

3. Privilege and Permission

Hyunchan, Park

<http://oslab.jbnu.ac.kr>

Division of Computer Science and Engineering

Jeonbuk National University

학습 내용

- J-Cloud 인스턴스 생성
- 관리 체계와 관련 명령어 실습
- 파일의 권한 체계와 관련 명령어 실습



개인 과제 3: 실습 및 c언어 복습

- 실습 과제

- 실습 내용에서 다루는 명령어를 모두 입력하고, 그 결과를 확인할 것
 - 동영상에서 수행한 내용
- 제출 방법
 - Old LMS, 개인 과제 3
 - Xshell 로그 파일 1개 제출
 - 파일 명: 학번.txt

- c언어 복습 과제

- JOTA: hw3-1, hw3-2 문제 수행

- 제출 기한

- 9/28 (월) 23:59 (지각 감점: 5%p / 12H, 1주 이후 제출 불가)

J-Cloud Instance Creation

- 이번 학기 사용할, 나만의 리눅스 머신을 만들어보자



유닉스 시스템을 이용하려면?

- 이미 만들어진 유닉스 시스템에 계정을 발급받아 사용
 - (장) 유닉스 시스템은 다수 사용자가 동시에 시스템에 접근해서 동시에 시스템을 활용할 수 있음
 - (단) 보통 시스템 관리 권한이 없으므로 원하는 환경을 구축하기 어려움
- 새로운 컴퓨터에 유닉스 시스템을 구축
 - (장) 관리 권한을 갖고 원하는 환경을 구축 가능
 - (단) 많은 시간, 비용, 노력이 필요함
- 가상 머신을 사용하는 방법
 - (장) 기존 윈도우 시스템을 활용해 독립된 유닉스 시스템 구축
 - 윈도우 10의 경우, 간단하게 Linux 환경을 구축할 수 있는 도구 제공
 - (단) 설치에 시간이 소요되고, 기존 시스템의 자원을 소요함
- 클라우드를 사용하는 방법!
 - (장) 적은 노력으로 빠르게 독립된 유닉스 시스템을 구축할 수 있음
 - (단) 새로운 시스템을 구축하는 것보다는 저렴하지만, 비용이 소요됨

J-Cloud



JCloud는 전북대학교 컴퓨터공학부에서 제공하는
교육 및 연구용 클라우드 서비스입니다.

전북대 학생들로 구성된 J-DevOps 팀이 OpenStack을 기반으로 구축하였고,
교육 및 연구 용도에 맞춰 필요한 사항을 수정하며 운영합니다.

2018년 2월 현재, 시험 운영 중이며 아래 두 가지 서비스를 제공 중입니다.

(로고를 클릭하면 해당 서비스로 이동합니다.)



이번 학기 사용할, 나만의 리눅스 머신!

- 장점

- Isolated: 나 혼자 사용하니까,
- Full privilege: 내 맘대로,
- Free: 아무거나 다 해도 됨. 거기에 공짜.
- Support: 다른 사람이 원격으로 접속하여 도움을 주기 쉬움
 - 관련 문의 시, 학번, 이름, SSH 접속 Port, 전화번호 꼭 기재할 것!
 - 예) 41983, 박현찬, unix.jbnu.ac.kr:19150, 010-3747-5266

- 단점

- 내 컴퓨터에 설치된 VM이 아니기 때문에 마음대로 다루기가 어렵다
 - 사실 조금만 익숙해지면 해결됨. VM 이기 때문에 불편한 점은 똑같음

사용 방법

- J-Cloud 시스템: <http://jcloud.jbnu.ac.kr>
- 소개 및 사용 방법: <https://jcloud-devops.github.io/>
 - Document 에서 아래 참조
 - 인스턴스 생성 방법
 - SSH 접속은 기존과 동일하게 Xshell 사용
- Unix 수강생들의 J-Cloud 계정은 모두 생성해놓은 상태
 - ID: unix학번, PW: 학번
 - 예) unix41983//41983
- 문의: jcloud@jbnu.ac.kr
- 참고: 학기 종료 직후, 인스턴스 및 계정 삭제됨
 - 컴퓨터공학과 학생들은 ID:학번/PW:이메일계정 으로 계정이 생성되어 있음
 - 타 학과 학생들은 별도 신청하여 사용 가능

수업용 인스턴스(가상머신) 만들기

- Details
 - Instance Name: 학번 (반드시 본인 학번 그대로 이용할 것. 채점과 관련있음)
 - Count: 1 (1인당 1대만 만들어서 사용할 것)
 - Availability Zone: “nova”
- Source: Boot source = Image, 항목 중 “Unix2020” 선택
 - Create New Volume: “No”
- Flavor: 2020unix.flavor
- Network: 2020unix.internal
- Keypair: 방금 만든 키페어 선택
- 그 외 기본 설정 사용

