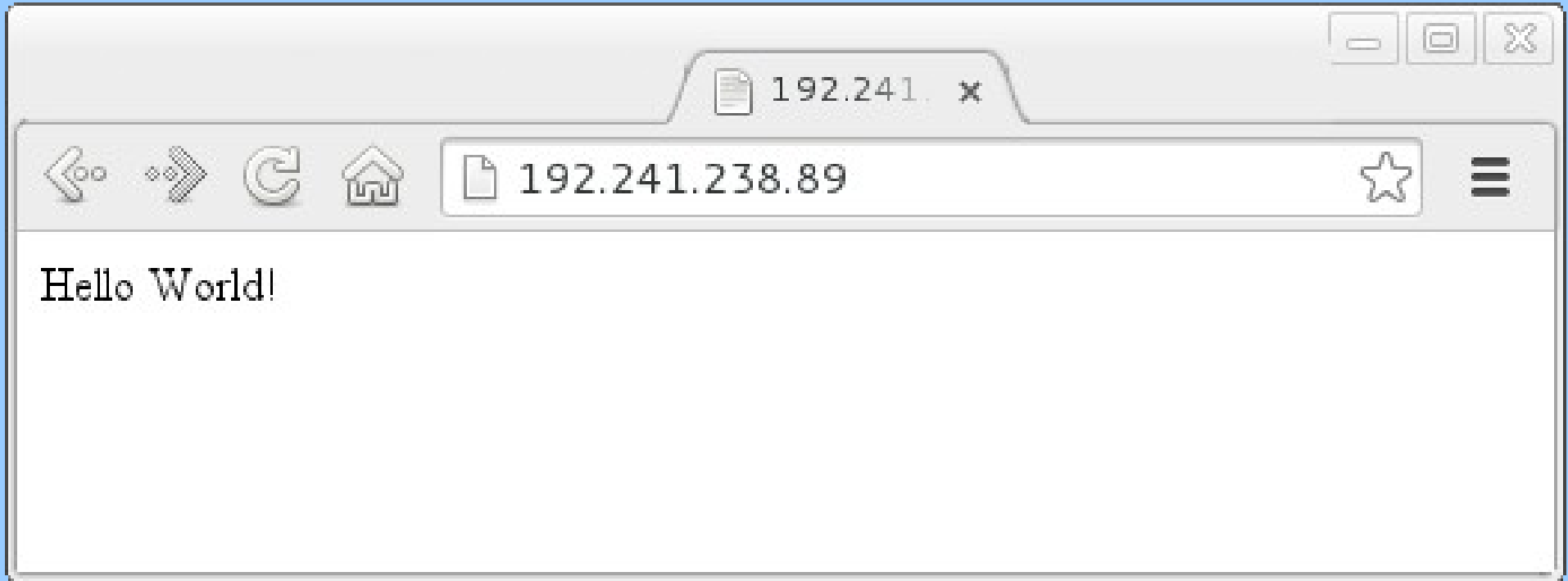
```
# In case of an error, restarting Apache will fail.  
# Error message will appear on the screen ...
```

```
gordon@pyyyc: ~$ sudo service apache2 restart  
Syntax error on line 5 of /etc/apache2/sites-enabled/webapp1:  
Invalid command 'XXXXXX', perhaps misspelled or defined by a module  
not included in the server configuration  
Action 'configtest' failed.  
The Apache error log may have more information.  
Failed!
```

```
# ... or in Apache's error log file:
```

