



인하공업전문대학
INHA TECHNICAL COLLEGE

서버구축실습

2023년 2학기



리눅스 서버구축 Stack

리눅스 네트워크
서비스 관리

리눅스 서버

네트워크 설정

텔넷 SSH

FTP

DB

Apache(웹서버)

NFS & SAMBA

리눅스
시스템 관리

디스크관리

패키지 관리

사용자 관리

리눅스
기본 명령

디렉터리와
파일

문서
편집

셸사용

접근
권한
설정

프로세스
관리

리눅스
개요 및 설치

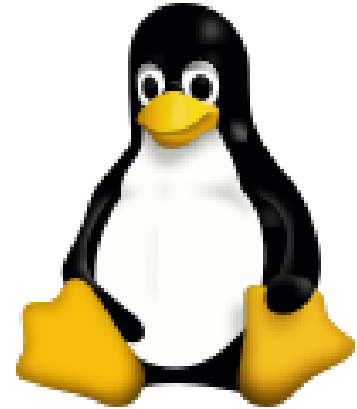
리눅스 개요

AWS 리눅스 인
스턴스 생성

명령 사용 환경

리눅스의 역사

- 1991년 9월 17일 리눅스는 리누스 토발즈(Linux Torvals)가 처음 출시
- 리눅스 커널에 기반을 둔 오픈소스 유닉스 계열 운영체제
- 배포판에는 리눅스 커널과 지원 시스템 소프트웨어, 그리고 라이브러리를 포함

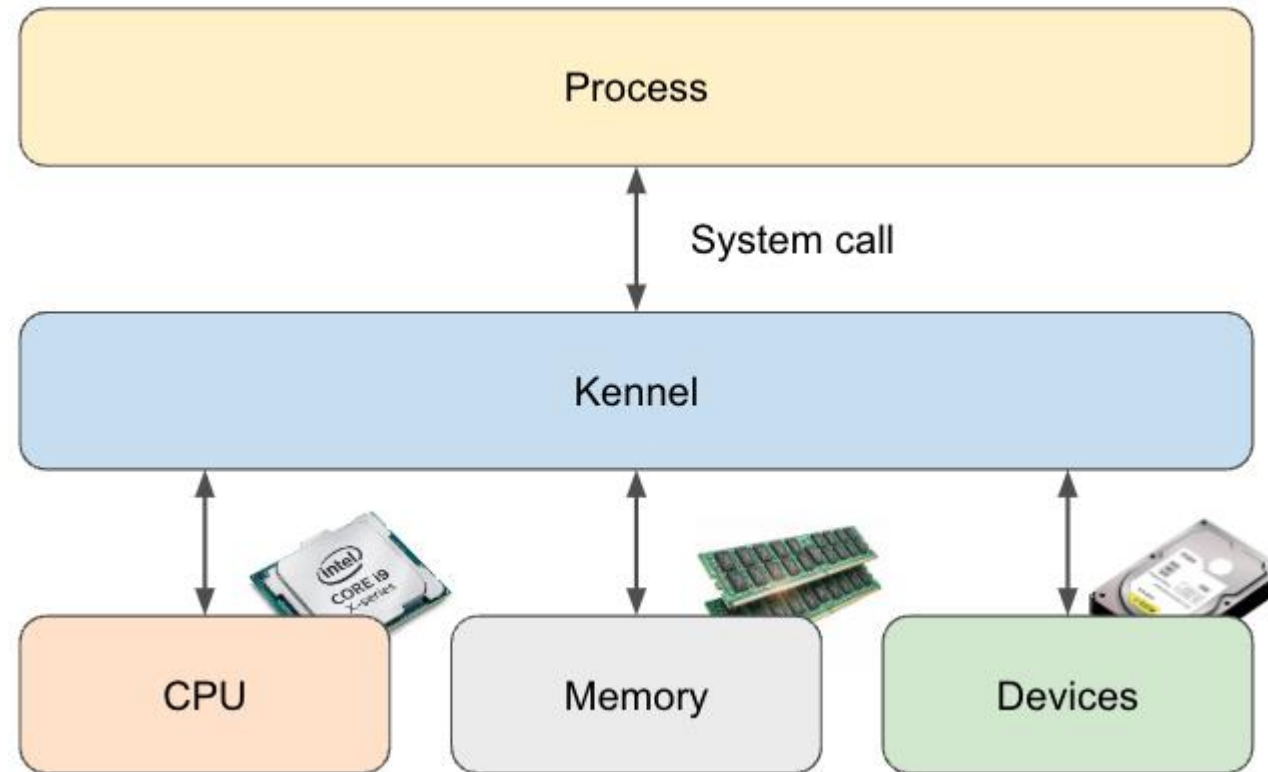


사용이유?

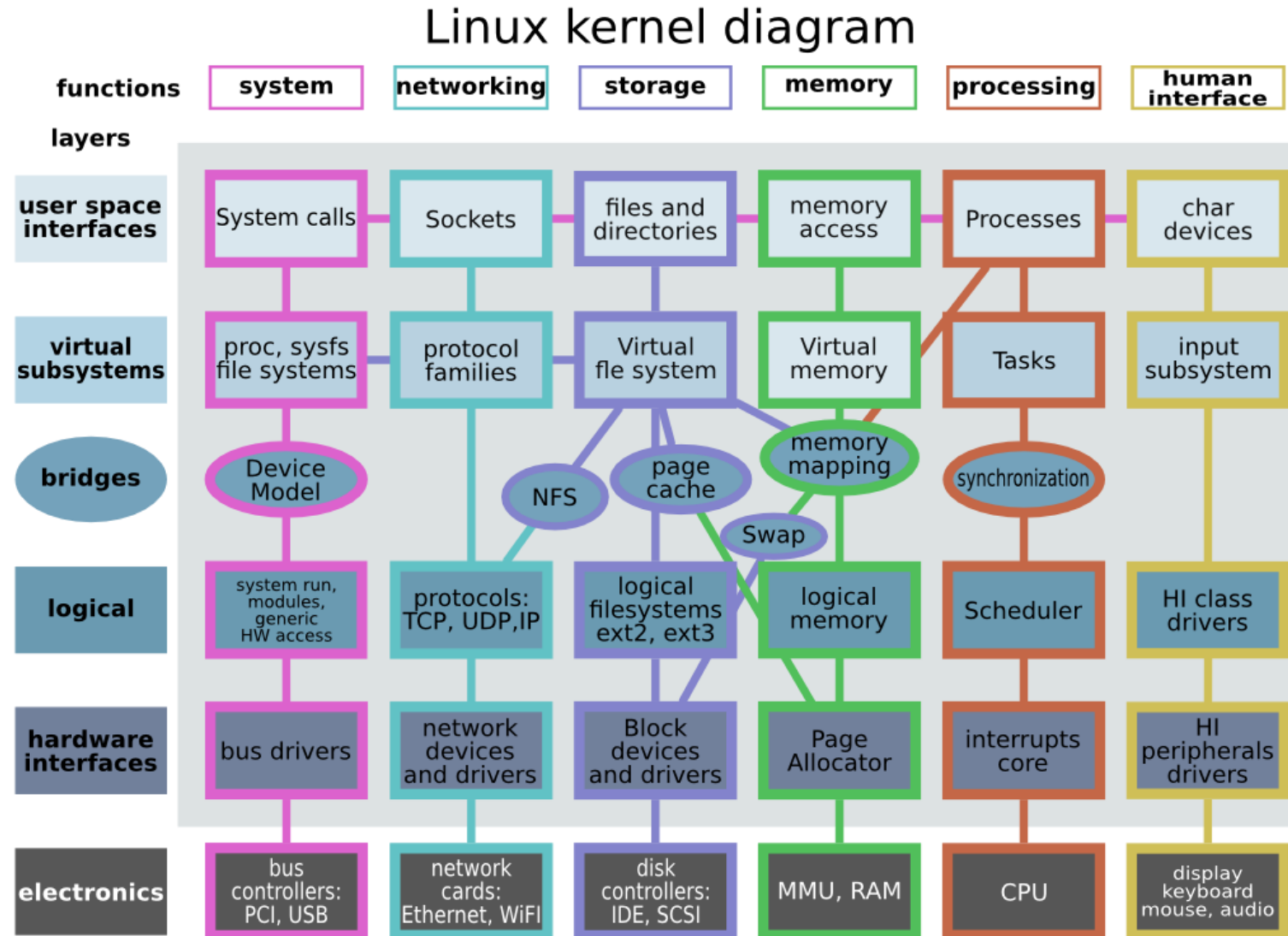
1. 리눅스는 공개 소프트웨어이며 무료로 사용가능
2. 유닉스와 완벽한 호환성을 유지
3. 서버용 운영체제로 많이 사용
4. 편리한 GUI환경 제공

커널이란?

