ubuntu

linux
system & network administration

Ömer Berat Sezer

PhD | Software Engineer | DevOps |
Solution Architect

linkedin.com/in/omerberatsezer/

github.com/omerbsezer

Why Linux System, Network Administration?

- Linux powers 90% of cloud infrastructure, servers, and embedded systems.
- Servers of 70-90% all over the world are Linux.
- You should learn Linux System, Network to control infrastructure on-prem and on cloud.

User and Group Management Basics

- Users: Individual accounts with unique IDs to access the system
- Groups: Logical collections of users to manage permissions.
- Core Commands:
  - useradd, usermod, userdel
  - groupadd, groupmod, groupdel
  - passwd for setting passwords.

User and Group Files

- /etc/passwd: Contains user account details.
- /etc/group: Lists group details.
- /etc/shadow: Stores encrypted passwords.

```
user@ubuntu: /$ cat /etc/group | grep groupname
# search groupname in the groups
user@ubuntu: /$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
user:x:1000:1000:,,,:/home/user:/bin/bash
```

User Commands

```
user@ubuntu: /$ sudo useradd -m username
# add a user
user@ubuntu: /$ sudo passwd username
# set password
user@ubuntu: /$ sudo usermod -aG sudo username
 # modify user
user@ubuntu: /$ sudo userdel -r username
# delete user
```

linkedin.com/in/omerberatsezer

Group Commands

```
user@ubuntu: /$ sudo groupadd developers
# add a group with ‘developers’
user@ubuntu: /$ sudo usermod -aG developers username
# add user to group
user@ubuntu: /$ groups username
# view group membership
user@ubuntu: /$ sudo groupdel developers
# delete a group
```

Permission Basics

Read (r), Write (w), Execute (x)
Levels: Owner, Group, Others

```
user@ubuntu: /$ ls  -la
drwxr-xr-- 2 user group 4096 Dec 25 14:00 myfolder
# Owner: Read, write, execute.
# Group: Read, execute.
#  Others: Read only.
user@ubuntu: /$ ls -l /path/to/directory
```

