

## DevOps for SpringBoot

단계	주제	내용
[1]	리눅스 기초 I (명령어와 파일 시스템)	- 리눅스 디렉토리 구조 이해- 기본 명령어 (ls, cd, mkdir, rm, cp, mv, cat, chmod 등)- 사용자 및 퍼미션 이해- 실습: 파일 조작, 사용자 추가, 권한 변경
[2]	리눅스 기초 II (WSL2 네트워크 실습)	- 네트워크 명령어: ping, curl, ss, netstat, ip- 패키지 설치: apt update, apt install- 프로세스/포트 확인: ps aux, top, ss -tuln- 실습: 네트워크 점검, 패키지 관리, 실행 중인 프로세스 추적
[3]	Docker & Spring Boot 컨테이너화	- Docker 기본 구조 및 명령어 실습- Dockerfile 작성- Spring Boot 컨테이너화- Docker Compose 로 DB 연동 실습
[4]	Jenkins 설치 및 CI 구성	- Docker 로 Jenkins 설치- Jenkins 초기 설정 및 Git 연동- Jenkins Job 생성 및 빌드 자동화- Jenkinsfile 작성: Build + Test
[5]	Kubernetes 설치 및 수동 배포	- Minikube 설치 및 kubectl 설정- Kubernetes 개념 (Pod, Deployment, Service)- Spring Boot 앱 수동 배포 (YAML 작성)- 실습: 서비스 접속 및 로깅 확인
[6]	Jenkins → Kubernetes 자동 배포 (CD)	- Jenkins 에 kubectl 연동 (kubeconfig)- Jenkinsfile 수정: Build → Deploy 자동화- 실습: Git Push → Jenkins → K8s 자동배포 구성
[7]	Ingress-Nginx + 경로 기반 라우팅	- Ingress Controller 설치 (Minikube Addon)- 도메인/경로 기반 서비스 분리- 실습: /api, /admin, /user 등으로 분기 라우팅 구성
[8]	Prometheus + Grafana 모니터링	- Prometheus 설치 및 Spring Boot 와 연동- actuator, micrometer 설정- Grafana 설치 및 대시보드 구성- 실습: JVM 메모리, 요청 수, 응답 속도 시각화
[9]	전체 통합 배포 흐름 구성	- Git → Jenkins → Docker → K8s → Ingress → Grafana- 장애 복구 및 배포 실패 대응 실습- 실습: 실제 시나리오 기반 전체 배포 테스트

## 환경구성

- OS: Windows 11
- 환경: WSL2 (Ubuntu), Docker Desktop, Minikube, Jenkins, Prometheus, Grafana
- 백엔드: Spring Boot (Maven 기반)

