



# Compiler (컴파일러)

- Project1

# Discord 질문 전용 채널 운영

---

- Discord 채널 (JBNU\_Compiler\_2025\_1)
  - <https://discord.gg/H3rmwFd4> (초대링크)
  - 공지사항 채널: 수업 및 프로젝트 관련 공지 게시 (lms 에도 동일 공지)
  - 수업-질문 채널: 수업 관련 질문
  - 프로젝트-질문 채널: 프로젝트 관련 질문
- 되도록 discord 에서 소통 추천!!
  - 혹은 메일로 문의: hyukwoo.park@jbnu.ac.kr

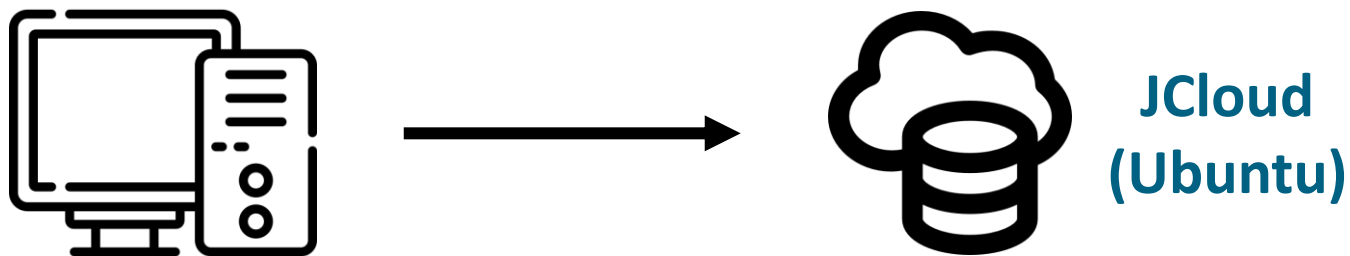
## Project1

---

[프로젝트 개발 환경 설정]

# JCloud

- 클라우드 상의 가상 머신 (Ubuntu) 제공
  - [jcloud.jbnu.ac.kr](http://jcloud.jbnu.ac.kr) 접속 후 학교 구글 계정으로 로그인
  - KeyPair 만들기 (처음 한번만)
    - JCloud 접속할 때 마다 필요하므로 생성 후 다운로드하여 관리 필요!!
  - 인스턴스 만들기
    - Ubuntu 24.04 이미지로 생성
- 사용 가이드: <https://jhelper.jbnu.ac.kr/JCloud/1UserGuide>



# JCloud 생성

JCloud

students(113.198.66.75)

hyukwoo.park@jbnu.ac.kr

프로젝트

API 액세스

Compute

개요

인스턴스

이미지

키 페어

서버 그룹

볼륨

네트워크

인증

프로젝트 / Compute / 인스턴스

인스턴스

Instance ID =

필터

인스턴스 시작

인스턴스 삭제

기타 작업

5 항목 표시

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	europa	ubuntu.24.04.image.v2	10.0.0.23	micro	castle	Active	cpuzone-az	None	Running	5일, 22시간	스냅샷 생성
<input type="checkbox"/>	pluto	ubuntu.24.04.image.v2	10.0.0.84	micro	castle	Active	cpuzone-az	None	Running	5일, 22시간	스냅샷 생성
<input type="checkbox"/>	dmatch	ubuntu.24.04.image.v2	10.0.0.222	micro	castle	Active	cpuzone-az	None	Running	1주	스냅샷 생성
<input type="checkbox"/>	202346544	ubuntu.24.04.image.v2	10.0.0.86	micro	202346544-key	Active	nova	None	Running	1주, 4일	스냅샷 생성
<input type="checkbox"/>	compiler_test	ubuntu.24.04.image.v2	10.0.0.197	micro	hwpark-key	Active	nova	None	Running	1주, 5일	스냅샷 생성

# JCloud 생성

인스턴스 시작

세부 정보 \*

소스 \*

Flavor \*

네트워크 \*

네트워크 포트

보안 그룹

Key Pair

구성

서버 그룹

스케줄러 힌트

메타데이터

인스턴스 소스는 인스턴스를 생성할 때 사용되는 템플릿입니다. 인스턴스의 스냅샷 이미지를 사용할 수 있습니다. (이미지 스냅샷), 볼륨 또는 볼륨스냅샷 (활성화된 경우). 또한 새 볼륨을 생성할 때 영구적인 스토리지를 사용하도록 선택할 수 있습니다.

부팅 소스 선택

Image

새로운 볼륨 생성

예

아니오

할당됨

0 항목 표시

이름

업데이트 완료

크기

포맷

가시성

아래에서 사용 가능한 항목에서 선택

0 항목 표시

▼ 사용 가능 1

하나 선택



필터 또는 전체 텍스트 검색은 여기를 클릭하십시오.