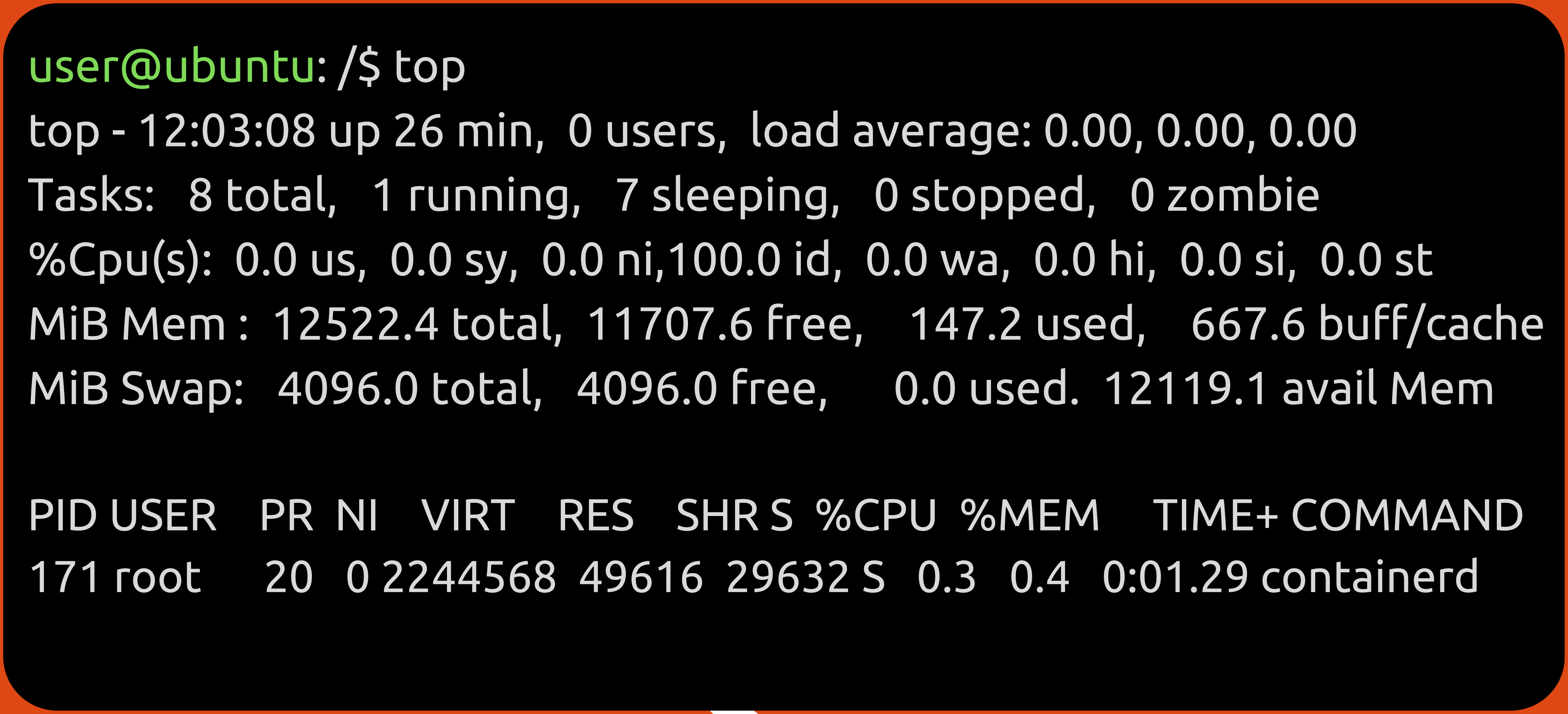
Permission Commands

- Change permissions: chmod
- Change ownership: chown

```
user@ubuntu: /$ chmod 755 file.txt
# (Owner: rwx, Group: r-x, Others: r-x)
user@ubuntu: /$ chown username:developers file.txt
user@ubuntu: /$ chmod +x script.sh
```

Process Monitoring

- Real-time resource usage: top/htop
- List running processes: ps



```
user@ubuntu: /$ top
top - 12:03:08 up 26 min,  0 users,  load average: 0.00, 0.00, 0.00
Tasks:   8 total,   1 running,   7 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  12522.4 total,  11707.6 free,    147.2 used,    667.6 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.  12119.1 avail Mem

PID USER     PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
171 root      20   0 2244568  49616  29632 S   0.3   0.4   0:01.29 containerd
```

