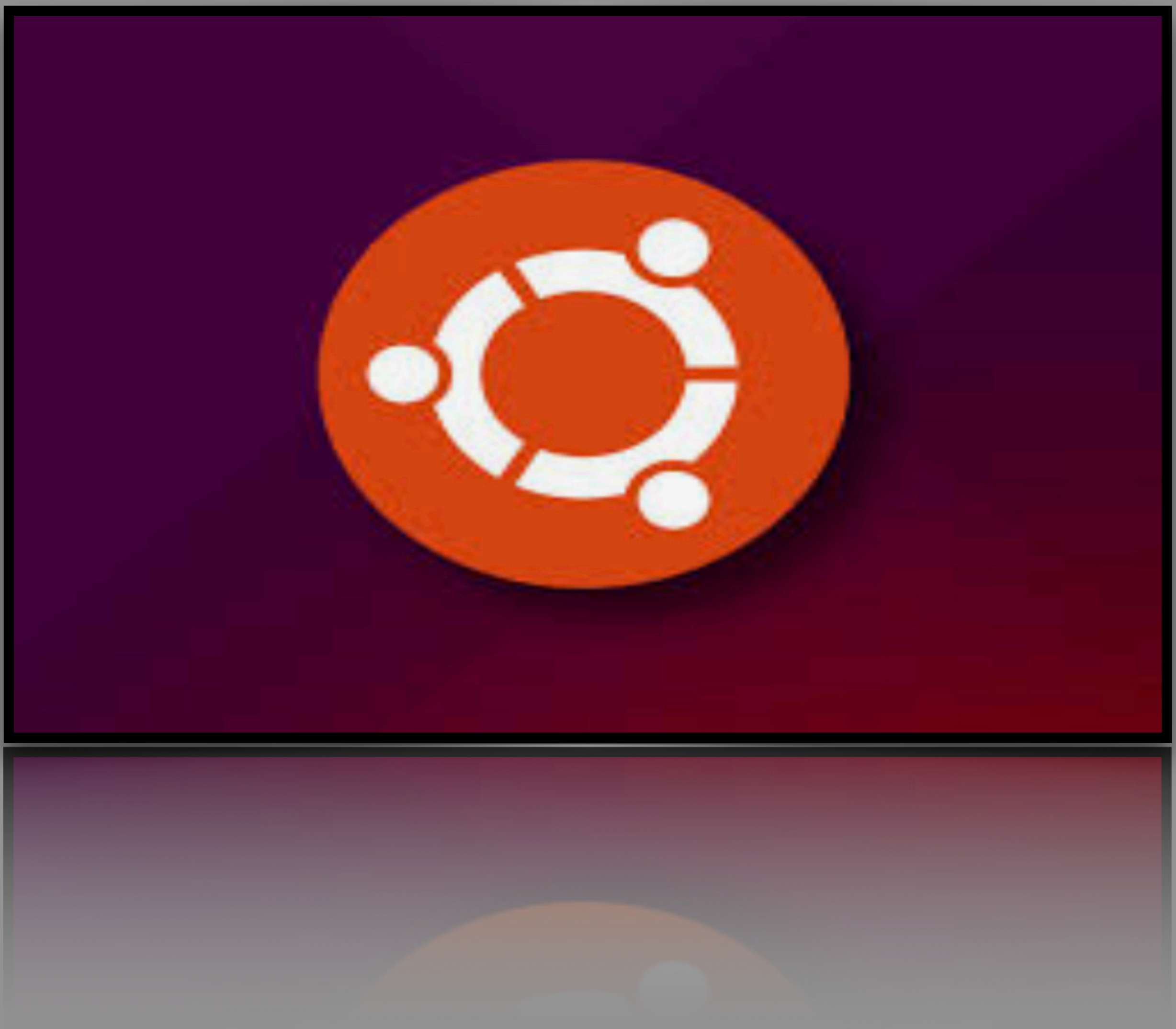
UBUNTU SERVER



ADD USER

```
$ apt update
$ apt upgrade

# Create Sudo User
$ adduser (username)
$ usermod -aG sudo (username)

# Change from root to sudo user
$ su - (username)

# Unlock the root account. Root access is rarely neeeded with the sudo command.
$ sudo passwd root
# Set new password

# Re-Lock the root account to remove the password and lock root:
$ sudo passwd -dl root
```

```
# Go Back to root
$ exit
```

SSH SECURITY

If the Root Account Uses SSH Key Authentication

If you logged in to your root account using SSH keys, then password authentication is disabled for SSH. You will need to add a copy of your local public key to the new user's ~/.ssh/authorized_keys file to log in successfully.

Since your public key is already in the root account's ~/.ssh/authorized_keys file on the server, we can copy that file and directory structure to our new user account in our existing session.