1 항목 표시

이름

업데이트 완료

크기

포맷

가시성

▶ ubuntu.24.04.image.v2

12/31/24 4:59 AM

6.52 GB

QCOW2

공용



1 항목 표시

✕ Cancel

< 뒤로

Next >

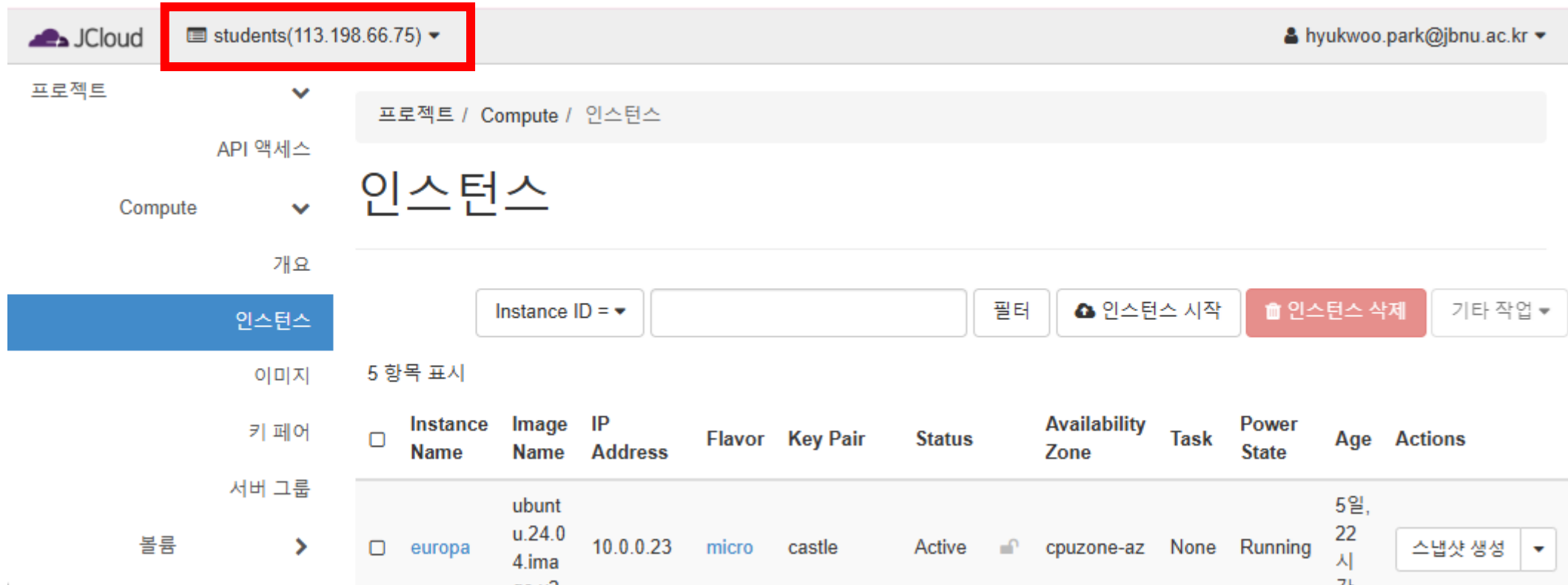
☁ 인스턴스 시작

# JCloud 생성

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	europa	ubuntu.24.04.image.v2	10.0.0.23	micro	castle	Active		cpuzone-az	None	Running	5일, 22시간	스냅샷 생성 ▼
<input type="checkbox"/>	pluto	ubuntu.24.04.image.v2	10.0.0.84	micro	castle	Active		cpuzone-az	None	Running	5일, 22시간	스냅샷 생성 ▼
<input type="checkbox"/>	dmatch	ubuntu.24.04.image.v2	10.0.0.222	micro	castle	Active		cpuzone-az	None	Running	1주	스냅샷 생성 ▼
<input type="checkbox"/>	202346544	ubuntu.24.04.image.v2	10.0.0.86	micro	202346544-key	Active		nova	None	Running	1주, 4일	스냅샷 생성 ▼
<input type="checkbox"/>	compiler_test	ubuntu.24.04.image.v2	10.0.0.197	micro	hwpark-key	Active		nova	None	Running	1주, 5일	스냅샷 생성 ▼

# JCloud 접속

- <https://jhelper.jbnu.ac.kr/JCloud/2PortGuide> 참고
- IP 확인
  - JCloud 왼쪽 상단에서 개인별 할당된 IP 확인



The screenshot shows the JCloud web interface. At the top, there is a header bar with the JCloud logo, a dropdown menu showing 'students(113.198.66.75)' (highlighted with a red box), and a user profile 'hyukwoo.park@jbnu.ac.kr'. Below the header, there is a sidebar with navigation links: '프로젝트', 'API 액세스', 'Compute', '개요', '인스턴스' (selected), '이미지', '키 페어', '서버 그룹', and '볼륨'. The main content area shows the '인스턴스' (Instances) page. It includes a breadcrumb '프로젝트 / Compute / 인스턴스' and a title '인스턴스'. Below the title, there is a search bar 'Instance ID =', a '필터' button, and three action buttons: '인스턴스 시작', '인스턴스 삭제', and '기타 작업'. A table of instances is displayed, showing 5 items. The first instance is 'europa' with IP address 10.0.0.23.

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
europa	ubuntu.24.04.ima	10.0.0.23	micro	castle	Active	cpuzone-az	None	Running	5월, 22 시	스냅샷 생성



# JCloud 접속

## • 포트포워딩

- 포트 포워딩 개념: 공용 IP의 특정 포트를 내부 인스턴스의 지정 port로 연결하여 접근.
  - (예제) 113.198.66.74:19xxx -> capstone 프로젝트의 xxx번 인스턴스의 7777 포트
- 접속하려는 인스턴스의 내부 네트워크 번호 확인
  - 내부 IP의 마지막 숫자(10.0.0.xxx)를 세 자리로 맞춰 사용
  - (예제) 내 인스턴스의 내부 IP가 10.0.0.231 인 경우, 231 이 번호
- 아래 포트포워딩 참고하여, 포트 번호 완성
  - (예제) 231 인스턴스의 8080 port로 접근하려면, 10231

