

New Ubuntu 14.04 Server Checklist >
Additional Recommended Steps for... ▼

By: Justin Ellingwood

 Subscribe

Additional Recommended Steps for New Ubuntu 14.04 Servers


371

Posted November 4, 2014  370.5k GETTING STARTED SECURITY SYSTEM TOOLS UBUNTU

Introduction

After setting up the bare minimum configuration for a new server, there are some additional steps that are highly recommended in most cases. In this guide, we'll continue the configuration of our servers by tackling some recommended, but optional procedures.

Prerequisites and Goals

Before you start this guide, you should run through the [Ubuntu 14.04 initial server setup guide](#). This is necessary in order to set up your user accounts, configure privilege elevation with `sudo`, and lock down SSH for security.

Once you have completed the guide above, you can continue with this article. In this guide, we will be focusing on configuring some optional but recommended components. This will involve setting our system up with a firewall, Network Time Protocol synchronization, and a swap files.

Configuring a Basic Firewall

Firewalls provide a basic level of security for your server. These applications are responsible for denying traffic to every port on your server with exceptions for ports/services you have approved. Ubuntu ships with a tool called `ufw` that can be used to configure your firewall policies. Our basic strategy will be to lock down everything that we do not have a good reason to keep open.

Before we enable or reload our firewall, we will create the rules that define the exceptions to our policy. First, we need to create an exception for SSH connections so that we can maintain access for remote administration.

The SSH daemon runs on port 22 by default and `ufw` can implement a rule by name if the default has not been changed. So if you have **not** modified SSH port, you can enable the exception by typing:

```
sudo ufw allow ssh
```

If you have modified the port that the SSH daemon is listening on, you will have to allow it by specifying the actual port number, along with the TCP protocol:

```
sudo ufw allow 4444/tcp
```

This is the bare minimum firewall configuration. It will only allow traffic on your SSH port and all other services will be inaccessible. If you plan on running additional services, you will need to open the firewall at each port required.

If you plan on running a conventional HTTP web server, you will need to allow access to port 80:

```
sudo ufw allow 80/tcp
```

If you plan to run a web server with SSL/TLS enabled, you should allow traffic to that port as well:

```
sudo ufw allow 443/tcp
```

If you need SMTP email enabled, port 25 will need to be opened:

```
sudo ufw allow 25/tcp
```

After you've finished adding the exceptions, you can review your selections by typing:

```
sudo ufw show added
```

If everything looks good, you can enable the firewall by typing:

```
sudo ufw enable
```

You will be asked to confirm your selection, so type "y" if you wish to continue. This will apply the exceptions you made, block all other traffic, and configure your firewall to start automatically at boot.

Remember that you will have to explicitly open the ports for any additional services that you may configure later. For more in-depth information, check out our article on [configuring the ufw firewall](#).

Configure Timezones and Network Time Protocol Synchronization

The next step is to set the localization settings for your server and configure the Network Time Protocol (NTP) synchronization.

The first step will ensure that your server is operating under the correct time zone. The second step will configure your system to synchronize its system clock to the standard time maintained by a global network of NTP servers. This will help prevent some inconsistent behavior that can arise from out-of-sync clocks.

Configure Timezones

Our first step is to set our server's timezone. This is a very simple procedure that can be accomplished by reconfiguring the `tzdata` package:

```
sudo dpkg-reconfigure tzdata
```

You will be presented with a menu system that allows you to select the geographic region of your server:

Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time zones in which they are located.

Geographic area:

- Africa
- America**
- Antarctica
- Australia
- Arctic Ocean
- Asia
- Atlantic Ocean
- Europe
- Indian Ocean
- Pacific Ocean
- System V timezones
- US
- None of the above

<Ok>

<Cancel>

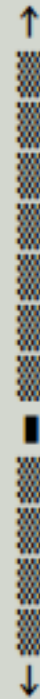
After selecting an area, you will have the ability to choose the specific time zone that is appropriate for your server:

Configuring tzdata

Please select the city or region corresponding to your time zone.

Time zone:

Mazatlan
Menominee
Merida
Metlakatla
Mexico_City
Miquelon
Moncton
Monterrey
Montevideo
Montreal
Montserrat
Nassau
New_York
Nipigon



<Ok>

<Cancel>

Your system will be updated to use the selected timezone, and the results will be printed to the screen:

```
Current default time zone: 'America/New_York'
```

```
Local time is now:      Mon Nov  3 17:00:11 EST 2014.
```

```
Universal Time is now:  Mon Nov  3 22:00:11 UTC 2014.
```

Next, we will move on to configure NTP.

