

CentOS

Group member:

Chong Hong Yao	2370277-DCS
Wong Chee Hang	2370088-DCS
Ng Kah Dung	2370282-DCS
Quee You Xin	2370283-DCS

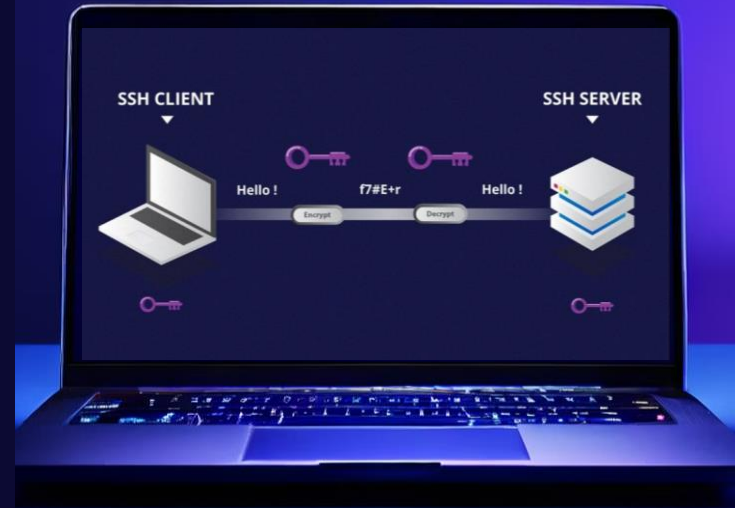


SSH Configuration

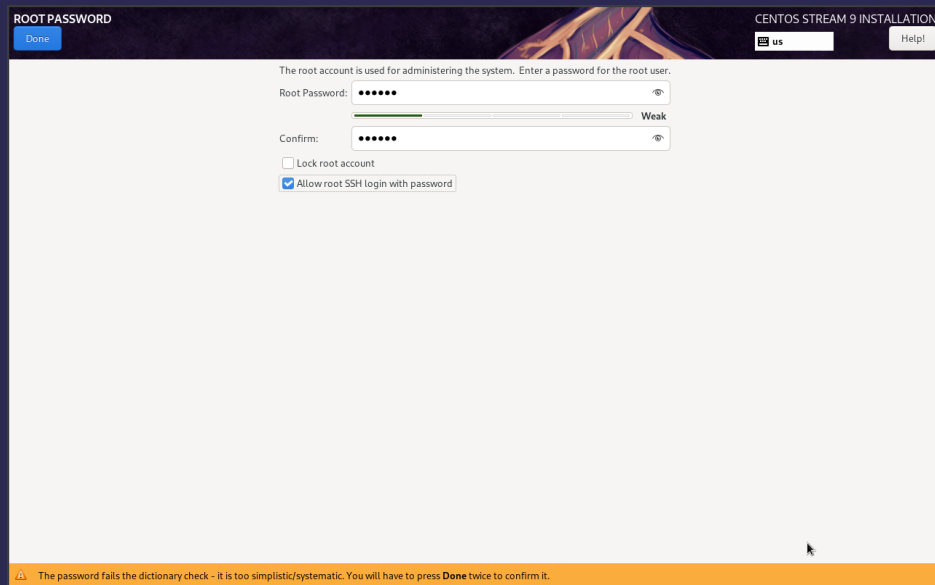
The use of ssh is to a method for securely sending commands to a computer over an unsecured network.

Example:

1. How to setup.
2. How to access.
3. How to test SSH.



Setup SSH




The screenshot shows the 'ROOT PASSWORD' screen in the CentOS Stream 9 installation window. The window title is 'CENTOS STREAM 9 INSTALLATION'. In the top left corner, there is a 'Done' button. In the top right corner, there is a user selection dropdown showing 'us' and a 'Help!' button. The main text reads: 'The root account is used for administering the system. Enter a password for the root user.' Below this, there are two password input fields: 'Root Password:' and 'Confirm:'. The 'Root Password' field has a strength indicator bar that is mostly green but ends with a 'Weak' label. Below the password fields, there are two checkboxes: 'Lock root account' (unchecked) and 'Allow root SSH login with password' (checked). At the bottom of the window, there is an orange warning banner that says: 'The password fails the dictionary check - it is too simplistic/systematic. You will have to press Done twice to confirm it.'


To allow root access


enable root SSH login with password
allowed you to have full access of the
system.

Setup SSH

The root account is used for administering the system. Enter a password for the root user.

Root Password: 

 Weak

Confirm: 

☐ Lock root account

☒ Allow root SSH login with password

To allowed root access

enable root SSH login with password allowed you to have full access of the system.

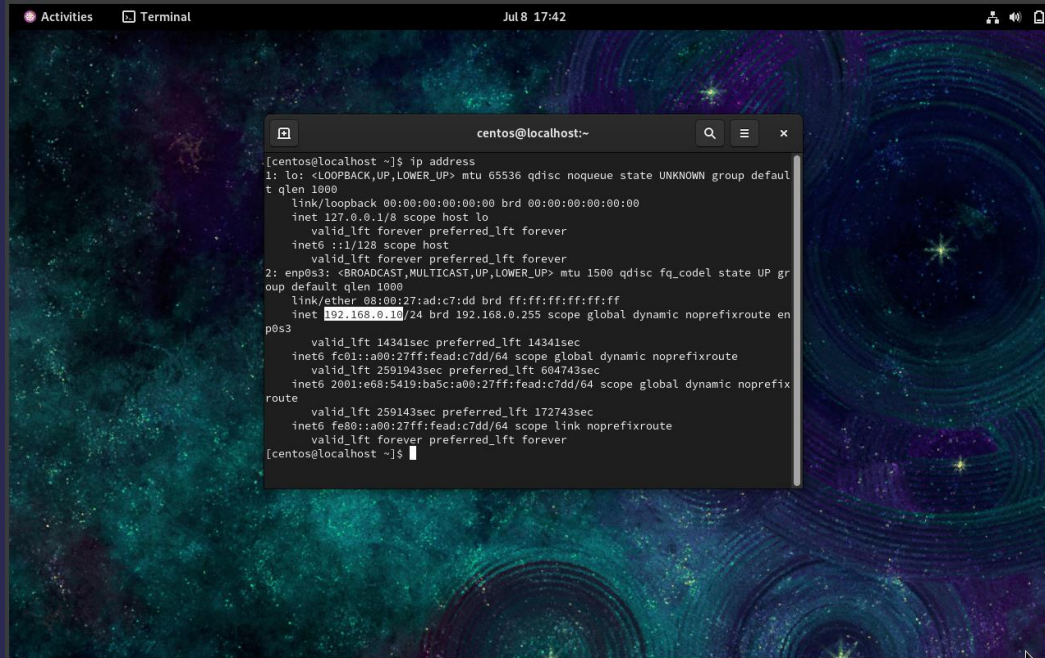
Access SSH

First, open the Terminal then, find the ip address by typing "ip address"

#EXPLAIN

#to get find the Ip address for centos.

