TurtleWolf / docker-run-it-ubuntu-bin-bash

haha, i'm sure you meant condensed, but the image of you condescending four chapters into a list is pretty hilarious... "what do you think you are, a book? you'll end up fitting on one page, just wait and see..." Edit

Manage topics

52 commits | 1 branch | 0 releases | 1 contributor | MIT

Branch: master ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

TurtleWolf 3rd draft, ready for library printing — Latest commit b2abe13 40 minutes ago

captured_Images	2nd draft of outline, code blocks for each section	18 hours ago
.env	Update .env	22 days ago
.gitignore	Initial commit	23 days ago
Dockerfile	added docker demos to edit	22 days ago
LICENSE	Initial commit	23 days ago
README.md	3rd draft, ready for library printing	40 minutes ago
UbuntuServersU1804.pdf	updating notes on section of chapter 2	6 days ago
docker-compose.yml	Update docker-compose.yml	22 days ago

README.md

https://www.youtube.com/watch?v=Cvrqmq9A3tA&index=1&list=PLETG2T1KvnipSA8vKmzju_unzl44jeyCa

https://www.twitch.tv/videos/347820755

L inux on

W indows in

D ocker LAMP

`APACHE` can be replaced with `NGINX`
`MySQL` should be replaced with `MariaDB`

Show info like number of containers, etc

```
$ docker info
```

List all containers (Even if not running)

```
$ docker container ls -a
```

Get logs (Use name or ID)

```
$ docker container logs [NAME]
```

Stop all running containers

```
$ docker container stop $(docker ps -aq)
```

To remove a running container use force(-f)

```
$ docker container rm -f [ID]
```

Remove all containers

```
$ docker rm $(docker ps -aq)
```

Remove all images

```
$ docker rmi $(docker images -a -q)
```

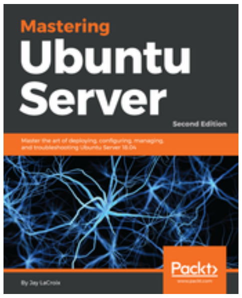
`$ docker run --name u1804 -dit -p 8080:80 ubuntu:18.04 //bin/bash`

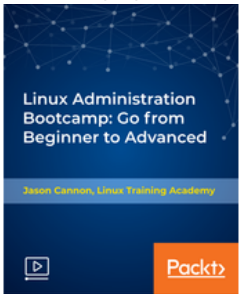
detached interactive terminal on port 8080 named u1804

```
$ docker stop u1804
$ docker start u1804
```

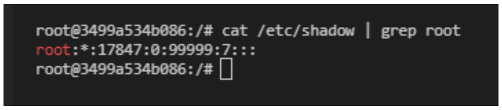
`$ docker attach u1804`

u1804 represents containerID
root@u1804:/# `ls -l`
root@u1804:/# `apt-get update`
root@u1804:/# `apt-get install sudo`
root@u1804:/# `apt-get install -y curl`
root@u1804:/# `curl https://www.google.com`





Chapter 2. Managing Users



```
root@3499a534b086:/# cat /etc/shadow | grep root
root:*:17847:0:99999:7:::
root@3499a534b086:/#
```

Understanding when to use `root`

```
apt install tmux
E: Could not open lock file /var/lib/dpkg/lock - open (13: Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root?
sudo apt install tmux
```

Creating and removing `users`

```
sudo useradd -d /home/jdoe -m jdoe
ls -l /home
nano /usr/sbin/adduser
sudo userdel dscully
ls -l /home
sudo mv /home/dscully /store/file_archive
sudo mkdir -p /store/file_archive
sudo userdel -r dscully
sudo rm -r /home/dscully
```

sudo rm -r / *(space)* **home/dscully**

Understanding the `/etc/passwd` & `/etc/shadow files`

```
cat /etc/passwd
sudo cat /etc/shadow
sudo cat /etc/shadow | grep root
mulder:$6$TPxx8Z.:16809:0:99999:7:::
sudo passwd -S <username>
```

root@u1804:/# `adduser jane_doe`

root@u1804:/# `cat /etc/shadow | grep root`

any password ?

```
user: * :password changed:7days between changes:max:warning:disable:8thN:9?
```

...

encrypted password

```
user: x :password changed:7days between changes:max:warning:disable:8thN:9?
```

...

lockout login

```
user: ! :password changed:7days between changes:max:warning:disable:8thN:9?
```

...

add a user to `sudo` as a secondary group

jane_doe@u1804:~$ `sudo usermod -aG sudo jane_doe`

...

switch user

root@u1804:/# `su - jane_doe` ...

list all files in long form

jane_doe@u1804:~$ `ls -al`

...

add user group

jane_doe@u1804:~$ `sudo groupadd admins`

...

modify secondary group to include user

jane_doe@u1804:~$ `sudo usermod -aG admins jane_doe`

...

lock password - will not affect SSH (see ch 15)

jane_doe@u1804:~$ `sudo passwd -l root`

jane_doe@u1804:~$ `sudo cat /etc/shadow | grep root`

root@u1804:/# `su - root`

...

unlock password

jane_doe@u1804:~$ `sudo passwd -u <username>`

...

/etc/shadow

jane_doe@u1804:~$ `sudo chage -l root`

...

Distributing default configuration files with `/etc/skel`

```
ls -la /etc/skel
```

force password change - should move this into `/etc/skel` **?**, for default configuration (see ch 1)
jane_doe@u1804:~$ `sudo chage -d 0 <username>`
...

Switching users

```
sudo passwd
sudo su -
su - <username>
sudo su - <username>
```

Pluggable Authentication Module (PAM):
jane_doe@u1804:~$ `sudo apt install libpam-cracklib`
...
install Nano
jane_doe@u1804:~$ `sudo apt install nano`
...
configure password requirements in PAM
jane_doe@u1804:~$ `sudo nano /etc/pam.d/common-password`
! (use a 2nd TTY to prevent lock out)
...

Managing groups

```
ls -l
-rw-r--r-- 1 root bind 490 2013-04-15 22:05 named.conf
cat /etc/group
sudo groupadd admins
sudo groupdel admins
sudo usermod -aG admins myuser
sudo usermod -g <group-name> <username>
man usermod
sudo usermod -d /home/jsmith jdoe -m
sudo usermod -l jsmith jdoe
sudo gpasswd -d <username> <grouptoremove>
sudo gpasswd -a <username> <group>
```

...
add a user to `sudo` as a secondary group
jane_doe@u1804:~$ `sudo usermod -aG sudo <username>`
jane_doe@u1804:~$ `sudo usermod -aG sudo jane_doe`
may use another group (such as `wheel`)
...
configure `sudo` group or user access
jane_doe@u1804:~$ `sudo visudo`
checks to make sure your changes follow the correct syntax `/etc/sudoers` `jane_doe ALL=(ALL:ALL) ALL`
`charlie ubuntu-server=(jane_doe:admins) /usr/bin/apt,/usr/sbin/reboot,/usr/sbin/shutdown`
(root or username) TTY IP=(USER:GROUP) COMMANDS
! It's always a good idea to use full paths when editing `sudo` command permissions
...

Managing passwords and password policies

`lock` password per `<username>`

jane_doe@u1804:~$ `sudo passwd -l <username>`

`unlock` password per `<username>`

jane_doe@u1804:~$ `sudo passwd -u <username>`

...

`list` expiration of a user's `password`

jane_doe@u1804:~$ `sudo chage -l <username>`

(set to zero would force a password change)

jane_doe@u1804:~$ `sudo chage -d 0 <username>`

(review changes)

jane_doe@u1804:~$ `sudo chage -l <username>`

`expiration` of a user's `password` *(Maximum days until a change is required)*

jane_doe@u1804:~$ `sudo chage -M 90 <username>`

`expiration` of a user's `password` *(minimum days until a change is required)*

jane_doe@u1804:~$ `sudo chage -m 5 dscully`

...

`install` Pluggable Authentication Module (`PAM`)

jane_doe@u1804:~$ `sudo apt install libpam-cracklib`

...

`edit` ..*(/ `etc` / `pam.d` / `common-password`)*

jane_doe@u1804:~$ `sudo nano /etc/pam.d/common-password`

```
password required pam_pwhistory.so remember=99 use_authok
```

`difference` *(at least three characters have to be different)*

```
difok=3
```

`obscure` *(prevents simple passwords from being used)*

```
obscure
```

Configuring administrator access with sudo

`modify` secondary `Group` *to include* **`user`**

jane_doe@u1804:~$ `sudo usermod -aG sudo <username>`

...

nano `save changes`

```
Ctrl + W
```

nano `exit editor`

```
Ctrl + X
```

...

configure `visudo` *default `Editor` to* **`vim`**

jane_doe@u1804:~$ `sudo EDITOR=vim visudo`

`etc / sudoers`

```
Object type : User : Group : Other's
-dl : rwx : rwx : rwx
```

...

