

Ansible 자동화로 구현한 클라우드 환경 DevOps

- 통합 문제 풀이 -

제출일	2020. 12. 31
훈련기관	(주)솔데스크
강사님	김기연 강사님
조원	1조 최시울, 김민영, 황유지, 허윤하, 김선경

목 차

1. 개요
 - 1.1. 개요
 - 1.2. 소프트웨어 환경
 - 1.3. 보고서 포맷
2. OS 부팅 과정
 - 2.1. 리눅스 부팅 과정
 - 2.2. 설치 소스 부팅 옵션
3. SWAP
 - 3.1. SWAP 공간
 - 3.2. SWAP 추가 및 확인
 - 3.3. SWAP 삭제 및 확인

개 요

이 보고서는 K-Digital 수업 LINUX I, II 파트를 마치며 그룹별로 주어진 문제에 대해 고민해 보고 답을 작성해 정리한 보고서입니다.

저희 1조는 공통 주제인 OS 부팅, SWAP 과정과 원격 또는 로컬 접속 시 환경 파일에 대한 정리까지 작성해보았습니다. OS부팅 과정 부분에서는 OS 중 리눅스 RHEL 7&8의 부팅 과정과 SWAP 파티션에 대해 설명합니다. SWAP 파티션의 정의와 개념을 서술하고 실무에서 어떻게 사용되는지 실습을 통해 학습할 수 있습니다.

SWAP의 정의와 필요한 이유에 대해 설명하고 실습을 통해 SWAP을 추가, 제거하는 방법에 대해 알 수 있습니다.

