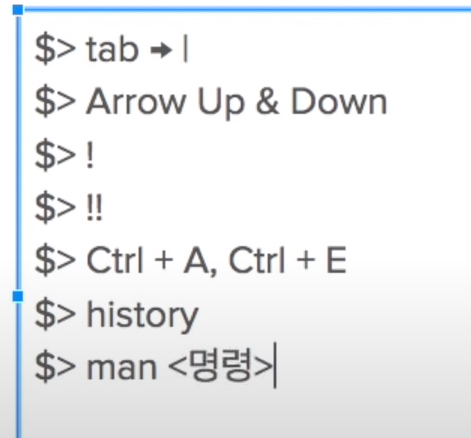
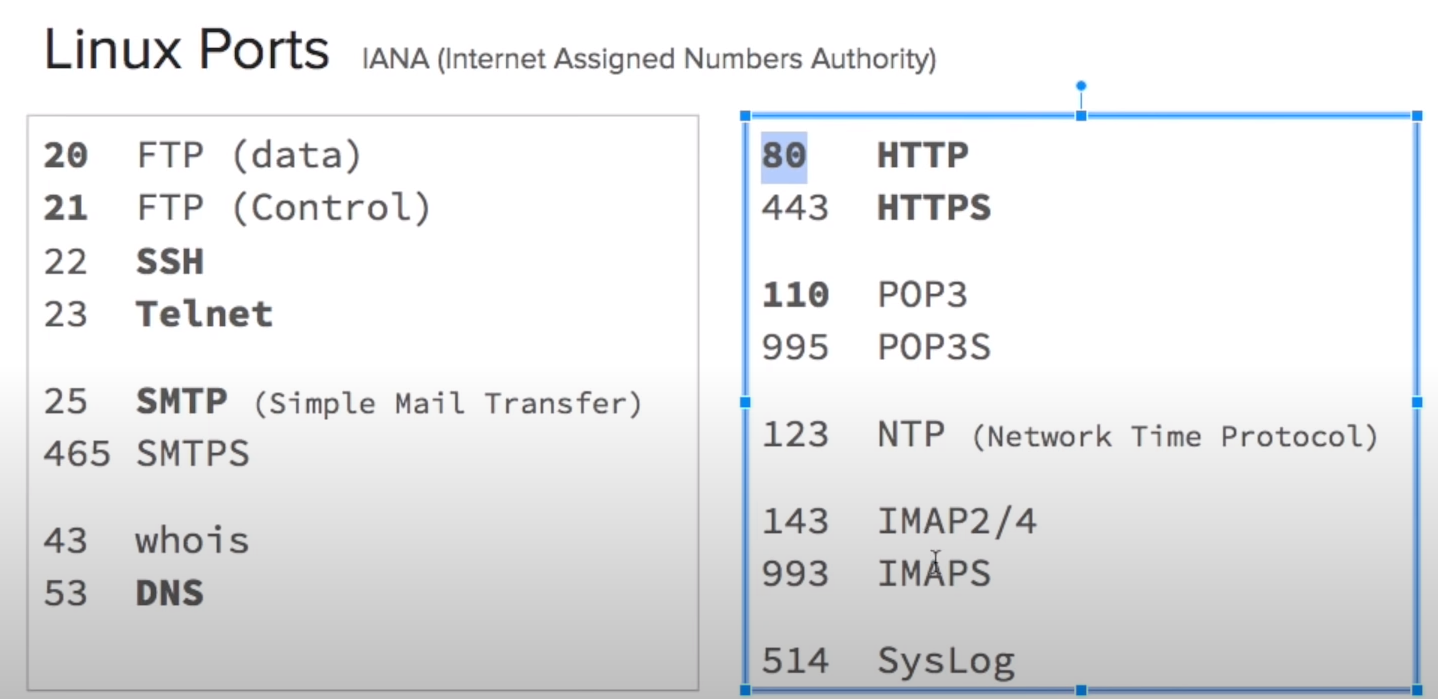
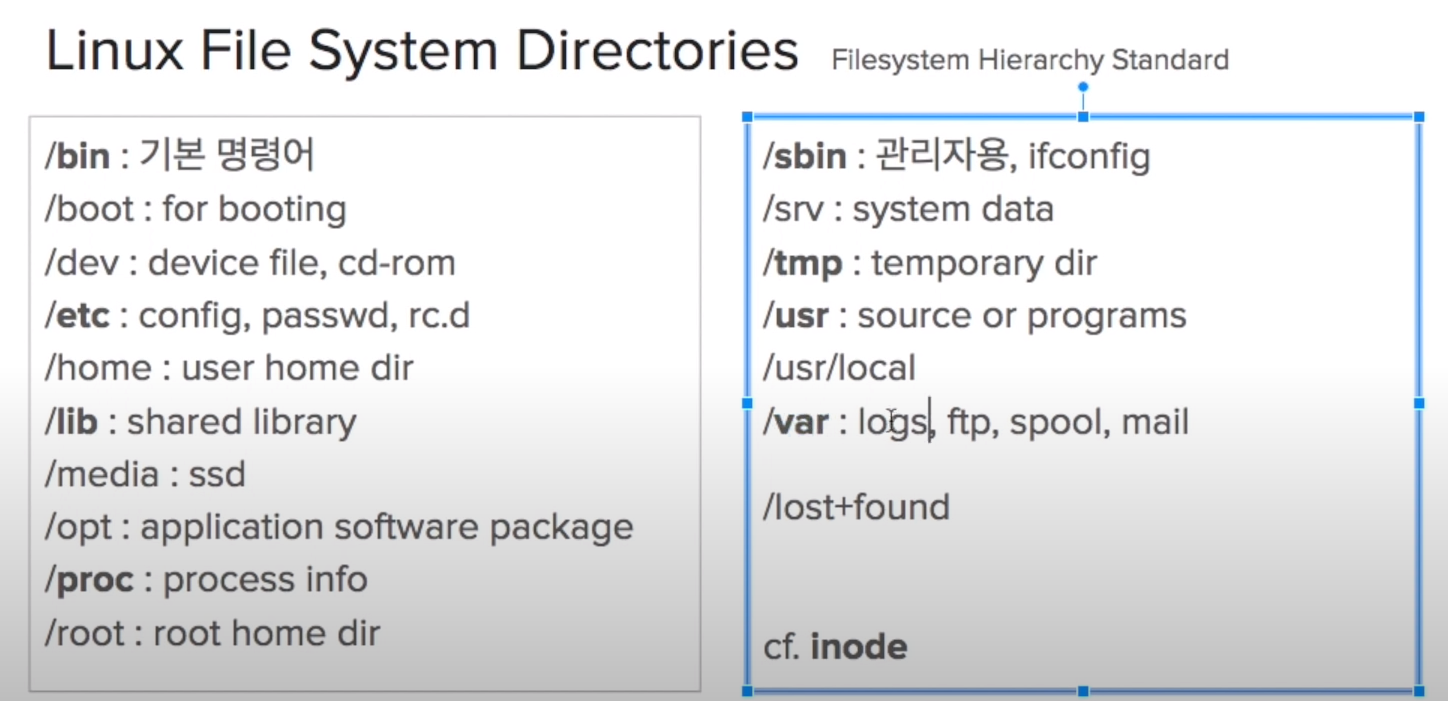
**리눅스**



