Lab 7

Lab Title: Environment Variables, PATH, UFW, and SSH Key Authentication

Submitted to: Engr. Muhammad Shoaib

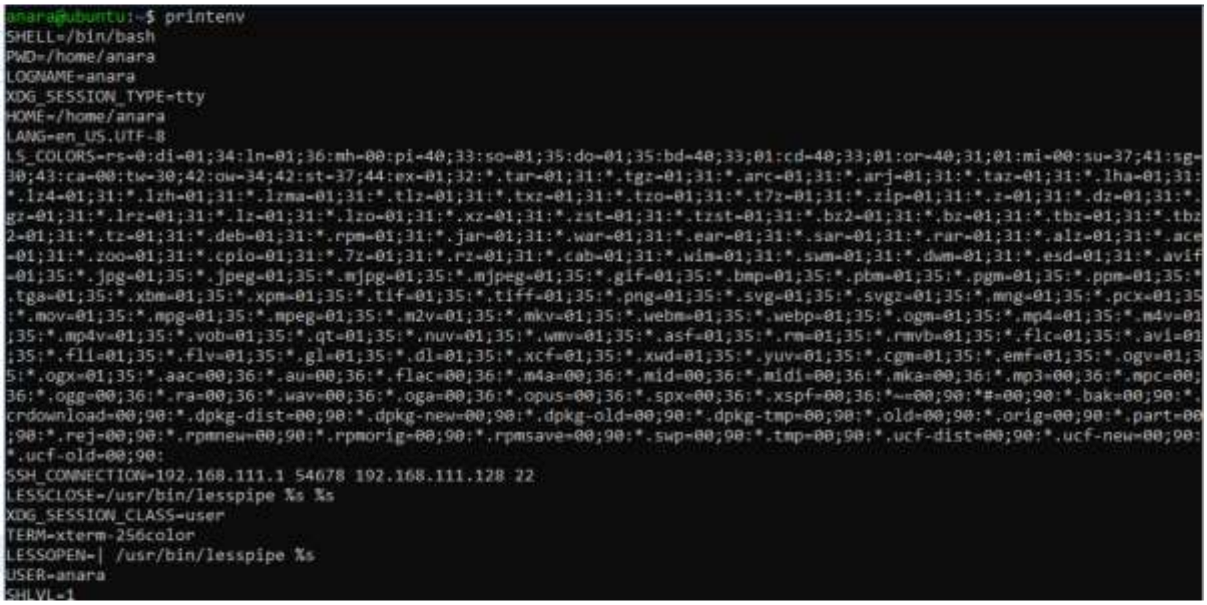
Course Title: Cloud Computing

Submitted by: Anara Hayat

Reg#No: 2023-BSE-008

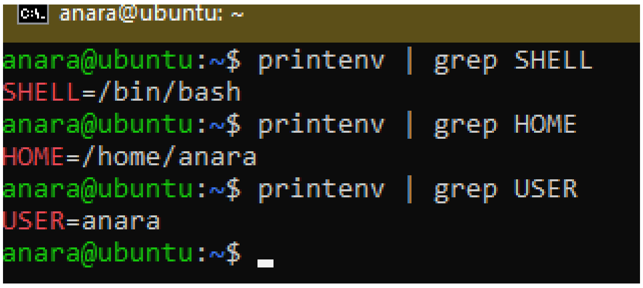
Task 1 — Print & filter environment variables

1. **Print all environment variables:**



```
anara@ubuntu:~$ printenv
SHELL=/bin/bash
PWD=/home/anara
LOGNAME=anara
XDG_SESSION_TYPE=tty
HOME=/home/anara
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=
30;43:ca=00:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:
*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.
gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz
2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace
=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.avif
=01;35:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*
.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35
:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01
;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01
;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;3
5:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;
36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:*~=00;90:*#=00;90:*.bak=00;90:*.
crdownload=00;90:*.dpkg-dist=00;90:*.dpkg-new=00;90:*.dpkg-old=00;90:*.dpkg-tmp=00;90:*.old=00;90:*.orig=00;90:*.part=00
;90:*.rej=00;90:*.rpmnew=00;90:*.rpmorig=00;90:*.rpmsave=00;90:*.swp=00;90:*.tmp=00;90:*.ucf-dist=00;90:*.ucf-new=00;90:
*.ucf-old=00;90:
SSH_CONNECTION=192.168.111.1 54678 192.168.111.128 22
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=anara
SHLVL=1
```

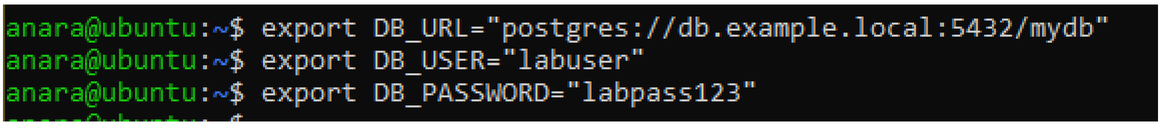
2. **Filter for SHELL, HOME and USER — run these greps together and capture one combined screenshot:**



```
anara@ubuntu:~$ printenv | grep SHELL
SHELL=/bin/bash
anara@ubuntu:~$ printenv | grep HOME
HOME=/home/anara
anara@ubuntu:~$ printenv | grep USER
USER=anara
anara@ubuntu:~$ 
```

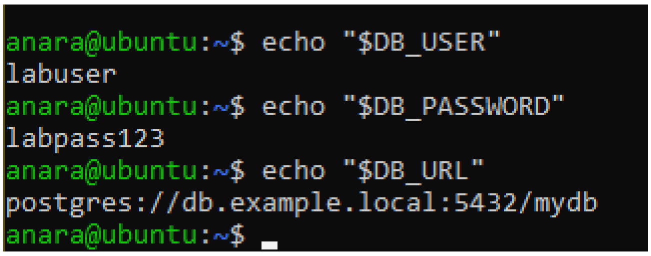
Task 2 — Export DB_* variables temporarily and observe scope

1. **Define all DB_* variables (run the three exports one after another). Capture them in one screenshot showing the three export commands and their execution:**



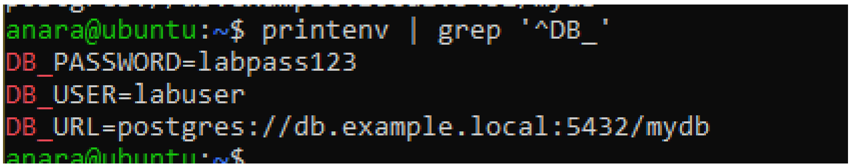
```
anara@ubuntu:~$ export DB_URL="postgres://db.example.local:5432/mydb"
anara@ubuntu:~$ export DB_USER="labuser"
anara@ubuntu:~$ export DB_PASSWORD="labpass123"
anara@ubuntu:~$
```

2. **Echo the three variables (run the three echo commands together) and capture one screenshot showing their outputs:**



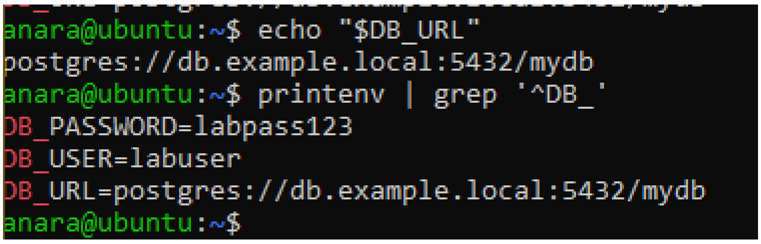
```
anara@ubuntu:~$ echo "$DB_USER"
labuser
anara@ubuntu:~$ echo "$DB_PASSWORD"
labpass123
anara@ubuntu:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
anara@ubuntu:~$ 
```

3. **Show all DB_ variables with a single grep command (capture that output):**



```
anara@ubuntu:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
anara@ubuntu:~$
```

