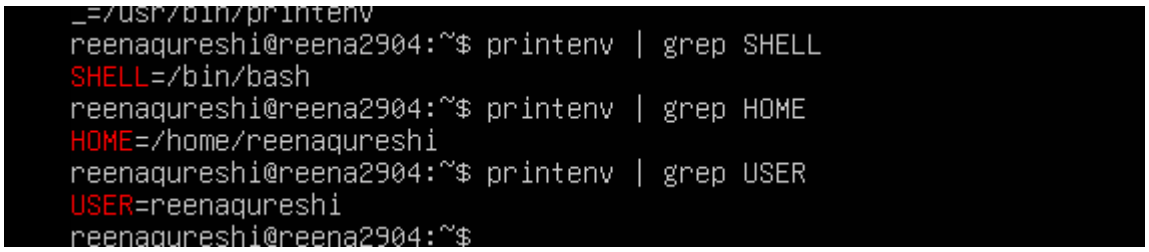
Lab 7

Task 1 — Print & filter environment variables

Print all environment variables:

printenv

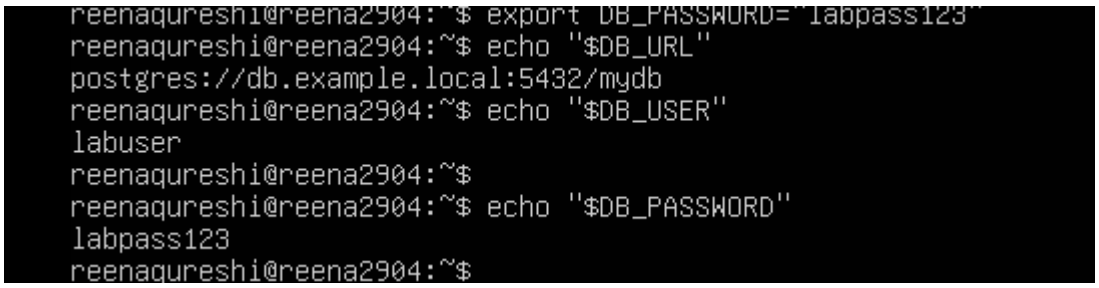
Filter for SHELL, HOME and USER — run these greps together and capture one combined screenshot:



Task 2 — Export DB_* variables temporarily and observe scope

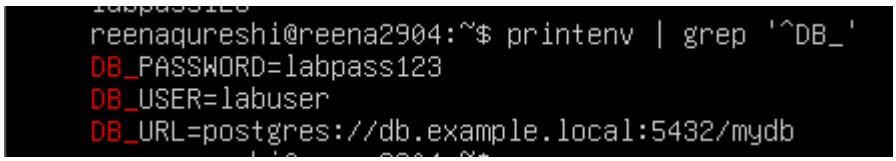
Define all DB_* variables (run the three exports one after another). Capture them in one screenshot showing the three export commands and their execution:

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



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

printenv | grep '^DB_'



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:

echo "$DB_URL"

printenv | grep '^DB_'

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

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

vim ~/.bashrc

add at the end:

Lab 7 persistent DB variables

export DB_URL="postgres://db.example.local:5432/mydb"

export DB_USER="labuser"

export DB_PASSWORD="labpass123"

```
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
export DB_URL="postgres://db.example.local:5432/mydb"

export DB_USER="labuser"

export DB_PASSWORD="labpass123"
".bashrc" 121L, 3883B written
reenaqureshi@reena2904:~$ _
```

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:

source ~/.bashrc

echo "$DB_URL"

echo "$DB_USER"

echo "$DB_PASSWORD"

printenv | grep '^DB_'

```
    reenaqureshi@reena2904:~$ source ~/.bashrc
    reenaqureshi@reena2904:~$ echo "$DB_URL"
    postgres://db.example.local:5432/mydb
    reenaqureshi@reena2904:~$ echo "$DB_USER"
    labuser
    reenaqureshi@reena2904:~$ echo "$DB_PASSWORD"
    labpass123
    reenaqureshi@reena2904:~$ printenv | grep '^DB_'
    DB_PASSWORD=labpass123
    DB_USER=labuser
    DB_URL=postgres://db.example.local:5432/mydb
```