Keypair 만들기

- 미리 생성해서 꼭 다운로드 받고, 파일 잘 관리할 것
 - J-Cloud 상의 가상머신에 접속할 때 사용하는 인증파일

JCloud 2020unix(203.254.143.135) admin

Project / Compute / Key Pairs

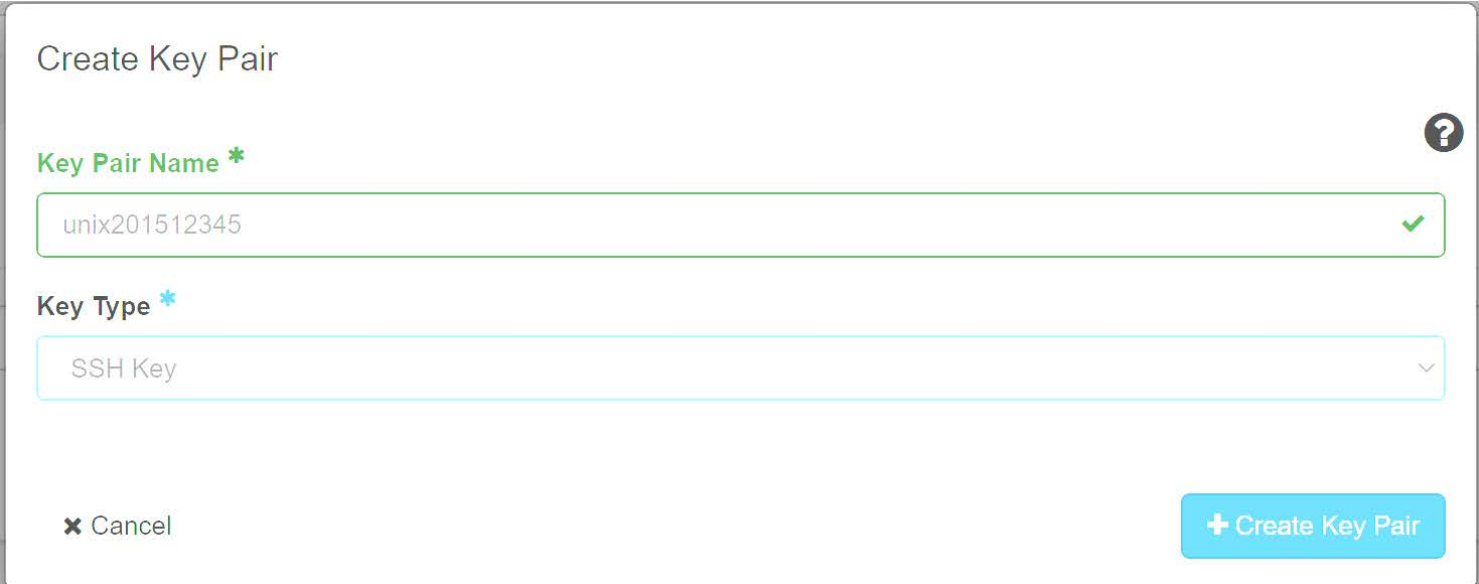
Key Pairs

Displaying 7 items

Name	Type	Action
hcpark-temp	ssh	Delete Key Pair
hcpark-ussuri	ssh	Delete Key Pair
khj_test	ssh	Delete Key Pair
test	ssh	Delete Key Pair
test_key	ssh	Delete Key Pair
testkey	ssh	Delete Key Pair

Keypair 만들기

- 키페어 이름: unix학번
- 타입: SSH key
- Create Key Pair 클릭하면, .pem 파일 자동 다운로드 됨
- 인스턴스 생성 이후, 접속 할 때 x-Shell 에 등록하여 사용함



Create Key Pair

Key Pair Name *

unix201512345 ✓

Key Type *

SSH Key

✕ Cancel + Create Key Pair

J-Cloud 에서 인스턴스 생성

- 인스턴스 생성 후 화면

- 본인의 인스턴스에 대해 IP 확인하고, SSH 접속하여 다음 내용들을 진행할 것

The screenshot shows the JCloud web interface. The top navigation bar includes the JCloud logo, a project selector (2020unix(203.254.143.135)), and a user profile (admin). The left sidebar contains navigation links: Project, API Access, Compute, Overview, Instances (highlighted), Images, Key Pairs, Server Groups, Volumes, Network, Admin, and Identity. The main content area is titled 'Instances' and shows a table of running instances. Two instances are listed: 'test2.hcpark' and 'couch.hcpark.test'. The first instance is highlighted with a red box. The table columns are: Instance Name, Image Name, IP Address, Flavor, Key Pair, Status, Availability Zone, Task, Power State, Age, and Actions.