Configure NTP Synchronization

Now that you have your timezone set, we should configure NTP. This will allow your computer to stay in sync with other servers, leading to more predictability in operations that rely on having the correct time.

For NTP synchronization, we will use a service called `ntp`, which we can install from Ubuntu's default repositories:

```
sudo apt-get update
sudo apt-get install ntp
```

This is all that you have to do to set up NTP synchronization on Ubuntu. The daemon will start automatically each boot and will continuously adjust the system time to be in-line with the global NTP servers throughout the day.

Click [here](#) if you wish to learn more about NTP servers.

Create a Swap File

Adding "swap" to a Linux server allows the system to move the less frequently accessed information of a running program from RAM to a location on disk. Accessing data stored on disk is much slower than accessing RAM, but having swap available can often be the difference between your application staying alive and crashing. This is especially useful if you plan to host any databases on your system.

Note

Although swap is generally recommended for systems utilizing traditional spinning hard drives, using swap with SSDs can cause issues with hardware degradation over time. Due to this consideration, we do not recommend enabling swap on DigitalOcean or any other provider that utilizes SSD storage. Doing so can impact the reliability of the underlying hardware for you and your neighbors.

If you need to improve the performance of your server, we recommend upgrading your Droplet. This will lead to better results in general and will decrease the likelihood of contributing to hardware issues that can affect your service.

Advice about the best size for a swap space varies significantly depending on the source consulted. Generally, an amount equal to or double the amount of RAM on your system is a good starting point.

Allocate the space you want to use for your swap file using the `fallocate` utility. For example, if we need a 4 Gigabyte file, we can create a swap file located at `/swapfile` by typing:

```
sudo fallocate -l 4G /swapfile
```

After creating the file, we need to restrict access to the file so that other users or processes cannot see what is written there:

```
sudo chmod 600 /swapfile
```

We now have a file with the correct permissions. To tell our system to format the file for swap, we can type:

```
sudo mkswap /swapfile
```

Now, tell the system it can use the swap file by typing:

```
sudo swapon /swapfile
```

Our system is using the swap file for this session, but we need to modify a system file so that our server will do this automatically at boot. You can do this by typing:

```
sudo sh -c 'echo "/swapfile none swap sw 0 0" >> /etc/fstab'
```

With this addition, your system should use your swap file automatically at each boot.

Where To Go from Here?

You now have a very decent beginning setup for your Linux server. From here, there are quite a few places you can go. First, you may wish to snapshot your server in its current configuration.

Take a Snapshot of your Current Configuration

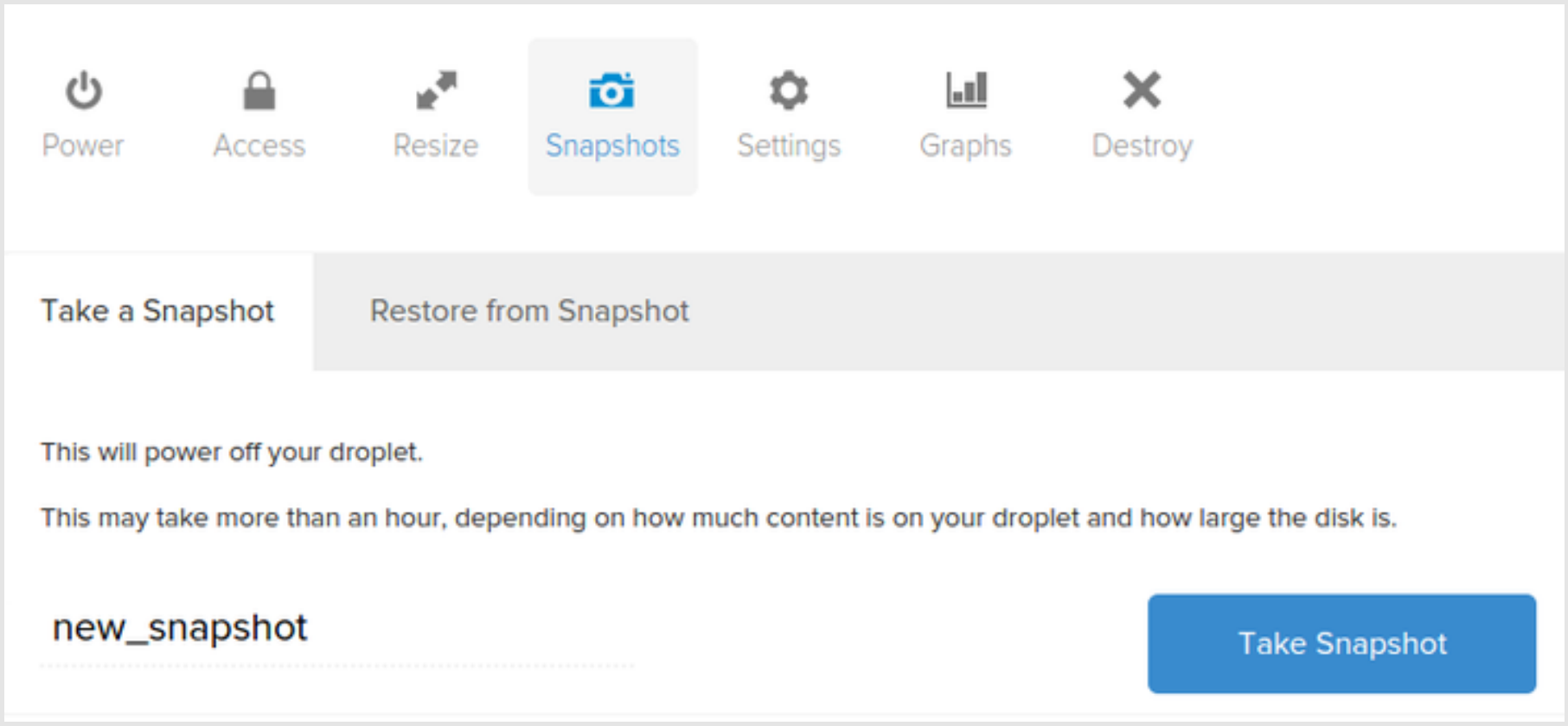
If you are happy with your configuration and wish to use this as a base for future installations, you

