Holberton Student Server Configuration

V 0.4 20160623

Editors:

William McCann Rona Chong Electra Chong

John Spence Dora Korpar

[0. Open Items](#h.t8sgqvwhw0q5)

[1. Environment](#h.z0ym6kat26oa)

[1.1 Ubuntu](#h.48fqdx7h9nu5)

[2. Installed Software and Configuration](#h.ry5eg22fkyqh)

[2.0 Emacs](#h.wi7fv1vl7ppt)

[2.1 git](#h.970fn3w2gawr)

[2.2.1 Nginx (or Apache web server):](#h.z004wvi76se1)

[2.2.2 Apache(or nginx web server)](#h.2w5iz1vbd1ur)

[2.3 PHP5](#h.k6naiorudf3c)

[2.4 Python](#h.27e0ax5n2ocp)

[2.5 Bind](#h.7x93fq5xp7ea)

[2.6 Nagios (optional)](#h.42u5jp4iv78u)

[2.7 WaveFront Proxy and Collectd Daemon](#h.qespvet9hvmm)

[2.8 mysql](#h.z72svllmig4p)

[2.8 HAProxy](#h.2diqhbsltaj9)

[3.0 Firewall](#h.qphp67kztdnu)

[4. Anticipated Failure Modes](#h.txyr3wwn0j3h)

[5. Redundancy](#h.xdttacxvonnl)

[6. Load Balancing](#h.vvc08mfotl83)

[7. Backup](#h.r2xrytgsht3)

[8. Monitoring](#h.g5kajhnzri5c)

[8. SLA/Uptime Requirements](#h.lyczwk8csoos)

[9. Testing](#h.z7zoai7vuuan)

[10. Appendices](#h.822abkmy1xm9)

# 0. Open Items

To do List:

1. Script to restart services after re-boot
2. Script to rebuild servers if they are corrupted??
3. Ensure everyone has slave bind server
4. Backup and/or secondary host of websites?
5. Group review of plan
6. Mentor Review of this document
7. Test PagerDuty

# 1. Environment

## 1.1 Ubuntu

OS: Ubuntu Version: 14.04.1 LTS

Installed by Sylvain Kalache on OVH

Login: ssh -i [id\_rsa file] admin@[IP address]

Password: Use password for private key

You can also set up a password

Hosted at: [INSERT IP ADDR HERE]

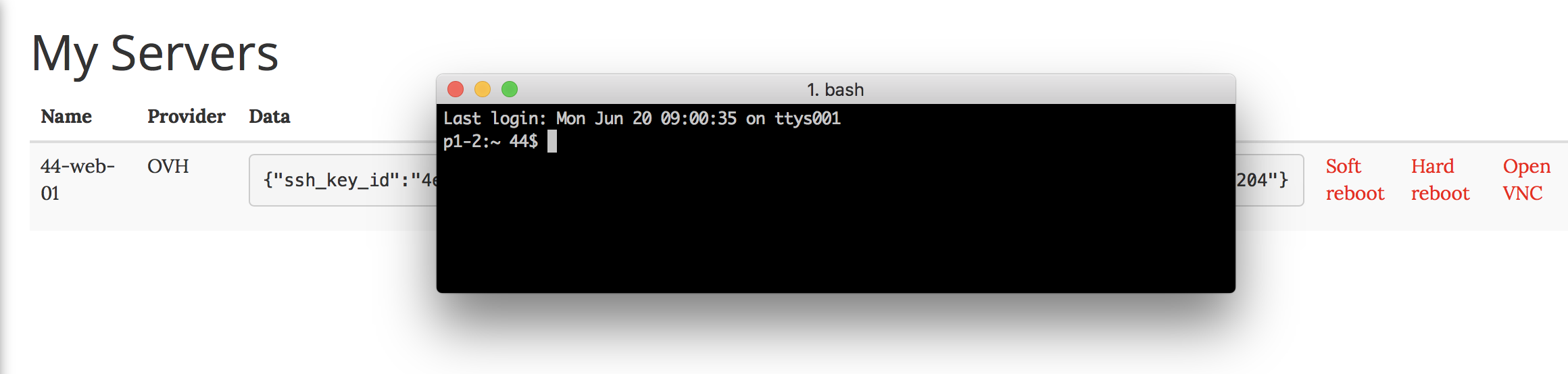
OVH Contact: Sylvain Kalache

OVH Login: Sylvain Kalache

OVH Password: Sylvain Kalache

OVDS SLA: Sylvain Kalache

We have limited ability to manage our servers at https://intranet.hbtn.io/dashboards/my\_server



Soft Reboot: Reboot

Hard Reboot: Virtual “power down” for 15 minutes

Open VNC: Login with username and password. You must set up a password beforehand.

Some uses:

- your server doesn’t have a network interface

- you rejected SSH connections in iptables, such that you can’t connect to your server via SSH

- your server OS doesn’t boot (grub,etc)

# 2. Installed Software and Configuration

## 2.0 Emacs

Version: emacs24

Install using: sudo apt-get install emacs24

## 2.1 git

Install using: sudo apt-get install git

From command line: git config --global user.name <your name here> user.email <your email here>

## 2.2.1 Nginx (or Apache web server):

[John Spence 20160701]

https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-14-04-lts

sudo apt-get update  
sudo apt-get install nginx

To stop your web server, you can type:

sudo service nginx stop

To start the web server when it is stopped, type:

sudo service nginx start

To stop and then start the service again, type:

sudo service nginx restart

We can make sure that our web server will restart automatically when the server is rebooted by typing:

sudo update-rc.d nginx defaults

## 2.2.2 Apache(or nginx web server)

Apache: Version: 2.4.20(?)

Install from: <http://httpd.apache.org/download.cgi>

or simply using: sudo apt-get update

sudo apt-get install apache2

Instructions to follow at: [x](https://httpd.apache.org/docs/current/install.html)

(Secondary instructions: [x](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04))

For Apache: Place website code in /var/www/html

Optionally: follow guidelines @ [x](https://www.digitalocean.com/community/tutorials/how-to-configure-the-apache-web-server-on-an-ubuntu-or-debian-vps) for configuring Apache server

Create this Upstart file: /etc/init/apache.conf (not yet sure if this does anything that the system v file doesn’t already do)

-------------------------

start on runlevel [2345]

stop on runlevel [!2345]

expect daemon

respawn limit 10 30

[more code to get Apache to respawn in case of kill]

## 2.3 PHP5

(RChong 20160617 this is mostly for if any part of your website uses PHP)

Version: PHP5

Install using: sudo apt-get install libapache2-mod-php5;

sudo a2enmod php5

Instructions to follow at: [x](https://help.ubuntu.com/community/ApacheMySQLPHP)

## 2.4 Python

Version: 2.7 and 3.4? (Rona Chong *This may come pre-installed on server, not sure.)*

(Daniel Alzugaray I think it came pre-installed as well)

## 

## 2.5 Bind

Version: bind9

Install using: sudo apt-get update

sudo apt-get install bind9

sudo apt-get install dnsutils

**\*\*\*Also see transcribed PASCAL @GANDI.NET’s presentation on Bind, etc. in Appendices\*\*\***

Edit this file: sudo emacs /var/cache/bind/p/spence.tech

--------------------------

;

; BIND data file for local loopback interface

;

$TTL 300

@ IN SOA localhost. root.localhost. (

8 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS spencetech

@ IN A 158.69.70.204

www CNAME @

spencetech IN A 158.69.70.204

--------------------------

Troubleshooting:

cat /var/log/syslog

sudo service bind9 restart

rndc reload

Example 2

--------------------------

;

; BIND data file for local loopback interface

;

$TTL 604800

@ IN SOA localhost. root.localhost. (

2 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS subdir.johnserrano.tech.