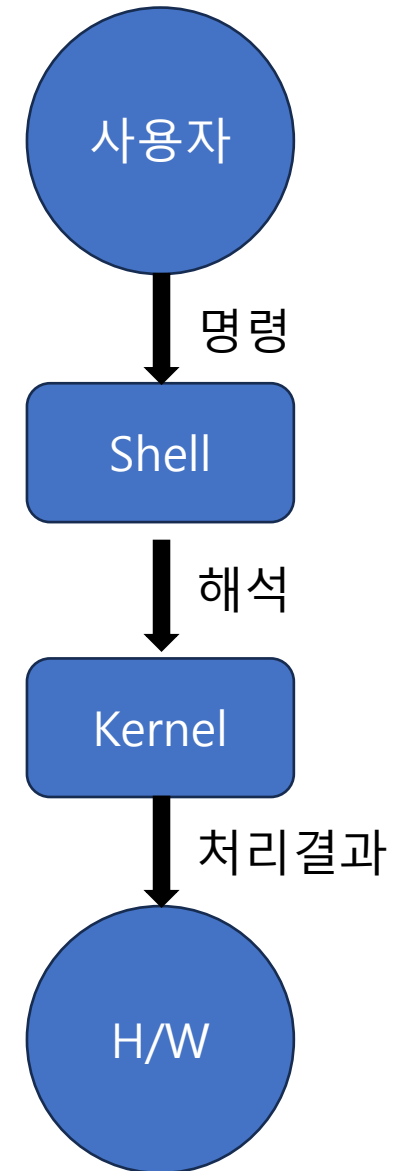
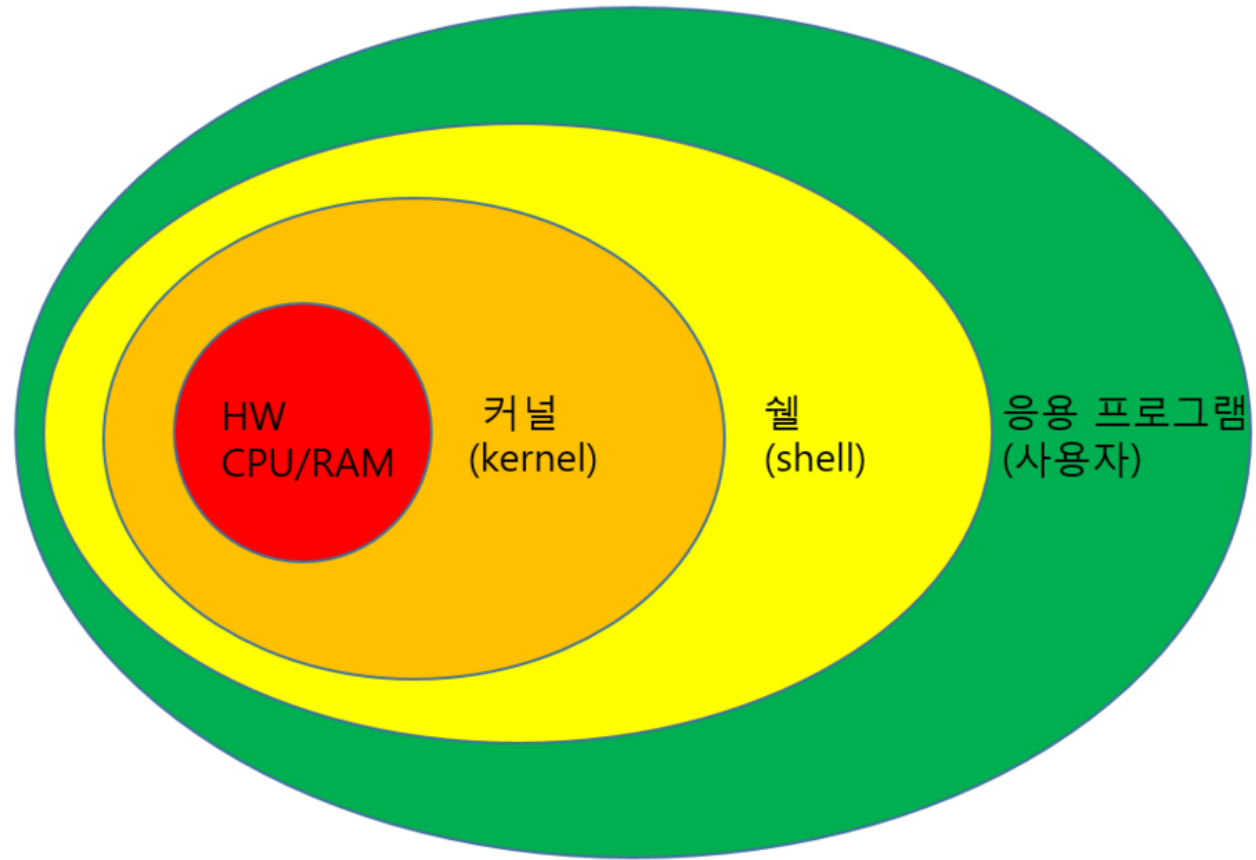
- 하드웨어 리소스(CPU, 메모리, 스토리지 등) 관리
- 소프트웨어에서 하드웨어 및 리소스에 대한 액세스를 추상화



