



Cloud Computing Lab

Submitted To:

Sir Shoaib & Sir Waqas

Submitted By:

Hamail Fatima

Roll No:

2023-BSE-023

Section:

5(A)

Cloud Computing Lab 7

Lab Tasks

Task 1 – Print & filter environment variables

Goal: Show environment variables and filter them using grep.
Commands and required screenshots (grouped as requested):

1. Print all environment variables:

printenv

- Save screenshot as: task1_printenv_all.png

```
admin@Hamail-ALAM MINGW64 ~
$ printenv
ProgramFiles(x86)=C:\Program Files (x86)
!::=:
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
EFC_16580_1592913036=
SHELL=/usr/bin/bash
NUMBER_OF_PROCESSORS=8
PROCESSOR_LEVEL=6
TERM_PROGRAM_VERSION=3.8.1
MINGW_PREFIX=/mingw64
PKG_CONFIG_PATH=/mingw64/lib/pkgconfig:/mingw64/share/pkgconfig
USERDOMAIN_ROAMINGPROFILE=HAMAIL-ALAM
HOSTNAME=Hamail-ALAM
PROGRAMFILES=C:\Program Files
MSYSTEM=MINGW64
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
ORIGINAL_TEMP=/tmp
MINGW_CHOST=x86_64-w64-mingw32
OS=windows_NT
HOMEDRIVE=C:
HOMEPATH=/
HOME=C:\Users\admin
MSYSTEM_CARCH=x86_64
USERDOMAIN=HAMAIL-ALAM
PWD=/c/Users/admin
USERPROFILE=C:\Users\admin
OneDriveConsumer=C:\Users\admin\OneDrive
MANPATH=/mingw64/local/man:/mingw64/share/man:/usr/local/man:/usr/share/man:/usr
/man:/share/man
MINGW_PACKAGE_PREFIX=mingw-w64-x86_64
ALLUSERSPROFILE=C:\ProgramData
ORIGINAL_PATH=/mingw64/bin:/usr/bin:/c/Users/admin/bin:/c/Program Files (x86)/VM
ware/VMware Player/bin:/c/Windows/system32:/c/Windows/c/Windows/System32/wbem:/c/Windows/System32/WindowsPowerShell/v1.0:/c/Windows/System32/openSSH:/c/MingW/b
in:/cmd:/c/Users/admin/bin:/c/Users/admin/AppData/Local/Programs/Python/Python313/Scripts:/c/Users/admin/AppData/Local/Programs/Python/Python313:/c/Users
/admin/AppData/Local/Microsoft/WindowsApps:/c/Users/admin/AppData/Local/Programs
/Microsoft VS Code/bin
CommonProgramW6432=C:\Program Files\Common Files
HOME=/c/Users/admin
USERNAME=admin
SSH_ASKPASS=/mingw64/bin/git-askpass.exe
PLINK_PROTOCOL=ssh
OneDrive=C:\Users\admin\OneDrive
COMSPEC=C:\Windows\system32\cmd.exe
TMPDIR=/tmp
APPDATA=C:\Users\admin\AppData\Roaming
SYSTEMROOT=C:\Windows
LOCALAPPDATA=C:\Users\admin\AppData\Local
```

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

printenv | grep SHELL

printenv | grep HOME

printenv | grep USER

- Save screenshot as: task1_grep_shell_home_user.png (single screenshot showing all three grep outputs together)

```
admin@Hamail-ALam MINGW64 ~
$ printenv | grep SHELL
printenv | grep HOME
printenv | grep USER
SHELL=/usr/bin/bash
HOMEDRIVE=C:
HOME=/c/Users/admin
HOMEPATH=\Users\admin
USERDOMAIN_ROAMINGPROFILE=HAMAIL-ALAM
USERDOMAIN=HAMAIL-ALAM
USERPROFILE=C:\Users\admin
ALLUSERSPROFILE=C:\ProgramData
USERNAME=admin
```

⌚ Screenshots:

- task1_printenv_all.png
- task1_grep_shell_home_user.png

Task 2 – Export DB_* variables temporarily and observe scope

Goal: Create env variables with export in the current shell, verify them, then close shell and show variables are gone.

Per the requested grouping rule: capture all the variable-definition (export) commands in a single screenshot; capture the echo/print checks grouped logically.

