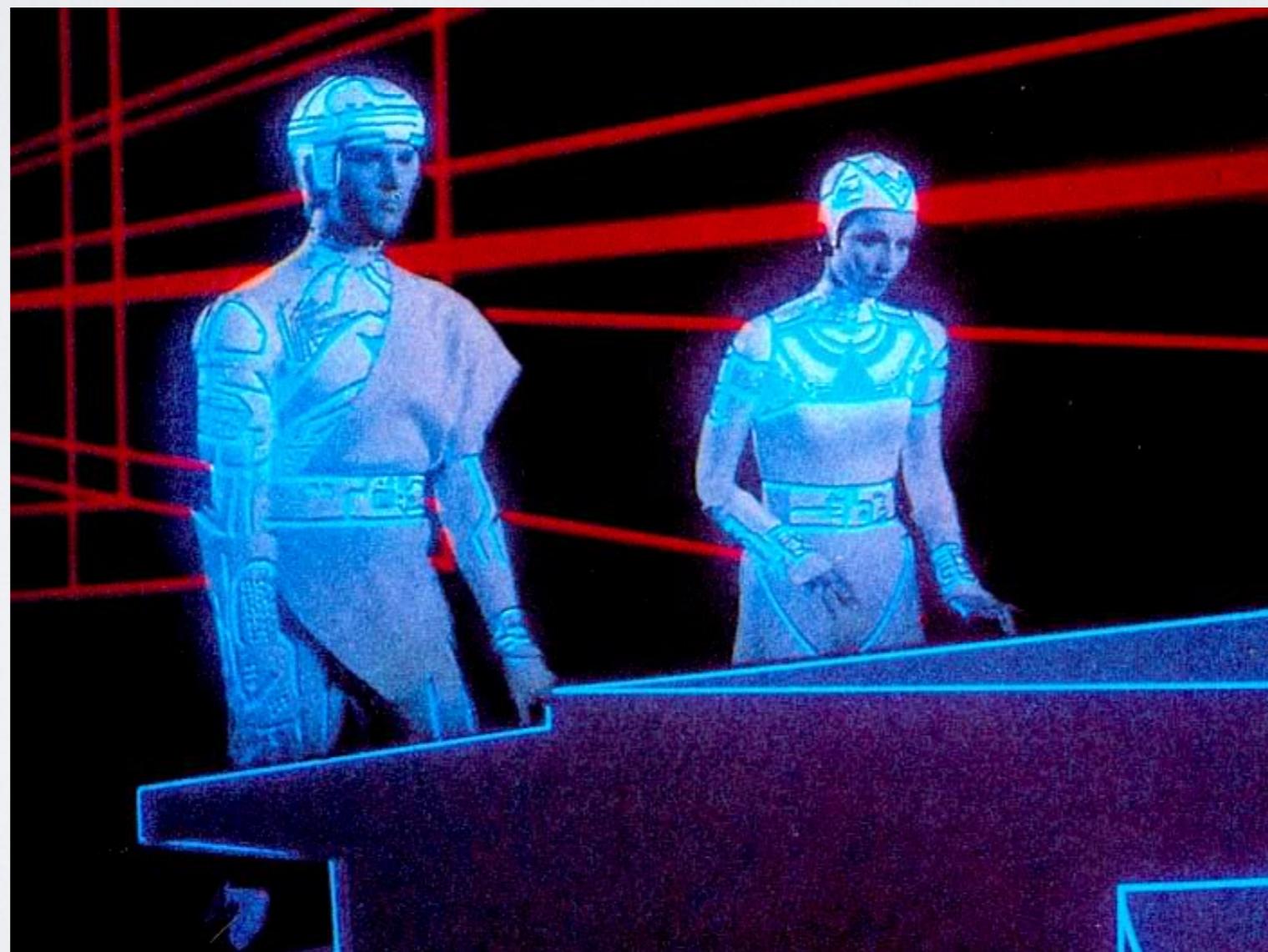


# THE ACCIDENTAL SYSADMIN

Or, how I learned to stop worrying and love the terminal



A **system administrator**, or **sysadmin**, is a person who is responsible for the upkeep, configuration, and reliable operation of computer systems; especially multi-user computers, such as servers.



[System administrator - Wikipedia, the free encyclopedia](#)  
[https://en.wikipedia.org/wiki/System\\_administrator](https://en.wikipedia.org/wiki/System_administrator) Wikipedia ▾

# WHAT IS A SYSADMIN?

I have no idea what I'm doing.



# I LOVE YOU, LAMP

We're going to set up and configure  
a LAMP server using Digital Ocean

# WHAT IS LAMP?

LAMP is an acronym for a “web application stack”, consisting of:

- Linux
- Apache
- MySQL
- PHP



# WHY DIGITAL OCEAN?

- Fast to set up and deploy
- Simple, uncluttered interface
- Free to get started
- \$10 credit when signing up via  
<http://bit.ly/accidentalsysadmin>



# “DROPLET” IS A FUNNY WORD

The screenshot shows the DigitalOcean Control Panel interface for creating a new droplet. The top navigation bar includes the DigitalOcean logo, a user profile for 'Quinn', and a search bar with the URL <https://cloud.digitalocean.com/droplets/new>. Below the header, there are links for Droplets, Images, DNS, API, and Support, along with a settings gear icon.

The main section is titled 'Create Droplet' with a help icon. The first step is 'Droplet Hostname', where the value 'AccidentalSysadmin' is entered. To the right, under 'Your Droplet', the host name is listed as 'AccidentalSysadm..'. The next step is 'Select Size', which displays five size options:

Size	Price	Memory	CPU	Disk	Transfer
\$5/mo	\$0.007/hour	512 MB	1 CPU	20 GB SSD Disk	1000 GB Transfer
<b>\$10/mo</b>	<b>\$0.015/hour</b>	1 GB	1 CPU	30 GB SSD Disk	2 TB Transfer
\$20/mo	\$0.030/hour	2 GB	2 CPUs	40 GB SSD Disk	3 TB Transfer
\$40/mo	\$0.060/hour	4 GB	2 CPUs	60 GB SSD Disk	4 TB Transfer
\$80/mo	\$0.119/hour	8 GB	4 CPUs	80 GB SSD Disk	5 TB Transfer
\$160/mo					
\$320/mo					
\$480/mo					
\$640/mo					

The '\$10/mo' plan is highlighted with a blue background. To the right of the size table, under 'Region', 'Image', 'Settings', and 'SSH Keys', their current values are listed as 'none'. A help icon is located at the bottom right.

# GO GIANTS!

DigitalOcean Control Panel Quinn

Digital Ocean, Inc. [US] <https://cloud.digitalocean.com/droplets/new>

Droplets Images DNS API Support ⚙️

## Select Region

 New York <table border="1"><tr><td>3</td><td>2</td><td>1</td></tr></table>	3	2	1	 Amsterdam <table border="1"><tr><td>3</td><td>2</td><td>1</td></tr></table>	3	2	1	 San Francisco <b>1</b>	 Singapore <table border="1"><tr><td>1</td></tr></table>	1	 London <table border="1"><tr><td>1</td></tr></table>	1
3	2	1										
3	2	1										
1												
1												

Frankfurt  

1
---

## Your Droplet

**Hostname** AccidentalSysadm..

**Size** \$10/mo

**Region** San Francisco 1

**Image** 6.5 x64

**Settings** none

**SSH Keys** none

## Select Image

Distributions Applications Snapshots Backups 💡

# PICK AN OS, ANY OS

DigitalOcean Control Panel Quinn

Digital Ocean, Inc. [US] <https://cloud.digitalocean.com/droplets/new>

Droplets Images DNS API Support ⚙️

## Select Image

Distributions Applications Snapshots Backups

 UBUNTU Select Version ▾	 FREEBSD Select Version ▾	 FEDORA Select Version ▾	 DEBIAN Select Version ▾
 COREOS Select Version ▾	 CENTOS 6.5 x64 ▾		

Your Droplet

**Hostname** AccidentalSysadm..

**Size** \$10/mo

**Region** San Francisco 1

**Image** 6.5 x64

**Settings** none

**SSH Keys** none

## Available Settings

# MORE OPTIONS?!

The screenshot shows the DigitalOcean Control Panel interface. At the top, there's a navigation bar with icons for Home, Back, Forward, and Refresh, followed by the URL "Digital Ocean, Inc. [US] https://cloud.digitalocean.com/droplets/new". On the right of the bar are user profile icons for "Quinn" and other standard browser controls.

