GlusterFS - despersed Type Volume 생성 실습

GFS Volume Features – Dispersed Volume

GFS Volume type
이해 편의상 비슷한 HW RAID 레벨의 특성을 같이 기재하였다
- Distribute Volume (RAID 0)
- Replicate Volume (RAID 1)
- Distribute + Replicate Volume (RAID 0+1)
- **Dispersed Volume (RAID 5,6)**
- Dispersed + Replicate Volume (RAID 50 , 60)

Dispersed Volume?
HW RAID Level 에서의 비슷한 유형을 찾는다면 RAID 5 와 6의 중간 단계에 있다.
최소 3개 이상의 Brick 부터 구성이 가능하며
볼륨 구성 시 복구 가능한 분산 패리티의 수량을 명시할 수 있어(Redundancy)
RAID 5 보다 복구 유연성이 좋다고 볼 수 있다.

Replicate 보다 가용량을 확보하며 최소한의 안전장치를 확보한다고 볼 수 있다.

Distribute 과의 차이점?
같은 분산의 의미이지만 Distribute 는 순수한 파일을 각 Brick 에 분배하는 것이고
Dispersed 는 erasure coding 방식 (패리티와 사실상 유사) 의
인코딩된 전용 데이터를 만들어 Brick 에 분산한다.

Create Command
e.x)
```
$ gluster volume create NEW-VOLNAME [disperse-data COUNT] [redundancy COUNT] [transport tcp | rdma | tcp,rdma] NEW-BRICK...
```

disperse-data COUNT : 실제 사용되는 용량에 들어가는 Brick 의 수

redundancy COUNT : 데이터 분산 패리티의 수량 지정 가능.
해당 수량만큼 Brick 을 가진 물리 디스크가 Failed 되어도 클러스터는 유지된다.

RAID 5가 1개의 패리티. RAID 6 이 2개의 패리티만이 가능한 것보다 유연하다.
이 값만 지정해 줄 시 자동적으로 disperse-data COUNT 가 계산된다.

RMW(Read-Modify-Write) 방식으로 데이터를 기록하기 때문에
효율적인 I/O 를 위한 Brick 비율이 존재한다
(e.x DISPERSE 6 REDUNDANCY 2)

GFS Volume Create

100GB * 3개의 Brick 을 Redundancy 1개 (RAID 5) 방식으로 구성

Peer Probe
Node01)

노드들을 우선 인식해야 한다. 마스터 자기 자신은 제외

```
gluster peer probe 211.239.151.197
gluster peer probe 211.239.151.198
gluster peer status
```

Volume Create
노드마다 100GB 씩 할당한 파티션을 묶어서 클러스터를 구성한다.

```
gluster volume create data disperse-data 2 redundancy 1 transport tcp 211.239.151.196:/gfs_node 211.239.151.197:/gfs_node 211.239.151.198:/gfs_node force
```

```
# check
volume create: data: success: please start the volume to access data
```

기본 상태 확인
```
gluster volume info
```

볼륨 시작. 처음 생성시 stop 되어 있으니 반드시 시작해야 함
```
gluster volume start
volume start: data: success
```

볼륨 확인
```
gluster volume status
```

```
Status of volume: data
Gluster process                                 TCP Port  RDMA
Port  Online  Pid
------------------------------------------------------------
----------------
Brick 211.239.151.196:/gfs_node                   49152     0
Y       65695
Brick 211.239.151.197:/gfs_node                   49152     0
Y       13261
Brick 211.239.151.198:/gfs_node                   49152     0
Y       13513
Self-heal Daemon on localhost                   N/A       N/A
Y       65712
Self-heal Daemon on 211.239.151.198             N/A       N/A
Y       13530
Self-heal Daemon on 211.239.151.197             N/A       N/A
Y       13278

Task Status of Volume data
------------------------------------------------------------
----------------
```

마운트 및 용량 확인
(All Node)
brick 2개 만큼의 용량 확보 되었고 1개는 redundancy(패리티) 상태이다.

```
mount -t glusterfs gfs-node01:/data /mnt
df -h /mnt
Filesystem        Size  Used Avail Use% Mounted on
gfs-node01:/data  200G  2.1G  198G   2% /mnt
```

fstab 자동 마운트 등록
클러스터이기 때문에 어떤 노드의 엔드포인트를 지정하여도 마운트에 문제는 없다.

```
echo "gfs-node01:data /mnt glusterfs defaults,_netdev 0 0" >>
/etc/fstab
```

Node Fail-over TEST

(노드 다운)
** 1 node down

Node 하나를 Poweroff 시 약 1분 정도의 딜레이가 걸리지만
클러스터는 정상 작동한다
gluster status volume

** 2 node down
# Redundancy 가 1개이므로 노드 2개가 다운되면 클러스터 볼륨은 연결 실패하게 된다.
[root@node01 mnt]# ll
ls: cannot open directory .: Transport endpoint is not connected

(노드 업)
** 1 node up
노드 2개가 살아났을 경우 남아있던 노드는 자동으로 다시 연결된다.
up 노드는 리마운트 해주어야 한다.

[root@node01 mnt]# ll
ls: cannot open directory .: Transport endpoint is not connected
[root@node01 mnt]# ll
total 0
-rw-r--r--. 1 root root 0 Jan 13 10:42 a
-rw-r--r--. 1 root root 0 Jan 13 10:44 b
-rw-r--r--. 1 root root 0 Jan 13 10:44 c

[root@node01 mnt]# gluster volume status
Status of volume: data
Gluster process   TCP Port RDMA Port Online Pid

Brick 211.239.151.196:/gfs_node   49152   0   Y   67046
Brick 211.239.151.197:/gfs_node   49153   0   Y   3551
Self-heal Daemon on localhost   N/A   N/A   Y   67063
Self-heal Daemon on 211.239.151.197   N/A   N/A   Y   3108

Task Status of Volume data

** 2 node up
정상화 된 것으로 보인다.
다수의 다량 데이터 검증은 필요하다

** 3 (all) node up
부팅 순서에 따라 자동 마운트는 안 되지만 스토리지 볼륨은 정상 online 된다.

Volume Expansion (ADD-BRICK)

node03 에 100GB Brick 3EA 추가할 것이다..
Dispersed 타입 볼륨은 3(1패리티) 비율의 Brick 으로 구성하였고
6(2 패리티) 조합이 밸런스가 권장된다.
- 기존 GFS Volume 에 증설
- 기존 브릭 교체 실습

```
# Check now
[root@node02 ~]# gluster volume info | egrep -iE "brick|t
ype"
Type: Disperse
Number of Bricks: 1 x (2 + 1) = 3
Transport-type: tcp
Bricks:
Brick1: 211.239.151.196:/gfs_node
Brick2: 211.239.151.197:/gfs_node
Brick3: 211.239.151.198:/gfs_node

# 추가할 BRICK 파티셔닝 ( sdc , sdd , sde 3개 구성하자 )
fdisk /dev/sdc <<EOF
n
p



w
EOF
mkfs.xfs -i size=512 /dev/sdc1
mkdir -p /add_brick1
mount /dev/sdc1 /add_brick1

echo "/dev/sdc1        /add_brick1        xfs      defaults
0 0" >> /etc/fstab
```

```
# check
[root@node03 ~]# df -h | grep brick
/dev/sdc1               100G   33M  100G   1% /add_bric
1
/dev/sdd1               100G   33M  100G   1% /add_bric
2
/dev/sde1               100G   33M  100G   1% /add_bric
3

# Volume ADD issue
3개 미만의 브릭을 추가하려고 하면 에러가 발생한다
[root@node03 ~]# gluster volume add-brick data gfs-
node03:/add_brick1
volume add-brick: failed: Incorrect number of bricks supplied
1 with count 3

새 Peer ( 물리적 노드 ) 가 아닌 같은 서버에 Brick 다수 중첩될 경우 경고 발생
[root@node03 ~]# gluster volume add-brick data gfs-
node03:/add_brick1   gfs-node03:/add_brick2   gfs-
node03:/add_brick
3
volume add-brick: failed: Multiple bricks of a disperse volume
are present on the same server. This setup is not optimal.
Bricks should be on different nodes to have best fault
tolerant configuration. Use 'force' at the end of the command
if you want to override this behavior.

force 옵션을 요구함
volume add-brick: success

# Volume brick check
체크 시 6개의 브릭 중 2개가 패리티로 설정된 균형비율 확인
[root@node03 ~]# gluster volume info | egrep -iE "brick"
Number of Bricks: 2 x (2 + 1) = 6
Bricks:
Brick1: 211.239.151.196:/gfs_node
Brick2: 211.239.151.197:/gfs_node
Brick3: 211.239.151.198:/gfs_node
Brick4: gfs-node03:/add_brick1
Brick5: gfs-node03:/add_brick2
Brick6: gfs-node03:/add_brick3
```

```
# Volume Size Check
brick 추가시 volume size 도 라이브하게 변경되었다.
[root@node03 ~]# df -h | grep mnt
gfs-node01:data          400G  4.2G  396G   2% /mnt


# Rebalance
추가된 Brick 으로의 고른 데이터 분산을 위해 사용
gluster volume rebalance data start
volume rebalance: data: success: Rebalance on data has been
started successfully. Use rebalance status command to check
status of the rebalance process.
ID: 9782e187-6cda-4e2b-aae9-0b78746d69fa


# check
1) status 에서 내용을 확인할 수 있다
Task Status of Volume data
------------------------------------------------------------
----------------
Task                 : Rebalance
ID                   : 9782e187-6cda-4e2b-aae9-0b78746d69fa
Status               : completed

2) 리밸런스 후 추가된 brick 에 패리티가 일부 분산된 내용을 확인할 수 있었다
[root@node03 add_brick3]# ll ( 리밸런스 전 )
total 0
[root@node03 add_brick3]# ll ( 리밸런스 후 )
total 8
-rw-r--r--. 2 root root 512 Jan 13 14:57 d
[root@node03 add_brick3]#
```

Setting Up Volumes – Gluster Docs

[AWS] EC2 직렬 콘솔에서 응급모드 진입 설정 – Amazon Linux 2

리눅스에서 패스워드 리셋이나 마운트 에러 등의 이슈로 응급 모드로 진입하여 작업해야 할 때가 있는데 AWS는 기본적으로 원격 관리이기 때문에 콘솔 화면이 보이지 않습니다.