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
test2.hcpark	unix2020	10.0.0.55	2020unix.flavor	test	Active	nova	None	Running	2일, 4시간	Create Snapshot
couch.hcpark.test	Couch.v1	10.0.0.150	jdevops.flavor	hcpark-ussuri	Active	nova	None	Running	1주, 4일	Create Snapshot

SSH 클라이언트를 이용한 접근

• <https://jcloud-devops.github.io/user-guide.html>

매우 중요!!!

- SSH 연결 방법을 확인하고, 본 수업에서는 아래 공용 IP 사용함
- 공용 IP: 203.254.143.135 (unix.jbnu.ac.kr)
- SSH 포트 번호: 19xxx
 - xxx는 본인 인스턴스 IP 의 끝자리. 예) 10.0.0.3 → 19003, 10.0.0.199 → 19199
- 접속 User ID: ubuntu
 - Password는 없음. 키페어가 패스워드 역할을 함

• 예시) 본인 instance IP 가 J-Cloud에서 아래와 같이 10.0.0.55 인 경우

<input type="checkbox"/>	Instance Name	Image Name	IP Address
<input type="checkbox"/>	test2.hcpark	unix2020-ubuntu20.04-code-server-couch-ssh7777	10.0.0.55

연결

사용자 인증

로그인 프롬프트

로그인 스크립트

SSH

보안

터널링

SFTP

TELNET

RLOGIN

SERIAL

프록시

연결

일반

이름(N): 0. 2020unix.test

프로토콜(P): SSH

호스트(H): unix.jbnu.ac.kr

포트 번호(O): 19055

연결

사용자 인증

로그인 프롬프트

로그인 스크립트

SSH

보안

터널링

SFTP

TELNET

RLOGIN

SERIAL

프록시

연결 유지

터미널

키보드

VT 모드

고급

모양

연결 > 사용자 인증

인증 방법과 기타 관련 매개 변수들을 선택하십시오.

이 섹션은 로그인 할 때 시간을 절약하기 위해 사용할 수 있습니다. 그러나 보안을 중요시하는 경우 이 섹션을 비워 두는 것이 좋습니다.

방법(M): Public Key

설정(S)...

사용자 이름(U): ubuntu

암호(P):

사용자 키(K): hcpark-ussuri

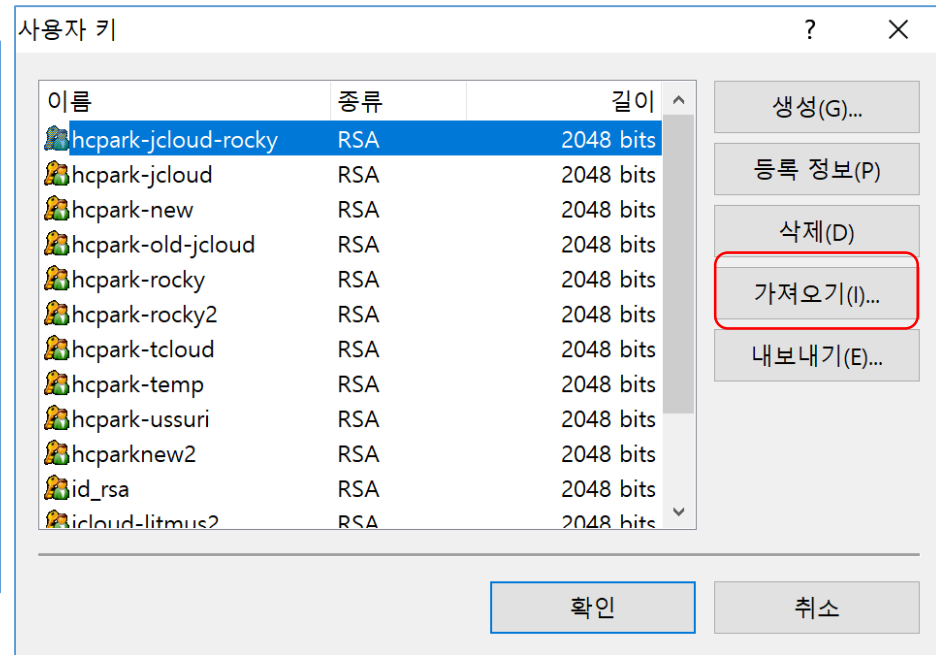
찾아보기(B)...