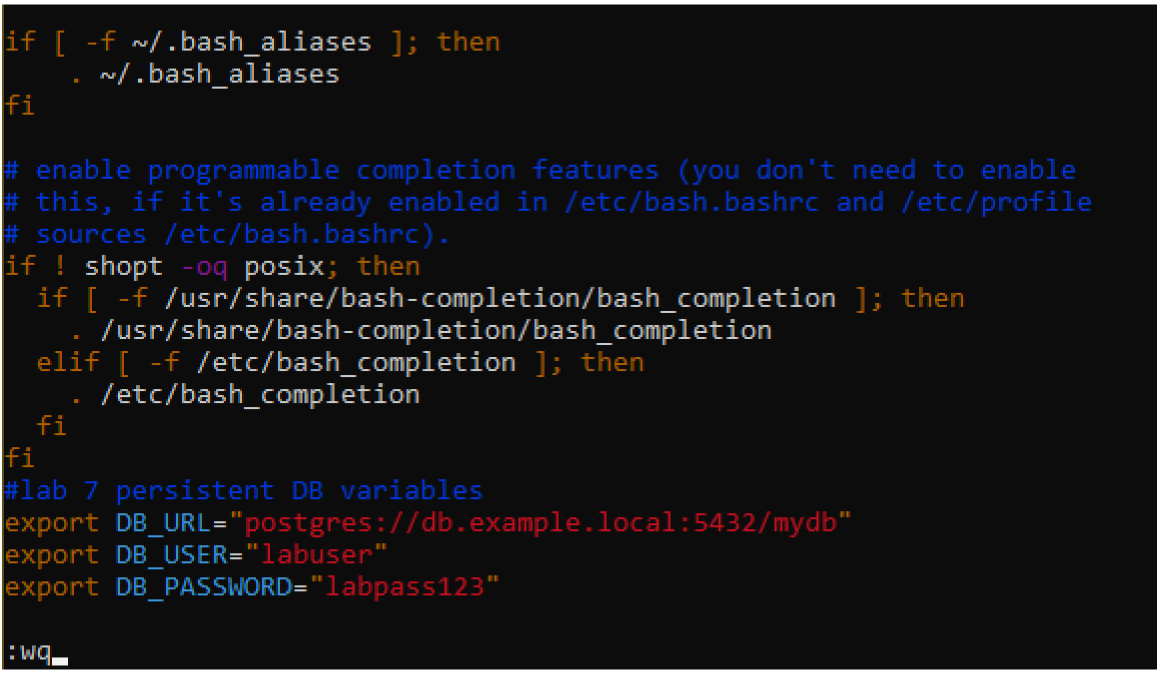
4. **Close the bash session (e.g., exit) and reopen a new terminal. Verify the variables are gone by running the echo(s) and the grep together; capture both checks in one screenshot:**



```
anara@ubuntu:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
anara@ubuntu:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
anara@ubuntu:~$
```

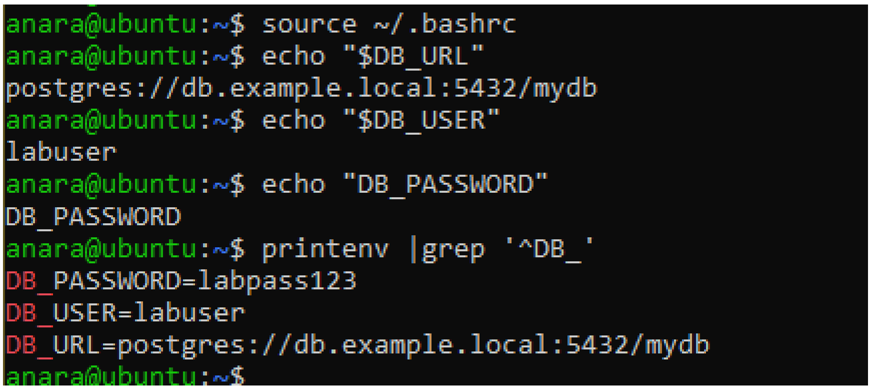
Task 3 — Make DB_* variables persistent in ~/.bashrc

1. **Open ~/.bashrc in an editor and append the three export lines. Capture the editor showing the three lines added (single screenshot):**



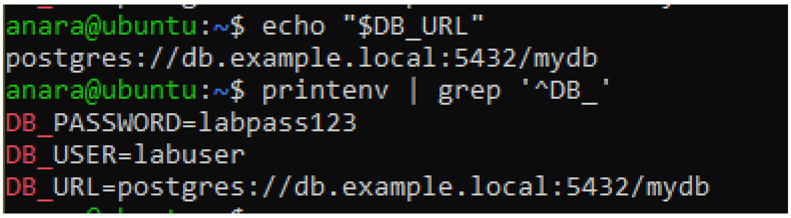
```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
#lab 7 persistent DB variables
export DB_URL="postgres://db.example.local:5432/mydb"
export DB_USER="labuser"
export DB_PASSWORD="labpass123"

:wq
```

2. Source ~/.bashrc and capture the source command in one screenshot together with the next verification commands (grouped): run source ~/.bashrc and then immediately run the three echoes and a single grep, capturing all of these in one screenshot:



```
anara@ubuntu:~$ source ~/.bashrc
anara@ubuntu:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
anara@ubuntu:~$ echo "$DB_USER"
labuser
anara@ubuntu:~$ echo "DB_PASSWORD"
DB_PASSWORD
anara@ubuntu:~$ printenv |grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
anara@ubuntu:~$
```

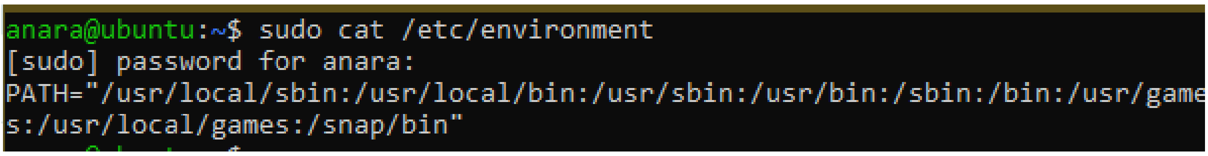
3. Close and reopen terminal. Verify persistence by running one echo and the grep together — capture both in one screenshot:



```
anara@ubuntu:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
anara@ubuntu:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

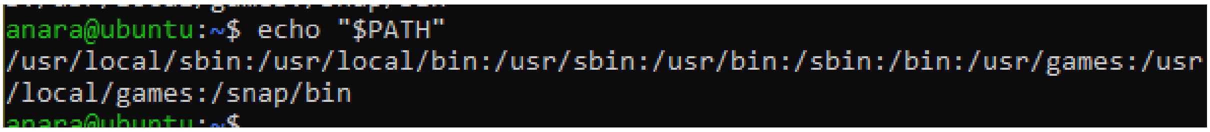
Task 4 — System-wide environment variable, welcome script, and PATH

1. View /etc/environment:



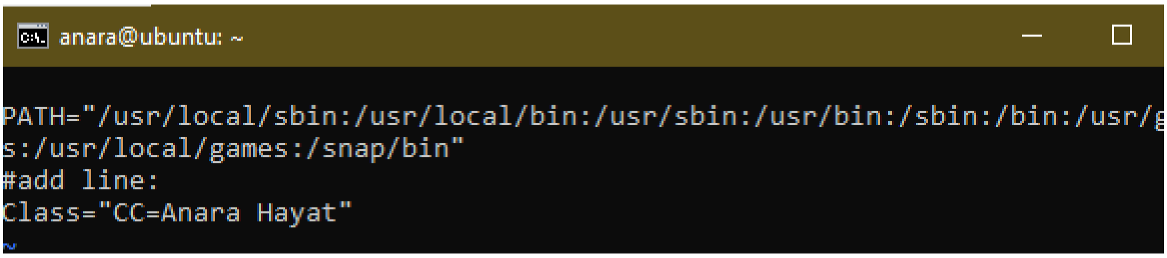
```
anara@ubuntu:~$ sudo cat /etc/environment
[sudo] password for anara:
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/game
s:/usr/local/games:/snap/bin"
```

2. Show current PATH:

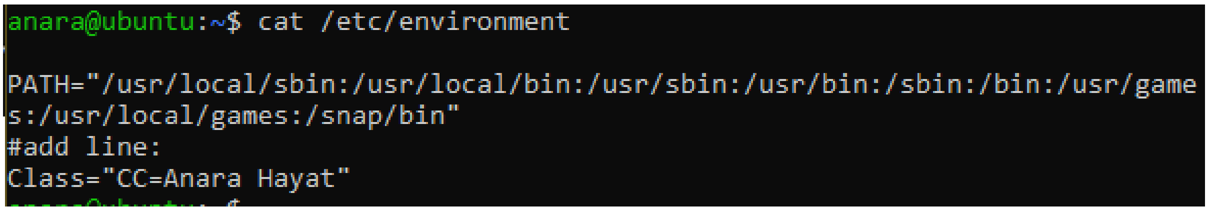


```
anara@ubuntu:~$ echo "$PATH"
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr
/local/games:/snap/bin
```

3. Edit /etc/environment and add Class:

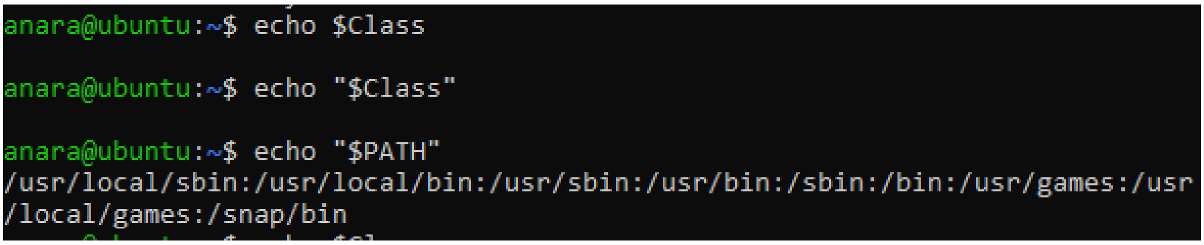