can take a snapshot of your server through the DigitalOcean control panel. Starting in October of 2016, snapshots cost \$0.05 per gigabyte per month, based on the amount of utilized space within the filesystem.

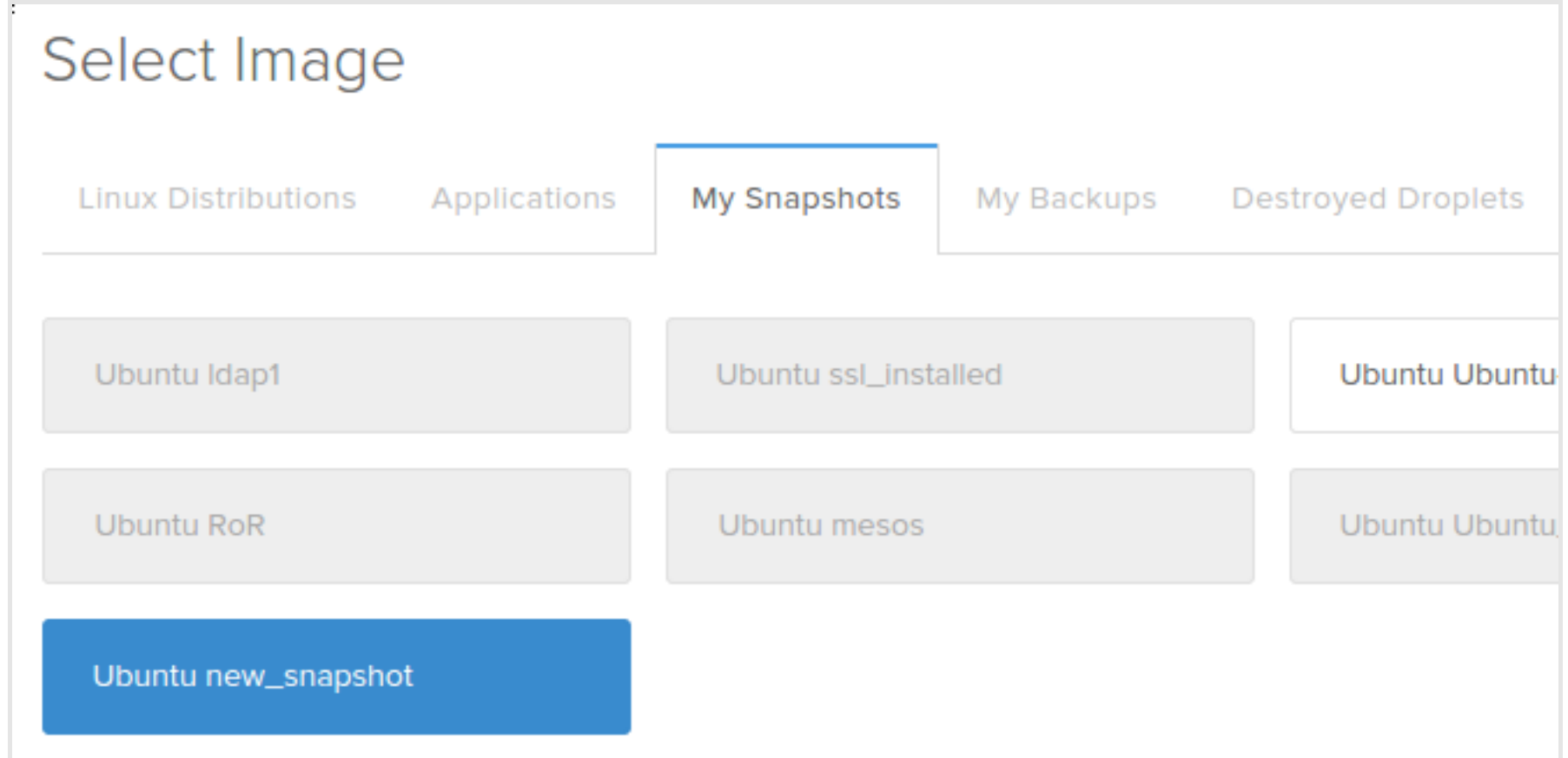
To prepare for the snapshot, shutdown your server from the command line. Although it is possible to take a snapshot of a running system, powering down leads to better guarantees that the filesystem will be consistent:

```
sudo poweroff
```