Close and reopen terminal. Verify persistence by running one echo and the grep together

echo "$DB_URL"

printenv | grep '^DB_'

```
reenaqureshi@reena2904:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
reenaqureshi@reena2904:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
reenaqureshi@reena2904:~$
```

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

View /etc/environment:

sudo cat /etc/environment

```
reenaqureshi@reena2904:~$ sudo cat /etc/environment
[sudo] password for reenaqureshi:
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
reenaqureshi@reena2904:~$
```

Show current PATH:

echo "$PATH"

```
reenaqureshi@reena2904:~$ echo "$PATH"
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

Edit /etc/environment and add Class:

sudo vim /etc/environment

add line: Class="CC-<your_class_name>"

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
class= ReenaQureshi-0522_
```

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:

```
reenaqureshi@reena2904:~$ echo "$Class"
CC-5B-REENAQURESHI_052
reenaqureshi@reena2904:~$ echo "$PATH"
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
reenaqureshi@reena2904:~$
```

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

cat > ~/welcome <<'EOF'

#!/bin/bash

echo "Welcome to Cloud Computing $USER"

EOF

chmod +x ~/welcome

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lo
reenaqureshi@reena2904:~$ cat > ~/welcome <<'EOF'
> #!/bin/bash
>
> echo "Welcome to Cloud Computing $USER"
>
> EOF
reenaqureshi@reena2904:~$
reenaqureshi@reena2904:~$ cat ~/welcome
#!/bin/bash

echo "Welcome to Cloud Computing $USER"

reenaqureshi@reena2904:~$ chmod +x ~/welcome
```

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

```
reenaqureshi@reena2904:~$ cd ~
reenaqureshi@reena2904:~$ ./~welcome
-bash: ./~welcome: No such file or directory
reenaqureshi@reena2904:~$ ./welcome
Welcome to Cloud Computing reenaqureshi
reenaqureshi@reena2904:~$
```

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:

vim ~/.bashrc

add at end:

PATH=$PATH:~

```
export DB_USER="labuser"

export DB_PASSWORD="labpass123"
PATH=$PATH:~
".bashrc" 122L, 3896B written
reenaqureshi@reena2904:~$
```

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

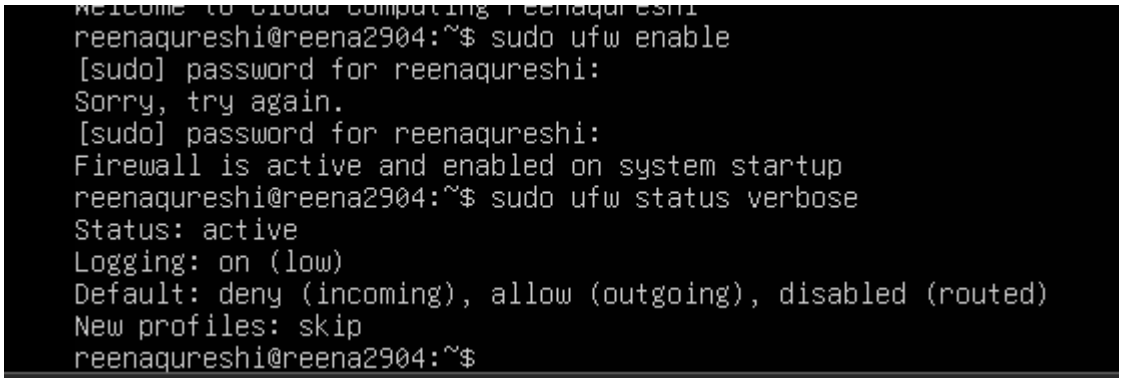
```
reenaqureshi@reena2904:~$ source ~/.bashrc
reenaqureshi@reena2904:~$ cd ~
reenaqureshi@reena2904:~$ ./welcome
Welcome to Cloud Computing reenaqureshi
reenaqureshi@reena2904:~$ _
```

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

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

sudo ufw enable

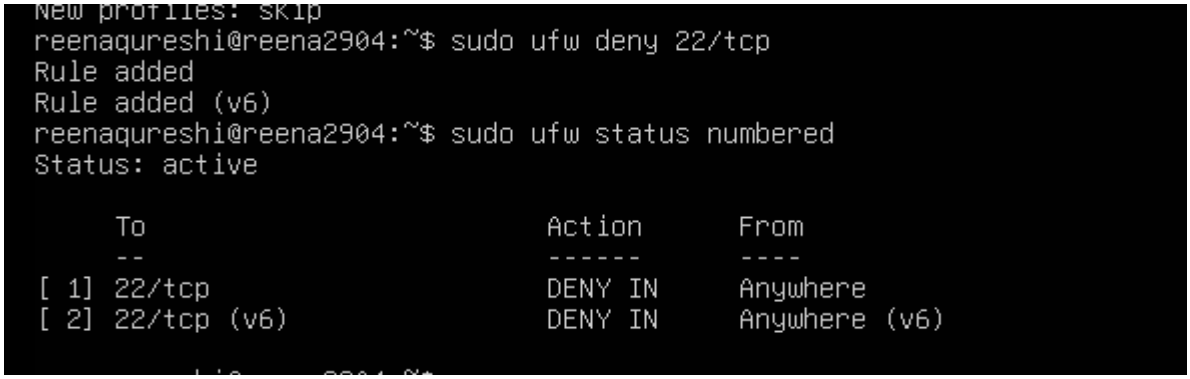
sudo ufw status verbose



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

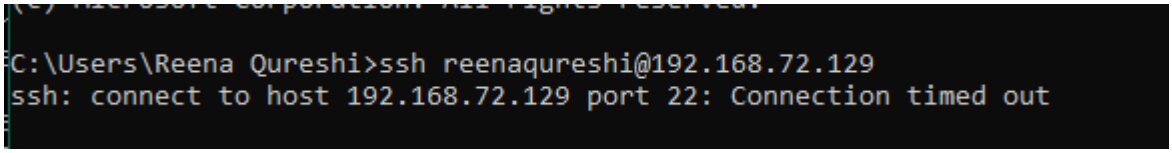
sudo ufw deny 22/tcp

sudo ufw status numbered



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

ssh username@<server_ip>



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

sudo ufw allow 22/tcp

sudo ufw reload

sudo ufw status

