# How to Use the Linux history Command

Introduction

The **`history`** command in Linux is a built-in shell tool that displays a list of commands used in the terminal session. **`history`** allows users to reuse any listed command without retyping it.

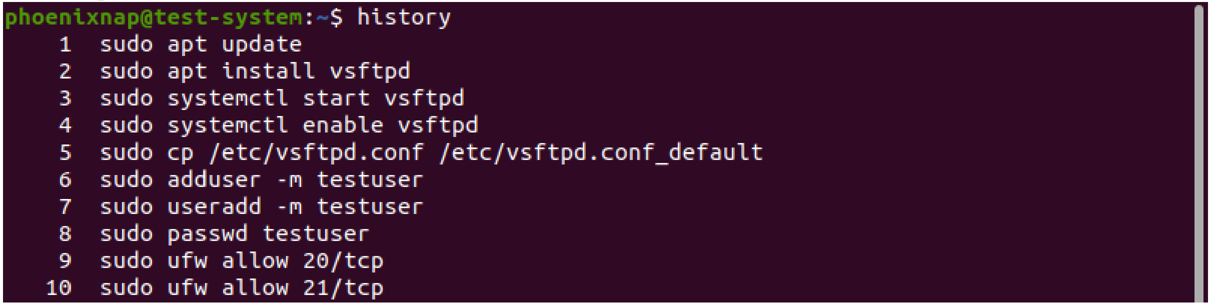
Prerequisites

- A system running Linux.
- An account with sudo privileges.
- Access to the terminal window.

## How to Use Linux history Command

Using the **`history`** command without options displays the list of commands used since the start of the terminal session:

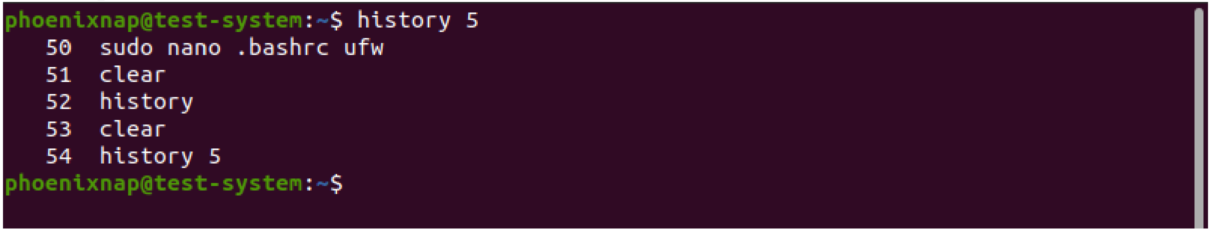
```
history
```



```
phoenixnap@test-system:~$ history
    1  sudo apt update
    2  sudo apt install vsftpd
    3  sudo systemctl start vsftpd
    4  sudo systemctl enable vsftpd
    5  sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
    6  sudo adduser -m testuser
    7  sudo useradd -m testuser
    8  sudo passwd testuser
    9  sudo ufw allow 20/tcp
   10  sudo ufw allow 21/tcp
```

To display the command history list with a limited number of entries, append that number to the **`history`** command. For instance, to show only the latest five entries, use:

```
history 5
```



```
phoenixnap@test-system:~$ history 5
   50  sudo nano .bashrc ufw
   51  clear
   52  history
   53  clear
   54  history 5
phoenixnap@test-system:~$
```

Once you close the terminal, the Bash shell saves new command history entries in the *.bash_history* file.

### Use Date and Timestamps

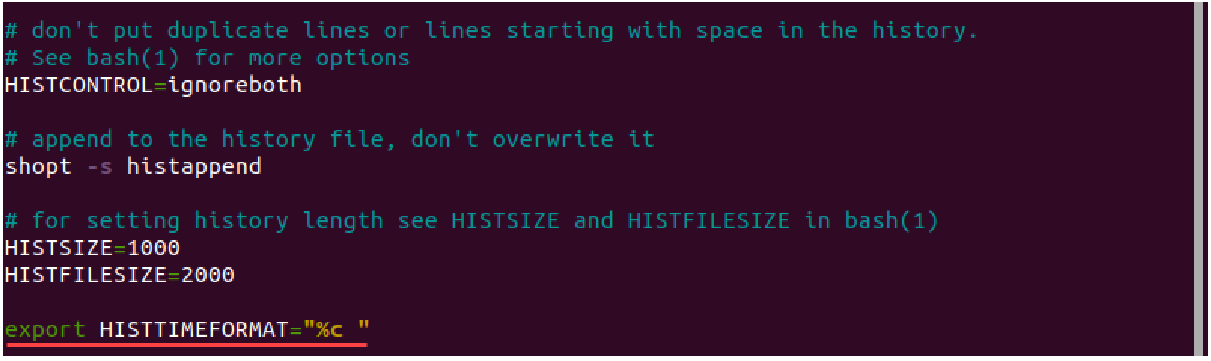
The *.bashrc* file stores the Bash shell settings. Modifying this file allows you to change the output format of the **`history`** command.