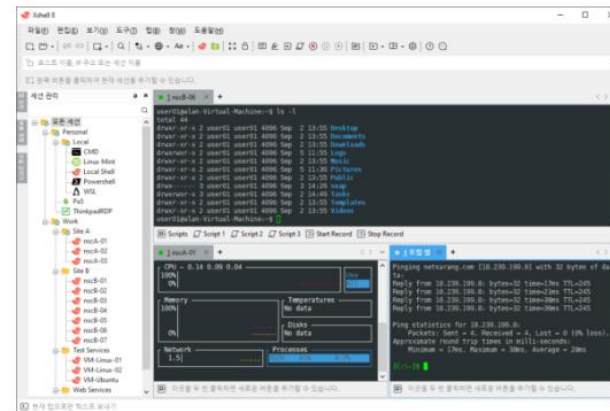
```
- 80 (HTTP): 18xxx  
- 443 (HTTPS): 17xxx  
- 7777 (SSH): 19xxx  
- 3000 (범용): 13xxx  
- 8080 (범용): 10xxx
```

# JCloud 접속 - 터미널

- SSH 접속: xshell 이용 (<https://jhelper.jbnu.ac.kr/JCloud/3SSHGuide> **참고**)
- 파일전송: xftp 이용
- <https://www.netsarang.com/ko/free-for-home-school/> 에서 다운로드 및 설치
- 학교 메일 인증(jbnu.ac.kr) 후 사용



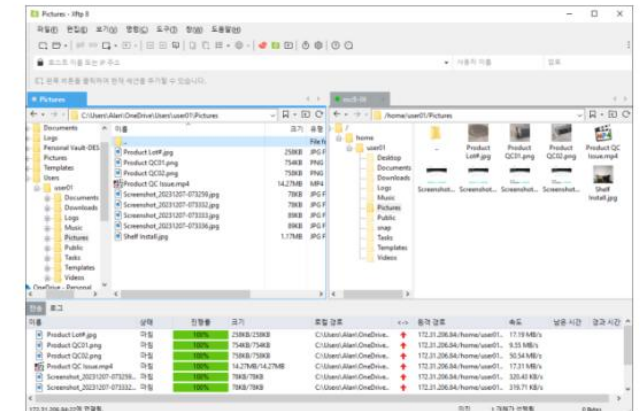
업계 최고의 강력한 SSH 클라이언트



다운로드



네트워크를 통한 수고스럽지 않은 파일 전송



다운로드

# SSH 접속하기: 접속 주소 입력 (중요!!)

- 포트포워딩 주소에 맞게 호스트 IP와 포트 번호 입력
  - ① 프로젝트 IP 확인 → “호스트” 칸에 입력
  - ② 인스턴스 번호 확인 → “포트번호” 칸에 입력

(참고) 일반적인 주소 표시 형식

113.198.66.75 : 19231

호스트 주소 (IP)    포트 번호

openstack ① students(113.198.66.75)

프로젝트 / Compute / 인스턴스

호스트 IP와 포트 번호  
인스턴스

Instance Name	Image Name	IP Address	Flavor
12345	ubuntu.2004.image	② 10.0.0.231	jcode.s1.flavor

새 세션 등록 정보

범주(C):

- 연결
  - 사용자 인증
    - 로그인 프롬프트
    - 로그인 스크립트
  - SSH
    - 보안
    - 터널링
    - SFTP
  - TELNET
  - RLOGIN
  - SERIAL
  - 포록시
  - 연결 유지
- 터미널
  - 키보드
  - VT 모드
  - 고크