```
reenaqureshi@reena2904: $ sudo ufw allow 22/tcp
Rule updated
Rule updated (v6)
reenaqureshi@reena2904:~$ sudo ufw allow 22/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
reenaqureshi@reena2904:~$ sudo ufw reload
Firewall reloaded
reenaqureshi@reena2904:~$ sudo ufw status
Status: active

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

reenaqureshi@reena2904:~$
```

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

ssh username@<server_ip>

```
  Memory usage: 17%                IPV4 address for ens33: 192.168.76.
  Swap usage:   0%

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

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

Expanded Security Maintenance for Applications is not enabled.

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

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts.

Last login: Sat Oct 25 11:11:07 2025 from 192.168.76.1
reenaqureshi@reena2904:~$
```

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

A. On Windows host (client) — group related client actions:

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

ssh-keygen -t ed25519 -f ~/.ssh/id_lab7 -C "lab_key"

ls -la ~/.ssh

```
reenaqureshi@reena2904:~$ 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/reenaqureshi/.ssh/id_lab7
Your public key has been saved in /home/reenaqureshi/.ssh/id_lab7.pub
The key fingerprint is:
SHA256:IFxImhMWlPFa9L/uXBzPB8twh1Dz7GhfhZp3vPbUsVY lab_key
The key's randomart image is:
+--[ED25519 256]--+
| .*=o..     o    |
| ..B.o     . + . |
|  + = o   .   + .|
|   + . o   . * ..|
|  .     S o O +.E|
|         o O * o*|
|        . o = o=o|
|       o .   .o..|
|       .+       .|
+----[SHA256]-----+
reenaqureshi@reena2904:~$ ls -la ~/.ssh
total 20
drwx------  2 reenaqureshi reenaqureshi 4096 Nov  7 05:05 .
drwxr-x--- 24 reenaqureshi reenaqureshi 4096 Nov  7 04:39 ..
-rw-------  1 reenaqureshi reenaqureshi    0 Sep 26 10:18 authorized_keys
-rw-------  1 reenaqureshi reenaqureshi  399 Nov  7 05:05 id_lab7
-rw-r--r--  1 reenaqureshi reenaqureshi   89 Nov  7 05:05 id_lab7.pub
-rw-r--r--  1 reenaqureshi reenaqureshi  142 Oct 22 16:16 known_hosts
reenaqureshi@reena2904:~$
```

Show the public key content (single screenshot):

type $env:USERPROFILE\.ssh\id_lab7.pub

or on Git Bash: cat ~/.ssh/id_lab7.pub

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

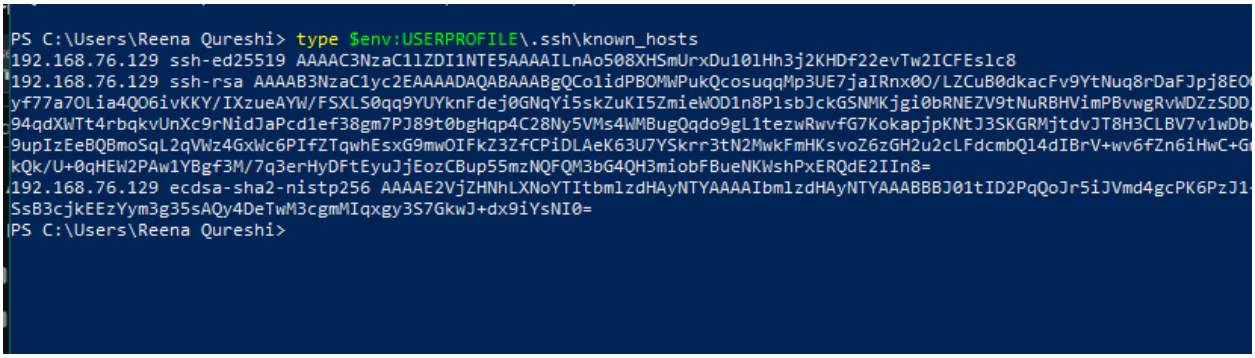
Clear contents (PowerShell)

Clear-Content $env:USERPROFILE\.ssh\known_hosts

View the file (should be empty)

type $env:USERPROFILE\.ssh\known_hosts