`sudo group`

```
%sudo ALL=(ALL:ALL) ALL
```

`user`

```
root ALL=(ALL:ALL) ALL
```

...

```
TTY : User : Group : Command
```

note that it's best to use full paths

charlie could run these : **`commands`**

```
charlie ALL=(ALL:ALL) /usr/sbin/reboot,/usr/sbin/shutdown
```

but not others

```
Sorry, user charlie is not allowed to execute '/usr/bin/apt install tmux' as root on ubuntu-server.
```

limited to certain **`terminal`**

```
charlie ubuntu-server=(ALL:ALL) /usr/bin/apt
```

restrict **`user`** *&* **`group`** *options*

```
charlie ubuntu-server= /usr/bin/apt
```

restrict to certain **`( user : group )`**

```
charlie ubuntu-server=(dscully:admins) ALL
```

Setting permissions on files and directories

```
ls -l
```

```
  -rw-rw-rw- 1 doctor doctor   5          Jan 11   12:52 welcome
  -rw-r--r--    1 root     root       665       Feb 19   2014 profile
  -rwxr-xr-x 1 dalek    dalek      35125   Nov  7    2013 exterminate
```

```
-rw-rw-rw- 1 doctor doctor 5 Jan 11 12:52 welcome
-rwxr-xr-x 1 dalek dalek 35125 Nov 7 2013 exterminate
```

```
  -rw-r--r-- 1   sue   accounting      35125  Nov  7  2013 budget.txt
  drwxr-x--- 1   bob   sales           35125  Nov  7  2013 annual_projects
```

```
chmod o-r budget.txt
chmod o-r /home/sue/budget.txt
chmod 770 -R mydir
```

```
  `ls -l`
  `-rw-rw-rw- 1 doctor doctor    5 Jan 11 12:52 welcome`
  `-rwxr-xr-x 1 dalek dalek      35125 Nov  7  2013 exterminate`
```

```
sudo chown sue myfile.txt
sudo chown -R sue mydir
sudo chown sue:sales myfile.txt
# sudo chown sales myfile.txt
```

remove `read` from file permissions for `other's`

jane_doe@u1804:~$ `sudo chmod o-r /home/sue/budget.txt`

...

octal permission patterns

jane_doe@u1804:~$ `chmod 600 filename.txt` (same as) `chmod -rw------- filename.txt`
jane_doe@u1804:~$ `chmod 740 filename.txt` (same as) `chmod -rwxr----- filename.txt`
jane_doe@u1804:~$ `chmod 770 filename.txt` (same as) `chmod -rwxrwx--- filename.txt`
jane_doe@u1804:~$ `chmod 770 -R dir_name` (recursive directories)

...

change ownership of directory recursively

jane_doe@u1804:~$ `sudo chown -R jane_doe:admins dir_name`

...

change group ownership

jane_doe@u1804:~$ `sudo chgrp sales myfile.txt`

Q&A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

`customize TTY prompt`

root@u1804:/# `echo 'export PS1="[\u@\h \w]\$ "' >> ~/.bash_profile`

```
root@u1804:/# nano ~/.bash_profile
```

```
root@u1804:/# exit
```

Chapter 3. Storage Volumes

Understanding the Linux filesystem

`/`
The beginning of the **filesystem**, all directories are underneath this

`/home`
User home directories

`/root`
The home directory for `root` (**root** doesn't have a directory underneath `/home`)

`/media`
For removable media, such as flash drives

`/mnt`
For volumes that are intended to stay mounted for a while

`/opt`
Additional software packages (some programs are installed here, not as common)

`/bin`
Essential user binaries (ls, cp, and so on)

`/proc`
Virtual filesystem for OS-level components

`/usr/bin`
A majority of user commands

`/usr/lib`
Libraries

`/var/log`
Log files

Managing Storage Volumes

```
apt install tmux
apt install tmux
```

Using symbolic and hard links

hard link file1 to file3

```
jane_doe@u1804:~$ ln file1 file3
```

list inode number

```
jane_doe@u1804:~$ ls -i
```

remove file3

```
jane_doe@u1804:~$ rm file3
```

symlink file1 to file3

```
jane_doe@u1804:~$ ln -s file1 file3
```

Viewing disk usage

disk filesystem in **human readable**
jane_doe@u1804:~$ `df -h`
will show available `cyber` space

disk filesystem inodes
jane_doe@u1804:~$ `df -i`
shows available `inodes`

disk usage
jane_doe@u1804:~$ `du -hsc *`
will show `disk usage` in `human` readable, `summary` of `current` working directory total

install **NCurses Disk Usage**
jane_doe@u1804:~$ `sudo apt install ncdu`

disk usage
jane_doe@u1804:~$ `ncu -x`
`-x` limit to the current filesystem
during interface; `d` would delete

Adding additional storage volumes

disk functions *-list* jane_doe@u1804:~$ `sudo fdisk -l`
utility for `listing`, `creating` or `deleting` disk partions

follow display messages
jane_doe@u1804:~$ `dmesg --follow`
When done, press `Ctrl + C` on your keyboard:

list block devices
jane_doe@u1804:~$ `lsblk`
`/dev/sda`
`/dev/sdb`
`/dev/sdc`

adding a new volume
to `/etc/fstab` file

disk functions
jane_doe@u1804:~$ `sudo fdisk /dev/sdb` ..*(volume path)*
`m` *for menu*
`n` *new partition*
`enter` *default partion number*
`1G` *partion size*
`w` *write changes*
`enter` *save changes*

Partitioning and formatting volumes

disk function ..*(utility again)*
jane_doe@u1804:~$ `sudo fdisk -l`
review added disk partion
(***or to try again***)
jane_doe@u1804:~$ `sudo fdisk`
`g` *new **GPT** layout*
`o` *new MBR layout*

disk format *(partition `ext4`)*
jane_doe@u1804:~$ `sudo mkfs.ext4 /dev/sdb1` ..*(volume path)*
or
disk format *(partition `xfs`)*
jane_doe@u1804:~$ `sudo mfs.xfs /dev/sdb1` ..*(volume path)*

disk function *(review)*

jane_doe@u1804:~$ `sudo fdisk -l`

make directory

jane_doe@u1804:~$ `sudo mkdir /mnt/vol1` ..*(volume path)*

Mounting and unmounting volumes

mount device to directory

jane_doe@u1804:~$ `sudo mount /dev/sdb1 /mnt/vol1` ..*(volume path)*

mount device to directory with `type` option ..*(usually un-necessary)*

jane_doe@u1804:~$ `sudo mount /dev/sdb1 -t ext4 /mnt/vol1` ..*(volume path)*

...

unmount device

jane_doe@u1804:~$ `sudo umount /mnt/vol1`

disk filesystem in **human readable** ..*(confirm unmounted)*

jane_doe@u1804:~$ `df -h`

Understanding the /etc/fstab file

block identification, UUID ..*(/ etc / f stab)*

jane_doe@u1804:~$ `blkid`

...

make new directory for extra storage ..*(/ mnt / extra_storage)*

jane_doe@u1804:~$ `sudo mkdir /mnt/extra_storage`

...

edit `/etc/fstab`

jane_doe@u1804:~$ `sudo nano /etc/fstab`

...

```
UUID=e51bcc9e-45dd-45c7 /mnt/extra_storage ext4 rw,auto 0 0
```

...

mounting volume (with **`auto`**)

jane_doe@u1804:~$ `sudo mount -a`

...

```
UUID=e51bcc9e-45dd-45c7 /mnt/ext_disk ext4 rw,noauto 0 0
```

...

mounting an external disk (with **`noauto`**) perhaps per back-up

jane_doe@u1804:~$ `sudo mount /mnt/ext_disk`

...

list everything that is mounted

jane_doe@u1804:~$ `mount`

Managing swap

SWAP-file

swap volume (with **`auto`**)

jane_doe@u1804:~$ `sudo swapon -a`

don't forget to edit `fstab`

...

`/swapfile none swap sw 0 0`

...

check memory

jane_doe@u1804:~$ `free -m`

...

file allocate

jane_doe@u1804:~$ `sudo fallocate -l 4G /swapfile`

creates a 4 gigabyte file

...

make swap

jane_doe@u1804:~$ `sudo mkswap /swapfile`

makes it the swap file

...

don't forget to edit **/** `etc` **/** `f stab`

`/swapfile none swap sw 0 0` **Activate SwapFile** (__ with **`auto`**)

jane_doe@u1804:~$ `sudo swapon -a`

LVM

Utilizing LVM volumes

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

`apt install tmux`

check if **lvm2** *is installed*

jane_doe@u1804:~$ `dpkg -s lvm2 | grep status`

should return `install ok installed` *if it is installed already*

...

install lvm2 *(Logical Volume Management)*

jane_doe@u1804:~$ `sudo apt install lvm2`

...

disk functions

jane_doe@u1804:~$ `sudo fdisk -l`

should list partions

...

pvcreate *(create physical volumes)*

jane_doe@u1804:~$

```
    sudo pvcreate /dev/sdb
    sudo pvcreate /dev/sdc
    sudo pvcreate /dev/sdd
    sudo pvcreate /dev/sde
```

display Physical Volumes

```
jane_doe@u1804:~$ sudo pvdisplay
```

...

create Volume Group

```
jane_doe@u1804:~$ sudo vgcreate vg-test /dev/sdb1
```
..*(volume path)*

...

display Volume Groups

```
jane_doe@u1804:~$ vgdisplay
```

...

create Logical Volume `-n` *name* , `-L` *size ?* , *group name* ,

```
jane_doe@u1804:~$ sudo lvcreate -n myvol1 -L 10g vg-test
```

...

display Logical Volumes

```
jane_doe@u1804:~$ sudo lvdisplay
```

...

make file system , *(format logical volume)*

```
jane_doe@u1804:~$ sudo mkfs.ext4 /dev/vg-test/myvol1
```

...

mount device *to* **directory**

```
jane_doe@u1804:~$ sudo mount /dev/vg-test/myvol1 /mnt/lvm/myvol1
```

...

disk filesystem in **human readable** ..*(confirm volume is mounted & it's size)*

```
jane_doe@u1804:~$ df -h
```

...

Extend Logical Volume *(use the remaining space)*

```
jane_doe@u1804:~$ sudo lvextend -n /dev/vg-test/myvol1 -l +100%FREE
```

(should return)

```
 Logical volume vg-test/myvol1 successfully resized.
```

...

disk filesystem in **human readable** ..*(confirm, still need to resize file system)*

```
jane_doe@u1804:~$ df -h
```

...