## DevOps 란?

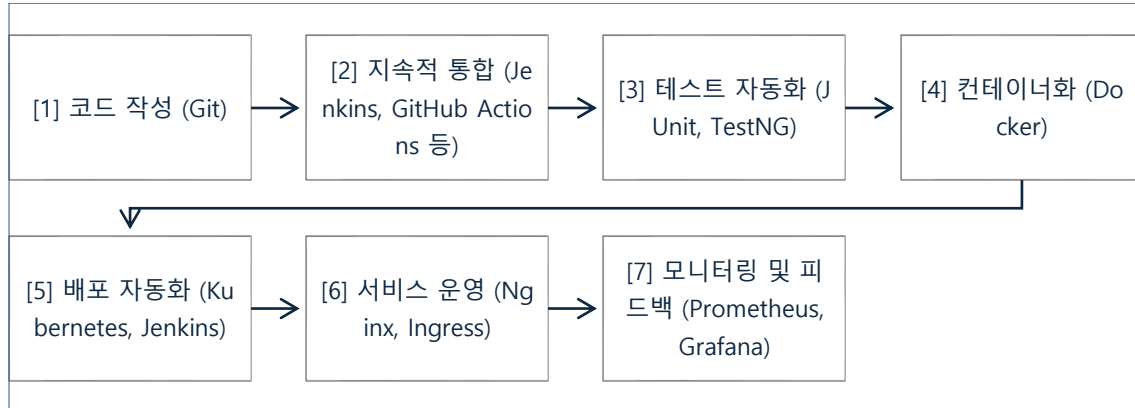


개발(Dev) 팀과 운영(Ops) 팀이 협업과 자동화를 통해 빠르게 고품질의 서비스를 지속적으로 배포할 수 있도록 돕는 방법론

## DevOps의 핵심 개념

개념	설명
지속적 통합 (CI) Continuous Integration	개발자가 코드를 자주 통합하도록 하여, 빌드와 테스트 자동화로 품질을 빠르게 검증
2. 지속적 배포 (CD) Continuous_ Deployment/Delivery	코드 변경이 자동으로 운영 환경에 배포되도록 하여 출시 시간을 단축
3. 자동화 (Automation)	인프라, 테스트, 배포를 자동화해 휴먼 에러 감소 + 일관성 유지
4. 모니터링 (Monitoring)	서비스 상태를 실시간 관찰해 장애를 빠르게 감지하고 대응
5. 협업과 문화 (Culture)	사일로 제거: 개발, 운영, QA 팀이 공동 목표를 가지고 일하는 문화

## DevOps 흐름 (Pipeline)



## DevOps 주요 도구

목적	대표 도구
버전 관리	Git, GitHub, GitLab
CI/CD 파이프라인	Jenkins, GitHub Actions, GitLab CI
컨테이너	Docker
오케스트레이션	Kubernetes (k8s), Minikube
모니터링	Prometheus, Grafana
배포 게이트웨이	Nginx, Ingress Controller



## 추천 시스템 사양

항목	권장
CPU	Intel i5/i7 or AMD Ryzen 5 이상 (4~8 코어)
RAM	16GB 이상 (최소 12GB)
Storage	SSD (Docker Desktop 속도에 중요)
OS	Windows 11 (WSL2 기본 지원)

**개발용 Docker+WSL+DB+Spring 전체 사용 -> 16GB 이상 권장**

## 권장 메모리 요구량 { 풀스택 개발 환경 구성 요소 }

구성	설명	메모리 사용량(예상)
WSL2 + Ubuntu	리눅스 실행 환경	1.5~2GB
Docker Desktop	컨테이너 실행 기반	1.5~2GB
Node.js + React (프론트엔드)	개발 서버, Vite, Webpack 등	1~2GB
Spring Boot (백엔드)	Java 기반 서버	1~2GB
MySQL or MongoDB	데이터베이스	1~1.5GB
Nginx or Apache	프록시/정적 서버용	100~300MB
Jenkins, Redis 등	CI/CD 또는 캐시용	1~2GB (선택)
브라우저(Chrome 등)	디버깅 및 실행 확인	1~2GB

## 리눅스 설치 :

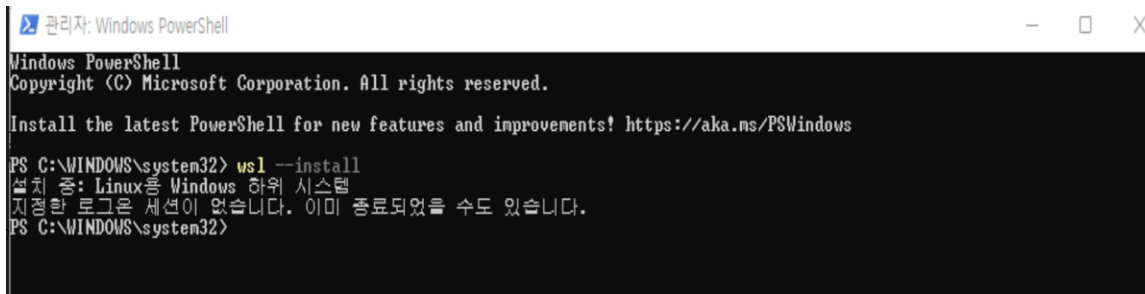
<https://learn.microsoft.com/en-us/windows/wsl/install-manual>