```
logout
Connection to 192.168.76.129 closed.
PS C:\Users\Reena Qureshi> Clear-Content $env:USERPROFILE\.ssh\known_hosts
PS C:\Users\Reena Qureshi> type $env:USERPROFILE\.ssh\known_hosts
PS C:\Users\Reena Qureshi>
```



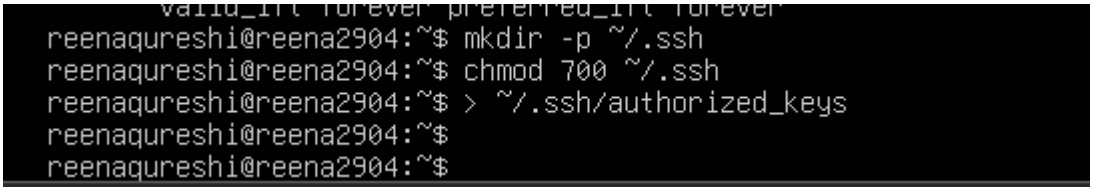
B. On Ubuntu server — group related server-side commands:

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:

mkdir -p ~/.ssh

chmod 700 ~/.ssh

> ~/.ssh/authorized_keys



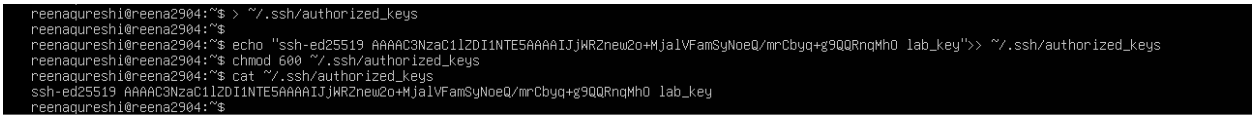
Append the public key, set file permissions, and show the resulting authorized_keys (capture commands and resulting file content in one screenshot):