The main content area has a header "Available Settings" with four checkboxes: "Private Networking", "Backups", "IPv6", and "User Data". Below this is a section titled "Add SSH Keys (Optional)" containing a text input field with placeholder text "quinn@macbook..." and a link "+ Add SSH Key". A note below explains that adding an SSH key is a recommended security measure; if not added, a root password will be sent via email.

To the right, under "Your Droplet", are several configuration details:

- Hostname**: AccidentalSysadm..
- Size**: \$10/mo
- Region**: San Francisco 1
- Image**: 6.5 x64
- Settings**: none
- SSH Keys**: none

A large green button at the bottom center says "Create Droplet".

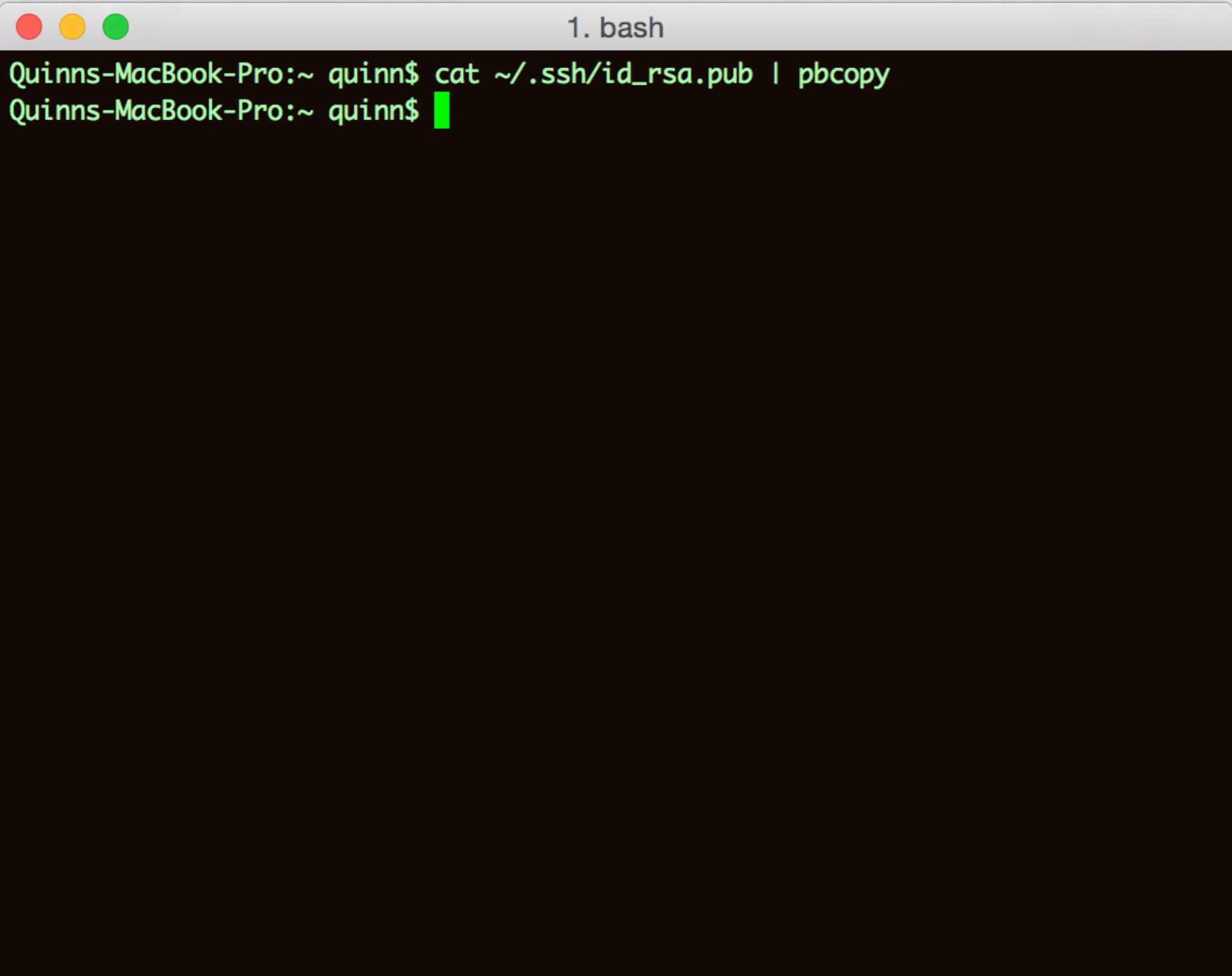
# KILL YOUR PASSWORD

ssh-keygen

```
Quinns-MacBook-Pro:~ quinn$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/quinn/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/quinn/.ssh/id_rsa.
Your public key has been saved in /Users/quinn/.ssh/id_rsa.pub.
The key fingerprint is:
6c:e1:34:38:ea:fe:59:1c:2b:de:0c:93:d0:7e:2a:87 quinn@Quinns-MacBook-Pro.local
The key's randomart image is:
+--[ RSA 2048]----+
|                               |
|                               |
|       .                     |
|       o +                   |
|       o = o                 |
|       o . S                 |
|       . o + o               |
|       ..* =                 |
|       .E..@                 |
|       .+= o                 |
+-----+
Quinns-MacBook-Pro:~ quinn$
```

# COPY TO CLIPBOARD

```
cat ~/.ssh/id_rsa.pub | pbcopy
```



A screenshot of a macOS terminal window titled "1. bash". The window has three colored close buttons (red, yellow, green) at the top left. The title bar also displays "1. bash". The main area of the terminal shows the command being run: "Quinns-MacBook-Pro:~ quinn\$ cat ~/.ssh/id\_rsa.pub | pbcopy". The command is completed, as indicated by the prompt "Quinns-MacBook-Pro:~ quinn\$". The background of the terminal is dark, and the text is white.

# VI IS THE TEXT EDITOR FROM HELL

Honestly, getting your head around Vi is the hardest part of any of this.

I Am Devloper  
@iamdevloper

+ Follow

I've been using Vim for about 2 years now,  
mostly because I can't figure out how to exit  
it.