### WSL2 + Ubuntu 설치 및 설정 case1 :

#### 1. WSL2 기능 활성화 : 설치 후 시스템 재부팅 필요

```
# PowerShell (관리자 권한)에서 실행
wsl --install
```

만일 아래와 같다면 수동설치



```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
설치 중: Linux를 Windows 하위 시스템
지정된 로그인 세션이 없습니다. 이미 종료되었을 수도 있습니다.
PS C:\WINDOWS\system32>
```

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-
Linux /all /norestart
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all
/norestart
```

명령어	역할	설명
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux	WSL 1 기능 켜기	리눅스 명령 실행 가능
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform	WSL 2 용 가상 머신 기능	리눅스 커널 실행 기반
/norestart	재시작 없이 적용	필요시 수동 재부팅 필요

< 재부팅 > -> 아래실행

wsl --set-default-version 2
wsl --install -d Ubuntu
wsl --list --online
wsl --list --verbose

## 2. Ubuntu 실행 후 초기 세팅

wsl --install -d Ubuntu
# 사용자 계정 설정 후 아래 명령 실행
sudo apt update
sudo apt upgrade -y
# 유용한 기본 도구 설치
sudo apt install -y curl wget unzip vim git net-tools htop build-essential

## 3. WSL2 네트워크 확인 & 설정 팁

WSL2 특성상 외부 접속 시 localhost 대신 WSL IP 또는 localhost:포트 포워딩 설정 필요

# 현재 IP 주소 확인
ip a
# 포트 점유 확인
ss -tuln

## WSL2 + Ubuntu 설치 및 설정 case2 :

- Microsoft Store 열기 -> Ubuntu 24.04 LTS 검색 ->

"설치" 버튼 클릭 -> 설치 완료 후 시작 메뉴 → Ubuntu 24.04 실행,

Windows Subsystem for Linux (WSL) 찾아서 실행

## WSL2 + Ubuntu 설치 및 설정 case3 :

### 공식 WSL 2 커널 업데이트 설치

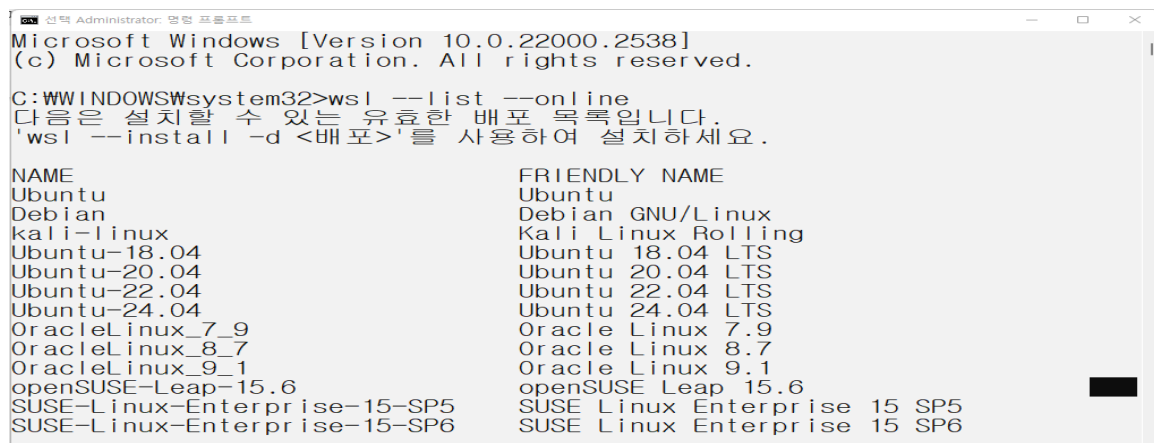
1. 링크 클릭: <https://aka.ms/wsl2kernel>
2. 직접 다운로드:

[https://wslstorestorage.blob.core.windows.net/wslblob/wsl\\_update\\_x64.msi](https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi)

```
C:\WINDOWS\system32>wsl --set-default-version 2
```

### 목록확인

```
wsl --list --online
```



```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>wsl --list --online
다음은 설치할 수 있는 유효한 배포 목록입니다.
'wsl --install -d <배포>'를 사용하여 설치하세요.