Open the *.bashrc* file using a text editor such as Nano:

```
sudo nano .bashrc
```

To change the output format to include date and timestamps, add the following line to the *.bashrc* file:

```
export HISTTIMEFORMAT="%c "
```



```
# 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

export HISTTIMEFORMAT="%c "
```

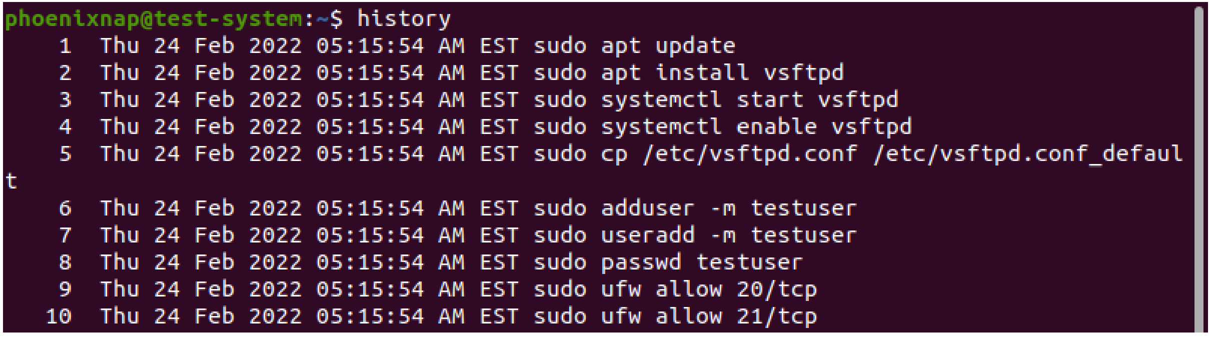
**Note:** The blank space before the closed quotation marks prevents the timestamp from connecting to the command name, making the history list easier to read.

Using different arguments after **`HISTTIMEFORMAT`** allows you to customize the level of detail in the timestamp:

- **`%d`**: Day
- **`%m`**: Month
- **`%y`**: Year
- **`%H`**: Hour
- **`%M`**: Minutes
- **`%S`**: Seconds
- **`%F`**: Full date (Y-M-D format)
- **`%T`**: Time (H:M:S format)
- **`%c`**: Complete date and timestamp (Day-D-M-Y H:M:S format)

Save the changes to the *.bashrc* file, relaunch the terminal, and run the **`history`** command to confirm the new output format:

```
history
```

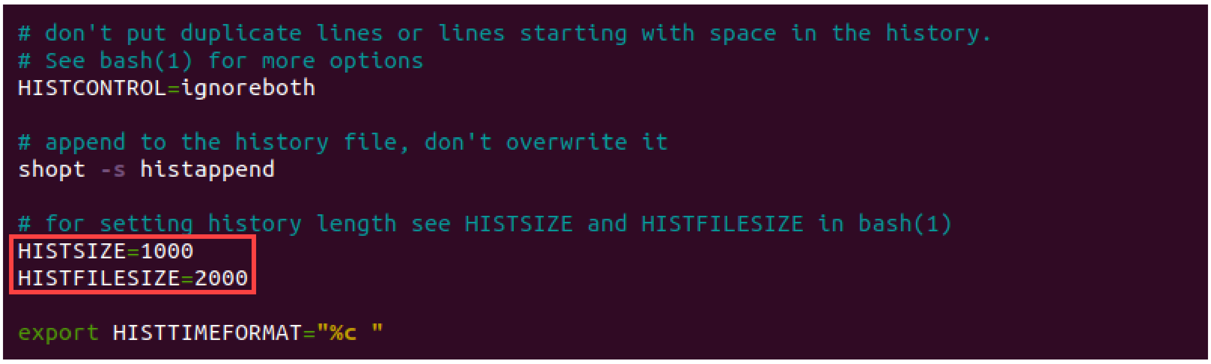


```
phoenixnap@test-system:~$ history
    1  Thu 24 Feb 2022 05:15:54 AM EST sudo apt update
    2  Thu 24 Feb 2022 05:15:54 AM EST sudo apt install vsftpd
    3  Thu 24 Feb 2022 05:15:54 AM EST sudo systemctl start vsftpd
    4  Thu 24 Feb 2022 05:15:54 AM EST sudo systemctl enable vsftpd
    5  Thu 24 Feb 2022 05:15:54 AM EST sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_defaul
t
    6  Thu 24 Feb 2022 05:15:54 AM EST sudo adduser -m testuser
    7  Thu 24 Feb 2022 05:15:54 AM EST sudo useradd -m testuser
    8  Thu 24 Feb 2022 05:15:54 AM EST sudo passwd testuser
    9  Thu 24 Feb 2022 05:15:54 AM EST sudo ufw allow 20/tcp
   10  Thu 24 Feb 2022 05:15:54 AM EST sudo ufw allow 21/tcp
```