Now, in the DigitalOcean control panel, you can take a snapshot by visiting the "Snapshots" tab of your server:



After taking your snapshot, you will be able to use that image as a base for future installations by selecting the snapshot from the "My Snapshots" tab for images during the creation process:



Additional Resources and Next Steps

From here, your path depends entirely on what you wish to do with your server. The list of guides below is in no way exhaustive, but represents some of the more common configurations that users turn to next:

- [Setting up a LAMP \(Linux, Apache, MySQL/MariaDB, PHP\) stack](#)
- [Setting up a LEMP \(Linux, Nginx, MySQL/MariaDB, PHP\) stack](#)
- [Installing the WordPress CMS on an Apache web server](#)
- [Installing the WordPress CMS on an Nginx web server](#)
- [Installing the Drupal CMS on an Apache web server](#)
- [Installing Node.js](#)
- [Installing Ruby on Rails and RVM](#)
- [Installing Laravel, a PHP framework](#)
- [Installing Puppet to manage your infrastructure](#)

Conclusion

By this point, you should know how to configure a solid foundation for your new servers.

Hopefully, you also have a good idea for your next steps. Feel free to explore the site for more ideas that you can implement on your server.

By: Justin Ellingwood

 Upvote (371)

 Subscribe

Tutorial Series

New Ubuntu 14.04 Server Checklist

When creating a new Ubuntu 14.04 server, there are some basic steps that you should take to ensure that your server is secure and configured properly. This tutorial series covers connecting to your server and general security best practices, and provides links to articles that will help you start running your own web server or application.

Show Tutorials

Spin up an SSD cloud server in under a minute.

Simple setup. Full root access.
Straightforward pricing.

DEPLOY SERVER



Related Tutorials

[How to Block Unwanted SSH Login Attempts with PyFilter on Ubuntu 16.04](#)

[How to Create an Encrypted File System on a DigitalOcean Block Storage Volume](#)

[How To Host a Website Using Cloudflare and Nginx on Ubuntu 16.04](#)

[Solution Deep Dive: Building a Highly Available Web Application with Web Processing and Stor...](#)

[How To Install and Secure Memcached on CentOS 7](#)

36 Comments

B *I*      



B *I*      



Leave a comment...

Logged in as:
agiansante00



Notify me of replies
to my comment

Comment

^ paulyeohlz November 7, 2014
♡

3 Should it not be `sudo chmod 600 /swapfile` rather than `sudo chown 600 /swapfile` ?

[Reply](#) · [Report](#)

^ jellingwood MOD November 7, 2014
♡

1 Yep, definitely. Thanks for spotting the mistake!

[Reply](#) · [Report](#)

^ lingzhong July 14, 2015
♡

1 [deleted]

[Reply](#) · [Report](#)

