**Use of READ for variable pass:**

1. #!/bin/bash
2. *# Demonstrate how read actually works*
3. echo What cars do you like?
4. read car1 car2 car3
5. echo Your first car was: $car1
6. echo Your second car was: $car2
7. echo Your third car was: $car3
8. ./cars.sh
9. What cars do you like?
10. *Jaguar Maserati Bentley*
11. Your first car was: Jaguar
12. Your second car was: Maserati
13. Your third car was: Bentley
14. ./cars.sh
15. What cars do you like?
16. *Jaguar Maserati Bentley Lotus*
17. Your first car was: Jaguar
18. Your second car was: Maserati
19. Your third car was: Bentley Lotus

**Compare two numbers with if else bash**

1. echo "enter two numbers";
2. read a b;
3. echo "a=$a";
4. echo "b=$b";
5. if [ $a \> $b ];
6. then
7. echo "a is greater than b";
8. else
9. echo "b is greater than a";
10. fi;

**Create multiple directory using for loop**

for($counter = 1; $counter <= 9; $counter++)

{

`mkdir -p $counter/{1..9}`; //Executing Unix Command

}

**Everyday backup:**

tar -zcf ~/backups/daily/backup-$(date +%Y%m%d).tar.gz /home/ubuntu/linux\_tests/ 90daysofdevops

find ~/backups/daily/\* -mtime +7 -delete

rsync -a --delete /home/ubuntu/backups/ [ubuntu@172.31.10.200:~/backups](mailto:ubuntu@172.31.10.200:~/backups)

**Users list in etc/passwd**

awk -F':' '{ print $1}' /etc/passwd

# Service command

The **service command** is used to run a [System V init script](https://bash.cyberciti.biz/wiki/index.php?title=System_V_init_script&action=edit&redlink=1). Usually all system V init scripts are stored in [/etc/init.d](https://bash.cyberciti.biz/guide/etc/init.d) directory and service command can be used to start, stop, and restart the [daemons](https://bash.cyberciti.biz/guide/Daemons) and other services under Linux. All scripts in [/etc/init.d](https://bash.cyberciti.biz/guide/etc/init.d) accepts and supports at least the start, stop, and restart commands.

## Syntax

The general syntax is as follows:

service SCRIPT-Name COMMAND

The COMMAND can be at least start, stop, status, and restart.

### The stop command

The syntax is as follows:

service SCRIPT-Name stop

To stop the sshd service, enter:

service sshd stop

### The start command

The syntax is as follows:

service SCRIPT-Name start

To start the sshd service, enter:

service sshd start

### The status command

The syntax is as follows:

service SCRIPT-Name status

To get current status of the sshd service, enter:

service sshd status

**Git commands useful for you:**

**Clone and push to your repo**

1. **Create a new repository at**[**github.com**](https://github.com/)**.** (this is your repository)

* Give it the same name as the other repository.
* Don't initialize it with a README, .gitignore, or license.

1. **Clone the other repository to your local machine.** (if you haven't done so already)

* git clone https://github.com/other-account/other-repository.git

1. **Rename the local repository's current 'origin' to 'upstream'.**

* git remote rename origin upstream

1. **Give the local repository an 'origin' that points to your repository.**

* git remote add origin https://github.com/your-account/your-repository.git

1. **Push the local repository to your repository on github.**

* git push origin main

**create new branch --make changes and push to new branch**

git checkout -b <branch-name>

changes

then git status

git add –all

git push origin <new branch>

for more: https://www.youtube.com/watch?v=uaeKhfhYE0U

**Systemctl vs service command**

**Systemctl have enable and service doesn’t so enable when reboot and disable when reboot.**

**But service simply check service**

**curl offers upload and sending capabilities.** **Wget only offers plain HTTP POST support**. HTTP multipart/form-data sending, which allows users to do HTTP "upload" and in general emulate browsers and do HTTP automation to a wider extent.

##use fish shell for auto suggestion. Just adding this comment for reference of tools

git rm --cached <filePath> **does not unstage** a file, it actually **stages the removal of the file(s)** from the repo (assuming it was already committed before) but leaves the file in your working tree (leaving you with an untracked file).

git reset -- <filePath> will **unstage** any staged changes for the given file(s).

That said, if you used git rm --cached on a new file that is staged, it would basically look like you had just unstaged it since it had never been committed before.