리눅스의 커널

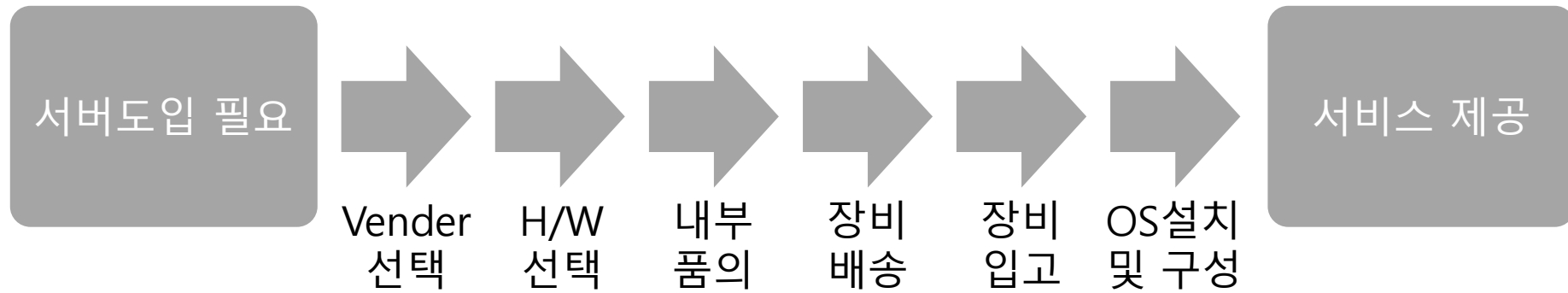


커널과 Shell



AWS 도입 배경

물리서버 도입

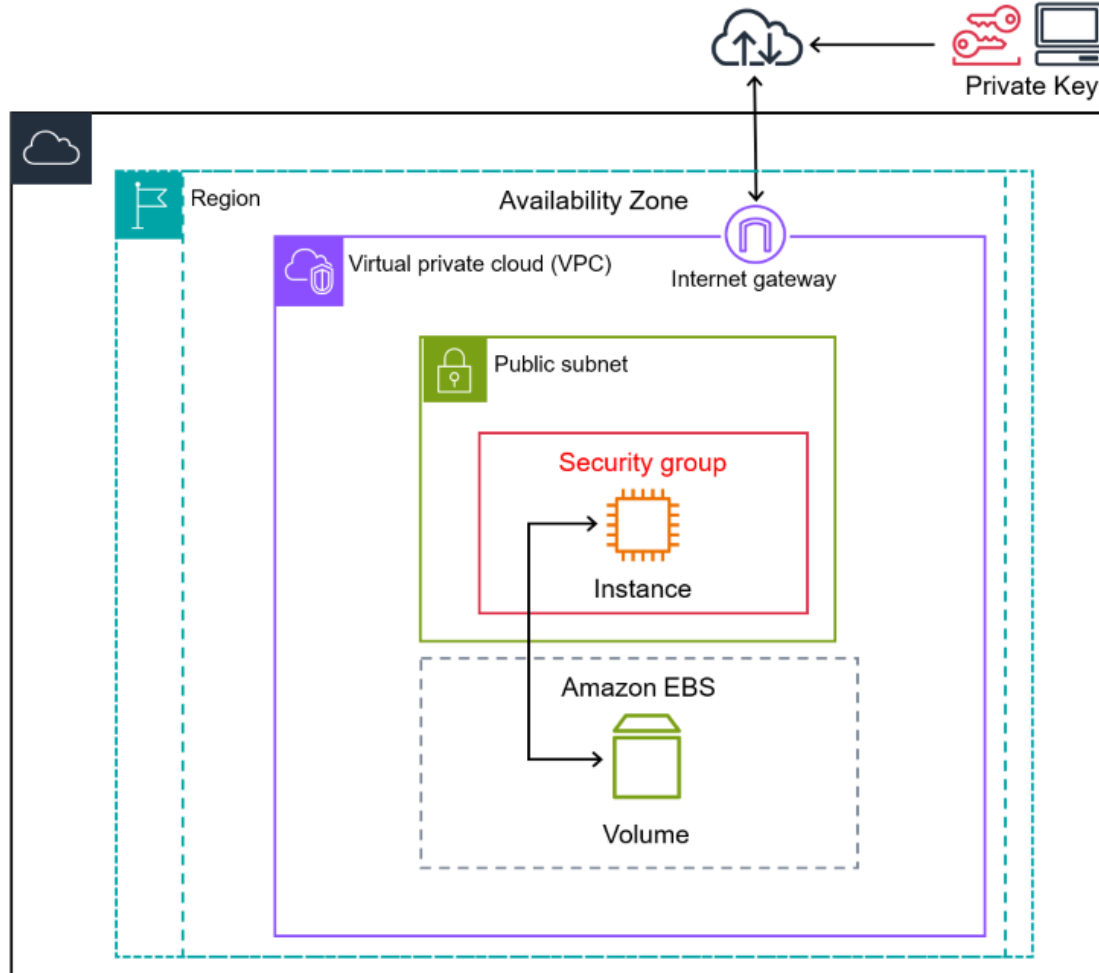


AWS 도입



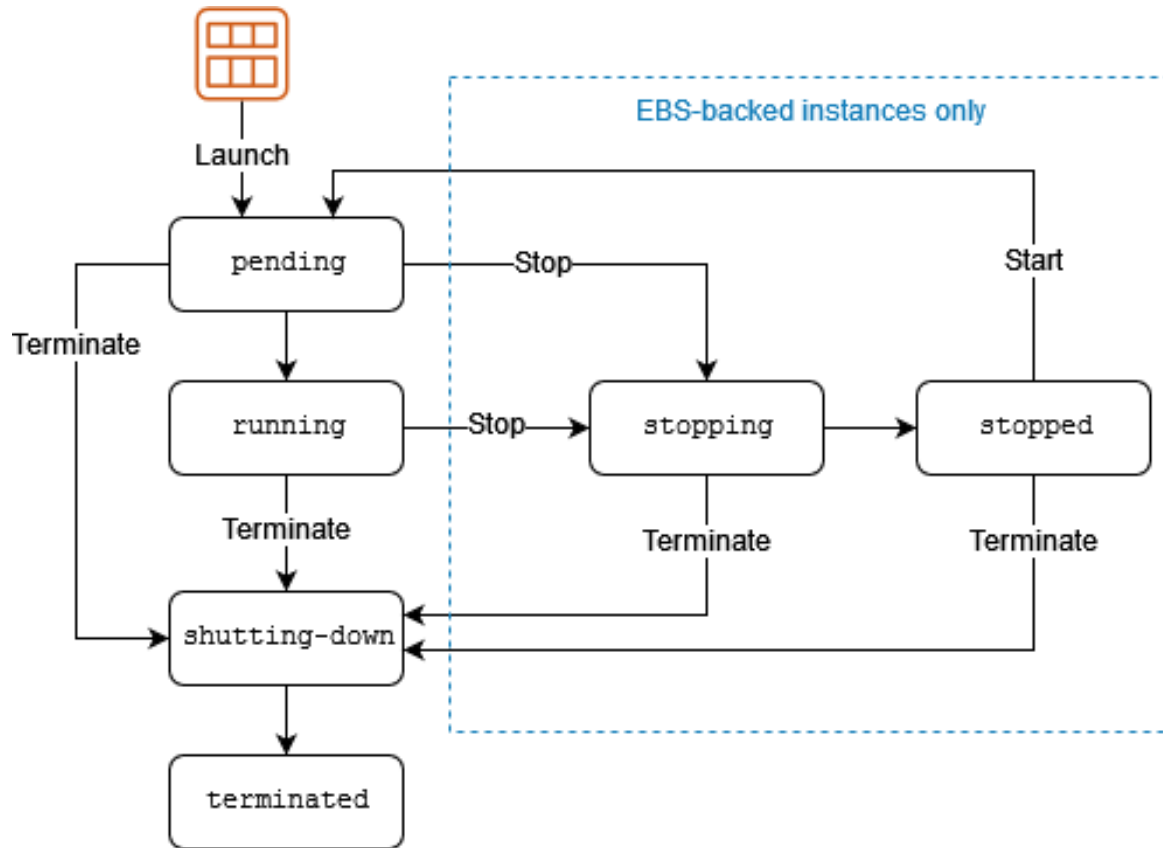
- **효율성** : 실제 서버구축에 들어가는 시간적 비용 단축 가능.
- **비용 절감** : 사용한 만큼 요금 지불

AWS EC2



- Amazon Elastic Compute Cloud(Amazon EC2)
- 클라우드에서 **확장 가능한 컴퓨팅 용량**을 제공
- 사용한 만큼 비용을 지불하기 때문에 탄력적인
이라는 의미의 Elastic

인스턴스 수명주기



- **Pending** : AMI가 실행된 후 준비 상태
- **Running** : 사용한 만큼 요금 지불
- **Shutting-down** : 인스턴스가 종료할 준비 상태
- **Terminated** : 인스턴스가 영구적으로 삭제된 상태 (시작불가)
- **Stopping** : 인스턴스를 중지할 준비 상태
- **Stopped** : 인스턴스가 종료되고 사용 불가, 언제든지 다시 시작 가능

AWS 로그인



로그인

☒ 루트 사용자

무제한 액세스 권한이 필요한 작업을 수행하는 계정 소유자입니다. [자세히 알아보기](#)

☐ IAM 사용자

일일 작업을 수행하는 계정 내 사용자입니다. [자세히 알아보기](#)

루트 사용자 이메일 주소

username@example.com

다음

계속 진행하는 경우 [AWS 고객 계약](#) 또는 [AWS 서비스에 대한 기타 계약 및 개인 정보 보호 정책](#)에 동의하게 됩니다. 이 사이트는 필수 쿠키를 사용합니다. 자세한 내용은 [쿠키 고지](#)를 참조하세요.

_____ AWS를 처음 사용하십니까? _____

AWS 계정 새로 만들기



축하합니다.

AWS(Amazon 웹 서비스) 구독을 업데이트해 주셔서 감사합니다. 이제 고객님의 관리 콘솔에는 모든 AWS 서비스가 업데이트됩니다.

AWS Management Console로 이동

AWS 콘솔 홈



서비스

검색

[알트+S]



스토어

Jo

콘솔 홈 정보

기본 레이아웃으로 재설정

+ 위젯 추가

최근에 방문한 서비스 정보



최근에 방문한 서비스 없음

자주 방문하는 AWS 서비스 중 하나를 살펴봅니다.

[IAM](#) [EC2](#) [S3](#) [RDS](#) [Lambda](#)

[모든 서비스 보기](#)

AWS 시작



[AWS 시작하기](#)

AWS를 최대한 활용하는 데 도움이 되는 기초 지식을 배우고 유용한 정보를 찾아봅니다.



[교육 및 자격증](#)

AWS 전문가로부터 배우고 기술과 지식을 발전시킵니다.



[AWS의 새로운 소식](#)

새로운 AWS 서비스, 기능 및 리전을 검색합니다.

AWS EC2



- New EC2 Experience ×
Tell us what you think
- EC2 대시보드
- EC2 글로벌 보기
- 이벤트
- ▼ 인스턴스
- 인스턴스
- 인스턴스 유형
- 시작 템플릿
- 스팟 요청
- Savings Plans
- 예약 인스턴스
- 전용 호스트
- 용량 예약
- ▼ 이미지
- AMI
- AMI 카탈로그
- ▼ Elastic Block Store
- 볼륨
- 스냅샷

리소스 EC2 글로벌 보기

유럽 (스톡홀름) 리전에서 다음 Amazon EC2 리소스를 사용하고 있음:

인스턴스(실행 중)	0	로드 밸런서	0	배치 그룹	0
보안 그룹	1	볼륨	0	스냅샷	0
인스턴스	0	전용 호스트	0	키 페어	0
탄력적 IP	0	Auto Scaling 그룹	0		

인스턴스 시작

서버 마이그레이션

참고: 인스턴스는 유럽 (스톡홀름) 리전에서 시작됩니다.

예약된 이벤트

계정 속성

기본 VPC
vpc-009e74a6662fbf717

설정

Data protection and security [New](#)

영역

EC2 직렬 콘솔

기본 크레딧 사양

콘솔 실험

추가 정보

시작 안내서

설명서

모든 EC2 리소스

포럼

요금

문의처

서비스 상태 AWS Health 대시보드

리전

유럽 (스톡홀름)

영역

영역 이름	영역 ID
eu-north-1a	eun1-az1

AWS 인스턴스 구축



[EC2](#) > [인스턴스](#) > 인스턴스 시작

인스턴스 시작 [정보](#)

Amazon EC2를 사용하면 AWS 클라우드에서 실행되는 가상 머신 또는 인스턴스를 생성할 수 있습니다. 아래의 간단한 단계에 따라 빠르게 시작할 수 있습니다.

이름 및 태그 [정보](#)

이름

예: 내 웹 서버

[추가 태그 추가](#)

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) [정보](#)

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보십시오.

Quick Start



 더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

▼ 요약

인스턴스 개수 [정보](#)

1

[소프트웨어 이미지\(AMI\)](#)

Amazon Linux 2023 AMI 2023.1.2...[더 보기](#)
ami-0c147c2e2b026f094

[가상 서버 유형\(인스턴스 유형\)](#)

t3.micro

[방화벽\(보안 그룹\)](#)

새 보안 그룹

[스토리지\(볼륨\)](#)

1개의 볼륨 - 8GiB

❗ 프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다. ✕

[취소](#)

[인스턴스 시작](#)

[명령 검토](#)

AWS 인스턴스 구축

미국 동부 (버지니아 북부)	us-east-1
미국 동부 (오하이오)	us-east-2
미국 서부 (캘리포니아)	us-west-1
미국 서부 (오레곤)	us-west-2
아시아 태평양 (뭄바이)	ap-south-1
아시아 태평양 (오사카)	ap-northeast-3
아시아 태평양 (서울)	ap-northeast-2
아시아 태평양 (싱가포르)	ap-southeast-1
아시아 태평양 (시드니)	ap-southeast-2
아시아 태평양 (도쿄)	ap-northeast-1
캐나다 (중부)	ca-central-1
유럽 (프랑크푸르트)	eu-central-1
유럽 (아일랜드)	eu-west-1
유럽 (런던)	eu-west-2
유럽 (파리)	eu-west-3

이름 및 태그 정보

이름

추가 태그 추가

1. 지역 선택
2. 이름 기재

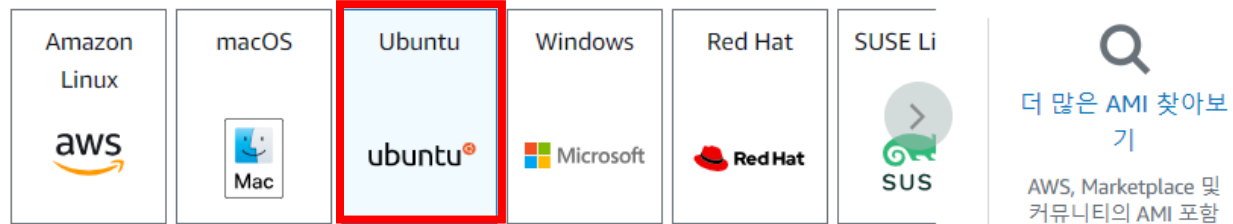
AWS 인스턴스 구축

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보십시오.