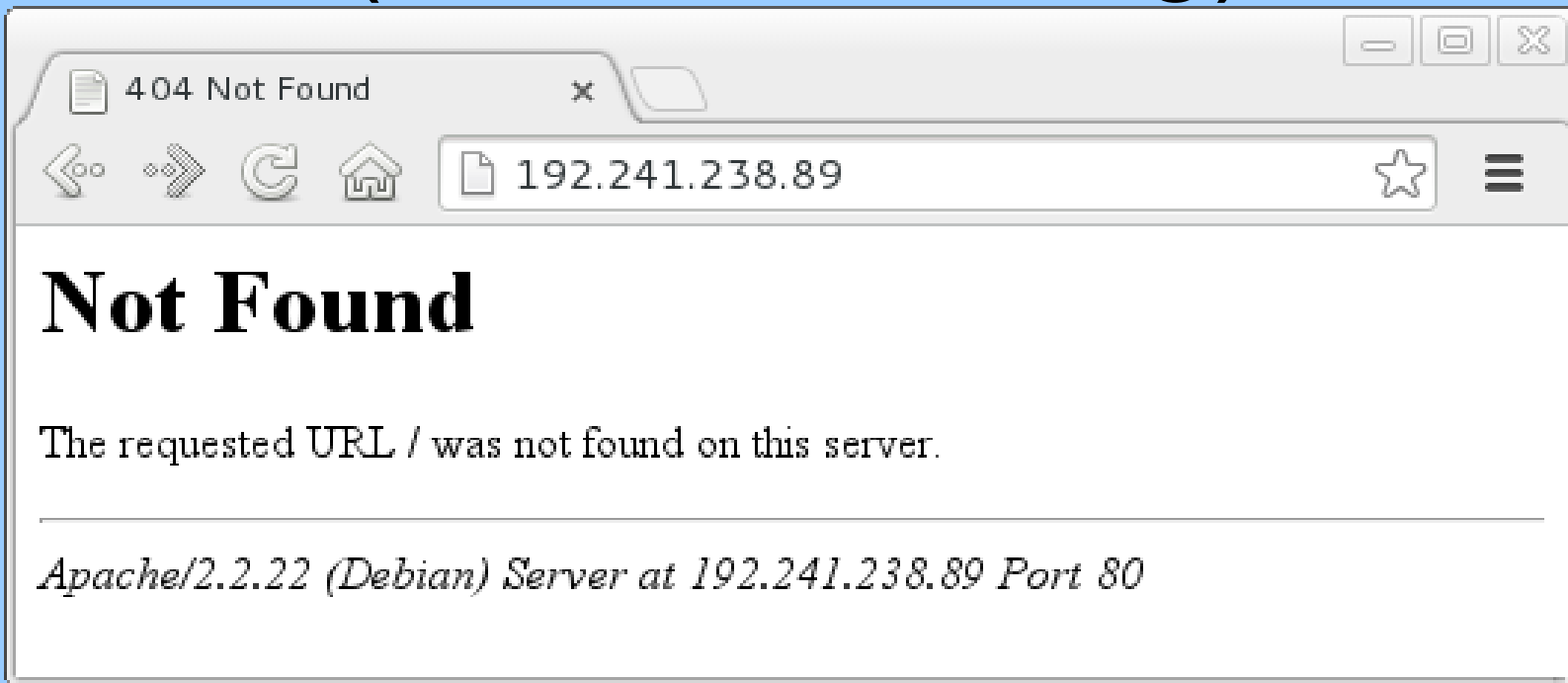
```
gordon@pyyyc: ~$ sudo tail /var/log/apache2/error.log  
[ ... ]
```

Test Apache/WSGI/Flask Application



```
gordon@pyyyc: ~$ sudo tail -f /var/log/apache2/access.log
173.183.194.88 - - [22/Aug/2013:03:47:43 +0000] "GET / HTTP/1.1" 200
483 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/27.0.1453.110 Safari/537.36"
```

Test Apache/WSGI/Flask Application (Troubleshooting)



```
gordon@pyyyc: ~$ sudo tail -f /var/log/apache2/error.log
[Fri Aug 23 03:41:28 2013] [error] [client 173.183.194.88] Target WSGI
script not found or unable to stat: /home/gordon/webapp1/myapp.wsgi
```

The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- **Setup DNS record**
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Setup DNS record (1 of 4)