NAME                                FRIENDLY NAME
Ubuntu                              Ubuntu
Debian                              Debian GNU/Linux
kali-linux                           Kali Linux Rolling
Ubuntu-18.04                         Ubuntu 18.04 LTS
Ubuntu-20.04                         Ubuntu 20.04 LTS
Ubuntu-22.04                         Ubuntu 22.04 LTS
Ubuntu-24.04                         Ubuntu 24.04 LTS
OracleLinux_7_9                      Oracle Linux 7.9
OracleLinux_8_7                      Oracle Linux 8.7
OracleLinux_9_1                      Oracle Linux 9.1
openSUSE-Leap-15.6                   openSUSE Leap 15.6
SUSE-Linux-Enterprise-15-SP5         SUSE Linux Enterprise 15 SP5
SUSE-Linux-Enterprise-15-SP6         SUSE Linux Enterprise 15 SP6
```

### 설치

```
C:\WINDOWS\system32>wsl --install -d Ubuntu-24.04
```

## 1. 리눅스 ?

유닉스(UNIX)를 기반으로 만들어진 **오픈소스 운영체제 커널**, 사용자가 자유롭게 수정, 재배포할 수 있으며 서버, 클라우드, 임베디드 등 다양한 환경에서 사용된다.

리눅스(Linux)는 커널(운영체제의 핵심)이고, 우분투(Ubuntu)는 **리눅스 커널을 기반으로 만든 하나의 완성된 리눅스 운영체제**이다.

### ◇ 구성 요소

- **커널 (Kernel):** 하드웨어와 소프트웨어를 중재하는 핵심
- **셸 (Shell):** 명령어 입력/실행 환경 (bash, zsh 등)
- **유틸리티:** ls, cp, chmod 등 명령어 모음

항목	리눅스 (Linux)	우분투 (Ubuntu)
종류	커널(핵심 엔진)	리눅스 배포판 중 하나
만든 곳	Linus Torvalds	Canonical Ltd
구성	커널만 제공	커널 + 셸 + 명령어 + 패키지 + GUI 등
사용자용 가용유무	직접 사용 불가	설치하면 바로 사용 가능
이름 종류	리눅스 커널 6.8 등 <b>kernel.org</b>	Ubuntu 22.04, 24.04 등 <b>ubuntu.com</b>

## 2. 리눅스 역사

<b>1969</b>	벨 연구소에서 유닉스(UNIX) 개발 시작
<b>1991</b>	핀란드의 리누스 토르발스가 리눅스 커널 0.01 발표
<b>1992</b>	GPL(자유소프트웨어 라이선스)로 리눅스 공개 전환
<b>1994</b>	리눅스 커널 1.0 정식 릴리스
<b>2000년대~</b>	RedHat, Debian, Ubuntu, CentOS 등 배포판 활성화
<b>최근</b>	리눅스는 클라우드, 서버, AI, IoT, 모바일(Android)의 핵심 운영체제가 됨



---

### 3. 리눅스의 특징

- 오픈소스: 소스코드 공개, 누구나 기여 가능
  - 다중 사용자 / 다중 작업 (Multi-user / Multitasking)
  - 강력한 네트워크 지원 (웹 서버, 방화벽, DNS 등)
  - 모듈식 구조: 필요한 구성요소만 설치 가능
  - 안정성, 보안성 높음
- 

### 4. 현재 리눅스 커널 버전 체계

- 리눅스 커널은 <https://kernel.org> 에서 유지됨
- 버전 규칙: X.Y.Z
  - X: 메이저 버전
  - Y: 안정/보완
  - Z: 패치 레벨

**\*2025 년 기준 최신 커널: 6.9.x (2025 년 6 월 기준)**

---

### 5. 주요 리눅스 배포판

계열	배포판	설명
Debian 계열	Ubuntu, Kali, Raspberry Pi OS	데스크탑, 교육, 해킹툴, 라즈베리파이 등
RedHat 계열	CentOS, Rocky, Alma, RHEL	서버, 기업용에서 많이 사용
Arch 계열	Arch, Manjaro	심플한 롤링 업데이트 방식
기타	Fedora, openSUSE, Alpine	데스크탑 실험, 경량 컨테이너 등

---

---

## 6. 온라인 학습 리소스

유형	설명	링크
<b>Linux Journey</b>	리눅스 초보자를 위한 여정식 튜토리얼	<a href="#">바로가기</a>
<b>The Linux Command Line Book</b>	CLI 명령어 전자책 (무료 PDF)	<a href="#">바로가기</a>
<b>FreeCodeCamp - 리눅스 강의 (2 시간)</b>	명령어 중심 전체 흐름 강의	<a href="#">바로가기</a>

## 리눅스 구조

<b>사용자 (User)</b> 쉘(Bash, Zsh 등)	시스템과 상호작용하는 주체. 사람이 직접 사용하는 계층.
<b>유틸리티/명령어</b> ls, cp, mv, apt, vim...	/bin, /usr/bin, /usr/local/bin 등에 위치 파일 조작, 시스템 관리, 텍스트 편집 등 다양한 작업 수행 apt는 패키지 설치 도구, vim은 텍스트 편집기 등으로 다양한 기능 제공
<b>시스템 라이브러리</b> glibc, OpenSSL, etc.	프로그램이 직접 커널과 통신하지 않고 라이브러리를 통해 간접 접근 /lib, /usr/lib, /lib64 등의 경로에 위치
<b>커널 (Linux Kernell)</b> read(), write(), fork()	리눅스 운영체제의 핵심. 하드웨어와 소프트웨어 간 연결 담당 메모리 관리, 프로세스 관리, 파일 시스템 제어, 장치 드라이버 제어 네트워크 관리
<b>하드웨어 (Hardware)</b> CPU, RAM, 디스크, etc.	커널이 각 장치를 드라이버를 통해 제어하며 사용자 명령을 하드웨어에서 실행 가능하게 함