^ rgreenjr November 11, 2014
♡

1 Very helpful post, thanks.

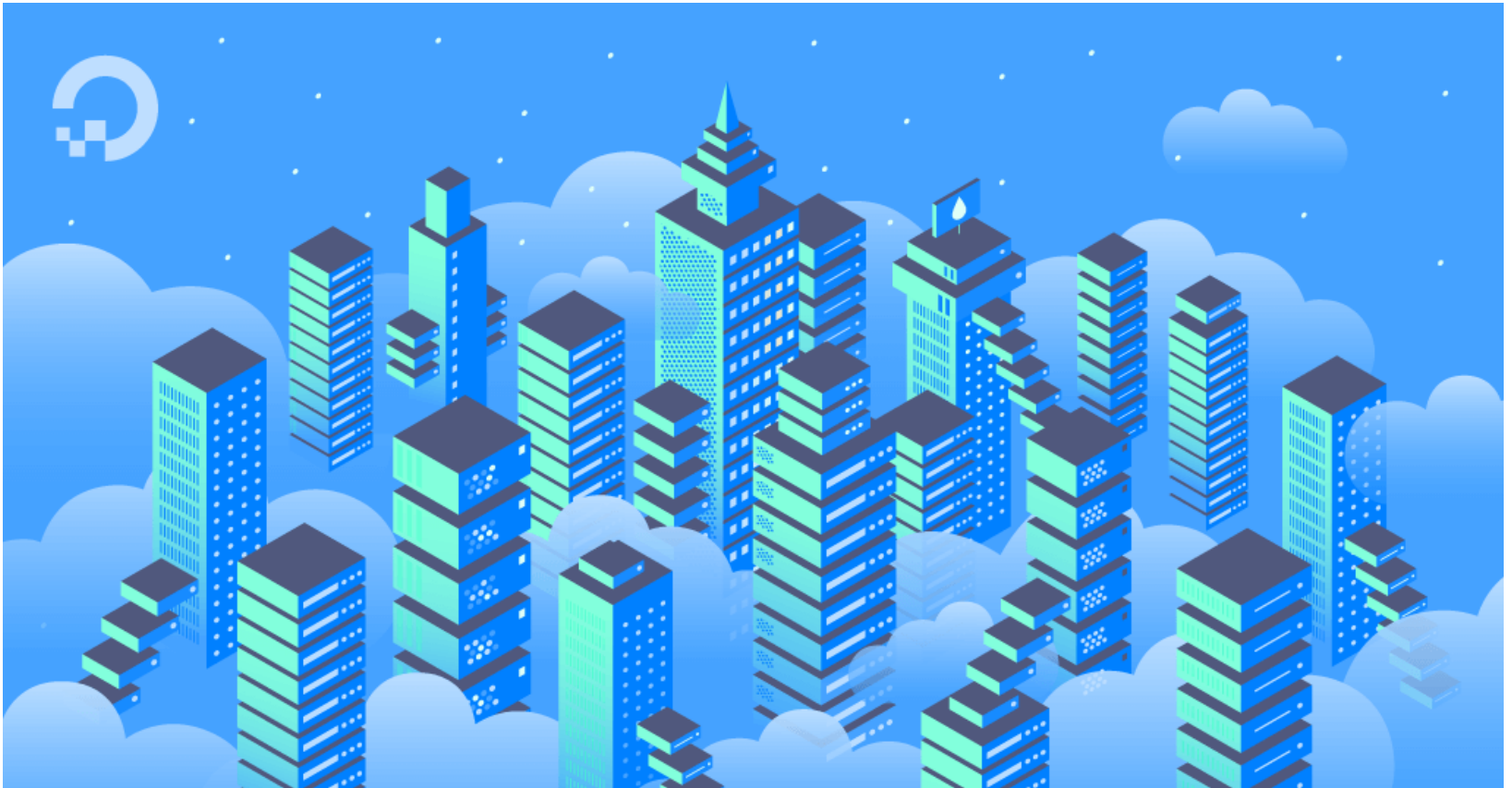
[Reply](#) · [Report](#)

^ SequoiaSemper December 4, 2014
♡

3 If you want to use NTP Synchronization you should enable port 123 in the firewall:

```
sudo ufw allow 123/tcp
```

per <https://www.digitalocean.com/community/tutorials/how-to-set-up-time-synchronization-on-ubuntu-12-04>



How To Set Up Time Synchronization on Ubuntu 12.04

by Etel Sverdlov

This tutorial covers how to set up time synchronization and ensure that servers are synced using the NTP protocol so that all servers are displaying the same time. When a server's time is mismatched, emails sent out from the misconfigured server may arrive 3 minutes in

[Reply](#) · [Report](#)

^ [warwick](#) December 5, 2014

◦ Should that be UDP? As in:

```
sudo ufw allow 123/udp
```

On my server I did this:

```
sudo ufw allow ntp
```

And after:

```
sudo ufw status
```

It shows only 123, not 123/tcp or 123/udp

Does anyone know which one is correct / better?

[Reply](#) · [Report](#)

^ [mavs](#) December 10, 2014

- o Hey, If you check out the specifications for NTP v4 you can find the statement "**The NTP packet is a UDP datagram**". Therefore, 123/udp should be the better choice.

[Reply](#) · [Report](#)

^ [joshwitz](#) December 12, 2014

- 10 Actually, opening port 123 is not necessary at all for NTP. ufw allows all *outgoing* connections by default, and unless you are setting up your own time server for others to ping, incoming can (and should) be closed.