#COMMAND: "ip address"

A screenshot of a Linux terminal window with a dark background and a colorful, abstract pattern. The terminal title bar shows 'Activities', 'Terminal', and the date 'Jul 8 17:42'. The prompt is 'centos@localhost:~'. The command 'ip address' has been executed, and the output shows details for the loopback interface 'lo' and the ethernet interface 'enp0s3'. The IP address for 'enp0s3' is 192.168.0.18.

```
centos@localhost:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:c7:dd brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.18/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 14341sec preferred_lft 14341sec
    inet6 fc01::a00:27ff:fead:c7dd/64 scope global dynamic noprefixroute
        valid_lft 2591943sec preferred_lft 604743sec
    inet6 2001:e68:5419:ba5c:a00:27ff:fead:c7dd/64 scope global dynamic noprefixroute
        valid_lft 259143sec preferred_lft 172743sec
    inet6 fe80::a00:27ff:fead:c7dd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
centos@localhost:~$
```

Access SSH

```
centos@localhost:~  
[centos@localhost ~]$ ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:ad:c7:dd brd ff:ff:ff:ff:ff:ff  
    inet 192.168.0.10/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 14341sec preferred_lft 14341sec  
    inet6 fc01::a00:27ff:fead:c7dd/64 scope global dynamic noprefixroute  
        valid_lft 2591943sec preferred_lft 604743sec  
    inet6 2001:e68:5419:ba5c:a00:27ff:fead:c7dd/64 scope global dynamic noprefixroute  
        valid_lft 259143sec preferred_lft 172743sec  
    inet6 fe80::a00:27ff:fead:c7dd/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
[centos@localhost ~]$
```

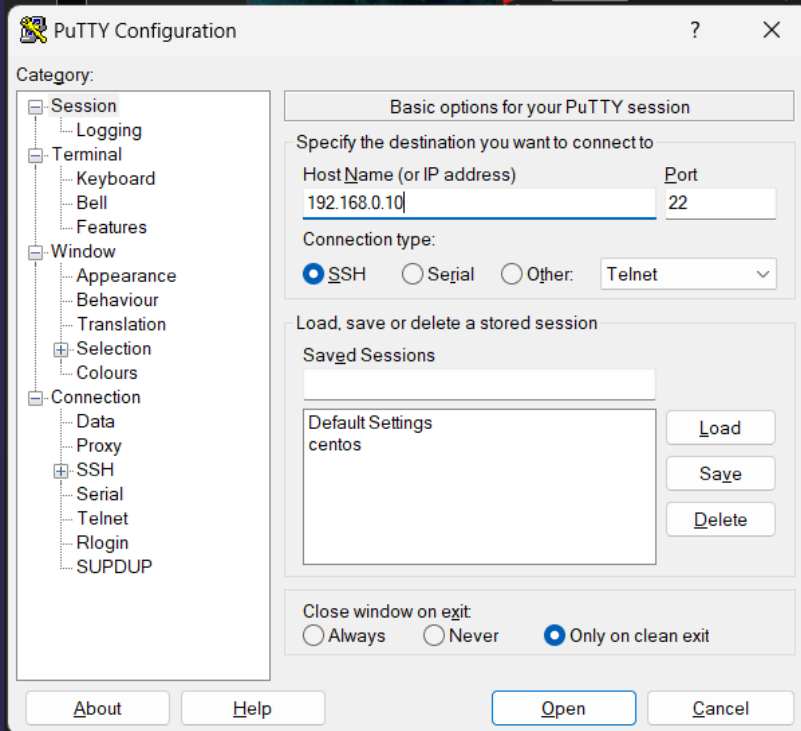
First, open the Terminal then, find the ip address by typing "ip address"

Access SSH

```
centos@localhost:~  
[centos@localhost ~]$ ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:ad:c7:dd brd ff:ff:ff:ff:ff:ff  
    inet 192.168.0.10/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 14341sec preferred_lft 14341sec  
    inet6 fc01::a00:27ff:fead:c7dd/64 scope global dynamic noprefixroute  
        valid_lft 2591943sec preferred_lft 604743sec  
    inet6 2001:e68:5419:ba5c:a00:27ff:fead:c7dd/64 scope global dynamic noprefixroute  
        valid_lft 259143sec preferred_lft 172743sec  
    inet6 fe80::a00:27ff:fead:c7dd/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
[centos@localhost ~]$
```

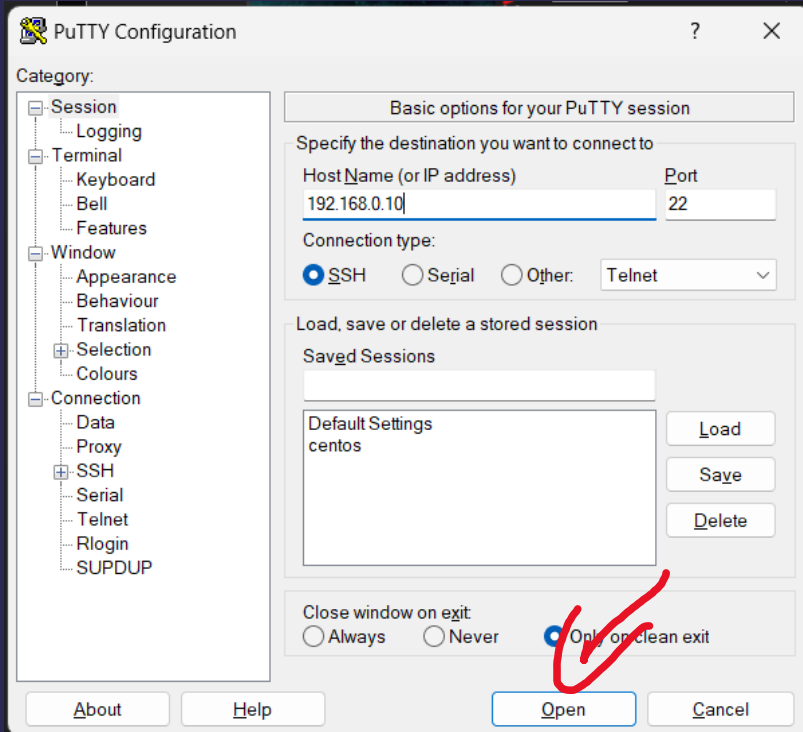
Second, remember the ip address.

Access SSH



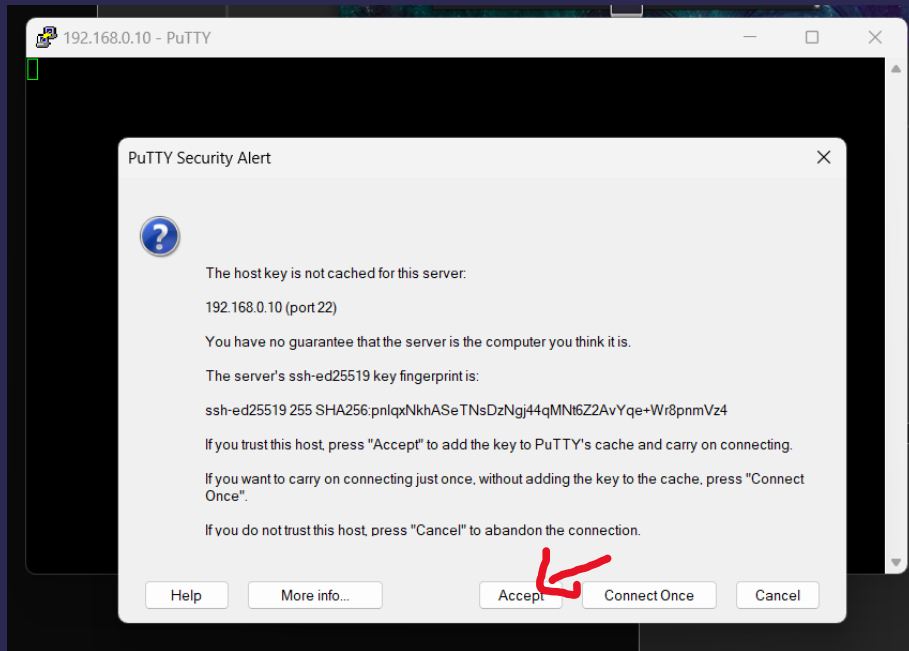
Third, open PuTTY on windows and type the ip address you remembered.