연결

일반

이름(N): JCloud 12345

프로토콜(P): SSH

호스트(H): 113.198.66.75

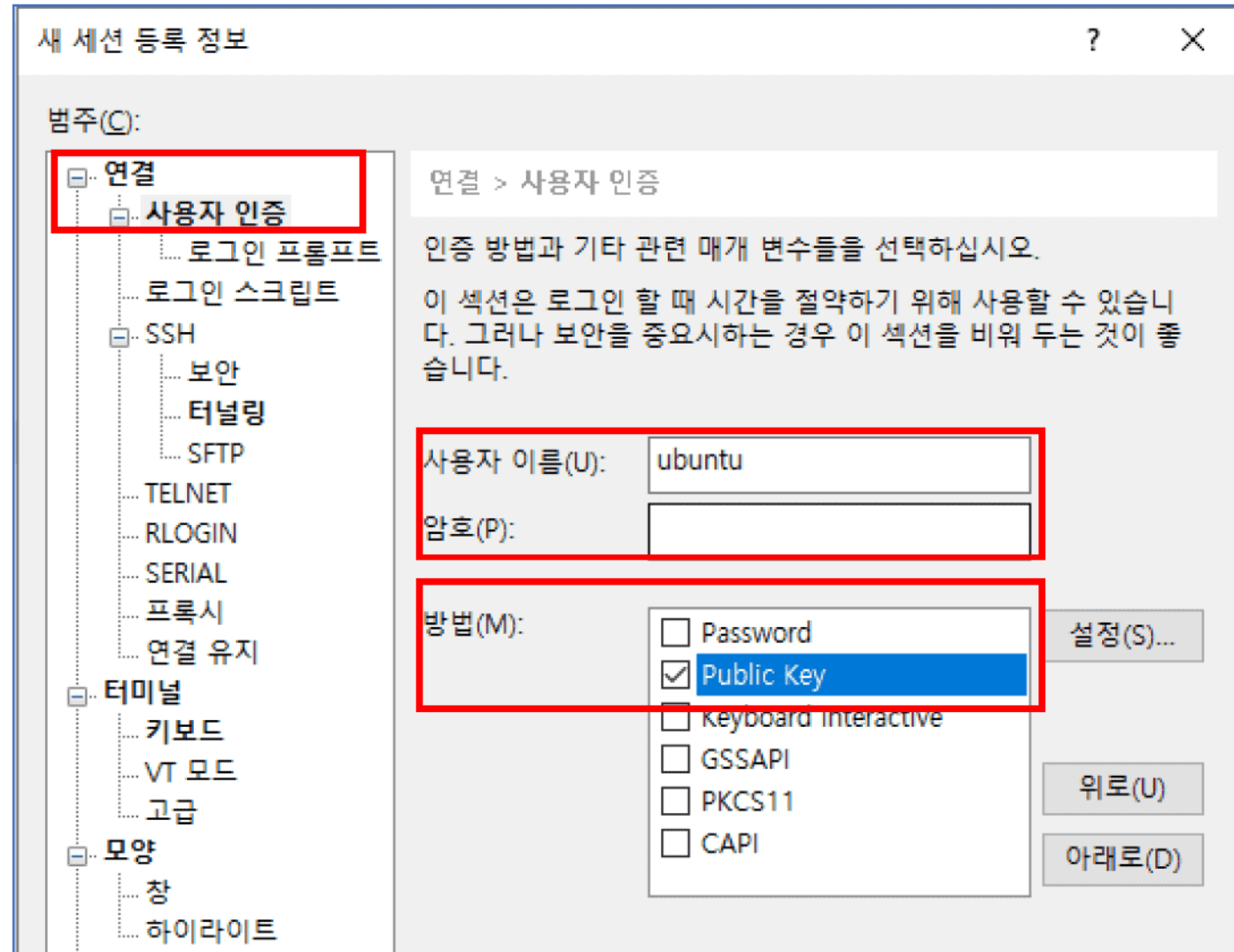
포트 번호(P): 19231

설명(D):

다시 연결

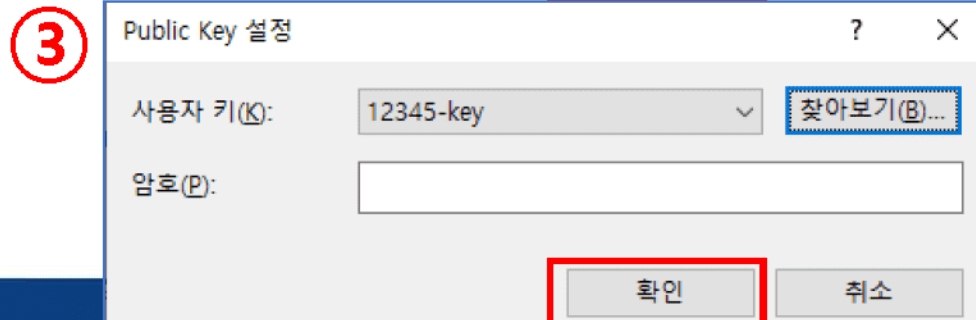
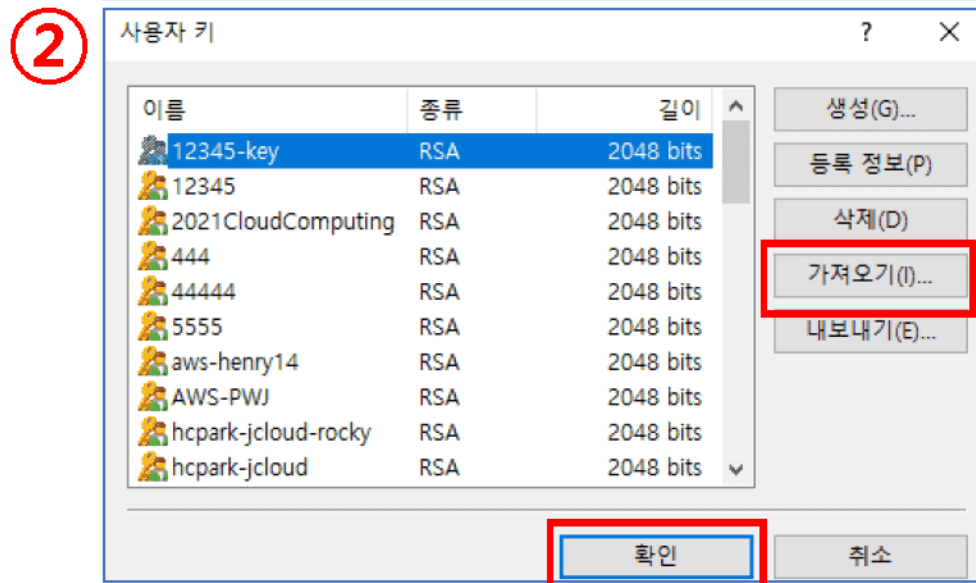
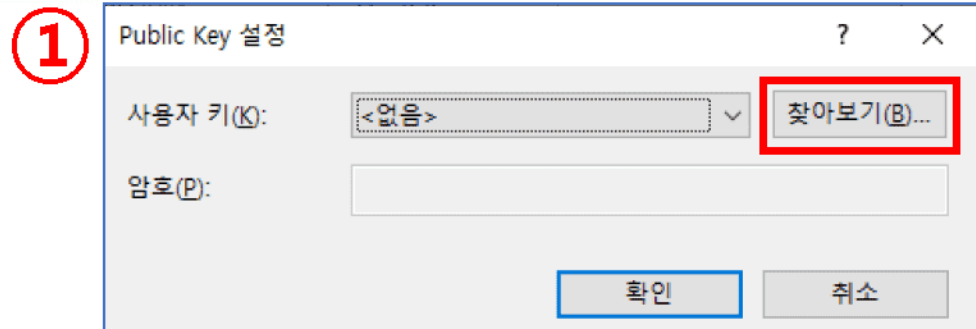
# SSH 접속하기: 사용자 인증 정보 입력 (중요!!)