```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/g
s:/usr/local/games:/snap/bin"
#add line:
Class="CC=Anara Hayat"
~
```



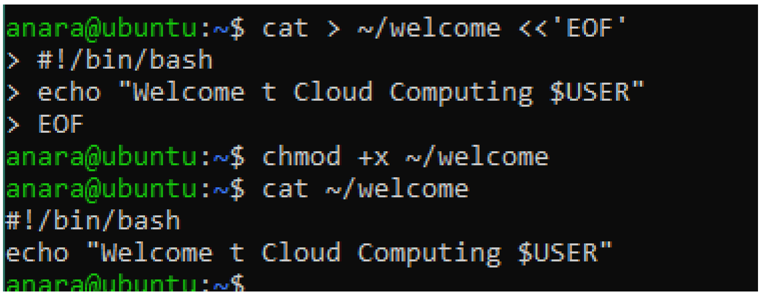
```
anara@ubuntu:~$ cat /etc/environment

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/game
s:/usr/local/games:/snap/bin"
#add line:
Class="CC=Anara Hayat"
```

4. **Re-login or open a new shell and show Class and PATH together (grouped prints): run echo $Class and echo $PATH together and capture in a single screenshot**



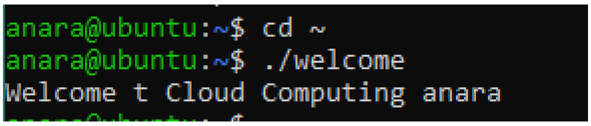
```
anara@ubuntu:~$ echo $Class

anara@ubuntu:~$ echo "$Class"

anara@ubuntu:~$ echo "$PATH"
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr
/local/games:/snap/bin
```

5. **Create welcome script at your home directory (~/welcome) and make it executable (capture the heredoc creation and chmod together in one screenshot if possible):**



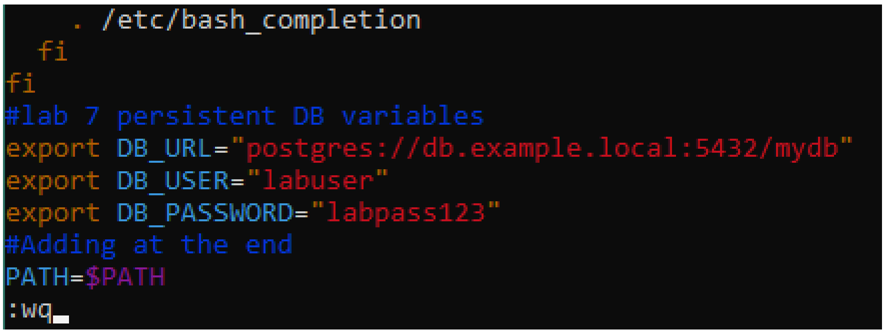
```
anara@ubuntu:~$ cat > ~/welcome <<'EOF'
> #!/bin/bash
> echo "Welcome t Cloud Computing $USER"
> EOF
anara@ubuntu:~$ chmod +x ~/welcome
anara@ubuntu:~$ cat ~/welcome
#!/bin/bash
echo "Welcome t Cloud Computing $USER"
anara@ubuntu:~$
```

6. **Run the script from your home directory using ./welcome:**



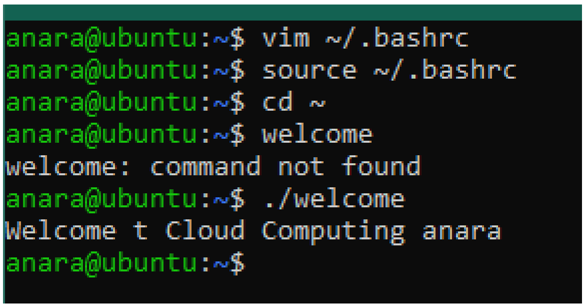
```
anara@ubuntu:~$ cd ~
anara@ubuntu:~$ ./welcome
Welcome t Cloud Computing anara
```

7. **Add your home directory to PATH in ~/.bashrc. NOTE: per your instruction we do not include an export PATH line here — only add the PATH modification line in the file. Capture the editor showing that PATH line in one screenshot:**



```
    . /etc/bash_completion
  fi
fi
#lab 7 persistent DB variables
export DB_URL="postgres://db.example.local:5432/mydb"
export DB_USER="labuser"
export DB_PASSWORD="labpass123"
#Adding at the end
PATH=$PATH
:wq
```

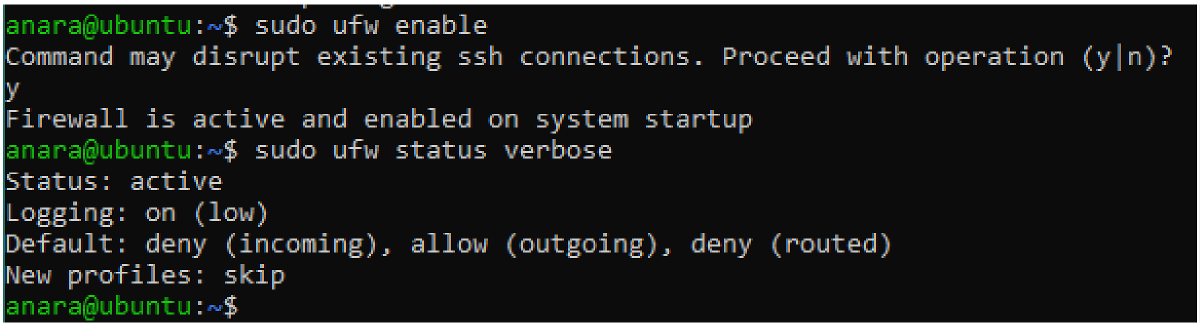
8. **Apply the change and run welcome — capture these runtime commands in a separate screenshot (must be taken separately from the editor screenshot):**



```
anara@ubuntu:~$ vim ~/.bashrc
anara@ubuntu:~$ source ~/.bashrc
anara@ubuntu:~$ cd ~
anara@ubuntu:~$ welcome
welcome: command not found
anara@ubuntu:~$ ./welcome
Welcome t Cloud Computing anara
anara@ubuntu:~$
```

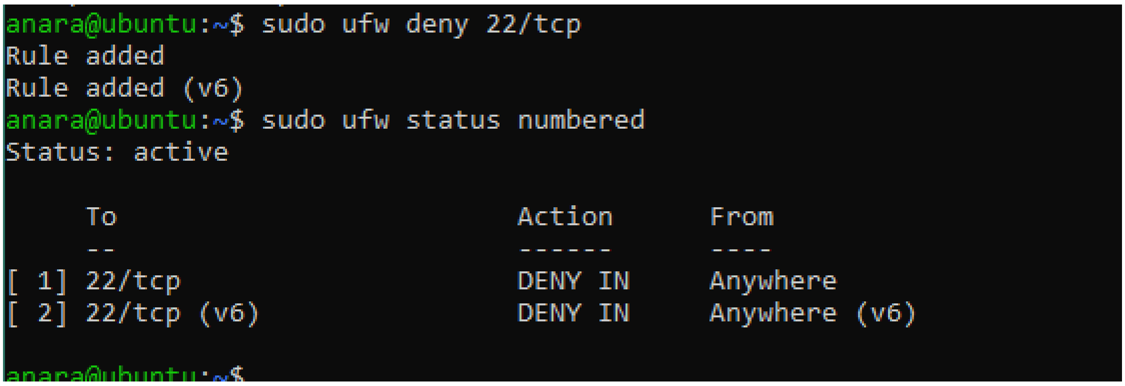
Task 5 — Block and allow SSH using ufw (firewall)

1. **Enable ufw and show status (group both commands in one screenshot if you run them together):**



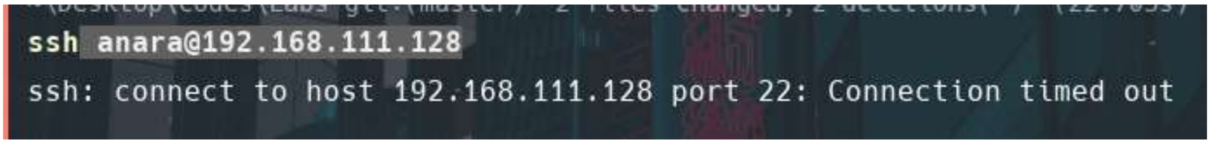
```
anara@ubuntu:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)?
y
Firewall is active and enabled on system startup
anara@ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip
anara@ubuntu:~$
```

2. **Deny TCP port 22 and show status (run deny and status numbered together and capture in one screenshot). Use short form as requested:**



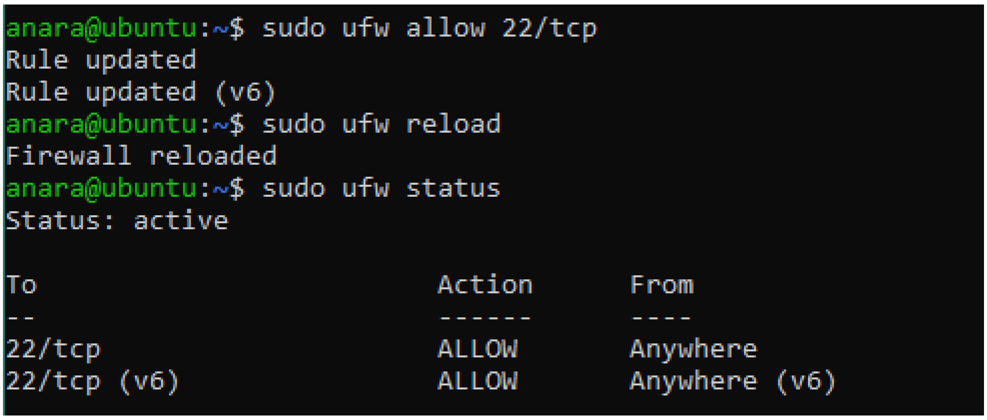
```
anara@ubuntu:~$ sudo ufw deny 22/tcp
Rule added
Rule added (v6)
anara@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     DENY IN     Anywhere
[ 2] 22/tcp (v6)                DENY IN     Anywhere (v6)