Tab : 문장 자동완성

History : 실행한 명령어 목록을 보여준다

! : !뒤에 명령어의 일부분을 입력하면 history에 사용된 명령어중에 일치하는 것을 찾아서 실행

!! : 방금전 사용한 명령어 사용 ( arrow up과 동일)

Ctrl + A : 현재 명령문에서 맨 앞으로 커서이동

Ctrl + E : 현재 명령문에서 맨 뒤로 커서이동

Adduser <유저이름> : 파일 생성하고 할 수 있는 유저생성

Ps -ef : Cpu를 쓰고있는 프로세스 목록

Password <유저이름> : 패스워드 새로발급

Deluser <유저이름> : 유저 삭제 (디렉토리는 남아있음)

Rmdir <디렉토리명>:디렉토리 삭제

Rm -rf <디렉토리명> : 강제로 디렉토리 삭제

Su - <유저명> : 해당 유저로 변경

Touch <파일명> :파일생성

Ls -al(=ll): 모든 파일을 한 줄씩 다보여줘(숨긴파일(시스템파일)도 모두)

Ls -cf(=l) : 숨기지않은 파일들만 보여줘

(su) apt-get update : apt-get 은 새로운 프로그램을 install하기 위한 프로그램으로써 최신 버전으로 update 하는 것

(su) apt-get install <이름> : 해당 프로그램 다운

Vim : 에디터다.

Head -5 <파일명> : 파일의 처음부분부터 5번째줄 까지 내용 보여줘

Tail -5 <파일명> : 파일의 마지막에서 5번째줄 까지 내용 보여줘

Which <파일명> : 해당 파일의 위치

Clear : 쉘화면 지우기

Echo “문자열” > <파일명>: 해당 문자열을 파일에 덮어씌우기

Echo “문자열 >> <파일명> : 해당 문자열을 파일에 추가하기

Cd .. : 상위 디렉토리로 이동

Cd - : 전 디렉토리로 이동

Mv <파일명> <위치> : 해당 파일을 해당위치로 이동

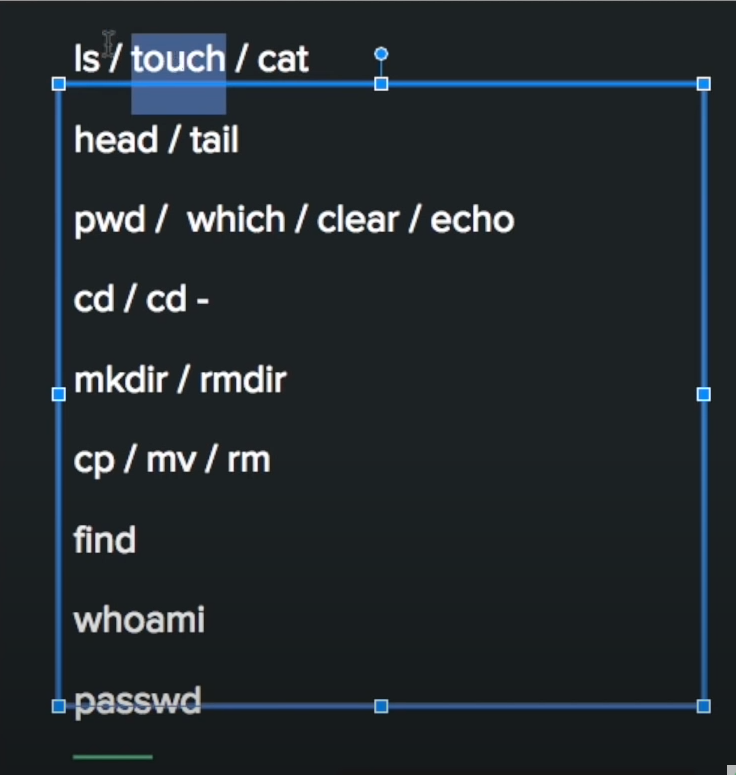
Find . -name <파일명> : 해당 파일의 위치를 찾는다.