Resize File-System ..*(ext4)*

```
jane_doe@u1804:~$ sudo resize2fs /dev/mapper/vg--test-myvol1
```

(should return)

```
 The filesystem on /dev/mapper/vg--test-myvol1 is now 5241856 (4k) blocks long.
```

...

disk filesystem in **human readable** ..*(added space now usable)*

```
jane_doe@u1804:~$ df -h
```

...

Extend Volume Group *(add additonal volumes to group)*

```
jane_doe@u1804:~$
```

```
  sudo vgextend vg-test /dev/sdc
  sudo vgextend vg-test /dev/sdd
  sudo vgextend vg-test /dev/sde
```

(should return)

```
 Volume group "vg-test" successfully extended
```

...

display Physical Volumes *(confirm additional physical volumes attached)*

```
jane_doe@u1804:~$ sudo pvdisplay
```

...

Extend Logical Volume *(extend logical volume 10 gigabytes)*

```
jane_doe@u1804:~$ sudo lvextend -L+10g /dev/vg-test/myvol1
```

...

resize file-system *(make free space available to filesystem)*

```
jane_doe@u1804:~$ sudo resize2fs /dev/vg-test/myvol1
```

create Logical Volume `-s` *snapshot* , `-n` *name* , `-L` *maximu size ? , group name* **/** *volume ,*

jane_doe@u1804:~$ `sudo lvcreate -s -n mysnapshot -L 4g vg-test/myvol1`

(should return)

```
Logical volume "mysnapshot" created.
```

...

logical volume size *(monitor it's size)*

jane_doe@u1804:~$ `lvs`

...

logical volume convert

jane_doe@u1804:~$ `sudo lvconvert --merge vg-test/mysnapshot`

(should return)

```

  Merging of volume mysnapshot started.
  myvol1: Merged: 100.0%`

```

...

logical volume size *(recheck)*

jane_doe@u1804:~$ `lvs`

...

remove logical volume

jane_doe@u1804:~$ `sudo lvremove vg-test/myvol1`

...

remove logical group

jane_doe@u1804:~$ `sudo vgremove vg-test`

Understanding RAID

RAID - Redundant Array of Inexpensive Disks

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

disk functions *-list*

jane_doe@u1804:~$ `sudo fdisk -l`

(one is hardware, multilple is software)

...

Multiple Disk And Disk Administration

jane_doe@u1804:~$ `mdadm`

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 4. Networks

Host Name

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

hostname

jane_doe@u1804:~$ `hostname`

...

Host Name Control - *set host name*

jane_doe@u1804:~$ `sudo hostnamectl set-hostname dev2.mynetwork.org`

...

concatenate / `etc` **/** `hostname`)_

jane_doe@u1804:~$ `cat /etc/hostname`

...

edit / `etc` **/** `hostname`) - *(previous to 15.04, edit maunually)*

jane_doe@u1804:~$ `edit /etc/hostname`

...

```
unable to resolve host dev.mynetwork.org
```

...

edit / `etc` **/** `hosts`) - *(edit maunually)*

jane_doe@u1804:~$ `edit /etc/hosts`

managing Network Interfaces

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

currently assigned IP address

jane_doe@u1804:~$ `ip addr show`

(or shortened to)

jane_doe@u1804:~$ `ip a`

...

state of interface *(toggling up & down)*

jane_doe@u1804:~$

```
sudo ip link set enp0s3 down
sudo ip link set enp0s3 up
```

_older systems would _ **edit /** `etc` **/** `udev` **/** `rules.d` **/** `70-persistent-net-rules`
jane_doe@u1804:~$ `cat /etc/udev/rules.d/70-persistent-net-rules`

...

`en` - *Ethernet*
`wl` - *Wireless*
`p` - *Bus Number*
`s` - *Slot*
`enp0s3` *(wired network, first bus in PCI slot 3)*

...

~~**InterFace Configuration**~~
~~jane_doe@u1804:~$ `ifconfig`~~
~~*(or)* jane_doe@u1804:~$ `/sbin/ifconfig`~~
~~*(deprecated, replace with* `ip` *)*~~

...

Internet Protocol
jane_doe@u1804:~$ `ip`
(iproute2 replaces net-tools)

...

Interface Down
jane_doe@u1804:~$ `sudo ifconfig enp0s3 down`
(~~iproute2 replaces~~ net-tools)

...

Interface Up
jane_doe@u1804:~$ `sudo ifconfig enp0s3 up`
(~~iproute2 replaces~~ net-tools)

...

Assigning static IP addresses

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

concatenate / `etc` **/** `netplan`)
jane_doe@u1804:~$ `cat /etc/netplan`
(something.yaml)

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
     addresses: [192.168.0.101/24, '2002:2::4/64']
     gateway4: 192.168.1.1
     nameservers:
       addresses: [192.168.1.1,8.8.8.8]
```

Apply NetPlan
jane_doe@u1804:~$ `sudo netplan apply`

legacy variant *(basically, any version of Ubuntu older than 17.10)*
concatenate / `etc` **/** `network` **/** `interfaces`)
jane_doe@u1804:~$ `cat /network/interfaces`

```
# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 10.10.96.1
    netmask 255.255.255.0
    broadcast 10.10.96.255
    dns-search local.lan
    dns-nameservers 10.10.96.1
```

restart networking
jane_doe@u1804:~$ `sudo systemctl restart networking.service`
(legacy variant , before systemD)
jane_doe@u1804:~$ `sudo /etc/init.d/networking restart`

install tmux - *terminal multiplexer*
jane_doe@u1804:~$ `sudo apt install tmux`
(activate)
jane_doe@u1804:~$ `tmux`
(demo)
jane_doe@u1804:~$ `top`
`Ctrl + B`
`D` *(should exit)*
...
reattach - *terminal multiplexer*
jane_doe@u1804:~$ `tmux a`
...
! *use* `tmux` *before* ***either*** `restarting` *technique*
...
show **IP addresses**
jane_doe@u1804:~$ `ip a`

NetworkManager

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

***NetworkManager** is a utility for managing network connectivity on your server, though it's largely been replaced with **Netplan**.*

Linux name resolution

concatenate / `etc` **/** `nsswitch.conf`)
`hosts: files dns`
...
concatenate / `etc` **/** `hosts`)
`10.10.96.124 minecraftserver`
...
*On legacy **Ubuntu** servers , there was a file ,* **/** `etc` **/** `resolv.conf`
jane_doe@u1804:~$ `cat /network/resolv.conf`
...
DNS nameservers *that the server is currently pointing to*
jane_doe@u1804:~$ `systemd-resolve --status |grep DNS\ Servers`

OpenSSH

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

(confirm **SSH Daemon** *installation)*

jane_doe@u1804:~$ `which sshd`

(should return) `/usr/sbin/sshd`

...

install OpenSSH-Server

jane_doe@u1804:~$ `sudo apt install openssh-server`

...

(confirm **SSH Client** *installation)*

jane_doe@u1804:~$ `which ssh`

(should return) `/usr/sbin/ssh`

...

install OpenSSH-Client

jane_doe@u1804:~$ `sudo apt install openssh-client`

...

SSH *status*

jane_doe@u1804:~$ `systemctl status ssh`

...

SSH *start*

jane_doe@u1804:~$ `sudo systemctl start ssh`

...

SSH *enable*

jane_doe@u1804:~$ `sudo systemctl enable ssh`

...

legacy variant *(14.04 & 12.04)*

jane_doe@u1804:~$ `sudo service ssh start`

jane_doe@u1804:~$ `sudo update-rc.d ssh defaults`

...

listening ports , *restrict output to SSH*

jane_doe@u1804:~$ `sudo netstat -tulpn |grep ssh`

...

SSH *connect via* **IP address**

jane_doe@u1804:~$ `ssh 10.10.96.10`

...

SSH *connect* **user @** *, via* **IP address**

jane_doe@u1804:~$ `ssh fmulder@10.10.96.10`

...

SSH *, port , user @ , IP address*

jane_doe@u1804:~$ `ssh -p 2242 fmulder@10.10.96.10`

...

exit

jane_doe@u1804:~$ `exit`

or **`Ctrl+D`** *, especially if you have processes to leave running in the background*

SSH key management

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

Generate SSH Key

jane_doe@u1804:~$ `ssh-keygen`

(default location) `/home/<user>/.ssh`

(passphrase , optional)

...

id_rsa *&* **id_rsa.pub**

jane_doe@u1804:~$ `ls -l /home/<user>/.ssh`

jane_doe@u1804:~$ `ls -l /home/jane_doe/.ssh`

...

SSH *transmit* ***public*** *key to a target server,*

jane_doe@u1804:~$ `ssh-copy-id -i ~/.ssh/id_rsa.pub unicorn`

(default location)

`~/.ssh/authorized_keys`

...

start **SSH** *agent*

jane_doe@u1804:~$ `eval $(ssh-agent)`

...

unlock key *(via agent)*

jane_doe@u1804:~$ `ssh-add ~/.ssh/id_rsa`

...

change pass-phrase

jane_doe@u1804:~$ `ssh-keygen -p`

`Enter` *accepts default file* `id_rsa`

simplifying SSH connections with a `config file`

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

edit / `home` **/** `<user>` **/** `.ssh` **/** `config )`

jane_doe@u1804:~$ `nano /home/jane_doe/.ssh/config`

```
host myserver
    Hostname 192.168.1.23
    Port 22
    User jdoe

Host nagios
    Hostname nagios.local.lan
    Port 2222
    User nagiosuser
```

SSH *(with `config`)*
jane_doe@u1804:~$ `ssh nagios`
(same as..)_
jane_doe@u1804:~$ `ssh -p 2222 nagiosuser@nagios.local.lan`

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 5. Packages

package management