paste public key name id_lab7.pub from Windows client into the echo below

echo "ssh-ed25519 AAAA... yourpublickey ... comment" >> ~/.ssh/authorized_keys

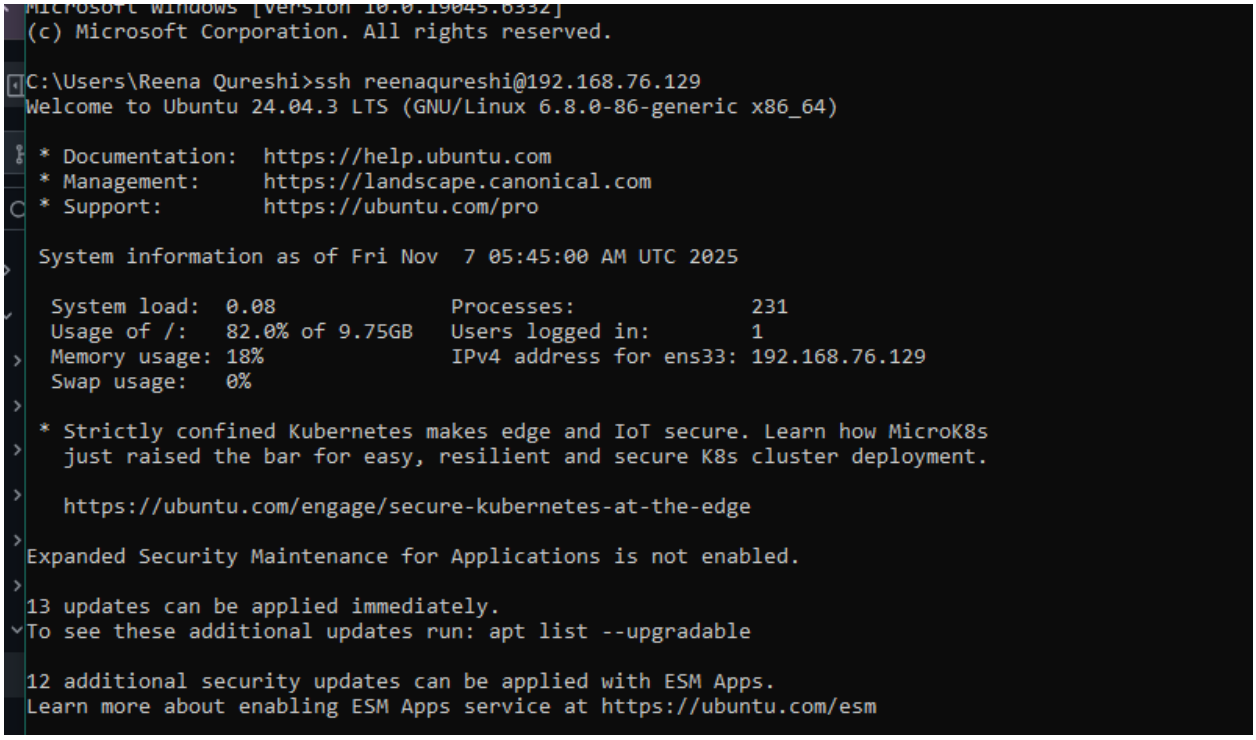
chmod 600 ~/.ssh/authorized_keys

cat ~/.ssh/authorized_keys



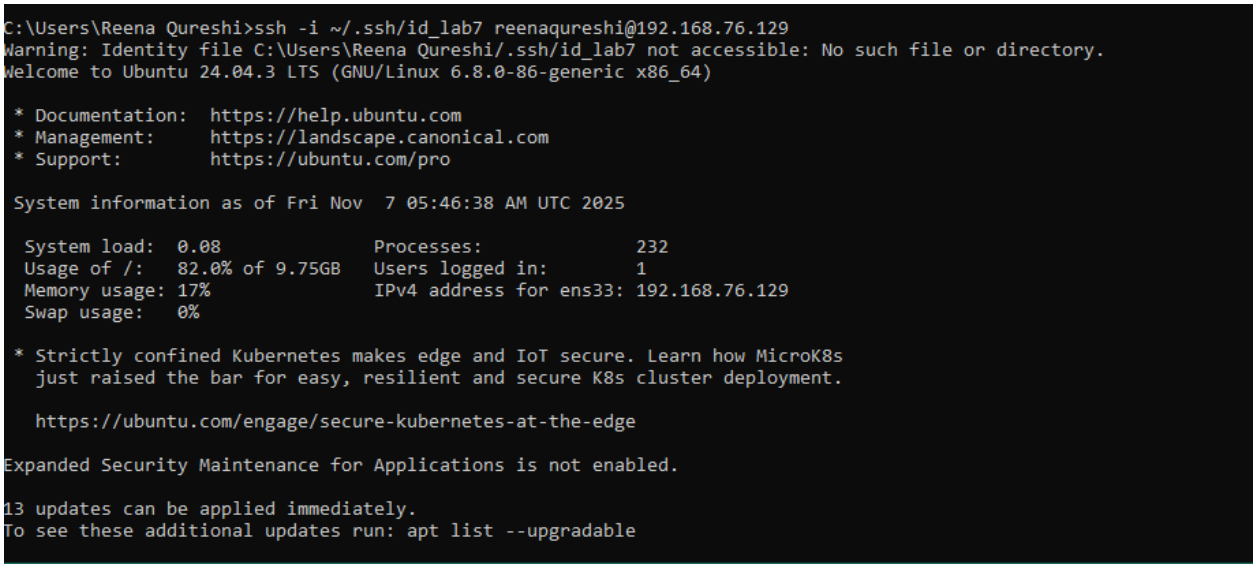
From Windows host test passwordless login (capture successful login in one screenshot):

ssh username@<server_ip>



Also demonstrate explicit identity usage (single screenshot):

ssh -i ~/.ssh/id_lab7 username@<server_ip>



Exam Evaluation Questions:

Q1: Quick Environment Audit