Access SSH



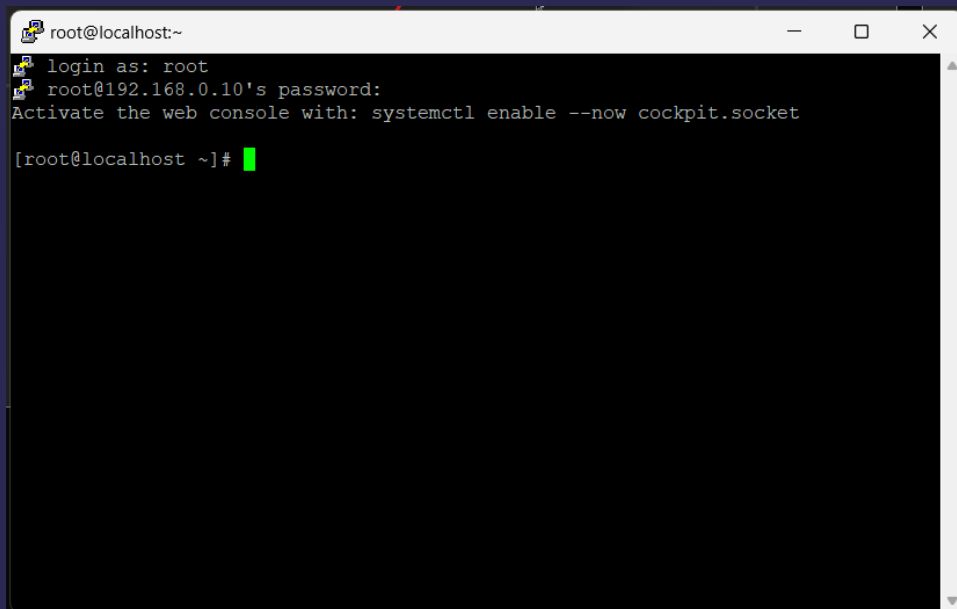
Fourth, click open

Access SSH



This will popup and need to click accept.

Access SSH

A terminal window titled 'root@localhost:~' with standard window controls. The terminal shows an SSH login sequence: 'login as: root', 'root@192.168.0.10's password:', and a system message 'Activate the web console with: systemctl enable --now cockpit.socket'. The prompt '[root@localhost ~]#' is followed by a green cursor.

```
root@localhost:~  
login as: root  
root@192.168.0.10's password:  
Activate the web console with: systemctl enable --now cockpit.socket  
[root@localhost ~]#
```

Now you are in SSH type your user name and password to login.

1st Test SSH (shutdown)

Use the command "shutdown -h now" to test if the shutdown -h now command only works in root.

```
root@localhost:~#  
login as: root  
root@192.168.1.24's password:  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last login: Wed Jul 10 13:11:24 2024 from 192.168.1.23  
[root@localhost ~]# shutdown -h now  
[root@localhost ~]#
```

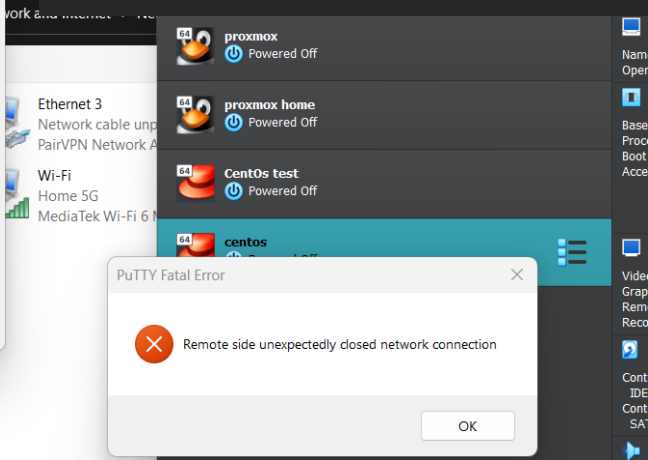
#EXPLAIN

#shutdown the OS.

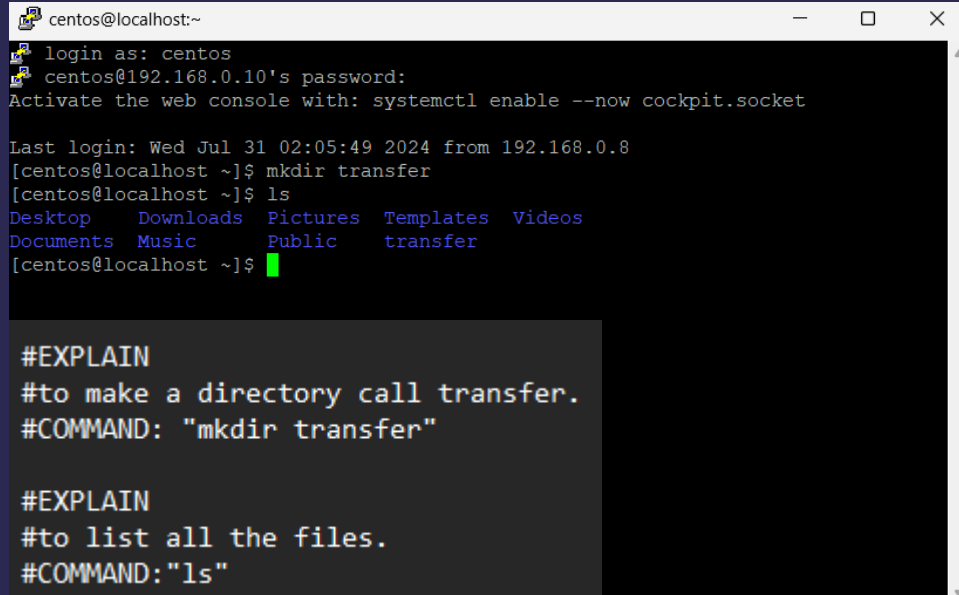
#-h to get the system be halted or bring it to a abrupt stop.

#now is to tell to do it now.

#COMMAND: "shutdown -h now"



2th Test SSH (windows to centos)

A terminal window titled 'centos@localhost:~' showing the process of logging into a CentOS system via SSH. The user 'centos' is prompted for a password. After login, the system shows the last login time and the user's home directory. The user then runs 'mkdir transfer' to create a new directory, followed by 'ls' to list the contents of the home directory, which now includes the 'transfer' folder. A dark grey box at the bottom of the terminal contains explanatory text for the commands.

```
centos@localhost:~  
login as: centos  
centos@192.168.0.10's password:  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last login: Wed Jul 31 02:05:49 2024 from 192.168.0.8  
[centos@localhost ~]$ mkdir transfer  
[centos@localhost ~]$ ls  
Desktop    Downloads  Pictures   Templates  Videos  
Documents  Music      Public     transfer  
[centos@localhost ~]$
```

#EXPLAIN
#to make a directory call transfer.
#COMMAND: "mkdir transfer"