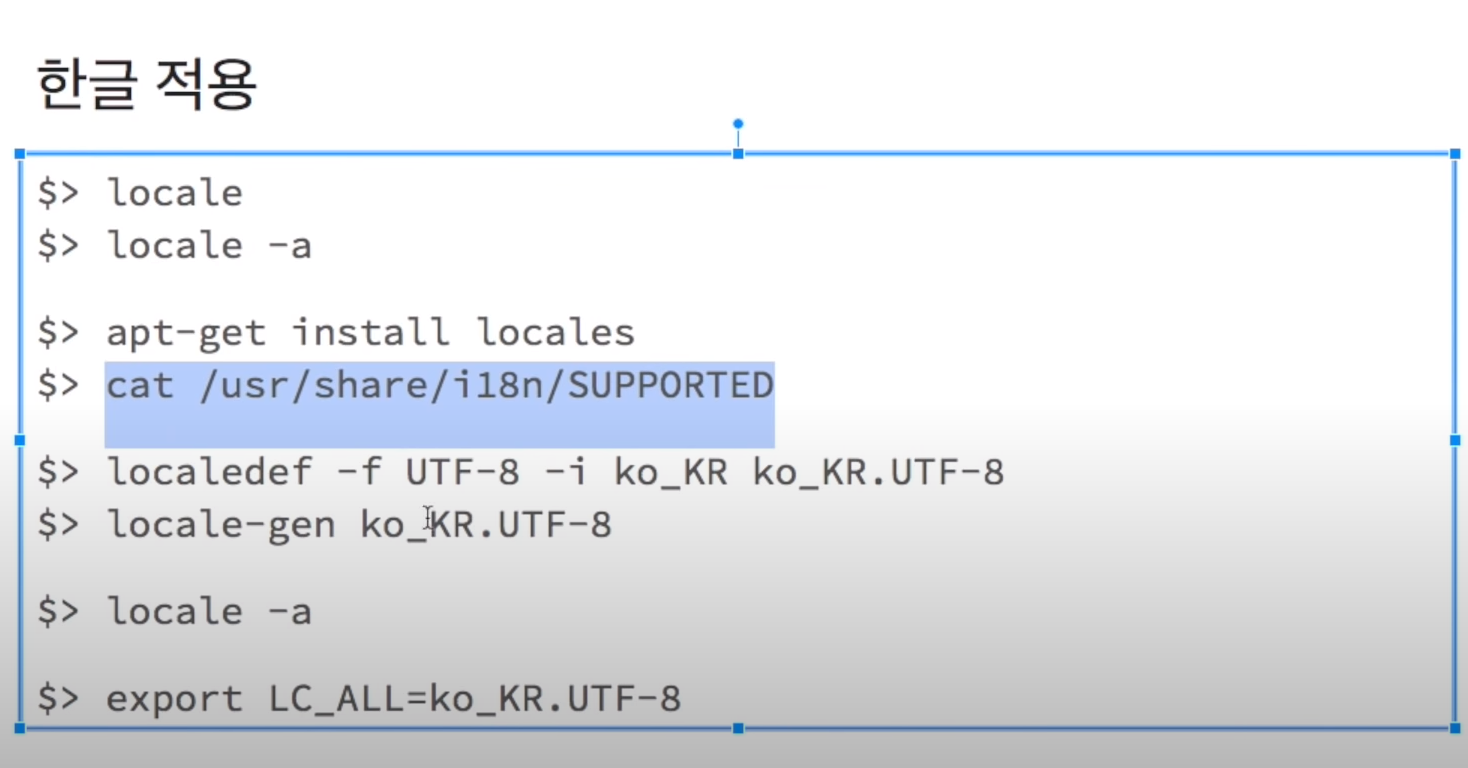
* . : 현재 디렉토리 위치의 하위에 있는 모든 곳에서 찾아라
* -name : 파일명을 지정해주기 위해 사용