암호(A):

사용자 키:
본인 Keypair 선택

X-Shell 키페어 등록

- 세션 등록 정보->연결->사용자 인증->사용자 키 찾아보기
- 가져오기->다운로드 받은 .pem 파일 선택->목록에 추가되면 선택



J-Cloud 관련 팁

- 크롬 브라우저를 사용할 것
 - 특히 인터넷 익스플로러는...좀...이제 그만 좀...
- 도움말 및 비밀번호 변경 방법
 - 오른쪽 위 본인 계정 버튼을 클릭하면,
 - 도움말: Help 클릭 -> JCloud Documentation page 로 이동
 - 비밀번호 변경: Settings 클릭 -> (화면 전환 후 왼쪽 메뉴에서) change password
- 파일을 주고 받으려면? (X-Shell 이용 시)
 - 파일 업로드: Drag & drop
 - 파일 다운로드
 - `$ sz sample.txt`
 - 이후 다운로드 받을 폴더 선택
 - 참고: 관련 패키지는 lrzsz

Privilege



시스템 관리자

- 시스템 관리자의 역할
 - 사용자, 그룹 등록 및 삭제
 - 소프트웨어 설치, 업그레이드, 삭제
 - 하드웨어 추가 설치
 - 시스템 보안
 - 데이터 백업
- Windows의 관리자 계정: ADMINistrator
- Unix: Super user
- 다른 용어: Privileged user or account
 - 반대말: non-privileged



시스템 관리자

- 슈퍼유저(superuser)

- 시스템을 관리할 수 있는 사용자
- “root”: 슈퍼유저가 사용하는 계정
- Root 로그인 시, Shell 의 모양이 \$ 가 아닌 # 으로 표시됨
 - 명령어 사용법을 검색했을 때, 앞이 # 으로 표시되어있으면 관리 권한으로 수행해야 한다는 뜻

- 슈퍼유저 로그인 방법

- 직접 root 계정으로 로그인
- 다른 계정으로 로그인 후
 - su 명령 사용, root 패스워드 입력
 - sudo -s 명령 사용, 본인 패스워드 입력
 - (다시 돌아오려면 exit)

```
ubuntu@test2:~$ su
Password:
root@test2:/home/ubuntu# exit
exit
ubuntu@test2:~$ sudo -s
root@test2:/home/ubuntu#
```

- (참고) 클라우드 인스턴스는 보통 기본 계정에 관리 권한이 부여되어 있음

사용자 계정 추가 명령어

- 사용자 계정 추가

```
# useradd [옵션] 사용자명
```

새로운 사용자 계정을 생성한다.

참고: # adduser [옵션] 사용자명

새로운 사용자 정보 입력이 보다 편리하고, 사용자 환경을 설정함

- 패스워드 설정

- # passwd 사용자명

- 관련 파일: /etc/passwd, /etc/shadow

- 사용자 계정 삭제

```
# userdel [-r] 사용자명
```

사용자 계정을 삭제한다.

-r 옵션을 사용하면 사용자의 홈 디렉터리도 삭제한다.



예제

```
root@unix:/home/ubuntu# useradd newuser1
root@unix:/home/ubuntu# ls -al /home/newuser1
ls: cannot access '/home/newuser1': No such file or directory
root@unix:/home/ubuntu# adduser newuser2
Adding user `newuser2' ...
Adding new group `newuser2' (1108) ...
Adding new user `newuser2' (1107) with group `newuser2' ...
Creating home directory `/home/newuser2' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for newuser2
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
root@unix:/home/ubuntu# userdel newuser1
root@unix:/home/ubuntu# ls /home/newuser2
root@unix:/home/ubuntu# ls -al /home/newuser2
total 20
drwxr-xr-x  2 newuser2 newuser2 4096 Sep 14 13:55 .
drwxr-xr-x 109 root      root      4096 Sep 14 13:55 ..
-rw-r--r--  1 newuser2 newuser2  220 Sep 14 13:55 .bash_logout
-rw-r--r--  1 newuser2 newuser2 3771 Sep 14 13:55 .bashrc
-rw-r--r--  1 newuser2 newuser2  807 Sep 14 13:55 .profile
root@unix:/home/ubuntu#
```