- 사용자 인증 메뉴 선택
- 사용자 이름: “ubuntu”
- 암호: 비워두기
- 방법: “Public Key” 만 선택
  - “Public Key” 를 클릭하고,
  - 오른쪽 “설정” 버튼 클릭



# SSH 접속하기: 사용자 인증 정보 입력 (중요!!)

- ① 화면에서 “찾아보기” 클릭
- ② 화면에서 “가져오기” 클릭 후,
  - 다운로드 받은 키페어 파일 “**학번-key.pem**” 선택
  - 왼쪽 목록에 등록된 본인 key를 선택한 후, “확인” 클릭
- ③ 사용자 키 선택된 것 확인
  - 암호는 비워둘 것
  - “확인” 클릭
- 모든 정보 입력 완료
- 세션 등록 정보 창에서 “확인” 클릭



# JCloud 접속: XFTP

## • 새 세션 정보 등록

- 개인 JCloud 호스트 및 포트 번호 입력
- 사용자 이름: “ubuntu” 입력
- 암호: 비워두기
- 방법: “Public Key” 만 선택
- 오른쪽 “설정” 클릭
- Public Key 설정에서 다운받은 키페어 가져오기
- 모든 정보 입력 완료
- 세션 등록 정보 창에서 “확인”
- 그 후 저장된 세션으로 “연결”

새 세션 등록 정보

일반 옵션 고급

사이트

이름(N): JCloud

호스트(H): 113.198.66.75

프로토콜(R): SFTP

포트 번호(O): 19231

프록시 서버(X): <없음>

설명(D):

로그인

☐ 익명 로그인(L) ☐ SSH 키 에이전트(Xagent) 사용(G)

☐ 인증 프로파일 사용(A)

인증 프로파일(F): <지정하지 않음>

사용자 이름(U): ubuntu

암호(P):

방법(M): ☐ Password ☒ Public Key ☐ Keyboard interactive ☐ GSSAPI

설정(S)...

찾아보기(W)...

찾아보기(B)...

설정(I)...

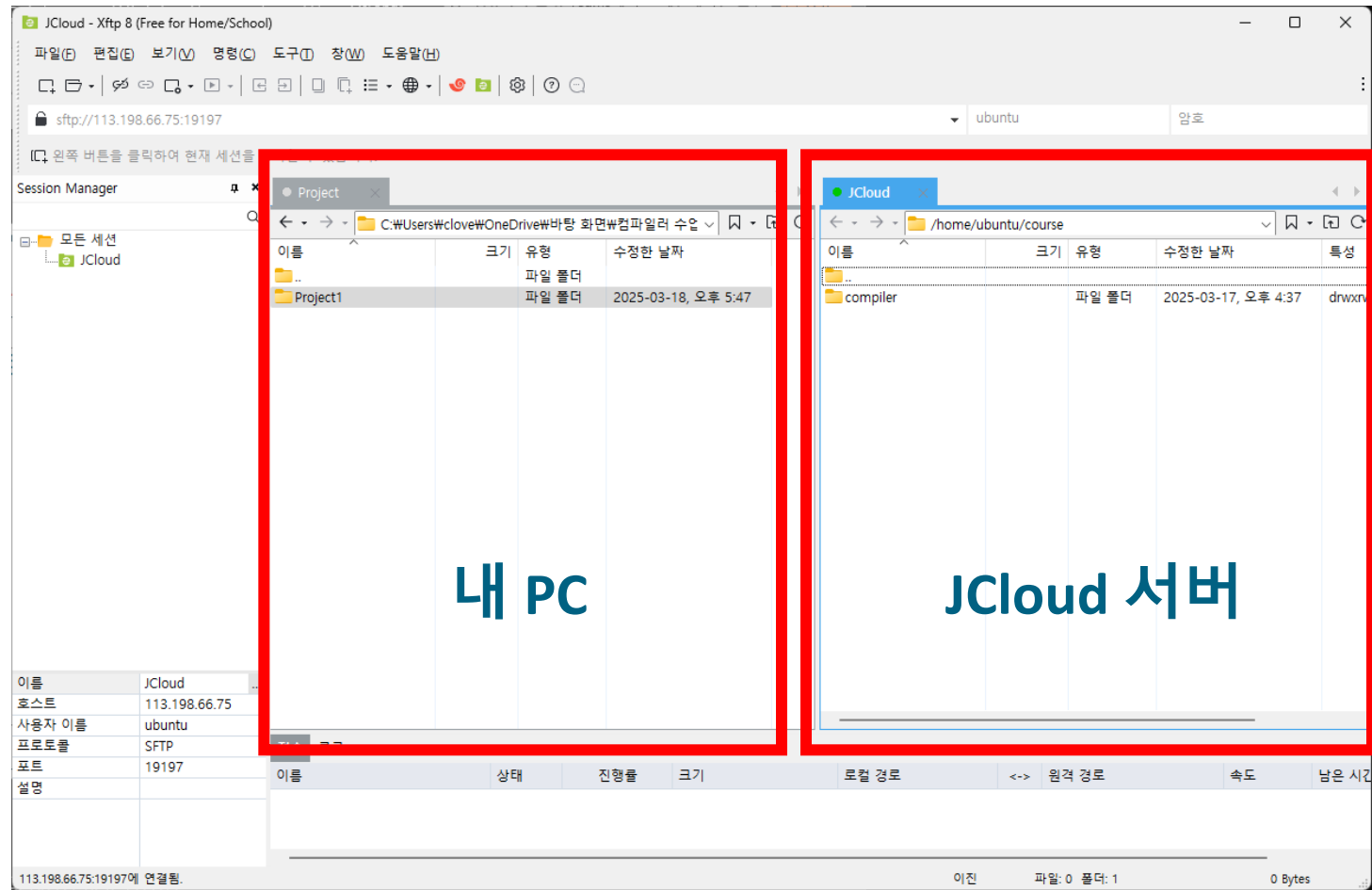
위로(U)