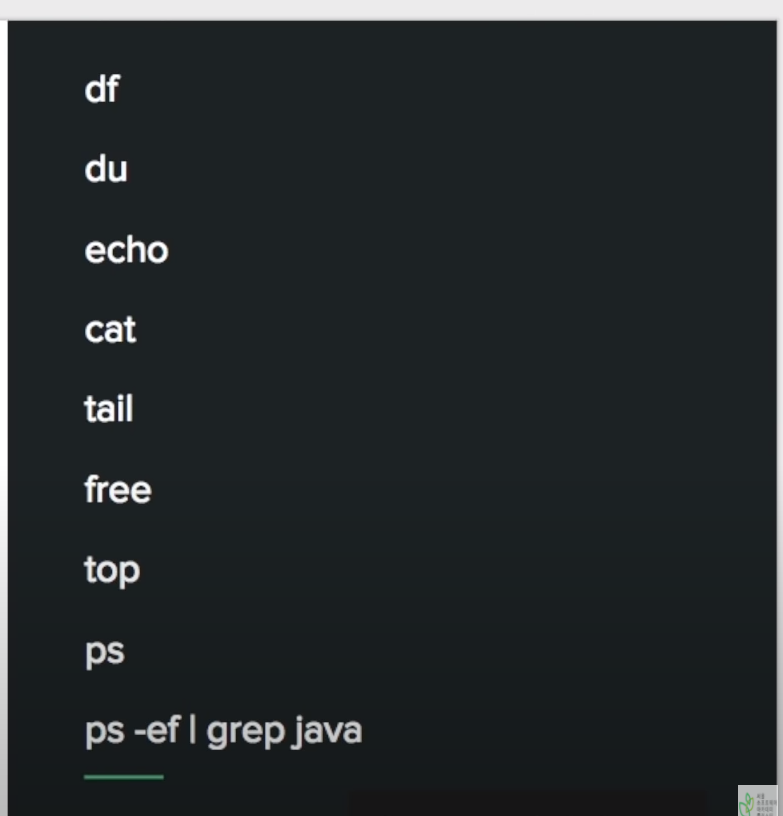


~ : 내 home 디렉토리를 의미한다.

* 따라서 현재 위치는 내 home디렉토리 안에 ttt 디렉토리





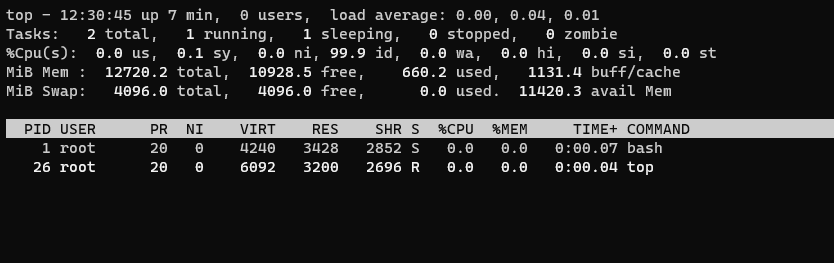


Df : 디스크를 몇 % 사용했는지 확인

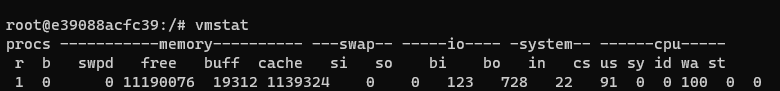
Du <디렉토리> : 해당 디렉토리가 사용하고 있는 량

Free (-m) : 메모리 사용량 -m 메가바이트 단위로

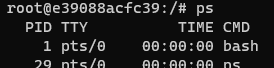
Top : cpu의 사용량을 보여준다.



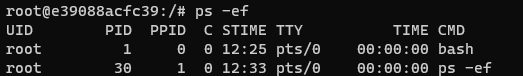
Vmstat : top의 내용을 한줄로 보여준다



Ps : 어떤 프로세스가 있는지 확인



Ps -ef : 좀 더 자세하게



Ps -ef 에는 grep 사용가능



Echo ‘#!/bin/sh’ > tt.sh : tt.sh 파일 안에 ‘#!/bin/sh’을 넣어라

Echo “echo 123” >> tt.sh : 기존의 tt.sh 파일 내옹에 “echo 123”을 추가해라

Cat tt.sh -> 

Sh tt.sh : tt.sh 파일의 명령어를 실행시킴

* Tt.sh 파일 안에있는 echo문이 실행되어서 123 이라는 결과값을 보여줌
* 

Chmod : 권한 변경

Chmod 745 tt.sh : tt.sh 파일의 권한을 rwxr—r-x로 변경해라

Chmod +x tt.sh : tt.sh 파일의 권한에 모두 x(실행)이 가능하도록 권한 부여

Chown <유저이름>:<그룹이름> <파일명> : 파일의 소유자를 변경한다.

Mv : 파일을 이동시킬 때나, 파일명을 변경시킬 때 사용가능

Mv <이전파일명> <새로운파일명>

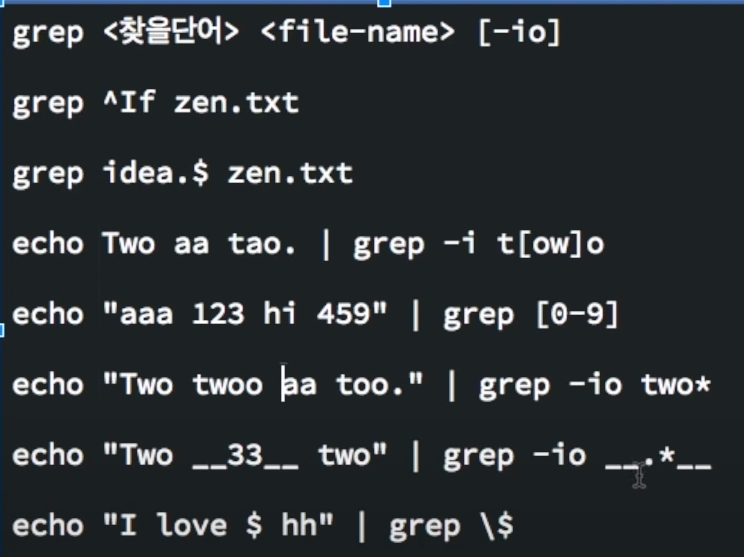
Mv <파일명> <이동위치>

Mv <파일명> <이동위치>/<새로운파일명>

Ln (-s) <목적지> <링크명: (심볼링) 링크



Grem 정규표현식 으로 단어 찾기





Ctrl+f : 페이지 업

Ctrl+u/b : 페이지 다운

W:단어 이동

H,j,k,l : 방향키

Cw: 한단어를 바꿔라

Dw: 한단어를 삭제해라

Shift+A : 라인 제일 끝에서 입력모드로

V : 블록지정(드레그)