[Reply](#) · [Report](#)

^ [mavs](#) December 12, 2014

- o That's a reasonable point. Thank you!

[Reply](#) · [Report](#)

^ [jonathonbatson](#) December 6, 2014

- o if I have changed my default ssh port .. and added a rule in ufw for the new port, should i then disallow access to the default ssh port ?

[Reply](#) · [Report](#)

^ [kamaln7](#) **MOD** December 6, 2014

- 1 Yes, you don't need port 22 anymore because nothing is using/listening on it.

[Reply](#) · [Report](#)

^ [newbie](#) May 12, 2016

- o @kamaln7

How to disable/deny any port? actually i am using remote server for database and mistakenly enable port 80, 443 etc. now it shows like this after giving deny command

```
adminx@xionic-db:~$ sudo ufw show added
```


Added user rules (see 'ufw status' for running firewall):

```
ufw allow 22
ufw deny 80/tcp
ufw deny 443/tcp
ufw deny 25/tcp
```

is it ok? or im doing something wrong?

[Reply](#) · [Report](#)

 [kamaln7](#) MOD May 12, 2016

 ¹ Hi @newbie! If you run `sudo ufw status numbered`, you should see a list of rules with a number next to each one. For example:

```
Status: active
```

	To	Action	From
	--	-----	----
[1]	22	ALLOW IN	15.15.15.0/24
[2]	80	ALLOW IN	Anywhere

You can then run `sudo ufw delete 2` to delete any rule you want (in this case, rule #2),

[Reply](#) · [Report](#)