The screenshot shows the FreeDNS website interface. At the top, the browser address bar displays 'freedns.afraid.org'. The main heading reads 'FreeDNS - Free DNS - Dynamic DNS - Static DNS subdomain and domain hosting'. On the left, there are two navigation menus: 'For Members:' and 'For Everybody:'. The 'For Members:' menu includes links for Main Menu, Domains, Subdomains, Web Forward, Dynamic DNS, IPv6 Reverse, Backup DNS, Preferences, Registry, and Logout. The 'For Everybody:' menu includes links for Home, FAQ, News, DNS Stats, and AUP/TOS. The main content area features a heading 'Free DNS Hosting, Dynamic DNS Hosting, Static DNS Hosting, subdomain and domain hosting.' followed by a paragraph explaining why it's free and a list of 'Possible Uses' with 10 bullet points. On the right side, there is a 'DNS Auth Trace' section with an input field for 'your.domain.com' and a 'Trace' button. Below this is a statistics table showing 1,463,433 members, 2,076 premium users, 5,389,426 records, and 704,344 zones. A promotional message at the bottom right offers '+50 subdomains +3 stealth flags Wildcard DNS Just \$5 a month! Go premium today!'.

FreeDNS - Free DNS - Dynamic DNS - Static DNS subdomain and domain hosting

For Members:

- [[Main Menu](#)]
- [[Domains](#)]
- [[Subdomains](#)]
- [[Web Forward](#)]
- [[Dynamic DNS](#)]
- [[IPv6 Reverse](#)]
- [[Backup DNS](#)]
- [[Preferences](#)]
- [[Registry](#)]
- [[Logout](#)]

For Everybody:

- [[Home](#)]
- [[FAQ](#)]
- [[News](#)]
- [[DNS Stats](#)]
- [[AUP/TOS](#)]

Free DNS Hosting, Dynamic DNS Hosting, Static DNS Hosting, subdomain and domain hosting.

Why is it free? It's quite simple. We wanted a challenge... that's it.

Possible Uses:

- Host your own site on your own connection from home/work/school/etc
- Access your computer with a name (like zeus.afraid.org or yourdomain.com) instead of a numeric IP address
- Run your own http server, ftp server, or anything you want to install on your computer/server
- Fetchable URL to update your IP instantly on our network if you have a dynamic address
- Hosts even work for your LAN. If you have a LAN connected to the internet you can point hosts to private IP addresses (even private IPv6 addresses) and they will work within your network
- Let your friends point theirname.yourdomain.com to their own connection
- Use web forwarding to transparently redirect a hostname to another URL. Let our servers handle the redirection
- afraid.org has been un-interrupted for hundreds of days at a time
- afraid.org is operated from multiple redundant high capacity well connected servers

DNS Auth Trace

Members:	1,463,433
Premium:	2,076
Records:	5,389,426
Zones:	704,344

+50 subdomains
+3 stealth flags
Wildcard DNS
Just \$5 a month!
[Go premium today!](#)

Setup DNS record (2 of 4)

The screenshot shows the 'Subdomains' management page on freedns.afraid.org. The browser address bar displays 'freedns.afraid.org/subdomain/'. The page features a navigation menu on the left, a central area for adding subdomains, and a right-hand sidebar with a 'DNS Auth Trace' tool and account statistics.

For Members:

- [[Main Menu](#)]
- [[Domains](#)]
- [[Subdomains](#)]
- [[Web Forward](#)]
- [[Dynamic DNS](#)]
- [[IPv6 Reverse](#)]
- [[Backup DNS](#)]
- [[Preferences](#)]
- [[Registry](#)]
- [[Logout](#)]

For Everybody:

- [[Home](#)]
- [[FAQ](#)]
- [[News](#)]
- [[DNS Stats](#)]
- [[AUP/TOS](#)]

[Add a subdomain](#)

Subdomains

DNS Auth Trace

Last IP: **174.0.163.213**

UserID:	agn123
Account Type:	Basic
Members:	1,463,436
Premium:	2,076
Records:	5,389,426
Zones:	704,345

+50 subdomains
+3 stealth flags
Wildcard DNS
Just \$5 a month!
[Go premium today!](#)

Setup DNS record (3 of 4)

Add a new subdomain

Type:	A ▾	explanation
Subdomain:	pyyyc	
Domain:	mooo.com (public) ▾	
Destination:	192.241.238.89	
TTL:	<i>For our premium suppo</i> seconds (optional)	
Wildcard:	<input type="checkbox"/> Enabled for all subscribers (more info)	

Save!