소프트웨어 환경 및 버전

소프트웨어	설명
LINUX	CentOS 8.3
VM	VMware Workstation 15.5.6

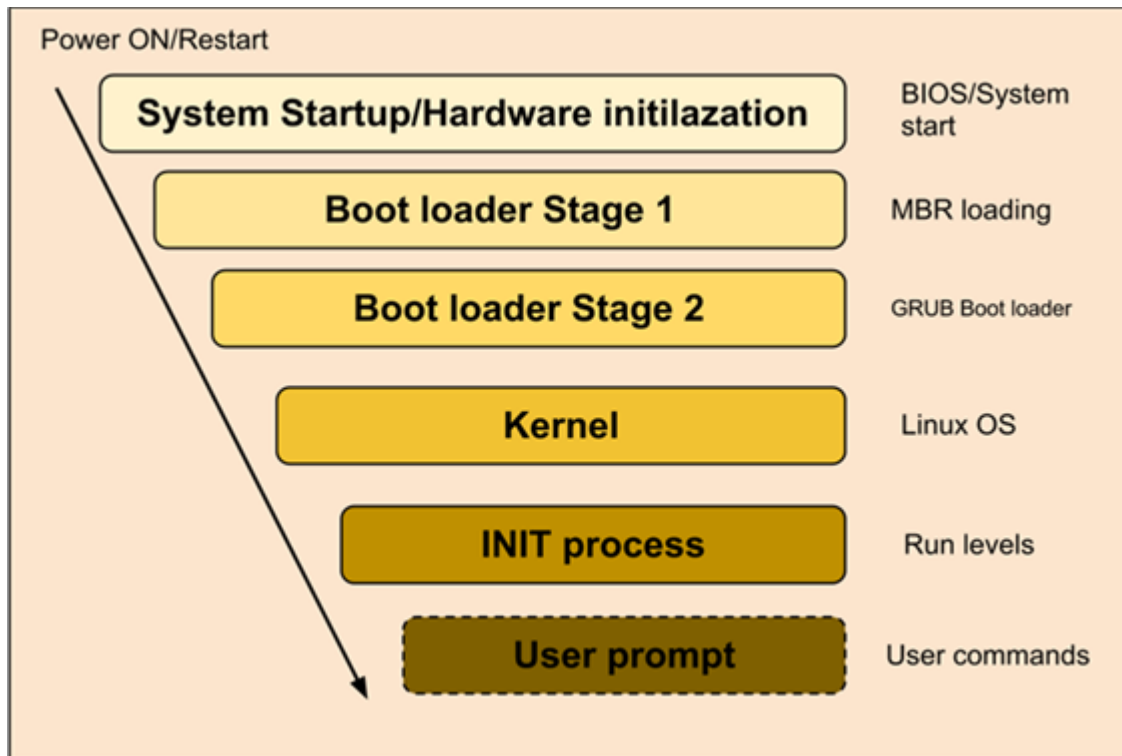
보고서 포맷

전체 보고서 : 맑은 고딕 10pt, 줄 간격 150%

셀 작성 : 굴림체 8pt, 표 안에 작성

OS 부팅 과정

리눅스 부팅 과정



· 0 단계 : 전원 스위치 ON 단계

- 메인보드의 ROM-BIOS에 있는 BIOS 프로그램을 자동 실행합니다.
- BIOS 프로그램은 전원공급과 함께 메모리의 특정 번지(예:FFFF0H)에 자동 로드됩니다.
- CPU는 전원공급과 함께 특정 번지(예:FFFF0H)의 BIOS 프로그램(명령들)을 자동 실행 합니다.

· 1단계 : BIOS 단계



그림 8-3 바이오스 단계의 세부 동작

- 자체 진단 기능(POST(Power On Self Test))
- CMOS 검사, CPU, MEMORY, 그래픽카드, 키보드, 마우스 등 각종 장치들의 이상 유무를 검사하고 이들 장치(하드웨어들)을 초기화시킵니다.
- 부팅 매체 검색과 부트로더 실행합니다.
- POST 과정이 이상 없이 진행 완료되면 검색된 부팅 매체(하드디스크, CD-ROM, 플로피 디스크 등)에서 부트로더(예:GRUB, LILO)를 불러들입니다. 예를 들어, 하드디스크가 부팅 매체로 선택되었다면 하드디스크의 부팅 파티션에 있는 0번 섹터에 있는 부트로더(Boot Loader, 즉, GRUB)을 읽어 들이게 됩니다.
- 부트로더(GRUB)가 메모리에 적재되면 BIOS는 종료되고, 시스템 제어권은 부트로더(GRUB)이 갖게 됩니다.

· 2 단계 : 부트로더(GRUB/LILO) 단계

부트 로더는 일반적으로 여러 운영체제 중에서 부팅할 운영체제를 선택할 수 있도록 메뉴를 제공합니다. 리눅스의 대표적 부트로더는 GRUB와 LILO입니다.

- 만약 부팅 로더로 GRUB를 사용하고 있다면, GRUB은 실행과 함께 /boot/grub/grub.conf 파일을 읽어서 어떤 부팅 메뉴(커널)로 부팅을 할 것인가를 결정하게 됩니다. 자동 결정 또는 사용자 선택) 이 화면이 GRUB이 화면(파란화면)에 나타나는 첫 번째 화면입니다.
- GRUB은 커널(kernel) 이미지를 불러들이고 시스템 제어권을 커널에게 넘겨줍니다.

· 3 단계 : 커널 단계

- 커널은 swapper 프로세스(PID 0번)를 호출합니다.
- swapper는 커널이 사용할 각 장치 드라이브들을 초기화하고 init 프로세스(PID 1번)를 실행하게 됩니다.
- init 프로세스가 실행되면서 /etc/inittab 파일을 읽어들이어서 그 내용들을 차례대로 실행합니다.

· 4 단계 : init 프로세스 단계

init 실행 단계에서는 다양한 서비스를 동작시킵니다. 각 서비스가 시작하는 과정이 화면에 메시지로 출력됩니다.

```
[ 0.004000] Spectre V2 : Spectre mitigation: LFENCE not serializing, switchin
g to generic retpoline
[ 1.597182] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
[ 2.173085] sd 32:0:0:0: [sda] Assuming drive cache: write through
[ 2.173909] sd 32:0:1:0: [sdb] Assuming drive cache: write through
[ 2.175401] sd 32:0:2:0: [sdc] Assuming drive cache: write through
[ 2.176375] sd 32:0:3:0: [sdd] Assuming drive cache: write through
/dev/sda1: clean, 183785/1310720 files, 2025607/5242368 blocks
```

다음 사진은 부팅 후 dmesg 명령이나 more /var/log/boot.log 명령으로 확인할 수 있습니다.

```
yanju@ubuntu:~$ dmesg | more
[ 0.000000] Linux version 4.18.0-25-generic (build@lgw01-amd64-033) (gcc ver
sion 7.4.0 (Ubuntu 7.4.0-1ubuntu1~18.04.1)) #26~18.04.1-Ubuntu SMP Thu Jun 27 07
:28:31 UTC 2019 (Ubuntu 4.18.0-25.26~18.04.1-generic 4.18.20)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.18.0-25-generic root=UUI
D=ec549f21-01e4-447b-8da5-3b427913535b ro find_preseed=/preseed.cfg auto nopromp
t priority=critical locale=en_US quiet
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Centaur CentaurHauls
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regl
sters'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes,
using 'compact' format.
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000009e7ff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000009e800-0x00000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000dc000-0x0000000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000100000-0x00000000000007fedffff] usable
```

설치 소스 부팅 옵션

inst.repo=

이 부팅 옵션은 설치 소스를 지정해 줍니다. **inst.repo=cdrom**. **inst.repo=** 이 옵션의 대상은 다음 매체 중 하나 여야 합니다.