## View the Size of the History Buffer

The *.bashrc* file contains two entries that control the size of the history buffer:

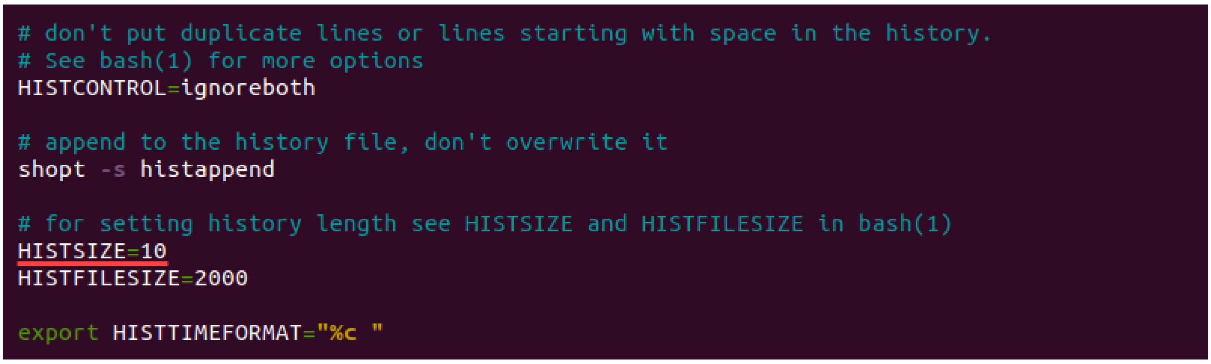
- **`HISTSIZE`**: The maximum number of entries for the history list.
- **`HISTFILESIZE`**: The maximum number of entries in the *.bash_history* file.



```
# 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

export HISTTIMEFORMAT="%c "
```

Editing the **`HISTSIZE`** and **`HISTFILESIZE`** values changes how the Bash shell displays and saves the command history.

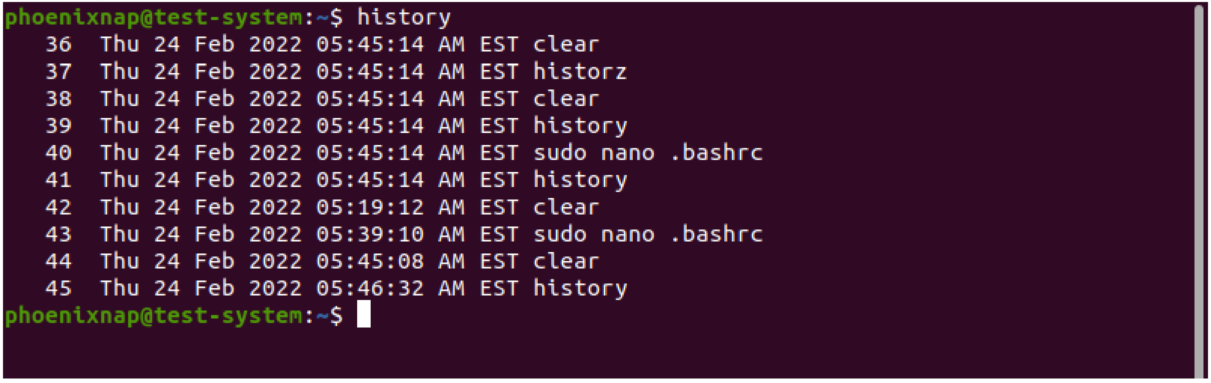
For instance, changing the **`HISTSIZE`** value to **`10`** makes the **`history`** command list show a maximum of 10 latest entries.



```
# 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=10
HISTFILESIZE=2000

export HISTTIMEFORMAT="%c "
```