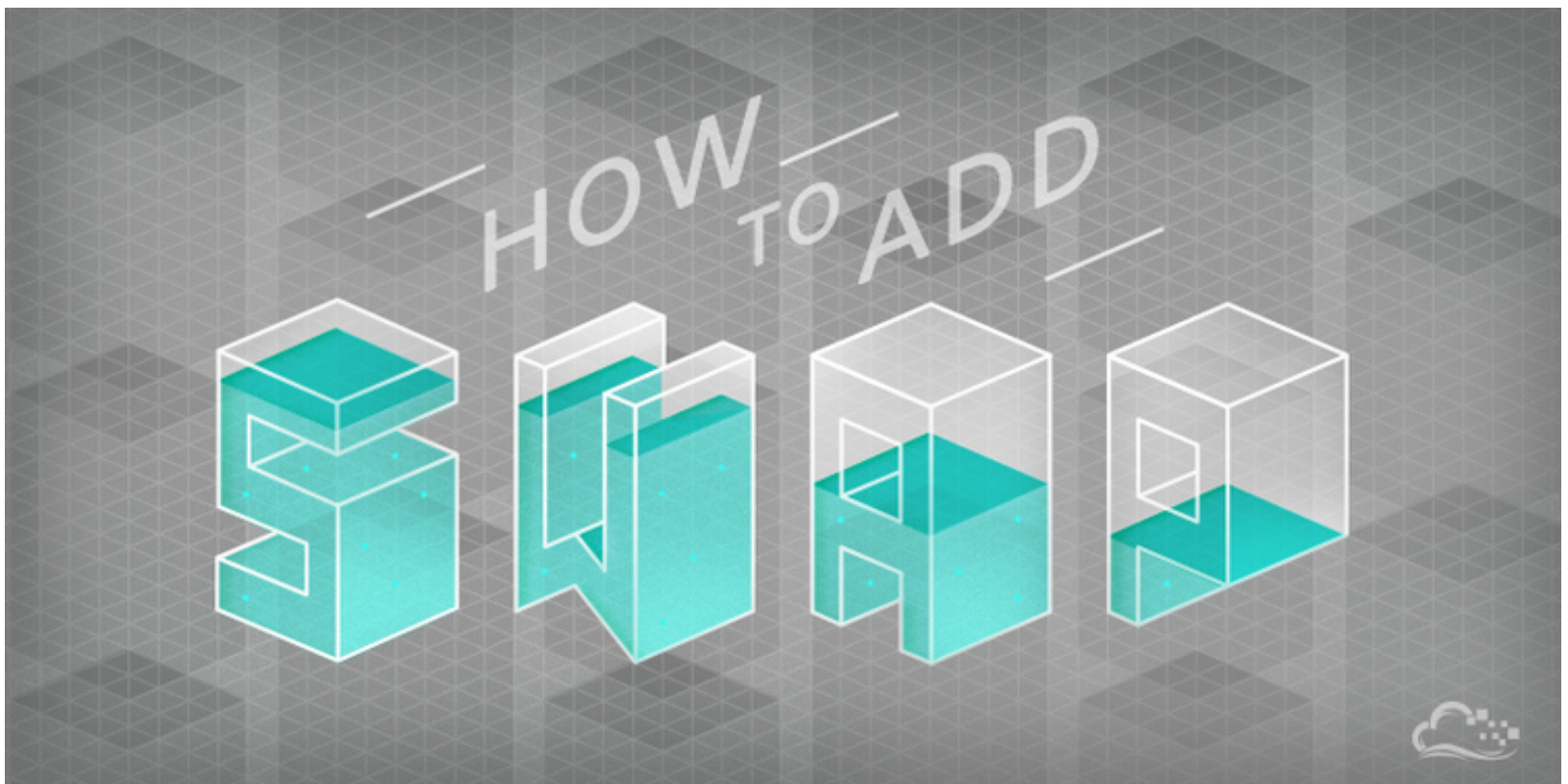
 [frankis](#) December 21, 2014

  [deleted]

[Reply](#) · [Report](#)

 [frankis](#) December 21, 2014

  I recommend also tweaking the swapfile as per this guide from Justin.



How To Add Swap on Ubuntu 14.04

by Justin Ellingwood

Swap space can be used as an "overflow" area for your system when you run out of RAM. The operating system can store data that would normally be kept in RAM on the hard drive in a specially formatted file. In this guide, we'll demonstrate how to create and use one of

[Reply](#) · [Report](#)

^ [chihchengyuan](#) December 29, 2014



◦ I get this, what happened and how to solve it?

```
sudo dpkg-reconfigure tzdata
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
locale: Cannot set LC_CTYPE to default locale: No such file or directory
locale: Cannot set LC_ALL to default locale: No such file or directory
/usr/bin/locale: Cannot set LC_CTYPE to default locale: No such file or
directory
/usr/bin/locale: Cannot set LC_ALL to default locale: No such file or
```


directory

Current default time zone: 'Asia/Taipei'

Local time is now: Mon Dec 29 14:21:43 CST 2014.

Universal Time is now: Mon Dec 29 06:21:43 UTC 2014.

[Reply](#) · [Report](#)

^ [dannl](#) February 4, 2015



1. as root, i added the line `LANG="en_US.UTF-8"` to `/etc/default/locale`
2. exited the ssh connection because `. /etc/default/locale` didn't seem to reload it for the current session
3. re-established the ssh connection
4. ran `locale` to confirm that the errors mentioned above were gone

[Reply](#) · [Report](#)

^ [idiazroncero](#) January 15, 2015



- Some people might find (I did) imposible to create a swap file if running on a VPS.

Command `swapon` returns an error: "swapon: swap: swapon failed: Operation not permitted"

If your "swapon" is not responding, this might be the cause.

I am running Ubuntu 14.04 on a VPS provided by OVH, an european-based international ISP. Turns out that some of the virtualized operating systems ISPs use are modified to automatically provide swap memory (in my case, swap = 1/8 RAM). They also prevent root user from creating its own swap memory files.

When I run `free -m` i can see I had 128MB of swap memory from the very beginning.

total	used	free	shared	buffers	cached
-------	------	------	--------	---------	--------

Mem: 1024 325 698 2 0 285

-/+ buffers/cache: 39 984

Swap: 128 0 128

This is an obvious shortcoming.

[Reply](#) · [Report](#)

^ [autorun](#) February 10, 2015
♡

1 Hello,

I'm thinking about using your great services! However, I wonder if this guide trilogy is almost enough to deploy a normal local store Wordpress website, (just talking about the server mantainment). I would also use Cloudflare to stop spam and ddos attacks.

I mean, apart from updating packages, I'm worried about security, and any other kind of configuration not explained here.

Are this firewall settings enough for the most common attacks?

Should I take a special consideration on admin pages apart from SSL certificates?

Because, the fact that having my own VPS is great, and **I know the Linux basics and I think I can configure my server! but I'm not server admin**, so I'm not sure if my website will be a lot more naked than other managed servers (bluehost, etc)

[Reply](#) · [Report](#)

^ [mktdom](#) February 23, 2015
♡

o Hi,

I get a "Operation not permitted" on swapon, does it important ? how to fix it

```
me@vps:~$ sudo fallocate -l 4G /swapfile
```

```
me@vps:~$ sudo chmod 600 /swapfile
```

```
me@vps:~$ sudo mkswap /swapfile
```

```
Setting up swapspace version 1, size = 4194300 KiB
```

```
no label, UUID=468432d3-ea0d-4502-b326-ac101044c914
```

```
me@vps:~$ sudo swapon /swapfile
```

```
swapon: /swapfile: swapon failed: Operation not permitted
```

Thkx

[Reply](#) · [Report](#)

^ [jasons](#) March 13, 2015
♡

o [@mktdom](#) using 14.04 x64 on digital ocean micro droplet i had no problem making the 4gb swap file.