Nitro Hypervisor 를 지원하는 T3 급 이상의 인스턴스는 AWS Console 에서 직접 시리얼 콘솔을 접근할 수 있는 옵션이 추가가 되었는데

GRUB Timeout 설정 시간이 짧아서 응급모드 진입이 어려운 이슈가 있습니다.
Amazon Linux 2 OS 의 GRUB 타임아웃 시간을 늘려서 확인할 수 있도록 해 봅니다.

마운트 실패로 인한 응급 모드의 큰 틀은 다음의 2가지로 분류가 됩니다.

1) `GRUB TIMEOUT` 변경
2) 응급 모드 진입 후 `mount` 옵션에 `nofail` 추가 시, 실패해도 진행 가능

IAM 접근 최소 조건

`# IAM`이 `Administrator Group` 이라면 상관 없으나, 디테일 권한 관리될 경우 다음의 정책을 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:EnableSerialConsoleAccess",
                "ec2:DisableSerialConsoleAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

GRUB TIMEOUT 값 수정하기

root 권한에서 Default GRUB 의 값을 수정 및 추가합니다.
Console Serial 은 root 패스워드가 콘솔 로그인 가능해야 합니다.
아마존의 타임아웃 권장값은 1입니다만 넉넉하게 10(초)로 변경합니다.

```
sed -i 's/GRUB_TIMEOUT=0/GRUB_TIMEOUT=10/g' /etc/default/grub
sed  -i  's/GRUB_TERMINAL="ec2-console"/GRUB_TERMINAL="console
serial"/g' /etc/default/grub
 echo  -e  GRUB_SERIAL_COMMAND=\"serial  --speed=115200\"  >>
/etc/default/grub

# check
cat /etc/default/grub

GRUB_CMDLINE_LINUX_DEFAULT="console=tty0
console=ttyS0,115200n8   net.ifnames=0   biosdevname=0
nvme_core.io_timeout=4294967295    rd.emergency=poweroff
rd.shell=0"
GRUB_TIMEOUT=10
GRUB_DISABLE_RECOVERY="true"
GRUB_TERMINAL="console serial"
GRUB_X86_USE_32BIT="true"
GRUB_SERIAL_COMMAND="serial --speed=115200"

# 실제 GRUB 에 적용합니다
grub2-mkconfig -o /boot/grub2/grub.cfg
```

EC2 Reboot 후 직렬 콘솔 확인

10초 지연 후 지나가므로 e 키를 바로 눌러 응급모드 진입

응급모드 진입

해당 위치에 single 구문을 추가 후 Ctrl + x 로 부팅

FSTAB 에서 nofail 옵션 사용

추가 마운트 옵션에 `nofail` 을 넣으면 부팅 시 실패해도 정상 진행됩니다

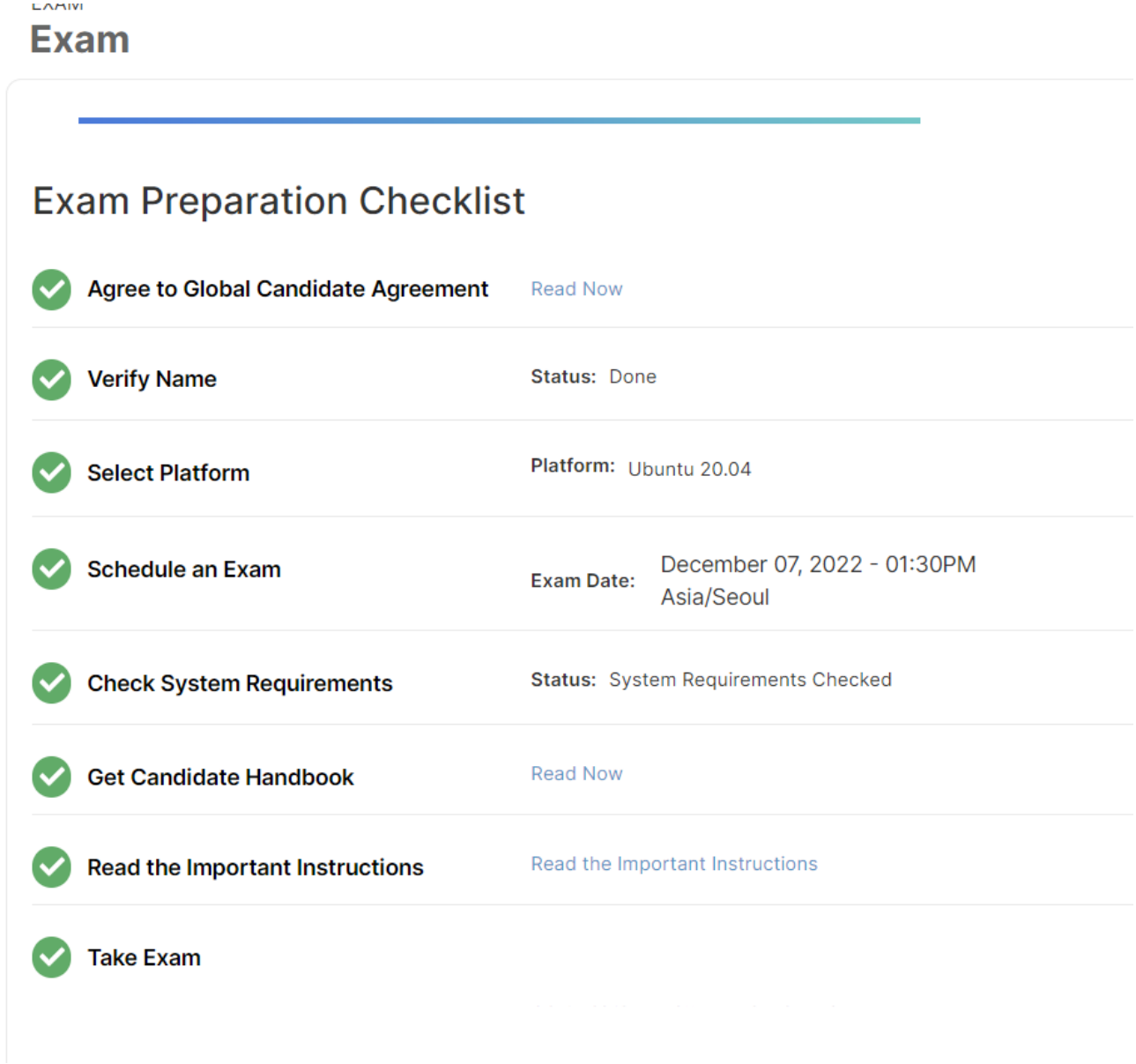
AWS 공식 링크(이미지 별)

2022.12.07 CKA(Certified Kubernetes Administrator) 시험 응시 소감

2022.06.25 부터 바뀐 시험환경 (PSI Secure Browser) 로 바뀌었다고 하는데 금번에 시험을 치루면서 환경에 대한 소감을 정리 하였습니다;

관련 업데이트 `URL :`
`https://training.linuxfoundation.org/bridge-migration-2021/`

시험 준비 – 최소 시험 시간 30분 전 입실 필수



CKA 는 원격 전용 시험이며 Schedule 된 시간 30분 전에
TAKE EXAM 버튼이 활성화 됩니다.

시험 시작 전 세팅 준비하는 시간으로 실제 시험시간에 카운팅되지는 않으나 늦어질 수록 실제 시험 시간이 늦어지므로 늦게 끝나게 됩니다.