```

/
├── bin    (기본 명령어 ls, cp 등)
├── boot   (부트로더, 커널 등)
├── dev    (디바이스 파일)
├── etc    (설정 파일)
├── home   (사용자 홈 디렉토리)
├── lib    (라이브러리)
├── media  (USB, 외장디스크 마운트)
├── opt    (추가 설치 프로그램)
├── proc   (가상 프로세스 정보)
├── root   (root 사용자 홈)
├── sbin   (관리자 명령어)
├── tmp    (임시 파일)
├── usr    (공유 명령어, 라이브러리 등)
└── var    (로그, 캐시, 메일 등)

```

## 셸 리눅스 도움말 보는 방법

명령어	사용 형식	설명	특징/장점	종료 방법
man	man 명령어	매뉴얼 페이지 (공식 문서)	가장 상세하고 신뢰도 높음	q
--help	명령어 --help	간단한 옵션 설명	빠르고 간단한 요약 확인 가능	자동 종료
info	info 명령어	구조화된 문서 정보	링크 구조로 설명, man 보다 깊이 있음	q
whatis	whatis 명령어	한 줄 요약 설명	매우 간단한 정의 제공	자동 종료
apropos	apropos 키워드	관련 명령어 검색	키워드 기반 검색, 학습 및 탐색용	자동 종료

**man ls**                      ls 명령어의 전체 매뉴얼 페이지

**ls --help**                      ls 의 사용법, 옵션 목록 요약 출력

**info ls**                      info 시스템의 ls 관련 문서 표시

**whatis ls**                      ls (1) - list directory contents

**apropos copy**                cp (1) - copy files and directories

## 리눅스 기본 명령어 정리

### 1. 디렉토리 및 위치 확인

명령어	설명	예시
pwd	현재 작업 디렉토리 경로 출력	/home/user
ls	디렉토리 내 파일 목록 출력	ls -l, ls -a
cd	디렉토리 이동	cd /etc, cd .., cd ~
tree	디렉토리 구조 트리 형식 출력	tree (설치 필요: apt install tree)

- Q1. 현재 사용자의 작업 디렉토리 경로를 출력하시오.
- Q2. 현재 디렉토리 내의 모든 파일 및 디렉토리를 자세히(long format) 확인하시오.
- Q3. 숨김 파일을 포함하여 현재 디렉토리의 전체 내용을 확인하시오.
- Q4. 홈 디렉토리(~)로 이동한 후, 현재 경로를 출력하시오.
- Q5. /etc 디렉토리로 이동한 후, 상위 디렉토리로 한 단계 이동하시오.
- Q6. /tmp 디렉토리에서 새로운 디렉토리 practice01 을 만들고, 그 안으로 이동하시오.
- Q7. practice01 디렉토리 안에 빈 파일 data.txt, memo.txt 를 만들고, ls -l 로 확인하시오.
- Q8. 현재 디렉토리 기준으로 트리 구조를 출력하시오. (설치 필요 시 설치)
- Q9. /home 디렉토리 전체 구조를 tree 명령어로 출력하시오. 단, 깊이는 2 단계까지만 제한하시오.
- cd 명령어를 사용하여 /var, /usr, /bin 디렉토리를 각각 순서대로 방문한