anara@ubuntu:~$
```

3. **From Windows host attempt to SSH (expected to fail) — capture the host-side SSH attempt in one screenshot:**



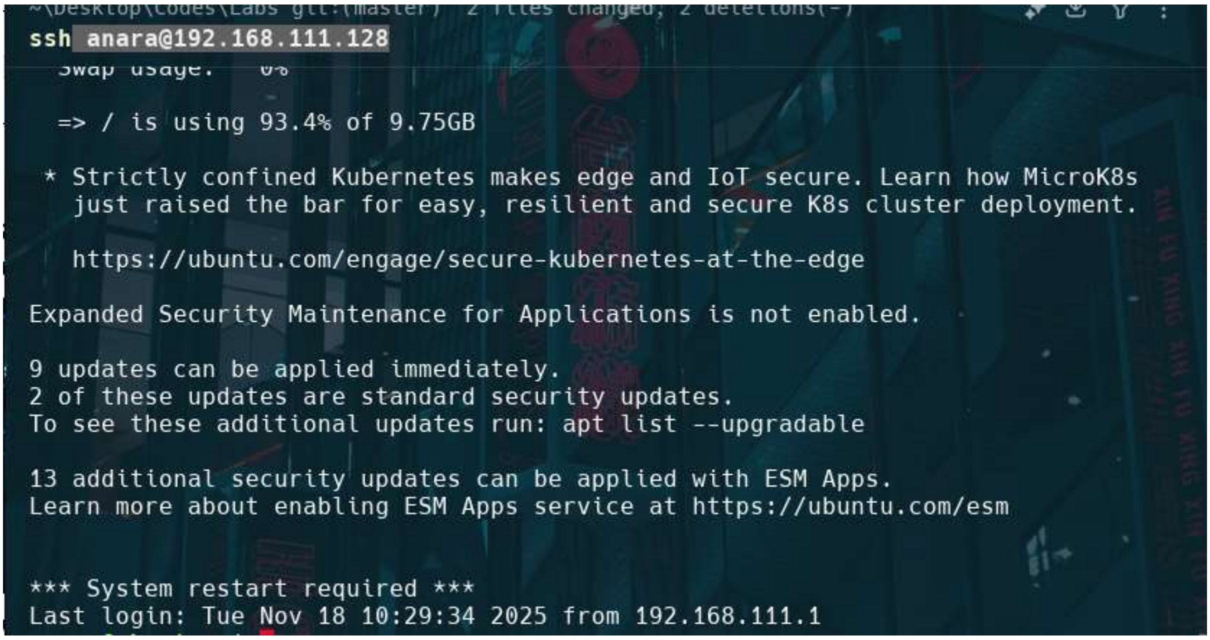
```
ssh anara@192.168.111.128
ssh: connect to host 192.168.111.128 port 22: Connection timed out
```

4. **Allow SSH back and reload, then show status (group allow, reload, status in one screenshot if run together). Use short form as requested:**



```
anara@ubuntu:~$ sudo ufw allow 22/tcp
Rule updated
Rule updated (v6)
anara@ubuntu:~$ sudo ufw reload
Firewall reloaded
anara@ubuntu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
```

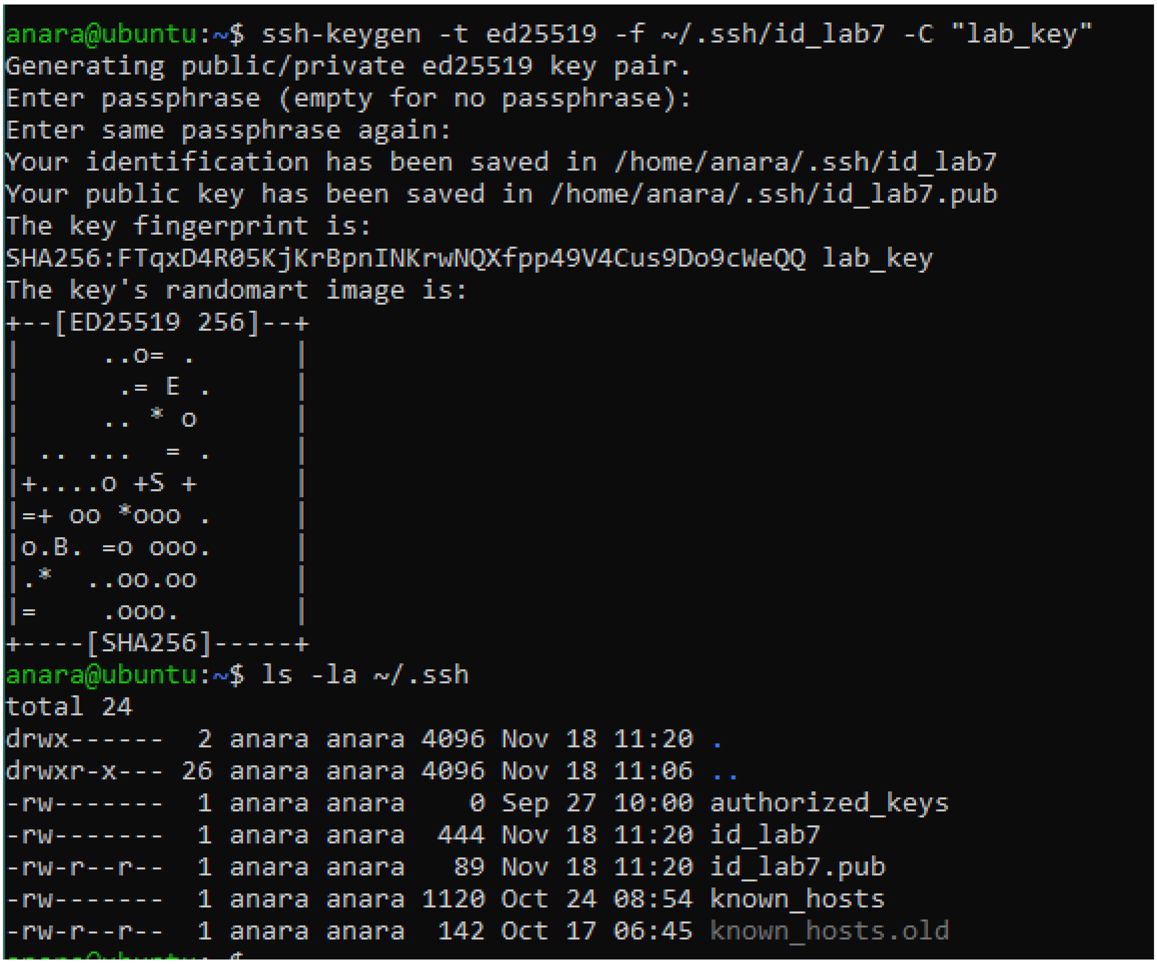
5. **From Windows host attempt SSH again (should succeed) — capture successful login in one screenshot:**



```
~\Desktop\Codes\Labs git:(master) 2 files changed, 2 deletions(-)
ssh anara@192.168.111.128
  Swap usage:   0%