```
apt install tmux
apt install tmux
```

hardware enablement updates

```
sudo apt install --install-recommends linux-generic-hwe-16.04
```

Debian vs Snap

```
apt install tmux
apt install tmux
```

Installing & removing software

```
sudo apt install openssh-server
sudo apt install <package1> <package2> <package3>
sudo apt-get install apache2
sudo apt update
sudo apt remove <package>
sudo apt remove <package1> <package2> <package3>
sudo apt remove --purge <package>
snap find <package>
snap find nmap
sudo snap install nmap
which nmap
sudo snap remove nmap
sudo snap refresh nmap
sudo snap refresh
```

Searching for packages

```
apt search <search term>
apt search apache php
apt-cache show libapache2-mod-php
```

managing repositories

```
deb http://us.archive.ubuntu.com/ubuntu/ bionic main restricted
sudo apt update
sudo apt-add-repository ppa:username/myawesomesoftware-1.0
```

Backing up & restoring Debian packages

```
dpkg --get-selections > packages.list
tmux install
sudo apt update
/usr/bin/dselect
sudo apt install dselect
sudo dselect update
sudo dpkg --set-selections < packages.list
sudo apt-get dselect-upgrade
```

Cleaning up orphaned apt packages

```
apt install tmux
apt install tmux
apt install tmux
```

Making use of Aptitude

```
sudo apt install aptitude
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
sudo aptitude unmarkauto <packagename>
sudo aptitude
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 6. Processes

Monitor & Controll Processes

```
apt install tmux
apt install tmux
```

PS Command

```
pidof vim
ps a
ps au
ps aux
ps aux | grep nginx
ps u -C nginx
ps aux --sort=-pcpu
ps aux --sort=-pcpu | head -n 5
ps aux --sort=-pmem | head -n 5
```

Managing jobs

```
sudo apt install vim-nox
[1]+ Stopped nano
ps au |grep nano
jay 1070 0.0 0.9 39092 7320 pts/0 T 15:53 0:00 nano
[1]- Stopped nano file1.txt
[2]+ Stopped nano file2.txt
fg 1
sudo apt install htop
htop &
```

misbehaving processes

```
sudo kill 31258
man 7 signal
sudo kill -9 31258
sudo killall myprocess
sudo killall -9 myprocess
```

htop

```
sudo apt install htop
htop
apt install tmux
apt install tmux
```

system processes

```
systemctl |grep ssh
systemctl status ssh
systemctl status -l ssh
sudo systemctl stop ssh
sudo systemctl start ssh
sudo systemctl enable ssh
sudo systemctl disable ssh
service ssh status
ssh start/running, process 907
sudo service ssh stop
ssh stop/waiting
sudo service ssh start
ssh start/running, process 1304
/etc/init.d/ssh start
/etc/init.d/ssh stop
/etc/init.d/ssh restart
/etc/init.d/ssh reload
/etc/init.d/ssh status
sudo systemctl enable myservice@myuser
```

Monitoring memory usage

```
free -m
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
cat /proc/sys/vm/swappiness
sudo sysctl vm.swappiness=30
/etc/sysctl.conf
vm.swappiness = 30
```

scheduling Tasks with Cron

```
crontab -l
sudo crontab -u jdoe -l
no crontab for jdoe
crontab -e
EDITOR=vim crontab -e
m h dom mon dow command
3 0 * * 4 /usr/local/bin/cleanup.sh
* 0 * * * /usr/bin/apt-get update
0 1 1 * * /usr/local/bin/run_report.sh
```

load average

```
cat /proc/loadavg
0.36, 0.29, 0.31
1.87, 1.53, 1.22
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 7. Services

Setting Up Network Services

```
apt install tmux
apt install tmux
```

Planning an IP address scheme

```
Network: 192.168.1.0/24
Network equipment: 192.168.1.1 - 192.168.1.10
Servers: 192.168.1.11 - 192.168.1.99
DHCP: 192.168.1.100 - 192.168.1.240
Reservations: 192.168.1.241 - 192.168.1.254
```

Serving IP addresses with isc-dhcp-server

```
sudo apt install isc-dhcp-server
systemctl status isc-dhcp-server
sudo systemctl stop isc-dhcp-server
sudo mv /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.orig
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

```
default-lease-time 86400;
max-lease-time 86400;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option domain-name "local.lan";
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.240;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
}
```

```
default-lease-time 43200;
max-lease-time 86400;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option domain-name "local.lan";
authoritative;
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.240;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
}
```

```
INTERFACESv4=""
INTERFACESv4="enp0s3"
sudo systemctl start isc-dhcp-server
sudo systemctl status isc-dhcp-server
sudo tail -f /var/log/syslog
May 5 22:07:36 hermes dhcpd: DHCPDISCOVER from 52:54:00:88:f8:bc via enp0s3
May 5 22:07:36 hermes dhcpd: DHCPOFFER on 192.168.1.103 to 51:52:01:87:f7:bc via enp0s3
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

```
lease 192.168.1.138 {
  starts 0 2016/05/06 16:37:30;
  ends 0 2016/05/06 16:42:30;
  cltt 0 2016/05/06 16:37:30;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 32:6e:92:01:1f:7f;
}
```

Setting up DNS with bind

```
sudo apt install bind9
// forwarders {
// 0.0.0.0;
// };
forwarders {
8.8.8.8;
8.8.4.4;
};
sudo systemctl restart bind9
systemctl status bind9
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.100 192.168.1.240;
option routers 192.168.1.1;
```

`option domain-name-servers 192.168.1.1;`

```
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.100 192.168.1.240;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.1;
}
```

```
dig www.packtpub.com
;; Query time: 98 msec
;; Query time: 1 msec
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "local.lan" IN {
type master;
file "/etc/bind/net.local.lan";
};
```

```
zone "local.lan" IN {
     type master;
     file "/etc/bind/net.local.lan";
};
```

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

```
 $TTL 1D
 @ IN SOA local.lan. hostmaster.local.lan. (

 201808161 ; serial

 8H ; refresh
 4H ; retry
 4W ; expire
 1D ) ; minimum
 IN A 192.168.1.1
 ;
 @ IN NS hermes.local.lan.
 fileserv        IN  A   192.168.1.3
 hermes     IN A    192.168.1.1
 mailserv        IN  A   192.168.1.5
 mail            IN  CNAME   mailserv.
 web01           IN  A   192.168.1.7
```

```
$TTL 1D
@ IN SOA local.lan. hostmaster.local.lan. (
201808161 ; serial
8H ; refresh
4H ; retry
4W ; expire
1D ) ; minimum
IN A 192.168.1.1
@ IN NS hermes.local.lan.
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

```
 fileserv        IN  A   192.168.1.3
 hermes        IN  A    192.168.1.1
 mailserv        IN  A   192.168.1.5
 mail            IN  CNAME   mailserv.
 web01           IN  A   192.168.1.7
```

```
sudo systemctl restart bind9
```

```
systemctl status bind9
```

```
cat /var/log/syslog | grep bind9
```

```
dig webserv.local.lan
```

```
dig www.packtpub.com
```

```
;; Query time: 1 msec
```

```
;; SERVER: 127.0.0.53#53(127.0.0.53)
```

```
;; WHEN: Sat Feb 10 10:00:59 EST 2018
```

```
;; MSG SIZE rcvd: 83
```

```
;; Query time: 1 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Feb 10 10:00:59 EST 2018
;; MSG SIZE  rcvd: 83
```

```
systemd-resolve --status |grep DNS Servers
```

Creating a secondary (slave) DNS server

```
options {
        directory "/var/cache/bind";
        forwarders {
                8.8.8.8;
                8.8.4.4;
        };
        dnssec-validation auto;

        auth-nxdomain no;
        listen-on-v6 { any; };
};
```

```
options {
        directory "/var/cache/bind";
        allow-transfer { localhost; 192.168.1.2; };
        forwarders {
                8.8.8.8;
                8.8.4.4;
        };
        dnssec-validation auto;

        auth-nxdomain no;
        listen-on-v6 { any; };
};
```

```
zone "local.lan" IN {
    type slave;
    masters { 192.168.1.1; };
    file "/var/lib/bind/net.local.lan";
};
```

```
dig @192.168.1.1 fileserv
```

```
dig @192.168.1.2 fileserv
```

```
May 06 13:19:47 ubuntu-server named[2615]: transfer of 'local.lan/IN' from 10.10.99.184#53: connected using 192.168.1.2#35275
May 06 13:19:47 ubuntu-server named[2615]: zone local.lan/IN: transferred serial 201602093
May 06 13:19:47 ubuntu-server named[2615]: transfer of 'local.lan/IN' from 10.10.99.184#53: Transfer status: success
May 06 13:19:47 ubuntu-server named[2615]: transfer of 'local.lan/IN' from 10.10.99.184#53: Transfer completed: 1 messages, 10 records, 290
May 06 13:19:47 ubuntu-server named[2615]: zone local.lan/IN: sending notifies (serial 201602093)
```

```
default-lease-time 86400;
max-lease-time 86400;
```

```
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option domain-name "local.lan";
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.240;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
}
```

```
default-lease-time 86400;
max-lease-time 86400;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option domain-name "local.lan";
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.240;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1, 192.168.1.2;
}
```

Setting up an internet gateway

```
apt install tmux
```

```
apt install tmux
```