- Q10. 뒤, 다시 홈 디렉토리로 돌아오시오. 각 단계에서 pwd 로 경로를 확인하시오.

## 2. 파일 및 디렉토리 관리

명령어	설명	예시
touch	빈 파일 생성	touch file.txt
mkdir	새 디렉토리 생성	mkdir newfolder
cp	파일 복사	cp file.txt /tmp/
mv	파일 이동 또는 이름 변경	mv old.txt new.txt
rm	파일 또는 디렉토리 삭제	rm file.txt, rm -r folder/

- Q1. 현재 디렉토리에 report.txt 라는 빈 파일을 생성하시오.
- Q2. ~/practice 라는 이름의 새 디렉토리를 생성하시오.
- Q3. report.txt 파일을 practice 디렉토리로 복사하시오.
- Q4. practice 디렉토리로 이동하여 report.txt 의 이름을 log.txt 로 변경하시오.
- Q5. 현재 디렉토리에 note1.txt, note2.txt 파일을 각각 생성하시오.
- Q6. note1.txt 를 /tmp 디렉토리로 복사하고, 복사되었는지 확인하시오.
- Q7. note2.txt 를 /tmp 디렉토리로 **이동**시키고, 현재 디렉토리에서 없어졌는지 확인하시오.
- Q8. /tmp 디렉토리로 이동한 뒤, note2.txt 파일의 이름을 final.txt 로 변경하시오.
- Q9. 홈 디렉토리에 oldfile.txt 파일을 생성한 후, 삭제하시오.
- Q10. ~/practice 디렉토리 전체를 삭제하시오. (log.txt 파일 포함)
-

### 3. 파일 내용 확인

명령어	설명	예시
cat	파일 내용 전체 출력	cat file.txt
less, more	페이지 단위로 보기	less file.txt
head	앞부분 일부 출력	head -n 10 file.txt
tail	뒷부분 일부 출력	tail -n 20 file.txt

- Q1. sample.txt 라는 파일을 만들고, cat 명령어로 그 내용을 출력하시오.  
(파일에는 미리 내용 2~3 줄 작성)
- Q2. cat 명령어를 사용하여 여러 줄의 텍스트를 직접 입력하고 multi.txt 파일로 저장하시오.
- Q3. multi.txt 파일을 more 명령어로 한 페이지씩 넘기며 확인하시오.
- Q4. multi.txt 파일을 less 명령어로 열고, 위/아래 화살표를 사용해 탐색한 후 종료하시오.
- Q5. head 명령어를 사용하여 multi.txt 의 **처음 5 줄**을 출력하시오.
- Q6. tail 명령어를 사용하여 multi.txt 의 **마지막 5 줄**을 출력하시오.
- Q7. head 명령어를 이용해 **처음 10 줄**을 출력하되, 옵션을 명시적으로 사용하시오. (-n)
- Q8. tail -f 명령어로 log.txt 파일을 **실시간으로 감시**하고, 터미널에서 내용을 추가해 확인하시오.
- Q9. cat 명령어를 사용하여 multi.txt 와 sample.txt 두 개의 파일 내용을 한 번에 출력하시오.
- Q10. less 를 사용하여 텍스트 파일을 열고, 검색(/문자열) 기능으로 특정 단어를 찾아보시오.

#### 4. 검색 및 필터링

명령어	설명	예시
grep	특정 문자열 검색	grep "error" log.txt
find	파일 또는 디렉토리 검색	find / -name "passwd"

다음내용을 사용하여 실습을 진행하시오