  => / is using 93.4% of 9.75GB

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

13 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


*** System restart required ***
Last login: Tue Nov 18 10:29:34 2025 from 192.168.111.1
```

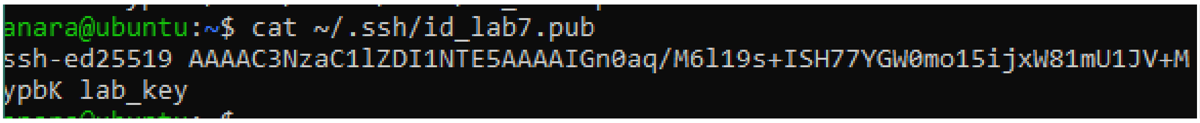
Task 6 — Configure SSH key-based login from Windows host

1. **Generate ed25519 key pair (if needed) and show the generated files in one screenshot (run ssh-keygen and then list ~/.ssh):**



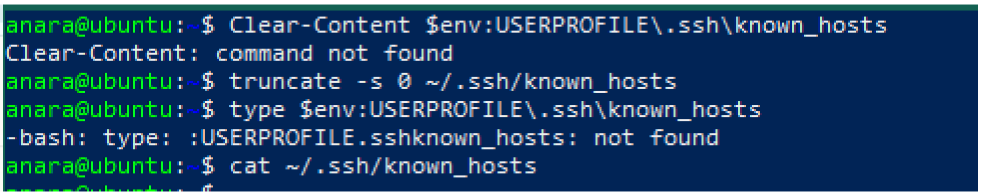
```
anara@ubuntu:~$ ssh-keygen -t ed25519 -f ~/.ssh/id_lab7 -C "lab_key"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/anara/.ssh/id_lab7
Your public key has been saved in /home/anara/.ssh/id_lab7.pub
The key fingerprint is:
SHA256:FTqxD4R05KjKrBpnINKrwNQXfpp49V4Cus9Do9cWeQQ lab_key
The key's randomart image is:
+--[ED25519 256]--+
|     ..o= .      |
|      .= E .     |
|     .. * o      |
| .. ...  = .     |
|+....o +S +      |
|=+ oo *ooo .     |
|o.B. =o ooo.     |
|.*  ..oo.oo      |
|=    .ooo.       |
+----[SHA256]-----+
anara@ubuntu:~$ ls -la ~/.ssh
total 24
drwx------  2 anara anara 4096 Nov 18 11:20 .
drwxr-x--- 26 anara anara 4096 Nov 18 11:06 ..
-rw-------  1 anara anara    0 Sep 27 10:00 authorized_keys
-rw-------  1 anara anara  444 Nov 18 11:20 id_lab7
-rw-r--r--  1 anara anara   89 Nov 18 11:20 id_lab7.pub
-rw-------  1 anara anara 1120 Oct 24 08:54 known_hosts
-rw-r--r--  1 anara anara  142 Oct 17 06:45 known_hosts.old
```

2. **Show the public key content (single screenshot):**



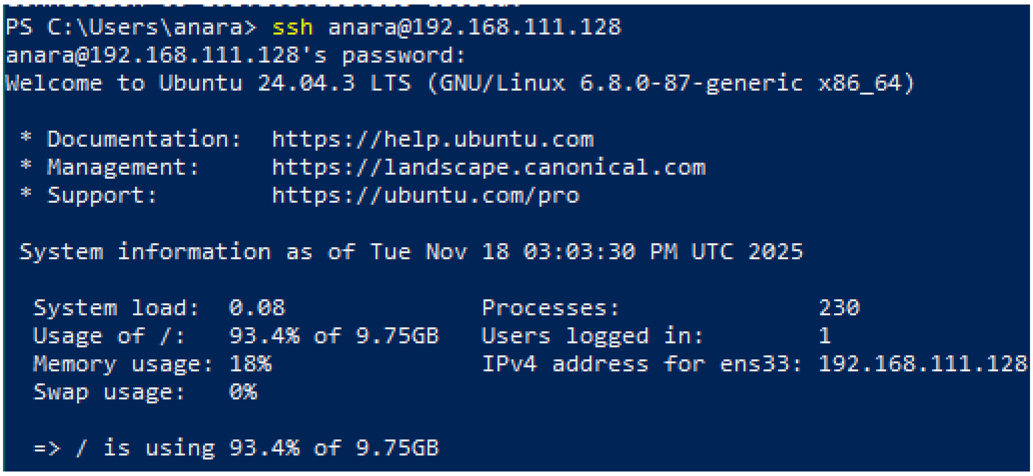
```
anara@ubuntu:~$ cat ~/.ssh/id_lab7.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGn0aq/M6l19s+ISH77YGW0mo15ijxW81mU1JV+M
ypbK lab_key
```

3. **Clear the known_hosts file content and verify it is empty (single screenshot):**



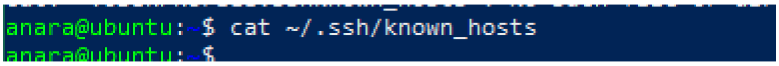
```
anara@ubuntu:~$ Clear-Content $env:USERPROFILE\.ssh\known_hosts
Clear-Content: command not found
anara@ubuntu:~$ truncate -s 0 ~/.ssh/known_hosts
anara@ubuntu:~$ type $env:USERPROFILE\.ssh\known_hosts
-bash: type: :USERPROFILE.sshknown_hosts: not found
anara@ubuntu:~$ cat ~/.ssh/known_hosts
```

4. **Connect to the Ubuntu server using the standard SSH command (this will prompt to accept the server host key because known_hosts is empty). Capture the connection prompt/accept step in one screenshot:**



```
PS C:\Users\anara> ssh anara@192.168.111.128
anara@192.168.111.128's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Nov 18 03:03:30 PM UTC 2025

  System load:  0.08               Processes:             230
  Usage of /:   93.4% of 9.75GB    Users logged in:       1
  Memory usage: 18%                IPv4 address for ens33: 192.168.111.128
  Swap usage:   0%

  => / is using 93.4% of 9.75GB
```

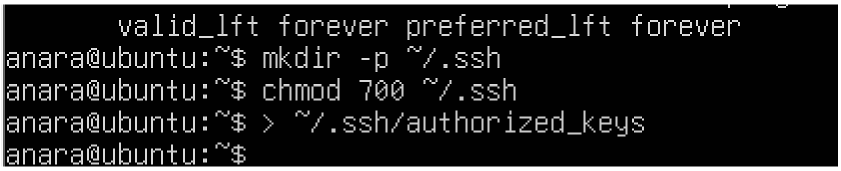
5. **After the successful connection, view the known_hosts file to show the server host key was added (single screenshot):**