Keeping your clock in sync with NTP

```
sudo apt install ntp
```

```
systemctl status ntp
```

```
pool 0.ubuntu.pool.ntp.org iburst
pool 1.ubuntu.pool.ntp.org iburst
pool 2.ubuntu.pool.ntp.org iburst
pool 3.ubuntu.pool.ntp.org iburst
```

```
#restrict 192.168.123.0 mask 255.255.255.0 notrust
```

```
restrict 192.168.1.0 mask 255.255.255.0
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 8. Files

Sharing & Transferring Files

```
apt install tmux
```

```
apt install tmux
```

File server considerations

```
apt install tmux
apt install tmux
```

Sharing files with Windows users via Samba

```
sudo apt install samba
sudo systemctl stop smbd
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.orig
```

```
[global]
server string = File Server
workgroup = WORKGROUP
security = user
map to guest = Bad User
name resolve order = bcast hosts wins
include = /etc/samba/smbshared.conf
```

```
[global]
server string = File Server
workgroup = WORKGROUP
security = user
map to guest = Bad User
name resolve order = bcast hosts wins
include = /etc/samba/smbshared.conf
```

```
[Documents]
    path = /share/documents
    force user = myuser
    force group = users
    public = yes
    writable = no

[Public]
    path = /share/public
    force user = myuser
    force group = users
    create mask = 0664
    force create mode = 0664
    directory mask = 0777
    force directory mode = 0777
    public = yes
    writable = yes
```

```
path = /share/documents
force user = myuser
force group = users
public = yes
writable = no
```

```
create mask = 0664
force create mode = 0664
directory mask = 0777
force directory mode = 0777
```

```
public = yes
writable = yes
sudo systemctl start smbd
//myserver/shared/documents /mnt/documents cifs username=myuser,noauto 0 0
sudo apt install smbclient cifs-utils
sudo mount /mnt/documents
sudo mount -t cifs //myserver/Documents -o username=myuser /mnt/documents
```

Setting up NFS shares

```
sudo mkdir /exports
sudo apt install nfs-kernel-server
sudo mv /etc/exports /etc/exports.orig
/exports/backup
/exports/documents
/exports/public
```

```
  /exports *(ro,fsid=0,no_subtree_check)
  /exports/backup 192.168.1.0/255.255.255.0(rw,no_subtree_check)
  /exports/documents 192.168.1.0/255.255.255.0(ro,no_subtree_check)
  /exports/public 192.168.1.0/255.255.255.0(rw,no_subtree_check)
```

```
man export
# sudo Domain = localdomain
sudo systemctl restart nfs-kernel-server
systemctl status -l nfs-kernel-server
sudo apt install nfs-common
sudo mount myserver:/documents /mnt/documents
/exports *(ro,fsid=0,no_subtree_check)
sudo exportfs -a
```

Transferring files with `Rsync`

```
sudo rsync -r /home/myusr /backup
sudo rsync -a /home/myuser /backup
-rlptgoD
sudo rsync -av /home/myuser /backup
sudo rsync -av /home/myuser admin@192.168.1.5:/backup
sudo rsync -av --delete /src /target
sudo rsync -avb --delete /src /target
sudo rsync -avb --delete --backup-dir=/backup/incremental /src /target
CURDATE=$(date +%m-%d-%Y)
export $CURDATE
sudo rsync -avb --delete --backup-dir=/backup/incremental/$CURDATE /src /target
sudo rsync -avb --delete --backup-dir=/backup/incremental/08-17-2018 /src /target
```

Transferring files with `SCP`

```
/usr/bin/scp
apt install tmux
scp myfile.txt jdoe@192.168.1.50:/home/jdoe
scp myfile.txt jdoe@192.168.1.50:
scp jdoe@192.168.1.50:myfile.txt .
scp -r /home/jdoe/downloads/linux_iso jdoe@192.168.1.50:downloads
scp -r /home/jdoe/downloads/linux_iso jdoe@192.168.1.50:/home/jdoe/downloads
scp -P 2222 -r /home/jdoe/downloads/linux_iso jdoe@192.168.1.50:downloads
scp -rv /home/jdoe/downloads/linux_iso jdoe@192.168.1.50:downloads
```

Mounting remote directories with `SSHFS`

```
sudo apt install sshfs
sshfs myuser@192.168.1.50:/share/myfiles /mnt/myfiles
sudo umount /mnt/myfiles
umount: /mnt/myfiles: Permission denied
fusermount -u /mnt/myfiles
myuser@192.168.1.50:/share/myfiles /mnt/myfiles fuse.sshfs rw,noauto,users,_netdev 0 0
mount /mnt/myfiles
```

Q & A

1. $ `sudo`

2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 9. Databases

DataBase Management

```
apt install tmux
```

```
apt install tmux
```

Preparations for setting up a database server

```
apt install tmux
```

```
apt install tmux
```

Installing MariaDB

```
sudo apt install mariadb-server
```

```
sudo apt install mysql-server
```

```
systemctl status mariadb
```

```
sudo mysql_secure_installation
```

```
  Enter current password for root (enter for none):
  Set root password? [Y/n]
  Remove anonymous users? [Y/n]
  Disallow root login remotely? [Y/n]
  Remove test database and access to it? [Y/n]
  Reload privilege tables now? [Y/n]
```

```
sudo mariadb
```

```
mariadb -u root -p
```

```
UPDATE mysql.user SET plugin = 'mysql_native_password' WHERE USER='root';
```

```
FLUSH PRIVILEGES;
```

```
Now, you'll be able to access the MariaDB shell as root with native authentication:
```

```
mysql -u root -p
```

```
MariaDB [(none)]>
```

MariaDB configuration

```
debian.cnf
```

```
debian-start
```

```
mariadb.cnf
```

```
my.cnf
```

```
my.cnf.fallback
```

```
conf.d
```

```
mariadb.conf.d
```

```
  [client]
  host     = localhost
  user     = root
  password =
  socket    = /var/run/mysqld/mysqld.sock
  [mysql_upgrade]
  host      = localhost
  user      = root
  password =
  socket    = /var/run/mysqld/mysqld.sock
  basedir   = /usr


  # The MariaDB/MySQL tools read configuration files in the following order:
  # 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,
  # 2. "/etc/mysql/conf.d/*.cnf" to set global options.
  # 3. "/etc/mysql/mariadb.conf.d/*.cnf" to set MariaDB-only options.
  # 4. "~/.my.cnf" to set user-specific options.


  !includedir /etc/mysql/conf.d/
  !includedir /etc/mysql/mariadb.conf.d/
```

Managing MariaDB databases

```
MariaDB [(none)]>
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
CREATE USER 'admin'@'%' IDENTIFIED BY 'password';
CREATE USER 'admin'@'192.168.1.%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost';
FLUSH PRIVILEGES;
mariadb -u admin -p
mariadb -u admin -p<password>
GRANT SELECT ON *.* TO 'readonlyuser'@'localhost' IDENTIFIED BY 'password';
CREATE DATABASE mysampledb;
SHOW DATABASES;
SELECT HOST, USER, PASSWORD FROM mysql.user;
GRANT SELECT ON mysampledb.* TO 'appuser'@'localhost' IDENTIFIED BY 'password';
GRANT ALL ON mysampledb.* TO 'appuser'@'localhost' IDENTIFIED BY 'password';
SHOW GRANTS FOR 'appuser'@'localhost';
DELETE FROM mysql.user WHERE user='myuser' AND host='localhost';
USE mysampledb;
CREATE TABLE Employees (Name char(15), Age int(3), Occupation char(15));
INSERT INTO Employees VALUES ('Joe Smith', '26', 'Ninja');
SELECT * FROM Employees;
DELETE FROM Employees WHERE Name = 'Joe Smith';
DROP TABLE Employees;
DROP DATABASE mysampledb;
mysqldump -u admin -p --databases mysampledb > mysampledb.sql
sudo mariadb < mysampledb.sql
```

Setting up a slave database server

```
[mysql]
[mysqld]
log-bin
binlog-do-db=mysampledb
server-id=1
bind-address = 127.0.0.1
bind-address = 0.0.0.0
GRANT REPLICATION SLAVE ON *.* to 'replicate'@'192.168.1.204' identified by 'slavepassword';
sudo systemctl restart mariadb
FLUSH TABLES WITH READ LOCK;
mysqldump -u admin -p --databases mysampledb > mysampledb.sql
mariadb -u root -p < mysampledb.sql
[mysqld]
server-id=2
sudo systemctl restart mariadb
CHANGE MASTER TO MASTER_HOST="192.168.1.184", MASTER_USER='replicate', MASTER_PASSWORD='slavepassword';
UNLOCK TABLES;
SHOW SLAVE STATUSG;
Slave_IO_State: Waiting for master to send event
START SLAVE;
SHOW SLAVE STATUSG;
USE mysampledb;
INSERT INTO Employees VALUES ('Optimus Prime', '100', 'Transformer');
USE mysampledb;
SELECT * FROM Employees;
sudo netstat -tulpn |grep mysql
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN 946/mysqld
GRANT REPLICATION SLAVE ON *.* to 'replicate'@'192.168.1.204' identified by 'slavepassword';
FLUSH PRIVILEGES
CHANGE MASTER TO MASTER_HOST="192.168.1.184", MASTER_USER='replicate', MASTER_PASSWORD='slavepassword';
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 10. Serving Web Content