Steps and required screenshots:

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:

```
export DB_URL="postgres://db.example.local:5432/mydb"
export DB_USER="labuser"
export DB_PASSWORD="labpass123"
```

- Save screenshot as: task2_exports_all.png (single screenshot showing all three export commands shown/executed)

```
admin@Hamail-ALam MINGW64 ~
$ export DB_URL="postgres://db.example.local:5432/mydb"
export DB_USER="labuser"
export DB_PASSWORD="labpass123"
```

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

```
echo "$DB_URL"
echo "$DB_USER"
echo "$DB_PASSWORD"
• Save screenshot as: task2_echoes_all.png
```

```
admin@Hamail-ALam MINGW64 ~
$ echo "$DB_URL"
echo "$DB_USER"
echo "$DB_PASSWORD"
postgres://db.example.local:5432/mydb
labuser
labpass123
```

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

```
printenv | grep '^DB_'
• Save screenshot as: task2_printenv_grep_db.png
```

```
admin@Hamail-Alam MINGW64 ~
$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

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:

```
echo "$DB_URL"
printenv | grep '^DB_'
• Save screenshot as: task2_after_restart_checks.png (single screenshot showing echo (empty) and printenv | grep '^DB_' with no results)
```

```
admin@Hamail-Alam MINGW64 ~
$ echo "$DB_URL"
printenv | grep '^DB_'
postgres://db.example.local:5432/mydb
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

Optional combined evidence (if you want to also save a combined screenshot showing steps 1-3 together for convenience):

```
admin@Hamail-Alam MINGW64 ~
$ printenv | grep '^DB_'
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

- task2_export_echo_printenv_combined.png
- 📸 Screenshots (required):
task2_exports_all.png
task2_echoes_all.png
task2_printenv_grep_db.png
task2_after_restart_checks.png
(Optional) task2_export_echo_printenv_combined.png

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

Goal: Add DB_* variables to ~/.bashrc, reload, and verify persistence. Grouped captures: show the three export lines in ~/.bashrc together, and group the post-source checks into one screenshot.

Steps and required screenshots:

1. 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"
```

- Save screenshot as: task3_bashrc_added.png (single screenshot showing the three export lines in the editor)

```
source ~/bashrc
```

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:

```
source ~/bashrc
```

```
echo "$DB_URL"
```

```
echo "$DB_USER"
```

```
echo "$DB_PASSWORD"
```

```
printenv | grep '^DB_'
```

- Save screenshot as: task3_source_and_verification.png (single screenshot showing source, the three echoes, and the grep output)

```
admin@Hamail-Altam MINGW64 ~
$ source ~/bashrc
echo "$DB_URL"
echo "$DB_USER"
echo "$DB_PASSWORD"
printenv | grep '^DB_'
postgres://db.example.local:5432/mydb
labuser
labpass123
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```

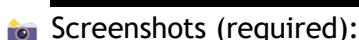
3. Close and reopen terminal. Verify persistence by running one echo and the grep together – capture both in one screenshot:

```
echo "$DB_URL"
```

```
printenv | grep '^DB_'
```

- Save screenshot as: task3_after_restart_persistent.png (single screenshot showing echo with value and grep output listing DB_ variables)

```
admin@Hamail-Altam MINGW64 ~
$ echo "$DB_URL"
printenv | grep '^DB_'
postgres://db.example.local:5432/mydb
DB_PASSWORD=labpass123
DB_USER=labuser
DB_URL=postgres://db.example.local:5432/mydb
```



task3_bashrc_added.png

task3_source_and_verification.png

task3_after_restart_persistent.png

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

Goal: Add Class variable to /etc/environment, view PATH, create a welcome script at ~/welcome, make it executable, and add PATH entry in ~/.bashrc so welcome can be executed without ./. Capture grouped screenshots as applicable.

Steps and required screenshots (grouping applies to "print with grep" type commands and grouped variable definitions – in this task there is a single system variable definition so a standard per-action capture is used):

- ## 1. View /etc/environment:

```
sudo cat /etc/environment
```

- Save screenshot as: task4_etc_environment_before.png

```
hamail@ubuntu-lab:~$ sudo cat /etc/environment
[sudo] password for hamail:
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
hamail@ubuntu-lab:~$ _
```

- ## 2. Show current PATH:

```
echo "$PATH"
```

- Save screenshot as: task4_echo_path_before.png

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
hamall@ubuntu-lab:~$ echo "$PATH"
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
hamall@ubuntu-lab:~$
```