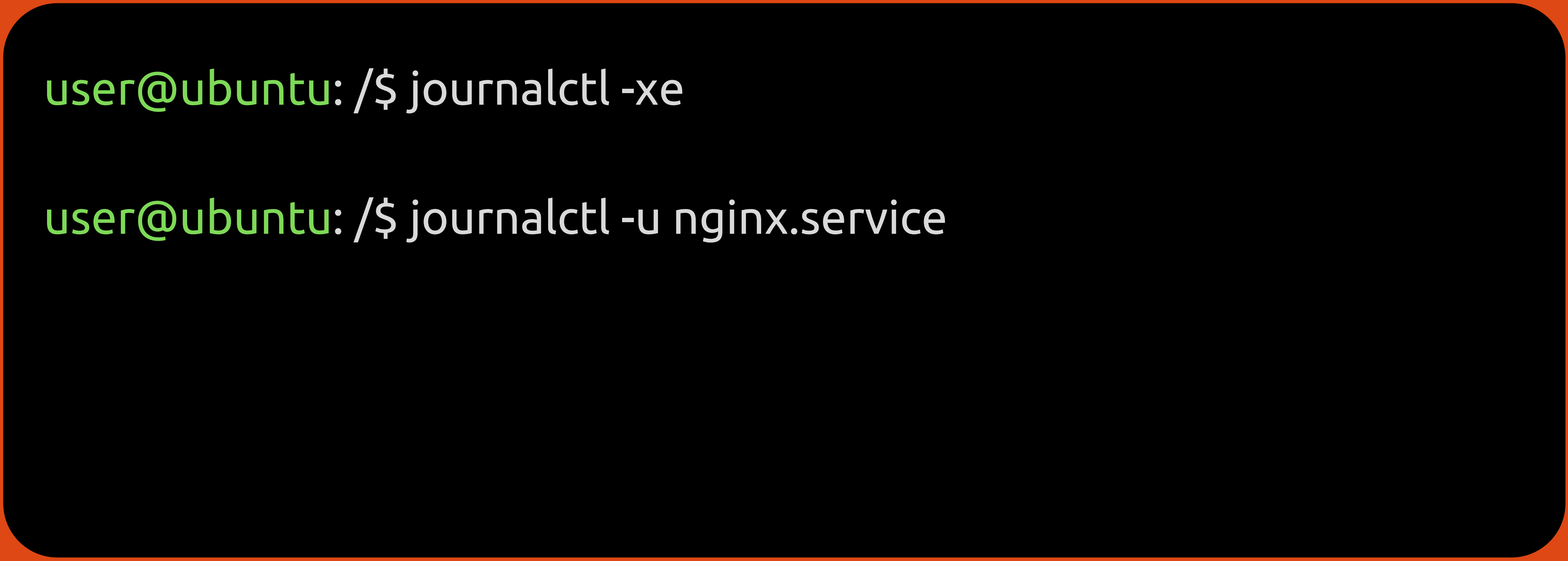
Disk Monitoring

- Disk space: df
- Directory size: du
- app for file/directory size: ncdu

```
user@ubuntu: /$ df -h
Filesystem     Size     Used   Avail    Use%   Mounted on
/dev/sdb       251G     20G  219G      9%         /
tmpfs          6.2G       0     6.2G      0%        /mnt
user@ubuntu: /$ du -sh /var/log
user@ubuntu: /$ ncdu
# traverse between directories
```

Log Monitoring

- View system logs: journalctl
- Kernel logs: dmesg
- App logs: /var/log/app.log

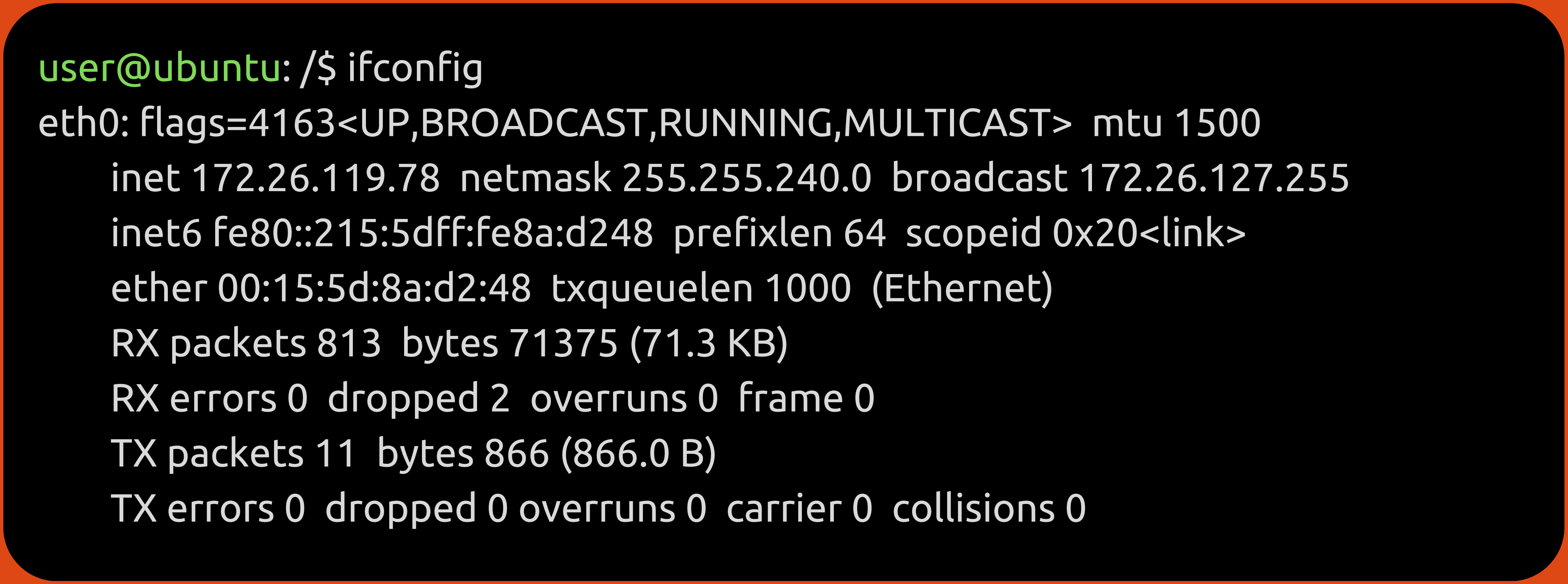


```
user@ubuntu: /$ journalctl -xe

user@ubuntu: /$ journalctl -u nginx.service
```