@ IN A 158.69.78.254

www CNAME johnserrano.tech.

subdir IN A 158.69.78.254

--------------------------

Edit this file:

sudo emacs /etc/bind/named.conf.local

--------------------------

//

// Do any local configuration here

//

// Consider adding the 1918 zones here, if they are not used in your

// organization

//include "/etc/bind/zones.rfc1918";

zone "spence.tech" {

type master; // This is a master zonefile

file "/var/cache/bind/p/spence.tech";

allow-query { any; };

};

--------------------------

[William McCann 20160615 note: I had an issue w/ BIND stopping. If you see error something like “rndc: connect failed: 127.0.0.1#953: connection refused” then try the command “sudo service bind9 start”]

[Dora Korpar 20160616 note: I ran into William’s issue and more. This is a great resource for configuring everything for bind9 and named: <http://ubuntuforums.org/showthread.php?t=1543750> ]

From project 168

https://intranet.hbtn.io/projects/168

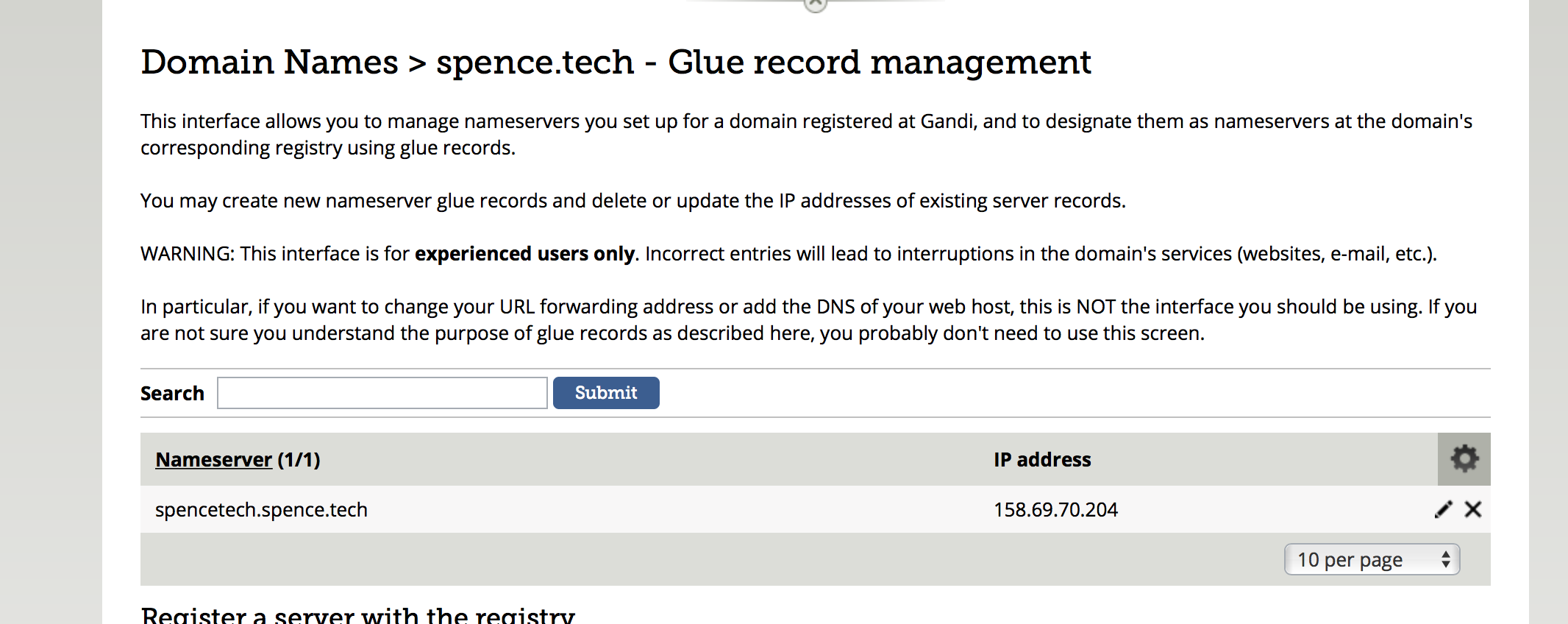
“You want to find one person (or more) willing to host your Gandi domain name zone on their server, which will be a slave zone hosted on their [Bind https://en.wikipedia.org/wiki/BIND server](https://en.wikipedia.org/wiki/BIND).”

Hints:

* The [presentation given](https://rawgit.com/kalou/stuff/master/dns_ws/#/35) <https://rawgit.com/kalou/stuff/master/dns_ws/#/35> during the Keynote contains most of what you need to achieve this.
* Make sure to use the dig command to verify that the slave is well setup
* Remember to have an allow-query { none; }; default option as a generic security measure in the current configuration - that might bite you.”

[John Spence 20160623 We need to add instructions for adding slave zones]

You must modify the glue records at gandi:



## 2.6 Nagios (optional)

## 2.7 WaveFront Proxy and Collectd Daemon

[By John Spence 20160621]

Please see the email from Conor Beverland of Wavefront in, below, concept 39, “On Call” at <https://intranet.hbtn.io/concepts/39> and Project 129, “On Call” at https://intranet.hbtn.io/projects/129.

Wavefront is one the more advanced of several monitoring services that were suggested (Nagios, DataDog, etc.). WaveFront will be used in an upcoming project/session.

Install the Wavefront proxy on your Ubuntu instance. Also install the collectd daemon, which collects data about your system performance communicates that data to the Wavefront proxy which in turn communicates with the Wavefront websites/servers, where you can monitor performance. Separately, under PagerDuty, is a description of how to connect Wavefront to PagerDuty.