(참고) 자동완성 기능과 입력 중단

- Tab key
 - 경로명, 커맨드 등을 입력할 때, tab 을 입력하면 자동 완성 수행
 - 혹은 현재 입력 상태에서 추가로 입력 가능한 옵션들을 보여줌
- 현재 입력하던 내용을 중단하고 싶다면?
 - CTRL+C (^C)
- 예제)
 - “vi /etc/p” 까지만 입력한 후, tab을 입력하면
 - /etc 내의 p 로 시작하는 파일을 모두 보여줌. 그리고 입력하던 명령어는 그대로 유지됨
 - “a” 를 추가로 입력한 후, tab을 입력하면
 - /etc 내의 pa 로 시작하는 파일을 모두 보여줌
 - 이후 s 를 입력하고, ctrl+c 를 입력하여 입력을 중단함

```
ubuntu@unix:~$ vi /etc/p
pam.conf          passwd-          pm/              popularity-contest.conf  protocols
pam.d/            perl/           polkit-1/        profile           python3/
passwd           pki/            pollinate/       profile.d/        python3.8/
ubuntu@unix:~$ vi /etc/pa
pam.conf pam.d/  passwd  passwd-
ubuntu@unix:~$ vi /etc/pas^C
ubuntu@unix:~$
```



사용자와 그룹

- Group
 - 여러 사용자를 하나의 그룹으로 묶어 관리할 수 있음
 - 특히 권한 공유를 위해 주로 사용함
 - 예) /var/temp 라는 디렉토리를 여러 학생이 함께 사용해야할 경우, 각 사용자 권한을 각각 수정하기보다, 같은 그룹에 소속시킨 다음 해당 그룹에 수정 권한을 부여함
 - 한 사용자는 최소 하나의 그룹에는 속해야 함
 - 이를 해당 사용자의 primary group 이라고 함
 - 한 사용자는 여러 그룹에 속할 수 있음
- 기본 사용자 그룹
 - 새로운 ID를 만들 경우, 해당 ID와 동일한 이름으로 새로운 그룹이 생성되고,
 - 자동으로 새로운 사용자가 해당 그룹에 속하며, primary group 으로 지정됨
- 사용자와 그룹은 unique ID and number 로 관리됨
 - /etc/passwd 에서 확인
 - 그룹 정보는 /etc/group

vi /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/s
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd:/
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd:/usr/sbin/r
systemd-timesync:x:102:104:systemd Time Synchronization,,:/run/system
messagebus:x:103:106:./nonexistent:/usr/sbin/nologin
syslog:x:104:110:./home/syslog:/usr/sbin/nologin
_apt:x:105:65534:./nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uidd:x:107:112:./run/uidd:/usr/sbin/nologin
tcpdump:x:108:113:./nonexistent:/usr/sbin/nologin
sshd:x:109:65534:./run/sshd:/usr/sbin/nologin
landscape:x:110:115:./var/lib/landscape:/usr/sbin/nologin
pollinate:x:111:1:./var/cache/pollinate:/bin/false
systemd-coredump:x:999:999:systemd Core Dumper:./usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:998:100:./var/snap/lxd/common/lxd:/bin/false
unix201512345:x:1001:1001:././home/unix201512345:/bin/bash
unix41983:x:1002:1002:./home/unix41983:/bin/bash
mytest:x:1003:1003:./home/mytest:/bin/bash
unix201918284:x:1004:1004:./home/unix201918284:/bin/bash
unix201711081:x:1005:1005:./home/unix201711081:/bin/bash
unix201514804:x:1006:1006:./home/unix201514804:/bin/bash
unix201614912:x:1007:1007:./home/unix201614912:/bin/bash
unix201846140:x:1008:1008:./home/unix201846140:/bin/bash
```

root:x:0:0:root:/root:/bin/bash

① ②③④ ⑤ ⑥ ⑦

① 필드 1 : 사용자명

② 필드 2 : 패스워드(/etc/shadow 파일에 암호화되어 있음)

③ 필드 3 : 사용자 계정 uid

④ 필드 4 : 사용자 계정 gid

⑤ 필드 5 : 사용자 계정 이름(정보)

⑥ 필드 6 : 사용자 계정 홈 디렉토리

⑦ 필드 7 : 사용자 계정 로그인 셸

* <https://webdir.tistory.com/129>



그룹 관리 명령어

- 그룹 추가

```
# groupadd [옵션] 그룹명
```

새로운 그룹을 생성한다.

- 그룹 삭제

```
# groupdel [-r] 그룹명
```

그룹을 삭제한다.

해당 그룹이 특정 사용자의 primary group 인 경우, 삭제 불가

- 현재 소속 그룹 확인

```
# groups [사용자명]
```

사용자의 소속 그룹을 확인한다.

뒤에 사용자명이 없을 경우, 현재 로그인한 사용자 정보 출력



그룹 관리 명령어

- 사용자에게 소속 그룹 추가