Installing and configuring Apache

```
sudo apt install apache2
systemctl status apache2
sudo a2ensite acmeconsulting.com.conf
sudo systemctl reload apache2
sudo a2dissite acmeconsulting.com.conf
sudo systemctl reload apache2
# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf
```

```
  <VirtualHost *:80>
      ServerAdmin webmaster@localhost
```

```
    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>



<VirtualHost 192.168.1.104:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/acmeconsultingErrorLog ${APACHE_LOG_DIR}/acmeconsulting.com-error.log
    CustomLog ${APACHE_LOG_DIR}/acmeconsulting.com-access.log combined
</VirtualHost>



<VirtualHost *:80>
    ServerName acmeconsulting.com
    DocumentRoot /var/www/acmeconsulting
</VirtualHost>

<VirtualHost *:80>
    ServerName acmesales.com
    DocumentRoot /var/www/acmesales
</VirtualHost> v
```

Installing additional Apache modules

```
apt search libapache2-mod
sudo apt install libapache2-mod-php7.2
apache2 -l
Which module(s) do you want to enable (wildcards ok)?
sudo a2enmod php7.2
sudo a2enmod php7.2
Module php7.2 already enabled
Enabling module php7.2.
To activate the new configuration, you need to run:
service apache2 restart
sudo a2dismod php7.2
Module php7.2 disabled.
To activate the new configuration, you need to run:
service apache2 restart
```

Securing Apache with SSL

```
sudo netstat -tulpn |grep apache
tcp6 0 0 :::80 :::* LISTEN 2791/apache2
tcp6 0 0 :::80 :::* LISTEN 3257/apache2
tcp6 0 0 :::443 :::* LISTEN 3257/apache2
sudo a2enmod ssl
sudo systemctl restart apache2
```

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                DocumentRoot /var/www/html

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

        SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
        SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

                <FilesMatch ".(cgi|shtml|phtml|php)$">
                                SSLOptions +StdEnvVars
                </FilesMatch>
```

```
                  <Directory /usr/lib/cgi-bin>
                                  SSLOptions +StdEnvVars
                  </Directory>

          </VirtualHost>
  </IfModule>
```

```
sudo a2ensite default-ssl.conf
```

```
sudo mkdir /etc/apache2/certs
```

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/certs/mysite.key -out /etc/apache2/certs/mysite.crt
```

```
  Country Name (2 letter code) [AU]:US
  State or Province Name (full name) [Some-State]:Michigan
  Locality Name (eg, city) []:Detroit
  Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
  Organizational Unit Name (eg, section) []:IT
  Common Name (e.g. server FQDN or YOUR name) []:myserver.mydomain.com
  Email Address []:webmaster@mycompany.com
```

```
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
```

```
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

```
# SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
```

```
# SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

```
SSLCertificateFile /etc/apache2/certs/mysite.crt
```

```
SSLCertificateKeyFile /etc/apache2/certs/mysite.key
```

```
sudo systemctl reload apache2
```

```
openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr
```

```
ServerName mydomain.com:443
```

```
  <IfModule mod_ssl.c>
          <VirtualHost *:443>
                  ServerName acmeconsulting.com:443

                 ServerAdmin webmaster@localhost

                  DocumentRoot /var/www/acmeconsulting

                  ErrorLog ${APACHE_LOG_DIR}/acmeconsulting.com-error.log
                  CustomLog ${APACHE_LOG_DIR}/acmeconsulting.com-access.log combined

                  SSLEngine on

          SSLCertificateFile      /etc/apache2/certs/acmeconsulting/acme.crt
          SSLCertificateKeyFile /etc/apache2/certs/acmeconsulting/acme.key

                  <FilesMatch ".(cgi|shtml|phtml|php)$">
                                  SSLOptions +StdEnvVars
                  </FilesMatch>
                  <Directory /usr/lib/cgi-bin>
                                  SSLOptions +StdEnvVars
                  </Directory>

          </VirtualHost>
  </IfModule>
```

Installing and configuring NGINX

```
sudo apt install nginx
```

```
/etc/nginx/sites-available/acmesales.com
```

```
sudo ln -s /etc/nginx/sites-available/acmesales.com /etc/nginx/sites-enabled/acmesales.com
```

```
sudo systemctl reload nginx
```

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/acmesales.com
```

```
listen 80 default_server;
```

```
listen [::]:80 default_server;
```

```
listen 80;
```

```
listen [::]:80;
```

```
server_name acmesales.com www.acmesales.com;
```

```
root /var/www/html;
```

```
root /var/www/acmesales.com;
```

```
  server {
          listen 80;
          listen [::]:80;

          root /var/www/acmesales.com;

          index index.html index.htm index.nginx-debian.html;

          server_name acmesales.com www.acmesales.com;

          location / {
                  try_files $uri $uri/ =404;
          }
  }
```

```
listen 443 ssl;
```

```
listen [::]:443 ssl;
```

```
ssl_certificate /etc/certs/cert.pem;
```

```
ssl_certificate_key /etc/certs/cert.key;
```

```
ssl_session_timeout 5m;
```

```
  server {
          listen 443 ssl;
          listen [::]:443 ssl;

          root /var/www/html;

          index index.html index.htm index.nginx-debian.html;

          server_name acmesales.com www.acmesales.com;

           ssl_certificate /etc/certs/cert.pem;
           ssl_certificate_key /etc/certs/cert.key;
           ssl_session_timeout 5m;
          location / {
                  try_files $uri $uri/ =404;
          }
  }
```

```
return 301 https://$host$request_uri;
```

Setting up failover with keepalived

```
nmap -sP 192.168.1.0/24
```

```
sudo apt install keepalived
```

```
Condition: start condition failed
```

```
sudo mkdir /etc/keepalived
```

```
  global_defs {
      notification_email {
      myemail@mycompany.com
      }
      notification_email_from keepalived@mycompany.com
```

```
    smtp_server 192.168.1.150
    smtp_connect_timeout 30
    router_id mycompany_web_prod
}
vrrp_instance VI_1 {
    smtp_alert
    interface enp0s3
    virtual_router_id 51
    priority 100

    advert_int 5
    virtual_ipaddress {
    192.168.1.200
    }
}
```

```
global_defs {
    notification_email {
    myemail@mycompany.com
    }
    notification_email_from keepalived@mycompany.com
    smtp_server 192.168.1.150
    smtp_connect_timeout 30
    router_id mycompany_web_prod
}
```

```
vrrp_instance VI_1 {
    smtp_alert
    interface enp0s3
    virtual_router_id 51
    priority 100

    advert_int 5



    virtual_ipaddress {
    192.168.1.200
}
```

```
sudo systemctl start keepalived
```

```
sudo systemctl status -l keepalived
```

```
<html>
    <title>keepalived test</title>
    <body>
        <p>This is server #1!</p>
    </body>
</html>
```

```
sudo systemctl stop keepalived
```

Setting up and configuring Nextcloud

```
wget <URL of Nextcloud>
```

```
wget https://download.nextcloud.com/server/releases/nextcloud-13.0.0.zip
```

```
unzip nextcloud-13.0.0.zip
```

```
sudo apt install unzip
```

```
sudo mv nextcloud /var/www/html/nextcloud
```

```
sudo chown www-data:www-data -R /var/www/html/nextcloud
```

```
/etc/apache2/sites-available/nextcloud.conf
```

```
Alias /nextcloud "/var/www/html/nextcloud/"

<Directory /var/www/html/nextcloud/>
  Options +FollowSymlinks
  AllowOverride All
```

```
   <IfModule mod_dav.c>
    Dav off
   </IfModule>

   SetEnv HOME /var/www/html/nextcloud
   SetEnv HTTP_HOME /var/www/html/nextcloud
  </Directory>
```

```
sudo a2ensite nextcloud.conf
```

```
sudo apt install libapache2-mod-php7.2 php7.2-curl php7.2-gd php7.2-intl php7.2-mbstring php7.2-mysql php7.2-xml
php7.2-zip
```

```
sudo systemctl restart apache2
```

```
CREATE DATABASE nextcloud;
```

```
GRANT ALL ON nextcloud.* to 'nextcloud'@'localhost' IDENTIFIED BY 'super_secret_password';
```

```
http://192.168.1.100/nextcloud
```

```
http://nextcloud.yourdomain.com
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Martin Fjordvald,
Clement Nedelcu
Nginx
HTTP Server
Fourth Edition
Harness the power of Nginx to make the most of your infrastructure and serve pages faster than ever before
Packt>

Chapter 11. Shell Techniques

Learning Advanced Shell Techniques

```
apt install tmux
```

```
apt install tmux
```

Understanding the Linux shell

```
cat /etc/passwd
```

Bash history

```
!100
```

```
systemctl status apache2
```

```
history -d 100
```

```
mariadb -u root -pSuperSecretPassword
```

some useful command-line tricks

```
sudo !!
sudo apt update && sudo apt install apache2
sudo apt update; sudo apt install apache2
echo $?
alias install="sudo apt install"
install tmux
alias i="sudo apt install"
i tmux
alias cpu10='ps auxf | sort -nr -k 3 | head -10'
alias mem10='ps auxf | sort -nr -k 4 | head -10'
alias lsmount='mount |column -t'
alias c=clear
```

```
alias i='sudo apt install'
alias cpu10='ps auxf | sort -nr -k 3 | head -10'
alias mem10='ps auxf | sort -nr -k 4 | head -10'
alias lsmount='mount |column -t'
```

```
cd -
```

Redirecting output

```
cat /var/log/syslog | grep apache2
echo "this is a test" >> ~/myfile.txt
echo "this is a test" > ~/myfile.txt
find /etc -name *apache*
```

```
find: '/etc/lvm/backup': Permission denied
find: '/etc/lvm/archive': Permission denied
find: '/etc/vpnc': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/libvirt/secrets': Permission denied
find: '/etc/polkit-1/localauthority': Permission denied
```

```
find /etc -name *apache* 2>/dev/null
find etc -name *apache* 1> ~/myfile.txt
```