Useful commands and troubleshooting

sudo emacs /opt/wavefront/wavefront-proxy/conf/wavefront.conf

sudo /etc/init.d/wavefront-proxy start

sudo /etc/init.d/wavefront-proxy restart

tail /var/log/collectd.log

cat /var/log/collectd.log | grep write\_tsdb  
tail -f /var/log/syslog

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | from: | | **Sylvain Kalache** <sylvain@holbertonschool.com> | | | to: | | SF Winter 2016 Batch <sf-0116@holbertonschool.com> | | | cc: | | Conor Beverland <conor@wavefront.com>,  Jason Mueller <jason.mueller@wavefront.com>,  Guillaume Salva <guillaume@holbertonschool.com> | | | date: | | Mon, Jun 13, 2016 at 8:30 PM | | | subject: | | Wavefront getting started | | | mailing list: | | sf-0116@holbertonschool.com Filter messages from this mailing list | | | signed-by: | | holbertonschool-com.20150623.gappssmtp.com | |   Our friends at [Wavefront](https://twitter.com/wavefronthq) <https://twitter.com/wavefronthq> are offering Holberton School a free account for you folks to monitor your websites and server and can then [integrate it with PagerDuty](https://community.wavefront.com//docs/DOC-1056) <https://community.wavefront.com//docs/DOC-1056> (API integration).  It’s easy to send data to Wavefront. You just need to install the [Wavefront Proxy](https://try.wavefront.com/help/redirect?to=/docs/DOC-1034)  https://try.wavefront.com/help/redirect?to=/docs/DOC-1034 and then stream some data to it. The proxy can be configured to accept data in several data formats including OpenTSDB and Graphite making it easy to integrate with anything which emits data in either of those formats ([Statsd](https://github.com/wavefrontHQ/StatsD), https://github.com/wavefrontHQ/StatsD Collectd, [Diamond](https://try.wavefront.com/help/redirect?to=message/1481) <https://try.wavefront.com/help/redirect?to=message/1481> and many others).  You can collect whatever data you want but, we suggest starting with [Collectd](https://collectd.org/) https://collectd.org to get data from your machines. To do so:   1. Install the proxy itself — This uses an install.sh script maintained here: <https://github.com/wavefrontHQ/install>  *[John Spence 20160621: Don’t use the install script from* [*https://github.com/wavefrontHQ/install*](https://github.com/wavefrontHQ/install) *because it lacks the --next flag and so installs Wavefront 3.14, whereas we need Wavefront 3.15. Use the install script below: sudo bash -c "$(curl -sL https://goo.gl/c70QCx)" -- --proxy --next --server https://try.wavefront.com/api/ --token YOUR\_API\_TOKEN]  You must get your API token from your account at Wavefront.  If you see this: cat: /opt/wavefront/wavefront-proxy/conf/wavefront.conf: No such file or directory You may have installed 3.14  You should receive this message:* The Wavefront Proxy has been successfully installed. To test sending a metric, open telnet to the port 2878 and type my.test.metric 10 into the terminal and hit enter. The metric should appear on Wavefront shortly. Additional configuration can be found at /opt/wavefront/wavefront-proxy/conf/wavefront.conf. A service restart is needed for configuration changes to take effect. admin@44:~$ ]  *[John Spence 20160615 : Cloned this repository to my personal github account and holberton account. Consider downloading files]*     1. Login to Wavefront, click the Gear Icon (top right hand corner), click your email and scroll to the API Access section. Click Generate to create a token.    2. Run the following command on your Linux server: sudo bash -c "$(curl -sL<https://goo.gl/c70QCx>)" -- --proxy --next --server<https://try.wavefront.com/api/> --token YOUR\_API\_TOKEN replacing <YOUR\_API\_TOKEN> with the token you generated in the previous step.   ***[John Spence 20160615: Note that my installation experience was inconsistent with these instructions, possibly because I used interactive mode.] [William McCann 20160615: Nothing would install for me with this command or the non-interactive command. I’m currently in email dialogue with Conor from Wavefront to resolve the issue. I pass on the info when it’s solved)***    3. Edit the file /opt/wavefront/wavefront-proxy/conf/wavefront.conf, uncomment line 20 #prefix=production and replace production with a 3 letter prefix starting with s, E.g.: prefix=s10 where 10 is your 2-digit student number.    4. Restart the proxy by running sudo /etc/init.d/wavefront-proxy restart    5. You should see your proxy (your hostname) listed on [this page](https://try.wavefront.com/manage/agents). <https://try.wavefront.com/manage/agents> Any data that it emits should be prefixed with your student number. *[John Spence 20160621: As of this writing, two students 944 and 45) are using Wavefront with their server:*  ] 2. Install Collectd    1. On the same machine where the Proxy is running, run this command sudo bash -c "$(curl -sL<https://goo.gl/c70QCx>)" -- --collectd --proxy\_address localhost --proxy\_port 4242 --overwrite\_collectd\_config    2. Run the same command on other machines as necessary (replacing localhost with the hostname of the Proxy server).    3. Collectd will be using the default configuration https://github.com/wavefrontHQ/install/blob/master/collectd.conf. (John Spence 20160615 Forked to personal and holberton-students github account) We think it’s a good starting point but feel free to explore the [collectd documentation](https://collectd.org/documentation/manpages/collectd.conf.5.shtml) and modify the configuration as you like. 3. Explore the data, learn more and create an Alert!    1. Navigate to this dashboard: <https://try.wavefront.com/dashboard/collectd-holberton>, select your Student ID from the variable. You should then see your server(s) in the Server dropdown and be able to select and graph them — if you don’t see your Student ID as an option right away then give it a couple of minutes to make sure that Wavefront has finished indexing your data.  *[John Spence 20160619 A snapshop of the current dashboard. As of this writing, WaveFront is working on a bug which may affect this output:*  *]*    2. There’s lots of [documentation on the Community](https://try.wavefront.com/help/redirect) about getting started with Wavefront. One of the most useful documents is the [language reference](https://try.wavefront.com/help/redirect?to=docs/DOC-1011) https://community.wavefront.com/docs/DOC-1011, If you click [“Dashboards → Blank Chart”](https://try.wavefront.com/chart) <https://try.wavefront.com/chart> you can start experimenting with some of your own queries.    3. Take a look at the [Alert which Sylvain](https://try.wavefront.com/alert/1462501986919) <https://try.wavefront.com/alert/1462501986919> has already created. Clone it and modify it for your own student metric prefix. Send the alert to your own PD API key. |

2.7.1 CollectD notes:

<https://collectd.org>  
  
Installation Instructions are in the email above and restated here:

sudo bash -c "$(curl -sL https://goo.gl/c70QCx)" -- --collectd --proxy\_address localhost --proxy\_port 4242 --overwrite\_collectd\_config

Upon successful installation you should see:

SUCCESS

CollectD has been successfully installed and configured. Additional configurations can be found at /etc/collectd/managed\_config/. Check /var/log/collectd.log for errors regarding writing metrics to the Wavefront Proxy by grepping for write\_tsdb

admin@44:~$

The Collectd/Wavefront installation is configured by default to communicate with the localhost. The idea is that you install the Wavefront proxy once and then you could send data to it from multiple different machines. Say you install the proxy on a machine with hostname: [server1.company.com](http://server1.company.com/) , then if collectd on that machine points at “localhost” it will work.

However, if you then go and install collectd on a second machine, then the proxy is no longer listening on “localhost”, it’s listening on [server1.company.com](http://server1.company.com/). If that’s the case all you would need to do is:

edit /etc/collectd/managed\_config/10-wavefront.conf and switch: Host "localhost" from localhost to the hostname where the proxy is running.

## 2.8 mysql

[John Spence 20160708]

Instructions here:

<https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-14-04>

sudo apt-get update

sudo apt-get install mysql-server  
 sudo mysql\_secure\_installation  
 sudo mysql\_install\_db

To start mysqld at boot time you have to copy

support-files/mysql.server to the right place for your system

You can start the MySQL daemon with:

cd /usr ; /usr/bin/mysqld\_safe &

You can test the MySQL daemon with mysql-test-run.pl

cd /usr/mysql-test ; perl mysql-test-run.pl

Checking the status of MySQL server by running 'service mysql status', I get

the following: "mysql start/running, process 1705" (your process number is

likely different).