🔍 수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

Quick Start



Amazon Machine Image(AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type
ami-08766f81ab52792ce (64비트(x86)) / ami-0ff124a3d7381bfec (64비트(Arm))
가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

프리 티어 사용 가능

설명

Canonical, Ubuntu, 20.04 LTS, amd64 focal image build on 2023-05-17

아키텍처

64비트(x86)

AMI ID

ami-08766f81ab52792ce

확인된 공급 업체

AMI : Ubuntu Server 20.04 LTS 선택
아키텍처 : 64비트 X86

AWS 인스턴스 구축

▼ 인스턴스 유형 정보

인스턴스 유형

t2.micro

프리 티어 사용 가능

패밀리: t2 1 vCPU 1 GiB 메모리 현재 세대: true

온디맨드 RHEL base 요금: 0.0744 USD 시간당

온디맨드 Linux base 요금: 0.0144 USD 시간당

온디맨드 SUSE base 요금: 0.0144 USD 시간당

온디맨드 Windows base 요금: 0.019 USD 시간당

☐ 모든 세대

[인스턴스 유형 비교](#)

[Additional costs apply for AMIs with pre-installed software](#)

인스턴스 유형 : t2.micro

※ **프리 티어** 반드시 확인

AWS 인스턴스 구축



인스턴스 유형 (1/386+) ↗



🔍 Find resources by attribute or tag

< 1 2 3 4 5 6 7 8 ... > ⚙️

	인스턴스 유형 ▾	vCPU ▾	아키텍처 ▾	메모리(GiB) ▾	스토리지(GB) ▾	스토리지 유형 ▾	네트워크 성능 ▾
<input type="radio"/>	t2.nano	1	i386, x86_64	0.5	-	-	Low to Moderate
<input checked="" type="radio"/>	t2.micro	1	i386, x86_64	1	-	-	Low to Moderate
<input type="radio"/>	t2.small	1	i386, x86_64	2	-	-	Low to Moderate
<input type="radio"/>	t2.medium	2	i386, x86_64	4	-	-	Low to Moderate
<input type="radio"/>	t2.large	2	x86_64	8	-	-	Low to Moderate
<input type="radio"/>	t2.xlarge	4	x86_64	16	-	-	Moderate
<input type="radio"/>	t2.2xlarge	8	x86_64	32	-	-	Moderate
<input type="radio"/>	t3.nano	2	x86_64	0.5	-	-	Up to 5 Gigabit
<input type="radio"/>	t3.micro	2	x86_64	1	-	-	Up to 5 Gigabit
<input type="radio"/>	t3.small	2	x86_64	2	-	-	Up to 5 Gigabit
<input type="radio"/>	t3.medium	2	x86_64	4	-	-	Up to 5 Gigabit

AWS 인스턴스 구축

▼ 키 페어(로그인) 정보

키 페어를 사용하여 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스를 시작하기 전에 선택한 키 페어에 대한 액세스 권한이 있는지 확인하세요.

키 페어 이름 - 필수

선택



새 키 페어 생성

▼ 키 페어(로그인) 정보


키 페어를 사용하여 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스를 시작하기 전에 선택한 키 페어에 대한 액세스 권한이 있는지 확인하세요.

키 페어 이름 - 필수

ubuntuKey



새 키 페어 생성

 ubuntuKey.ppk

키 페어 생성



키 페어 이름

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

키 페어 이름 입력

이름에는 최대 255개의 ASCII 문자를 포함할 수 있습니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

키 페어 유형

☒ RSA

RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519

ED25519 암호화된 프라이빗 및 퍼블릭 키 페어

프라이빗 키 파일 형식

☐ .pem

OpenSSH와 함께 사용

☒ .ppk

PuTTY와 함께 사용



메시지가 표시되면 프라이빗 키를 사용자 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

취소

키 페어 생성

AWS 인스턴스 구축

▼ 네트워크 설정 정보

[편집](#)

네트워크 정보

vpc-009e74a6662fbf717

서브넷 정보

기본 설정 없음(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당 정보

활성화

방화벽(보안 그룹) 정보

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 특정 트래픽이 인스턴스에 도달하도록 허용하는 규칙을 추가합니다.

☒ 보안 그룹 생성☐ 기존 보안 그룹 선택

다음 규칙을 사용하여 'launch-wizard-1'(이)라는 새 보안 그룹을 생성합니다.

☒ 에서 SSH 트래픽 허용
인스턴스 연결에 도움

위치 무관
0.0.0.0/0

☒ 인터넷에서 HTTPS 트래픽 허용
예를 들어 웹 서버를 생성할 때 엔드포인트를 설정하려면☒ 인터넷에서 HTTP 트래픽 허용
예를 들어 웹 서버를 생성할 때 엔드포인트를 설정하려면

⚠ 소스가 0.0.0.0/0인 규칙은 모든 IP 주소에서 인스턴스에 액세스하도록 허용합니다. 알려진 IP 주소의 액세스만 허용하도록 보안 그룹을 설정하는 것이 좋습니다. ✕

보안 그룹 생성 : 인스턴스에 접근 할 수 있는 보안 정책 새로 생성

SSH(22), HTTPS(443), HTTP(80) 허용

AWS 인스턴스 구축



▼ 스토리지 구성 정보

[어드밴스드](#)

1x 8 GiB gp2 루트 볼륨 (암호화되지 않음)

8GB, gp2선택

i 프리 티어를 사용할 수 있는 고객은 최대 30GB의 EBS 범용(SSD)또는 마그네틱 스토리지를 사용할 수 있습니다. **×**

새 볼륨 추가

선택한 AMI에 인스턴스가 허용하는 것보다 많은 인스턴스 스토어 볼륨이 포함되어 있습니다. AMI에서 처음 0개의 인스턴스 스토어 볼륨에만 액세스할 수 있습니다.

0x 파일 시스템

[편집](#)

AWS 인스턴스 구축

▼ 요약

인스턴스 개수 정보

1

소프트웨어 이미지(AMI)

Canonical, Ubuntu, 20.04 LTS, ...[더 보기](#)
ami-04341a215040f91bb

가상 서버 유형(인스턴스 유형)

t2.micro

방화벽(보안 그룹)

새 보안 그룹

스토리지(볼륨)

1개의 볼륨 - 8GiB

i 프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

취소

인스턴스 시작

[명령 검토](#)

[EC2](#) > [인스턴스](#) > 인스턴스 시작

🟢 성공

인스턴스를 시작했습니다. (i-07f18f21657d23bd8)

▶ 로그 시작

다음 단계

결제 및 프리 티어 사용 알림 생성

비용을 관리하고 높은 금액의 청구서를 방지하려면 결제 및 프리 티어 사용 임계값에 대한 이메일 알림을 설정합니다.

[결제 알림 생성](#)

인스턴스에 연결

인스턴스가 실행되면 로컬 컴퓨터에서 인스턴스에 로그인합니다.

[인스턴스에 연결](#)

[자세히 알아보기](#)

RDS 데이터베이스 연결

EC2 인스턴스와 데이터베이스 간의 트래픽 흐름을 허용하도록 연결을 구성합니다.

[RDS 데이터베이스 연결](#)

[새 RDS 데이터베이스 생성](#) [자세히 알아보기](#)

[모든 인스턴스 보기](#)

AWS 인스턴스 구축

[EC2](#) > [인스턴스](#) > [i-067f8cd067b43482e](#) > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-067f8cd067b43482e (ubuntuA)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID

 i-067f8cd067b43482e (ubuntuA)

연결 유형

☒ EC2 Instance Connect를 사용하여 연결

퍼블릭 IPv4 주소가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

☐ EC2 인스턴스 연결 엔드포인트를 사용하여 연결

프라이빗 IPv4 주소 및 VPC 엔드포인트가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

퍼블릭 IP 주소

 54.180.118.41

사용자 이름

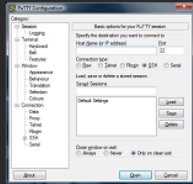
인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ubuntu을(를) 사용합니다.

ubuntu

참고: 대부분의 경우 기본 사용자 이름 ubuntu은(는) 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

PuTTY 설치

<https://www.putty.org/>



Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

[Download PuTTY](#)

Download PuTTY: latest release (0.79)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: **Stable** · [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.79, released on 2023-08-26.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.79 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

MSI ('Windows Installer')

64-bit x86: [putty-64bit-0.79-installer.msi](#) ([signature](#))