```
reenaqureshi@reena2904:~$ printenv
SHELL=/bin/bash
CREDENTIALS_DIRECTORY=/run/credentials/getty@tty1.service
MEMORY_PRESSURE_WRITE=c29tZSAyMDAwMDAgMjAwMDAuMAA=
XDG_SEAT=seat0
PWD=/home/reenaqureshi
LOGNAME=reenaqureshi
XDG_SESSION_TYPE=tty
SYSTEMD_EXEC_PID=1337
HOME=/home/reenaqureshi
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:*.tbz2=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;35:*.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:
MEMORY_PRESSURE_WATCH=/sys/fs/cgroup/system.slice/system-getty.slice/getty@tty1.service/memory.pressure
INVOCATION_ID=1ebdf7ec07bc4e908d88b09f38bfeccc
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=linux
LESSOPEN=| /usr/bin/lesspipe %s
USER=reenaqureshi
```

```
_=/usr/bin/printenv
reenaqureshi@reena2904:~$ printenv | grep SHELL
SHELL=/bin/bash
reenaqureshi@reena2904:~$ printenv | grep HOME
HOME=/home/reenaqureshi
reenaqureshi@reena2904:~$ printenv | grep USER
USER=reenaqureshi
reenaqureshi@reena2904:~$
```

Q2: Short-lived Student Info

```
EXPORT: command not found
reenaqureshi@reena2904:~$ EXPORT db_url="POSTGRES://DB.EXAMPLE.LOCAL:5432/MYDB"
EXPORT: command not found
reenaqureshi@reena2904:~$ export DB_URL="postgres://db.example.local:5432/mydb"
reenaqureshi@reena2904:~$ export DB_USER="labuser"
reenaqureshi@reena2904:~$ export DB_PASSWORD="labpass123"
reenaqureshi@reena2904:~$
```

```
reenaqureshi@reena2904:~$ export DB_PASSWORD="labpass123"
reenaqureshi@reena2904:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
reenaqureshi@reena2904:~$ echo "$DB_USER"
labuser
reenaqureshi@reena2904:~$
reenaqureshi@reena2904:~$ echo "$DB_PASSWORD"
labpass123
reenaqureshi@reena2904:~$
```

```
reenaqureshi@reena2904:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

```
Failed to connect to https://changelogs.ubuntu.com/met


reenaqureshi@reena2904:~$ echo "$DB_URL"

reenaqureshi@reena2904:~$ printenv | grep '^DB_'
reenaqureshi@reena2904:~$
```

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

```
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
export DB_URL="postgres://db.example.local:5432/mydb"

export DB_USER="labuser"

export DB_PASSWORD="labpass123"
".bashrc" 121L, 3883B written
reenaqureshi@reena2904:~$
```

```
reenaqureshi@reena2904:~$ source ~/.bashrc
reenaqureshi@reena2904:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
reenaqureshi@reena2904:~$ echo "$DB_USER"
labuser
reenaqureshi@reena2904:~$ echo "$DB_PASSWORD"
labpass123
reenaqureshi@reena2904:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