Understanding variables

```
myvar='Hello world!'
echo $myvar
mydir="/etc"
ls $mydir
env
read age
echo $age
echo "Please enter your age"
read age
echo "Your age is $age"
```

Writing simple scripts

```
nano ~/myscript.sh
```

```
#!/bin/bash
echo "My name is $USER"
echo "My home directory is $HOME"
```

```
chmod +x ~/myscript.sh
```

```
~/myscript.sh
```

```
"My name is jay"
```

```
"My home directory is /home/jay"
```

```
#!/bin/bash
sudo apt install apache2
sudo apt install libapache2-mod-php7.2
sudo a2enmod php
sudo systemctl restart apache2
```

```
#!/bin/bash

# Install Apache if it's not already present
if [ ! -f /usr/sbin/apache2 ]; then
    sudo apt install -y apache2
    sudo apt install -y libapache2-mod-php7.2
    sudo a2enmod php
    sudo systemctl restart apache2
fi
```

```
# Install Apache if it's not already present
```

```
if [ ! -f /usr/sbin/apache2 ]; then
```

```
#!/bin/bash
myvar=10
if [ $myvar -eq 1]; then
    echo "The variable equals 1"
else
    echo "The variable doesn't equal 1"
fi
```

```
#!/bin/bash
myvar=1
while [ $myvar -le 15 ]
do
    echo $myvar
    ((myvar++))
done
```

```
myvar=1
```

```
while [ $myvar -le 15 ]
```

```
echo $myvar
```

```
((myvar++))
```

```
done
```

```
#!/bin/bash
turtles='Donatello Leonardo Michelangelo Raphael'
for t in $turtles
do
echo $t
done
```

```
turtles='Donatello Leonardo Michelangelo Raphael'
```

```
echo $turtles
```

```
for t in $turtles
```

```
do
```

```
echo $t
```

```
done
```

Putting it all together: Writing an rsync backup script

```
rsync -avb --delete --backup-dir=/backup/incremental/08-17-2018 /src /target
```

```
rsync -avb --delete --backup-dir=/backup/incremental/$CURDATE /src /target
```

```
#/bin/bash
CURDATE=$(date +%m-%d-%Y)
if [ ! -f /usr/bin/rsync ]; then
    sudo apt install -y rsync
fi
rsync -avb --delete --backup-dir=/backup/incremental/$CURDATE /src /target
```

```
chmod +x backup.sh
```

```
mv backup.sh /usr/local/bin
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 12. Virtualization

Virtualization

```
apt install tmux
```

Setting up a virtual machine server

```
egrep -c '(vmx|svm)' /proc/cpuinfo
sudo groupadd kvm
sudo apt install bridge-utils libvirt-bin qemu-kvm qemu-system
systemctl status libvirtd
sudo systemctl stop libvirtd
sudo chown root:kvm /var/lib/libvirt/images
sudo chmod g+rw /var/lib/libvirt/images
sudo usermod -aG kvm <user>
sudo systemctl start libvirtd
sudo systemctl status libvirtd
sudo apt install ssh-askpass virt-manager
sudo cp /etc/libvirt/libvirtd.conf /etc/libvirt/libvirtd.conf.orig
unix_sock_group = "libvirtd"
unix_sock_group = "kvm"
unix_sock_ro_perms = "0777"
unix_sock_ro_perms = "077 0 "
sudo systemctl restart libvirtd
sudo chown root:kvm /var/lib/libvirt/images/ISO
sudo chmod g+rw var/lib/libvirt/images/ISO
```

Creating virtual machines

```
apt install tmux
```

Bridging the virtual machine network

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: false
  bridges:
    br0:
      interfaces: [enp0s3]
      dhcp4: true
      parameters:
        stp: false
        forward-delay: 0
```

```
sudo netplan apply
```

Simplifying virtual machine creation with cloning

```
sudo rm /etc/ssh/ssh_host_*
sudo dpkg-reconfigure openssh-server
```

Managing virtual machines via the command line

```
virsh list
virsh start my_vm
virsh shutdown my_vm
virsh suspend my_vm
virsh resume my_vm
virsh destroy my_vm
```

Q & A

1. $ sudo
2. $ adduser, useradd
3. $ rm jane_doe
4. $ /etc/password & /etc/shadow
5. $ /etc/skel
6. $ su jane_doe
7. $ sudo groupadd accounting
8. $ visudo
9. $ sudo adduser jdoe
10. $ chmod, chown

Chapter 13. Containers

"..so now we have a Dockerfile, what do we do with it? Turn it into an image of course!
Use the docker build command from within the directory that contains the Dockerfile."

```
$ docker build -t ubuntu:1804 .
```

Running Containers

```
apt install tmux
apt install tmux
```

What is containerization?

```
apt install tmux
apt install tmux
```

Understanding the differences between Docker and LXD

```
apt install tmux
apt install tmux
```

Installing Docker

```
sudo apt install docker.io
systemctl status docker
sudo usermod -aG docker <yourusername>
```

Managing Docker containers

```
docker search ubuntu
docker pull ubuntu
docker images
docker rmi 0458a4468cbc
docker run -it ubuntu:latest /bin/bash
docker start 353c6fe0be4d
docker attach 353c6fe0be4d
docker run -dit ubuntu /bin/bash
docker run -dit -p 8080:80 ubuntu /bin/bash
docker attach dfb3e
sudo apt install apache2
apt update
apt install apache2
/etc/init.d/apache2 start
sudo apt install nano
/etc/init.d/apache2 start
docker commit <Container ID> ubuntu/apache-server:1.0
docker run -dit -p 8080:80 ubuntu/apache-server:1.0 /bin/bash
docker stop <Container ID>
```

Automating Docker image creation with Dockerfiles

```
FROM ubuntu
MAINTAINER Jay <myemail@somewhere.net>
# Update the container's packages
RUN apt update; apt dist-upgrade -y
RUN apt install -y apache2 vim-nox
RUN echo "/etc/init.d/apache2 start" >> /etc/bash.bashrc
docker build -t packt/apache-server:1.0 .
docker run -dit -p 8080:80 packt/apache-server:1.0 /bin/bash
```

```
FROM ubuntu
MAINTAINER Jay <myemail@somewhere.net>
# Update the container's packages
RUN apt update; apt dist-upgrade -y
RUN apt install -y apache2 vim-nox
RUN echo "/etc/init.d/apache2 start" >> /etc/bash.bashrc
docker build -t packt/apache-server:1.0 .
docker run -dit -p 8080:80 packt/apache-server:1.0 /bin/bash
```

Managing LXD containers

```
sudo snap install lxd
```

```
sudo usermod -aG lxd <yourusername>
```

```
lxd init
```

```
lxc launch ubuntu:18.04 mycontainer
```

```
lxc exec mycontainer Bash
```

```
lxc exec mycontainer -- sudo --login --user ubuntu
```

```
lxc config set mycontainer boot.autostart 1
```

```
sudo apt update && sudo apt install apache2
```

```
curl <container_ip_address>
```

```
lxc profile create external
```

```
Profile extbr0 created
```

```
lxc network edit external
```

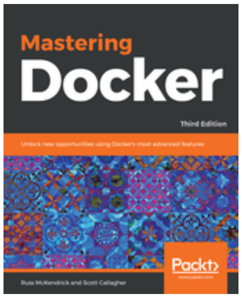
```
  description: External access profile
  devices:
  eth0:
  name: eth0
  nictype: bridged
  parent: br0
  type: nic
```

```
lxc launch ubuntu:18.04 mynewcontainer -p default -p external
```

```
lxc profile add mycontainer external
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`


Mastering
Docker
Third Edition
Unlock new opportunities using Docker's most advanced features
Packt
Russ McKendrick and Scott Gallagher

Chapter 14. Ansible

Automating Server Configuration with Ansible

```
apt install tmux
```

```
apt install tmux
```

Understanding the need for configuration management

```
apt install tmux
```

```
apt install tmux
```

Why Ansible?

```
apt install tmux
apt install tmux
```

Creating a Git repository

```
sudo apt install git
git config user.email "you@example.com"
git config user.name "John Doe"
git clone https://github.com/myusername/ansible.git
echo "this is a test" > testfile.txt
git add testfile.txt
git commit -m "initial commit"
git push origin master
git add myplaybook.yml
git commit -m "insert message about the commit here"
git push origin master
```

Getting started with Ansible

```
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible
/etc/sudoers.d/ansible
ansible ALL=(ALL) NOPASSWD: ALL
sudo chown root:root /etc/sudoers.d/ansible
sudo chmod 440 /etc/sudoers.d/ansible
sudo su - ansible
ssh 192.168.1.123 sudo ls /etc
```

Making your servers do your bidding

```
sudo touch /etc/ansible/hosts
chown ansible /etc/ansible/hosts
chmod 600 /etc/ansible/hosts
192.168.1.145
192.168.1.125
192.168.1.166
myhost1.mydomain.com
myhost2.mydomain.com
myhost3.mydomain.com
[defaults]
inventory = /path/to/hosts
ansible all -m ping
```

```
192.168.1.145 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

```
---
- hosts: all
  become: true
  tasks:
  - name: Install htop
    apt: name=htop
```

```
ansible-playbook packages.yml
- hosts: all
become: true
tasks:
- name: Install htop
apt: name=htop
```

```
---
- hosts: all
  become: true
  tasks:
  - name: Install htop
    apt: name=htop
  - name: Install git
    apt: name=git
  - name: Install vim-nox
    apt: name=vim-nox
```

```
---
- hosts: all
  become: true
  tasks:
  - name: Install packages
    apt: name={{item}}
    with_items:
      - htop
      - git
      - vim-nox
```

```
---
- hosts: all
  become: true
  tasks:
  - name: copy SSH motd
    copy: src=motd dest=/etc/motd
```