- (1) .treeinfo 파일을 포함하는 디렉토리 구조인 설치 가능한 트리
- (2) DVD (물리적 디스크)
- (3) 하드 드라이브 또는 시스템에 액세스 할 수 있는 ISO image

다음은 inst.repo= 설치 소스 부팅 옵션에 대해 표로 설명합니다.

소스 유형	부팅 옵션 방식	소스 형식
CD / DVD 드라이브	inst.repo=cdrom:<device>	물리 디스크로 설치
설치 가능한 트리	inst.repo=hd:<device>:/<path>	디렉토리 및 파일의 전체 사본 이 있는 DVD 의 image 파일
NFS 서버	inst.repo=nfs:[option:]<server>/<path>	디렉토리 및 파일의 전체 사본 이 있는 DVD 의 image 파일
HTTP 서버	inst.repo=http://<host>/<path>	디렉토리 및 파일의 전체 사본 이 있는 DVD 의 image 파일
HTTPS 서버	inst.repo=https://<host>/<path>	
FTP 서버	inst.repo=ftp://<username>:<password>@<host>/<path>	
HMC	inst.repo=hmc	

다음 형식으로 디스크 장치 이름을 설정 합니다.

- 커널 장치 이름 (예 /dev/sda1)
- 파일 시스템 레이블의 예 : LABEL=FLASH 또는 LABEL=RHEL8
- 파일 시스템 UUID의 예 : UUID=8176c7bf-04ff-403a-a832-9557f94e61db

inst.addrepo=

주 저장소와 함께 또 다른 설치 소스로 사용할 수 있는 추가 저장소를 추가할 수 있습니다. 이 부팅 옵션은 한 번의 부팅 중에 여러번 사용할 수 있습니다.

다음은 inst.addrepo 부팅 옵션에 대한 설명입니다.

설치 유형	부팅 옵션 형식	추가 설명
URL에 설치 가능한 트리	inst.addrepo=REPO_NAME, [http, https,ftp]://<host>/ <path>	주어진 URL에서 설치 가능한 트리를 찾습니다.
NFS 경로에서 설치 가능한 트리	inst.addrepo=REPO_NAME,n fs://<server>:/<path>	주어진 NFS 경로에서 설치 가능한 트리를 찾습니다. 호스트 뒤에는 콜론이 필요합니다.
설치 환경	inst.addrepo=REPO_NAME,fi le://<path>	설치 환경의 지정된 위치에서 설치 가능한 트리를 찾습니다. 이 옵션을 사용하려면 설치 프로그램이 사용가능한 소프트웨어 그룹을 로드하기전에 저장소를 미리 마운트 해야 합니다.
하드 드라이브	inst.addrepo=REPO_NAME,h d=:<device>:<path>	<path>에 지정된 <device> 파티션을 마운트 합니다. 경로를 지정하지 않으면 설치 프로그램은 <device>에서 유효한 ISO를 찾습니다.

S W A P

SWAP 공간

Linux의 스왑 공간은 실제 메모리(RAM)의 용량이 거의 찼을 때 사용됩니다. 시스템에 더 많은 메모리 리소스가 필요하고 RAM이 가득 차면 메모리의 비활성화된 페이지가 스왑 공간으로 이동하게 됩니다. 스왑 공간은 RAM이 적은 시스템에서 도움이 될 수 있지만 RAM을 대체할 정도로 많은 스왑 공간을 만드는 것은 좋지 않습니다. 스왑 공간은 물리적 메모리보다 액세스 시간이 느린 하드 드라이브에 있습니다.

스왑 공간 크기에 대한 권장사항은 다음과 같습니다.