아래로(D)

연결 확인 취소

# JCloud 접속: XFTP

- JCloud 와 내 PC 간 파일 전송
  - 드래그&드롭
  - Ctrl+C/Ctrl+V 모두 가능





# vi Editor

---

- Xshell(SSH) 로 JCloud(Ubuntu) 접속하여 개발 시 vi Editor 필요

- vi (vim): Linux 환경에서 기본으로 제공되는 텍스트 편집기

- vi 설치 (대부분 Ubuntu에서 기본 설치되어 있음)

```
sudo apt install vim
```

- vi 로 파일 열기

```
vi example.txt
```

- 3가지 모드

- 명령 모드
    - 입력 모드
    - 명령 라인 모드



# vi Editor

---

- 명령 모드

- vi 실행 시 기본적으로 명령 모드
- 방향키를 통해 커서 이동

- 입력 모드

- 텍스트를 입력할 수 있는 모드
- 명령 모드에서 **i 키** 누르면 입력 모드로 전환
- 입력 모드에서 **ESC 키** 누르면 명령 모드로 전환

# vi Editor

- 명령 라인 모드

- 파일 저장, 종료 등의 작업 수행하는 모드
- 명령 모드에서 **:키** 누르면 명령 라인 모드로 전환

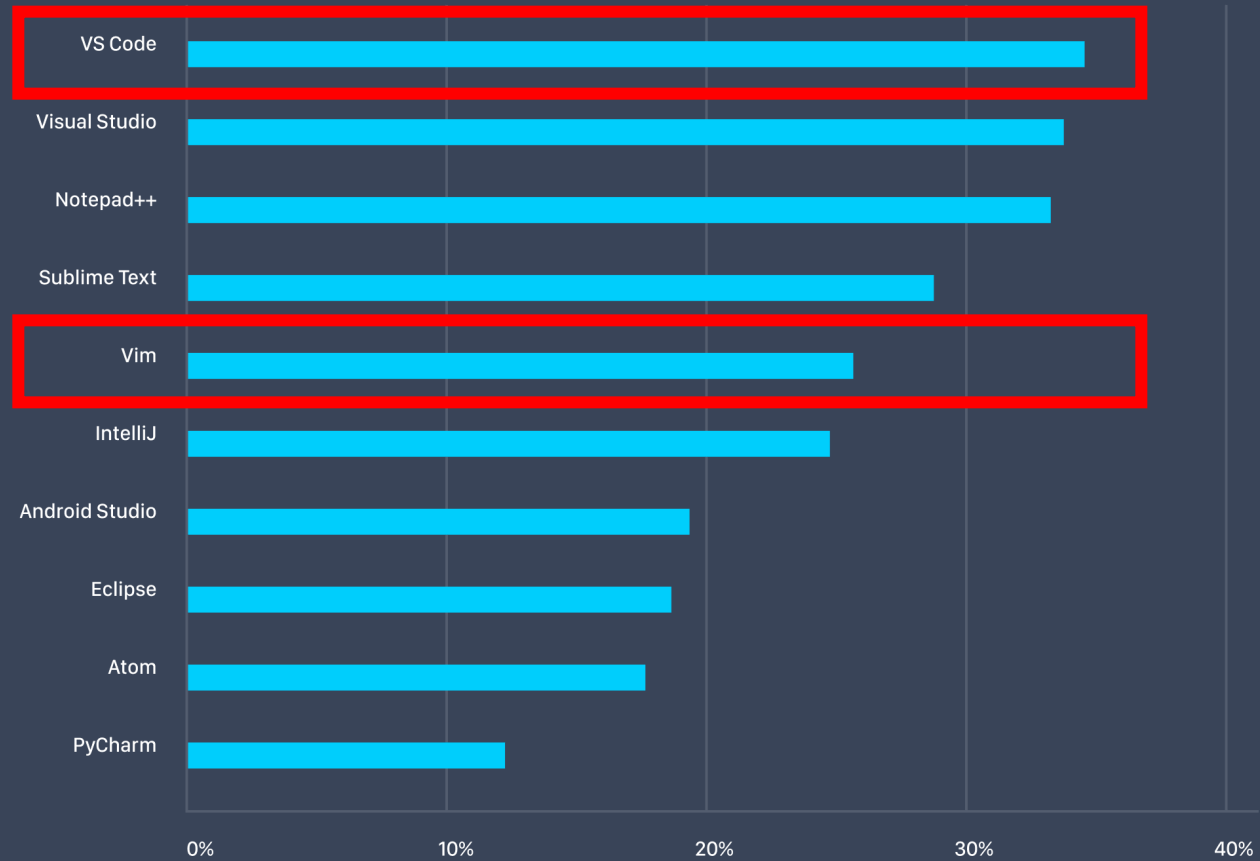
명령어	설명
:w	파일 저장
:q	vi 종료 (저장되지 않았을 경우 종료 불가)
:wq 또는 zz	저장 후 종료
:q!	저장하지 않고 강제 종료