RETWEETS 12,667 FAVORITES 6,328

3:26 PM - 17 Feb 2014

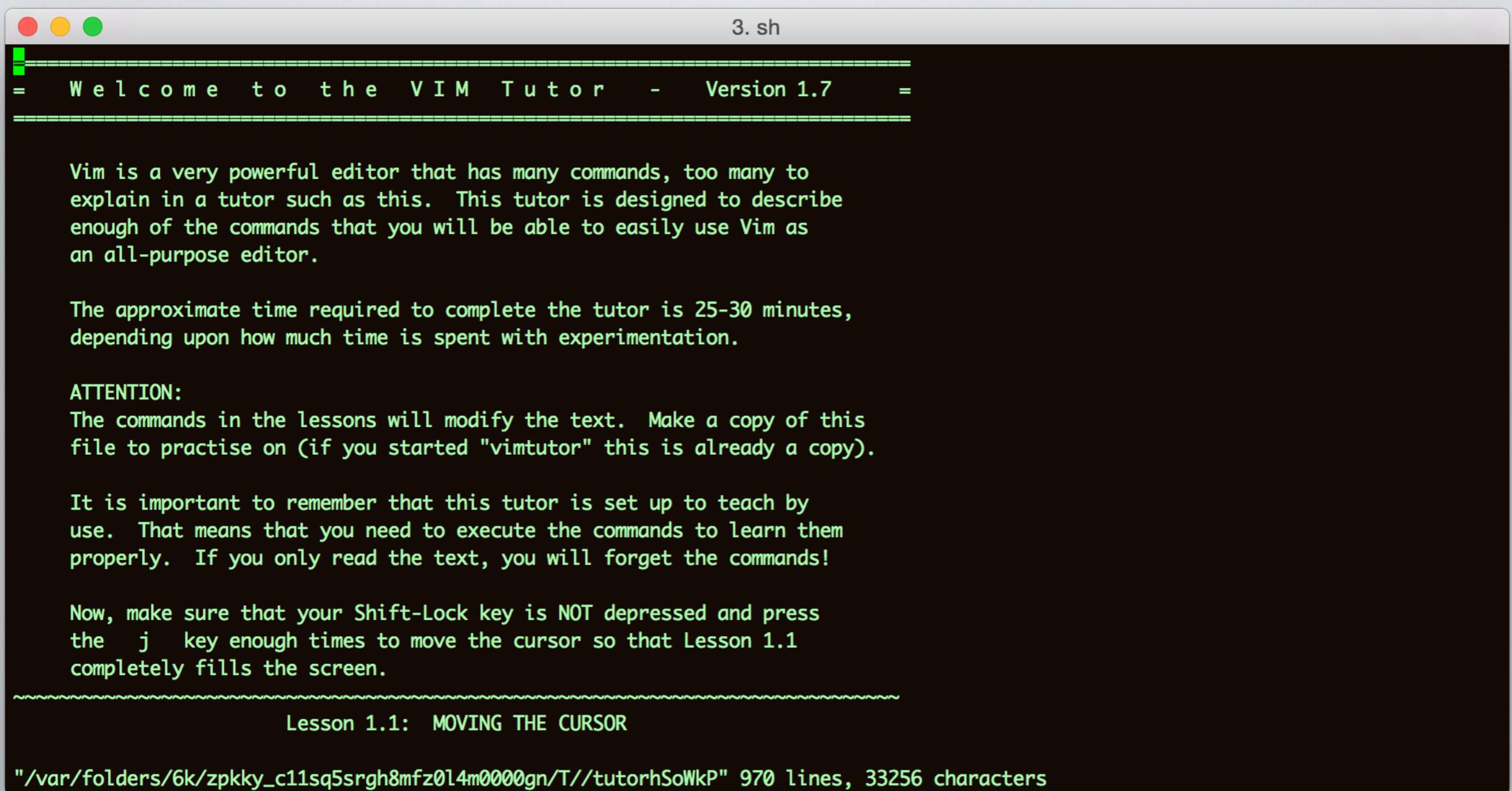
• • •

“Some people say **vye**, some say **vee-eye** (the vi manual suggests this) and some Roman numerologists say **six**.”

<http://ss64.com/bash/syntax-pronounce.html>

# BUT, VI/VIM WILL TEACH YOU HOW TO USE IT

vimtutor



A screenshot of a terminal window titled "3.sh". The window contains the following text:

```
= Welcome to the VIM Tutor - Version 1.7 =
=====
Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this. This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text. Make a copy of this
file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use. That means that you need to execute the commands to learn them
properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press
the j key enough times to move the cursor so that Lesson 1.1
completely fills the screen.

=====
```

Lesson 1.1: MOVING THE CURSOR

/var/folders/6k/zpkky\_c11sq5srgh8mfz0l4m0000gn/T//tutorhSoWkP" 970 lines, 33256 characters

# SET UP YOUR DOMAIN

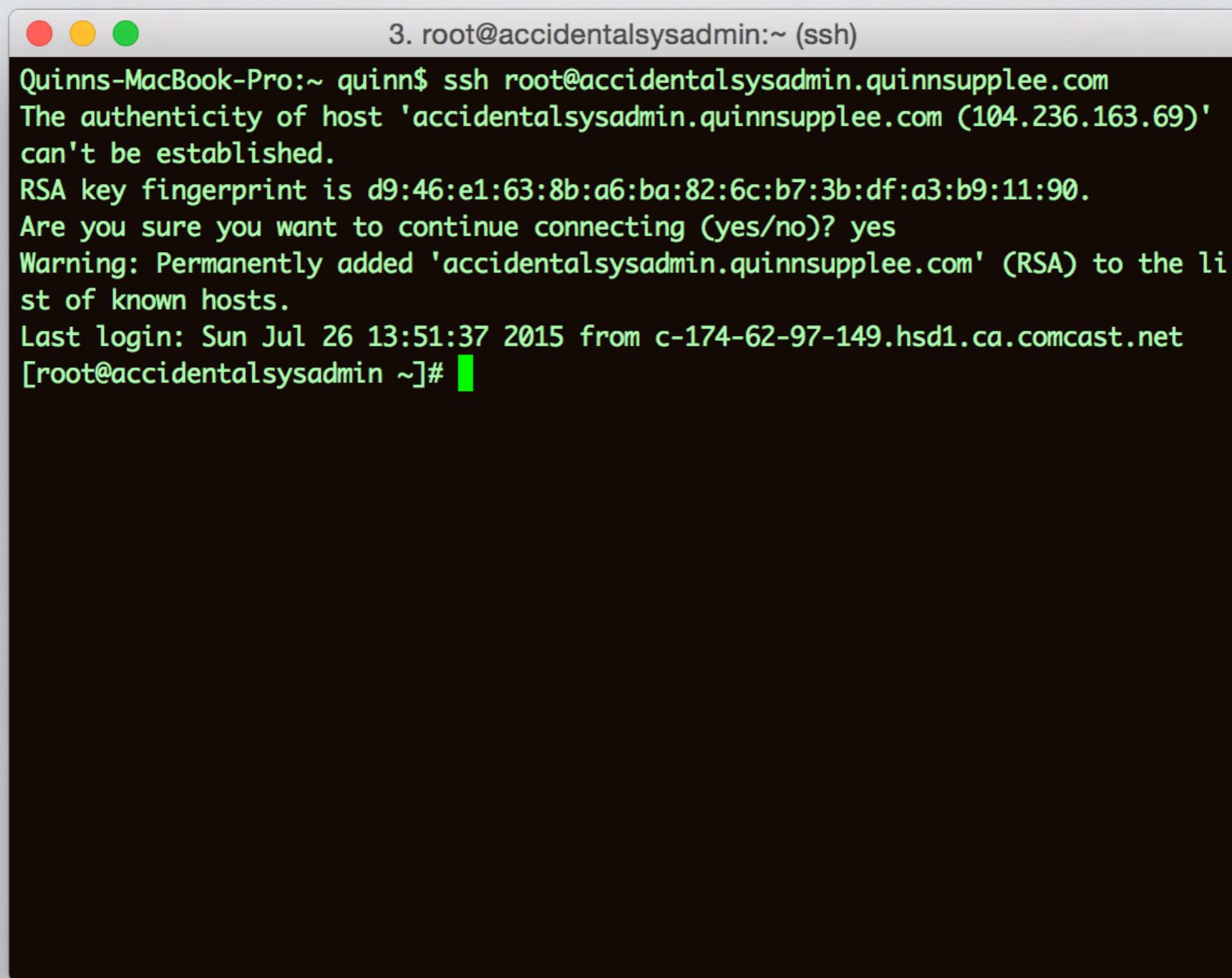
Or, sub-domain

The screenshot shows the Hover domain management interface. At the top, there's a navigation bar with links for PRICING, EMAIL, BLOG, WEBMAIL, OUR STORY, and HELP. Below that is a green header bar with links for FIND A DOMAIN, TRANSFER, RENEW, EARN HOVER CREDIT, and YOUR ACCOUNT. A promotional message for a .CO domain sale is displayed. The main content area shows the domain `quinnsupplee.com` with status information: active, registered on 2010-02-02, and renewing on 2016-02-02. The DNS tab is selected, showing a list of records. The first record, for host `accidentalsysadmin` with type A and value 104.236.163.69, is highlighted with a red border. Other records listed are for host `@` with type MX and values 10 mx1.emailsrvr.com and 20 mx2.emailsrvr.com.