```
# usermod -G 그룹명 사용자명
```

사용자에게 새로운 소속 그룹을 지정한다.

* 그룹명에 comma (,)를 사용해 여러 그룹을 지정할 수 있음. 공백 문자 사용하면 안됨.

* 만약, 현재 소속된 그룹명이 포함되어 있지 않다면, 해당 그룹에서는 제거됨.

(다만, primary group 에서는 제거되지 않음)

(옵션)

-a : -G 옵션과 함께 사용하며, 새로운 그룹을 append 하기만 함. (-aG 로 수행)

-g 그룹명: 해당 그룹으로 소속을 시키며, 동시에 primary group 으로 지정한다.

(참고: usermod 는 본래 사용자의 다양한 정보를 수정하는 명령:

홈 디렉토리, 로그인 시 사용할 shell, UID 등)



예제

```
root@unix:/home/ubuntu# useradd user1
root@unix:/home/ubuntu# useradd user2
root@unix:/home/ubuntu# groupadd group1
root@unix:/home/ubuntu# groupadd group2
root@unix:/home/ubuntu# groups user1
user1 : user1
root@unix:/home/ubuntu# groups user2
user2 : user2
root@unix:/home/ubuntu# █
```

```
root@unix:/home/ubuntu# groups user1 user2
user1 : user1
user2 : user2
root@unix:/home/ubuntu# usermod -G group1 user1
root@unix:/home/ubuntu# groups user1
user1 : user1 group1
root@unix:/home/ubuntu# usermod -G group2 user1
root@unix:/home/ubuntu# groups user1
user1 : user1 group2
root@unix:/home/ubuntu# usermod -G group1,group2 user1
root@unix:/home/ubuntu# groups user1
user1 : user1 group1 group2
root@unix:/home/ubuntu# usermod -G user1 user1
root@unix:/home/ubuntu# groups user1
user1 : user1
```

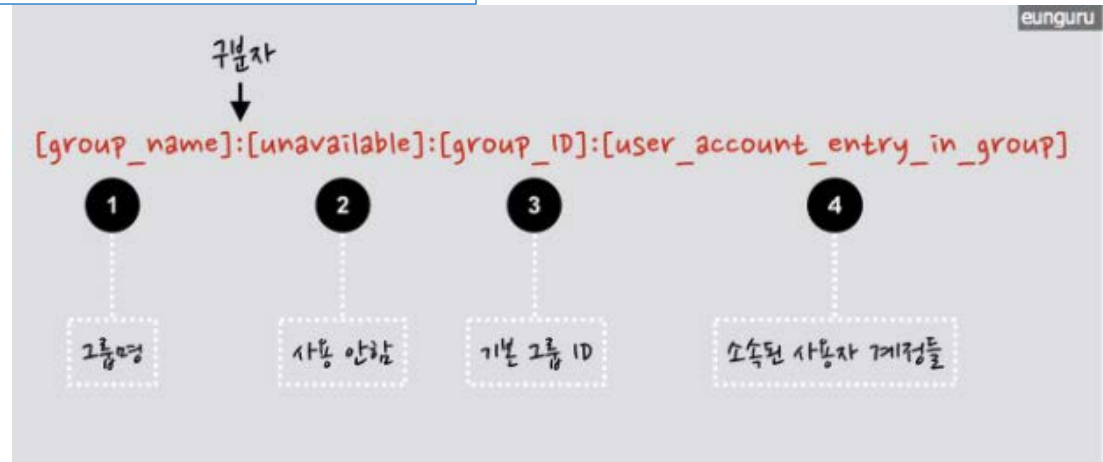


/etc/group

- 그룹에 관한 정보 및 현재 그룹의 소속 사용자 표시
- 이 파일을 직접 수정하여 그룹 정보를 변경할 수 있으나, 오타 등의 위험이 있음

```
root@unix:/home/ubuntu# usermod -G group1 user1
root@unix:/home/ubuntu# usermod -G group1 user2
root@unix:/home/ubuntu# vi /etc/group
```

```
unix201911832:x:1089:
unix201818743:x:1090:
unix201918323:x:1091:
unix201414246:x:1092:
unix201716403:x:1093:
unix201513451:x:1094:
unix201811162:x:1095:
unix201918328:x:1096:
unix201818139:x:1097:
unix201911839:x:1098:
unix201911840:x:1099:
unix201514723:x:1100:
unix201414253:x:1101:
unix201918336:x:1102:
unix201716419:x:1103:
unix201918334:x:1104:
unix201711097:x:1105:
newgroup:x:1106:newuser2,mytest
newuser2:x:1108:mytest
user1:x:1109:
user2:x:1110:
group1:x:1111:user1,user2
group2:x:1112:
"/etc/group" 170L, 3264C
```



* <https://eunguru.tistory.com/88>

특정 계정 혹은 그룹에 관리 권한 부여

1. 사용자를 sudo 그룹에 포함시킴

- Sudo 그룹: 관리 권한을 가진 그룹. (Super group)
- `# usermod -aG sudo user1`
- 사용자를 생성할 때 부터 sudo 그룹에 포함시키기
 - `# useradd -m -G sudo gunsmoke`

2. /etc/sudoers 파일 수정

- 관리 권한을 가진 유저를 관리하는 파일
- 오타로 인해 시스템 오류가 발생할 수 있으니, 가급적 사용하지 않음
- 편의상 해당 파일을 직접 수정할 수도 있지만,
- /etc/sudoers.d/ 경로 아래에 새로운 파일을 추가하는 것이 바람직함