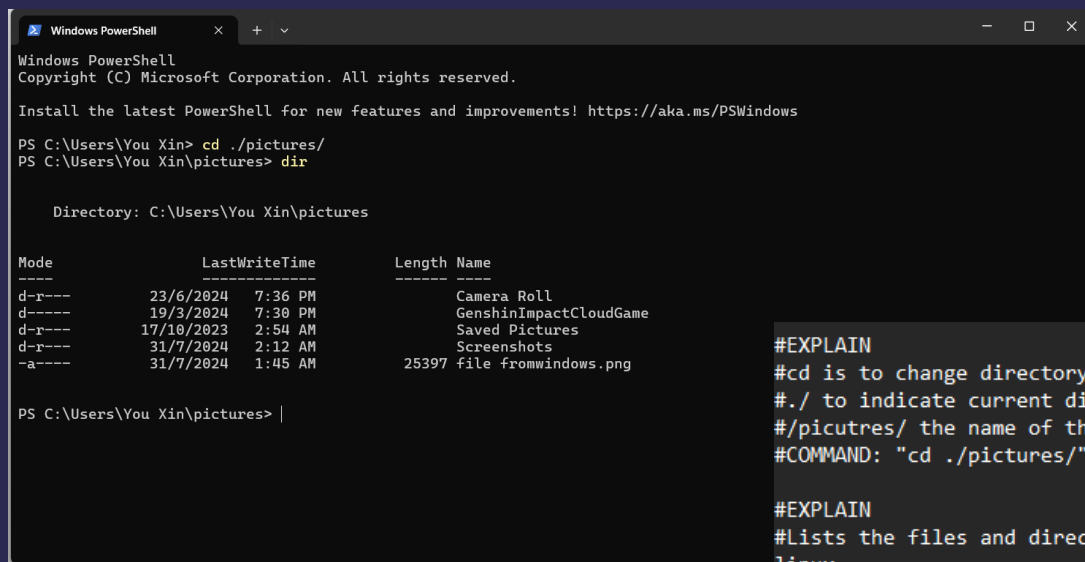
#EXPLAIN
#to list all the files.
#COMMAND: "ls"

First, you are in SSH type your user name and password to login.

Second, type "mkdir transfer" to make a folder name transfer.

Third, use the command "ls" to see the folder you have made.

2th Test SSH (windows to centos)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\You Xin> cd ./pictures/
PS C:\Users\You Xin\pictures> dir

    Directory: C:\Users\You Xin\pictures

Mode                LastWriteTime         Length Name
----                -
d-r-----          23/6/2024   7:36 PM             Camera Roll
d-----          19/3/2024   7:30 PM   GenshinImpactCloudGame
d-r-----         17/10/2023   2:54 AM             Saved Pictures
d-r-----          31/7/2024   2:12 AM             Screenshots
-a-----          31/7/2024   1:45 AM    25397 file fromwindows.png

PS C:\Users\You Xin\pictures> |
```

First open Windows PowerShell.

Second, type "cd ./pictures/" to change directory then select a file you want to transfer to centos.

#EXPLAIN

#cd is to change directory.

#./ to indicate current directory.

#/picutres/ the name of the folder.

#COMMAND: "cd ./pictures/"

#EXPLAIN

#Lists the files and directories in the specified directory same as "ls" in linux.

#COMMAND:"dir"

2th Test SSH (windows to centos)

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\You Xin> cd ./pictures/
PS C:\Users\You Xin\pictures> dir

Directory: C:\Users\You Xin\pictures

Mode                LastWriteTime         Length Name
----                -
d-r---           23/6/2024   7:36 PM             Camera Roll
d-----          19/3/2024   7:30 PM      GenshinImpactCloudGame
d-r---          17/10/2023   2:54 AM             Saved Pictures
d-r---          31/7/2024   2:14 AM             Screenshots
-a----          31/7/2024   1:45 AM      25397 file fromwindows.png

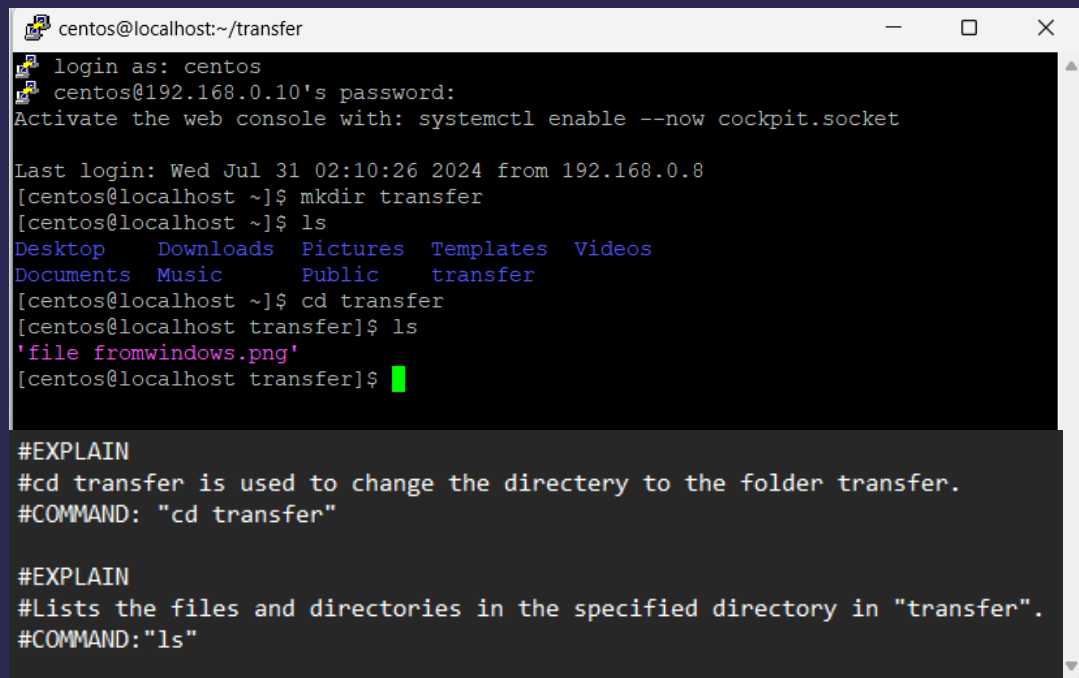
PS C:\Users\You Xin\pictures> scp "file fromwindows.png" centos@192.168.0.10:/home/centos/transfer
centos@192.168.0.10's password:
file fromwindows.png                                100%  25KB   4.9MB/s   00:00
PS C:\Users\You Xin\pictures> |
```

#EXPLAIN

#scp stands for Secure Copy Protocol used for securely transferring files and directories between computers over a network.
#"file fromwindows.png" is the file you want to transfer.
#centos@192.168.0.10:/home/centos/transfer is where you want to send to and what folder you want your sent file to be in.
#/picutres/ the name of the folder.
#COMMAND: "scp "file fromwindows.png" centos@192.168.0.10:/home/centos/transfer"

First Type "scp "file fromwindows.png"
centos@192.168.0.10:/home/centos/transfer
sfer" to transfer the file to centos then
wait the progress bar to finish.

2th Test SSH (windows to centos)



```
centos@localhost:~/transfer
login as: centos
centos@192.168.0.10's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Jul 31 02:10:26 2024 from 192.168.0.8
[centos@localhost ~]$ mkdir transfer
[centos@localhost ~]$ ls
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music      Public    transfer
[centos@localhost ~]$ cd transfer
[centos@localhost transfer]$ ls
'file fromwindows.png'
[centos@localhost transfer]$
```

#EXPLAIN
#cd transfer is used to change the directory to the folder transfer.
#COMMAND: "cd transfer"

#EXPLAIN
#Lists the files and directories in the specified directory in "transfer".
#COMMAND: "ls"

To check the file you have transfer go back to PuTTY. First, type "cd transfer" to go to change directory.