HOST	TYPE	VALUE
accidentalsysadmin	A	104.236.163.69
@	MX	10 mx1.emailsrvr.com
@	MX	20 mx2.emailsrvr.com

# NOW LET'S START HACKIN'

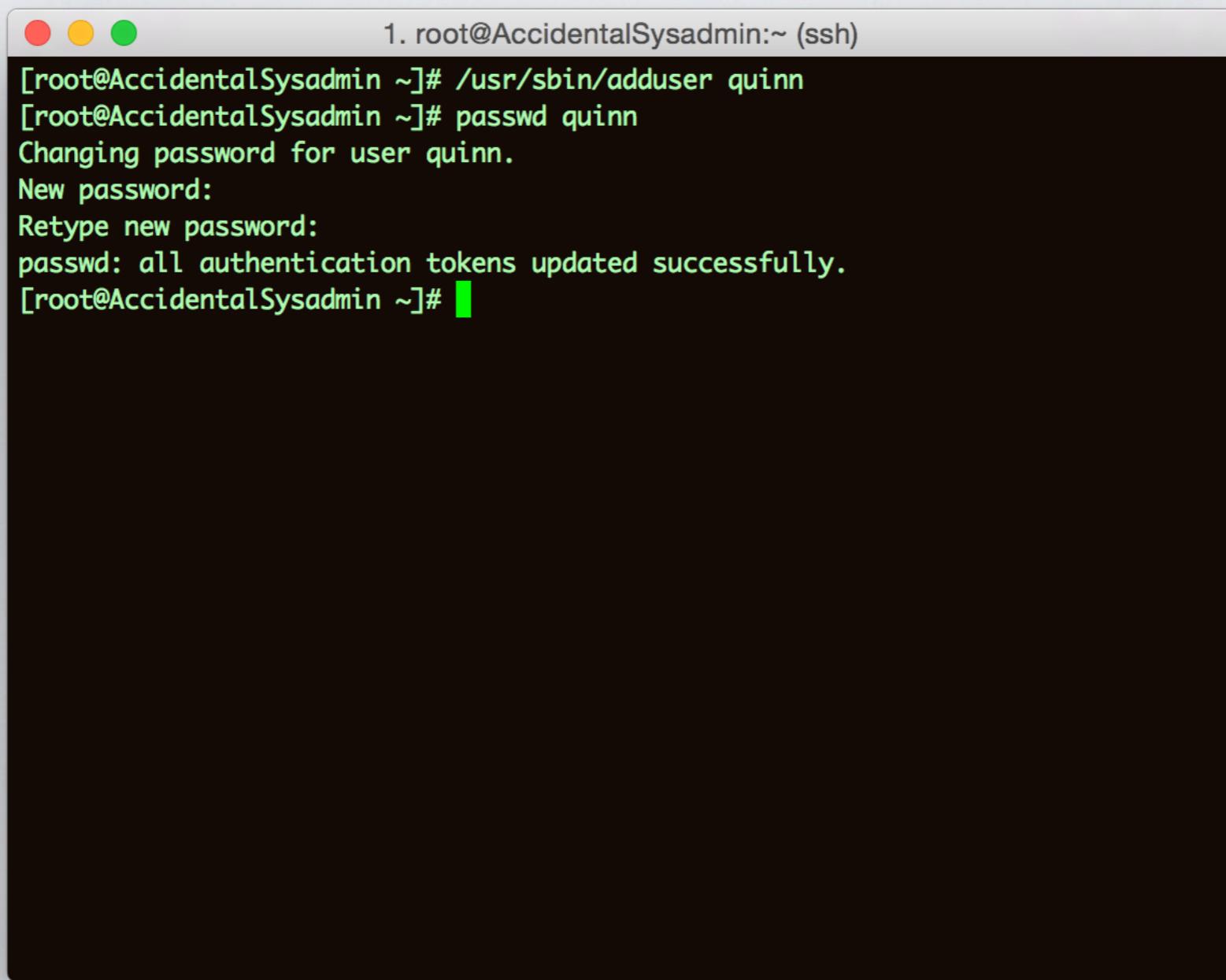
```
ssh root@accidentalsysadmin.quinnsupplee.com
```



3. root@accidentalsysadmin:~ (ssh)  
Quinns-MacBook-Pro:~ quinn\$ ssh root@accidentalsysadmin.quinnsupplee.com  
The authenticity of host 'accidentalsysadmin.quinnsupplee.com (104.236.163.69)'  
can't be established.  
RSA key fingerprint is d9:46:e1:63:8b:a6:ba:82:6c:b7:3b:df:a3:b9:11:90.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'accidentalsysadmin.quinnsupplee.com' (RSA) to the li  
st of known hosts.  
Last login: Sun Jul 26 13:51:37 2015 from c-174-62-97-149.hsd1.ca.comcast.net  
[root@accidentalsysadmin ~]#

# CREATE A NEW USER

```
/usr/sbin/adduser quinn  
passwd quinn
```



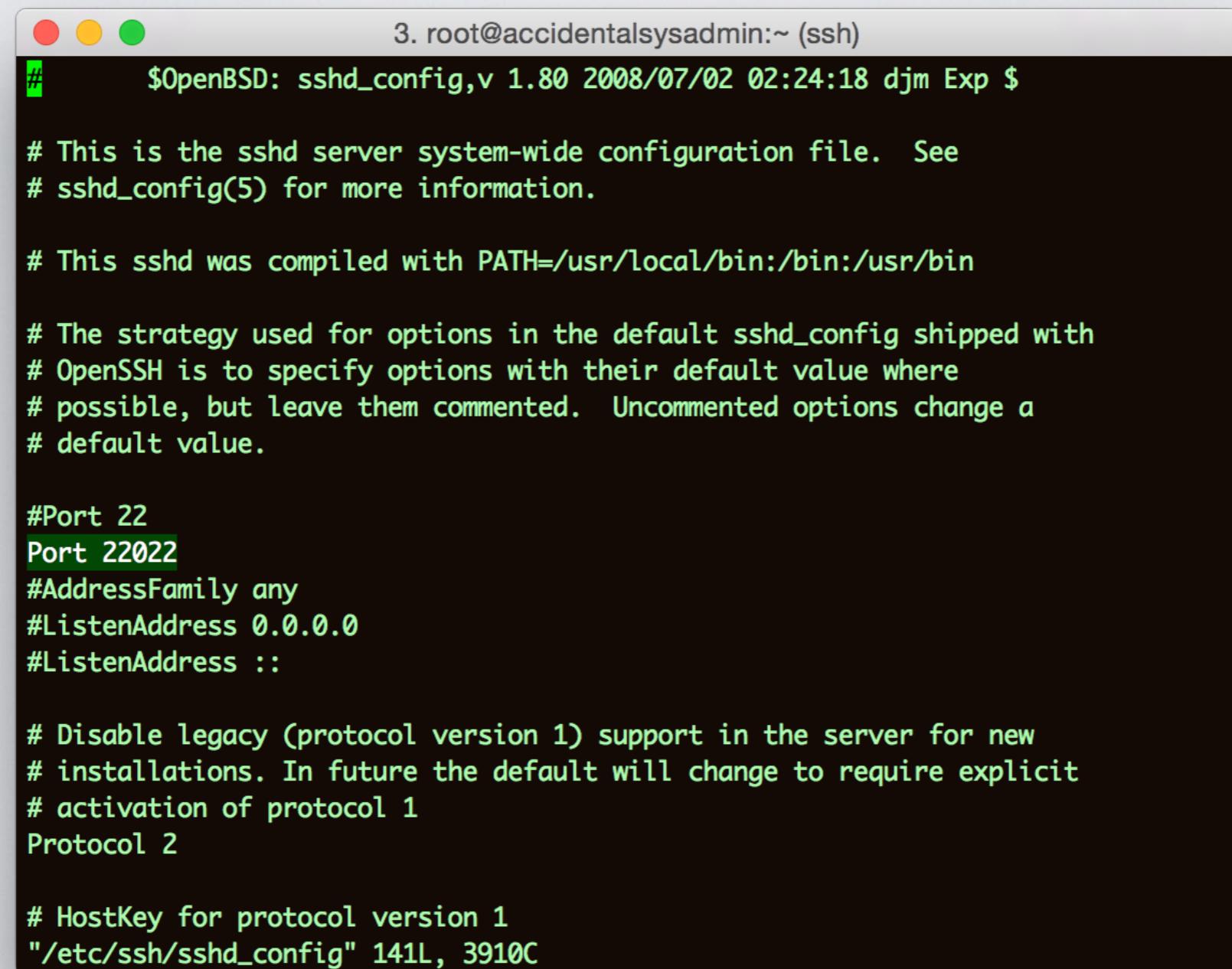
1. root@AccidentalSysadmin:~ (ssh)

```
[root@AccidentalSysadmin ~]# /usr/sbin/adduser quinn  
[root@AccidentalSysadmin ~]# passwd quinn  
Changing password for user quinn.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@AccidentalSysadmin ~]#
```