 # 로그인 및 시험 등록은 해당 URL 에서 이루어 집니다.
https://trainingportal.linuxfoundation.org/learn/course/certified-kubernetes-administrator-cka/exam/exam

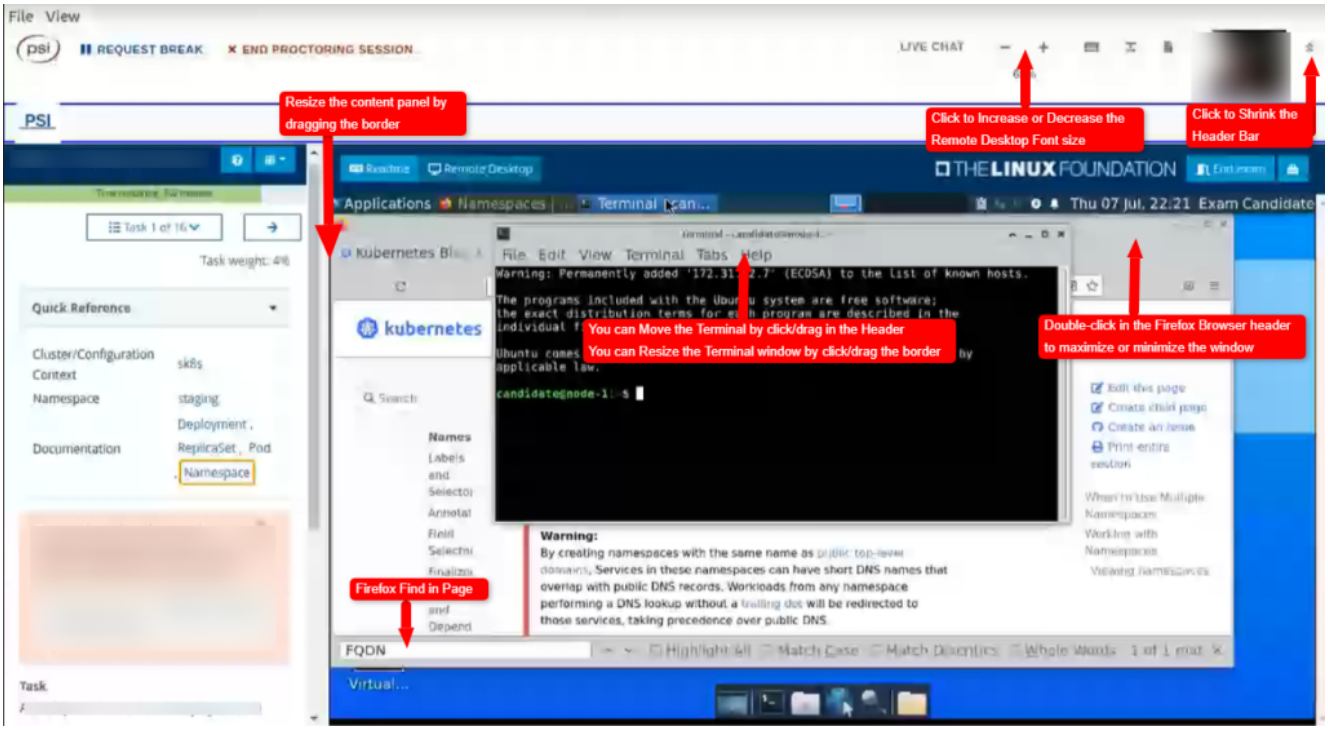
TAKE EXAM 을 누른 후 PSI 전용 브라우저를 설치할 수 있습니다.
클라이언트 환경에 Cotana, 원격(zoon, bluejean) 툴, VPN 등의 프로세스가 메모리에 떠 있을 경우 프로그램이 실행되지 않으므로 프로세스 종료해야 합니다
실행한 후 브라우저의 기능적인 신분증 촬영, 안면 촬영 등의 신원 확인 절차를 진행합니다.

이후 시험 감독관이 CHAT 으로 지시하는 내용들을 잘 따르면 됩니다.
 # 시험감독관은 저의 경우 다음의 상황을 요구했습니다
- 시험실 전체를 비추기
- 책상 밑 아래 비추기
- 책상에 노트북(데스크탑) 외 물건 치우기
(저의 경우 무선 마우스와, HDMI 용 집게를 치우라고 했습니다
마우스는 CHAT 으로 설명하고 허락 받았습니다)
- 셔츠 걷어서 팔, 손바닥(앞/뒤) 비춰보기
- 마스크 벗어서 보이지 않도록 칠판 스크린 뒤에 감추기

점검이 끝나면 협조에 고맙다고 하고 라벨이 달린 물병은 소지할 수 없음을 추가 안내합니다.

시험관이 원격에서 승인 시, 화면에 시험 랩이 나타나게 됩니다.
그 때부터 시간이 실제로 적용됩니다.

시험 환경 – 느림



저의 경우 회사 사무실에서 무선 와이파이로 진행했습니다.
외부 모니터는 지원하지 않기에 14인치 노트북을 사용하였는데
화면이 많이 작은데에다 해외망 원격 랩 시뮬이라 그런지 반응이 많이 느려 고생을 했습니다.

시험 진행시간은 2시간입니다.
문제는 총 17문제이며, 문제 내 2-3가지 연속된 과정을 물어볼 경우
부분점수가 배정됩니다.

단답형 문제는 2~4점
연속형 문제는 6~12점까지 배점이 매우 큽니다.

Flag 버튼이 있어서 문제를 순서대로 풀 필요는 없습니다.

시험 환경 URL :
https://docs.linuxfoundation.org/tc-docs/certification/tips-cka-and-ckad#adjusting-font-and-windows-in-the-examui

복사&붙여넣기

브라우저의 가상 실습 랩 안에서 Firefox 브라우저를 이용한
Kubernetes.io 의 Document 서치만 허용되는데
북마크 사용할 수 없는데에다 반응이 원활하지 않으므로
사전에 YAML 템플릿 위치 등을 숙지해 두는 것을 권장합니다.

복사-붙여넣기는 Ctrl+Shift+C , V 로 가능하다고 하지만
마우스 오른쪽으로 직접 Copy&Paste 를 권장합니다.

번역 & 플러그인

PSI 브라우저의 정해진 가상 랩 안에서 실행되는 것으로
크롬과는 달리 기본으로 번역 기능이 지원되지 않는 Firefox에
번역 플러그인을 설치할 수 없습니다.
문제의 영어 형식은 간단하기 때문에 크게 문제되지는 않을 것입니다.

시험관과는 CHAT 으로만 소통이 가능합니다.
화장실을 가기 전 반드시 전달하고 가야 하며 시험시간은 그대로 흘러갑니다.

점수

총점 100점에서 **66점(2/3) 이상 획득시 Pass** 입니다.
24시간 이내에 메일 및 결과 확인할 수 있게 되어 있습니다.
저의 경우에는 정확히 24시간 뒤 결과가 회신되었습니다.

Fail 시 Reschedule 기회를 한번 더 준다고 합니다.

바로바로 해결하기 어려운 문제는 Flag 찍어놓고 다른 문제부터 빠르게 푸는 것을 권장합니다.
반응도 느리고 망설이면 생각보다 시간이 부족했습니다.

기타

시험 접수 시 이름을 한글로 했을 경우
실제 자격증에 이름이 표기가 되지 않을 수 있습니다.
여권/신용카드에 기재된 영문 네임과 동일하게 접수 권장드립니다.

growpart 으로 볼륨 확장

지난번에 이어 기본 시스템에 디스크(볼륨)를 추가 확장 하는 방법을 해보겠습니다.

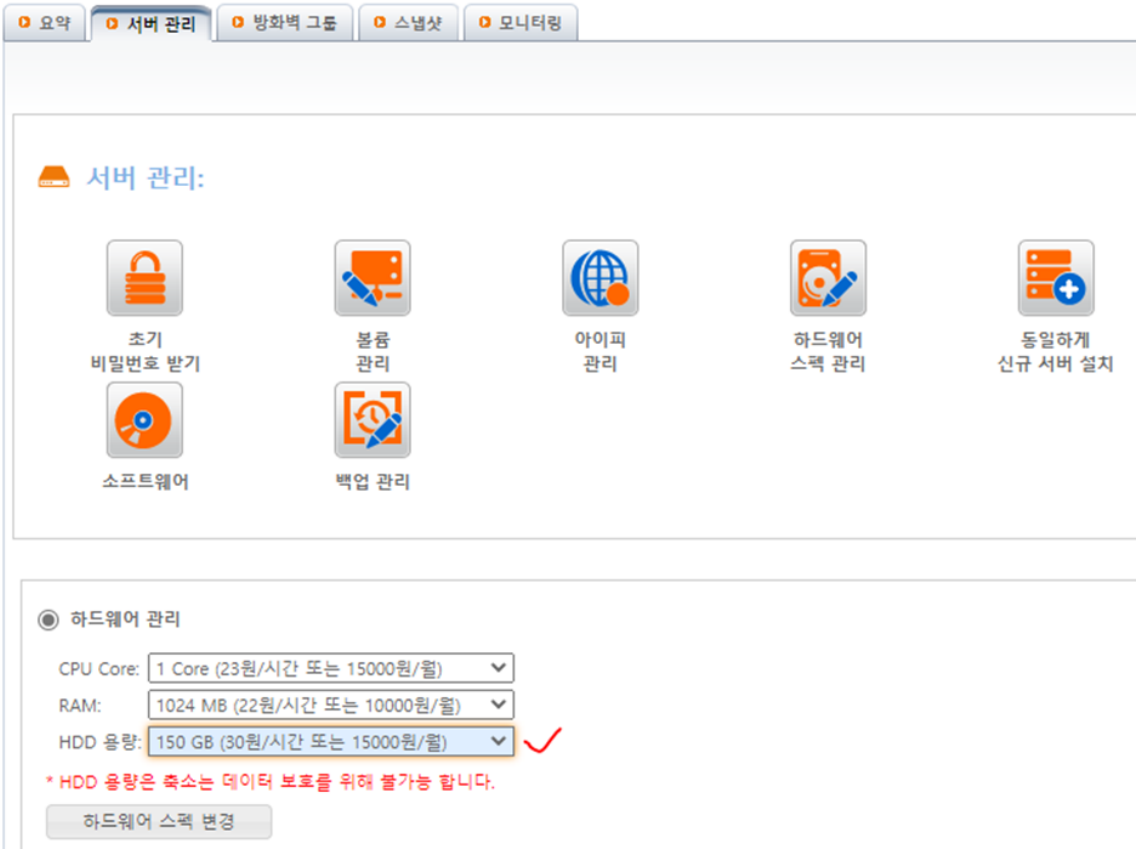
현재 `/home` 파티션(80G)에 볼륨(50G) 추가하여 확장 하는 방법을 알아 보겠습니다.