Now, update the mysql config: `sudo emacs /etc/mysql/my.cnf`

Comment out the line "bind-address = 127.0.0.1"

restart mysql: `service mysql restart`

Now update iptable configuration:

`iptables -L --line-numbers`

Insert the new rule at the beginning for each IP address (104.7.14.91, 199.116.74.197, 199.116.74.198, 54.172.4.200):

`iptables -I INPUT 2 -s 199.116.74.197 -p tcp --dport 3306 -j ACCEPT`

login to mysql with user root: `mysql -u root -p`

enter your root password.

## 2.8 HAProxy

[John Spence 20160701]

HAProxy is installed on your load-balancing ubuntu server.

Good installation explanation here:

https://www.howtoforge.com/tutorial/ubuntu-load-balancer-haproxy/

# 3.0 Firewall

[Revised John Spence 20160711]

Student’s best links for iptables:

<https://help.ubuntu.com/community/IptablesHowTo>

<https://help.ubuntu.com/community/IptablesHowTo#Editing_iptables>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-iptables-on-ubuntu-14-04>

<http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html#toc7>

<https://www.linode.com/docs/security/firewalls/control-network-traffic-with-iptables>

<http://superuser.com/questions/769814/how-to-block-all-ports-except-80-443-with-iptables>

<https://www.dnsknowledge.com/q-a/what-ports-need-to-be-open-for-dns-operate/>

<https://www.centos.org/forums/viewtopic.php?t=3974>

The program “iptables” comes with your server. It uses a table to keep track of what ports are open and which are blocked.

Use the command “sudo iptables -L” to see that table. It will probably start out looking like this:

Chain INPUT (policy ACCEPT)  
target prot opt source destination  
  
Chain FORWARD (policy ACCEPT)  
target prot opt source destination  
  
Chain OUTPUT (policy ACCEPT)  
target prot opt source destination

[RChong 20160620]:

To see active & open ports on server prior to configuring iptables use:

‘sudo netstat -anltp’

111 (used by rpcbind, a utility related to a filesharing service that’s unnecessary for us)

53 (used for DNS by named/bind)

22 (used for SSH)

953 (used for rndc by named/bind, on loopback address)

80 (used by httpd/your web server)

You first have to tell the table which connections you want to keep open, and then give it a “close the rest” command. Do this in the right order, because if you close everything first you’ll close the port you use to send command to your server and be locked out.

Per the spec:

“The golden rule is to close all ports that don't need to be opened.

In your case you are hosting a web application, only 3 ports have to be open on to the Internet:

* HTTP 80
* HTTPS 443 (for SSL)
* SSH 22
* DNS 53
* Be careful not to lock you out of the server (by blocking port SSH 22)

[John Spence 20160711 The following sequence will enable iptables to work with WaveFront and meet other minimum requirements, except for mysql (see that section) and, other services I am not using like uptimeRobot, etc.]

sudo su

root@44:/home/admin# iptables -F

root@44:/home/admin# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT^C

root@44:/home/admin# iptables -A INPUT -p udp -m udp --dport 53 -j ACCEPT

root@44:/home/admin# iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT

root@44:/home/admin# iptables -A INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT

root@44:/home/admin#

root@44:/home/admin# iptables -A INPUT -p tcp -m tcp -m multiport --dports 22,53,80,443,4242,2878 -j ACCEPT

root@44:/home/admin# iptables -A INPUT -p tcp -m tcp -m multiport --dports 22,53,80,443,4242,2878 -j ACCEPT

root@44:/home/admin# iptables -A INPUT -m conntrack -j ACCEPT --ctstate RELATED,ESTABLISHED

root@44:/home/admin# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

root@44:/home/admin#

root@44:/home/admin# iptables -A INPUT -j DROP

Chain OUTPUT (policy ACCEPT)

target prot opt source destination root@44:/home/admin#

root@44:/home/admin# iptables -L

Chain INPUT (policy ACCEPT)

target prot opt source destination

ACCEPT udp -- anywhere anywhere udp dpt:domain

ACCEPT tcp -- anywhere anywhere state NEW tcp dpt:domain

ACCEPT udp -- anywhere anywhere state NEW udp dpt:domain

ACCEPT tcp -- anywhere anywhere tcp multiport dports ssh,domain,http,https,4242,2878

ACCEPT tcp -- anywhere anywhere tcp multiport dports ssh,domain,http,https,4242,2878

ACCEPT all -- anywhere anywhere ctstate RELATED,ESTABLISHED

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

DROP all -- anywhere anywhere

Chain FORWARD (policy ACCEPT)

target prot opt source destination

To Test specific ports:

To check which ports are open or closed, follow the directions in the spec. That means on your server running the “nc -l ###” command (where the ### is the number of the port you want to test). Then on another computer running the “telnet xxx.xxx.xxx.xxx ###” command (where the xxx.xxx.xxx.xxx is your server’s IP address and the ### is the port you want to test). If the port is open, you should be able to type things on the telnet computer and hit enter and have them appear on the server.

To Make persistent after reboot:

One set of commands to make iptables persistent after reboots:

Write your rules to a files with: sudo iptables-save > /etc/iptables.conf

And then set them to automatically reload every time with: sudo iptables-restore < /etc/iptables.conf

[John Spence 20160711 Ubuntu has a built in firewall \*service\*, ufw, that is associated with iptables (not a service, but rather a kernal setting) that seems easier to administer. I have not tested this.

3.6 mySql Configuration

(John Spence 20160615: We can use the client application on the mac to access a database on another machine (vs the server application, the installation instructions for which are in the installation section above and the configuration for which are just below)

Notes provided after install:

---------------------------------------------------------------------------------------------------------------

To start mysqld at boot time you have to copy

support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !

To do so, start the server, then issue the following commands:

/usr/bin/mysqladmin -u root password 'new-password'

/usr/bin/mysqladmin -u root -h vagrant-ubuntu-trusty-64 password 'new-password'

Alternatively you can run:

/usr/bin/mysql\_secure\_installation

which will also give you the option of removing the test

databases and anonymous user created by default. This is

strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:

cd /usr ; /usr/bin/mysqld\_safe &

You can test the MySQL daemon with mysql-test-run.pl

cd /usr/mysql-test ; perl mysql-test-run.pl

One configuration example, not necessarily correct, but to illustrate the options:

---------------------------------------------------------------------------------------------------------------

In order to log into MySQL to secure it, we'll need the current

password for the root user. If you've just installed MySQL, and

you haven't set the root password yet, the password will be blank,

so you should just press enter here.

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL

root user without the proper authorisation.

Set root password? [Y/n] n

... skipping.

By default, a MySQL installation has an anonymous user, allowing anyone

to log into MySQL without having to have a user account created for

them. This is intended only for testing, and to make the installation

go a bit smoother. You should remove them before moving into a

production environment.

Remove anonymous users? [Y/n] n

... skipping.

Normally, root should only be allowed to connect from 'localhost'. This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y

... Success!

By default, MySQL comes with a database named 'test' that anyone can

access. This is also intended only for testing, and should be removed

before moving into a production environment.

Remove test database and access to it? [Y/n] Y

- Dropping test database...

ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist

... Failed! Not critical, keep moving...

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far

will take effect immediately.

Reload privilege tables now? [Y/n] Y

---------------------------------------------------------------------------------------------------------------

To allow a specific IP to access mySQL through its port:

sudo iptables -I INPUT -p tcp -s 104.7.14.91 --dport 3306 -j ACCEPT

3.6 Startup scripts

(John Spence 20160616) Some services and configurations may need to be run automatically upon warm or cold reboot. This list needs to be completed:

-Iptables

-?