# CHANGE DEFAULT SSH PORT

Change the line that says “Port 22”

Port 22022



3. root@accidentalsysadmin:~ (ssh)

```
$OpenBSD: sshd_config,v 1.80 2008/07/02 02:24:18 djm Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options change a
# default value.

#Port 22
Port 22022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

# Disable legacy (protocol version 1) support in the server for new
# installations. In future the default will change to require explicit
# activation of protocol 1
Protocol 2

# HostKey for protocol version 1
"/etc/ssh/sshd_config" 141L, 3910C
```

# DISABLE ROOT LOGIN

```
PermitRootLogin no
```



3. root@accidentalsysadmin:~ (ssh)

```
#ServerKeyBits 1024

# Logging
# obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PermitRootLogin no

#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile    .ssh/authorized_keys
#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody
```

# GIVE YOURSELF SUPERPOWERS

Edit your **sudo** configuration to grant root power to your new user

```
/usr/sbin/visudo
```

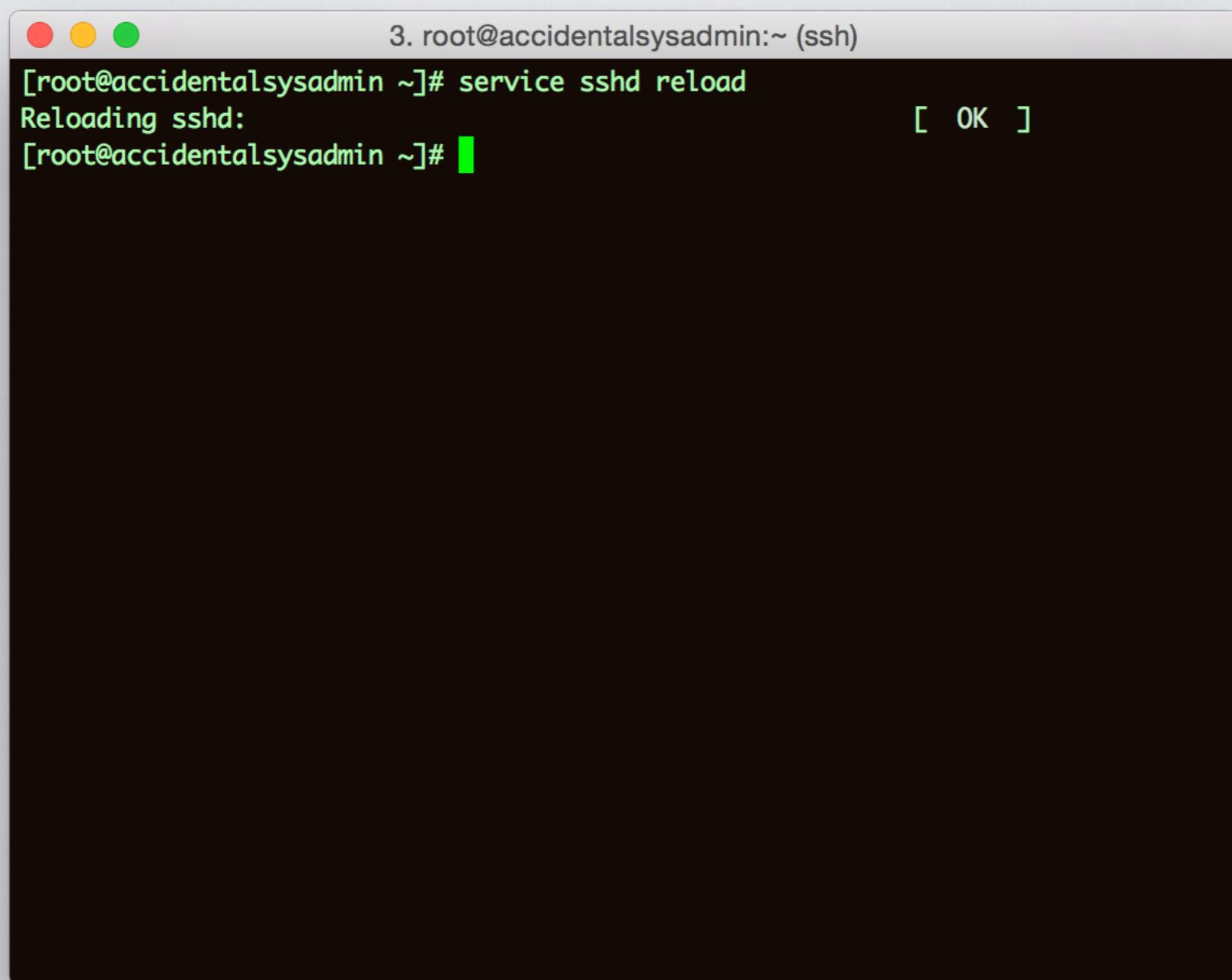
Find the line “root ALL=(ALL) ALL”,  
and add, below it:

```
"quinn          ALL=(ALL)          ALL"
```

Save and exit Vi (*Hahahahaha!*)

# RELOAD SSH SERVICE

```
service sshd reload
```



3. root@accidentalsysadmin:~ (ssh)

```
[root@accidentalsysadmin ~]# service sshd reload
Reloading sshd: [ OK ]
```