```
root@unix:~# vi /etc/sudoers.d/my_sudo_users
root@unix:~# cat /etc/sudoers.d/my_sudo_users
user1 ALL=(ALL) ALL
user2 ALL=(ALL) NOPASSWD:ALL
```

← Sudo 사용 시, 사용자 패스워드를 입력하여야 함
← Sudo 사용 시, 사용자 패스워드 입력하지 않아도 됨



예제

```
root@unix:/home/ubuntu# login user1
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

System information as of Mon Sep 14 14:54:23 UTC 2020

System load:  0.0           Processes:            143
Usage of /:   15.2% of 19.21GB Users logged in:      3
Memory usage: 25%          IPv4 address for ens3: 10.0.0.7
Swap usage:   0%

 * Kubernetes 1.19 is out! Get it in one command with:

    sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

36 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Sep 14 14:49:56 UTC 2020 on pts/0
No directory, logging in with HOME=/
$ sudo -s
[sudo] password for user1:
# exit
```

```
root@unix:/home/ubuntu# login user2
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

System information as of Mon Sep 14 15:01:30 UTC 2020

System load:  0.0           Processes:            144
Usage of /:   15.2% of 19.21GB Users logged in:      3
Memory usage: 25%          IPv4 address for ens3: 10.0.0.7
Swap usage:   0%

 * Kubernetes 1.19 is out! Get it in one command with:

    sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

36 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

No directory, logging in with HOME=/
$ sudo -s
# █
```



Permission for a File



파일의 접근 권한

- 각 파일에는 소유 사용자, 소유 그룹, 그에 따른 접근 권한이 존재함
 - 소유자, 소유그룹, 기타 사용자에게 대해 읽기/쓰기/실행 여부를 제어할 수 있음
- Stat 명령어를 사용했을 때의 예

```
unix201512345@unix:~$ stat newfile
```

```
File: newfile
Size: 14          Blocks: 8          IO Block: 4096   regular file
Device: fc01h/64513d Inode: 270312       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1001/unix201512345)   Gid: ( 1001/unix201512345)
Access: 2020-09-08 08:27:04.169145344 +0000
Modify: 2020-09-08 08:27:00.185212452 +0000
Change: 2020-09-08 08:27:00.185212452 +0000
Birth: -
```



소유자 변경: chown(change owner)

- 사용법

```
$ chown 사용자 파일
```

```
$ chown [-R] 사용자 디렉터리
```

파일 혹은 디렉터리의 소유자를 지정된 사용자로 변경한다.

-R 옵션: 디렉터리 아래의 모든 파일과 하위 디렉터리에 대해서도 소유자를 변경한다.

- 예

```
$ chown hong cs1.txt
```

chown: changing ownership of 'cs1.txt': 명령을 허용하지 않음

```
$ su
```

암호:

```
$ chown hong cs1.txt
```

```
$ ls -l cs1.txt
```

```
-rw-r--r--. 1 hong cs 2088 10월 21 16:25 cs1.txt
```


그룹 변경: chgrp(change group)

- 사용법