Y : 복사

Yy: 라인하나 복사

P : 붙혀넣기

Dd : 한 개 라인 삭제

. :

Colon command

W : 저장

:q: vi 빠져나감

:q! : 현재 작업 저장안하고 나감

:! : 커맨드 라인으로 잠깐 나가기 (명령어 확인하기 위해)

:!(명령어) : 커맨드라인에서 명령을 실행 (엔터누르면 다시 돌아옴)

:wq : 저장하고 나감

Set nu : 라인 번호 보여줘

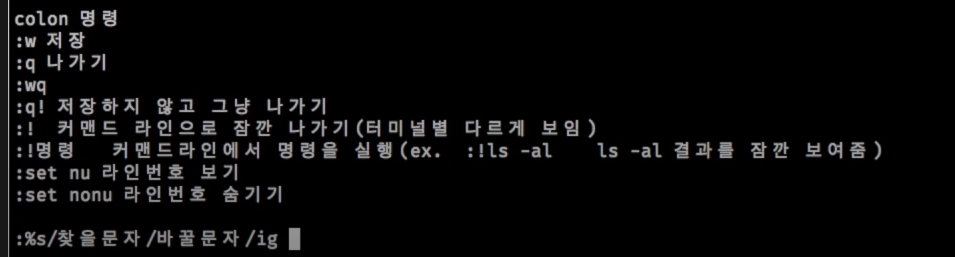
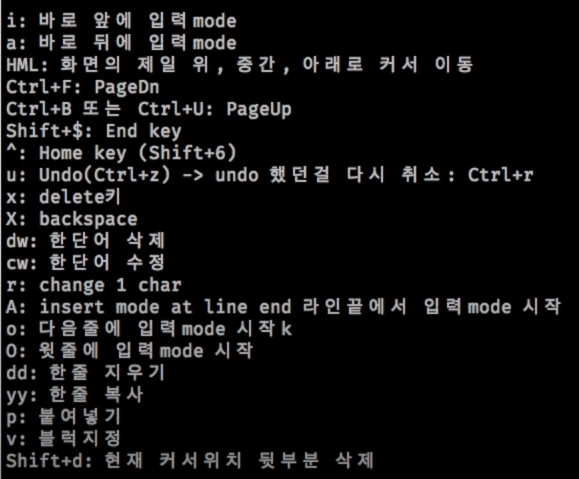
Set nonu : 라인 번호 지워