```
anara@ubuntu:~$ cat ~/.ssh/known_hosts
anara@ubuntu:~$
```

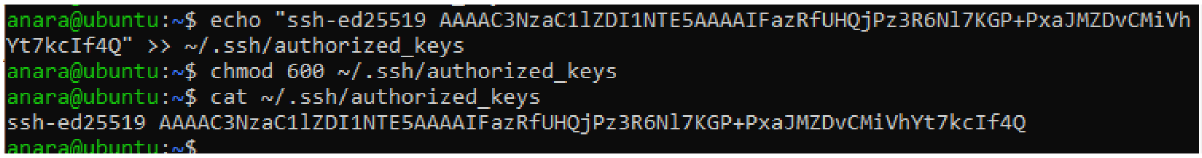
B. On Ubuntu server — group related server-side commands:

1. **Prepare the ~/.ssh directory and clear authorized_keys (this will create the directory if missing, set the correct directory permissions, and truncate the authorized_keys file). Capture this command sequence and its output in one screenshot:**



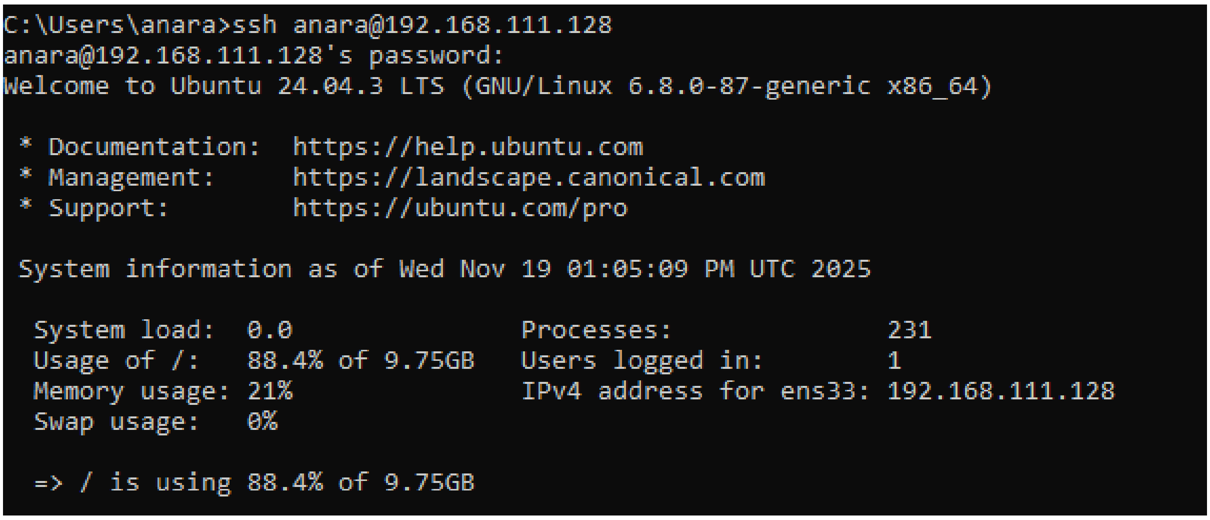
```
       valid_lft forever preferred_lft forever
anara@ubuntu:~$ mkdir -p ~/.ssh
anara@ubuntu:~$ chmod 700 ~/.ssh
anara@ubuntu:~$ > ~/.ssh/authorized_keys
anara@ubuntu:~$
```

2. **Append the public key, set file permissions, and show the resulting authorized_keys (capture commands and resulting file content in one screenshot):**



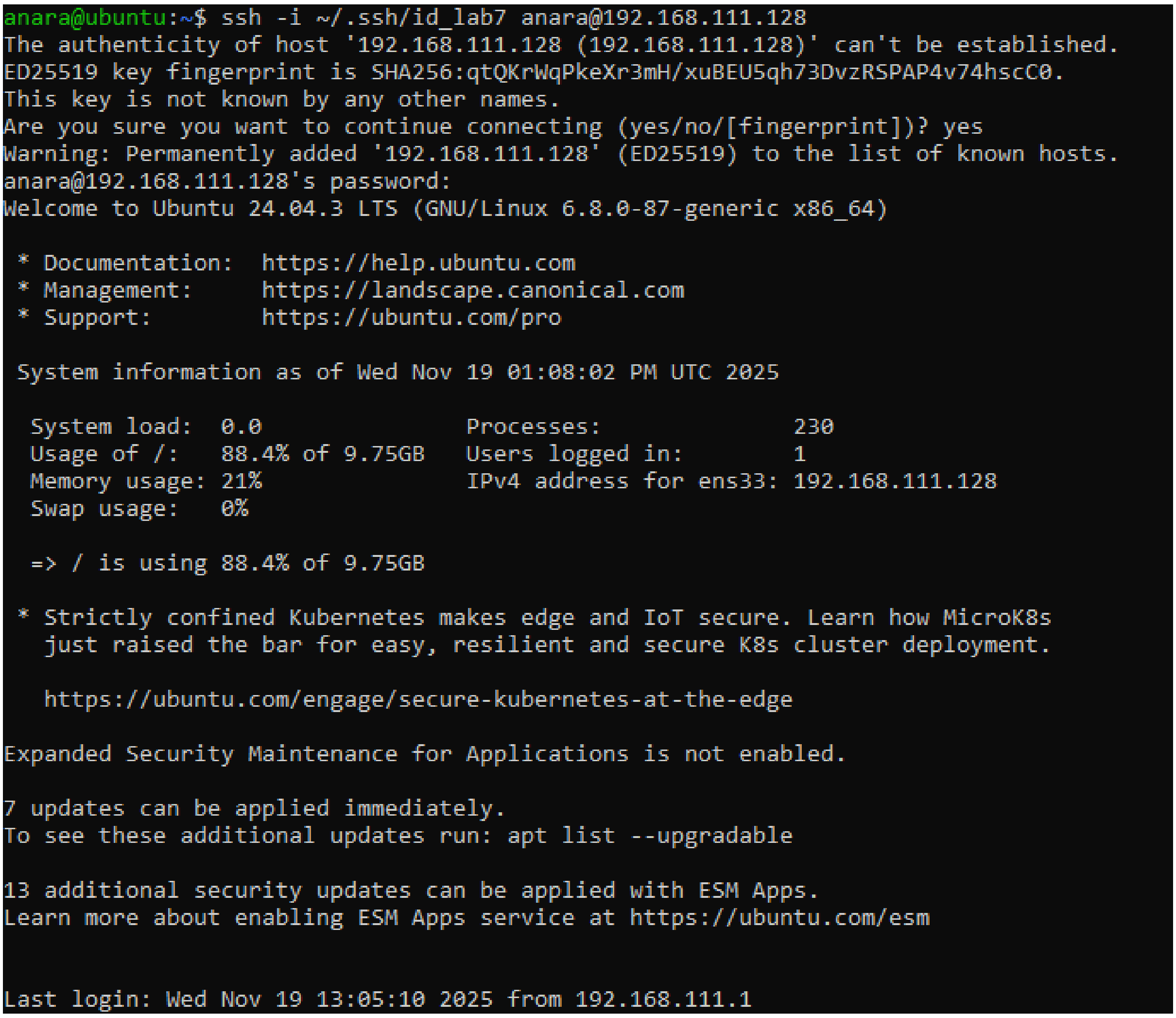
```
anara@ubuntu:~$ echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFazRfUHQjPz3R6Nl7KGP+PxaJMZDvCMiVh
Yt7kcIf4Q" >> ~/.ssh/authorized_keys
anara@ubuntu:~$ chmod 600 ~/.ssh/authorized_keys
anara@ubuntu:~$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFazRfUHQjPz3R6Nl7KGP+PxaJMZDvCMiVhYt7kcIf4Q
anara@ubuntu:~$
```

3. **From Windows host test passwordless login (capture successful login in one screenshot):**



```
C:\Users\anara>ssh anara@192.168.111.128
anara@192.168.111.128's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Nov 19 01:05:09 PM UTC 2025

  System load:  0.0               Processes:             231
  Usage of /:   88.4% of 9.75GB   Users logged in:       1
  Memory usage: 21%               IPv4 address for ens33: 192.168.111.128
  Swap usage:   0%

  => / is using 88.4% of 9.75GB
```

4. Also demonstrate explicit identity usage (single screenshot):

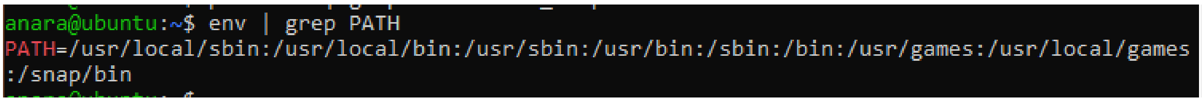


```
anara@ubuntu:~$ ssh -i ~/.ssh/id_lab7 anara@192.168.111.128
The authenticity of host '192.168.111.128 (192.168.111.128)' can't be established.
ED25519 key fingerprint is SHA256:qtQKrWqPkeXr3mH/xuBEU5qh73DvzRSPAP4v74hscC0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.111.128' (ED25519) to the list of known hosts.
anara@192.168.111.128's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Nov 19 01:08:02 PM UTC 2025

  System load:  0.0               Processes:             230
  Usage of /:   88.4% of 9.75GB   Users logged in:       1
  Memory usage: 21%               IPv4 address for ens33: 192.168.111.128
  Swap usage:   0%

  => / is using 88.4% of 9.75GB

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

13 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Wed Nov 19 13:05:10 2025 from 192.168.111.1
```

Exam Evaluation Questions

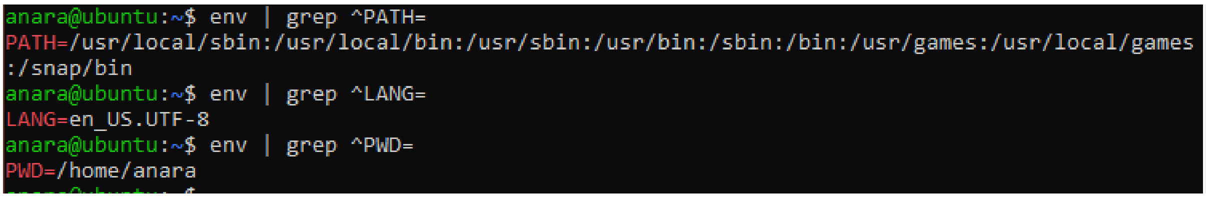
Q1: Quick Environment Audit

1. **Run a single command to display environment variables and capture its output**



```
anara@ubuntu:~$ env | grep PATH
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
:/snap/bin
```

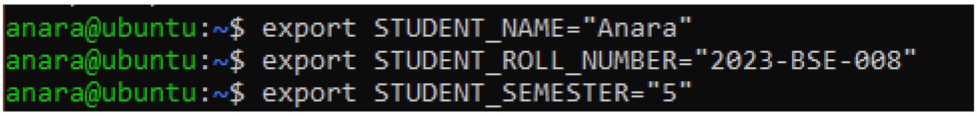
2. **In the same terminal session, run three filters (one per line) to show values for PATH, LANG, and PWD, then capture a single screenshot showing the three outputs together.**