```
$ chgrp 그룹 파일
```

```
$ chgrp [-R] 그룹 디렉터리
```

파일 혹은 디렉터리의 그룹을 지정된 그룹으로 변경한다. -R 옵션을 사용하면 지정된 디렉터리 아래의 모든 파일과 하위 디렉터리에 대해서도 그룹을 변경한다.



일반적으로 사용하는 방법

- `chown ubuntu:ubuntu <filename>`
 - 해당 파일의 소유자 및 그룹을 한번에 수정함
- `chown -R ubuntu:ubuntu .`
 - 현재 디렉토리 이하의 모든 파일에 대해 모두 적용



접근 권한(permission mode)

- 파일에 대한 읽기(r), 쓰기(w), 실행(x) 권한

권한	파일	디렉터리
r	파일에 대한 읽기 권한	디렉터리 내에 있는 파일명을 읽을 수 있는 권한
w	파일에 대한 쓰기 권한	디렉터리 내에 파일을 생성하거나 삭제할 수 있는 권한
x	파일에 대한 실행 권한	디렉터리 내로 탐색을 위해 이동할 수 있는 권한

- 소유자(owner)/그룹(group)/기타(others)로 구분하여 관리
- 예: `rwX r-X r-X`



접근권한의 예

접근권한	의미	8진수
rw-rwxrwx	소유자, 그룹, 기타 사용자 모두 읽기,쓰기,실행 가능	777
rw-r-xr-x	소유자만 읽기,쓰기,실행 가능, 그룹, 기타 사용자는 읽기,실행 가능	755
rw-rw-r--	소유자와 그룹만 읽기,쓰기 가능, 기타 사용자는 읽기만 가능	664
rw-r--r--	소유자만 읽기,쓰기 가능, 그룹과 기타 사용자는 읽기만 가능	644
rw-r-----	소유자만 읽기,쓰기 가능 그룹은 읽기만 가능	640
rwX-----	소유자만 읽기,쓰기,실행 가능	700



접근권한 변경: chmod(change mode)

- 사용법

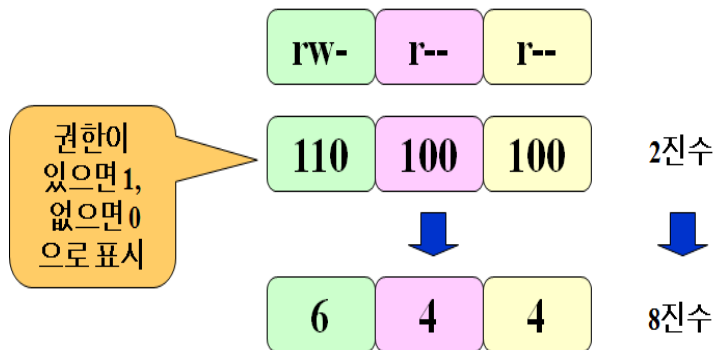
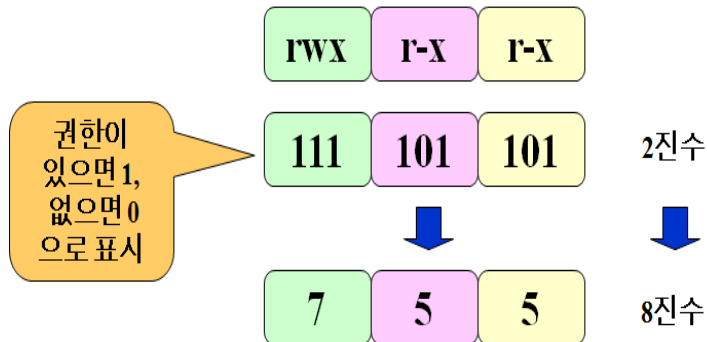
```
$ chmod [-R] 접근권한 파일 혹은 디렉터리
```

파일 혹은 디렉터리의 접근권한을 변경한다. -R 옵션을 사용하면 지정된 디렉터리 아래의 모든 파일과 하위 디렉터리에 대해서도 접근권한을 변경한다.



접근권한 표현: 8진수

• 접근권한 8진수 변환



• 사용 예

```
$ chmod 644 cs1.txt
$ ls -l cs1.txt
-rw-r--r-- 1 chang ... cs1.txt
```

접근권한	8진수
rw-rwxrwx	777
rw-r-xr-x	755
rw-rw-r--	664
rw-r--r--	644
rw-r-----	640
rw-x-----	700

접근권한 표현: 기호

- 기호를 이용한 접근권한 변경

사용자범위 연산자 권한
[u|g|o|a]⁺ [+|-|=] [r|w|x]⁺

구분	기호와 의미
사용자 범위	u(user:소유자), g(group:그룹), o(others:기타 사용자), a(all:모든 사용자)
연산자	+(권한 추가), -(권한 제거), =(권한 설정)
권한	r(읽기 권한), w(쓰기 권한), x(실행 권한)

기호를 이용한 접근권한 변경

- 예

```
$ chmod g-w cs1.txt
```

```
$ ls -l cs1.txt
```

```
-rw-r--r-- 1 chang cs 2088 4월 16일 13:37 cs1.txt
```

```
$ chmod o-r cs1.txt
```

```
$ ls -l cs1.txt
```

```
-rw-r----- 1 chang cs 2088 4월 16일 13:37 cs1.txt
```

```
$ chmod g+w,o+rw cs1.txt
```

```
$ ls -l cs1.txt
```

```
-rw-rw-rw- 1 chang cs 2088 4월 16일 13:37 cs1.txt
```