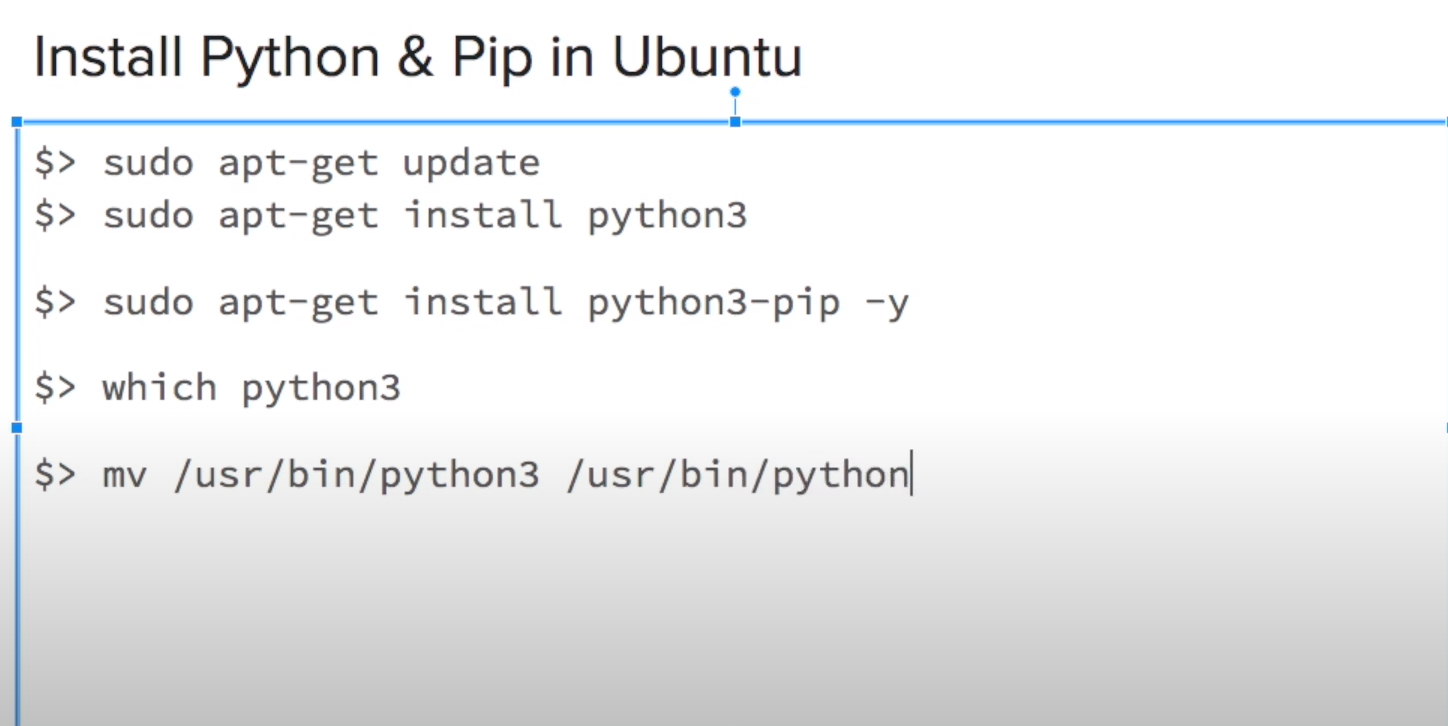
:라인번호 : 라인번호로 커서 이동

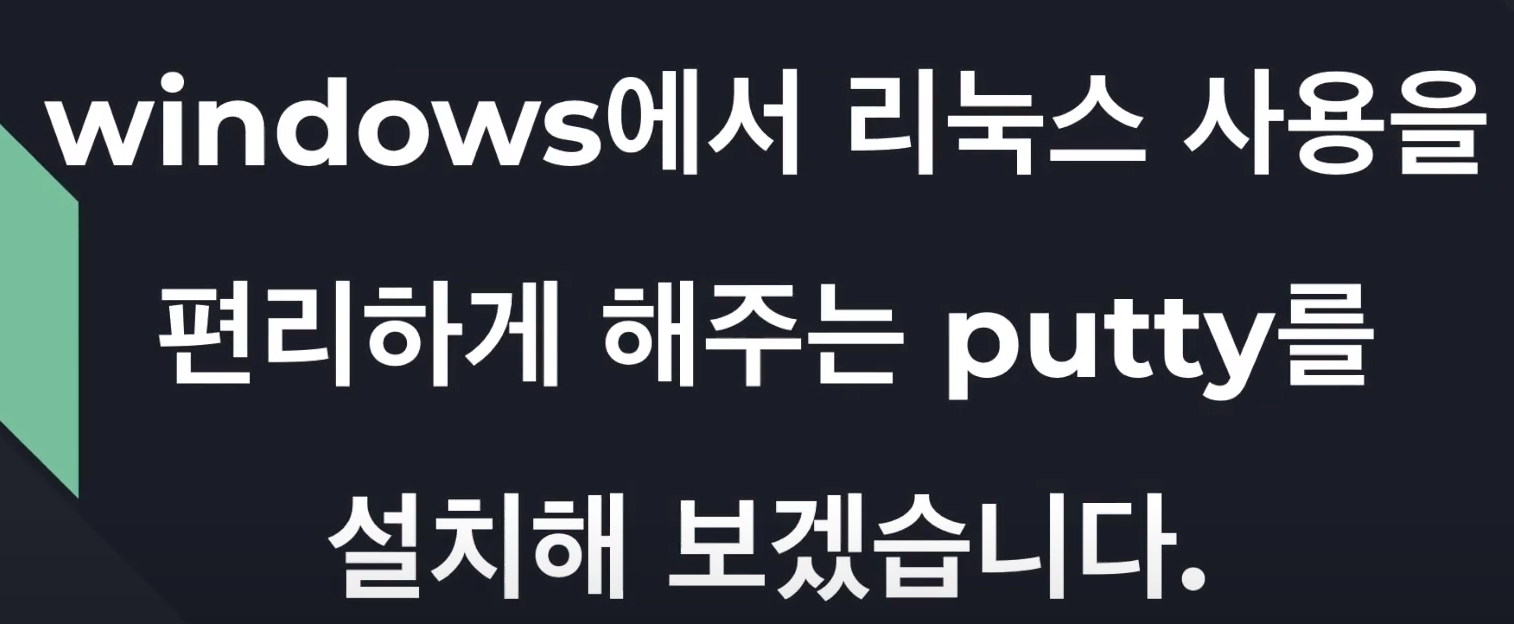
Paste :

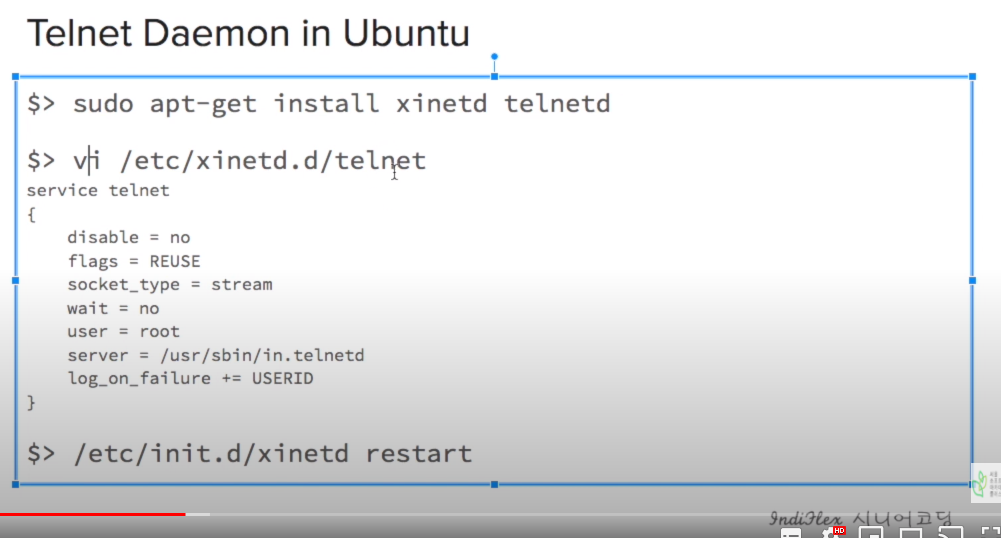
/검색어 : 검색

Vi <파일명> : 파일을 vi에디터로 열어라









/etc/init.d/xinetd restart : 텔넷 사용할 때마다 띄어줘야한다 .