Setup DNS record (4 of 4)

1 subdomains			
mooo.com		[add]	
<input type="checkbox"/>	<u>pyyyc.mooo.com</u>	A	192.241.238.89
delete selected		Add	

Testing DNS record

NOTE: DNS record update takes several minutes.

```
$ host pyyyc.moood.com
Unknown host pyyyc.moood.com

$ host pyyyc.moood.com
pyyyc.moood.com has address 127.0.0.2

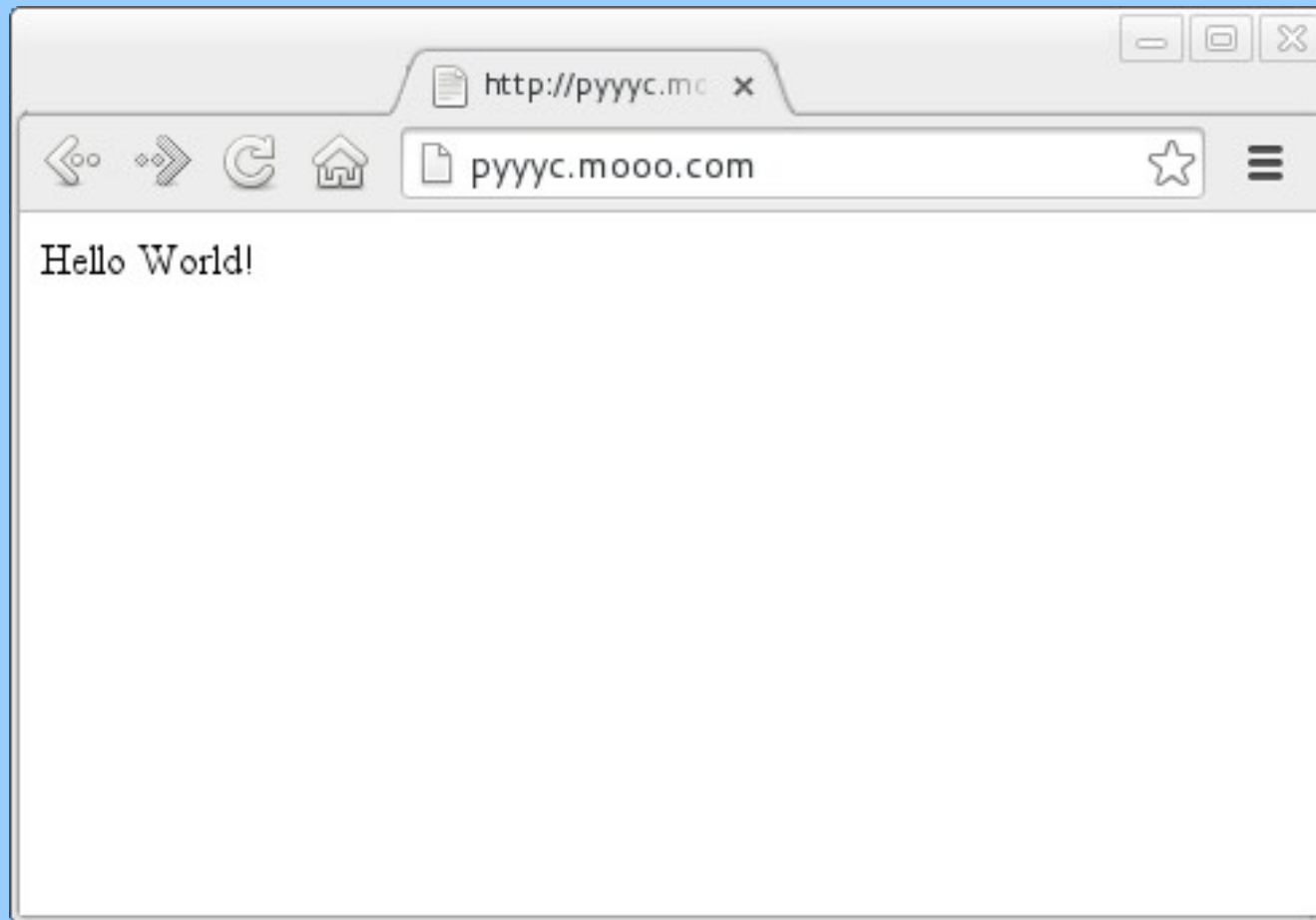
# Wait 5 minutes...

$ host pyyyc.moood.com
pyyyc.moood.com has address 192.241.238.89

$ ping pyyyc.moood.com
PING pyyyc.moood.com (192.241.238.89) 56(84) bytes of data.
64 bytes from 192.241.238.89: icmp_seq=1 ttl=52 time=73.1 ms
64 bytes from 192.241.238.89: icmp_seq=2 ttl=52 time=73.0 ms
...

$ ssh gordon@pyyyc.moood.com
gordon@pyyyc.moood.com's password: [ENTER PASSWORD]
```