[Reply](#) · [Report](#)

^ [iAbdulaziz](#) June 26, 2015
♡

o good tips, thanks

[Reply](#) · [Report](#)

^ [MarkBoris](#) *July 6, 2015*

- So when you first fire up a server from Digital Ocean the firewall is not enabled until you run `sudo ufw enable`. Correct?

[Reply](#) · [Report](#)

^ [jellingwood](#) **MOD** *July 13, 2015*

- @MarkBoris: Yep, that's right.

[Reply](#) · [Report](#)

^ [adelinghanayem](#) *October 6, 2015*

- I really don't what to say, this really can't be written or described better !

[Reply](#) · [Report](#)

^ [aral](#) *October 14, 2015*

- 3 I'd highly recommend that you set the timezone of your server to UTC (under 'configure timezones'). To do so, after:

```
sudo dpkg-reconfigure tzdata
```

Scroll down to the bottom of the list and select 'None of the above'. Then select 'UTC'.

This should result in output similar to:

```
Current default time zone: 'Etc/UTC'
Local time is now:        Wed Oct 14 09:55:25 UTC 2015.
Universal Time is now:    Wed Oct 14 09:55:25 UTC 2015.
```

For more information on why you should do this, see [The Worst Server Setup Mistake You Can Make... Setting the timezone to anything other than UTC.](#)

[Reply](#) · [Report](#)

^ [ddulic](#) *October 19, 2015*

- Thank you for the useful tips :)

[Reply](#) · [Report](#)

^ [domcamilliere](#) *October 29, 2015*



- So I used the command `sudo chmod 600 /swapfile` and received the following error:

```
swapon: /swapfile: read swap header failed
```

What's up with that? I then proceeded to type in `sudo mkswap /swapfile` THEN entered

```
sudo chmod 600 /swapfile
```

It makes more sense to me to create the swapfile first, then set permissions on it; no? After switching the order of the commands, I received no error. Perhaps my understanding is incorrect? I'd love some clarification.

Thanks!

[Reply](#) · [Report](#)

^ [domcamilliere](#) *October 29, 2015*



- So, none of this even worked initially. I ran `cat /proc/meminfo` and realized after everything a swap file was never created. So I followed the steps once more and there was my swap file! (although way too large and unnecessary as I used the 4G as mentioned in your doc). I have a 512mb RAM droplet and read Ubuntu's documentation stating that a swap file of more than double your RAM is really a waste (even with an ssd like our droplets utilize). So I ended up deleting my swapfile and making a new one with the proper 1G of space with consideration for the RAM I'm working with. Thanks for the great doc!

[Reply](#) · [Report](#)

^ [ajjonez](#) *November 2, 2015*



- is this tutorial necessary when one uses the one-click install for wordpress?

[Reply](#) · [Report](#)

^ [digitaldragon](#) *November 15, 2015*



- Did anyone else run into an issue where ufw is not enabled at boot?

For example, after reboot I get the following

```
xxx@xxx:~$ sudo ufw status
Status: inactive
```

I also found [this post](#) which helped me to enable ufw at boot

[Reply](#) · [Report](#)

^ [vs](#) December 22, 2015

- Why do you have to do `sudo sh -c 'echo "/swapfile none swap sw 0 0" >> /etc/fstab'` instead of just `sudo echo "/swapfile none swap sw 0 0" >> /etc/fstab`

[Reply](#) · [Report](#)

^ [jellingwood](#) **MOD** December 23, 2015

- @[vs](#): We wrap the command in `sh -c` because `sudo` elevates the permissions of your command, but it does not increase your privileges for the shell redirection (the `>>` that appends the output of our command to the `/etc/fstab` file). To get around this, we use `sudo` to give an entire shell instance (initiated by the `sh -c` command) elevated privileges. Since the entire subshell session has increased permissions, the redirection will work correctly.

An alternative is to not use redirection at all. We could instead pipe a regular `echo` statement into the `tee` command, which splits and forwards output to multiple locations. We can add the `-a` flag to append rather than overwrite the content at the destination. Preceding the `tee` command with `sudo` gives us the permissions to write to the protected `/etc/fstab` file:

```
$ echo "/swapfile none swap sw 0 0" | sudo tee -a /etc/fstab
```

This would produce the same results as the first command, but might be easier to remember.

[Reply](#) · [Report](#)


Load More Comments



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



Copyright © 2018 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#) 

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#) [Shop](#)

SCROLL TO TOP