Saving the changes to the *.bashrc* file, relaunching the terminal, and running the **`history`** command confirms the new output format:

```
history
```

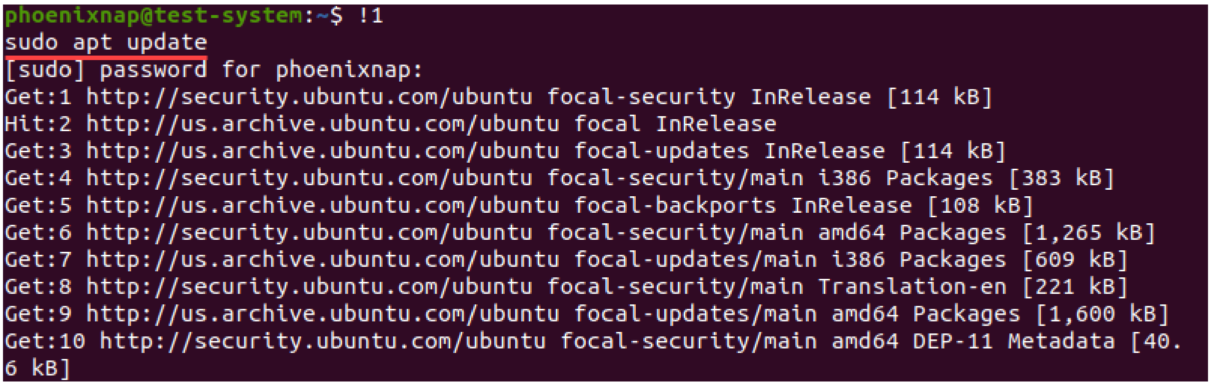


```
phoenixnap@test-system:~$ history
   36  Thu 24 Feb 2022 05:45:14 AM EST clear
   37  Thu 24 Feb 2022 05:45:14 AM EST historz
   38  Thu 24 Feb 2022 05:45:14 AM EST clear
   39  Thu 24 Feb 2022 05:45:14 AM EST history
   40  Thu 24 Feb 2022 05:45:14 AM EST sudo nano .bashrc
   41  Thu 24 Feb 2022 05:45:14 AM EST history
   42  Thu 24 Feb 2022 05:19:12 AM EST clear
   43  Thu 24 Feb 2022 05:39:10 AM EST sudo nano .bashrc
   44  Thu 24 Feb 2022 05:45:08 AM EST clear
   45  Thu 24 Feb 2022 05:46:32 AM EST history
phoenixnap@test-system:~$ 
```

## Repeat a Command

Running the **`history`** command allows you to reuse any of the commands on the list. For instance, to run the first command (**`sudo apt update`**) again, use:

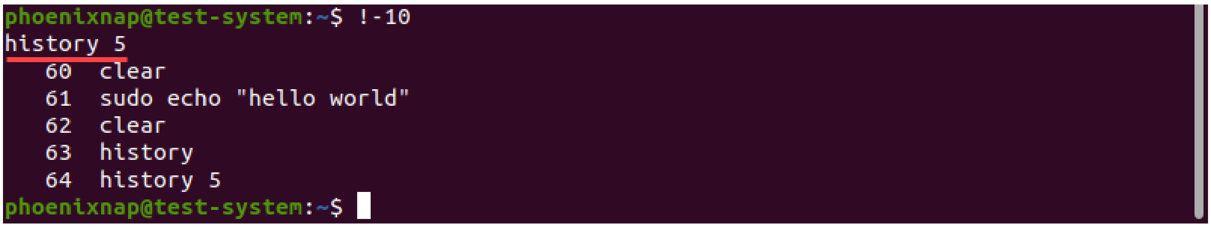
```
!1
```



```
phoenixnap@test-system:~$ !1
sudo apt update
[sudo] password for phoenixnap:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [383 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1,265 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [609 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [221 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,600 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.
6 kB]
```

Adding a dash (**-**) before the command number starts the count from the end of the list. For instance, to reuse the tenth last command (**`history 5`**), use:

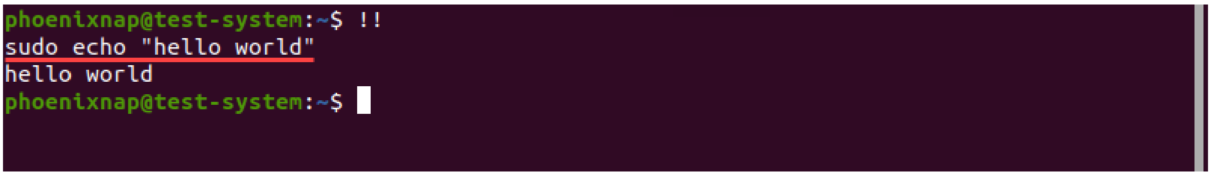
```
!-10
```



```
phoenixnap@test-system:~$ !-10
history 5
   60  clear
   61  sudo echo "hello world"
   62  clear
   63  history
   64  history 5
phoenixnap@test-system:~$
```

Use double exclamation points to repeat the last command:

```
!!
```

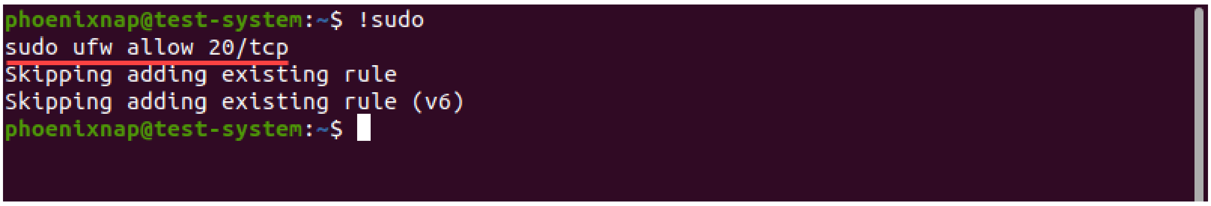


```
phoenixnap@test-system:~$ !!
sudo echo "hello world"
hello world
phoenixnap@test-system:~$
```

## Search a Command by String

Adding a string after the exclamation point runs the latest command that starts with that string. For example, to reuse the latest command that begins with **`sudo`**, use:

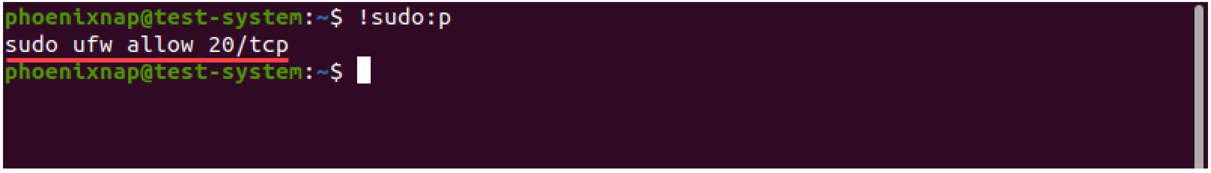
```
!sudo
```



```
phoenixnap@test-system:~$ !sudo
sudo ufw allow 20/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
phoenixnap@test-system:~$
```

Using this method can cause problems if the shell runs an unexpected command, especially when searching for a command that starts with **sudo**. As a precaution, adding the **:p** argument displays the command without running it, allowing you to review the command and decide if you want to execute it.

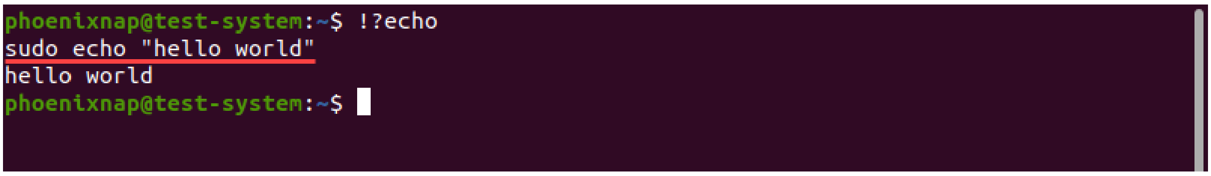
```
!sudo:p
```



```
phoenixnap@test-system:~$ !sudo:p
sudo ufw allow 20/tcp
phoenixnap@test-system:~$
```