Victory!



Automatic DNS updates (1 of 2)

Dynamic DNS Resources

Dynamic DNS Clients

This is the area where you can download one of the dynamic DNS clients which support FreeDNS

Programmer's XML API

For those who write their own update clients, this allows the downloading of account contents dynamic DNS URLs easily within custom applications for easy maintenance.

Rotate Authentication Token

If you suspect someone has gotten unauthorized access to your update URL, this utility will rotate **ALL** of your update URLs for security reasons. Your old update URLs will no longer be valid.

dynamic update candidates! (A records)

mooo.com

[add]

pyyyc.mooo.com

[Direct URL](#)

| [Wget Script](#)

| [Curl Script](#)

| [Edit Record](#)

| [quick cron example](#)

192.241.238.89

Automatic DNS updates (2 of 2)

```
# Edit crontab with "crontab -e"
# Add the listed line, based on FreeDNS's "quick cron example"

# View crontab with "crontab -l"

gordon@pyyyc: ~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# m h dom mon dow    command
3,8,13,18,23,28,33,38,43,48,53,58 * * * * sleep 28 ; wget -O - \
    http://freedns.afraid.org/dynamic/update.php? \
    Q1A0RWFiVmpINWN2b1hNak5UU3pxNHkw0jEwMDEx0TU4 >> \
    /tmp/freedns_pyyyc_mooo_com.log 2>&1 &
```

The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - **Setup Named-Virtual-Hosts, multiple Flask applications**
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Named Virtual Hosts (1 of 5)

Create multiple sub-domains on FreeDNS, pointing to the same IP.

subdomains			
ignorelist.com			[add]
<input type="checkbox"/>	yyc.ignorelist.com	A	192.241.238.89
mooo.com			[add]
<input type="checkbox"/>	pyyyc.mooo.com	A	192.241.238.89
<input type="checkbox"/>	voooodoo.mooo.com	A	192.241.238.89
<input type="button" value="delete selected"/>		Add	

Named Virtual Hosts (2 of 5)

Create multiple Flask applications.
(don't forget to update paths in WSGI files.)

```
gordon@pyyyc: ~$ find -iname "*.py" -o -iname "*.wsgi"
./webapp2/myapp.py
./webapp2/myapp.wsgi
./webapp3/myapp.py
./webapp3/myapp.wsgi
./webapp1/myapp.py
./webapp1/myapp.wsgi

gordon@pyyyc:~$ cat webapp3/myapp.py
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return '<b>Hello From YYC.ignoreList.com</b>(this is webapp3)'
[...]
```