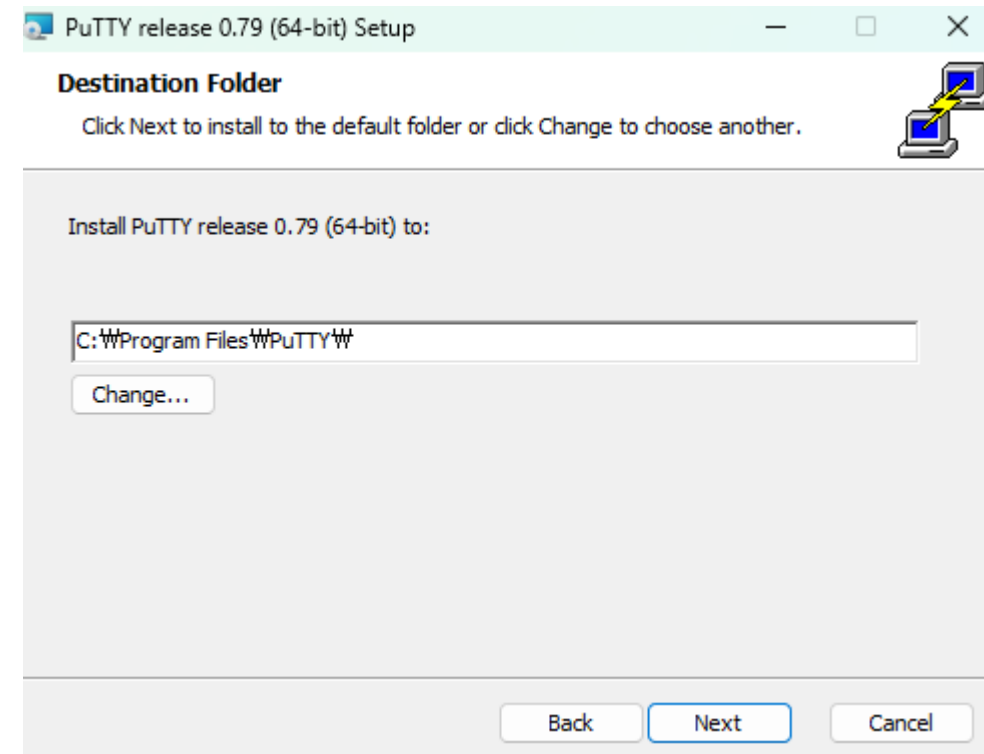
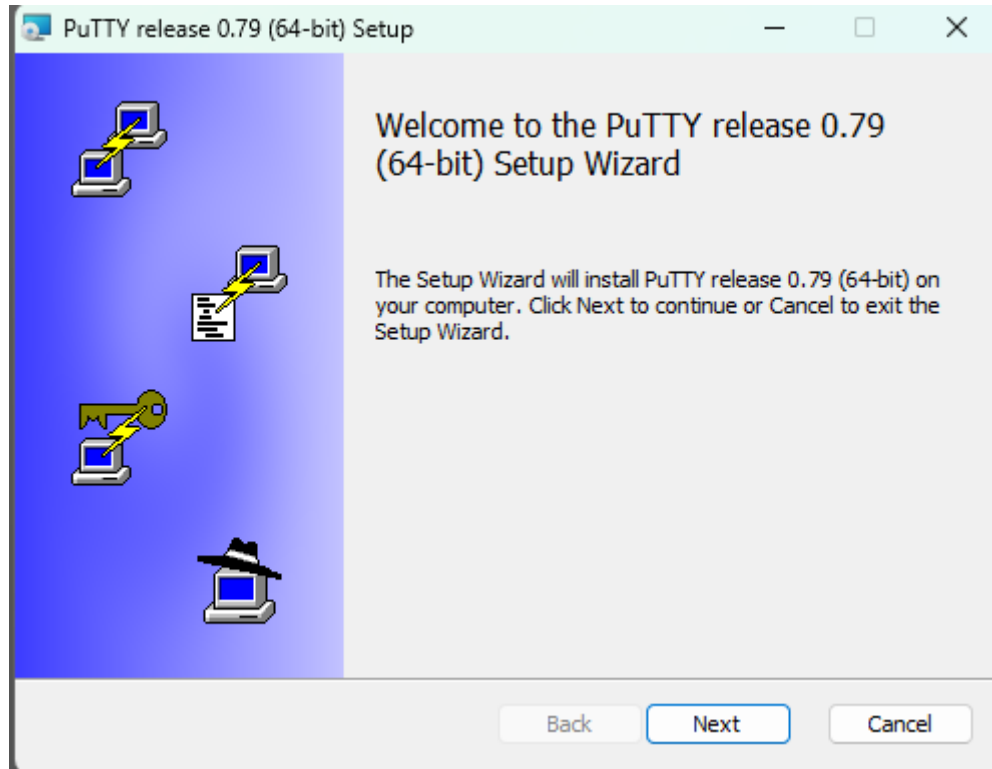
64-bit Arm: [putty-arm64-0.79-installer.msi](#) ([signature](#))

32-bit x86: [putty-0.79-installer.msi](#) ([signature](#))

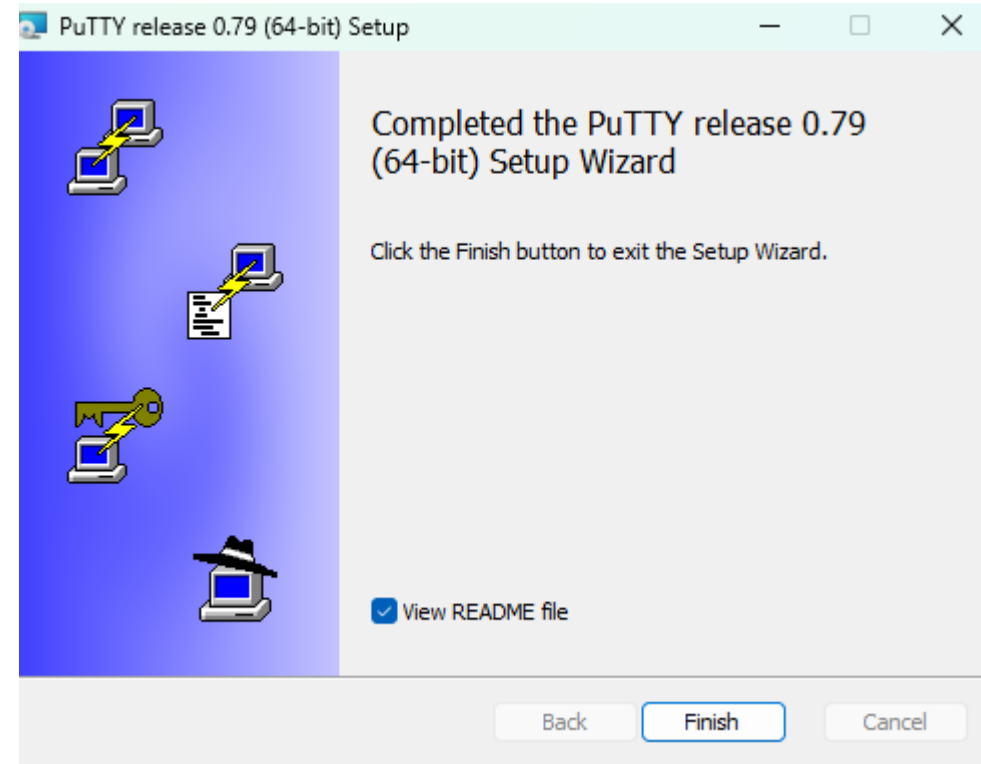
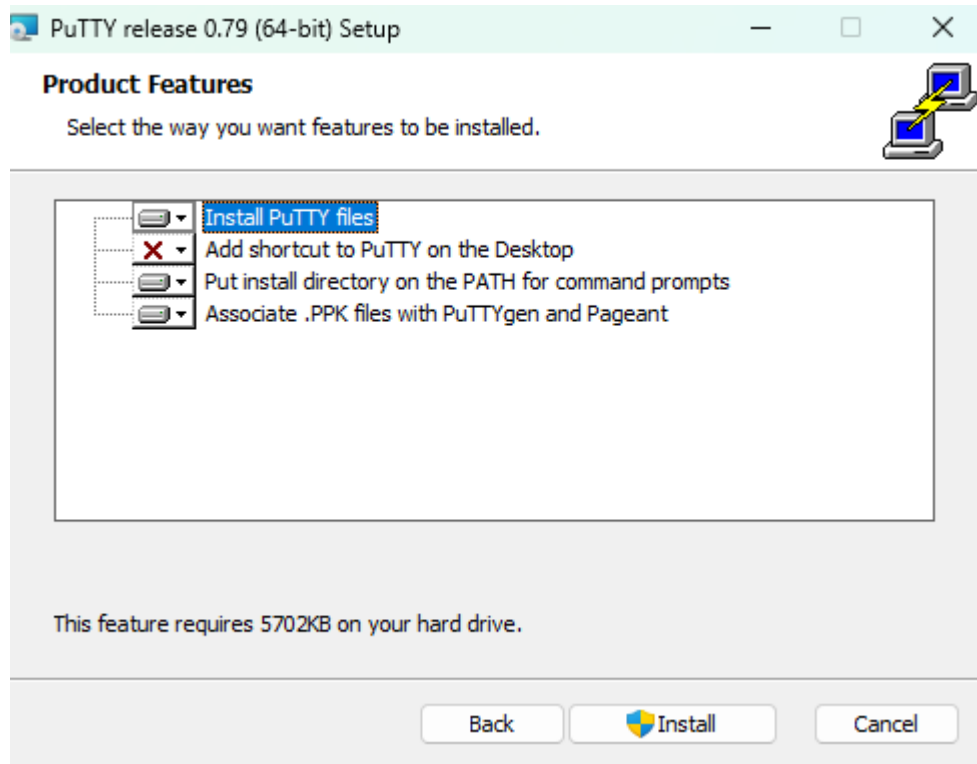
Unix source archive

.tar.gz: [putty-0.79.tar.gz](#) ([signature](#))