```
echo -e "Error occurred\nWarning issued\nFail to connect" > log.txt
touch emptyfile.txt
mkdir emptydir
```

- grep 을 사용해 memo.txt 파일에서 "error"라는 단어가 포함된 줄을 찾아  
Q1. 출력하시오.
- Q2. 대소문자를 구분하지 않고 "Warning"이라는 단어를 log.txt 에서 검색하시오.
- Q3. log.txt 에서 "fail"이라는 단어가 포함된 줄 번호와 함께 출력하시오.
- 현재 디렉토리에서 .txt 확장자를 가진 파일 중, "user"라는 단어를 포함한 줄을  
Q4. 모두 검색하시오.
- Q5. /etc/passwd 파일에서 "bash"라는 문자열이 **끝에 있는 줄만** 출력하시오.
- 현재 디렉토리 이하에서 이름이 config.json 인 파일을 find 명령어로 찾아 절대  
Q6. 경로로 출력하시오.
- Q7. find 를 사용하여 현재 디렉토리에서 7 일 이내에 수정된 .sh 파일을 검색하시오.
- Q8. 현재 위치에서 크기가 1MB 이상인 파일을 모두 검색하시오.
- Q9. 현재 디렉토리에서 빈 디렉토리를 찾아 모두 출력하시오.
- /var 디렉토리에서 이름에 log 라는 단어가 포함된 파일 또는 디렉토리를 모두  
Q10. 검색하시오.



---

## 5. 시스템 정보 및 리소스 확인

명령어	설명
uname -a	시스템 정보 출력
top	실시간 프로세스 확인
df -h	디스크 사용량 확인
free -h	메모리 사용량 확인

- Q1. 현재 실행 중인 리눅스 시스템의 커널 이름, 버전, 시스템 정보 전체를 출력하시오.
- Q2. 현재 사용 중인 리눅스가 32 비트인지 64 비트인지 확인하시오.
- Q3. top 명령어를 실행하여 실시간으로 CPU 사용량과 메모리 상태를 확인하시오.
- Q4. top 실행 중 현재 가장 많은 CPU 를 사용하는 프로세스를 확인하고 종료하시오.
- Q5. 전체 디스크 공간과 사용량을 사람이 읽기 쉬운 형태로 출력하시오.
- Q6. 특정 디렉토리(/home, /tmp)의 디스크 사용량을 확인하시오.
- Q7. 메모리의 전체 용량, 사용 중인 메모리, 남은 메모리를 확인하시오.
- Q8. 스왑(Swap) 메모리의 사용량을 확인하시오.
- Q9. free -h 명령어 결과에서 사용 가능한(available) 메모리가 몇 MB 인지 읽어내시오.
- Q10. 시스템의 하드웨어 정보를 한 줄 요약으로 정리하여 설명하시오. (uname -a 기반으로 서술)
-

## 6. 사용자 및 권한 관리

명령어	설명	예시
whoami	현재 로그인한 사용자 확인	
sudo	관리자 권한으로 명령 실행	sudo apt update
chmod	파일 권한 변경	chmod +x script.sh
chown	소유자 변경	chown user:user file.txt

Q1. 현재 로그인한 사용자의 이름을 출력하시오.

Q2. 현재 사용자의 UID 와 GID 를 확인하시오.

Q3. 일반 사용자 계정에서 sudo 명령어를 사용하여 apt update 를 실행하시오.

Q4. sudo 를 사용하여 /etc 디렉토리의 파일 목록을 출력하시오.