- vi 참고 자료

- <https://blockdmask.tistory.com/25>
- <https://withcoding.com/112>

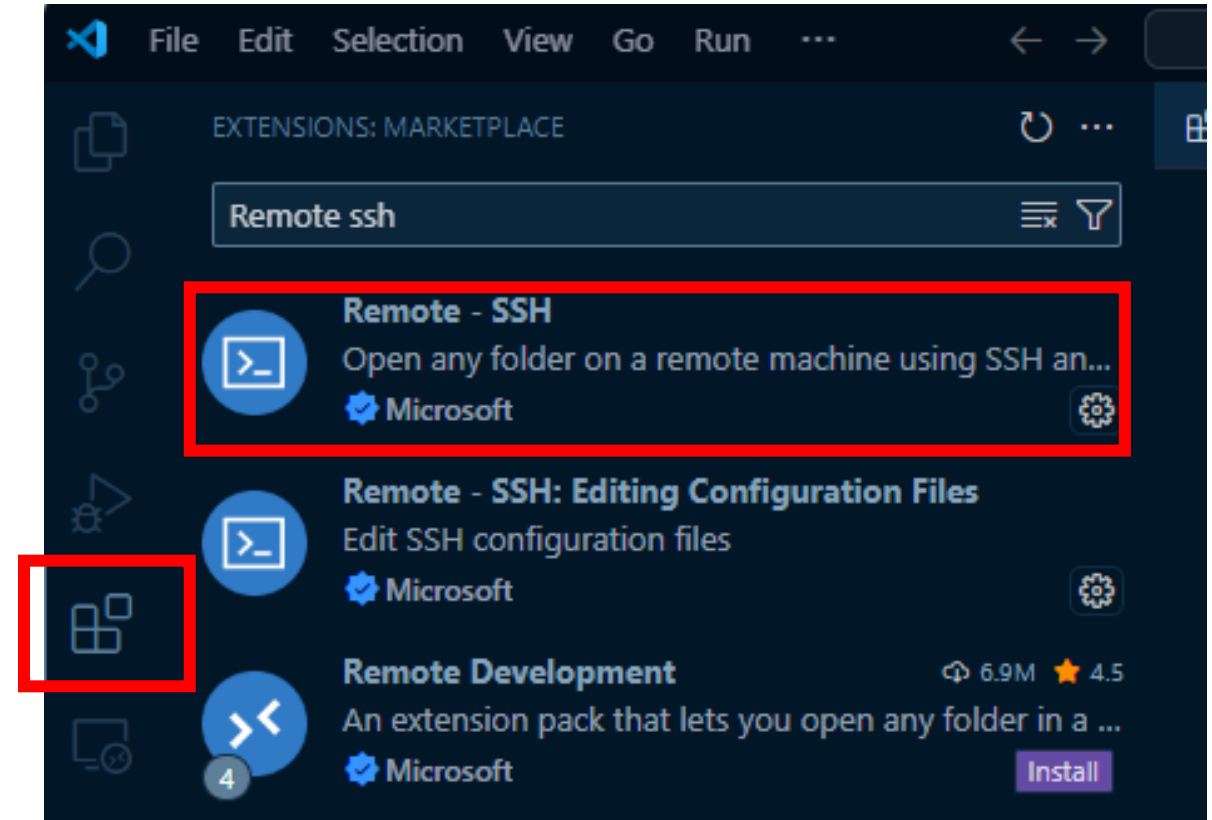
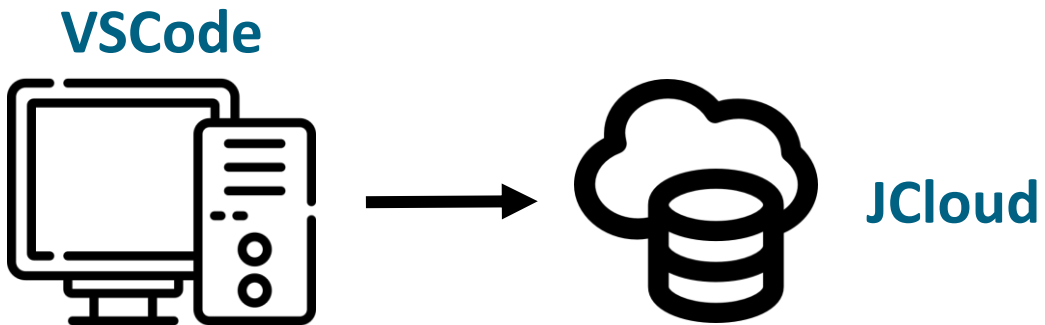
# VSCode & vi(vim) Editor