Network Basics

- Check network interfaces: ip a, ifconfig
- Check routing table: ip route
- Test connectivity: ping google.com

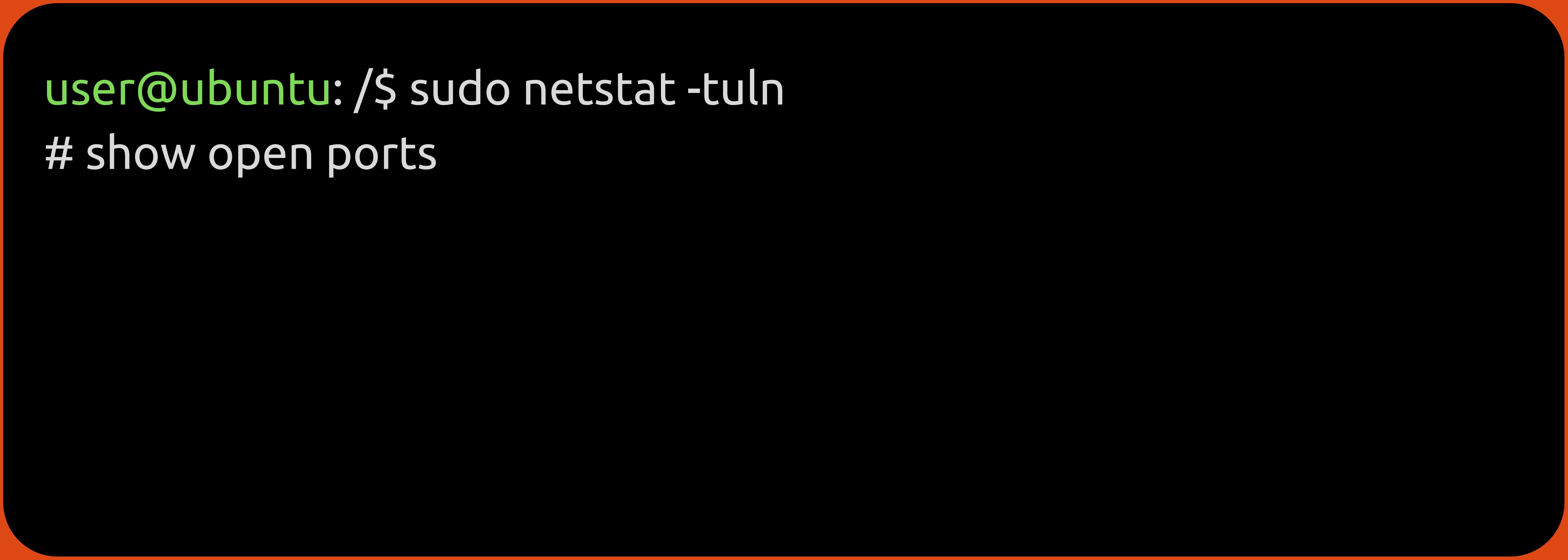


```
user@ubuntu: /$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 172.26.119.78  netmask 255.255.240.0  broadcast 172.26.127.255
      inet6 fe80::215:5dff:fe8a:d248  prefixlen 64  scopeid 0x20<link>
      ether 00:15:5d:8a:d2:48  txqueuelen 1000  (Ethernet)
      RX packets 813  bytes 71375 (71.3 KB)
      RX errors 0  dropped 2  overruns 0  frame 0
      TX packets 11  bytes 866 (866.0 B)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

linkedin.com/in/omerberatsezer

Network Monitoring

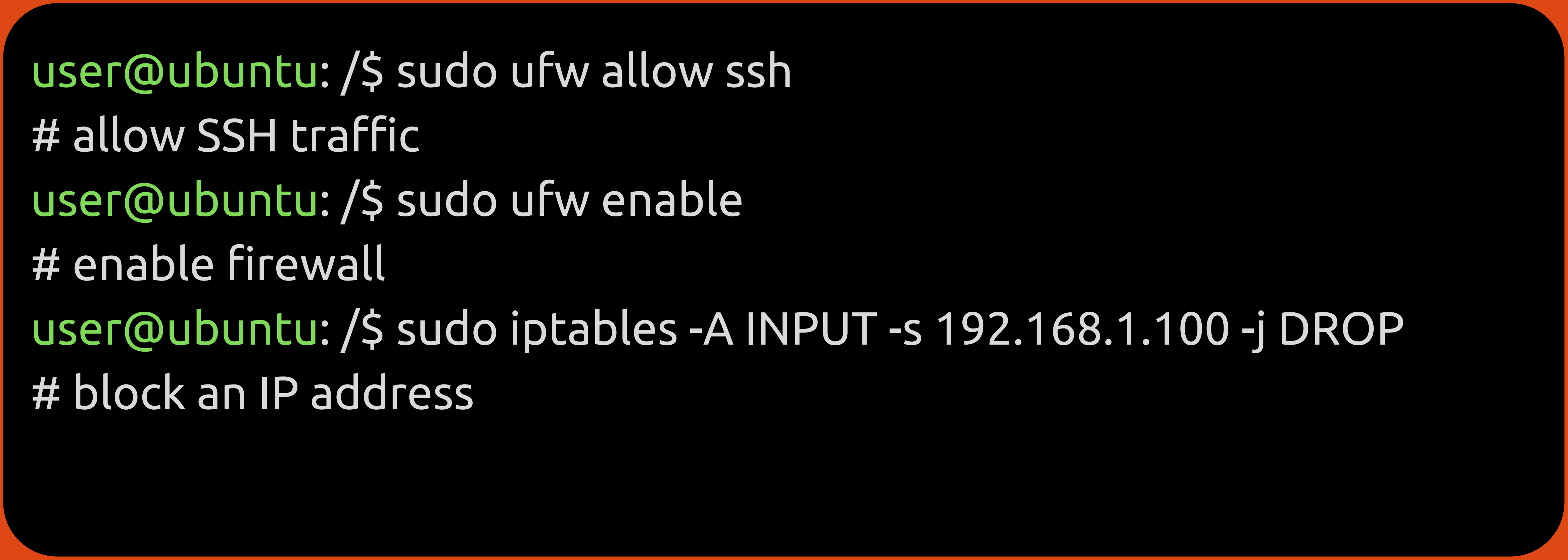
- Active connections: netstat/ss
- Bandwidth usage: iftop
- Open ports and services: nmap