시스템의 RAM 양	권장 스왑 공간	최대 절전 모드를 허용하는 경우 권장되는 스왑 공간
≤ 2GB	RAM 용량의 2배	RAM 용량의 3배
> 2GB - 8GB	RAM 용량과 동일	RAM 용량의 2배
> 8GB - 64GB	4GB 이상	RAM 용량의 1.5배
> 64GB	4GB 이상	최대 절전 모드는 권장되지 않습니다.

위 표를 통해 스왑 공간 및 최대 절전모드 지원과 관련하여 용량 설정에 도움을 줍니다.

스왑 공간을 여러 저장 장치에 분산하면 특히 빠른 드라이브, 컨트롤러 및 인터페이스가 있는 시스템에서 스왑 성능이 향상됩니다.

SWAP 추가 및 확인

이 부분에서는 더 많은 스왑 공간을 추가하는 방법에 대해 설명합니다.

· 스왑 파티션 만들기

1. /dev/sdb 디스크가 사용 중인지 확인합니다.

```
# df -h
```

2. /dev/sdb 디스크의 system partition id를 변경합니다.

```
# fdisk /dev/sdb
Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 82
```

3. 새 스왑 공간을 포맷 합니다.

```
# mkswap /dev/sdb1
```

4. SWAP 활성화합니다.

```
# swapon /dev/sdb1
```

5. 스왑이 성공적으로 확장되고 활성화되었는지 테스트하기 위해서 스왑 공간을 검사합니다.

```
# cat /proc/swaps
# free
```

· 스왑 파일 만들기 순서

1. 새 스왑 파일의 크기를 정하고 1024를 곱하여 블록 수를 결정합니다. 예를 들 64MB 스왑 파일의 블록 크기는 65536입니다.

2. 빈 파일을 만듭니다.

```
# dd if = /dev/zero of=/swapfile bs = 1024 count = 65536
```

3. 다음 명령을 사용하여 스왑 파일을 설정합니다.

```
# mkswap /swap/swapfile
```

4. 스왑 파일의 보안을 변경하여 다른 유저가 읽을 수 없도록 합니다.

```
# chmod 0600 /swap/swapfile
```

5. 부팅 시 스왑 파일을 활성화하려면 /etc/fstab에 아래 항목을 포함하여 편집합니다.

```
# vi /etc/fstab

/swap/swapfile swap swap defaults 0 0
```

6. 시스템이 새 /etc/fstab 구성을 등록하도록 마운트 장치를 재생성합니다.

```
# systemctl daemon-reload
```

7. 스왑파일을 즉시 활성화하려면 아래 항목을 입력합니다.

```
# swapon /swap/swapfile
```

8. 새 스왑 파일이 성공적으로 생성되고 활성화되었는지 확인하기 위해 스왑 공간을 검사합니다.

```
# cat /proc/swaps
# free
```

SWAP 삭제 및 확인

이 부분에서는 스왑 공간을 줄이는 방법에 대해 설명합니다.

· 스왑 파티션 삭제

1. 파티션들의 마운트를 해제하신 후 연결된 스와핑을 비활성화합니다.

```
# swapoff /dev/sdb1
```

2. /etc/fstab 파일에서 삭제할 스왑 파티션 항목을 삭제합니다.

```
# vi /etc/fstab
```

3. 스왑이 성공적으로 삭제되었는지 확인하기 위해 스왑 공간을 검사합니다.

```
# cat /proc/swaps
# free
```

스왑 파일 삭제

1. 다음 명령을 실행하여 스왑파일을 비활성화합니다.

```
# swapoff /swap/swapfile
```

2. /etc/fstab/ 파일에서 해당하는 항목을 제거합니다.

```
# vi /etc/fstab  
/swap/swapfile swap swap defaults 0 0
```

3. 실제 파일을 제거합니다.

```
# rm -rf /swap/*
```

결론

마무리

OS 중 리눅스 부팅과정에 대해 차례대로 알아 보았고 설치 소스 부팅옵션을 통해 디스크 등 다른 매체를 등록할 수 있습니다. 또한, SWAP의 파일 또는 파티션에 따라 다른 생성 방식에 대해 습득하고 활용할 수 있습니다.

참조 문서

- 리눅스 부팅과정 참고 문서 :

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/performing_a_standard_rhel_installation/getting-started_installing-rhel

https://www.linux.co.kr/lecture/lec_linux_01/lec-data/08data.pdf

- SWAP 참고 문서 :

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_storage_devices/getting-started-with-swap_managing-storage-devices#swap-space_getting-started-with-swap