Named Virtual Hosts (3 of 5)

Create Apache sites configuration files with **ServerName** directive:

```
gordon@pyyyc: ~$ cat /etc/apache2/sites-enabled/webapp2
<VirtualHost *:80>
    ServerName    vooodoo.mooo.com

    WSGIDaemonProcess test2 user=gordon group=www-data \
                                threads=5 home=/home/gordon/webapp2
    WSGIScriptAlias / /home/gordon/webapp2/myapp.wsgi

    <Directory /home/gordon/webapp2/>
        WSGIProcessGroup test2
        WSGIApplicationGroup %{GLOBAL}
        WSGIScriptReloading On
        Order deny,allow
        Allow from all
    </Directory>
</VirtualHost>
```

Named Virtual Hosts (4 of 5)

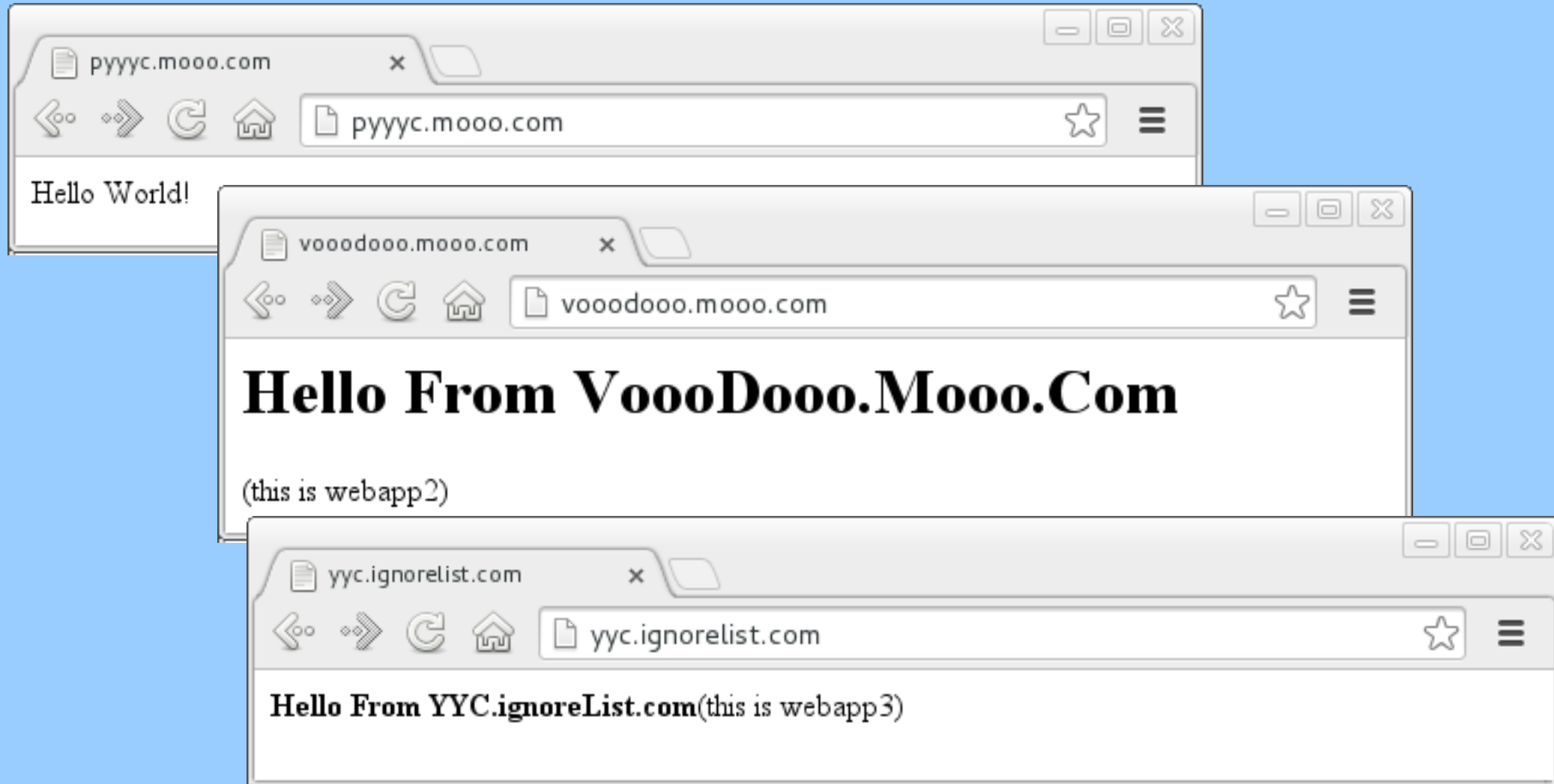
Create multiple Apache sites configuration files.
(Don't forget to update the paths)

```
gordon@pyyyc: ~$ ls -l /etc/apache2/sites-*
/etc/apache2/sites-available:
total 24
-rw-r--r-- 1 692 Mar 3 11:14 default
-rw-r--r-- 1 7251 Mar 3 11:14 default-ssl
-rw-r--r-- 1 486 Aug 23 03:40 webapp1
-rw-r--r-- 1 520 Aug 22 19:56 webapp2
-rw-r--r-- 1 521 Aug 22 19:59 webapp3

/etc/apache2/sites-enabled:
total 0
lrwxrwxrwx 1 26 Aug 22 04:09 webapp1 -> ../sites-available/webapp1
lrwxrwxrwx 1 26 Aug 23 03:24 webapp2 -> ../sites-available/webapp2
lrwxrwxrwx 1 26 Aug 22 19:54 webapp3 -> ../sites-available/webapp3
```