```
user@ubuntu: /$ sudo netstat -tuln
# show open ports
```

Firewall Config

- Enable firewall: ufw enable
- Check firewall status: ufw status
- List existing rules: sudo iptables -L -v



```
user@ubuntu: /$ sudo ufw allow ssh
# allow SSH traffic
user@ubuntu: /$ sudo ufw enable
# enable firewall
user@ubuntu: /$ sudo iptables -A INPUT -s 192.168.1.100 -j DROP
# block an IP address
```

Package Management Basics

- Debian-based: apt (e.g., Ubuntu).
- Red Hat-based: yum/dnf (e.g., CentOS, Fedora).
- Others: pacman, snap, zypper.

```
user@ubuntu: /$ sudo apt install nginx
# install nginx
user@ubuntu: /$ sudo apt-get install nginx
# install nginx
user@ubuntu: /$ sudo apt remove nginx
# install nginx
user@ubuntu: /$ sudo apt purge nginx
# remove the package and its associated
```

Package Management Basics

```
user@ubuntu: /$ sudo apt update
# update package index
user@ubuntu: /$ sudo apt upgrade
# upgrade installed packages
user@ubuntu: /$ sudo apt install nginx
# install a package
user@ubuntu: /$ sudo apt remove nginx
# remove a package
```

Automation Tools

- Automate routine tasks like backups, updates, and configuration.
- Tools: Ansible, Puppet, Chef.
- Multiple remote control without agent on node => Ansible