Second, type "ls" to see list the file in transfer.

2th Test SSH (Centos To Windows)

```
centos@localhost:~/transfer
login as: centos
centos@192.168.0.10's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Jul 31 02:10:26 2024 from 192.168.0.8
[centos@localhost ~]$ mkdir transfer
[centos@localhost ~]$ ls
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music      Public    transfer
[centos@localhost ~]$ cd transfer
[centos@localhost transfer]$ ls
'file fromwindows.png'
[centos@localhost transfer]$ echo "this file from centos" >> filefromcentos.txt
[centos@localhost transfer]$ cat filefromcentos.txt
this file from centos
[centos@localhost transfer]$
```

```
#EXPLAIN
#echo used to output the string "this file from centos"
#>> filefromcentos.txt the folder name.
#COMMAND: "echo "this file from centos" >> filefromcentos.txt"
```

```
#EXPLAIN
#to display the contents of the file form filefromcentos.txt.
#COMMAND:"cat filefromcentos.txt"
```

First create a file with the word this file is from centos inside the file and the file name is filefromcentos.txt by using the command "echo "this file from centos" >> filefromcentos.txt"

Second, use the command "cat filefromcentos.txt" to read the file contents.

2th Test SSH (Centos To Windows)

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\You Xin> cd ./pictures/
PS C:\Users\You Xin\pictures> dir

Directory: C:\Users\You Xin\pictures

Mode                LastWriteTime         Length Name
----                -
d-r-----          23/6/2024   7:36 PM             Camera Roll
d-r-----          19/3/2024   7:30 PM      GenshinImpactCloudGame
d-r-----         17/10/2023   2:54 AM             Saved Pictures
d-r-----          31/7/2024   2:56 AM             Screenshots
-a-----          31/7/2024   1:45 AM      25397 file fromwindows.png

PS C:\Users\You Xin\pictures> scp centos@192.168.0.10:/home/centos/transfer/filefromcentos.txt .
centos@192.168.0.10's password:
filefromcentos.txt                                     100% 22      8.4KB/s   00:00
PS C:\Users\You Xin\pictures> |
```

#EXPLAIN

#change directory to pictures.

#COMMAND: "cd ./pictures/"

#EXPLAIN

#list the files contents.

#COMMAND:"dir"

#EXPLAIN

#to transfer the file"filefromcentos.txt" from centos to windows.

#COMMAND:"scp centos@192.168.0.10:/home/centos/transfer/filefromcentos.txt"

First go back to Windows PowerShell
type "cd ./pictures/" to change direcerty.

Second, type "scp

[centos@192.168.0.10:/home/centos/transfer/filefromcentos.txt](#) ." Then type

password the file will transfer.

2th Test SSH (Centos To Windows)

```
Windows PowerShell
-----
d-r---      23/6/2024   7:36 PM          Camera Roll
d-----      19/3/2024   7:30 PM          GenshinImpactCloudGame
d-r---      17/10/2023   2:54 AM          Saved Pictures
d-r---      31/7/2024   2:56 AM          Screenshots
-a-----      31/7/2024   1:45 AM      25397 file fromwindows.png

PS C:\Users\You Xin\pictures> scp centos@192.168.0.10:/home/centos/transfer/filefromcentos.txt .
centos@192.168.0.10's password:
filefromcentos.txt                                100% 22      8.4KB/s   00:00
PS C:\Users\You Xin\pictures> dir

Directory: C:\Users\You Xin\pictures

Mode                LastWriteTime         Length Name
-----
d-r---      23/6/2024   7:36 PM          Camera Roll
d-----      19/3/2024   7:30 PM          GenshinImpactCloudGame
d-r---      17/10/2023   2:54 AM          Saved Pictures
d-r---      31/7/2024   3:08 AM          Screenshots
-a-----      31/7/2024   1:45 AM      25397 file fromwindows.png
-a-----      31/7/2024   3:08 AM         22 filefromcentos.txt

PS C:\Users\You Xin\pictures> cat filefromcentos.txt
this file from centos
PS C:\Users\You Xin\pictures> |
```

To check the file has been transfer or not

First, type "dir" to check.

Second, type "cat filefromcentos.txt" to read the file contents.

#EXPLAIN

#list the files contents.

#COMMAND:"dir"

#EXPLAIN

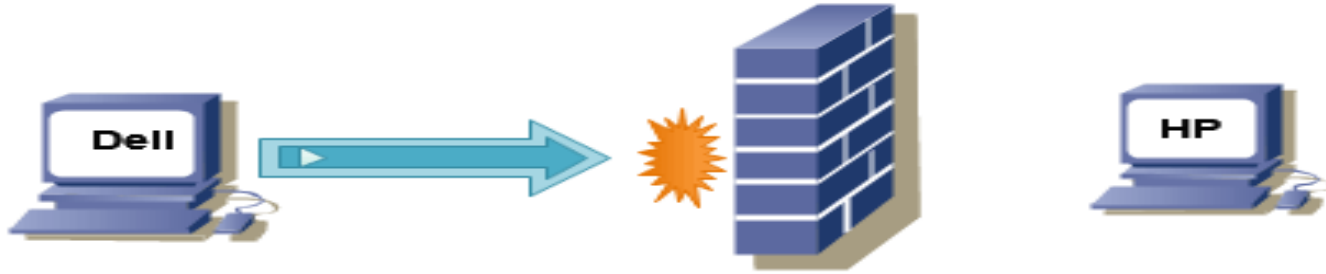
#to see the file contents in filefromcentos.txt.

#COMMAND:"cat filefromcentos.txt"

Firewall Configuration

A solid teal horizontal bar located below the title text.

Firewall Configuration on CentOS



Example :

- 1.Blocking and unblocking SHH Remote Access
- 2.blocking and unblocking IP Address

Firewall Status Checking

```
sudo systemctl status firewalld
```

```
#sudo:Executes the command with superuser privileges.
```

```
#systemctl:control and manage system services
```

```
#status firewalld:display the firewall current status
```

```
Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset
Active: active (running) since Thu 2024-07-04 10:24:23 +08; 1h 2min ago
Docs: man:firewalld(1)
Main PID: 863 (firewalld)
Tasks: 4 (limit: 62672)
Memory: 47.7M
CPU: 1.462s
CGroup: /system.slice/firewalld.service
└─863 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
```

```
sudo systemctl start firewalld
```

```
#start firewalld:active the firewall
```

Example 1

Blocking SSH Remote Access

1



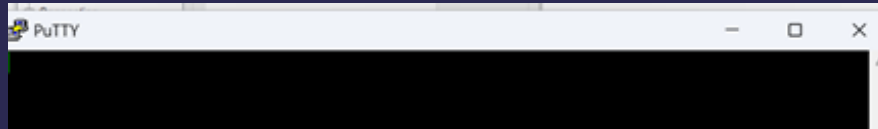
2

```
sudo firewall-cmd --zone=public --remove-service=ssh --permanent
#firewall-cmd: Used to add, remove, or modify firewall rules and services.
#zone=public :Targets the public firewall zone
#remove-service=ssh: block the ssh service
#permanent:Still valid after restarting the system
```

3

```
sudo firewall-cmd --reload
#reload:update the new rules
```

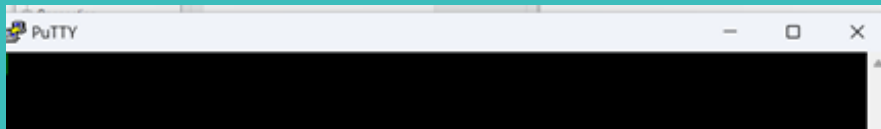
4



Example 1

Unlocking SSH Remote Access

1



2