Named Virtual Hosts (4 of 5)

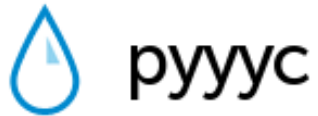
Restart Apache, and test!



The Plan

- Create new Droplet
- Setup
 - Create non-root user
 - Install Apache2
 - Install Python dev. Tools
 - Install Python's Flask module
- Create Minimal Flask application
 - Test as Stand-alone
 - Configure Apache2 + WSGI
- Setup DNS record
- Optional, Time permitting
 - Setup Named-Virtual-Hosts, multiple Flask applications
 - Shutdown droplet, save snapshot
destroy droplet, save money.

Create Snapshot



192.241.238.89 Off 512MB Ram 20GB SSD Disk San Francisco 1 Debian 7.0 x64



Power



Access



Resize



Snapshots



Backups



Settings



Graphs



Destroy

Take a Snapshot

Restore from Snapshot

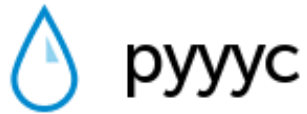
This will power off your droplet.

This may take more than an hour, depending on how much content is on your droplet and how large the disk is.

pyYYC-Demo|

Take Snapshot

Create Snapshot



Your snapshot is processing



192.241.238.89 Off 512MB Ram 20GB SSD Disk San Francisco 1 Debian 7.0 x64



Your event is processing...



Droplet History

Event	Initiated	Execution Time
Snapshot	Processing...	<div><div style="width: 10%;"></div></div>

Snapshots & Backups



Images

Take a Snapshot



Snapshots

pyYYC-Demo

Created: 08/22/13 21:17 in San Francisco 1



Droplet Backups

You have no backups currently

Create Droplet from Snapshot



Create Droplet



Select Image

Linux Distributions

Applications

My Images

My Backups

pyYYC-Demo

Create Droplet

Thank you!



Questions?



Image Credits

Dolphine -

<http://www.flickr.com/photos/nathangibbs/6812478307/>

DigitalOcean, FreeDNS screenshots -
A. Gordon

DigitalOcean Shark Logo -
Digital Ocean.