To search for a command that contains a string, but may not start with it, add a question mark next to the exclamation point. For instance, to reuse the last command that contains **echo**:

```
!?echo
```



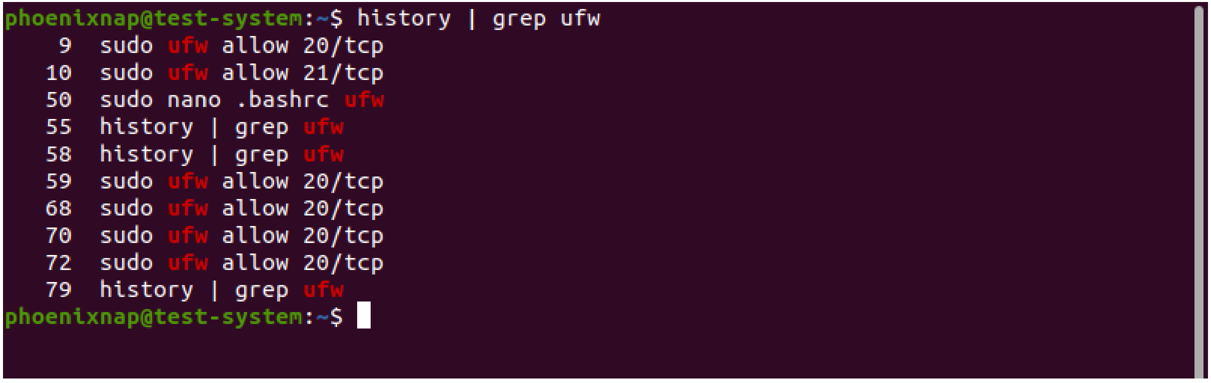
```
phoenixnap@test-system:~$ !?echo
sudo echo "hello world"
hello world
phoenixnap@test-system:~$
```

In the example above, the shell reuses the last command that contains the **echo** string even though the command starts with **sudo**.

### List the Matching Commands

Combining **history** and **grep** allows you to display a list of commands that contain a string. For example, to list all commands that contain **ufw**, use:

```
history | grep ufw
```



```
phoenixnap@test-system:~$ history | grep ufw
    9  sudo ufw allow 20/tcp
   10  sudo ufw allow 21/tcp
   50  sudo nano .bashrc ufw
   55  history | grep ufw
   58  history | grep ufw
   59  sudo ufw allow 20/tcp
   68  sudo ufw allow 20/tcp
   70  sudo ufw allow 20/tcp
   72  sudo ufw allow 20/tcp
   79  history | grep ufw
phoenixnap@test-system:~$
```

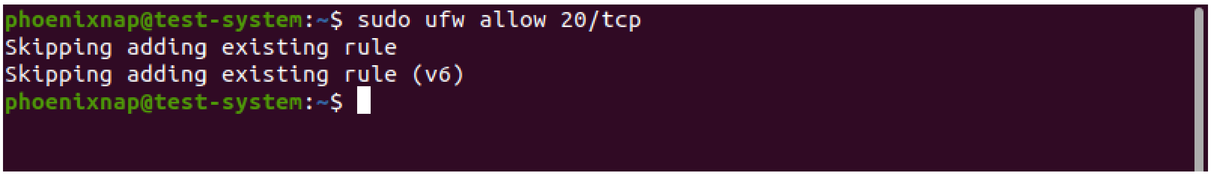
### Change the Executed Command

Use the following syntax to change the last executed command:

```
^[old string]^[new string]^
```

For instance, the **ufw** command to enable port 20 shows that the port is already enabled:

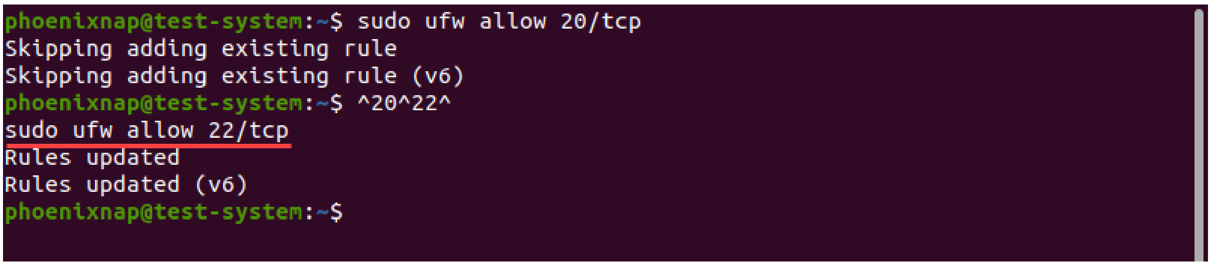
```
sudo ufw allow 20/tcp
```



```
phoenixnap@test-system:~$ sudo ufw allow 20/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
phoenixnap@test-system:~$
```