```
sudo firewall-cmd --zone=public --add-service=ssh --permanent  
#--add-service=ssh:unlock the ssh service
```

3

```
sudo firewall-cmd --reload  
#reload:update the new rules
```

4



4

```
sudo firewall-cmd --zone=public --remove-rich-rule='rule family="ipv4" source address="192.168.1.11" drop' --permanent  
#--remove-rich-rule:remove a firewall rule
```

1

```
C:\Users\hongy>ping 192.168.1.27
```

```
Pinging 192.168.1.27 with 32 bytes of data:  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 192.168.1.27:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Example 2

Blocking and unblock from a
specific IP address
(192.168.1.11)

3

```
C:\Users\hongy>ping 192.168.1.27
```

```
Pinging 192.168.1.27 with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.
```

```
Ping statistics for 192.168.1.27:  
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

2

```
sudo firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" source address="192.168.1.11" drop' --permanent  
#--add-rich-rule: Adds a firewall rule.  
# 'rule family="ipv4" source address="192.168.1.11" drop':  
# family="ipv4": Applies to IPv4 traffic.  
# source address="192.168.1.11": Targets traffic from IP 192.168.1.11.  
# drop: Blocks traffic from this IP address.
```

Example 2

```
C:\Users\hongy>ping 192.168.1.27
```

```
Pinging 192.168.1.27 with 32 bytes of data:  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64  
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 192.168.1.27:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Confirm that the IP connection is successful

Add Firewall Rule to Block IP

```
[hong123@localhost ~]$ sudo firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" source address="192.168.1.11" drop' --permanent
```

Confirm IP Connection is Blocked

```
C:\Users\hongy>ping 192.168.1.27
```

```
Pinging 192.168.1.27 with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.
```

```
Ping statistics for 192.168.1.27:  
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Remove Firewall Rule

```
[hong123@localhost ~]$ sudo firewall-cmd --zone=public --remove-rich-rule='rule family="ipv4" source address="192.168.1.11" drop' --permanent
```

1

2

3

4

A laptop is shown from a slightly elevated angle, its screen displaying a complex data visualization. The screen is filled with multiple overlapping charts, including line graphs, bar charts, and scatter plots, all rendered in a dark theme with various colors like blue, green, and orange. The data appears to be financial or technical in nature, with many small text labels and numerical values. The laptop is placed on a dark surface, and the background is a blurred cityscape at night, with warm, out-of-focus lights creating a bokeh effect. The overall scene is dimly lit, with the primary light source being the laptop screen and the ambient city lights.

-
- A laptop is shown from a slightly elevated angle, its screen displaying a complex data visualization. The screen is filled with multiple overlapping charts, including line graphs, bar charts, and scatter plots, all rendered in a dark theme with various colors like blue, green, and red. The data appears to be financial or technical in nature, with many small text labels and numerical values. The laptop is placed on a dark surface, and the background is a blurred cityscape at night, with warm, out-of-focus lights creating a bokeh effect. The overall scene is dimly lit, with the primary light source being the laptop screen and the ambient city lights.

3) Start sql service and check the status of that function or not?

1. For start the sql service, we need to use cmd 'sudo systemctl start mysqld.service'.

```
#EXPLAIN
#systemctl:utility to control the systemd system
#install mysql-server:install mysql-server
#start:Start the specified service
#mysqld.service:service name

#command:'sudo systemctl start mysqld.service'|
```

2. For check the status, we can use the cmd 'sudo systemctl status mysqld'.

```
#EXPLAIN
#STATUS:display the status of the specified service.
#mysqld:server service name

#command:'sudo systemctl status mysqld'|
```

```
[root@localhost hang]# sudo systemctl start mysqld.service
[root@localhost hang]# sudo systemctl status mysqld
● mysqld.service - MySQL 8.0 database server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; disabled; preset: >
   Active: active (running) since Sat 2024-07-27 12:50:44 +08; 43s ago
     Process: 6833 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, st>
     Process: 6855 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mysqld.service>
    Main PID: 6930 (mysqld)
      Status: "Server is operational"
        Tasks: 38 (limit: 23020)
       Memory: 456.2M
          CPU: 2.727s
      CGroup: /system.slice/mysqld.service
              └─6930 /usr/libexec/mysqld --basedir=/usr

Jul 27 12:50:37 localhost.localdomain systemd[1]: Starting MySQL 8.0 database s>
Jul 27 12:50:37 localhost.localdomain mysql-prepare-db-dir[6855]: Initializing >
Jul 27 12:50:44 localhost.localdomain systemd[1]: Started MySQL 8.0 database se>
```

3. If there has a error, we can checking the sql server intallation by using cmd 'sudo journalctl -u mysql-server'.

```
#EXPLAIN
#journalctl:querying and displaying logs
#-u:Specifies the unit to filter the logs
#mysql-server:the service unit name

#command:'sudo journalctl -u mysql-server'
```

4) Enable the service start automatically by using the cmd 'sudo systemctl enable mysqld'.

```
[root@localhost hang]# (sudo systemctl enable mysqld )  
Created symlink /etc/systemd/system/multi-user.target.wants/mysqld.service → /usr/lib/systemd/system/mysqld.service.  
[root@localhost hang]# sudo mysql_secure_intallation
```

```
#EXPLAIN  
#systemctl:utility to control the systemd system  
#enable:enable a service starts automatically  
#mysqld:service name  
|  
#command:'sudo systemctl enable mysqld'
```

5) secure our MySQL installation. We can use the cmd 'sudo mysql_secure_installation' to do this. Next, enter 'N' and set up a password.

```
[root@localhost hang]# sudo mysql_secure_installation
```

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords and improve security. It checks the strength of password and allows the users to set only those passwords which are secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n
Please set the password for root here.

New password: _____

Re-enter new password: _____

#EXPLAIN

#sudo:super user

#mysql_secure_installation:

improve the security of the MySQL server

#command:'sudo mysql_secure_installation'

6) Establish the connection with mysql using cmd

'mysql -u root -p'

Now we can use mysql to create database.

```
[root@localhost hang]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.36 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

#EXPLAIN

#mysql:MySQL command-line client

#-u root:log in as the MySQL root user

#-p:prompt you for the root user's password

#command:'mysql -u root -p'

Function testing

Create Database

```
mysql> CREATE DATABASE ahlimbakery;  
Query OK, 1 row affected (0.00 sec)
```

Create table

```
mysql> CREATE TABLE Employees (  
-> EmployeeID INT PRIMARY KEY,  
-> EmployeeName varchar(50),  
-> Position varchar(50)  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE Products (  
-> ProductID INT PRIMARY KEY,  
-> ProductName VARCHAR(50),  
-> ProductPrice INT  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE Customers (  
-> CustomerID INT PRIMARY KEY,  
-> CustomerName VARCHAR(50),  
-> Phone INT  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_ahlimbakery |  
+-----+  
| Customers              |  
| Employees              |  
| Products               |  
+-----+  
3 rows in set (0.00 sec)
```