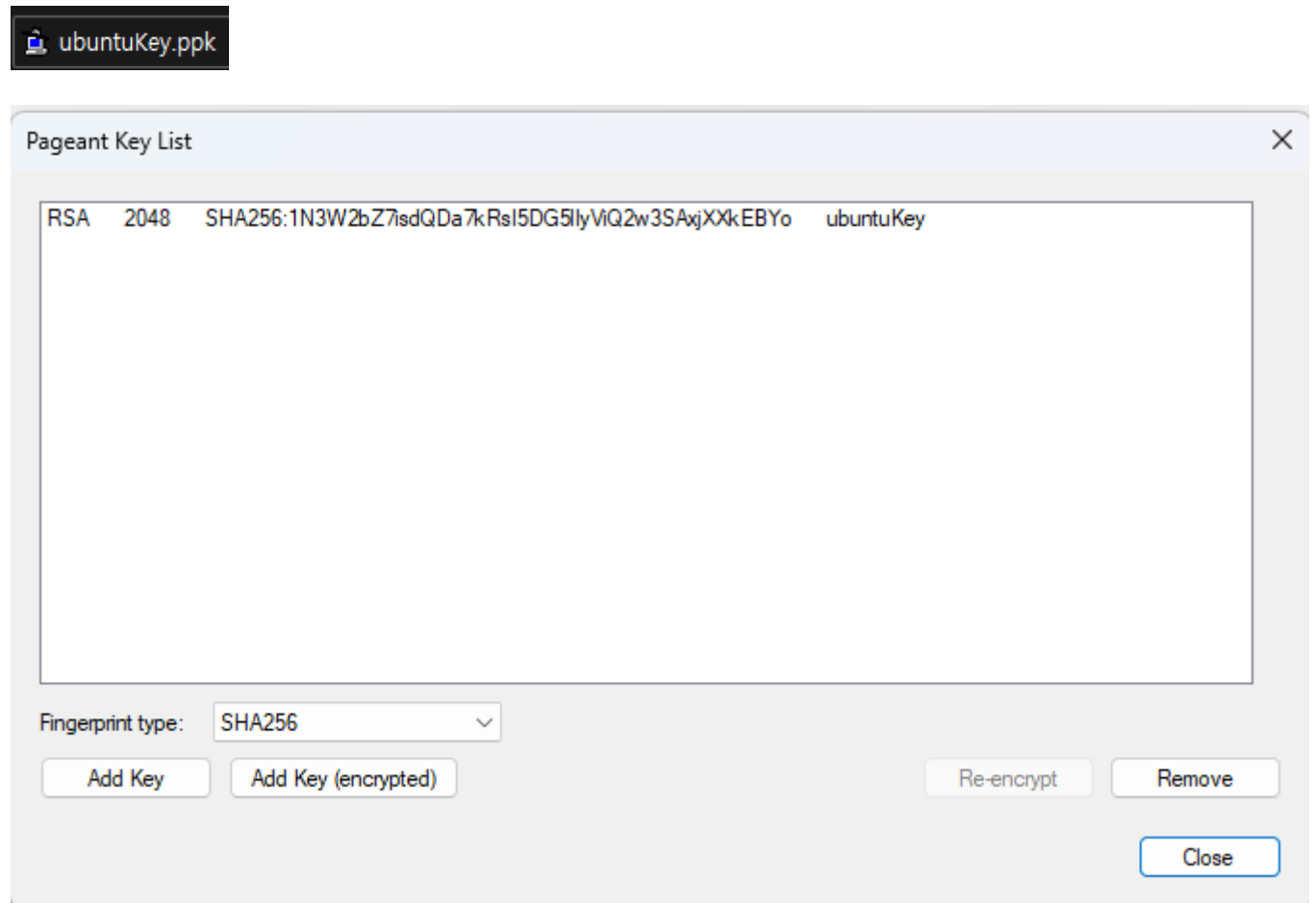
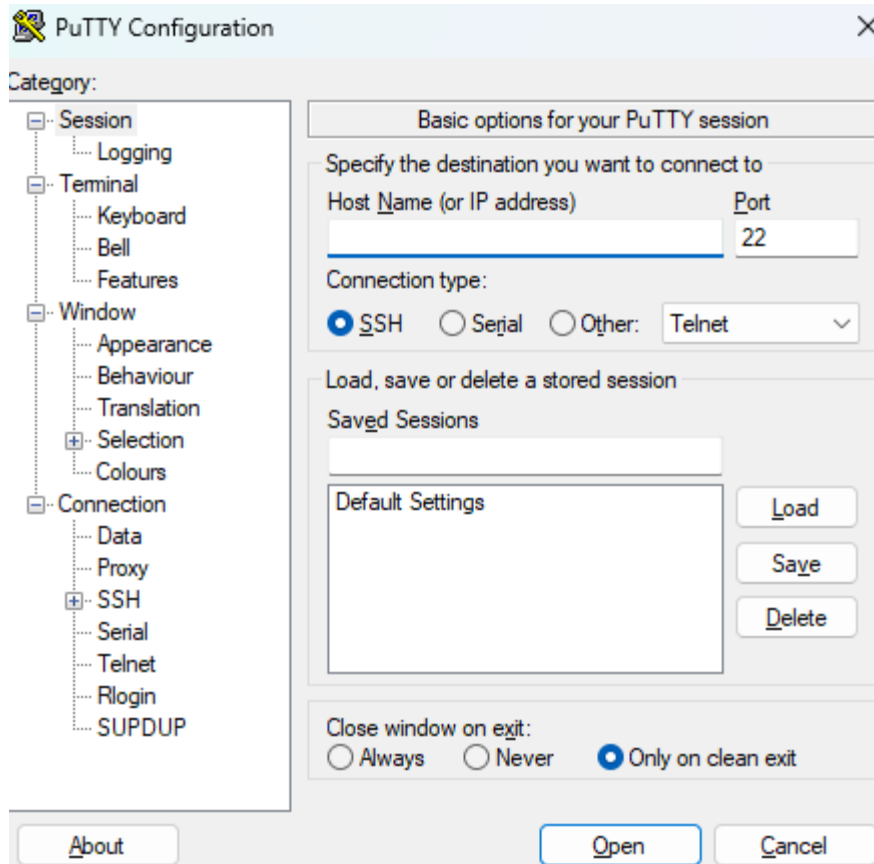
PuTTY 설치



PuTTY 설치



PuTTY 실행



PuTTY 접속

EC2 인스턴스 연결



Session Manager

SSH 클라이언트


EC2 직렬 콘솔


인스턴스 ID

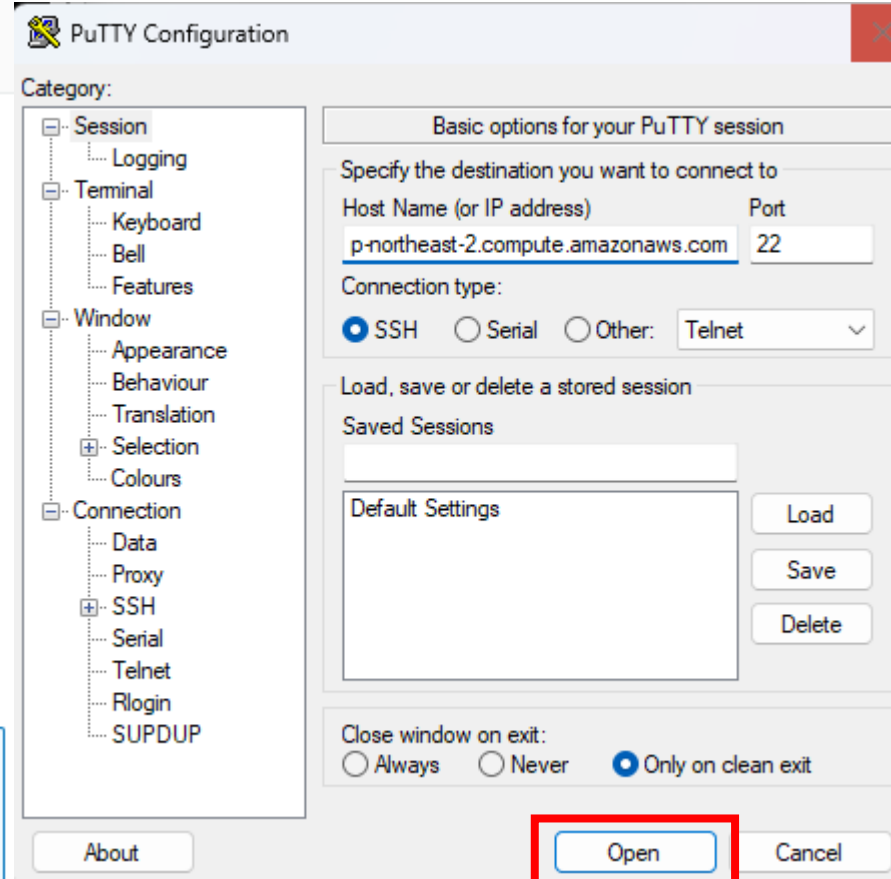
 i-04f6c17c1dea2eb61 (ubuntuServer)

1. SSH 클라이언트를 엽니다.
2. 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 ubuntuKey.pem입니다.
3. 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 없도록 합니다.
 `chmod 400 ubuntuKey.pem`
4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:
 `ec2-15-164-94-200.ap-northeast-2.compute.amazonaws.com`

예:

 `ssh -i "ubuntuKey.pem" ubuntu@ec2-15-164-94-200.ap-northeast-2.compute.amazonaws.com`

 참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.