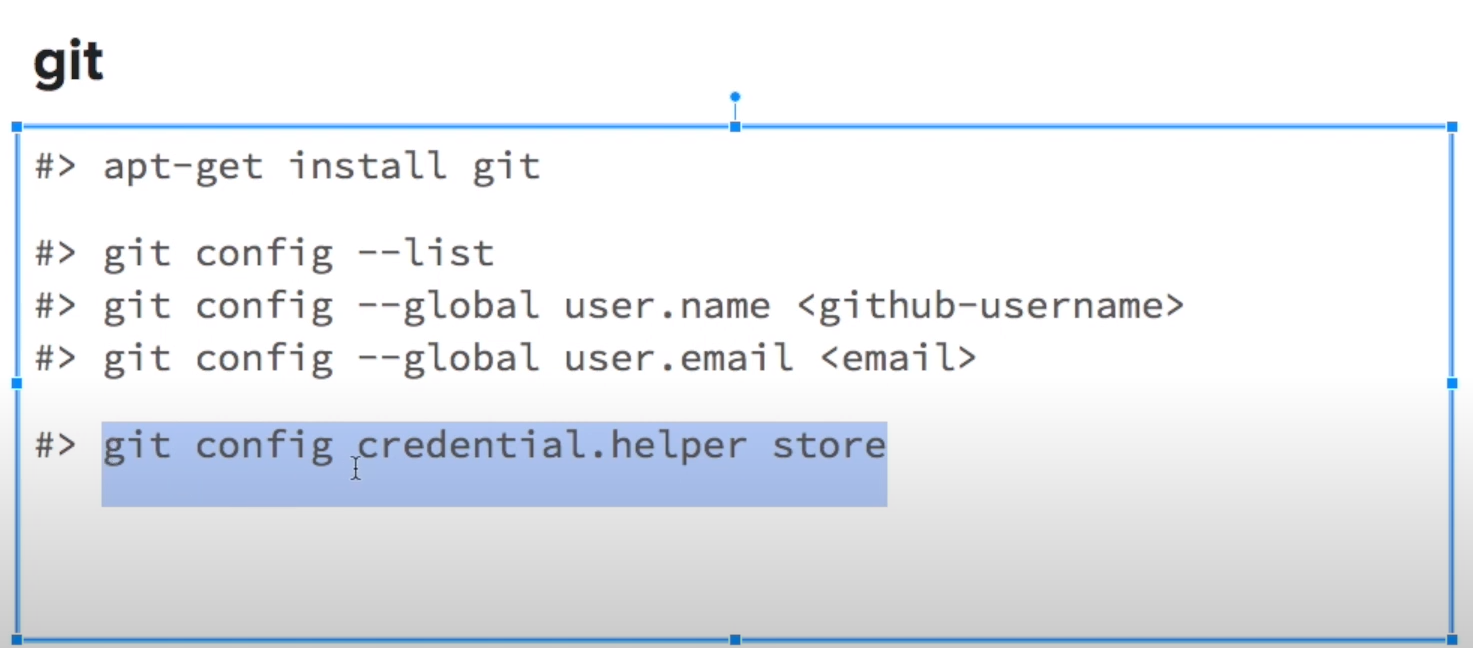
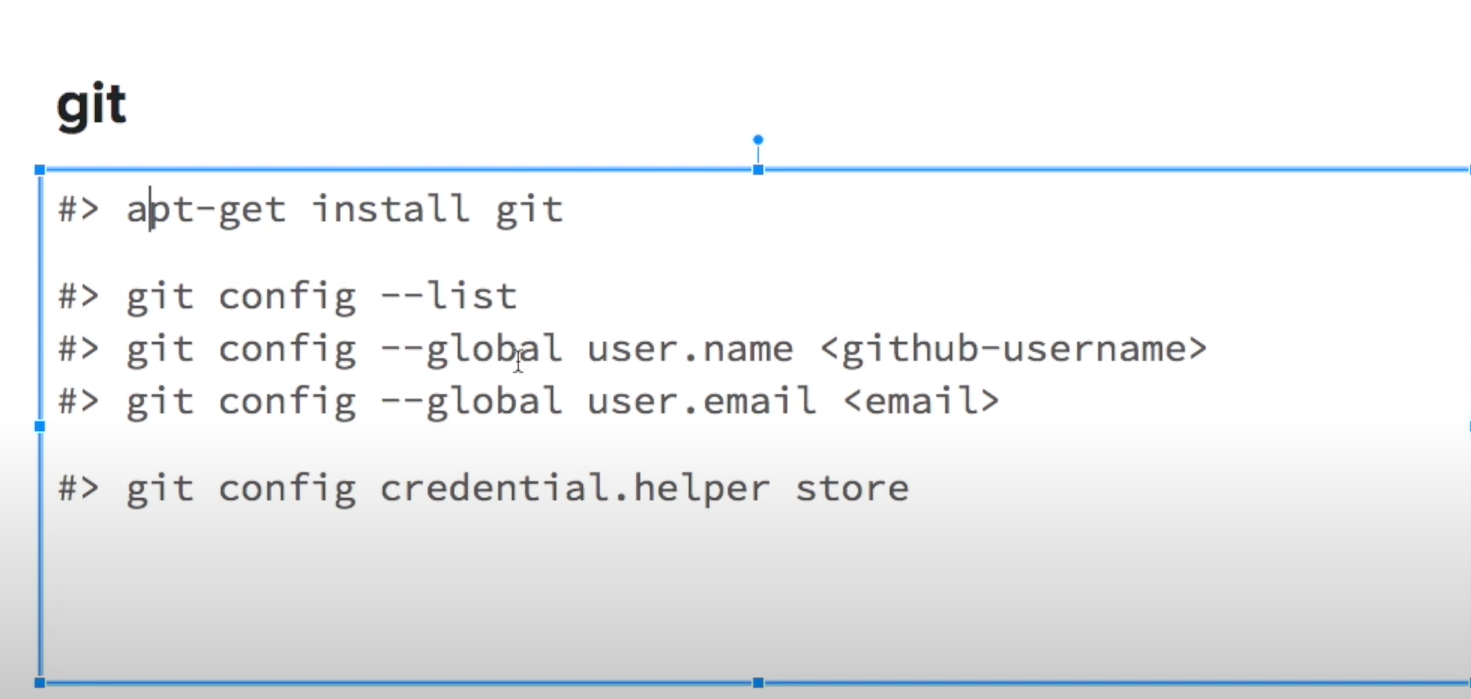