- ### 3. Edit /etc/environment and add Class:

```
sudo vim /etc/environment  
# add line: Class="CC-<your class name>"
```

- Save screenshot as: task4_etc_environment_edit_vim.png (editor with edit)

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"

- Save screenshot as: task4_etc_environment_after.png (cat or editor view showing the new Class line)

```
~  
~  
~  
~  
"/etc/environment" 3L, 1288 written  
hamail@ubuntu-lab:~$
```

- 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:
echo \$Class
echo "\$PATH"
 - Save screenshot as: task4_echo_class_and_path.png (single screenshot showing both

outputs)

```
~/etc/environment" 3L, 128B written  
hamail@ubuntu-lab:~$ echo $Class  
hamail@ubuntu-lab:~$ echo "$PATH"  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  
hamail@ubuntu-lab:~$
```

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):

```
cat > ~/.welcome <<'EOF'  
#!/bin/bash  
echo "Welcome to Cloud Computing $USER"  
EOF
```

```
chmod +x ~/.welcome
```

- Save screenshot as: task4_welcome_create_and_chmod.png (single screenshot showing heredoc creation command and chmod output/listing)

```
hamail@ubuntu-lab:~$ cat > ~/.welcome <<'EOF'  
#!/bin/bash  
echo "Welcome to Cloud Computing $USER"  
EOF  
  
chmod +x ~/.welcome  
hamail@ubuntu-lab:~$
```

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

```
cd ~  
.welcome  
• Save screenshot as: task4_welcome_run_dot.png
```

```
chmod +x ~/.welcome  
hamail@ubuntu-lab:~$ cd ~  
.welcome  
Welcome to Cloud Computing hamail  
hamail@ubuntu-lab:~$ |
```

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:

```
vim ~/.bashrc  
# add at end:  
PATH=$PATH:~
```

- Save screenshot as: task4_bashrc_path_line.png (editor screenshot showing the PATH line only)

```

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "##" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then

```

1,1

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

source ~/.bashrc

cd ~

welcome

- Save screenshot as: task4_bashrc_source_and_welcome.png (single screenshot showing the source command and the welcome output)

```

hamail@ubuntu-lab:~$ source ~/.bashrc
cd ~
welcome
Welcome to Cloud Computing hamail
hamail@ubuntu-lab:~$ |

```

📸 Screenshots (required):

task4_etc_environment_before.png
task4_echo_path_before.png
task4_etc_environment_edit_vim.png
task4_etc_environment_after.png
task4_echo_class_and_path.png
task4_welcome_create_and_chmod.png
task4_welcome_run_dot.png
task4_bashrc_path_line.png
task4_bashrc_source_and_welcome.png

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

Goal: Use ufw to deny and allow SSH then verify SSH connectivity changes from host. Save screenshots after each logical command/step; group related print checks when appropriate.

Steps and required screenshots:

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

```
sudo ufw enable  
sudo ufw status verbose
```

- Save screenshot as: task5_ufw_enable_and_status.png

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

2. 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
```

- Save screenshot as: task5_ufw_deny_22_and_status.png

```
hamail@ubuntu-lab:~$ sudo ufw deny 22/tcp  
sudo ufw status numbered  
Rule added  
Rule added (v6)  
Status: active  
  
 To           Action    From  
 --           -----   ----  
 [ 1] 22/tcp    DENY IN  Anywhere  
 [ 2] 22/tcp (v6) DENY IN  Anywhere (v6)  
  
hamail@ubuntu-lab:~$ |
```

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

```
ssh username@<server_ip>
```

- Save screenshot as: task5_ssh_attempt_blocked.png

```
hamail@ubuntu-lab:~$ ssh hamail@<server_ip>  
-bash: syntax error near unexpected token `newline'  
hamail@ubuntu-lab:~$ |
```

4. 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
```

- Save screenshot as: task5_ufw_allow_reload_status.png

```

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

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

hamail@ubuntu-lab:~$ |

```

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

ssh username@<server_ip>

- Save screenshot as: task5_ssh_success_after_allow.png

```

hamail@ubuntu-lab:~$ ssh hamail@192.168.114.129
hamail@192.168.114.129's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-86-generic x86_64)

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

System information as of Thu Nov 20 02:12:58 PM UTC 2025

System load: 0.03      Processes:            227
Usage of /: 85.7% of 9.75GB  Users logged in: 1
Memory usage: 16%      IPv4 address for ens33: 192.168.114.129
Swap usage: 0%

=> / is using 85.7% 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.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Nov 20 14:01:24 2025 from 192.168.114.1

```

 Screenshots (required):

- task5_ufw_enable_and_status.png
- task5_ufw_deny_22_and_status.png
- task5_ssh_attempt_blocked.png
- task5_ufw_allow_reload_status.png
- task5_ssh_success_after_allow.png

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

Goal: Copy your public key from the Windows host into the Ubuntu server's `~/.ssh/authorized_keys` to allow passwordless SSH. Save grouped screenshots for the client-side actions and the server-side edits/checks.