SSH

```
ubuntu@ip-172-31-38-210: ~  
Using username "ubuntu".  
Authenticating with public key "ubuntuKey" from agent  
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1036-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Mon Sep 18 10:19:02 UTC 2023  
  
System load:  0.01          Processes:            101  
Usage of /:   21.3% of 7.57GB Users logged in:          0  
Memory usage: 24%          IPv4 address for eth0: 172.31.38.210  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Mon Sep 18 10:15:26 2023 from 13.209.1.61  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-38-210:~$
```

SSH

SSH(Secure Shell) : 보안 셸

- 원격지 호스트 컴퓨터에 접속하기 위해 사용되는 인터넷 프로토콜
- 네트워크 프로토콜 중 하나로 컴퓨터와 컴퓨터가 인터넷과 같은 Public Network를 통해서 서로 통신을 할 때 보안적으로 안전하게 통신을 하기 위해 사용하는 프로토콜

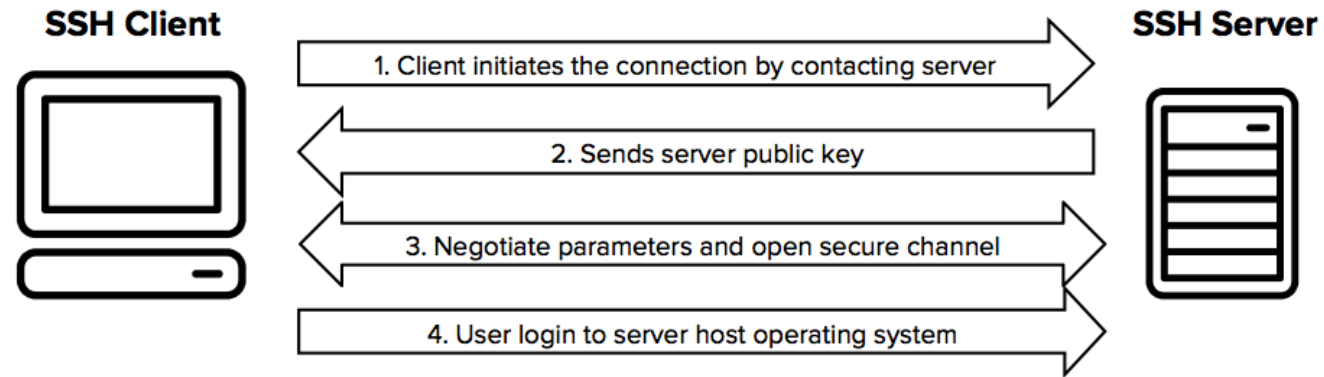
사용이유

- 기존 유닉스 시스템 셸에 접속하기 위해 사용하던 텔넷(Telnet)은 암호화가 이루어지지 않아 계정 정보 탈취의 위험성이 높았음
- 암호화 기능을 추가하여 나온 프로토콜

SSH 통신방식

SSH Key

- 서버에 비밀번호 대신 Key를 제출하는 방식
- 한 쌍의 키를 통해 접속하려는 클라이언트와 인증과정을 거침
- **Public Key**
 - 공개되어도 안전한키
 - 이 키를 통해 전송하기전 암호화
 - 암호화는 가능하지만 복호화는 불가능
- **Private Key**
 - 절대로 외부에 노출되어서는 안되는 Key
 - 본인의 컴퓨터 내부에 저장
 - 암호화된 메시지를 복호화 가능



기초 리눅스 명령 및 단축키

접속해제

\$ exit

\$ ^d → Ctrl + d

명령행 편집

문자삭제 → Delete or BackSpace

단어삭제 → Ctrl + w

행 삭제 → Ctrl + u

기초 리눅스 명령

명령의 구조와 사용

\$ ls

```
ubuntu@ip-172-31-38-210:~$ ls
```

\$ ls -a → 명령과 옵션을 사용 한 경우

```
ubuntu@ip-172-31-38-210:~$ ls -a
.  ..  .bash_logout  .bashrc  .cache  .profile  .ssh
```

\$ ls /tmp

```
ubuntu@ip-172-31-38-210:~$ ls /tmp
snap-private-tmp
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-ModemManager.service-kg6FRg
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-systemd-logind.service-GOn5Mg
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-systemd-resolved.service-Dppivh
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-systemd-timesyncd.service-AAyAc
f
```

\$ ls -a /tmp

```
ubuntu@ip-172-31-38-210:~$ ls -a /tmp
.
..
.ICE-unix
.Test-unix
.X11-unix
.XIM-unix
.font-unix
snap-private-tmp
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-ModemManager.service-kg6FRg
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-systemd-logind.service-GOn5Mg
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-systemd-resolved.service-Dppivh
systemd-private-bl2263ef125a4b4fa6b38a4e93cf5926-systemd-timesyncd.service-AAyAc
f
```


기초 리눅스 명령

현재 날짜와 시간 출력

\$ date

```
ubuntu@ip-172-31-38-210:~$ date  
Mon Sep 18 11:43:20 UTC 2023
```

현재 보이는 화면 삭제

\$ clear

매뉴얼

\$ man 명령

```
ubuntu@ip-172-31-38-210:~$ man clear  
clear(1)                                General Commands Manual          clear(1)  
  
NAME  
    clear - clear the terminal screen  
  
SYNOPSIS  
    clear [-T type] [-V] [-x]  
  
DESCRIPTION  
    clear clears your screen if this is possible, including its scrollback  
    buffer (if the extended "E3" capability is defined). clear looks in  
    the environment for the terminal type given by the environment variable  
    TERM, and then in the terminfo database to determine how to clear the  
    screen.  
  
    clear writes to the standard output. You can redirect the standard  
    output to a file (which prevents clear from actually clearing the  
    screen), and later cat the file to the screen, clearing it at that  
    point.  
  
OPTIONS  
    -T type  
        indicates the type of terminal. Normally this option is unneces-  
        sary, because the default is taken from the environment variable  
        TERM. If -T is specified, then the shell variables LINES and COLD-  
        UMNS will also be ignored.  
  
    -V    reports the version of ncurses which was used in this program, and  
        exits. The options are as follows:
```

리눅스의 파일과 디렉터리

- 리눅스는 유닉스의 시스템을 이어받아 시스템과 관련된 정보와 하드웨어 같은 장치를 모두 파일로 관리

파일의 종류

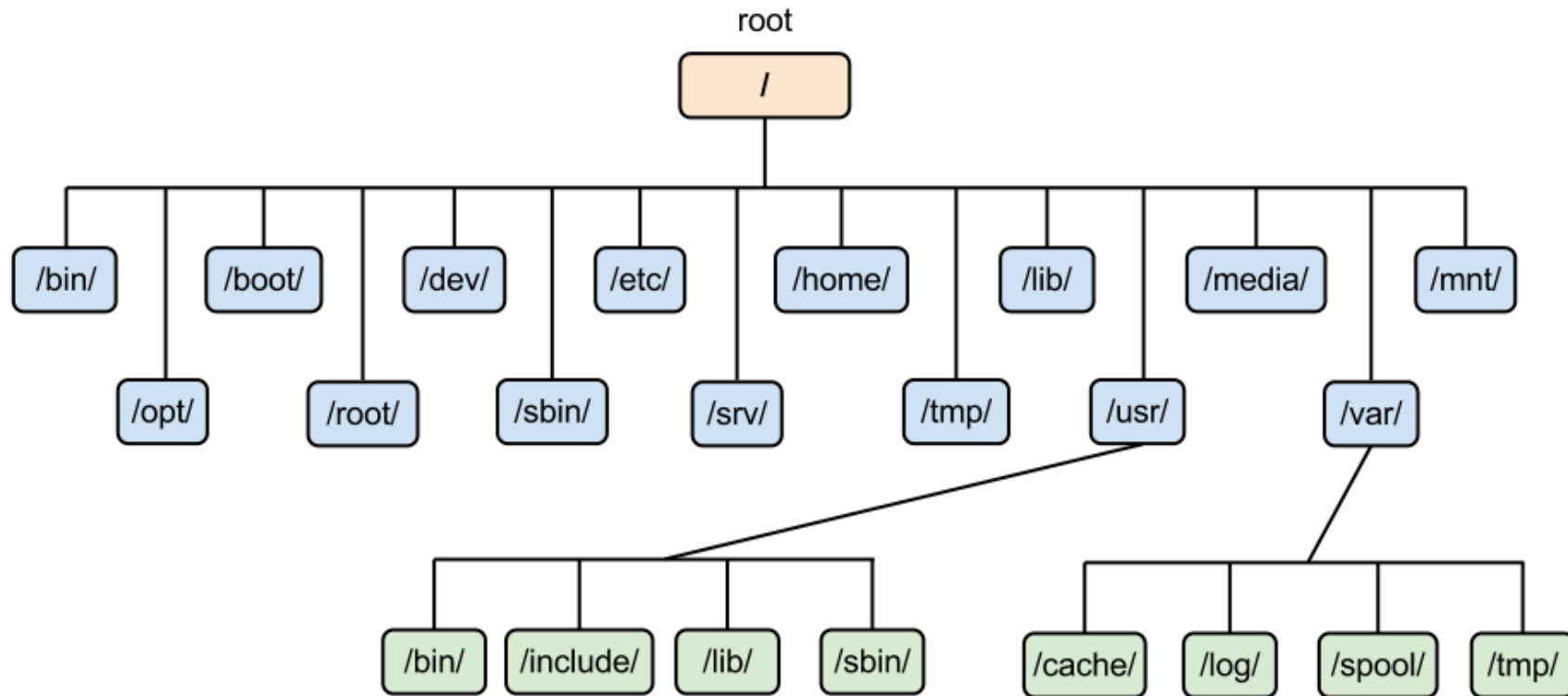
- 일반파일 : 데이터를 저장하는데 주로 사용(텍스트 파일, 실행파일, 이미지 파일 등)
- 디렉터리 : 해당 디렉터리에 저장된 파일, 하위 디렉터리에 대한 정보가 저장
- 심볼릭 링크 : 원본파일을 대신하도록 다른 파일명으로 지정한것(바로가기)
- 장치파일 : 하드디스크나 키보드 같은 장치를 파일로 취급한 것(/dev 디렉터리 아래에 주로 위치)

```
ubuntu@ip-172-31-38-210:~$ ls /
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
boot  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
ubuntu@ip-172-31-38-210:~$ file /bin
/bin: symbolic link to usr/bin
ubuntu@ip-172-31-38-210:~$ file /home
/home: directory
```

```
ubuntu@ip-172-31-38-210:~$ ls /dev
autofs      loop0      rtc        tty21      tty42      tty63      vcsa4
block       loop1      rtc0       tty22      tty43      tty7        vcsa5
btrfs-control  loop2      shm        tty23      tty44      tty8        vcsa6
char        loop3      snapshot   tty24      tty45      tty9        vcsu
console     loop4      snd        tty25      tty46      ttyS0       vcsu1
core        loop5      stderr     tty26      tty47      ttyS1       vcsu2
cpu         loop6      stdin      tty27      tty48      ttyS2       vcsu3
cpu_dma_latency  loop7      stdout     tty28      tty49      ttyS3       vcsu4
cuse        mapper     tty        tty29      tty5       ttyprintk   vcsu5
disk        mcelog     tty0       tty3       tty50      udmabuf     vcsu6
dma_heap    mem        tty1       tty30      tty51      uinput      vfio
ecryptfs    mqueue     tty10      tty31      tty52      urandom     vga_arbiter
fd          net        tty11      tty32      tty53      vcs         vhost-net
full        null       tty12      tty33      tty54      vcs1        vhost-vsock
fuse        nvram      tty13      tty34      tty55      vcs2        xen
hpet        port       tty14      tty35      tty56      vcs3        xvda
hugepages   ppp        tty15      tty36      tty57      vcs4        xvda1
hwrng       psaux      tty16      tty37      tty58      vcs5        xvda14
initctl     ptmx       tty17      tty38      tty59      vcs6        xvda15
input       pts        tty18      tty39      tty6       vcsa        zero
kmsg        random     tty19      tty4       tty60      vcsa1       zfs
log         rfkill     tty2       tty40      tty61      vcsa2
loop-control  root      tty20      tty41      tty62      vcsa3
```