Insert simple data to the table

```
mysql> INSERT INTO Employees(EmployeeID, EmployeeName, Position)
-> VALUES
-> (1, 'Aiman', 'Staff'),
-> (2, 'Aina', 'Staff'),
-> (3, 'Jack', 'Staff'),
-> (4, 'Jacky', 'Manager')
-> ;
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Customers (CustomerID, CustomerName, Phone)
-> VALUES
-> (1, 'xiang', '0136845458'),
-> (2, 'hang', '0129874210');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Products (ProductID, ProductName, ProductPrice)
-> VALUES
-> (1, 'Chocolate Cake', 45),
-> (2, 'Vanilla Cake', 45),
-> (3, 'Red Velvet Cake', 35),
-> (4, 'Brioche Bun', 5),
-> (5, 'Sesame Seed Bun', 6),
-> (6, 'Kaiser Roll', 8);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

DML

SELECT STATMENT

```
mysql> SELECT * FROM Products;
```

ProductID	ProductName	ProductPrice
1	Chocolate Cake	45
2	Vanilla Cake	45
3	Red Velvet Cake	35
4	Brioche Bun	5
5	Sesame Seed Bun	6
6	Kaiser Roll	8

6 rows in set (0.00 sec)

DELETE STATMENT

```
mysql> DELETE FROM Products
```

```
-> WHERE ProductID = 6;
```

```
Query OK, 1 row affected (0.01 sec)
```

ProductID	ProductName	ProductPrice
1	Chocolate Cake	45
2	Vanilla Cake	45
3	Red Velvet Cake	35
4	Brioche Bun	5
5	Sesame Seed Bun	6

5 rows in set (0.00 sec)

UPDATE STATEMENT

```
mysql> UPDATE Products
      -> SET ProductPrice = 50
      -> WHERE ProductName = 'Vanilla Cake';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM Products;
```

ProductID	ProductName	ProductPrice
1	Chocolate Cake	45
2	Vanilla Cake	50
3	Red Velvet Cake	35
4	Brioche Bun	5
5	Sesame Seed Bun	6

5 rows in set (0.00 sec)

Git Configuration and Usage



1. Download git on CentOS. Type the command “sudo yum install git” and type y to confirm.

```
Aug 7 13:
kahdung@loca

[kahdung@localhost ~]$ sudo yum install git
CentOS Stream 9 - BaseOS      2.6 kB/s | 6.0 kB  00:02
CentOS Stream 9 - BaseOS      613 kB/s | 8.2 MB 00:13
CentOS Stream 9 - AppStream    2.8 kB/s | 6.1 kB  00:02
CentOS Stream 9 - AppStream    588 kB/s | 20 MB  00:34
CentOS Stream 9 - Extras packages 4.4 kB/s | 5.9 kB 00:01
CentOS Stream 9 - Extras packages 12 kB/s | 18 kB  00:01
Dependencies resolved.

=====
Package      Arch      Version      Repository      Size
=====
Installing:
git          x86_64    2.43.5-1.el9 appstream       51 k
Installing dependencies:
git-core     x86_64    2.43.5-1.el9 appstream       4.4 M
git-core-doc noarch    2.43.5-1.el9 appstream       2.9 M
perl-Error    noarch    1:0.17029-7.el9 appstream       42 k
perl-Git      noarch    2.43.5-1.el9 appstream       38 k
perl-TermReadKey 2.38-11.el9 appstream       37 k
perl-lib      x86_64    0.65-481.el9 appstream       14 k

Transaction Summary
=====
Install 7 Packages

Total download size: 7.5 M
Installed size: 38 M
Is this ok [y/N]: y
Downloading Packages:
(1/7): git-2.43.5-1.el9.x86_64.rpm    195 kB/s | 51 kB  00:00
(2/7): perl-Error-0.17029-7.el9.noarch.rpm 480 kB/s | 42 kB  00:00
(3/7): perl-git-2.43.5-1.el9.noarch.rpm  183 kB/s | 38 kB  00:00
(4/7): perl-TermReadKey-2.38-11.el9.x86_64.rpm 102 kB/s | 37 kB  00:00
(5/7): perl-lib-0.65-481.el9.x86_64.rpm   48 kB/s | 14 kB  00:00
(6/7): git-core-2.43.5-1.el9.x86_64.rpm  500 kB/s | 4.4 MB 00:09
(7/7): git-core-doc-2.43.5-1.el9.noarch.rpm 270 kB/s | 2.9 MB 00:11
Total                                     659 kB/s | 7.5 MB 00:11

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
```

#EXPLAIN

sudo: Runs the command as an administrator.
yum: The tool that installs software on Linux.
install: Tells yum what to do
git: The name of the software want to install

Command: 'sudo yum install git'

2. Check version.

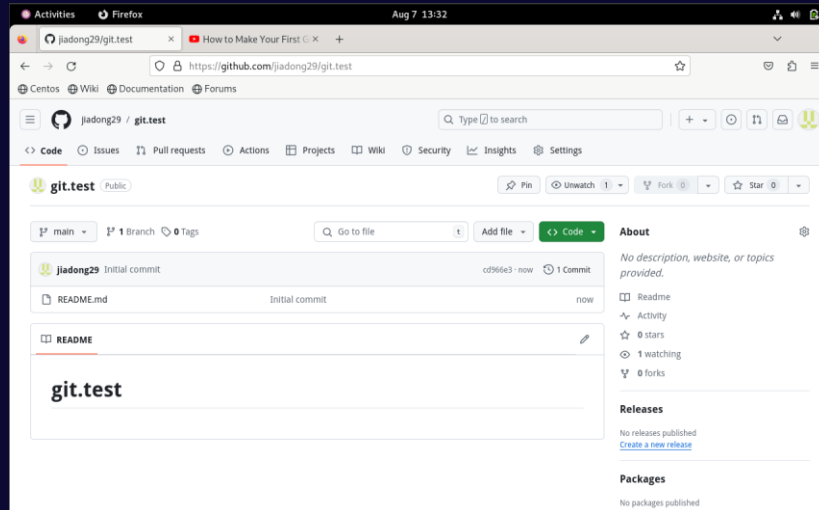
```
[kahdung@localhost ~]$ git --version
git version 2.43.5
[kahdung@localhost ~]$
```

#EXPLAIN

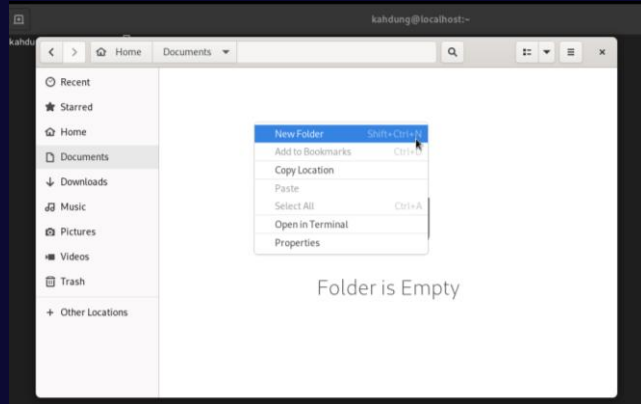
#git: Git version.
#--version: show version of the software is installed.

Command: 'git --version'

3. Create a new repository.



4. Open the file to create a new folder.



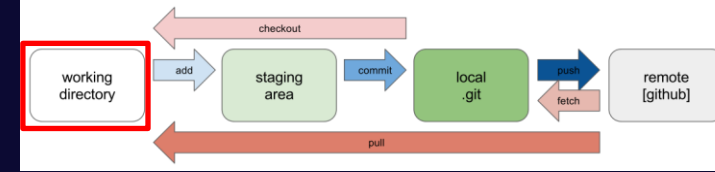
5. Open the Terminal view the current directory in the terminal. Clone a git repository Because we don't have a local repository yet, not even a working directory, So we need to create a version and upload it.