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

1. 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
```

- Save screenshot as: `task6_windows_sshkey_and_list.png` (single screenshot showing keygen result and `ls` of `.ssh` folder)

```
hamail@ubuntu-lab:~$ ssh-keygen -t ed25519 -f ~/.ssh/id_lab7 -C "lab_key"
ls -la ~/.ssh
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hamail/.ssh/id_lab7
Your public key has been saved in /home/hamail/.ssh/id_lab7.pub
The key fingerprint is:
SHA256:VYuHw1aXMTAi0u8Z7idmAkzZtcf/NssgAZ2e/gyE5lI lab_key
The key's randomart image is:
+--[ED25519 256]--+
|     . =+o |
| ..oo*+.+.. |
| o +B=o |
| o .oBo+ |
| o SE O . |
| o + * . . |
| o o + . . |
| o = * o.o |
| + o o +o |
+---[SHA256]---+
total 24
drwx----- 2 hamail hamail 4096 Nov 20 14:15 .
drwxr-x--- 25 hamail hamail 4096 Nov 20 14:05 ..
-rw----- 1 hamail hamail    0 Oct 29 13:04 authorized_keys
-rw----- 1 hamail hamail  399 Nov 20 14:15 id_lab7
-rw-r--r-- 1 hamail hamail   89 Nov 20 14:15 id_lab7.pub
-rw----- 1 hamail hamail  978 Oct 31 20:42 known_hosts
-rw-r--r-- 1 hamail hamail 142 Oct 31 20:42 known_hosts.old
hamail@ubuntu-lab:~$ |
```

2. Show the public key content (single screenshot):
type `$env:USERPROFILE\.ssh\id_lab7.pub`

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

- Save screenshot as: `task6_windows_public_key.png`

```

PS C:\Users\admin> ls $env:USERPROFILE\.ssh

Directory: C:\Users\admin\.ssh

Mode                LastWriteTime         Length Name
----                -----        ----
-a----       10/31/2025   6:36 PM          419 id_ed25519
-a----       10/31/2025   6:36 PM         108 id_ed25519.pub
-a----      11/21/2025   3:37 PM         399 id_lab7
-a----      11/21/2025   3:37 PM          90 id_lab7.pub
-a----     10/29/2025   6:19 PM        1677 known_hosts
-a----     10/29/2025   6:19 PM         926 known_hosts.old

PS C:\Users\admin> type $env:USERPROFILE\.ssh\id_lab7.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIaGzxx/M/3pPqrTa2HEZSK8ZYDutnOC+mt/wUzbVpI lab_key
PS C:\Users\admin>

```

- 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
```

- Save screenshot as: task6_windows_known_hosts_cleared_and_empty.png

```

PS C:\Users\admin> Clear-Content $env:USERPROFILE\.ssh\known_hosts
PS C:\Users\admin> type $env:USERPROFILE\.ssh\known_hosts
PS C:\Users\admin>

```

- 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:
ssh username@<server_ip>

```
# Accept the host key prompt (yes) and complete the login (enter password or key passphrase)
```

- Save screenshot as: task6_windows_ssh_accept_hostkey_and_login.png

```

PS C:\Users\admin> ssh hamail@192.168.114.129
The authenticity of host '192.168.114.129 (192.168.114.129)' can't be established.
ED25519 key fingerprint is SHA256:GUcU9+wF1MuDRrKDX9XNLxDFn6c+AcVDMZ8Z5yLlb8Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.114.129' (ED25519) to the list of known hosts.
hamail@192.168.114.129's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-86-generic x86_64)

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

System information as of Fri Nov 21 10:43:20 AM UTC 2025

System load: 0.03          Processes:           227
Usage of /: 85.8% of 9.75GB Users logged in:      1
Memory usage: 23%          IPv4 address for ens33: 192.168.114.129
Swap usage: 0%