```
anara@ubuntu:~$ env | grep ^PATH=
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
:/snap/bin
anara@ubuntu:~$ env | grep ^LANG=
LANG=en_US.UTF-8
anara@ubuntu:~$ env | grep ^PWD=
PWD=/home/anara
```

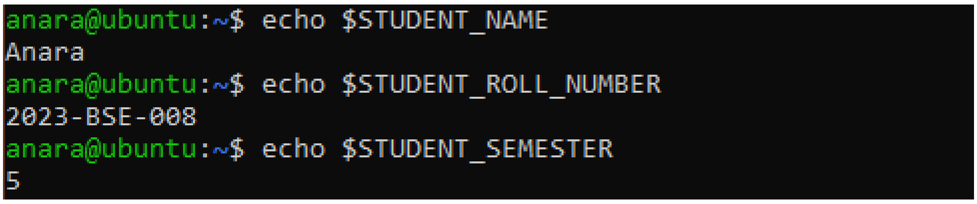
Q2: Short-lived Student Info

1. **In one terminal, set three variables (STUDENT_NAME, STUDENT_ROLL_NUMBER, STUDENT_SEMESTER) using export — execute all three consecutively and capture them in one screenshot (show the commands executed).**



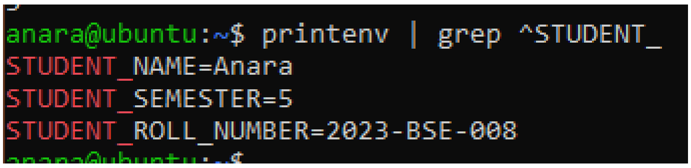
```
anara@ubuntu:~$ export STUDENT_NAME="Anara"
anara@ubuntu:~$ export STUDENT_ROLL_NUMBER="2023-BSE-008"
anara@ubuntu:~$ export STUDENT_SEMESTER="5"
```

2. **Still in the same session, print the three values with echo (grouped) and capture the outputs in one screenshot**



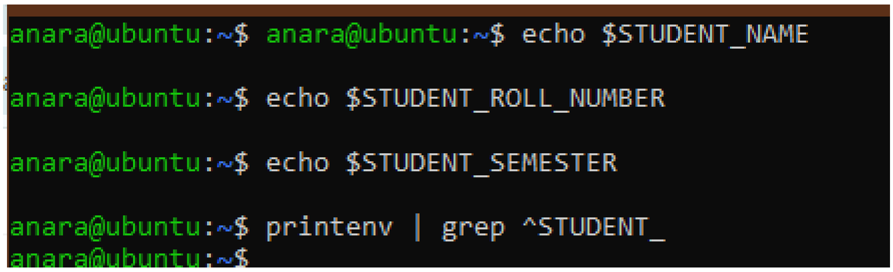
```
anara@ubuntu:~$ echo $STUDENT_NAME
Anara
anara@ubuntu:~$ echo $STUDENT_ROLL_NUMBER
2023-BSE-008
anara@ubuntu:~$ echo $STUDENT_SEMESTER
5
```

3. **Use a single printenv|grep command to list any STUDENT_ variables and capture the result**



```
anara@ubuntu:~$ printenv | grep ^STUDENT_
STUDENT_NAME=Anara
STUDENT_SEMESTER=5
STUDENT_ROLL_NUMBER=2023-BSE-008
```

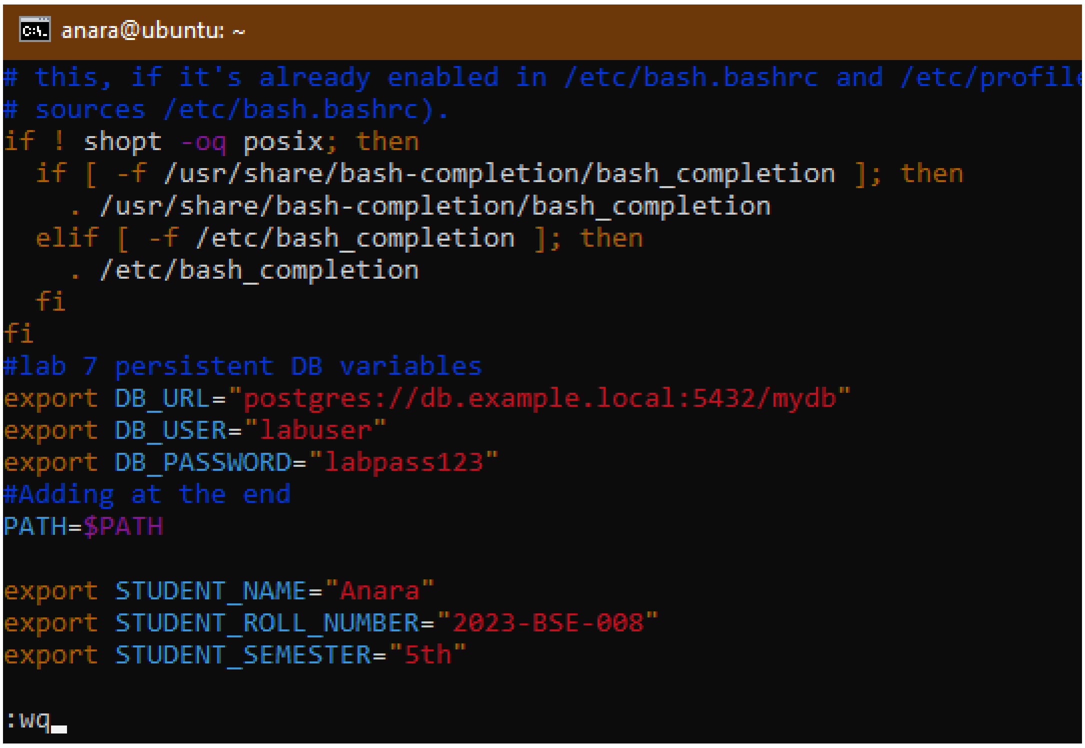
4. **Exit that shell, open a fresh terminal, and show that the STUDENT_ variables are not set (use echo and printenv|grep together) — capture in one screenshot**



```
anara@ubuntu:~$ anara@ubuntu:~$ echo $STUDENT_NAME

anara@ubuntu:~$ echo $STUDENT_ROLL_NUMBER

anara@ubuntu:~$ echo $STUDENT_SEMESTER

anara@ubuntu:~$ printenv | grep ^STUDENT_
anara@ubuntu:~$
```

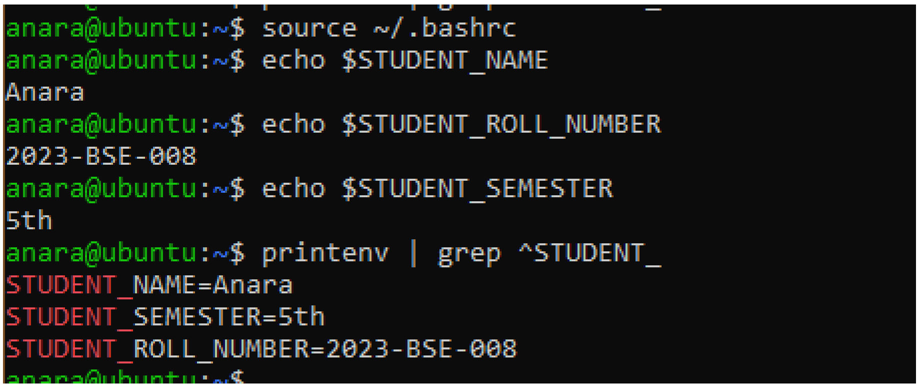
Q3: Make It Sticky (Persistence Check for Student Info)

1. **Edit ~/.bashrc and append the three STUDENT_* exports. Capture a screenshot of the editor showing the new lines**



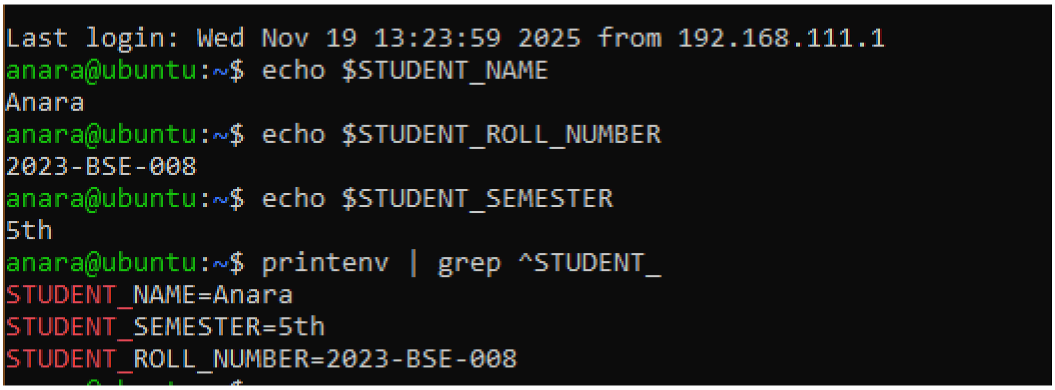
```
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
#lab 7 persistent DB variables
export DB_URL="postgres://db.example.local:5432/mydb"
export DB_USER="labuser"
export DB_PASSWORD="labpass123"
#Adding at the end
PATH=$PATH

export STUDENT_NAME="Anara"
export STUDENT_ROLL_NUMBER="2023-BSE-008"
export STUDENT_SEMESTER="5th"

:wq
```

2. **Reload your shell config with a single command and then verify the three variables and show printenv | grep '^STUDENT_' — capture these verification outputs together in one screenshot.**