```
Aug 7 13:37
kahdung@localhost:~/Documents

[kahdung@localhost Git test]$ pwd
/home/kahdung/Documents/Git test

[kahdung@localhost Git test]$ git clone https://github.com/jiadong29/git.test.git
Cloning into 'git.test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
[kahdung@localhost Git test]$
```

(Note: In the original image, a red arrow points from the URL in the terminal to the 'Clone' dialog box, and another red arrow points from the 'Clone' dialog box to the terminal output.)



6. Type 'ls' to check the git file.

```
[kahdung@localhost Git test]$ ls
git.test
```

#EXPLAIN

ls: List Files and Directories

Command: 'ls'

#EXPLAIN

pwd: find the current directory in your terminal

Command: 'pwd'

#EXPLAIN

git: Git version

clone: Cloning a local or remote repository.

Command: 'git clone'

7. Type 'touch' file name (hello.java) to create a folder and put some things in folder.

#EXPLAIN

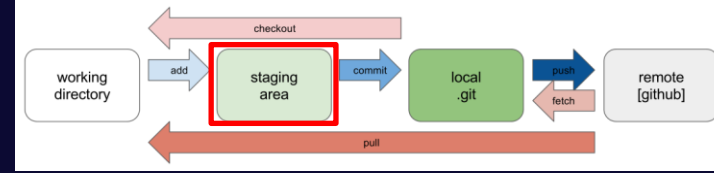
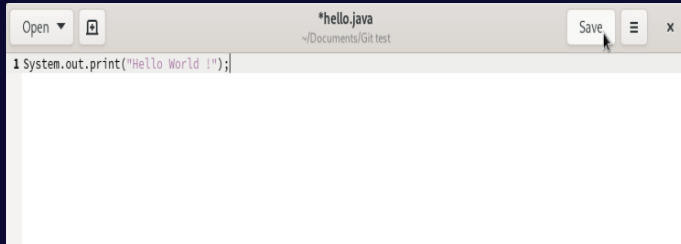
touch: create a file without any content.

Command: 'touch'

```
[kahdung@localhost Git test]$ ls  
git.test
```



```
[kahdung@localhost Git test]$ touch hello.java  
[kahdung@localhost Git test]$ ls  
git.test hello.java
```



8. Type 'git add' file name (hello.java) to adds the content from the working directory to the staging area for the next commit.

```
[kahdung@localhost git.test]$ git add hello.java
```

Check the status of the working directory and staging area.

```
[kahdung@localhost git.test]$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   hello.java
```

Successfully !

#EXPLAIN

git: Git version

add: Add file content to the index.

Command: 'git add'

#EXPLAIN

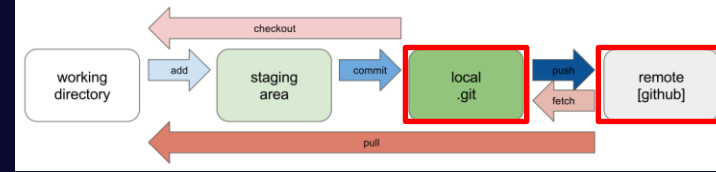
git: Git version

status: Displays the state of the working directory and the staging area.

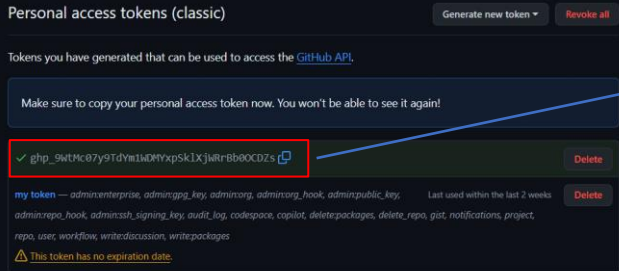
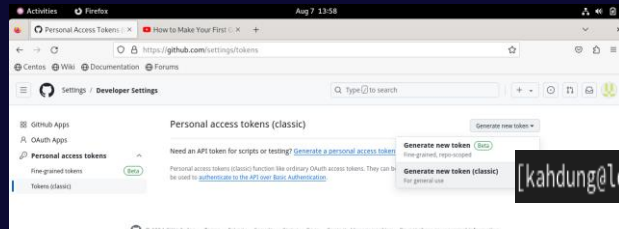
Command: 'git status'

9. Type 'git commit -m' it will save changes the file we made and add a commit message indicating that a new feature has been added to the login page.

```
[kahdung@localhost git.test]$ git commit -m "first java code"
[main a8d51bc] first java code
1 file changed, 1 insertion(+)
create mode 100644 hello.java
```



10. Create a new TOKEN in the Github settings.



Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Type 'git remote set-url origin' to manage set of tracked repositories.

```
[kahdung@localhost git.test]$ git remote set-url origin https://ghp_TxEN0kmr9qaiV3tmK9kLhiB0x7b7Z4b04pA@github.com/jiadong29/git.test
```

repositories

User name

#EXPLAIN

git: Git version

remote: Manage set of tracked repositories.

set-url: modifies the existing URL associated with the "origin" remote.

origin: typically the default name given to the remote repository.

Command: 'git remote set-url origin'

12. Pushes local changes (commits) to the remote repository name "origin" on the branch called "main".

```
[kahdung@localhost git.test]$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 3 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jiadong29/git.test
   cd966e3..a8d51bc  main -> main
[kahdung@localhost git.test]$
```

#EXPLAIN

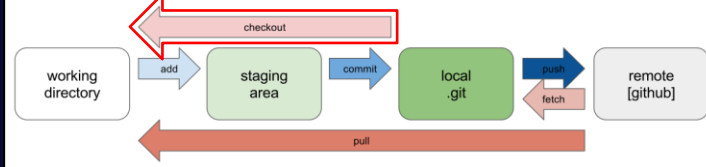
git: Git version

push: Update remote refs along with associated objects.

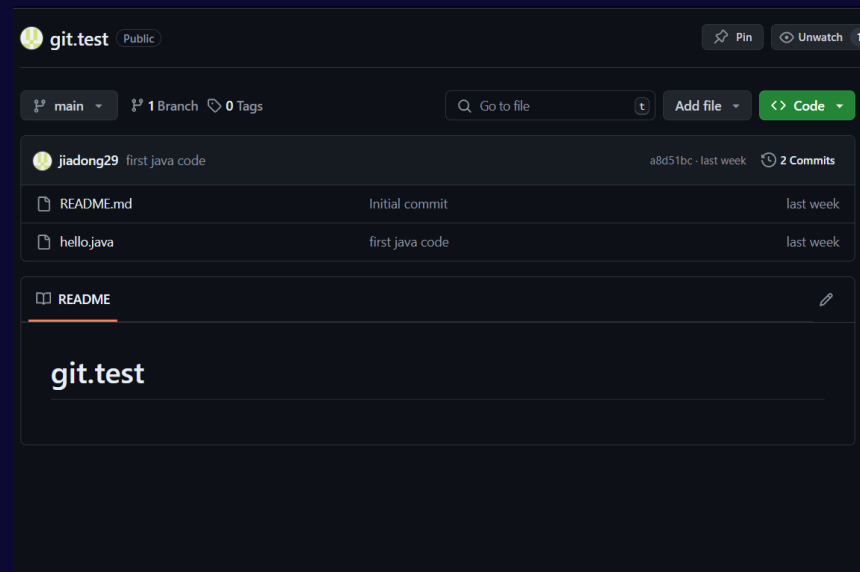
origin: typically the default name given to the remote repository.

main: This indicates the branch you're pushing to on the remote repository.

Command: 'git push origin main'



After git push, Open Github as you can see the folders was success uploaded.





**Thank
You!**