```

5. After the successful connection, view the known_hosts file to show the server host key was added (single screenshot):
type \$env:USERPROFILE\.ssh\known_hosts
- Save screenshot as: task6_windows_known_hosts_after_connect.png

```

connection to 192.168.114.129 closed.
PS C:\Users\admin> type $env:USERPROFILE\.ssh\known_hosts
192.168.114.129 ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIB+EaEx-fq/wzKt7JKB1WiU0u+0q1TiwYm0gGptMnoo7I
192.168.114.129 ssh-rsa AAAAB3NzaC1yc2EAAAQABAAABgQCZ2JX9g/pgI01gpmpwwuX0y8pzdXjUY7iGwSI72mF8ucMFVA2koWCl7l2zY9iJ0pQ
FiWv9gu+ZWpMsER981Taco7AaUtB00K/hnasG+9MOBNEB5DAqHbpLxmbQvmLyq4uGVuUpGNZE9Wc89ot79auTgMoO/7sHERr5+Z0+rPwsFV3k0uqEURVsX
M3RQZRj11SRxVB70YlwNQ2lQfjLa/7rKjICR+LrxjBz7g5hurQkcnKVd8yZ4oIh9iKzPUhWRV/Krgen5S46aRmb9+qxpvwHbgqCZ5C6smj1DCCh8CT/q5kr
nExe09tI/le8FDpVJ0uKdSA1U4aqInu3N52DnrSr4ZjmquSSpEvLs4w81lF31BxtPSvpvP702UAgadWLd74x7lMNSxb6mBbt07ihJjsVhZa1Fj78pSMV
qVQRQGqOTNUGOAaVuzzByCQzLionkP92zqdhXh9vBwvJpRopcD2lLQgewUPNRIs6ppTbXotVgKoMfGvVtILEPuk=
192.168.114.129 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoVTItbmlzdHAYNTYAAAIBmlzdHAYNTYAAAABBbfkbANDdaVQFTL11PyCakeDhHo9WbEo
8su87eS00UavElidyy/4V79lLhrmwU/YI85zJqXgEWfiJ2fz1uVv4bc=
PS C:\Users\admin>

```

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:
mkdir -p ~/.ssh
chmod 700 ~/.ssh
> ~/.ssh/authorized_keys
- Save screenshot as: task6_server_clear_authorized_keys.png

```

hamail@ubuntu-lab:~$ mkdir -p ~/.ssh
chmod 700 ~/.ssh
> ~/.ssh/authorized_keys

```

2. 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

```

- Save screenshot as: task6_server_add_key_and_show.png (single screenshot showing the commands and resulting authorized_keys content)

```
hamail@ubuntu-lab:~$ echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIaGzxz/M/3pPqrIa2HEZSK8ZYDutnOC+mt/wUzzbVpI lab_key" >> ~/ssh/authorized_keys
chmod 600 ~/ssh/authorized_keys
cat ~/ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIaGzxz/M/3pPqrIa2HEZSK8ZYDutnOC+mt/wUzzbVpI lab_key
hamail@ubuntu-lab:~$ |
```

3. From Windows host test passwordless login (capture successful login in one screenshot):
ssh username@<server_ip>

- Save screenshot as: task6_ssh_passwordless_login.png

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

System information as of Fri Nov 21 11:02:26 AM UTC 2025

System load: 0.57          Processes: 231
Usage of /: 85.8% of 9.75GB Users logged in: 1
Memory usage: 23%          IPv4 address for ens33: 192.168.114.129
Swap usage: 0%

=> / is using 85.8% 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.

1 update 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

*** System restart required ***
Last login: Fri Nov 21 11:00:08 2025 from 192.168.114.1
```

4. Also demonstrate explicit identity usage (single screenshot):
ssh -i ~/ssh/id_lab7 username@<server_ip>
- Save screenshot as: task6_ssh_with_identity_file.png

```

hamail@ubuntu-lab:~$ ssh -i $env:USERPROFILE\.ssh\id_lab7 hamail@192.168.114.129
Warning: Identity file :USERPROFILE.sshid_lab7 not accessible: No such file or directory.
hamail@192.168.114.129's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-86-generic x86_64)

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

System information as of Fri Nov 21 11:03:40 AM UTC 2025

System load: 0.67          Processes:      235
Usage of /:   85.8% of 9.75GB  Users logged in:   1
Memory usage: 24%           IPv4 address for ens33: 192.168.114.129
Swap usage:   0%

=> / is using 85.8% 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.

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

```

Important notes:

- Do NOT show or upload private key files.
- Ensure server-side permissions are strict: ~/.ssh 700, authorized_keys 600.