가상화/클라우드 환경에서 `growpart` 를 이용하여 디스크 용량을 유연하게 조절이 가능합니다.

아래 예제는 실제 사용중인 `OS Disk`의 용량을 가상화/클라우드에서 확장해주는 과정을 설명합니다.

참고로 `cloud-utils-growpart` 패키지는 설치 되어 있어야 합니다.

`FlexCloud` 콘솔에서 디스크 용량을 늘립니다.



HDD 용량 100G -> 150G 로 용량 증설 합니다.

1. 가상화 / 클라우드 플랫폼에서 해당 머신의 디스크 용량을 증설 해준다.

   (하드웨어 관리 > HDD 용량 에서 용량 추가 , 리부팅 됨)

```
# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0      2:0    1    4K  0 disk
sda      8:0    0  150G  0 disk
sda1   8:1    0   20G  0 part /
sda2   8:2    0   80G  0 part /home
# fdisk -l
```

```
Disk /dev/sda: 161.1 GB, 161061273600 bytes, 314572800 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 4096 bytes

I/O size (minimum/optimal): 4096 bytes / 4096 bytes

Disk label type: dos

Disk identifier: 0x000822ad

   Device Boot      Start         End      Blocks   Id  System

/dev/sda1   *        2048    41943039    20970496   83  Linux

/dev/sda2        41943040   209715199    83886080   83  Linux
```

2. 파티션 확장

growpart 명령어를 사용하여 /home 영역에 추가한 디스크를 전체 할당 합니다.

```
[root@localhost ~]# growpart /dev/sda 2

CHANGED: partition=2 start=41943040 old: size=167772160 end=209715200 new: size=272629727 end=314572767

[root@localhost ~]#
```

lsblk 에서 보면 용량 증가 부분 확인 됩니다.

```
[root@localhost ~]# lsblk

NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT

fd0      2:0    1    4K  0 disk
```

```
sda      8:0    0  150G  0 disk
□□sda1   8:1    0   20G  0 part /
□□sda2   8:2    0  130G  0 part /home
```

df -h 로 보면 아직 /home 파티션 용량 증가 안된 것으로 보입니다..

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        20G  1.5G   19G   8% /
devtmpfs        905M     0  905M   0% /dev
tmpfs           915M     0  915M   0% /dev/shm
tmpfs           915M  8.7M  906M   1% /run
tmpfs           915M     0  915M   0% /sys/fs/cgroup
/dev/sda2        80G   35M   80G   1% /home
```

3. 파티션 리사이즈를 합니다.

파일시스템에 종류에 따라 파티션 리사이즈 명령이 다르므로 blkid 명령으로 확인해 보고 맞는 명령어로 진행 합니다

파일시스템 종류 확인

```
[root@localhost ~]# blkid
```

/dev/sda1: UUID="67966418-a410-4461-8aba-fb45131b4737" TYPE="xfs"

/dev/sda2: UUID="ab5b8818-676a-448f-a827-0686fb4e7122" TYPE="xfs"

ext4 파일시스템일 경우 resize2fs 명령어를 사용

xfs파일시스템일 경우 xfs_growfs 명령어를 사용

여기서는 xfs 파일 시스템을 사용하고 있어 xfs_growfs 명령어를 사용 합니다.

```
[root@localhost ~]# xfs_growfs -d /home

meta-data=/dev/sda2              isize=512    agcount=4, agsize=5242880 blks

         =                       sectsz=4096  attr=2, projid32bit=1

         =                       crc=1        finobt=0 spinodes=0

data     =                       bsize=4096   blocks=20971520, imaxpct=25

         =                       sunit=0      swidth=0 blks

naming   =version 2              bsize=4096   ascii-ci=0 ftype=1

log      =internal               bsize=4096   blocks=10240, version=2

         =                       sectsz=4096  sunit=1 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0
```

```
data blocks changed from 20971520 to 34078715

[root@localhost ~]#
```

4. /home 파티션 용량 변경을 확인 합니다.

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        20G  1.5G   19G   8% /
devtmpfs        905M     0  905M   0% /dev
tmpfs           915M     0  915M   0% /dev/shm
tmpfs           915M  8.7M  906M   1% /run
tmpfs           915M     0  915M   0% /sys/fs/cgroup
/dev/sda2       130G   35M  130G   1% /home
tmpfs           183M     0  183M   0% /run/user/0
```

LVM 볼륨 확장

지난번에 이어 LVM시스템에 디스크(볼륨)를 추가 확장 하는 방법을 해보겠습니다.
볼륨(20G) 추가하여 현재 /home 파티션(70G)을 확장 하는 방법을 알아 보겠습니다.

1. 기존 디스크 확장

물리장치(sdb) 또는 볼륨 추가 후 표시 용량이 설정한(20G) 용량으로 표시되어야 합니다. 용량에 변화가 변화가 없으면 재부팅 합니다.

```
# lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0                     2:0    1    4K  0 disk
sda                     8:0    0  100G  0 disk
sda1                  8:1    0    1G  0 part /boot
sda2                  8:2    0   30G  0 part /
sda3                  8:3    0   69G  0 part
vg--home-lv--home 253:0    0   69G  0 lvm  /home
sdb                     8:16   0   20G  0 disk
```

2. 파티션 생성

/dev/sdb에 sdb1파트션을 만들고 Linux LVM으로 바꿔 줍니다.

```
# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.23.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x49a98a24.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal

I/O) size boundary is recommended, or performance may be impacted.

Command (m for help):
Command (m for help): n
Partition type:

   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
Using default value 41943039
Partition 1 of type Linux and of size 20 GiB is set
```

```
Command (m for help): p
Disk /dev/sdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: dos
Disk identifier: 0x49a98a24

Device Boot      Start         End      Blocks   Id  System

/dev/sdb1            2048    41943039    20970496   83  Linux

Command (m for help):
Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'
Command (m for help): p
Disk /dev/sdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

Disk label type: dos
Disk identifier: 0x49a98a24

Device Boot      Start         End      Blocks   Id  System

/dev/sdb1            2048    41943039    20970496   8e  Linux
LVM

Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

3. PV 생성

pvcreate 로 물리 볼륨 생성 및 pvdisplay 으로 확인 합니다.

```
[root@localhost ~]# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
```

```
[root@localhost ~]# pvdisplay

  --- Physical volume ---
  PV Name               /dev/sda3
  VG Name               vg-home
  PV Size               <69.00 GiB / not usable 3.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              17663
  Free PE               0
  Allocated PE          17663
  PV UUID               dFUFjg-ETp6-aCkV-UHpw-HpP7-e40T-7L00p3
  "/dev/sdb1" is a new physical volume of "<20.00 GiB"

  --- NEW Physical volume ---

  PV Name               /dev/sdb1
  VG Name
  PV Size               <20.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               VjYTMM-ECCZ-GWrd-WpNS-OxMI-qYns-pnEr9x
```

4. VG 확장

기존 볼륨 그룸 확인 하고 vgextend로 볼륨 그룹을 확장합니다.
볼륨 정보 확인

```
# vgdisplay

  --- Volume group ---

  VG Name               vg-home
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
```

```
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <69.00 GiB
  PE Size               4.00 MiB
  Total PE              17663
  Alloc PE / Size       17663 / <69.00 GiB
  Free  PE / Size       0 / 0
  VG UUID               MESw3F-j0gs-yzOo-bwR9-34nn-u0dP-qHOHEC
```

추가 볼륨을 확장

```
[root@localhost ~]# vgextend vg-home /dev/sdb1

  Volume group "vg-home" successfully extended
```

볼륨확장 후 정보 확인

```
[root@localhost ~]# vgdisplay

  --- Volume group ---

  VG Name               vg-home
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               88.99 GiB
  PE Size               4.00 MiB
  Total PE              22782
  Alloc PE / Size       17663 / <69.00 GiB
```

```
  Free  PE / Size       5119 / <20.00 GiB
  VG UUID               MESw3F-j0gs-yzOo-bwR9-34nn-u0dP-qHOHEC
```

5. LV 확장 (lvextend)

lvdisplay로 물리 볼륨을 확인한다.

```
# lvdisplay

  --- Logical volume ---

  LV Path                /dev/vg-home/lv-home
  LV Name                lv-home
  VG Name                vg-home
  LV UUID                AY0b7g-L0pN-DrTm-Lhgb-vqU5-T2W1-0cFe4e
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2022-05-11 16:20:59 +0900
  LV Status              available
  # open                 1
  LV Size                <69.00 GiB
  Current LE             17663
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:0
```

6. 리사이즈 (resize2fs)

작업전 용량 확인

```
[root@localhost ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on

/dev/sda2                    30G  1.3G   29G   5% /
devtmpfs                    484M     0  484M   0% /dev
tmpfs                       493M     0  493M   0% /dev/shm
tmpfs                       493M   19M  474M   4% /run
tmpfs                             493M      0   493M    0%
```

```
/sys/fs/cgroup
/dev/sda1                       1014M  140M  875M  14% /boot
/dev/mapper/vg--home-lv--home     68G   55M   65G   1% /home
```

현재 작업하는 파티션 파일시스템 종류 확인 해보고 맞는 것으로 진행 하면 됩니다.