```
anara@ubuntu:~$ source ~/.bashrc
anara@ubuntu:~$ echo $STUDENT_NAME
Anara
anara@ubuntu:~$ echo $STUDENT_ROLL_NUMBER
2023-BSE-008
anara@ubuntu:~$ echo $STUDENT_SEMESTER
5th
anara@ubuntu:~$ printenv | grep ^STUDENT_
STUDENT_NAME=Anara
STUDENT_SEMESTER=5th
STUDENT_ROLL_NUMBER=2023-BSE-008
anara@ubuntu:~$
```

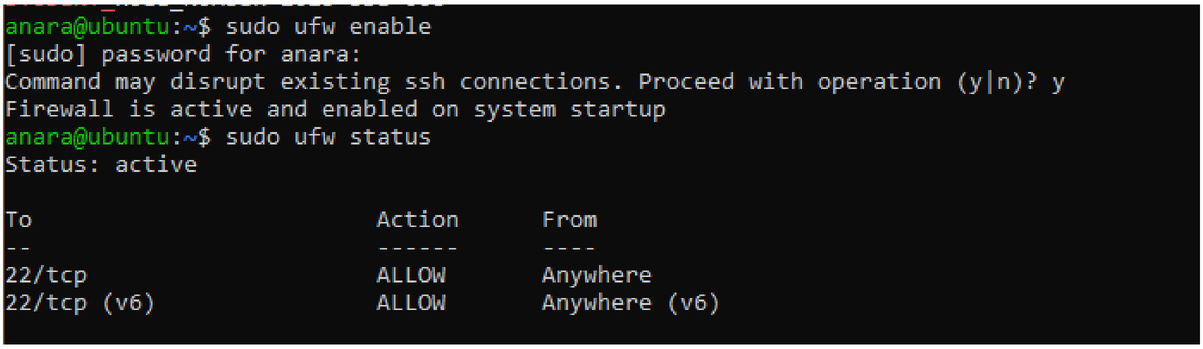
3. **Close and re-open a terminal and demonstrate the STUDENT_NAME variable is available (echo and printenv grep together) — capture in one screenshot.**



```
Last login: Wed Nov 19 13:23:59 2025 from 192.168.111.1
anara@ubuntu:~$ echo $STUDENT_NAME
Anara
anara@ubuntu:~$ echo $STUDENT_ROLL_NUMBER
2023-BSE-008
anara@ubuntu:~$ echo $STUDENT_SEMESTER
5th
anara@ubuntu:~$ printenv | grep ^STUDENT_
STUDENT_NAME=Anara
STUDENT_SEMESTER=5th
STUDENT_ROLL_NUMBER=2023-BSE-008
```

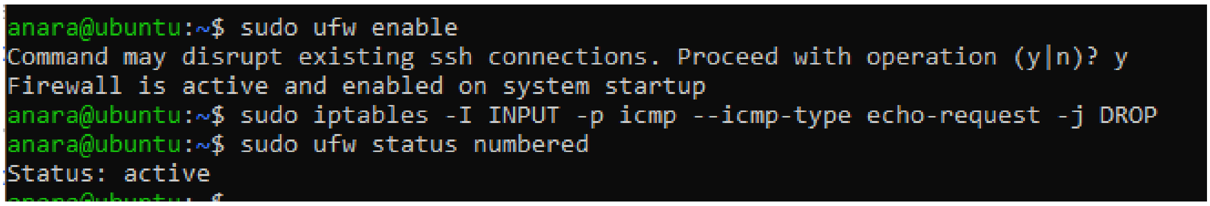
Q4: Firewall Rules: Block and Restore Ping (ICMP)

1. **Enable ufw and capture the enable command and status together in one screenshot.**



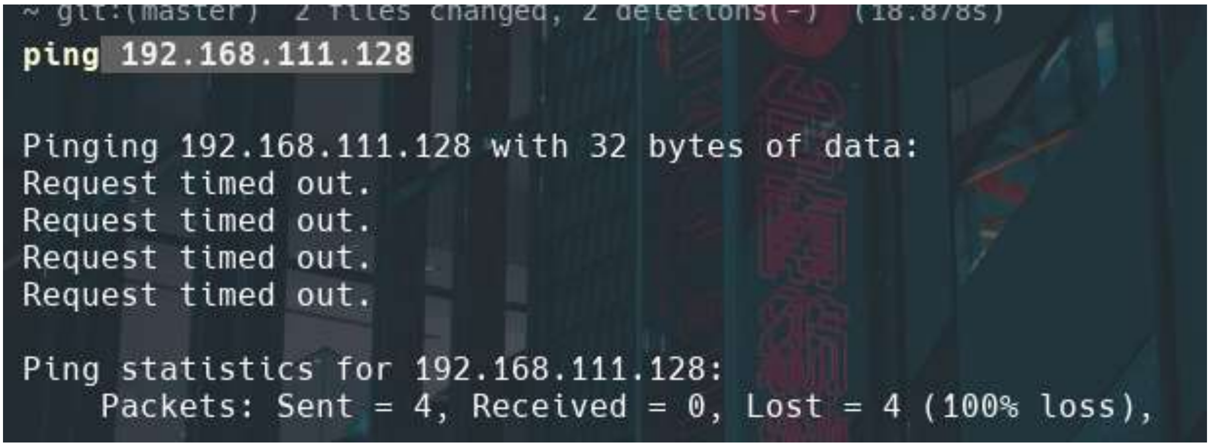
```
anara@ubuntu:~$ sudo ufw enable
[sudo] password for anara:
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
anara@ubuntu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
```

2. **Add a rule to block ping (ICMP echo) and show ufw status numbered in the same screenshot.**



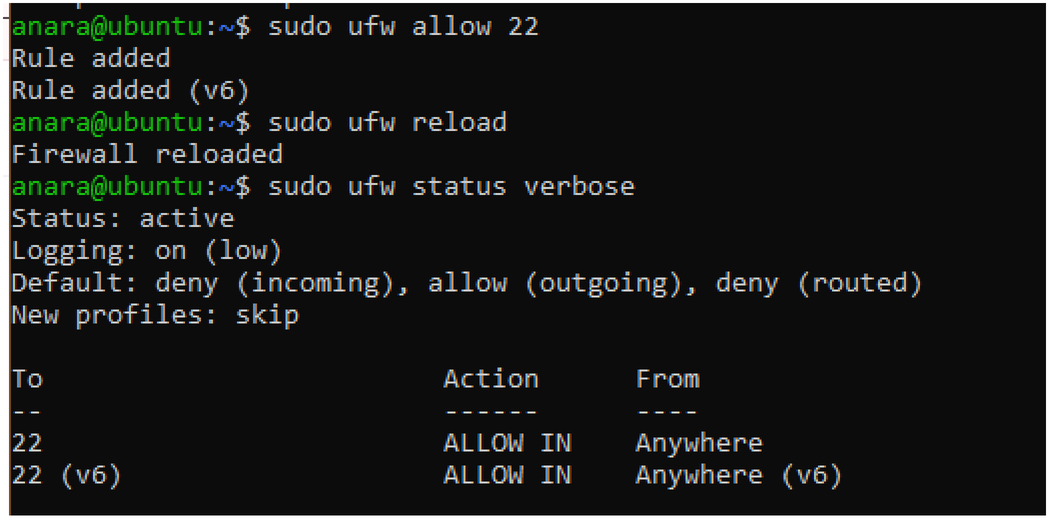
```
anara@ubuntu:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
anara@ubuntu:~$ sudo iptables -I INPUT -p icmp --icmp-type echo-request -j DROP
anara@ubuntu:~$ sudo ufw status numbered
Status: active
```

3. **From your Windows host (or another client), attempt to ping the server while the rule is active and capture the blocked/failing ping in one screenshot.**



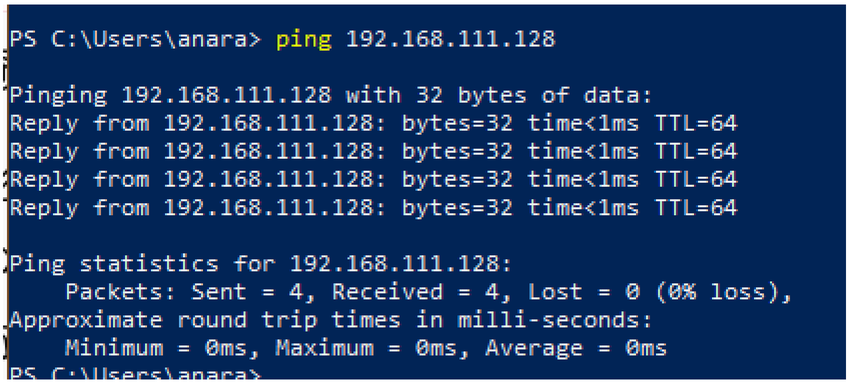
```
~ git:(master)  2 files changed, 2 deletions(-)  (18.878s)
ping 192.168.111.128

Pinging 192.168.111.128 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.111.128:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

4. **Re-allow ping (ICMP) (or remove the deny rule) and capture the allow/reload/status sequence in one screenshot**



```
anara@ubuntu:~$ sudo ufw allow 22
Rule added
Rule added (v6)
anara@ubuntu:~$ sudo ufw reload
Firewall reloaded
anara@ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    Anywhere
22 (v6)                    ALLOW IN    Anywhere (v6)
```

5. **From the client, ping the server again and capture successful replies in one screenshot**



```
PS C:\Users\anara> ping 192.168.111.128

Pinging 192.168.111.128 with 32 bytes of data:
Reply from 192.168.111.128: bytes=32 time<1ms TTL=64
Reply from 192.168.111.128: bytes=32 time<1ms TTL=64
Reply from 192.168.111.128: bytes=32 time<1ms TTL=64
Reply from 192.168.111.128: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.111.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\anara>
```

****