

Ansible 자동화로 구현한 클라우드 환 경 DevOps

통합 보고서

훈련기관 : (주)솔데스크

강사님 : 김기연 강사님

1조 : 김선경, 김민영, 최시율, 허윤하, 황유지

목차

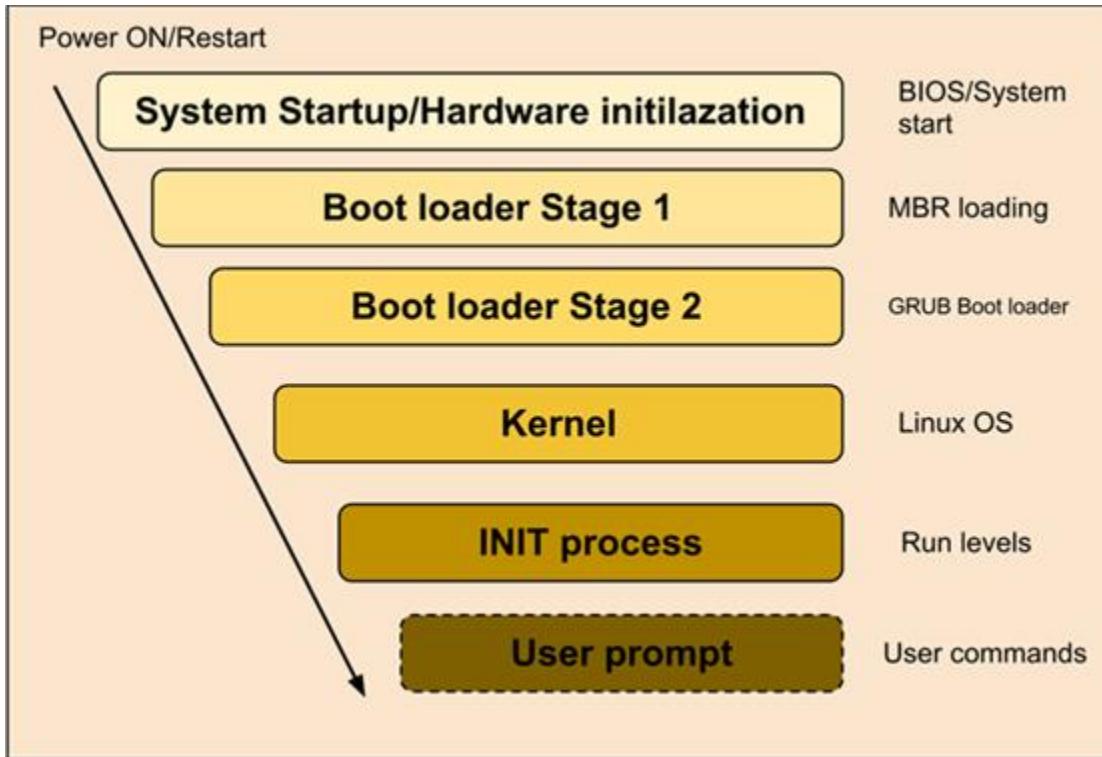
1. OS 부팅 과정

- I. 리눅스 부팅 과정
- II. 설치 소스 부팅 옵션

1. SWAP

- I. SWAP 공간
- II. SWAP 추가 및 확인
- III. SWAP 삭제 및 확인

리눅스 부팅 과정



리눅스 부팅 과정

● 0 단계 : 전원 스위치 ON 단계

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

● 1단계 : BIOS 단계

- 자체 진단 기능(POST(Power On Self Test))
- CMOS 검사, CPU, MEMORY, 그래픽카드, 키보드, 마우스 등 각종 장치들의 이상 유무를 검사하고 이들 장치(하드웨어들)를 초기화
- 부팅 매체 검색과 부트로더 실행
- POST 과정이 이상 없이 진행 완료되면 검색된 부팅 매체에서 부트로더를 로드
- 부트로더(GRUB)가 메모리에 적재되면 BIOS는 종료되고, 시스템 제어권은 부트로더(GRUB)이 갖게 됨

리눅스 부팅 과정

● 2 단계 : 부트로더 단계

- 부트 로더는 일반적으로 여러 운영체제 중에서 부팅할 운영체제를 선택할 수 있도록 메뉴를 제공 리눅스의 대표적 부트로더는 GRUB와 LILO 입
- GRUB은 커널(kernel) 이미지를 불러들이고 시스템 제어권을 커널에게 넘겨줌

● 3 단계 : 커널 단계

- 커널은 swapper 프로세스(PID 0번)를 호출
- swapper는 커널이 사용할 각 장치 드라이브들을 초기화하고 init 프로세스(PID 1번)를 실행
- init 프로세스가 실행되면서 /etc/inittab 파일을 읽어 들여서 그 내용들을 차례대로 실행



그림 8-3 바이오스 단계의 세부 동작

리눅스 부팅 과정

● 4 단계 : init 프로세스 단계

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

```
[ 0.004000] Spectre VZ : Spectre mitigation: LFENCE not serializing, switching 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
```

S W A P 공간

- 스왑 공간은 실제 메모리(RAM)의 용량이 차기 전에 사용한다.
- RAM이 적은 시스템에서는 도움이 될 수 있지만 RAM을 대체할 정도로 많은 스왑 공간은 만들지 않는다.
- 하드 드라이브에 위치에 있다.

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.

S W A P 추가

SWAP 파티션 추가

```
# df -h
```

/dev/sdb 디스크가 사용 중인지 확인

```
# fdisk /dev/sdb
```

/dev/sdb 디스크의 시스템 파티션 id 변경

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

새 스왑 공간 포맷

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

swap 활성화

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

swap이 성공적으로 추가 되었는지 확인

```
# free
```

SWAP 추가

SWAP 파일 추가

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

새 스왑 파일의 크기를 정하고 1024를 곱하여 블록 수 결정

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

스왑파일 생성

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

스왑파일 허가 권한 변경

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

부팅 시 스왑 파일 활성화

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

시스템이 /etc/fstab/의 변경된 내용을 등록하도록 마운트

```
# systemctl daemon-reload
```

트 장치 재생성

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

스왑 파일 즉시 활성화

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

확인을 위한 스왑 공간 검사

S W A P 삭제

SWAP 파티션 삭제

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

스왑 비활성화

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

/etc/fstab 파일에서 스왑 파티션 항목 삭제

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

swap이 성공적으로 삭제 되었는지 확인

```
# free
```

S W A P 삭제

SWAP 파일삭제

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

스왑 파일 비활성화

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

/etc/fstab에서 해당하는 항목 제거

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

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

실제 파일 제거