배쉬 쉘을 하나더 씌어서 실행시킨다.

Exit로 빠져나와도 docker ps를 해보면 컨테이너는 살아있음

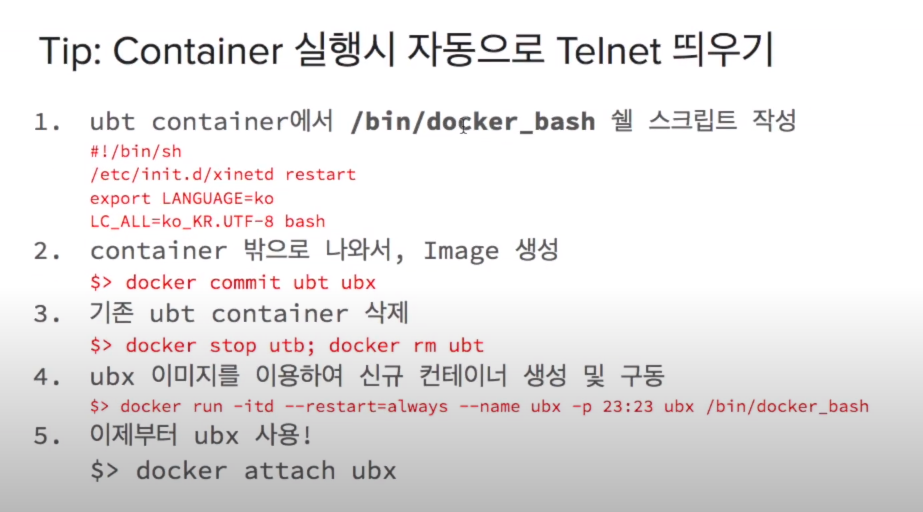
Attach 로 실행한 경우에는 exit로 빠져나오면 컨테이너도 중지됨

Detach (ctrl+p, q)로 빠져나와야 컨테이너도 활성화상태



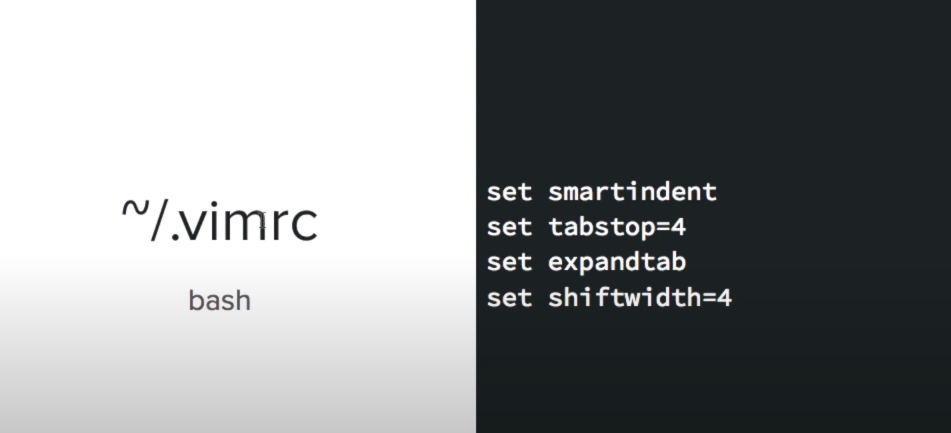
Git config credential.helper store : git과 연결한 정보를 저장해둔다 .

linux에서는 git을하면 계속 인증을 물어보는게 위의 조건을 해주면 기억했다가 자동으로 해줌



--restart=always : 실수로 컨테이너를 빠져나와도 종료되지않고 계속 활성화 상태로 되있게

-> container stop하면 멈춤



Vi 에디터 환경설정

4개 모두 tap에대한 설정이다 코드를 표준화하기 위해서



Alias s1=’~/s1.sh’ : 별명주기 ~/s1.sh를 변수 s1로 대체