-?

There are several options for startup scripts.

Two of them: crontab -e with @reboot

.conf file in init.d

Here is a link:

http://stackoverflow.com/questions/12973777/how-to-run-a-shell-script-at-startup

# 4. Anticipated Failure Modes

[John Spence 20160701]

Needs to be completed

# 5. Redundancy

[John Spence 20160701]

Summary:

* Second web server load balanced through HAProxy
* DNS Server installed on two other student’s ubuntu instances

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Engineer | ID | Server IP | PagerDuty ID | BIND, etc Redundancy 1 | BIND, etc Redundancy 2 |  |
| Damian Ali | 16 |  | P2QQWO7 | Alexandro de Oliveira | Asaia Palacios |  |
| Alexandro de Oliveira | 23 |  | PJJZ7UN | Asaia Palacios | augustin "gus" francis-boeuf |  |
| Asaia Palacios | 17 |  | PE10V6B | augustin "gus" francis-boeuf | Bennett Buchanan |  |
| augustin "gus" francis-boeuf | 19 |  | P5O6QLV | Bennett Buchanan | Bilal Khan |  |
| Bennett Buchanan | 16 |  | P0M7XRN | William McCann | Chandler Mayo |  |
| Bilal Khan | 40 |  | P67TDSA | Chandler Mayo | Daniel Alzugaray |  |
| Chandler Mayo | 33 |  | PEKNHAG | Daniel Alzugaray | Dora Korpar |  |
| Daniel Alzugaray | 49 |  | PREKU5B | Dora Korpar | Electra Chong |  |
| Dora Korpar | 28 |  | P0OI151 | Electra Chong | Gloria Bwandungi Mugarura |  |
| Electra Chong | 18 |  | PT63T2Y | Gloria Bwandungi Mugarura | Ian Wagener |  |
| Gloria Bwandungi Mugarura | 30 |  | PR4XKJS | Ian Wagener | John Serrano |  |
| Ian Wagener | 15 |  | PQ8NYSN | John Serrano | John Spence |  |
| John Serrano | 24 |  | PZ1C5Q7 | John Spence | Joe Greene |  |
| John Spence | 44 |  | PFXAQK2 | Joe Greene | Josh Parkin |  |
| Joe Greene | 43 |  | P3K0S1Y | Josh Parkin | Josquin "Jo" Gaillard |  |
| Josh Parkin | 14 |  | PB37AWQ | Josquin "Jo" Gaillard | Kris Bredemeier |  |
| Josquin "Jo" Gaillard | 16 |  | PV7YL8E | Kris Bredemeier | Zee Adams |  |
| Kris Bredemeier | 34 |  | PU2QFWD | William McCann | Marine Dejean |  |
| Zee Adams | 29 |  | PBMV00V | Marine Dejean | William McCann |  |
| Marine Dejean | 22 |  | P88M8YG | Zee Adams | Praylin Dinamoni |  |
| William McCann | 45 |  | PIVT8XQ | Bennett Buchanan | Kris Bredemeier |  |
| Praylin Dinamoni | 36 |  | PHI3CO4 | Rick Harris | Rona Chong |  |
| Rick Harris | 38 |  | PW3JNCJ | Rona Chong | Sravanthi Sinha |  |
| Rona Chong | 20 |  | PGLGPGM | Sravanthi Sinha | Steven Garcia |  |
| Sravanthi Sinha | 37 |  | PX06A71 | Steven Garcia | Tasneem Fara |  |
| Steven Garcia | 27 |  | PO6F394 | Tasneem Fara | Damian Ali |  |
| Tasneem Fara | 39 |  | PXAW6TN | Damian Ali | Alexandro de Oliveira |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# 6. Load Balancing

[John Spence 20160701]

Summary: HAProxy or Nginx(load balancing component) installed and configured on separate load balancing server with 9/1 load split on web server 1 and web server 2

# 7. Backup

Backup of websites

# 8. Monitoring

[By Electra Chong 20160621]

PagerDuty and the various monitoring tools are described here (read to end): <https://intranet.hbtn.io/concepts/39>

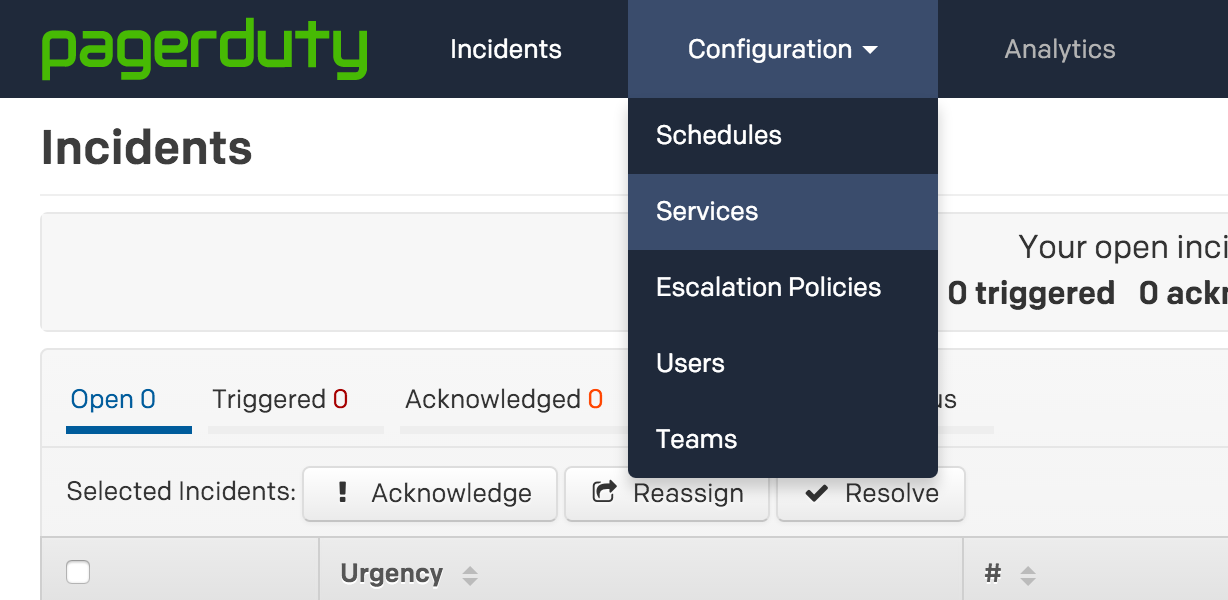
You should look for an email in your Holberton inbox by the subject of ‘[PagerDuty] Holberton Has Invited You to PagerDuty’ and accept the invite to register for PagerDuty.