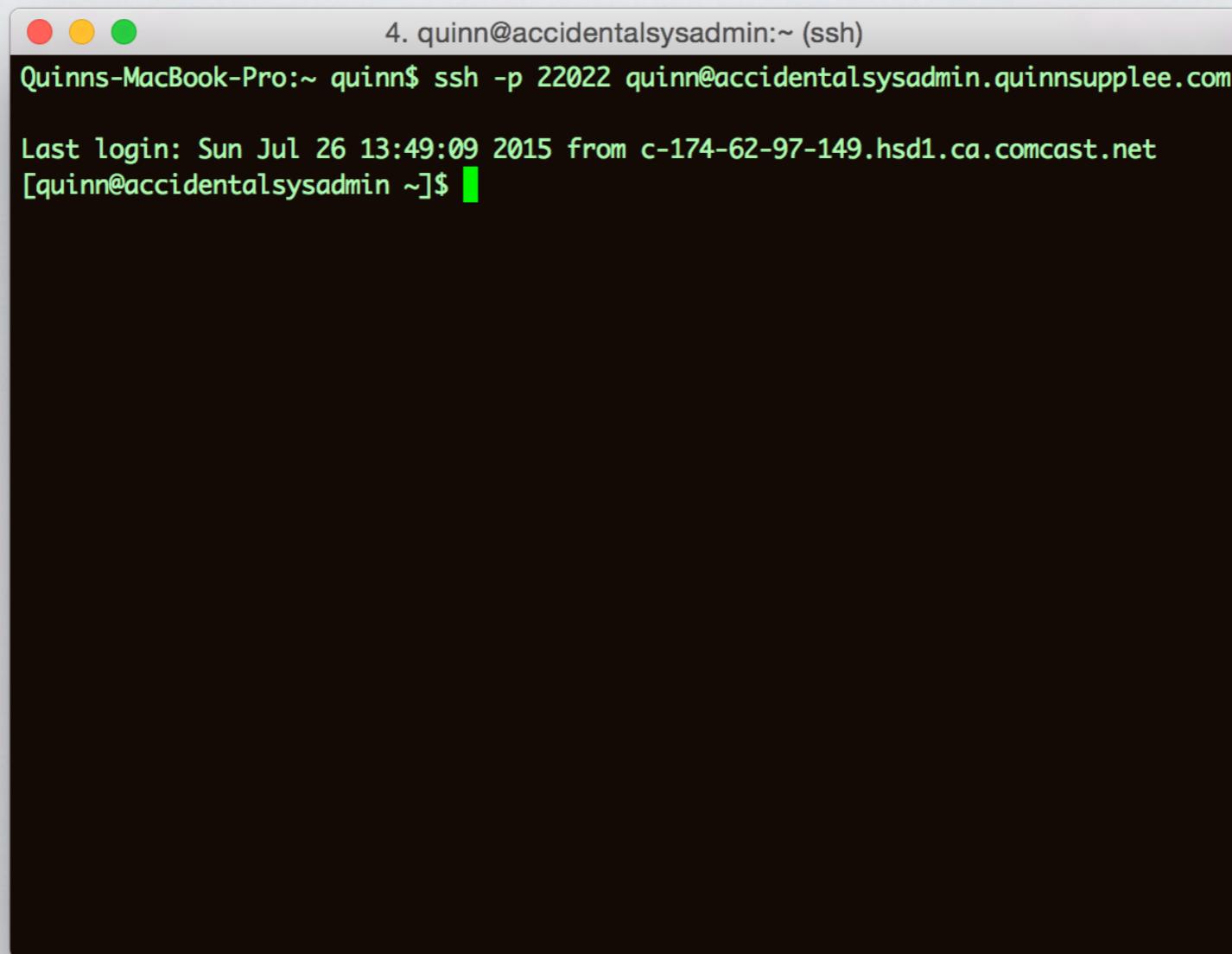
[root@accidentalsysadmin ~]#

A screenshot of a terminal window titled "3. root@accidentalsysadmin:~ (ssh)". The window shows a command-line interface where the user has run the command "service sshd reload". The output of the command is displayed, showing "Reloading sshd:" followed by a "[ OK ]" message in green text. The terminal has a dark background with light-colored text. The title bar includes three colored circles (red, yellow, green) typically used for window control.

# LOG IN AS YOUR NEW USER

Keep "root" session running for now, so in a new terminal window:

```
ssh -p 22022 quinn@accidentalsysadmin.quinnsupplee.com
```



A screenshot of a Mac OS X terminal window. The window title bar says "4. quinn@accidentalsysadmin:~ (ssh)". The main pane of the terminal shows the command "Quinns-MacBook-Pro:~ quinn\$ ssh -p 22022 quinn@accidentalsysadmin.quinnsupplee.com" followed by the output of the command: "Last login: Sun Jul 26 13:49:09 2015 from c-174-62-97-149.hsd1.ca.comcast.net [quinn@accidentalsysadmin ~]\$". The terminal has its characteristic dark gray background and light gray text.

# SET UP THE HOSTNAME

```
sudo /etc/sysconfig/network
```



```
4. quinn@accidentalsysadmin:~ (ssh)
[quinn@accidentalsysadmin ~]$ sudo vi /etc/sysconfig/network
[sudo] password for quinn:
[quinn@accidentalsysadmin ~]$ █
```

```
4. quinn@accidentalsysadmin:~ (ssh)
NETWORKING=yes
HOSTNAME=accidentalsysadmin.quinnsupplee.com
~
```



Yum is a package manager for Linux. To download and install many components of Linux, you just need to type:

```
sudo yum install <program>
```

The **Yellowdog Updater, Modified** (**yum**) is an open-source command-line package-management utility for **Linux** operating systems using the **RPM Package Manager**. Though **yum** has a command-line interface, several other tools provide graphical user interfaces to **yum** functionality.



Yum - Wikipedia

[https://en.wikipedia.org/wiki/Yellowdog\\_Updater,\\_Modified](https://en.wikipedia.org/wiki/Yellowdog_Updater,_Modified) Wikipedia ▾

OK, LET'S GET LAMP'D

# GIVE ME AN "A" (APACHE)

```
sudo yum install httpd
```

```
1. root@AccidentalSysadmin:/home (ssh)
----> Package httpd-tools.x86_64 0:2.2.15-39.el6.centos will be installed
----> Package mailcap.noarch 0:2.1.31-2.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
          Package           Arch    Version        Repository  Size
=====
Installing:
  httpd           x86_64  2.2.15-39.el6.centos   base      825 k
Installing for dependencies:
  apr             x86_64  1.3.9-5.el6_2         base     123 k
  apr-util        x86_64  1.3.9-3.el6_0.1       base      87 k
  apr-util-ldap   x86_64  1.3.9-3.el6_0.1       base      15 k
  httpd-tools     x86_64  2.2.15-39.el6.centos   base      75 k
  mailcap         noarch  2.1.31-2.el6          base      27 k

Transaction Summary
=====
Install      6 Package(s)

Total download size: 1.1 M
Installed size: 3.6 M
Is this ok [y/N]:
```

Hey! We just installed a web server!

# START APACHE

Once HTTPD is installed, let's tell it what the server name is

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

Find "ServerName" and add:

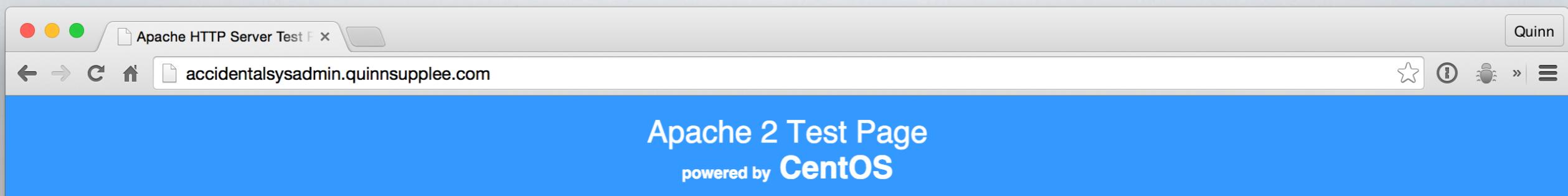
```
ServerName accidentalsysadmin.quinnsupplee.com
```

Save and exit

```
service httpd start
```

Now go see your site! <http://accidentalsysadmin.quinnsupplee.com>

# IT'S JUST LIKE A NEW BABY



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly.

---

**If you are a member of the general public:**

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](http://www.example.com), you should send e-mail to "webmaster@example.com".

**If you are the website administrator:**

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

**About CentOS:**

The Community ENTerprise Operating System (CentOS) Linux is a community-supported enterprise distribution derived from sources freely provided to the public by Red Hat. As such, CentOS Linux aims to be functionally compatible with Red Hat Enterprise Linux. The CentOS Project is the organization that builds CentOS. We mainly change packages to remove upstream vendor branding and artwork.

For information on CentOS please visit the [CentOS website](#).

**Note:**

CentOS is an Operating System and it is used to power this website; however, the webserver is owned by the domain owner and not the CentOS Project. **If you have issues with the content of this site, contact the owner of the domain, not the CentOS Project.**

# GIVE ME AN "M" (MYSQL)

```
sudo yum install mysql-server  
sudo service mysqld start
```

Now, MySQL should be running. Let's secure it!

```
sudo /usr/bin/mysql_secure_installation
```

Press ENTER to all prompts.

Be sure to provide a strong password and store it somewhere safe.

# GIVE ME A "P" (PHP)

Now we're going to install PHP with MySQL support

```
yum install php php-mysql
```

And we can check our PHP configuration by adding a quick info file:

```
vi info.php  
  
<?php phpinfo(); ?>
```

Save and exit

Then check it out at:

<http://accidentalsysadmin.quinnsupplee.com/info.php>

# CHECK THE PHP SWEETNESS

The screenshot shows a web browser window with the title bar "phpinfo()". The address bar contains the URL "accidentalsysadmin.quinnsupplee.com/info.php". The main content is a table titled "PHP Version 5.3.3" with the PHP logo in the top right corner. The table rows provide various configuration details:

System	Linux accidentalsysadmin.quinnsupplee.com 2.6.32-504.12.2.el6.x86_64 #1 SMP Wed Mar 11 22:03:14 UTC 2015 x86_64
Build Date	Jul 9 2015 17:39:38
Configure Command	<pre>./configure' '--build=x86_64-redhat-linux-gnu' '--host=x86_64-redhat-linux-gnu' '--target=x86_64-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib64' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/var/lib' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-libdir=lib64' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with-bz2' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--with-xpm-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '--with-kerberos' '--enable-ucd-snmp-hack' '--enable-shmop' '--enable-calendar' '--without-sqlite' '--with-libxml-dir=/usr' '--enable-xml' '--with-system-tzdata' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--disable-dom' '--disable-dba' '--without-unixODBC' '--disable-pdo' '--disable-xmlreader' '--disable-xmlwriter' '--without-sqlite3' '--disable-phar' '--disable-fileinfo' '--disable-json' '--without-pspell' '--disable-wddx' '--without-curl' '--disable-posix' '--disable-sysvmsg' '--disable-sysvshm' '--disable-sysvsem'</pre>
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional ini files	/etc/php.d/apcu.ini, /etc/php.d/bcmath.ini, /etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/gd.ini, /etc/php.d/imap.ini, /etc/php.d/iconv.ini