```
- hosts: all
  tasks:
    - name: Update all packages
      apt:
        update_cache: yes
        upgrade: dist
```

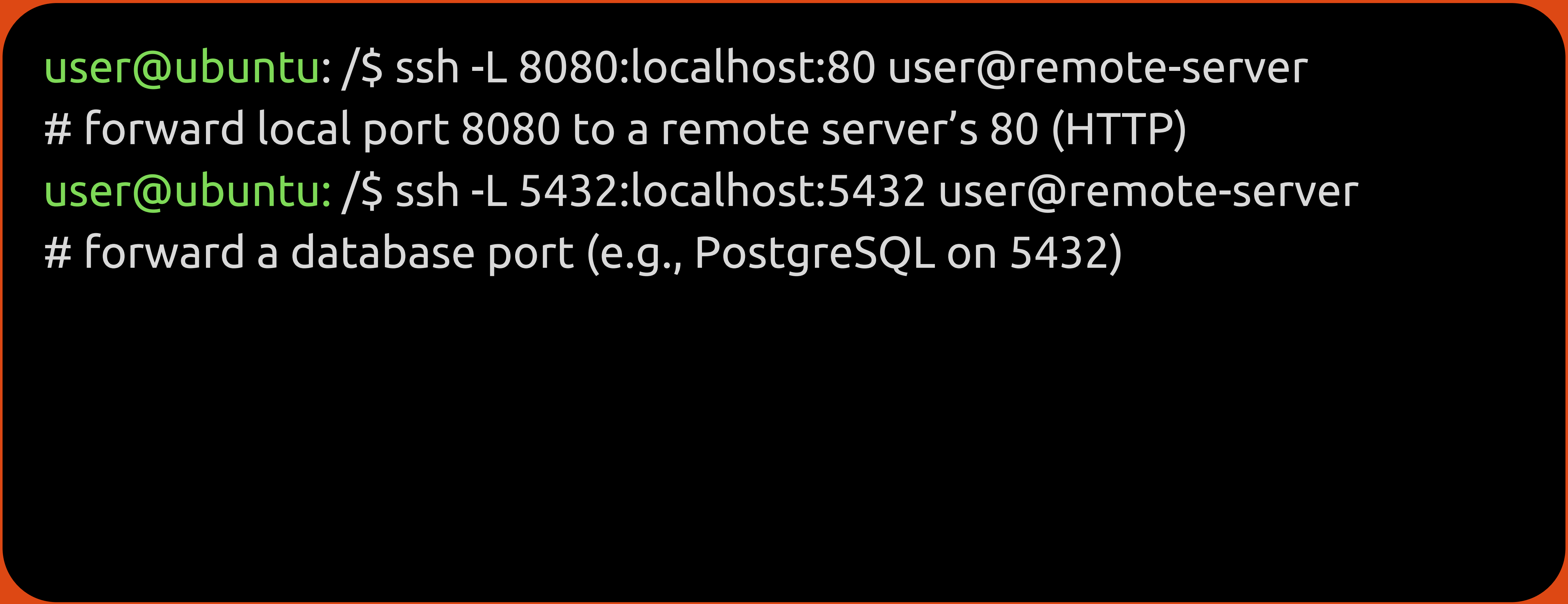
Secure Remote Access

- Connect to a remote server: ssh
- Copy a file to a remote server using SCP: scp
- Generate SSH keys: ssh-keygen

```
user@ubuntu: /$ ssh user@remote-server
# connect to a remote server
user@ubuntu: /$ ssh-keygen -t rsa -b 4096
# generate SSH keys for passwordless login
user@ubuntu: /$ ssh-copy-id user@remote-server
# copy SSH key to a remote server
user@ubuntu: /$ scp -r local/myfolder user@remote-server:/home/user/
# copy a directory recursively
```

SSH Local Port Forwarding

- To reach remote another port over SSH Port 22
- Traffic goes over the SSH tunneling



```
user@ubuntu: /$ ssh -L 8080:localhost:80 user@remote-server
# forward local port 8080 to a remote server’s 80 (HTTP)
user@ubuntu: /$ ssh -L 5432:localhost:5432 user@remote-server
# forward a database port (e.g., PostgreSQL on 5432)
```

Managing Services (systemctl)

- A service runs in the background without user interaction, often managing system functions
- e.g. ssh, nginx,

```
user@ubuntu: /$ systemctl status nginx
# check the status of a service
user@ubuntu: /$ sudo systemctl start nginx
# start/stop a service
user@ubuntu: /$ sudo systemctl enable nginx
# enable a service to start at boot
user@ubuntu: /$ sudo systemctl restart nginx
# restart a service
```

```
user@ubuntu:$  ######################
```

Follow for Tips on AWS, K8s, Docker, Linux
Ansible, DevOps, AI/ML

||
V

http://linkedin.com/in/omerberatsezer

http://github.com/omerbsezer

If you like, please share / repost / like
more people can view it

```
user@ubuntu:$ ######################
```