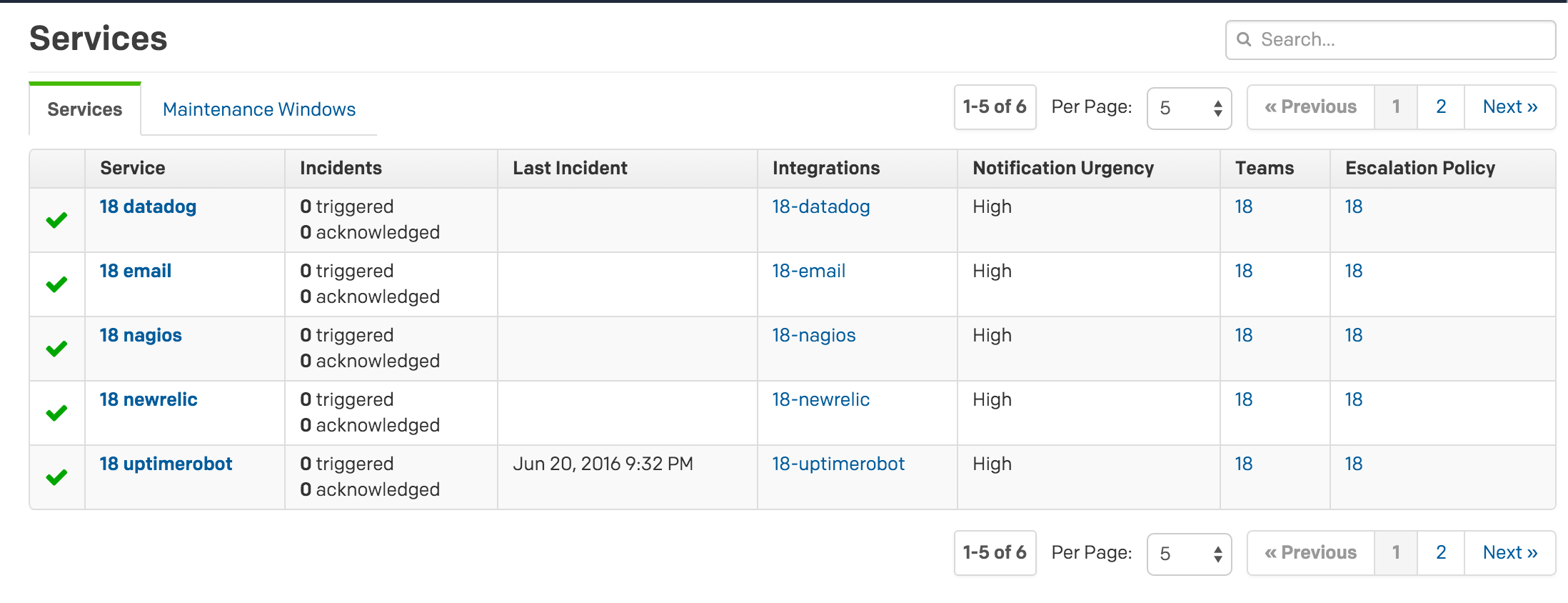
After signing in, you may see a box to the right called Resources on the Incidents page (click on pagerduty logo or Incidents in menu). The first link under Getting Started called ‘Setting up PagerDuty’ describes some basic features you may want to know, as well as how to set up services (section 6). **You need to set up your contact information to get calls/texts when PagerDuty creates alerts.**

<https://support.pagerduty.com/hc/en-us/articles/202828690-PagerDuty-Quick-Start-Guide-Core-Concepts>

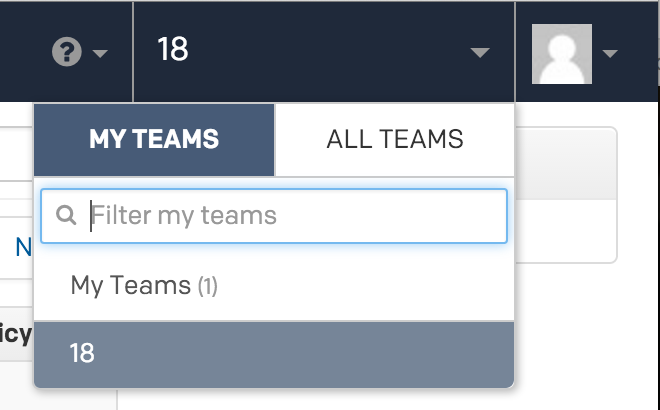
We don’t have to set up services because Sylvain already did this for the recommended tools, per the concept page, but it will show you where to check the services and [find the integration guides](https://www.pagerduty.com/integrations/).

Finding pre-configured services:





If you don’t see the right user # by the service name, make sure your account has been selected from the team in the menu.



Integration guides:

**New Relic:** (\* Per Sylvain we won’t be able to use this yet, from an email sent 06.06.16 by subject - On call: “NewRelic integration is not working with the type of account that you have, the new version of "Alerts" on NewRelic will work, you need to apply to be on the waiting list, here <https://alerts.newrelic.com/accounts/try-alerts>”) <https://www.pagerduty.com/docs/guides/new-relic-integration-guide/>

**Nagios:** <https://www.pagerduty.com/docs/guides/nagios-integration-guide/>

**Datadog:** \* Note I have not found a link to register for a free account yet. <https://www.pagerduty.com/docs/guides/datadog-integration-guide/>

For **Uptime Robot**, I was not able to find an integration guide so eventually I registered for a free Uptime Robot and set the notification contact to be the PagerDuty integration key when you click on the 18-uptimerobot link listed under the ‘Integrations’ column from the Services page (looks like an email, e.g. [18-uptimerobot@holbertonschool.pagerduty.com](mailto:18-uptimerobot@holbertonschool.pagerduty.com)). This triggered a PagerDuty alert as an email came in to the PagerDuty account, and you can verify the email address from the alert which displays the email contents.

I also was not able to find a **Wavefront** integration guide -- instead follow the instructions listed in the Wavefront section which were sent by the people there in an email forwarded by Sylvain.

# 8. SLA/Uptime Requirements

Uptime Requirements  
[by John Spence 20160621]

From <https://intranet.hbtn.io/projects/129>  
From June 6 to June 27, our uptime (serving a web page and not returning an HTTP 200) requirements are as follows:

Mandatory: 95% for the period to come which means that your website is down for 1 day 12 minutes or less  
For Fun: 98%: Uptime:10 hours 4 minutes 48 seconds or less downtime  
Infernum: 99%: 4 hours 2 minutes 24 seconds

[by Electra Chong 20160621]

For this project you should make sure you have implemented the **monitoring services** described in the above section. The project (link above) also gives a hint: **“Hint for this project: make sure that when your server gets restarted, your website automatically comes back up and running. (in your case that should simply mean that you webserver automatically starts)”**

To test this, you can reboot your server by “sudo reboot” and check if your website goes down, and is back up once you can ssh back into your server. (You will automatically be logged out and it will take a minute or two before the server is responsive again). You may want to have commands handy to [start apache](http://www.cyberciti.biz/faq/ubuntu-linux-start-restart-stop-apache-web-server/) or nginx back up manually if it doesn’t automatically restart (Google: “how to restart apache/nginx ubuntu 14.04”).

Most services automatically restart in case of server reboot, you just want to confirm in this case. If hypothetically they didn’t you could look up into start-up scripts using bash, cron, Upstart or System V.

Google hits for “how to start a service automatically ubuntu” and “restart apache service automatically”:

<http://stackoverflow.com/questions/2168518/bash-script-to-restart-apache-automatically>

<http://stackoverflow.com/questions/28498400/restart-apache-service-automatically-using-cron-12am-daily>

<https://www.digitalocean.com/community/tutorials/how-to-use-a-simple-bash-script-to-restart-server-programs>

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-linux-service-to-start-automatically-after-a-crash-or-reboot-part-1-practical-examples>

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-linux-service-to-start-automatically-after-a-crash-or-reboot-part-2-reference>

Whether or not they restart in case of the service crashing is another issue, and the digital ocean guides above describe how to test using the kill command. Electra: I tried looking into creating an Upstart script to restart Apache in case of it crashing, but it was no luck -- I couldn’t find the right command to include in the script, but it looks complicated because you may need to export some information about users and groups for the apache conf file. In any case, the project hint only mentions in case of server restart, so covering this scenario does not seem required (at this point in time).