```
reenaqureshi@reena2904:~$ echo "$DB_URL"
postgres://db.example.local:5432/mydb
reenaqureshi@reena2904:~$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
reenaqureshi@reena2904:~$
```

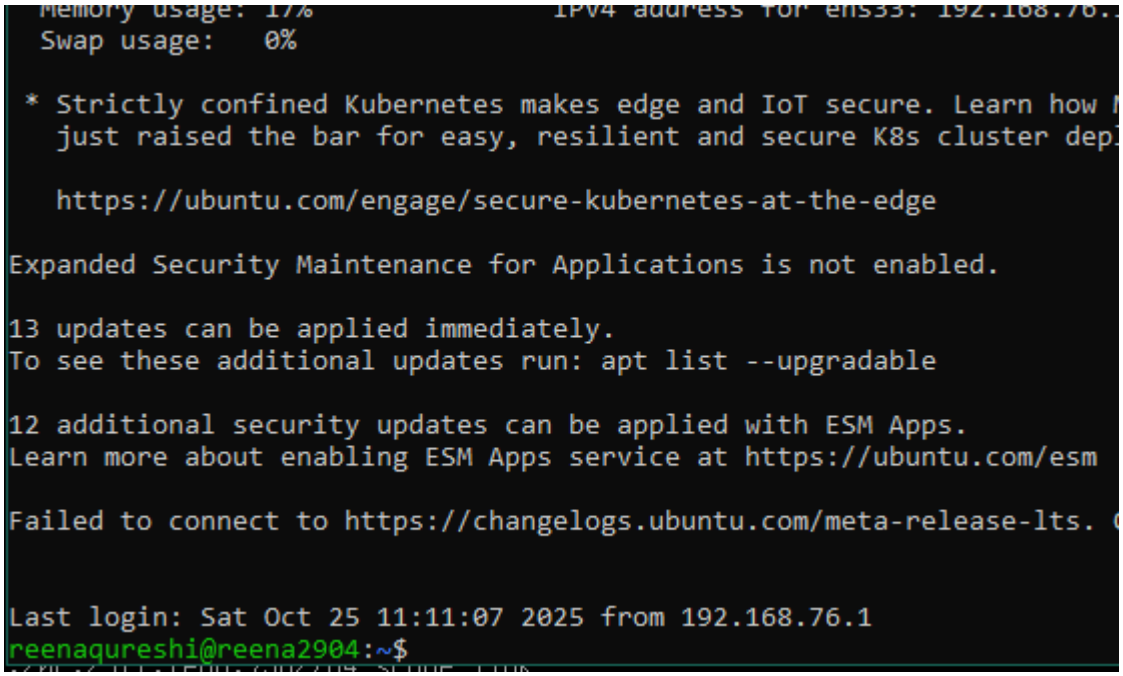
Q4: Firewall Rules – Block and Restore Ping (ICMP)

```
reenaqureshi@reena2904:~$ sudo ufw enable
[sudo] password for reenaqureshi:
Sorry, try again.
[sudo] password for reenaqureshi:
Firewall is active and enabled on system startup
reenaqureshi@reena2904:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
reenaqureshi@reena2904:~$
```

```
New profiles: skip
reenaqureshi@reena2904:~$ sudo ufw deny 22/tcp
Rule added
Rule added (v6)
reenaqureshi@reena2904:~$ sudo ufw status numbered
Status: active

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

```
C:\Users\Reena Qureshi>ssh reenaqureshi@192.168.72.129
ssh: connect to host 192.168.72.129 port 22: Connection timed out
```

```
Rule updated
Rule updated (v6)
reenaqureshi@reena2904:~$ sudo ufw allow 22/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
reenaqureshi@reena2904:~$ sudo ufw reload
Firewall reloaded
reenaqureshi@reena2904:~$ sudo ufw status
Status: active

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

reenaqureshi@reena2904:~$
```


Memory usage: 17%
IPv4 address for ens33: 192.168.76.
Swap usage: 0%
* Strictly confined Kubernetes makes edge and IoT secure. Learn how
just raised the bar for easy, resilient and secure K8s cluster dep
https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Expanded Security Maintenance for Applications is not enabled.
13 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
12 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts.
Last login: Sat Oct 25 11:11:07 2025 from 192.168.76.1
reenaqureshi@reena2904:~$