# THAT WAS EASY!

Let's install some additional PHP modules!

```
sudo yum install php-cli php-fpm php-cgi php-mysql php-xmlrpc php-curl php-gd php-apc php-pear php-imap php-mcrypt php-pspell
```

There are lots more you may be interested in. To get a list of available PHP modules, type:

```
yum search php-
```

# A FEW MORE THINGS

Now let's make sure PHP and MySQL start on every server reboot:

```
sudo chkconfig httpd on  
sudo chkconfig mysqld on
```

Let's install some other stuff we'll surely need:

```
sudo yum install zip unzip gzip rsync wget
```

# I CAN HAZ PHPMYADMIN?

Let's install and setup phpMyAdmin

```
sudo yum install wget
```

```
cd ~
```

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

```
sudo yum install phpmyadmin
```

# CONFIGURE PHPMYADMIN

Let's configure and secure phpMyAdmin

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

You want to replace all instances of **127.0.0.1**  
with your own **local IP address**

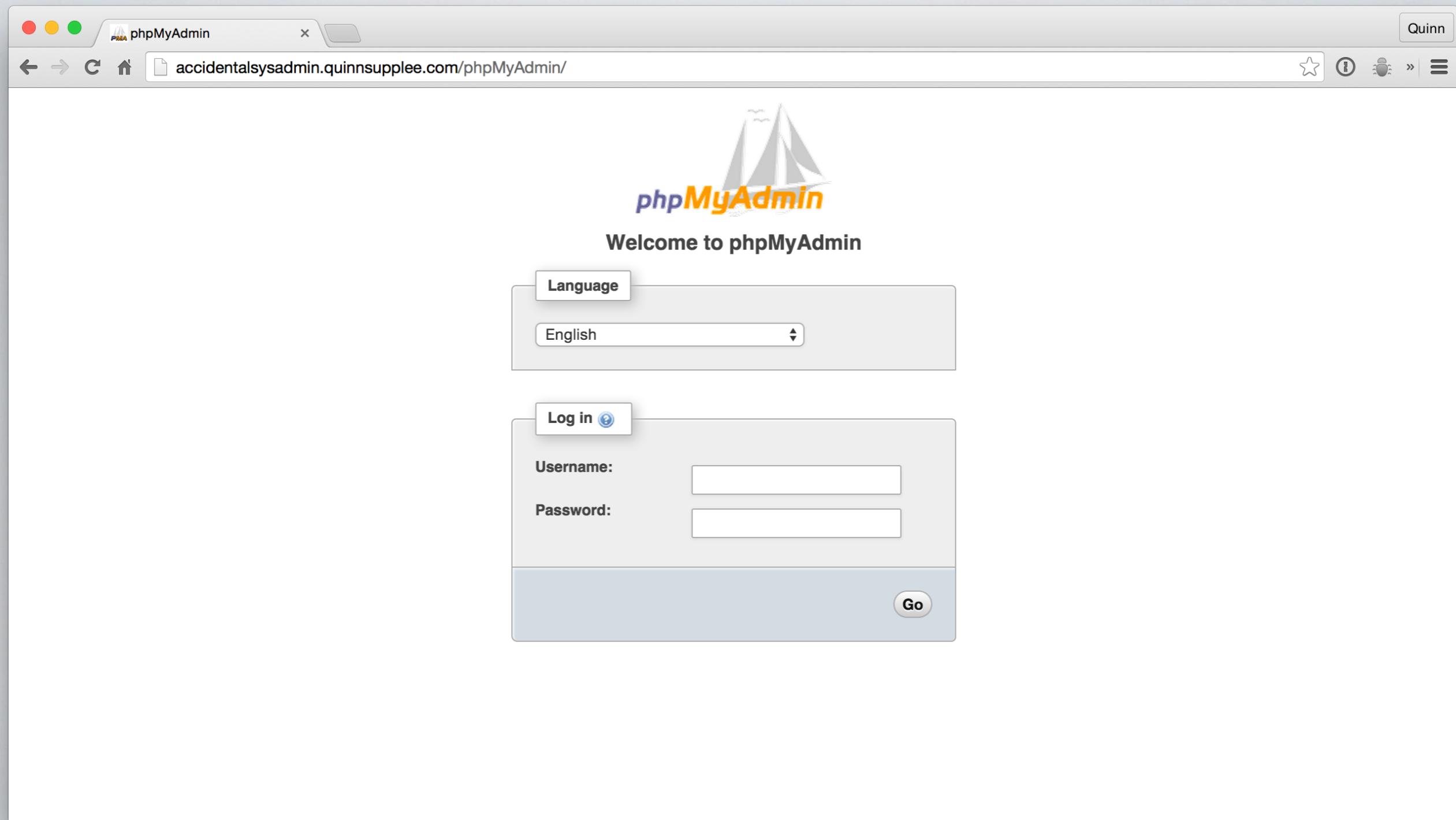
Do it manually or with a find-replace:

```
:%s/127.0.0.1/255.255.255.255/g
```

Now restart apache:

```
sudo service httpd restart
```

# VISIT YOUR NEW PHPMYADMIN



# LET'S INSTALL MOD\_SECURITY

```
sudo yum install pcre* libxml2* libcurl* lua*  
libtool openssl mod_security
```

Now let's add some custom WP-focused brute-force protection to it:

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

Add to end of file, right BEFORE closing </IfModule>...



## 5. vim

```
SecAction phase:1,nolog,pass,initcol:ip=%{REMOTE_ADDR},initcol:user=%{REMOTE_ADDR},id:5000134
<Locationmatch "/wp-login.php">
    # Setup brute force detection.

    # React if block flag has been set.
    SecRule user:bf_block "@gt 0" "deny,status:401,log,id:5000135,msg:'ip address blocked for 5 minutes, more than 10 login attempts in 3 minutes.'"

    # Setup Tracking. On a successful login, a 302 redirect is performed, a 200 indicates login failed.
    SecRule RESPONSE_STATUS "^302" "phase:5,t:none,nolog,pass,setvar:ip.bf_counter=0,id:5000136"
    SecRule RESPONSE_STATUS "^200" "phase:5,chain,t:none,nolog,pass,setvar:ip.bf_counter+=1,deprecatevar:ip.bf_counter=1/180,id:5000137"
    SecRule ip:bf_counter "@gt 10" "t:none,setvar:user.bf_block=1,expirevar:user.bf_block=300,setvar:ip.bf_counter=0"
</locationmatch>
~
~
~
~
~
~
-- INSERT --
```

<https://halfelf.org/2013/wp-login-protection-modsec/>

# WORDPRESS, THE NINJA METHOD

# LET'S INSTALL WP-CLI

WP-CLI is a Swiss Army Knife for WordPress management

```
cd ~  
  
curl -O https://raw.githubusercontent.com/wp-  
cli/builds/gh-pages/phar/wp-cli.phar  
  
sudo chmod +x wp-cli.phar  
  
sudo mv wp-cli.phar /usr/local/bin/wp
```

<http://wp-cli.org/>

# SET UP DB AND ROCK WP

Using phpMyAdmin, create a new user, a database, and grant all permissions:

**username: accidentalsys**  
**password: w0rdpr3ssSVCKS**

Back in Terminal:

```
cd /var/www/
sudo chown -R quinn.quinn html
cd html
wp core download
```

# RUN THE “5-MINUTE WP INSTALLER”



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="accidentalsys"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="accidentalsys"/>	Your MySQL username
Password	<input type="text" value="w0rdpr3ssSVCKS"/>	...and your MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if <code>localhost</code> does not work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

# LET'S SECURE OUR WP INSTALLATION

Make sure you aren't using "admin" as your admin username

Make sure you have a wicked-strong password

Install some super useful security plugins

```
wp plugin install sucuri-scanner
```

```
wp plugin install login-lockdown
```

Create and secure our WP Uploads directory

```
mkdir wp-content/uploads
```

```
sudo chown apache.apache wp-content/uploads
```

Enable and configure Sucuri Scanner and Login Lockdown plugins

# WP HARDENING, CONTINUED

Disable file editing from within WordPress. Add to **wp-config.php**:

```
define('DISALLOW_FILE_EDIT', true);
```

Disable PHP execution from within **wp-content** and **uploads** directories.