# 9. Testing

Tests we have run and results

# 10. Appendices

10.1 # DNS workshop

#### Pascal @gandi.net

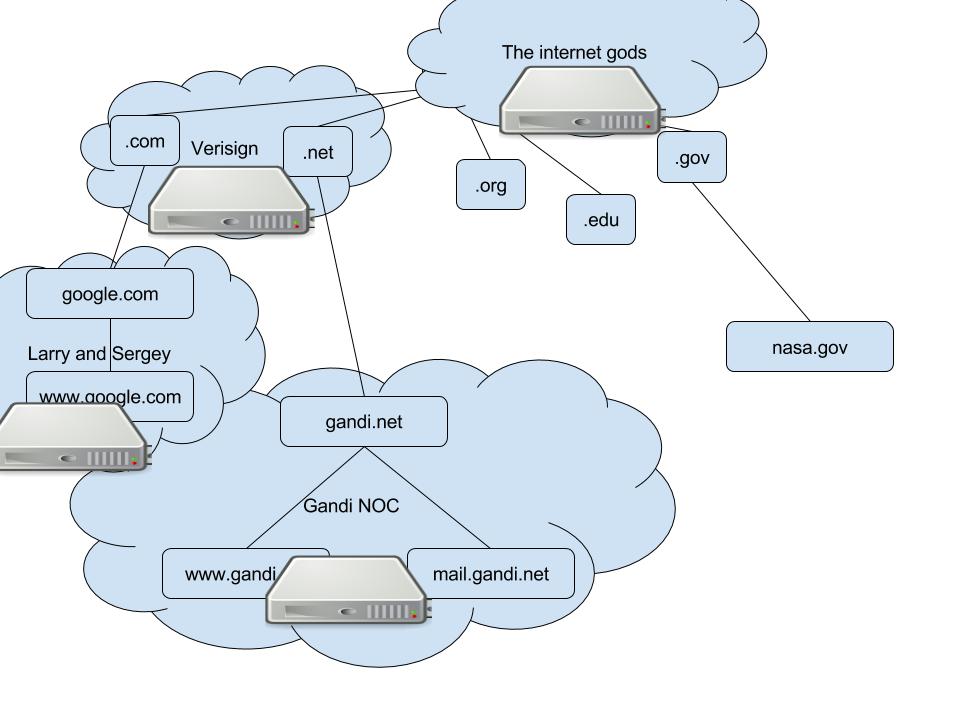
# The domain name system defines:

\* How to name all the things

\* With no conflicts

\* With delegation and separated management

\* A \*protocol\* to query for \*information\* associated with a \*name\*



The resolver asking the recursive caching nameserver

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

The resolver getting a reply from the recursive NS

/----------\ /-------------\

| Client | <= A "www.domain.com" 1.2.3.4 <== | Campus |

| Computer | | Name Server |

\----------/ \-------------/

But how does the recursive NS find this information ?

You, typing "www.domain.com":

You, typing "www.domain.com":

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

(You, waiting)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

|

? www.domain.com

v

/-------------\

| . (root) |

| Name Server |

\-------------/

(You, still waiting)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

^

DUNNO ASK TO .COM

|

/-------------\

| . (root) |

| Name Server |

\-------------/

(You, still waiting)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

|

kthxbye :/

v

/-------------\

| . (root) |

| Name Server |

\-------------/

(You, still waiting)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ =============== \-------------/

|

www.domain.com A ?

v

/-------------\ /-------------\

| .com (vsn) | | . (root) |

| Name Server | | Name Server |

\-------------/ \-------------/

(You, still waiting)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ ==============> \-------------/

^

DUNNO ASK DOMAIN.COM

|

/-------------\ /-------------\

| .com (vsn) | | . (root) |

| Name Server | | Name Server |

\-------------/ \-------------/

(You, still experiencing microsecond boredom)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ ================================= \-------------/

|

www.domain.com A ?

v

/-------------\ /-------------\ /-------------\

| domain.com | | .com (vsn) | | . (root) |

| Name Server | | Name Server | | Name Server |

\-------------/ \-------------/ \-------------/

(You, still experiencing microsecond boredom)

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ ================================= \-------------/

^

SURE THING, use 74.125.224.9, TTL is 300 seconds

|

/-------------\ /-------------\ /-------------\

| domain.com | | .com (vsn) | | . (root) |

| Name Server | | Name Server | | Name Server |

\-------------/ \-------------/ \-------------/

(That was quick, a few milliseconds)

/----------\ /-------------\

| Client | <= "www.domain.com" has 74.125.224. | Campus |

| Computer | | Name Server |

\----------/ \-------------/

SURE THING, use 74.125.224.9, TTL..

/-------------\ /-------------\ /-------------\

| domain.com | | .com (vsn) | | . (root) |

| Name Server | | Name Server | | Name Server |

\-------------/ \-------------/ \-------------/

Now what if you ask the same question again a few seconds after ?

/----------\ /-------------\

| Client | => Q? "www.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

The recursive NS cached this for TTL seconds, immediate answer

/----------\ /-------------\

| Client | <= A "www.domain.com" 1.2.3.4 <== | Campus |

| Computer | (294 seconds left) | Name Server |

\----------/ \-------------/

And what if you ask for dev.domain.com minutes later ?

/----------\ /-------------\

| Client | => Q? "dev.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ \-------------/

/-------------\ /-------------\ /-------------\

| domain.com | | .com (vsn) | | . (root) |

| Name Server | | Name Server | | Name Server |

\-------------/ \-------------/ \-------------/

Name server cached the information from . and .com about domain.com

/----------\ /-------------\

| Client | => Q? "dev.domain.com" Type=A ==> | Campus |

| Computer | | Name Server |

\----------/ ================================= \-------------/

|

dev.domain.com A ?

v

/-------------\ /-------------\ /-------------\

| domain.com | | .com (vsn) | | . (root) |

| Name Server | | Name Server | | Name Server |

\-------------/ \-------------/ \-------------/

</code></pre>

</section>

## Program

- Install a bind nameserver

- Configure an authoritative NS

- Change the config, basic rndc usage

- Add an authoritative zone on the NS

- Add a bit of security

- Handle the glue record

- Setup a slave zone

- SOA overview wrt Master/Slave setup

### Install dig on your server

------------------------------

sh# apt-get update

sh# apt-get install bind9

sh# apt-get install dnsutils

------------------------------

http://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/Bv9ARM.html

### We're setting up an authoritative-only nameserver.

Disallow recursion but allow queries from all clients:

Do the change in /etc/bind/named.conf.options:[Spence 20160616 Note that this conflicts with my example above, where we were told to edit /etc/bind/named.local]

------------------------------

options {

allow-query-cache { none; };

allow-query { any; };

recursion no;

}

------------------------------

### Reconfigure bind, check for errors in the logs

------------------------------

sh# service bind9 restart

------------------------------

Note that you also can use:

------------------------------

sh# rndc reconfig

------------------------------

to trigger a configuration reload. [Spence 20160617 Pascal also mentioned “rdnc reload”]

### Check that the changes are made, and recursion is refused:

------------------------------

sh# dig holbertonschool.com ANY @localhost

(Spence 20160616 i.e dig spence.tech ANY @ localhost)

------------------------------

Double check the flag and status code.