The simplest way to copy the files with the correct ownership and permissions is with the rsync command. This will copy the root user's .ssh directory, preserve the permissions, and modify the file owners, all in a single command. Make sure to change the highlighted portions of the command below to match your regular user's name:

```
    $ rsync --archive --chown=sammy:sammy ~/.ssh /home/
sammy
```

Now, open up a new terminal session and using SSH with your new username:

```
    $ ssh sammy@your_server_ip
```

You should be logged in to the new user account without using a password. Remember, if you need to run a command with administrative privileges, type sudo before it like this:

```
    $ sudo command_to_run
```

You will be prompted for your regular user password when using sudo for the first time each session (and periodically afterwards).

```
$ sudo ufw app list
$ sudo ufw allow OpenSSH
$ sudo ufw status
```

```
# Add Public Key authentication
$ cd .ssh & nano authorized_keys

# Disable root login
$ sudo nano /etc/ssh/sshd_config

# Disable password authentication
$ nano /etc/ssh/sshd_config

# Replace port 22 with a port between 1024 and 65536
$ nano /etc/ssh/sshd_config

# Configure your firewall to allow access to port 22s replacement
$ ufw allow 1024/UDP

$ systemctl restart sshd

# Add Fail2Ban and Portknocking to round out security.
```

CREATE BANNERS AT SYSTEM STARTUP

```
# How to display Banner in Command Line Terminal

$ nano /etc/ssh/sshd_config file.

# Add Link to Banner in config  /etc/ssh/my_banner

# Make sure you create a new file called:
$ nano /etc/ssh/my_banner file.
# Copy and paste contents of ASCII art into file.

# Reload sshd service. For instance:
$ systemctl reload ssh.service

$ apt install figlet toilet
$ figlet -c -k AGRI
```

SPEED-TEST

```
$ apt install speedtest-cli
$ speedtest
```

HISTORY COMMAND

```
# history command shows a list of last executed commands
$ history

# history command with corresponding timestamp
$ export HISTTIMEFORMAT='%F %T  '

# ignore duplicate commands entry made by user
$ export HISTCONTROL=ignoredups
```

WICKIT

```
# View Wikipedia summaries from the Command Line
$ apt install nodejs
$ npm install wikit -g

#Example. $ wikit Ubuntu
```

NETWORKING TOOLS

```
$ apt install net-tools
$ ifconfig
$ hostnamectl
$ ip addr show | grep inet
$ aria2 – downloading just about everything.
$ arpwatch – Ethernet Activity Monitor.
$ bmon – bandwidth monitor and rate estimator.
$ bwm-ng – live network bandwidth monitor.
$ curl – transferring data with URLs. (or try httpie)
$ darkstat – captures network traffic, usage statistics.
$ dhclient – Dynamic Host Configuration Protocol Client
$ dig – query DNS servers for information.
$ dstat – replacement for vmstat, iostat, mpstat, netstat
$ ethtool – controlling network drivers and hardware.
$ gated – gateway routing daemon.
$ host – DNS lookup utility.
$ hping – TCP/IP packet assembler/analyzer.
$ ibmonitor – shows bandwidth and total data transferred.
$ ifstat –  report network interfaces bandwidth.
$ iftop – display bandwidth usage.
```

$ ip – a command with more features than ifconfig

$ iperf3 – network bandwidth measurement tool

$ iproute2 – collection of utilities for controlling TCP/IP.

$ iptables – take control of network traffic.

$ IPTraf – An IP Network Monitor.

$ iputils – set of small useful utilities for Linux networking.

$ iw – a new CLI configuration utility for wireless devices.

$ jwhois (whois) – client for the whois service.

$ “lsof -i” – reveal information about your network sockets.

$ mtr – network diagnostic tool.

$ net-tools – utilities include: arp, hostname, ifconfig, netstat, rarp, route, plipconfig, slattach, mii-tool, iptunnel and ipmaddr.

$ ncat – improved re-implementation of netcat.

$ netcat – network utility reading/writing network

$ nethogs – a small ‘net top’ tool.

$ Netperf – Network bandwidth Testing.

$ netplan – Netplan is a utility for easily configuring networking on a linux system.

$ netsniff-ng – Swiss army knife for daily Linux network plumbing.

$ netwatch – monitoring Network Connections.

$ ngrep – grep applied to the network layer.

$ nload – display network usage.

$ nmap – network discovery and security auditing.

$ nmcli – a command-line tool for controlling NetworkManager and reporting network status.

$ nmtui – provides a text interface to configure networking by controlling NetworkManager.

$ nslookup – query Internet name servers interactively.

$ ping – send icmp echo_request to network hosts.

$ route – show / manipulate the IP routing table.

$ slurm – network load monitor.

$ snort – Network Intrusion Detection and Prevention System.

$ smokeping – keeps track of your network latency.

$ socat – establishes two bidirectional byte streams and transfers data between them.

$ speedometer – Measure and display the rate of data across a network.

$ speedtest-cli – test internet bandwidth using speedtest.net

$ ss – utility to investigate sockets.

$ ssh – secure system administration and file transfers over insecure networks.

$ tcpdump – command-line packet analyzer.

$ tcptrack – Displays information about tcp connections on a network interface.

$ telnet – user interface to the TELNET protocol.

$ tracepath – very similar function to traceroute.

$ traceroute – print the route packets trace to network host.

$ vnStat – network traffic monitor.

$ websocat – Connection forwarder from/to web sockets to/from usual sockets, in style of socat.

$ wget – retrieving files using HTTP, HTTPS, FTP and FTPS.

Wireless Tools for Linux – includes iwconfig, iwlist, iwspy, iwpriv and ifrename.

$ Wireshark – network protocol analyzer.