```
xfs인 경우 -> xfs_growfs
ext4 인 경우 -> resize2fs
```

파일시스템 종류 확인

```
[root@localhost ~]# blkid /dev/vg-home/lv-home

/dev/vg-home/lv-home:                   UUID="c29e7a45-
dc2f-46b0-9f9a-62aa7e17ad07" TYPE="ext4"

[root@localhost ~]# resize2fs /dev/vg-home/lv-home

resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vg-home/lv-home is mounted on /home; on-
line resizing required
old_desc_blocks = 9, new_desc_blocks = 12
The filesystem on /dev/vg-home/lv-home is now 23328768 blocks
long.

[root@localhost ~]# resize2fs /dev/vg-home/lv-home

resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vg-home/lv-home is mounted on /home; on-
line resizing required
old_desc_blocks = 9, new_desc_blocks = 12
The filesystem on /dev/vg-home/lv-home is now 23328768 blocks
long.
```

7. 용량 확인

```
# /home 파티션에 추가된 용량 확인 됩니다.

[root@localhost ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted
on
/dev/sda2                       30G  1.3G   29G   5% /
```

```
devtmpfs                        484M     0  484M   0% /dev
tmpfs                           493M     0  493M   0% /dev/shm
tmpfs                           493M   19M  474M   4% /run
tmpfs                                493M      0   493M    0%
/sys/fs/cgroup
/dev/sda1                      1014M  140M  875M  14% /boot
/dev/mapper/vg--home-lv--home    88G   59M   84G   1% /home
```

UFW 기본 설정

서비스 확인

```
systemctl status ufw
```

서비스 중지 및 비활성화

```
# 데몬 자체를 내리기 ( 비권장 )
systemctl stop ufw && systemctl disable ufw

# 기능 활성화 체크. inactive 이면 ufw 비 동작 상태
root:~# ufw status
Status: active
To                         Action      From
--                         ------      ----
Anywhere                   ALLOW       111.111.111.111

# ufw 활성화
ufw enable

# ufw 내리기
ufw disable
```

기본 룰 설정 파일 경로

```
cat /etc/ufw/user.rules
```

(예제) 특정 포트와 특정 IP만 접속 허용 시키기

```
# 우선 Inbound 모두 차단 ( 원격에서는 주의! )
ufw default deny incoming

# Outbound 는 모두 허용
ufw default allow outgoing

# 특정 포트 전체 허용 하기
ufw allow 3306/tcp

# 특정 아이피는 전체 접근 허용하기
ufw allow from 111.111.111.111/32

# 특정 아이피에 80 포트만 허용하기
ufw allow from 111.111.111.111/32 to any port 80 proto tcp

# 룰 리로드
ufw enable

# 부팅 시 ufw 자동으로 startup 올리기 ( 데몬 enable 과 별개이다 )
update-rc.d -f ufw defaults

### 모든 정책 리셋하기 ( 리셋 시 inactive 상태로 변환 )
ufw reset
```

로그 확인하기

```
# journalctl 의 커널 로그로 확인하면 외부 접근 차단 로그를 확인 가능
journalctl -xe

Aug 03 09:41:39 Docker-host kernel: [UFW BLOCK] IN=ens18 OUT=
MAC=be:45:a7:40:2f:02:76:d6:83:3e:ba:d3:08:00
SRC=211.239.150.48 >
Aug 03 09:41:40 Docker-host kernel: [UFW BLOCK] IN=ens18 OUT=
MAC=be:45:a7:40:2f:02:76:d6:83:3e:ba:d3:08:00
SRC=211.239.150.48
```

---

mysql8 xtrabackup (Docker) 백업 / 복구

Container 기반 Mysql8에서의 xtrabackup을 이용한 Full Backup / Restore 방법

몇일 전, Docker 를 이용한 DB 서버 이용 도중 DB 전체 백업(Full Backup)을 할 일이 생겼습니다.
백업 도중 Host machine 의 메모리 부족으로, OOM 이 동작하여 컨테이너가 계속 종료되는 일이 발생하여,
다른 백업 방법을 찾던 도중 오픈소스 백업 툴인 “xtrabackup” 을 이용하여 보았습니다.
mysqldump 와 대비해 장/단점이 뚜렷하여, 여러 방면으로 적용할 수 있어서 사용 방법을 포스팅 하도록 하겠습니다.

> ***xtrabackup 이란?***
>
> *– Mysql / Mariadb 등 에서 백업에 사용되는 오픈소스 백업 유틸리티*
> *– xtrabackup은 엔진 데이터를 그대로 복사하여 백업 / 복구 하는 방식입니다.*
> *– 대용량 백업의 경우mysqldump 대비 xtrabackup 이 서버의 리소스 사용 및 속도 면에서 매우 강력한 이점이 있다.*
> *– 단점 : 데이터를 복사하여 백업하는방식이기 때문에 대용량 백업을 진행할 시, Backup Server에 충분한 용량 확보 필요*

xtrabackup 설치

> *참고 사항*
>
> *xtrabackup 설치는 Host와 Mysql Container 내부 둘 다 설치해야 하며, 설치과정은 동일하다.*
> *(컨테이너 OS / Host OS확인 하여 각 버전에 맞춰 설치)*
>
> *해당 가이드에선 docker / docker-compose / mysql container 는 이미 설치 되어 있는 상태에서 시작한다.*

```
version: "3"
services:
    test_database:
        container_name: mysql8
        image: [redacted]
        environment:
          MYSQL_DATABASE: test_db
          MYSQL_ROOT_PASSWORD: 1234
          MYSQL_ROOT_HOST: '%'
        command:
          - mysqld
          - --character-set-server=utf8mb4
          - --collation-server=utf8mb4_unicode_ci
          - --default-authentication-plugin=mysql_native_password
        volumes:
              #          - ./etc/mysql:/etc/mysql:rw
          - ./etc/mysql/conf.d:/etc/mysql/conf.d:rw
          - ./var/lib/mysql:/var/lib/mysql:rw
          - ./bak:/home/bak
            #          - ./var/lib/mysql/mysql-files:/var/lib/mysql/mysql-files:rw
        logging:
         driver: "json-file"
         options:
          max-size: "10m"
          max-file: "10"
        ports:
          - 3306:3306
        ulimits:
          memlock:
            soft: -1
            hard: -1
          nofile:
            soft: 65536
            hard: 65536
```

** mysql docker-compose.yml*

주의 사항

xtrabackup 진행 시 mysql 데이터 복사가 진행될 때 로그가 매우 많이 쌓이기 때문에, 서버의 용량이 충분치 않다면 logrotate 설정이 필수로 되어 있어야 한다.

docker-compose up -d

- mysql8 container 기동 (-d : 백그라운드 실행)

docker ps

- Container ID 확인

docker exec -it [Container ID] /bin/bash

- Container 내부 접근

```
root@localhost:/home/data# docker-compose up -d
[+] Running 2/2
 # Network data_default  Created
 # Container mysql8      Started
root@localhost:/home/data# docker ps
CONTAINER ID   IMAGE
80bf3def47bd   [redacted]
root@localhost:/home/data# docker exec -it 80b /bin/bash
root@80bf3def47bd:/#
```

apt update -y && apt upgrade -y

- 패키지 최신화 진행

```
root@80bf3def47bd:/# apt update -y && apt upgrade -y
Get:1 http://security.debian.org/debian-security buster/updates InRelease [65.4 kB]
Get:2 http://deb.debian.org/debian buster InRelease [122 kB]
Get:3 http://deb.debian.org/debian buster-updates InRelease [51.9 kB]
Get:4 http://repo.mysql.com/apt/debian buster InRelease [22.1 kB]
Get:5 http://security.debian.org/debian-security buster/updates/main amd64 Packages [329 kB]
Get:6 http://deb.debian.org/debian buster/main amd64 Packages [7911 kB]
Get:7 http://deb.debian.org/debian buster-updates/main amd64 Packages [8788 B]
Get:8 http://repo.mysql.com/apt/debian buster/mysql-8.0 amd64 Packages [8447 B]
Fetched 8518 kB in 2s (4655 kB/s)
```

apt install -y wget lsb-release

- 설치에 필요한 패키지 설치
- wget : xtrabackup 홈페이지에서 패키지파일(.deb) 을 가져와 설치하기 위함
- lsb-release : OS의 배포판 버전 확인이 필요함

```
root@80bf3def47bd:/# apt install -y wget lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (10.2019051400).
The following additional packages will be installed:
  ca-certificates libpcre2-8-0 libpsl5 publicsuffix
The following NEW packages will be installed:
```

lsb_release -a

- OS 버전 확인