리눅스 파일 계층 구조

- 리눅스에서는 파일을 효율적으로 관리하기 위해 디렉터리를 계층적으로 구성
- 이를 트리(Tree)구조라 함
- 루트(root)디렉터리 : 최상단에 위치하며 "/" 로 표기



리눅스 파일 계층 구조

디렉터리	기능
dev	장치파일이 담긴 디렉터리
home	사용자 홈 디렉터리가 생성되는 디렉터리
media	DVD/CD나 USB 외부장치를 연결하는 디렉터리
opt	추가 패키지가 설치되는 디렉터리
root	root 계정의 홈 디렉터리, 루트(/) 디렉터리와 다른 것
sys	리눅스 커널과 관련된 파일이 있는 디렉터리
usr	기본 실행 파일과 라이브러리 파일, 헤더 파일 등 많은 파일이 있음 참고로 usr은 'Unix System Resource'의 약자
boot	부팅에 필요한 커널 파일
etc	리눅스 설정을 위한 각종 파일
lost+found	파일 시스템에 문제가 발생하여 복구할 경우, 문제가 되는 파일이 저장되는 디렉터리
mnt	파일 시스템을 임시로 마운트 하는 디렉터리
proc	프로세스 정보 등 커널 관련 정보가 저장되는 디렉터리
run	실행 중인 서비스와 관련된 파일이 저장 됨
srv	FTP나 Web 등 시스템에서 제공하는 서비스의 데이터가 저장됨
tmp	시스템 사용 중에 발생하는 임시 데이터가 저장됨, 이 디렉터리에 있는 파일은 재시작 하면 모두 삭제
var	시스템 운영 중에 발생하는 데이터나 로그 등 내용이 자주 바뀌는 파일이 주로 저장 됨

경로명 특징

절대 경로명

- 반드시 / 로 시작
- 루트 디렉터리부터 시작하여 특정 파일이나 디렉터리 위치에 이르기까지 중간에 있는 모든 디렉터리의 이름을 표시
- 특정 위치를 가리키는 절대 경로명은 항상 동일

상대 경로명

- /이외의 문자로 시작
- 현재 디렉터리를 기준으로 서브 디렉터리로 내려가면 그냥 서브 디렉터리 명으로 시작
- 현재 디렉터리를 기준으로 상위 디렉터리로 가려면 ..(마침표 두 개)로 시작
- 상대 경로명은 현재 디렉터리가 어디냐에 따라 달라짐

파일과 디렉터리 이름 규칙

- 파일과 디렉터리의 이름에는 /를 사용할 수 없다. /는 경로명에서 구분자로 사용하기 때문이다.
- 파일과 디렉터리의 이름에는 알파벳, 숫자, 하이픈(-), 언더스코어(_), 마침표(.)를 사용 할 수 있다.
- 파일과 디렉터리의 이름에는 /를 제외한 특수문자를 사용할 수 있으나 >, |, :, &와 공백문자의 사용은 피하는 것이 좋다.
- 파일과 디렉터리 이름의 영문은 대문자와 소문자를 구별하여 다른 글자로 취급한다.
- 파일과 디렉터리의 이름이 .(마침표)로 시작하면 숨김 파일로 간주한다.
- 파일 이름에는 널 문자("\0")를 사용하지 않는다.

예시

- 좋은 이름 예 : game.txt, hello.c, test, sample11
- 나쁜 이름 예 : &game, *dir, my home, game₩
- 사용할 수 없는 이름 예 : myhome/, /test, bad/name

디렉터리 관련 명령

현재 디렉터리 확인

\$ pwd

디렉터리 이동

\$ cd [디렉터리]

디렉터리 관련 명령

디렉터리 내용 확인

\$ ls

\$ ls [옵션] [디렉터리]

- a : 숨김 파일을 포함하여 모든 파일의 목록을 출력한다.
- d : 디렉터리 자체의 정보 출력
- i : 첫 번째 행에 inode 번호를 출력
- l : 파일의 상세정보 출력
- A : .(마침표)와 ..(마침표 두 개)를 제외한 모든 파일 목록을 출력
- F : 파일의 종류를 표시(*: 실행파일, /:디렉터리, @:심볼릭 링크)
- L : 심볼릭 링크 파일의 경우 원본 파일의 정보를 출력
- R : 하위 디렉터리의 목록 까지 출력

디렉터리 관련 명령

디렉터리 생성

\$ mkdir [옵션] [디렉터리]

-p : 하위 디렉터를 계층적으로 생성 할 때 중간 단계의 디렉터리가 없으면 자동으로 중간 단계 디렉터를 생성하고 지정한 디렉터를 생성

디렉터리 삭제

\$ rmdir [옵션] [디렉터리]

-p : 지정한 디렉터를 삭제하고, 그 디렉터리의 부모 디렉터리가 빈 디렉터리일 경우 부모 디렉터리도 자동 삭제

파일 관련 명령

파일 내용을 연속으로 출력

\$ cat [옵션] [파일]

-n : 행 번호를 붙여서 출력

파일 내용을 화면단위로 출력

\$ more [옵션] [파일]

+행 번호 : 출력을 시작할 행 번호를 지정

\$ less [파일]

파일 내용을 뒷부분 출력

\$ tail [옵션] [파일]

+행 번호 : 지정한 행부터 끝까지 출력

-숫자 : 화면에 출력할 행의 수를 지정(기본 값 10)

-f : 파일 출력을 종료하지 않고 주기적으로 계속 출력

파일 관련 명령

파일 복사 : 파일1(원본) 파일2로 복사

\$ cp [파일1(디렉터리1)] [파일2(디렉터리2)]

-i : 파일 2가 이미 존재하면 덮어쓸 것인지 물어봄

-r : 디렉터리를 복사할 때 지정

파일 이동 및 파일명 변경

\$ mv [파일1(디렉터리1)] [파일2(디렉터리2)]

-i : 파일 2(디렉터리2)가 이미 존재하면 덮어쓸 것인지 물어봄

파일 삭제

\$ rm [옵션] [파일 또는 디렉터리]

-i : 파일을 정말 삭제할 것인지 확인

-r : 디렉터리를 삭제 할 때 지정

파일 관련 명령

하드링크 만들기

\$ ln [옵션] [원본 파일] [링크 파일]

-s : 심볼릭 링크 파일을 생성

빈 파일 만들기, 접근/수정 시간 변경

\$ touch [-acm] [-r ref_file | -t time] [파일]

-a : 접근 시간만 변경

-m : 수정 시간만 변경

-t : [[CC]YY]MMDDhhmm[.ss]시간을 직접 입력

파일 관련 명령

파일 내용 검색

\$ grep [옵션] [패턴] [파일]

-i : 대문자, 소문자를 모두 검색

-l : 지정한 패턴이 포함된 파일명을 출력

-n : 행 번호 출력

파일 관련 명령

파일 찾기

\$ find [경로] [검색조건] [동작]

- 검색조건
 - name filename : 파일 명으로 검색
 - type 파일 종류 : 파일 종류로 검색
 - user loginID : 지정한 사용자가 소유한 모든 파일을 검색
 - perm 접근 권한 : 지정한 사용 권한과 일치하는 파일을 검색
- 동작
 - exec 명령 {} \; : 검색된 파일에 명령을 실행
 - ok 명령 {} \; 사용자의 확인을 받아서 명령을 실행
 - print : 검색된 파일의 절대 경로 명을 화면에 출력(기본동작)
 - ls : 검색결과를 긴 목록 형식으로 출력

파일 관련 명령

명령의 위치 찾기

\$ whereis [옵션] [파일]

-b : 바이너리 파일만 검색

-m : 매뉴얼 파일만 검색

-s : 소스 파일만 검색

\$ which [명령]

- 앨리어스나 PATH 환경 변수로 지정된 경로에서 파일을 찾음