Create a **.htaccess** file in **wp-content** and **wp-content/uploads**, and add:

```
<Files *.php>
deny from all
</Files>
```

# SET UP THE MAIN .HTACCESS FILE

Mine looks like this. Yours might look different.

```
# Block the include-only files.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^wp-admin/includes/ - [F,L]
RewriteRule !^wp-includes/ - [S=3]
RewriteRule ^wp-includes/[^\]+\.php$ - [F,L]
RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
RewriteRule ^wp-includes/theme-compat/ - [F,L]
</IfModule>

# Block access to wp-config.php
<files wp-config.php>
order allow,deny
deny from all
</files>

# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
```

# ENJOY YOUR NEW, SECURE WORDPRESS INSTALLATION

The screenshot shows a web browser window with the title bar "The Accidental Sysadmin" and the URL "accidentalsysadmin.quinnsupplee.com". The page content is as follows:

**The Accidental Sysadmin**  
Just another WordPress site

**Hello world!**

Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!

July 26, 2015 | 1 Comment | Edit

**RECENT POSTS**

Hello world!

**RECENT COMMENTS**

Mr WordPress on Hello world!

Proudly powered by WordPress

# OTHER THINGS OF INTEREST

## **Fail2Ban**

<http://www.fail2ban.org/>

## **ModPagespeed**

<https://developers.google.com/speed/pagespeed/module/>

## **DenyHosts**

<http://denyhosts.sourceforge.net/>

## **AWStats**

<http://www.awstats.org/>

## **Apache ModInfo**

[https://httpd.apache.org/docs/2.2/mod/mod\\_info.html](https://httpd.apache.org/docs/2.2/mod/mod_info.html)

## **Apache ModStatus**

[https://httpd.apache.org/docs/2.2/mod/mod\\_status.html](https://httpd.apache.org/docs/2.2/mod/mod_status.html)

# LINKS AND REFERENCES

**A command line interface for WordPress**

<http://bit.ly/1fyBsdh>

**Basic vi Commands**

<http://bit.ly/1OvsQQ3>

**Red Hat / CentOS Install mod\_security Apache Intrusion Detection And Prevention Engine**

<http://bit.ly/1gYweZl>

**How to Change the Hostname of a Linux System**

<http://bit.ly/1gf35ZK>

**Vi Cheat Sheet**

<http://bit.ly/1OvsWHz>

**What is my IP address?**

<http://bit.ly/1KsOtzT>

**How to harden Apache web server with mod\_security and mod\_evasive on CentOS**

<http://bit.ly/1estpxQ>

**Hardening WordPress**

<http://bit.ly/1MsblBT>

**WordPress Login Protection with ModSecurity**

<http://bit.ly/1Mvykgk>

# LINKS AND REFERENCES

**What are PHP extensions and libraries WP needs and/or uses?**

<http://bit.ly/1GT1vBV>

**How To Get Started With mod\_pagespeed with Apache on a CentOS and Fedora Cloud Server**

<http://do.co/1JjtFRF>

**How To Install DenyHosts on CentOS 6**

<http://do.co/1I3QQGb>

**How To Install Linux, Apache, MySQL, PHP (LAMP) stack On CentOS 6**

<http://do.co/1gf3qvz>

**How To Protect SSH with fail2ban on CentOS 6**

<http://do.co/1ImF6h9>

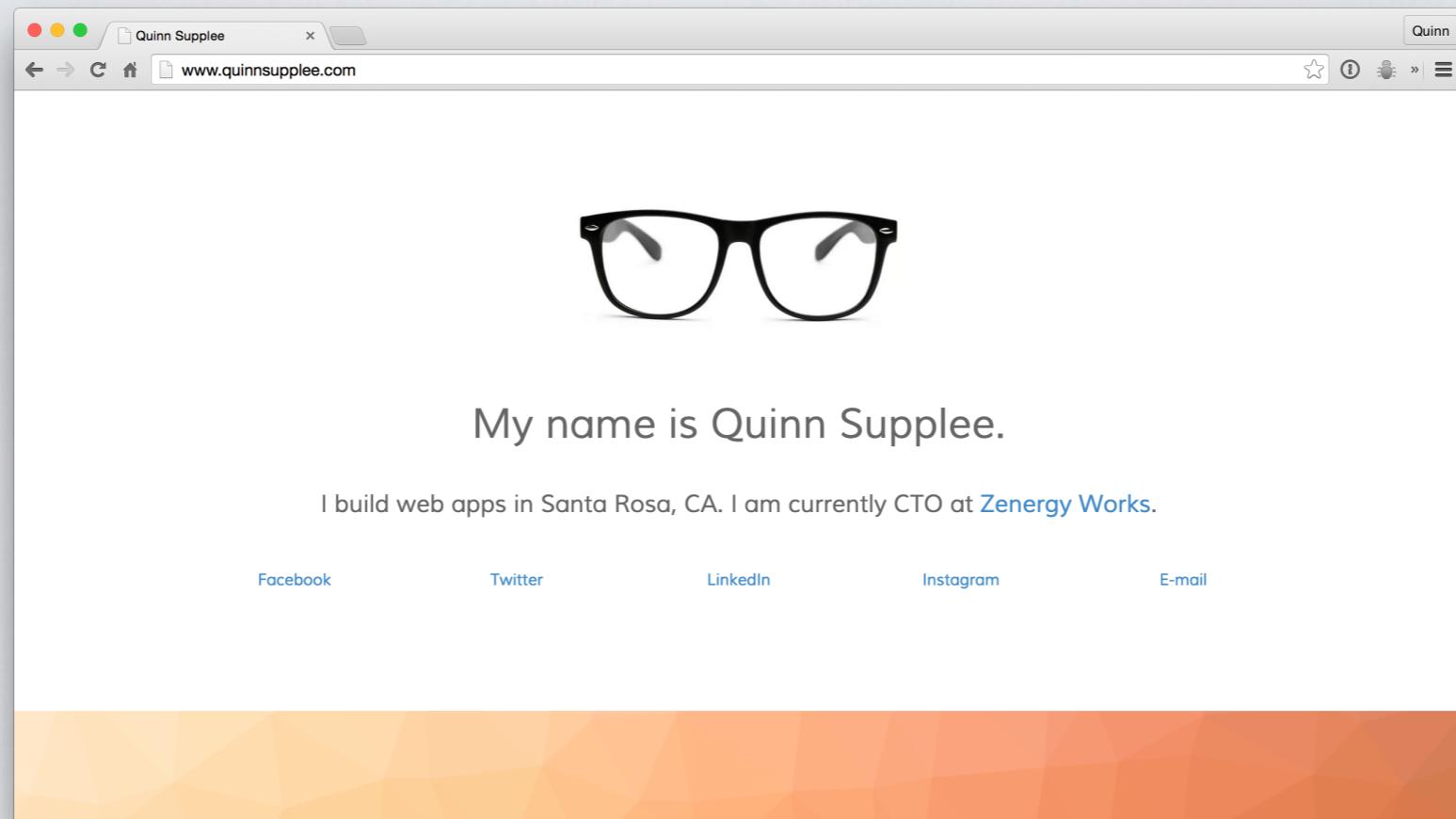
**How To Set Up SSH Keys**

<http://do.co/1IA3brm>

**How To Use SFTP to Securely Transfer Files with a Remote Server**

<http://do.co/1SJg6GI>

How to contact me: [quinn@quinnsupplee.com](mailto:quinn@quinnsupplee.com)



Presentation can be downloaded from:

<http://www.quinnsupplee.com/accidentalsysadmin>