Putting it all together – Automating web server deployment

```
---
- hosts: all
  become: true
  tasks:
  - name: Install Apache
    apt: name=apache2
```

```
---
- hosts: all
  become: true
  tasks:
  - name: Install Apache
    apt: name=apache2
  - name: Start the apache2 services
    service: name=apache2 state=started
```

```
<html>
<title>Ansible is awesome!</title>
<body>
    <p>Ansible is amazing. With just a small text file, we automated the setup of a web server!</p>
</body>
</html>
```

```
---
- hosts: all
  become: true
  tasks:
  - name: Install Apache
    apt: name=apache2
  - name: Start the apache2 services
    service: name=apache2 state=started
  - name: Copy index.html
    copy: src=index.html dest=/var/www/html/index.html
```

```
ansible-playbook apache.yml
```

Using Ansible's pull method

```
ansible-pull -U https://github.com/myusername/ansible.git
```

```
ansible-pull -U https://github.com/myusername/ansible.git myplaybook.yml
```

```
---
- hosts: localhost
  become: true
  tasks:
  - name: Install Apache
    apt: name=apache2
  - name: Start the apache2 services
    service: name=apache2 state=started
  - name: Copy index.html
    copy: src=index.html dest=/var/www/html/index.html
```

```
sudo su - ansible
```

```
ansible-pull -U https://github.com/myusername/ansible.git
```

```
ansible-pull -U h https://github.com/myusername/ansible.git apache.yml
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 15. Securing

Securing Your Server

```
apt install tmux
```

```
apt install tmux
```

Lowering your attack surface

```
sudo netstat -tulpn
```

```
sudo apt remove rpcbind
```

```
dpkg --get-selections > installed_packages.txt
```

Understanding and responding to CVEs

```
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
apt install tmux
```

Installing security updates

```
sudo apt upgrade
sudo systemctl restart bind9
/var/cache/apt/archives
sudo dpkg -i /path/to/package.deb
sudo dpkg -i /var/cache/apt/archives/linux-generic_4.15.0.7.8_amd64.deb
sudo apt -f install
```

Automatically installing patches with the Canonical Livepatch service

```
sudo snap install canonical-livepatch
sudo canonical-livepatch enable <token>
sudo canonical-livepatch status
```

Monitoring Ubuntu servers with Canonical's Landscape service

```
sudo add-apt-repository ppa:landscape/17.03
sudo apt update
sudo apt install landscape-server-quickstart
https://<IP_Address>/account/standalone/computers
```

```
sudo apt updatesudo apt install landscape-clientsudo landscape-config --computer-title "My Server" --account-name standalone --url https://<IP_ADDRESS>/message-system --ping-url http://<IP_Address>/ping
```

Securing OpenSSH

```
sudo systemctl restart ssh
Port 65332
ssh -p 65332 myhost
scp -P 65332 myfile myserver:/path/to/dir
AllowUsers larry moe curly
sudo groupadd sshusers
sudo usermod -aG sshusers myuser
AllowGroups admins sshusers gremlins
PermitRootLogin no
PasswordAuthentication no
```

```
Port 65332
Protocol 2
AllowUsers larry moe curly
AllowGroups admins sshusers gremlins
PermitRootLogin no
PasswordAuthentication no
```

Installing and configuring Fail2ban

```
sudo apt install fail2ban
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
#ignoreip = 127.0.0.1/8 ::1
Ignoreip = 127.0.0.1/8 ::1 192.168.1.0/24 192.168.1.245/24
bantime = 10m
maxretry = 5
port = ssh
port = 65332
sudo systemctl restart fail2ban
sudo systemctl status -l fail2ban
sudo fail2ban-client status
enabled = true
```

```
[apache-auth]
enabled = true
port     = http,https
logpath  = %(apache_error_log)
```

```
sudo systemctl restart fail2ban
sudo systemctl status -l fail2ban
sudo fail2ban-client status
```

```
Status
|- Number of jail: 2

  `- Jail list:    apache-auth, sshd
```

MariaDB best practices for secure database servers

```
ALL: 192.168.1.50
ALL: 192.168.1.0/255.255.255.0
ALL: 192.168.1.
ALL: ALL
ssh: 192.168.1.
ALL: ALL
GRANT SELECT ON mysampledb.* TO 'appuser'@'localhost' IDENTIFIED BY 'password';
GRANT SELECT ON mysampledb.* TO 'appuser'@'%' IDENTIFIED BY 'password';
GRANT SELECT ON mysampledb.* TO 'appuser'@'192.168.1.50' IDENTIFIED BY 'password';
GRANT SELECT ON mysampledb.* TO 'appuser'@'192.168.1.% IDENTIFIED BY 'password'
```

Setting up a firewall

```
sudo apt install ufw
sudo ufw status
sudo ufw allow from 192.168.1.156 to any port 22
sudo ufw allow from 192.168.1.0/24 to any port 22
sudo ufw allow from 192.168.1.50
sudo ufw allow 80
sudo ufw allow 443
sudo ufw enable
Firewall is active and enabled on system startup
sudo iptables -L
```

Encrypting and decrypting disks with LUKS

```
sudo apt install cryptsetup
sudo cryptsetup luksFormat /dev/sdb
WARNING!========This will overwrite data on /dev/sdb irrevocably.
Are you sure? (Type uppercase yes):
sudo cryptsetup luksOpen /dev/sdb backup_drive
sudo mkfs.ext4 -L "backup_drive" /dev/mapper/backup_drive
sudo mount /dev/mapper/backup_drive /media/backup_drive
sudo umount /media/backup_drive
sudo cryptsetup luksClose /dev/mapper/backup_drive
sudo cryptsetup luksOpen /dev/sdb backup_drive
sudo mount /dev/mapper/backup_drive /media/backup_drive
sudo cryptsetup luksChangeKey /dev/sdb -S 0
```

Locking down sudo

```
charlie ALL=(ALL:ALL) /usr/sbin/reboot,/usr/sbin/shutdown
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 16. TroubleShooting

Troubleshooting Ubuntu Servers

```
apt install tmux
apt install tmux
```

Evaluating the problem space

```
apt install tmux
apt install tmux
```

Conducting a root cause analysis

```
apt install tmux
apt install tmux
```

Viewing system logs

```
cat /var/log/apache2/access.log
tail /var/log/apache2/access.log
tail -n 100 /var/log/apache2/access.log
tail -f /var/log/apache2/access.log
less /var/log/apache2/access.log
cat /var/log/syslog |grep dhcp
zcat /var/log/syslog.2.gz
```

Tracing network issues

```
nslookup myserver.local
systemd-resolve --status | grep DNS Servers
lspci | grep -i net
lspci |grep -i net
```

```
01:00.1 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet
Controller (rev 12)02:00.0 Network controller: Intel Corporation Wireless 8260 (rev 3a)
```

Troubleshooting resource issues

```
df -hdf -i
sudo ncdu -x /home
du -cksh * | sort -hr | head -n 15
sudo touch /forcefsck
sudo touch /home/forcefsck
sudo iotop
```

Diagnosing defective RAM

```
apt install tmux
apt install tmux
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`
8. $ `visudo`
9. $ `sudo adduser jdoe`
10. $ `chmod, chown`

Chapter 17. Distasters

Preventing & Recovering from Disasters

```
apt install tmux
apt install tmux
```

Preventing disasters

```
apt install tmux
apt install tmux
```

Utilizing Git for configuration management

```
sudo apt install git
sudo mkdir /git
sudo chown jay:jay /git
git init --bare apache2
Initialized empty Git repository in /git/apache2/
git clone 192.168.1.101:/git/apache2
warning: You appear to have cloned an empty repository
sudo cp -rp /etc/apache2 /etc/apache2.bak
sudo mv /etc/apache2/* /git/apache2
sudo rm /etc/apache2
sudo find /git/apache2 -name '.?*' -prune -o -exec chown root:root {} +
sudo ln -s /git/apache2 /etc/apache2
ls -l /etc | grep apache2
lrwxrwxrwx 1 root root 37 2016-06-25 20:59 apache2 -> /git/apache2
sudo systemctl reload apache2
git add
git add <filename>
git commit -a -m "My first commit."
git push origin master
git clone 192.168.1.101:/git/apache2
git commit -a -m "Updated config files."
git push origin master
sudo apt install tig
git checkout 356dd6153f187c1918f6e2398aa6d8c20fd26032
git checkout master
git revert --no-commit 356dd6153f187c1918f6e2398aa6d8c20fd26032
git commit -a -m "The previous commit broke the application. Reverting."
git push origin master
```

Implementing a backup plan

```
URDATE=$(date +%m-%d-%Y)export $CURDATEsudo rsync -avb --delete --backup-dir=/backup/incremental/$CURDATE /src /target
/backup/incremental/8-16-2018
```

Replacing failed RAID disks

```
cat /proc/mdstat
[UU]
sudo hdparm -i /dev/sda
sudo fdisk -l
sudo sfdisk -d /dev/sda | sfdisk sudo /dev/sdb
sudo fdisk -l
sudo mdadm --manage /dev/md0 --add /dev/sdb1
mdadm: added /dev/sdb1
cat /proc/mdstat
```

Utilizing bootable recovery media

```
sudo add-apt-repository ppa:yannubuntu/boot-repair
sudo apt update && apt install boot-repair
```

Q & A

1. $ `sudo`
2. $ `adduser, useradd`
3. $ `rm jane_doe`
4. $ `/etc/password & /etc/shadow`
5. $ `/etc/skel`
6. $ `su jane_doe`
7. $ `sudo groupadd accounting`