📸 Screenshots (required):

task6_windows_sshkey_and_list.png
task6_windows_public_key.png
task6_windows_known_hosts_cleared_and_empty.png
task6_windows_ssh_accept_hostkey_and_login.png
task6_windows_known_hosts_after_connect.png
task6_server_clear_authorized_keys.png
task6_server_add_key_and_show.png
task6_ssh_passwordless_login.png
task6_ssh_with_identity_file.png

Exam Evaluation Questions

Q1: Quick Environment Audit

- Objective: Demonstrate you can inspect the current environment and extract a few key variables.
 - Actions & evidence:
- i. Run a single command to display environment variables and capture its output.
 - Save screenshot: EE_q1_env_all.png

```
hamail@ubuntu-lab:~$ printenv
SHELL=/bin/bash
PWD=/home/hamail
LOGNAME=hamail
XDG_SESSION_TYPE=tty
HOME=/home/hamail
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: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:*.lz=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.tz=01;31:*.tzs=01;31:*.bz=01;31:*.bz2=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:*.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:*.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:*.ogg=01;35:*.mp4=01;35:*.m4v=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:*
.ogg=00;36:*.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:*.rpmmew=00;90:*.rpmmrig=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=192.168.114.1 63083 192.168.114.129 22
LESSCLOSE=/usr/bin/lesspipe %s %
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %
USER=hamail
SHLVL=1
```

- ii. 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.
- Save screenshot: EE_q1_env_filters.png

```
hamail@ubuntu-lab:~$ echo $PATH
echo $LANG
echo $PWD
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
en_US.UTF-8
/home/hamail
hamail@ubuntu-lab:~$ |
```

Q2: Short-lived Student Info

- Objective: Show how temporary environment variables behave (session-scoped).
 - Actions & evidence:
- i. 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).
- Save screenshot: EE_q2_exports.png

```
hamail@ubuntu-lab:~$ export STUDENT_NAME="HamailFatima"
export STUDENT_ROLL_NUMBER="2023-BSE-023"
export STUDENT_SEMESTER="5"
hamail@ubuntu-lab:~$ |
```

- ii. Still in the same session, print the three values with echo (grouped) and capture the outputs in one screenshot.
- Save screenshot: EE_q2_echoes.png

```
hamail@ubuntu-lab:~$ echo $STUDENT_NAME
echo $STUDENT_ROLL_NUMBER
echo $STUDENT_SEMESTER
HamailFatima
2023-BSE-023
5
hamail@ubuntu-lab:~$ |
```

- iii. Use a single printenv|grep command to list any STUDENT_ variables and capture the result.
- Save screenshot: EE_q2_printenv_grep.png

```
hamail@ubuntu-lab:~$ printenv | grep STUDENT_
STUDENT_NAME=HamailFatima
STUDENT_SEMESTER=5
STUDENT_ROLL_NUMBER=2023-BSE-023
hamail@ubuntu-lab:~$ |
```

- iv. 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.
- Save screenshot: EE_q2_after_restart.png

```
hamail@ubuntu-lab:~$ printenv | grep STUDENT_
STUDENT_NAME=HamailFatima
STUDENT_SEMESTER=5
STUDENT_ROLL_NUMBER=2023-BSE-023
hamail@ubuntu-lab:~$ echo $STUDENT_NAME
printenv | grep STUDENT_
HamailFatima
STUDENT_NAME=HamailFatima
STUDENT_SEMESTER=5
STUDENT_ROLL_NUMBER=2023-BSE-023
hamail@ubuntu-lab:~$ |
```

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

- Objective: Demonstrate persistence of environment variables across sessions via shell configuration.
- Actions & evidence:
 - i. Edit ~/.bashrc and append the three STUDENT_* exports. Capture a screenshot of the editor showing the new lines.