```
root@80bf3def47bd:/# lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
```

- xtrabackup 다운로드 페이지 접근

URL : https://www.percona.com/downloads/Percona-XtraBackup-LATEST/

Download Percona XtraBackup 8.0

Version:

Percona XtraBackup 8.0.28-21

Software:

Debian GNU/Linux 10.0 ("buster")

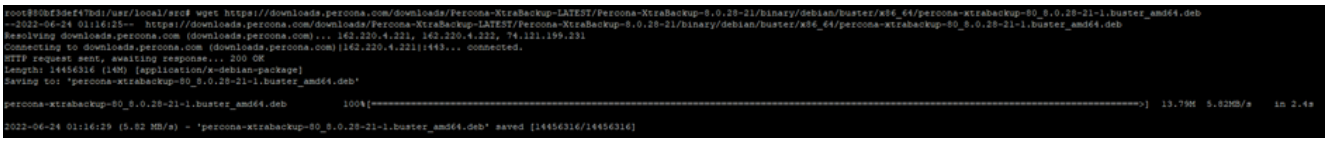
Prefer to install from our APT repository? View our online documentation to learn how.

Download All Packages Together	
Percona-XtraBackup-8.0.28-21-r78878e9b608-buster-x86_64-bundle.tar	427.5 MB
Download Packages Separately	
percona-xtrabackup-80_8.0.28-21-1.buster_amd64.deb	13.8 MB
percona-xtrabackup-dbg-80_8.0.28-21-1.buster_amd64.deb	406.3 MB
percona-xtrabackup-test-80_8.0.28-21-1.buster_amd64.deb	7.4 MB

- mysql8 버전 이상의 경우, xtrabackup 8 버전 이상을 이용해야한다.
- 다운로드 링크를 복사한다.

wget https://downloads.percona.com/downloads/Percona-XtraBackup-LATEST/Percona-XtraBackup-8.0.28-21/binary/debian/buster/x86_64/percona-xtrabackup-80_8.0.28-21-1.buster_amd64.deb

- 서버에 설치파일을 가져온다.



dpkg -i percona-xtrabackup-80_8.0.28-21-1.buster_amd64.deb

apt install -f -y // 의존성 패키지 설치

dpkg -i percona-xtrabackup-80_8.0.28-21-1.buster_amd64.deb

```
root@80bf3def47bd:/usr/local/src# dpkg -i percona-xtrabackup-80_8.0.28-21-1.buster_amd64.deb
(Reading database ... 10725 files and directories currently installed.)
Preparing to unpack percona-xtrabackup-80_8.0.28-21-1.buster_amd64.deb ...
Unpacking percona-xtrabackup-80 (8.0.28-21-1.buster) over (8.0.28-21-1.buster) ...
Setting up percona-xtrabackup-80 (8.0.28-21-1.buster) ...
Processing triggers for libc-bin (2.28-10+deb10u1) ...
```

xtrabackup // 설치 확인

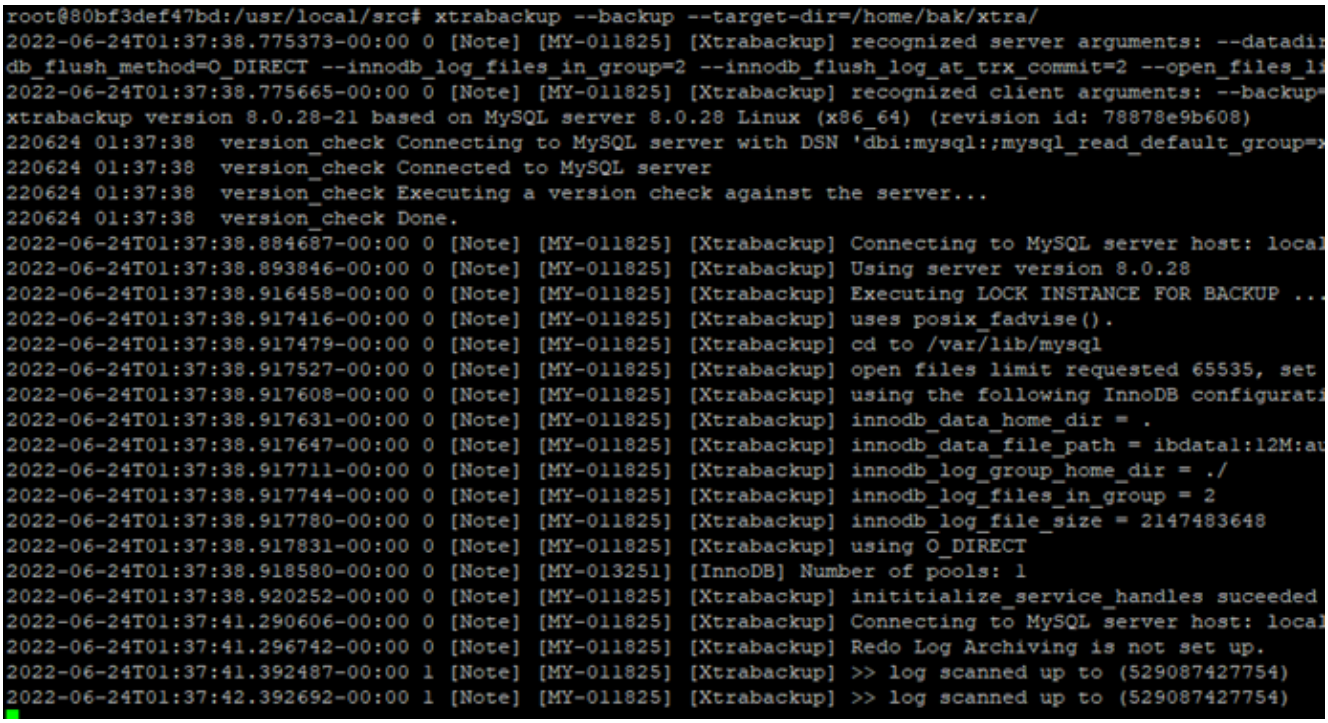
xtrabackup 을 이용한 백업 진행

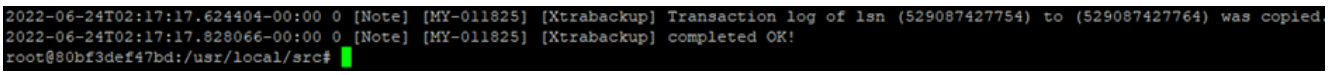
참고 사항

해당 가이드에선 Full backup을 대상으로 작성 하였습니다.

xtrabackup –backup –target-dir=/home/bak/xtra/

- 백업 진행
- 현재 매뉴얼에서는 가장 기본적인 백업 옵션만 넣어서 진행한다.





completed OK! 메세지 확인

xtrabackup을 이용한 복구

참고 사항

복구는 Host Machine 에서 진행한다.

exit // 컨테이너를 빠져나온다.

docker-compose down // 실행중이던 docker compose 종료

du -sh /home/data/var/lib/mysql // 삭제 전 데이터 파일 용량 확인

183G /home/data/var/lib/mysql

rm -rf /home/data/var/lib/mysql/ // 데이터 파일 전체 삭제

```
root@localhost:/home/data/var/lib# du -sh /home/data/var/lib/mysql
183G    /home/data/var/lib/mysql
root@localhost:/home/data/var/lib# rm -rf /home/data/var/lib/mysql/
root@localhost:/home/data/var/lib# du -sh /home/data/var/lib/mysql
du: cannot access '/home/data/var/lib/mysql': No such file or directory
root@localhost:/home/data/var/lib# mkdir /home/data/var/lib/mysql
```

xtrabackup –prepare –target-dir=/home/data/bak/xtra

- 백업파일 무결성 확인 (–prepare 옵션)

```
[Note] [MY-013072] [InnoDB] Starting shutdown...
[Note] [MY-013084] [InnoDB] Log background threads are being closed...
[Note] [MY-012980] [InnoDB] Shutdown completed; log sequence number 529087428118
[Note] [MY-011825] [Xtrabackup] completed OK!
```

[Xtrabackup] completed OK! 메시지 확인

xtrabackup –copy-back –target-dir=/home/data/bak/xtra/ –datadir=/home/data/var/lib/mysql/

- 복구 실행
- –copy-back 옵션은 백업 파일을 복사하여 복구하는 것 이며, –move-back 은 백업 파일을 복사하지 않고 이동시켜 복구를 진행한다.(–move-back 옵션은 원본 백업 파일은 삭제 된다.)

```
root@localhost:/home/data/var/lib# xtrabackup --copy-back --target-dir=/home/data/bak/xtra/ --datadir=/home/data/var/lib/mysql/
2022-06-27T11:30:46.968828+09:00 0 [Note] [MY-011825] [Xtrabackup] recognized server arguments: --datadir=/home/data/var/lib/mysql/
2022-06-27T11:30:46.969096+09:00 0 [Note] [MY-011825] [Xtrabackup] recognized client arguments: --copy-back=1 --target-dir=/home/data/bak/xtra/
xtrabackup version 8.0.28-21 based on MySQL server 8.0.28 Linux (x86_64) (revision id: 78878e9b608)
2022-06-27T11:30:46.969186+09:00 0 [Note] [MY-011825] [Xtrabackup] cd to /home/data/bak/xtra/
2022-06-27T11:30:46.969971+09:00 0 [Note] [MY-011825] [Xtrabackup] Copying undo_001 to /home/data/var/lib/mysql/undo_001
```