[Dora Korpar 20160616 maybe add an example of what the flags and status code should be/look like?]

(Spence 20160616 Agree. Anyone know how to do this?)

[RChong 20160617 Not absolutely sure about the flags part but I think it should look like this (at this point you haven’t made a zone file for your name server to use to provide an authoritative answer, and you disabled recursive querying on the name server. So using the local nameserver you configured, the dig would report status: REFUSED and ANSWER: 0, AUTHORITY: 0.):

;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 48462

;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; WARNING: recursion requested but not available

### What does a typical zone file look like ?

------------------------------------------------------------

@ IN SOA dns.gandi.net. hostmaster.gandi.net. (

1419475273 ; serial

10800 ; refresh (3 hours)

3600 ; retry (1 hour)

604800 ; expire (1 week)

10800 ; minimum (3 hours)

)

;; NS, MX entries at apex

IN NS a.dns.gandi.net.

NS b.dns.gandi.net.

NS c.dns.gandi.net.

MX 10 spool.mail.gandi.net.

MX 50 fb.mail.gandi.net.

;; Cheap load balancer

www IN A 10.5.3.1

A 10.5.3.2

;; Short TTL override

tmp 30 IN A 127.0.0.3

$ORIGIN seemingly.invalid.tfz.net.

@ 10800 IN CNAME 217.70.184.38

217.70.184.38 300 IN A 217.70.184.38

---------------------------------------------------------------

### We're going to add a zone now to this nameserver.

Use the /etc/bind/db.local as a template, install

a new file in /var/cache/bind/p/<yourdomain>.

Check that permissions for the new directory "p"

are correct.

# mkdir /var/cache/bind/p (make the directory which will house your zone file)

# chown bind:bind !$ (give your bind nameserver ownership over anything in the new directory?)

# cp /etc/bind/db.local !$/myfabulousdomain.tech (copy a template for a zone file from /etc/bind/ called ‘db.local’ to the new directory you made, ‘p’, and name the copy ‘myfabulousdomain.tech’)

[RChong 20160617 I put explanations in parentheses above, although mkdir and cp should be familiar commands. We had to do chown for one of our sysadmin projects as well. The commands are to create a new directory for the zone file, and copy a template file for your zone file into that new directory, which you will edit. $! expands to the last argument from the previous command, i.e. the directory you made, /var/cache/bind/p.]

### Now configure this zone to be loaded by bind at startup.

(Do this change in the /etc/bind/named.conf.local file)

------------------------------

zone "myfabulousdomain.tech" {

type master; // This is a master zonefile

file "/var/cache/bind/p/myfabulousdomain.tech";

}

------------------------------

### Reconfigure bind, check for errors in the logs

[RChong 20160617 You can check by restarting your bind ns (‘service bind9 restart’) then displaying the tail end of your syslog (‘tail /var/log/syslog’ on Ubuntu). Any errors caused by how you edited the conf files should show up.]

### Double check your zone is hosted properly.

# dig myfabulousdomain.tech @localhost

### Adapt the template until you have a proper setup, before we set the official nameservers for this domain at the registry.

- The nameserver should have a name, and resolve to this server's IP address

- You should have one NS record for the zone pointing to this nameserver's name

- Adjust the default TTL for a shorter value than the default 604k

- Have a www entry pointing to your web server machine

- Add any other records you like, such as a descriptive TXT record

- When you modify a zone, you can trigger a refresh using:

------------------------------

# rndc reload myfabulousdomain.tech

[RChong 20160617 At this point if you’ve written your zone file correctly, if you ‘dig <yourdomainhere> @localhost’ you should see an answer section (the IP address for your domain, as specified by your zone file), and I think maybe an authority section (your name server) in the output.

------------------------------

### Secure this nameserver a little more.

Anyone from outside, can have a full dump of your newly created zone

by simply asking for a zone transfer:

------------------------------

# dig myfabulousdomain.tech AXFR @X.X.X.X

------------------------------

In the global options, setup the allow-transfer {} option accordingly

to prevent that from happening.

------------------------------

options {

allow-transfer { none; };

}

------------------------------

Do this change in named.conf.options

### Create the glue record for the server in the zone using your registrar's website.

As finding the IP address for the server is mandatory in name lookups,

and this server is hosted in the domain name it's responsible for,

we need a hint from the parent zone about this server's IP address!

The registrar will push this IP address to the TLD so this hint is installed

properly.

### Change your NS to point to this new nameserver.

(Double check your NS works correctly ?)

[RChong 20160617: host -t ns <domain name> to find out authoritative name servers for domain.

Your nameserver should be listed. Then lookup your domains in the browser, and if your nameserver is working correctly, it should serve whatever’s stored on your webserver. Might take a while because of TTL and caching.]

### Find yourself a slave Nameserver teammate

First, allow them to transfer the zone, using a specific

allow-transfer option within the zone configuration

block of named.conf.local

------------------------------

zone "myfabulousdomain.tech" {

type master;

file "/var/cache/bind/primary/myfabulousdomain.tech";

allow-transfer { 1.2.3.4; };

}

------------------------------

### Normally, you should be a slave too

Let's replicate our teammate's zone

Setup your NS to become a slave of your teammate domain.

Create a subdirectory for slave zones, in /var/cache/bind/s/

Do not forget to fix permissions.

Create a zone for their domain in named.conf.local, declare it a

slave zone, setup the appropriate master and use the domain name

as a filename in this new directory for the zone.

------------------------------

zone "theirdomain.site" {

type slave;

file "/var/cache/bind/s/theirdomain.site";

masters { 1.2.3.5; };

}

------------------------------

### Check bind slave setup

Check that bind automatically downloads the zonefile from their server,

and stores it automatically in the new directory.

If not, look at logs on the master NS.

To trigger a zone transfer, use:

------------------------------

rndc retransfer domain.zone.name

------------------------------

### Check your master replicates correctly

Check, with dig, that the configured slave also has a copy of your own zone.

If that is the case, add the slave NS (you should ask for its hostname)

as a NS of your own domain in your zone, then on the registrar interface

so that TLD servers have the information.

Something like that:

------------------------------

@ IN NS myns

@ IN NS theirns.**dotterminated**.domain.name.

------------------------------

### A quick look at SOA fields

When you modify zone data, the slave will not update itself unless you

change the serial number of the zone to a higher number

(usually increments). A convention is to use today's date (20160401,

if appropriate) followed by a two-digits version number. That allows

for ~100 changes a day, which is way enough.

Double check this - change zone data, see in the logs that your NS

is notifying slaves, but that this change only gets propagated with the

right serial number increment.

On any slave, you can force a download using:

------------------------------

# rndc retransfer <zone>

------------------------------

Some things that are useful for this:

Sudo service bind9 restart whenever making a change to the named.conf files or a zone file

Checking /var/log/syslog to make sure changes kicked in fine (or if there were errors)

dig @[ipofhostingserver] -t SOA [domain here]

https://debian-administration.org/article/381/Updating\_bind\_serial\_numbers\_automatically