- Save screenshot: EE_q3_bashrc_editor.png

```
hamail@ubuntu-lab:~$ nano ~/.bashrc
```

- ii. 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.
- Save screenshot: EE_q3_after_source.png

```

hamail@ubuntu-lab:~$ source ~/.bashrc
echo $STUDENT_NAME
echo $STUDENT_ROLL_NUMBER
echo $STUDENT_SEMESTER
printenv | grep '^STUDENT_'
YourName
12345
5
STUDENT_NAME=YourName
STUDENT_SEMESTER=5
STUDENT_ROLL_NUMBER=12345
hamail@ubuntu-lab:~$ |

```

- iii. Close and re-open a terminal and demonstrate the STUDENT_NAME variable is available (echo and printenv grep together) – capture in one screenshot.
- Save screenshot: EE_q3_after_restart.png

```

hamail@ubuntu-lab:~$ echo $STUDENT_NAME
printenv | grep '^STUDENT_'
YourName
STUDENT_NAME=YourName
STUDENT_SEMESTER=5
STUDENT_ROLL_NUMBER=12345
hamail@ubuntu-lab:~$ |

```

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

- Objective: Demonstrate you can block ping (ICMP echo) traffic using ufw and then re-allow it; show effect from a client.
 - Actions & evidence:
- i. Enable ufw and capture the enable command and status together in one screenshot.
- Save screenshot: EE_q5_ufw_enable_status.png

```

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

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

hamail@ubuntu-lab:~$ |

```

- ii. Add a rule to block ping (ICMP echo) and show ufw status numbered in the same screenshot.
- Suggested
 - command example:
 - sudo ufw deny proto icmp from any to any
 - sudo ufw status numbered
 - Save screenshot: EE_q5_ufw_deny_ping_status.png

```

hamail@ubuntu-lab:~$ sudo ufw deny proto icmp from any to any
sudo ufw status numbered
ERROR: Unsupported protocol 'icmp'
Status: active

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

hamail@ubuntu-lab:~$ |

```

- iii. 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.
- Save screenshot: EE_q5_ping_blocked.png

```

hamail@ubuntu-lab:~$ ping 192.168.114.129
PING 192.168.114.129 (192.168.114.129) 56(84) bytes of data.
64 bytes from 192.168.114.129: icmp_seq=1 ttl=64 time=5.24 ms
64 bytes from 192.168.114.129: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 192.168.114.129: icmp_seq=3 ttl=64 time=0.658 ms
64 bytes from 192.168.114.129: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 192.168.114.129: icmp_seq=5 ttl=64 time=0.155 ms
64 bytes from 192.168.114.129: icmp_seq=6 ttl=64 time=0.160 ms
64 bytes from 192.168.114.129: icmp_seq=7 ttl=64 time=0.130 ms
64 bytes from 192.168.114.129: icmp_seq=8 ttl=64 time=0.067 ms

```

- iv. Re-allow ping (ICMP) (or remove the deny rule) and capture the allow/reload/status sequence in one screenshot.
- Suggested command example:
 - sudo ufw allow proto icmp from any to any
 - sudo ufw reload
 - sudo ufw status
 - Save screenshot: EE_q5_ufw_allow_ping_status.png

```

hamail@ubuntu-lab:~$ sudo ufw allow proto icmp from any to any
sudo ufw reload
sudo ufw status
[sudo] password for hamail:
ERROR: Unsupported protocol 'icmp'
Firewall reloaded
Status: active

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

hamail@ubuntu-lab:~$ |

```

- v. From the client, ping the server again and capture successful replies in one screenshot.
- Save screenshot: EE_q5_ping_success.png

```
hamaill@ubuntu-lab:~$ ping 192.168.114.129
PING 192.168.114.129 (192.168.114.129) 56(84) bytes of data.
64 bytes from 192.168.114.129: icmp_seq=1 ttl=64 time=14.5 ms
64 bytes from 192.168.114.129: icmp_seq=2 ttl=64 time=0.142 ms
64 bytes from 192.168.114.129: icmp_seq=3 ttl=64 time=0.107 ms
64 bytes from 192.168.114.129: icmp_seq=4 ttl=64 time=0.145 ms
64 bytes from 192.168.114.129: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 192.168.114.129: icmp_seq=6 ttl=64 time=0.124 ms
64 bytes from 192.168.114.129: icmp_seq=7 ttl=64 time=0.126 ms
64 bytes from 192.168.114.129: icmp_seq=8 ttl=64 time=0.291 ms
64 bytes from 192.168.114.129: icmp_seq=9 ttl=64 time=0.125 ms
64 bytes from 192.168.114.129: icmp_seq=10 ttl=64 time=0.133 ms
```