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

docker run -it ubuntu:latest /bin/bash

Edit

Manage topics

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

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

TurtleWolf escaped tildas — Latest commit e8f09f0 7 minutes ago

captured_Images	what's up with wildeCarde & exclamation in etc/shadow file?	a day ago
.env	Update .env	13 days ago
.gitignore	Initial commit	15 days ago
Dockerfile	added docker demos to edit	13 days ago
LICENSE	Initial commit	15 days ago
README.md	escaped tildas	7 minutes ago
UbuntuServersU1804.pdf	escaped tildas	7 minutes ago
docker-compose.yml	Update docker-compose.yml	13 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

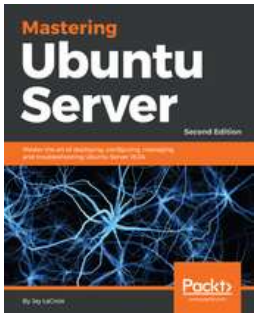
detached interactive terminal on port 8080 named u1804

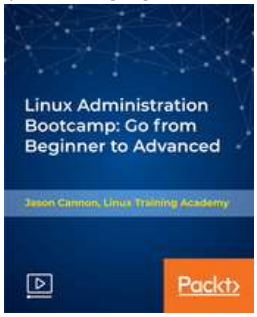
```
$ docker stop u1804
$ docker start u1804
```

$ docker attach u1804

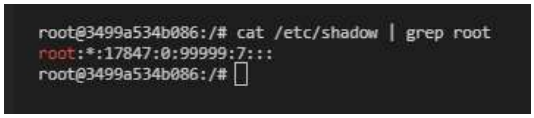
u1804 represents **containerID**

```
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



Managing Users

Understanding when to use `root`

Creating and removing `users`

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

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`

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

...
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

...
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

...

Configuring administrator access with sudo

...
configure `visudo` default editor to `vim`
jane_doe@u1804:~$ `sudo EDITOR=vim visudo`
`Object type : User : Group : Other's`
`-dl : rwx : rwx : rwx`
...

Setting permissions on files and directories

...
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` (would be the same as) `chmod -rw------- filename.txt`
jane_doe@u1804:\~$ `chmod 740 filename.txt` (would be the same as) `chmod -rwxr----- filename.txt`
jane_doe@u1804:\~$ `chmod 770 filename.txt` (would be the 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

`/`

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

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`

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

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*

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)*
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`

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`

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

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`

RAID - Redundant Array of Inexpensive Disks

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

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

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

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

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

(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

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

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

hardware enablement updates

Debian vs Snap

Installing & removing software

Searching for packages

managing repositories

Backing up & restoring Debian packages

Cleaning up orphaned apt packages

Making use of Aptitude

package management

package management

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

PS Command

Managing jobs

misbehaving processes

htop

system processes

Monitoring memory usage

scheduling Tasks with Cron

load average

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

Planning an IP address scheme

Serving IP addresses with isc-dhcp-server

Setting up DNS with bind

Creating a secondary (slave) DNS server

Setting up an internet gateway

Keeping your clock in sync with NTP

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

File server considerations

Sharing files with Windows users via Samba

Setting up NFS shares

Transferring files with `Rsync`

Transferring files with `SCP`

Mounting remote directories with SSHFS

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

Preparations for setting up a database server

Installing MariaDB

MariaDB configuration

Managing MariaDB databases

Setting up a slave database server

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

Installing additional Apache modules

Securing Apache with SSL

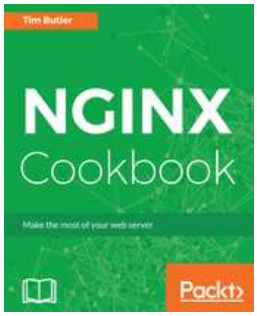
Installing and configuring NGINX

Setting up failover with keepalived

Setting up and configuring Nextcloud

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 11. Shell Techniques

Learning Advanced Shell Techniques

Understanding the Linux shell

Bash history

some useful command-line tricks

Redirecting output

Understanding variables

Writing simple scripts

Putting it all together: Writing an rsync backup script

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

Chapter 12. Virtualization

Setting up a virtual machine server

Creating virtual machines

Bridging the virtual machine network

Simplifying virtual machine creation with cloning

Managing virtual machines via the command line

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 .

Chapter 13. Running Containers

What is containerization?

Understanding the differences between Docker and LXD

Installing Docker

Managing Docker containers

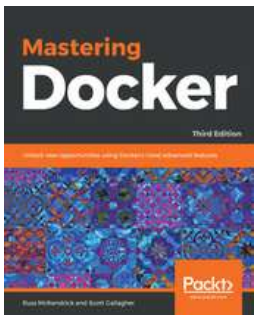
Automating Docker image creation with Dockerfiles

Managing LXD containers

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 14. Ansible

Automating Server Configuration with Ansible

Understanding the need for configuration management

Why Ansible?

Creating a Git repository

Getting started with Ansible

Making your servers do your bidding

Putting it all together – Automating web server deployment

Using Ansible's pull method

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

Chapter 15. Securing Your Server

Lowering your attack surface

Understanding and responding to CVEs

Installing security updates

Automatically installing patches with the Canonical Livepatch service

Monitoring Ubuntu servers with Canonical's Landscape service

Securing OpenSSH

Installing and configuring Fail2ban

MariaDB best practices for secure database servers

Setting up a firewall

Encrypting and decrypting disks with LUKS

Locking down sudo

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

Chapter 16. Troubleshooting Ubuntu Servers

Evaluating the problem space

Conducting a root cause analysis

Viewing system logs

Tracing network issues

Troubleshooting resource issues

Diagnosing defective RAM

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

Chapter 17. Preventing and Recovering from Disasters

Preventing disasters

Utilizing Git for configuration management

Implementing a backup plan

Replacing failed RAID disks

Utilizing bootable recovery media

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`