```
2022-06-27T15:45:18.761715+09:00 1 [Note] [MY-011825] [Xtrabackup] Done: Copying ./ibtmp1 to /home/data/var/lib/mysql/ibtmp1
2022-06-27T15:45:18.858858+09:00 0 [Note] [MY-011825] [Xtrabackup] completed OK!
```

[Xtrabackup] completed OK! // 메시지 확인

- 정상복구 확인 진행

docker-compose up -d // 컨테이너 기동

docker exec -it [Container ID] /bin/bash // 컨테이너 접근

du -sh /var/lib/mysql // 복구 후 용량 확인

183G /var/lib/mysql // 백업 전과 동일한 용량 확인

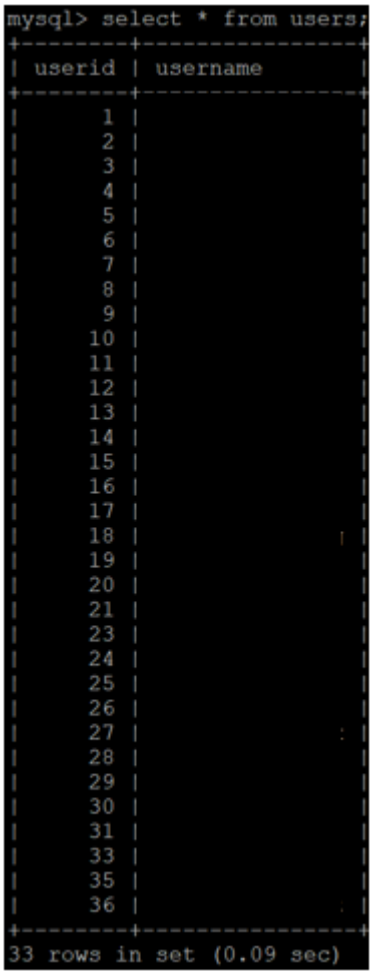
```
root@c8f98b39184d:/# du -sh /var/lib/mysql
183G    /var/lib/mysql
```

기존 데이터파일과 동일 용량 확인
mysql -u root -p // mysql 접근

show databases; // test db 확인

use zabbix;

select * from users; // 복구된 데이터 확인



삭제 전과 데이터가 동일한지 확인

참고 사항

MariaDB 10.1 이상 버전에선 Xtrabackup 대신 Mariabackup 을 사용하는 것 을 권장하며,
MariaDB 10.3 이상 버전부턴 Xtrabackup 자체가 지원 되지 않아, 반드시 Mariabackup 을 사용해야 한다.
– Mariabackup 은 암호화 , 압축된 테이블 및 기타 MariaDB 개선 사항과 함께 작동하도록 설계된 Xtrabackup의 포크 버전이다.
– 관련 URL : https://mariadb.com/kb/en/percona-xtrabackup-overview/#compatibility-with-m

---

PXE 튜토리얼

PXE 소개

Preboot execution environment 의 약자로써, 원격 네트워크 연결을 통한 OS 부팅을 가능하게 하는 표준

PXE 서버 사전 준비

Virtualbox 에서 테스트하며, Host-only Adapter 를 사용한다.

```
Virtualbox 실행 - File - Host Network Manager

IPv4 Address: 10.0.0.254/24
DHCP Server: None

CentOS 07 VM PXE:
  NIC 2EA 할당 (각 공인망/사설망)

CentOS 07 VM Client:
  NIC 1EA 할당 (사설망)
```

PXE 서버 네트워크 설정

```
# Public
cat << EOF | sudo tee /etc/sysconfig/network-scripts/ifcfg-
enp0s3
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
```

```
EOF

# Private
cat << EOF | sudo tee /etc/sysconfig/network-scripts/ifcfg-
enp0s8
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=no
NAME=enp0s8
DEVICE=enp0s8
ONBOOT=yes

IPADDR=10.0.0.11
PREFIX=24
EOF

sudo systemctl restart network
sudo systemctl status network
```

PXE 서버 DHCP 설치

```
sudo yum install -y dhcp

sudo mkdir /etc/dhcp/dhcpd.d/

cat << EOF | sudo tee /etc/dhcp/dhcpd.conf
option domain-name-servers     10.0.0.11;
## 10.0.0.11 -> PXE 서버 사설 IP

default-lease-time 600;
max-lease-time 7200;

authoritative;
## DHCPNAK send enable

filename        "pxelinux.0";
next-server     10.0.0.11;
## tftp 서버 IP

subnet 10.0.0.0 netmask 255.255.255.0 {
    range dynamic-bootp 10.0.0.200 10.0.0.253;
```

```
    option broadcast-address 10.0.0.255;

    # option routers 10.0.0.11;
    ## Client 가 자동 할당받는 Gateway IP
  ## 해당 예제에서는 사설망이므로 주석 처리
}

EOF

sudo systemctl enable --now dhcpd
sudo systemctl status dhcpd
```

PXE 서버 TFTP 설치

리눅스 부팅에 필요한 파일을 전송하기 위한 tftp 설치

```
sudo yum install -y syslinux tftp tftp-server
sudo cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

sudo systemctl enable --now tftp
sudo systemctl status tftp

tftp
(to) 10.0.0.11
tftp> verbose
tftp> trace
tftp> get pxelinux.0
```

PXE 서버 ISO 부팅 설정

```
wget
http://mirror.navercorp.com/centos/7/isos/x86_64/CentOS-7-x86_
64-Minimal-2009.iso

sudo mkdir /var/lib/tftpboot/centos7

sudo mount CentOS-7-x86_64-Minimal-2009.iso /mnt
## sudo mount -t iso9660 -o loop CentOS-7-x86_64-
Minimal-2009.iso /mnt

sudo cp /mnt/images/pxeboot/vmlinuz /var/lib/tftpboot/centos7/
```

```
sudo cp /mnt/images/pxeboot/initrd.img
/var/lib/tftpboot/centos7/
sudo cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot/

sudo mkdir /var/lib/tftpboot/pxelinux.cfg/

cat << EOF | sudo tee /var/lib/tftpboot/pxelinux.cfg/default
timeout 100
default menu.c32

menu title ######### PXE Boot Menu #########
label 1
   menu label ^1) Install CentOS 7
   kernel centos7/vmlinuz
                append     initrd=centos7/initrd.img
method=http://10.0.0.11/iso devfs=nomount

label 2
   menu label ^2) Boot from local drive
   localboot
EOF

sudo yum install -y httpd

cat << EOF | sudo tee /etc/httpd/conf.d/pxeboot.conf
Alias /iso /mnt
<Directory /mnt>
    Require ip 127.0.0.1 10.0.0.0/24
</Directory>
EOF

sudo systemctl enable --now httpd
sudo systemctl disable --now firewalld
```

PXE 서버 ISO 부팅 테스트

Virtualbox - VM - Settings

첫번째 NIC 은 dhcpd 서버에서 사용하는 NIC 과 동일한 대역의 장비로 설정한다.
(첫번째 NIC 이 다른 대역의 장비인 경우 미동작한다.)

VM Start - 단축키 F12 - Boot Manager - l) LAN (단축키 L 입력)

CentOS 설치화면을 확인할 수 있다.

PXE 서버 결론

PXE 서버와 다른 PC 의 네트워크 부팅을 통해 CentOS 설치를 가능하게 해보았다. 이외에도 initrd(=램디스크) 의 옵션에 따라 CentOS 자동 설치(kickstart)도 가능하고 nfs 에서 가져온 PXE 서버의 루트 파일시스템을 통해 원격 부팅도 가능하다.

TCPDUMP 기본 사용법 #01

TCPDUMP 사용법 및 예제

TCPDUMP 란?

명령줄 인터페이스에서 실행하는 네트워크 인터페이스의 패킷을 가로채는 (스니핑) 프로그램이다
TCP/IP 뿐만이 아닌 다른 여러 패킷의 정보를 출력할 수 있다.

기본 사용법

패킷을 수집하여 파일로 저장한 후 Wireshark 등의 패킷 분석 프로그램을 사용한다.
수많은 부가 옵션이 있으나 주로 쓰이는 몇 가지 옵션만 기술하였다.

패킷 수집 패턴 커맨드 (Basic / Expression / Header Flag)

Basic (필수 옵션)

기본적인 옵션값

-i : 네트워크 인터페이스가 여러 개일 경우 지정할 수 있다
-c : 스니핑 할 패킷의 개수를 정할 수 있다.

-v : 패킷 헤더의 정보를 조금 더 자세하게 보여준다 -vv 하나 더 붙이면 더욱 자세해진다
-n : 알려진 도메인 문자열을 IP 로 보이게 해준다. -nn 하나 더 붙이면 Port 도 숫자로 표시한다.
-w : 스니핑 한 패킷을 .pcap 타입 바이너리로 저장한다. -r 옵션으로 TXT 타입으로 로드할 수 있다.

 # 특정 인터페이스(-i) 의 모든 패킷을 100개만(-c) .pcap 파일 형식으로 저장(-w) 하기
해당 형식은 txt 가 아닌 Binary 타입이며 -r 옵션으로 읽거나 wireshark 에서 로드할 수 있다.

```
tcpdump  -i ens33 -vv -c 100 -w ttt.pcap
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
100 packets captured
100 packets received by filter
0 packets dropped by kernel
```

 # 특정 인터페이스(-i) 의 모든 패킷을 5개만(-c) DNS/PORT 번역하지 않고 IP 그대로 출력(-nn) 한다.