Q5. practice 디렉토리를 생성하고, 그 안에 test.txt 파일을 만들고, 현재 권한을 확인하시오.

Q6. test.txt 파일의 권한을 모든 사용자에게 읽기, 쓰기, 실행 권한이 있도록 변경하시오.

Q7. test.txt 파일의 권한을 소유자는 읽기/쓰기, 그룹은 읽기, 기타는 접근 불가로 변경하시오.

Q8. 현재 사용자가 소유한 test.txt 파일의 소유권을 myuser 사용자로 변경하시오.  
(sudo 사용)

Q9. chown 명령어를 사용하여 test.txt 파일의 소유자와 소유 그룹을 모두 myuser 로 변경하시오.

Q10. ls -l 명령어를 사용하여 소유자/그룹/권한 변경 사항이 적용되었는지 확인하시오.

---

## 7. 패키지 관리 (Ubuntu 기준)

명령어	설명	예시
apt update	패키지 목록 갱신	sudo apt update
apt install	패키지 설치	sudo apt install curl
apt remove	패키지 제거	sudo apt remove curl

Q1. 시스템의 패키지 목록을 최신 상태로 갱신하시오.

Q2. tree 명령어를 사용하기 위해 해당 패키지를 설치하시오.

Q3. htop 이라는 시스템 모니터링 도구를 설치하시오.

Q4. 설치된 htop 패키지가 제대로 작동하는지 확인하기 위해 실행하시오.

Q5. htop 명령어를 종료한 후, 해당 패키지를 시스템에서 삭제하시오.

Q6. tree 패키지를 삭제하시오.

Q7. 존재하지 않는 패키지 이름을 입력했을 때 어떤 에러 메시지가 출력되는지  
확인하시오. (예: apt install notapackage)

Q8. apt 명령어로 설치 가능한 패키지 목록을 검색하는 명령어를 찾아 실행하시오.

Q9. apt update 와 apt upgrade 의 차이를 설명하시오.

Q10. apt remove 와 apt purge 의 차이를 설명하시오.

---

---

## 8. 기타 명령어

명령어	설명	예시
echo	문자열 출력	echo "Hello Linux"
date	현재 날짜 및 시간 출력	
history	실행한 명령어 목록 보기	
clear	터미널 화면 정리	

Q1. echo 명령어를 사용하여 "Hello, Linux"라는 문장을 터미널에 출력하시오.

Q2. echo 명령어로 현재 사용자의 홈 디렉토리 경로를 출력하시오. (\$HOME 변수 사용)

Q3. echo 명령어를 사용하여 USER, SHELL 환경변수 값을 각각 출력하시오.

Q4. date 명령어를 사용하여 현재 날짜와 시간을 출력하시오.

Q5. date 명령어를 사용하여 현재 날짜만 YYYY-MM-DD 형식으로 출력하시오.

Q6. date 명령어를 사용하여 현재 시간을 HH:MM:SS 형식으로 출력하시오.

Q7. history 명령어를 사용하여 지금까지 터미널에 입력한 명령어 목록을 확인하시오.

Q8. history 에서 가장 최근 5 개의 명령어만 출력하시오.

Q9. clear 명령어를 사용하여 현재 터미널 화면을 깨끗하게 초기화하시오.

Q10. history 명령어를 사용하여 특정 명령어(echo)가 포함된 과거 기록을 검색하시오.