Use the syntax above to change the port number from 20 to 22:

```
^20^22^
```



```
phoenixnap@test-system:~$ sudo ufw allow 20/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
phoenixnap@test-system:~$ ^20^22^
sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
phoenixnap@test-system:~$
```

## Prevent Recording Commands in History

To prevent recording commands in the history list, temporarily disable recording by using:

```
set +o history
```

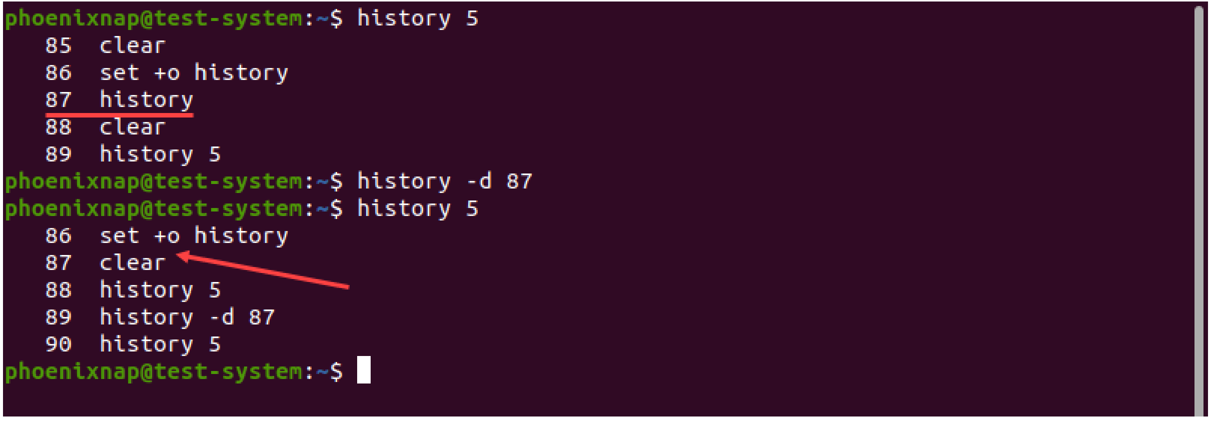
To re-enable recording, use:

```
set -o history
```

## Delete History

Use the `-d` option with the **`history`** command to delete a command from the history list. For instance, delete command number 87 with:

```
history -d 87
```



```
phoenixnap@test-system:~$ history 5
   85  clear
   86  set +o history
   87  history
   88  clear
   89  history 5
phoenixnap@test-system:~$ history -d 87
phoenixnap@test-system:~$ history 5
   86  set +o history
   87  clear
   88  history 5
   89  history -d 87
   90  history 5
phoenixnap@test-system:~$
```

Use the `-c` option to clear the whole history list:

```
history -c
```

## Update the History File

The Bash shell saves any updates to the command history list when you exit the terminal session. The **`history`** command also allows you to save changes while in the terminal session.

Using the **`-a`** option lets you append the command history entries from this session to the *.bash_history* file:

```
history -a
```

Another method is to use the **`-w`** option to save the entire history list to the *.bash_history* file:

```
history -w
```

Below are the lists of options available in the History Command:

| S No | File Name | Description |
|---|---|---|
| 1 | -c | It will clear the history list. It will delete all the entries of the ".bash_history" file. |
| 2 | -d | It will delete the history entry at the position offset. |
| 3 | -a | It will help to append the new history lines to the ".bash_history" file. |
| 4 | -n | It will read the history lines but not already read from the ".bash_history" file into the current history list. |
| 5 | -r | It is helping to read the contents of the ".bash_history" file and use it as the current history of the terminal. |
| 6 | -w | It will write the current history to the ".bash_history" file and help to overwrite the current history file contents. |
| 7 | -s | It will store the arguments in the ".bash_history" file as a single entry. |

Conclusion

After reading this tutorial, you should be able to use the **`history`** command in Linux to view, edit, and delete the command history list and reuse commands from it.

* Linux commands cheat sheet.