```
tcpdump  -i br0 -n -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on br0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:08:14.531606  ARP,  Request  who-has  8.8.8.8  tell 10.10.10.200, length 28
14:08:15.533556  ARP,  Request  who-has  8.8.8.8  tell 10.10.10.200, length 28
14:08:16.535615  ARP,  Request  who-has  8.8.8.8  tell 10.10.10.200, length 28
14:08:18.531679  ARP,  Request  who-has  8.8.8.8  tell 10.10.10.200, length 28
14:08:19.533632  ARP,  Request  who-has  8.8.8.8  tell 10.10.10.200, length 28
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

 # 더 자세한 패킷 정보를 출력 (-v , -vv) 한다.

```
tcpdump  -i br0 -n -vv -c 5
tcpdump: listening on br0, link-type EN10MB (Ethernet),
```

```
capture size 262144 bytes
14:10:46.548613 ARP, Ethernet (len 6), IPv4 (len 4), Request
who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:47.549640 ARP, Ethernet (len 6), IPv4 (len 4), Request
who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:48.551557 ARP, Ethernet (len 6), IPv4 (len 4), Request
who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:50.548614 ARP, Ethernet (len 6), IPv4 (len 4), Request
who-has 8.8.8.8 tell 10.10.10.200, length 28
14:10:51.549621 ARP, Ethernet (len 6), IPv4 (len 4), Request
who-has 8.8.8.8 tell 10.10.10.200, length 28
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

 # 특정 인터페이스에서(-i) 특정 IP 가 포함된 패킷(-nn, grep) 을 보여준다

```
tcpdump -i ens33 -nn | grep "1.2.3.4"
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size
262144 bytes
5 packets captured
6 packets received by filter
0 packets dropped by kernel
14:28:31.524052 IP 1.2.3.4.22 > 5.6.7.8.53707: Flags [P.], seq
4138122029:4138122237, ack 1068364849, win 1432, length 208
14:28:31.525940 IP 5.6.7.8.53707 > 1.2.3.4.22: Flags [.], ack
208, win 509, length 0
```

Expression (조건식)

조건에 부합하는 패킷 정보만을 보여준다. 조건의 가장 큰 구분은 다음과 같다. 조합할 수 있다.

1) Type (패킷의 종류 조건)
- host , net , port

2) Dir (패킷의 방향 조건)
- src, dst , src or dst, src and dst

3) Proto (패킷의 프토토콜 타입 조건)

- ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp, udp

4) 그 외
- 산술 연산자 ('and(&&)', 'or(||)', 'not(!)')
- gateway, broadcast, less, greater

 # 특정 인터페이스(-i) 에서 tcp 패킷이며 Destination Port 가 22인 패킷을 IP 로(-nn) 출력하기
tcpdump -i ens33 tcp dst port 22 -nn
```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
16:50:25.964928 IP 211.115.223.215.53707 > 211.239.150.48.22: Flags [.], ack 4194846013, win 511, length 0
16:50:26.016014 IP 211.115.223.215.53707 > 211.239.150.48.22: Flags [.], ack 161, win 511, length 0
16:50:26.060903 IP 211.115.223.215.53707 > 211.239.150.48.22: Flags [.], ack 321, win 510, length 0
```

 # 특정 인터페이스(-i) 에서 tcp 패킷"이며" Destination Port 가 80 "이거나" 443 인 패킷을 IP 로(-nn) 출력하기
tcpdump -i ens33 -nn tcp and dst port 80 or dst port 443
```
16:59:12.840778 IP 211.239.150.48.49514 > 8.8.8.8.443: Flags [F.], seq 5, ack 9, win 229, options [nop,nop,TS val 3924566459 ecr 3515313672], length 0
16:59:14.863982 IP 211.239.150.48.47286 > 8.8.8.8.80: Flags [S], seq 873533324, win 29200, options [mss 1460,sackOK,TS val 3924568482 ecr 0,nop,wscale 7], length 0
16:59:15.865581 IP 211.239.150.48.47286 > 8.8.8.8.80: Flags [S], seq 873533324, win 29200, options [mss 1460,sackOK,TS val 3924569484 ecr 0,nop,wscale 7], length 0
```

Header Flag

패킷 해더의 정의된 플래그 타입을 조건으로 한 결과값을 출력할 수 있다.

 SYN : Client 가 데이터를 보내기 전 Server 에 악수(동기)를 요청하는 가장 첫 단계
이 플래그를 통해 IN/OUT Port 통신을 체크할 수 있어서 자주 활용된다.

TCP Header

Interesting Parts	TCP header
tcp[0:2]	source port
tcp[2:2]	destination port
tcp[4:4]	sequence number
tcp[8:4]	acknowledgement number
tcp[12]	header length
tcp[13]	tcp flags
tcp[14:2]	window size
tcp[16:2]	checksum
tcp[18:2]	urgent pointer
tcp[20..60]	options or data

UDP Header

Interesting Parts	UDP Header
udp[0:2]	source port
udp[2:2]	destination port
udp[4:2]	datagram length
udp[6:2]	UDP checksum

Flags

Flags	Numerically	Meaning
---- --S- 0000 0010	0x02	normal syn
---A --S- 0001 0010	0x12	normal syn-ack
---A ---- 0001 0000	0x10	normal ack
--UA P--- 0011 1000	0x38	psh-urg-ack.
---A -R-- 0001 0100	0x14	rst-ack
---- --SF 0000 0011	0x03	syn-fin scan
--U- P--F 0010 1001	0x29	urg-psh-fin
-Y-- ---- 0100 0000	0x40	anything >= 0x40
XY-- ---- 1100 0000	0xC0	both reserved bits set
XYUA PRSF 1111 1111	0xFF	FULL_XMAS scan

 # SYN 패킷만 수집 하기

= TCP Flag(tcp[13]) 중에 SYN (0x02) 인 IP (-nn) 패킷 10개를(-c) txt 타입 result.log 로 저장(>) 하기

tcpdump -i ens33 -c 10 "tcp[13] == 0x02" -n -nn > result.log

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

 # cat resule.log

```
16:16:15.308872 IP 211.239.150.48.43192 > 61.100.13.230.10051: Flags [S], seq 3596242163, win 29200, options [mss 1460,sackOK,TS val 3921988927 ecr 0,nop,wscale 7], length 0
16:16:17.048128 IP 121.40.192.14.41265 > 211.239.150.48.4244: Flags [S], seq 3887030721, win 1024, length 0
```

[S] = SYN Flag

수집한 부분을 확인할 수 있으며

json 정렬이나, 조건식을 통해 다듬을 수도 있다.

```
Unskilled Attackers Pester Real Security Folks
==============================================
                     TCPDUMP FLAGS
Unskilled =  URG  =  (Not Displayed in Flag Field, Displayed elsewhere)
Attackers =  ACK  =  (Not Displayed in Flag Field, Displayed elsewhere)
Pester    =  PSH  =  [P] (Push Data)
Real      =  RST  =  [R] (Reset Connection)
Security  =  SYN  =  [S] (Start Connection)
Folks     =  FIN  =  [F] (Finish Connection)
          SYN-ACK =  [S.] (SynAcK Packet)
                     [.] (No Flag Set)
```

 # JSON 을 사용한 덤프파일 정렬 예시

result.log

서버 IP: 211.239.150.48

JSON:

```
[ { ipaddr1: '221.239.150.48',
    ipaddr2: '61.100.13.230',
    type: 'Outbound',
    port: '10051' },
  { ipaddr1: '121.40.192.14',
    ipaddr2: '221.239.150.48',
    type: 'Inbound',
    port: '4244' } ]

Table:
ipaddr1         ipaddr2         type      port
--------------  --------------  --------  -----
221.239.150.48  61.100.13.230   Outbound  10051
121.40.192.14   211.239.150.48  Inbound   4244
```

참고 자료

TCP 3 way handshark : https://websecurity.tistory.com/93

원격으로 CentOS 7 에서 내부 Windows 서버에 RDP 연결

CentOS 7 에서 Windows RDP 연결하기

진행 순서

```
# OS
CentOS 7.9 x86_64 minimal
로컬--> XRDP 를 통한 리눅스 GUI 접속 --> 리눅스 GUI 에서 내부 Windows RDP 접속
```

Linux GUI 환경 세팅

```
# GUI GroupInstall
root@localhost ~]# yum groups list | grep -i desktop
   Cinnamon Desktop
   MATE Desktop
   GNOME Desktop
   General Purpose Desktop
   LXQt Desktop
# GNOME 이 제대로 설치되지 않는다면 "Server with GUI" 그룹을 설치한다.
root@localhost ~] yum groupinstall "GNOME Desktop"

# GUI init
[root@localhost ~]# systemctl get-default
multi-user.target
[root@localhost ~]# systemctl set-default graphical.target
[root@localhost ~]# systemctl get-default
graphical.target
# Reboot 후 자동으로 GUI 모드로 부팅되는지 확인한다.
[root@localhost ~]# reboot
```

Linux 서버에 원격 접속 설정

```
# XRDP Install.
[root@localhost ~]# yum install epel-release
[root@localhost ~]# yum install xrdp
[root@localhost ~]# systemctl enable xrdp && systemctl start
xrdp

# selinux disable 및 iptables -F or  방화벽 정책에 tcp/3389 를 추가
해준다.
```

rdesktop 설치

```
# 컴파일러와 openssl-devel 이 패키지 컴파일 선행조건이다.
 yum -y install gcc openssl-devel
                                                          wget
https://github.com/rdesktop/rdesktop/releases/download/v1.8.6/
rdesktop-1.8.6.tar.gz
 tar xvzf rdesktop-1.8.6.tar.gz
 cd rdesktop-1.8.6/
 ./configure --disable-credssp --disable-smartcard
```

```
make
make install
```

Check

```
# 리눅스 서버에 RDP 로 접속 후, rdesktop -u [User] [ip]  입력한다.
root@localhost ~]# rdesktop -u administrator 10.10.10.5
Autoselected keyboard map en-us
Connection established using SSL.
WARNING: Remote desktop does not support colour depth 24;
falling back to 16
```