Most Popular Developer Environments



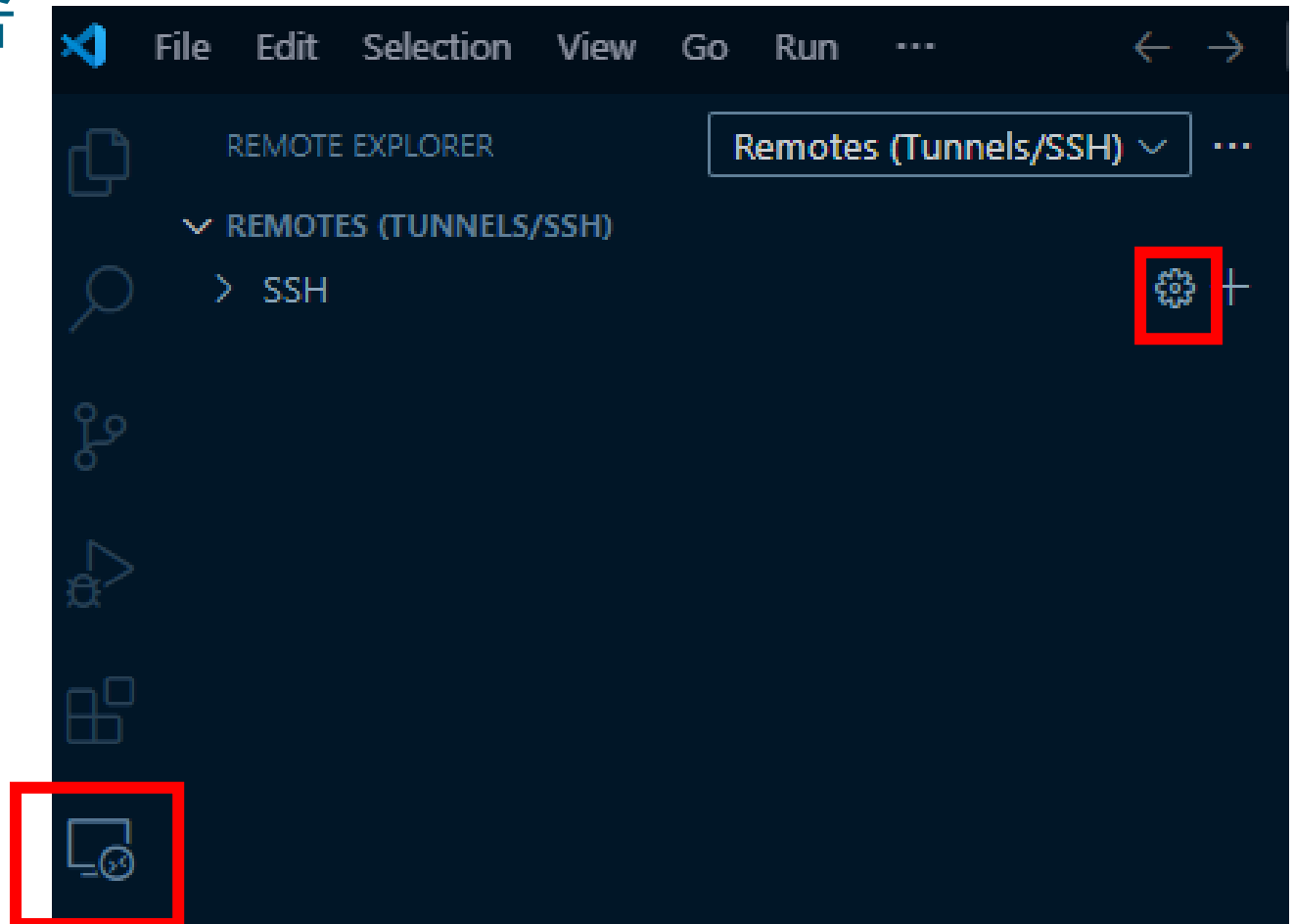
# JCloud 접속: VSCode

- VSCode 로 SSH 통해 원격 접속
  - Remove-SSH extension 이용
    - VSCode에서 Remote-SSH 설치



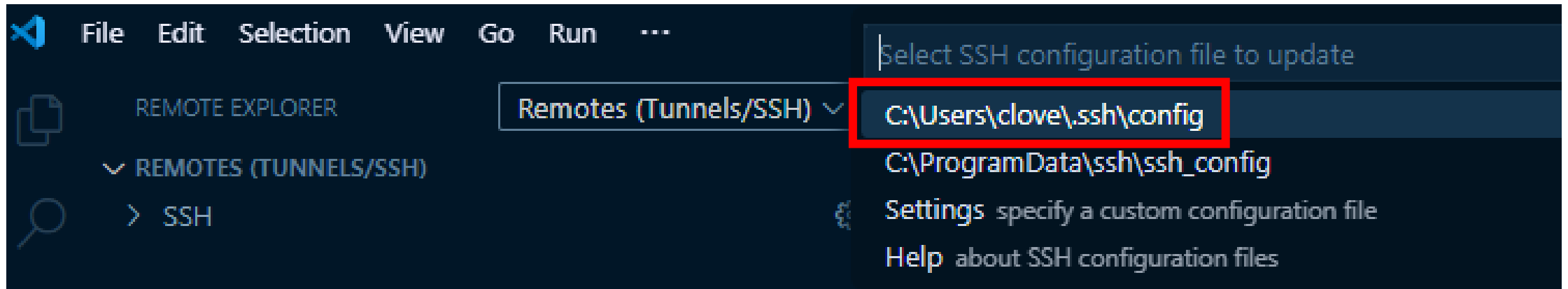
# JCloud 접속: VSCode

- VSCode 로 SSH 통해 원격 접속
  - 왼쪽 Remote Explorer
    - SSH 의 톱니바퀴(설정) 클릭



# JCloud 접속: VSCode

- VSCode 로 SSH 통해 원격 접속
  - 첫번째 설정파일 클릭 (config)
    - C:\Users\{사용자명}\.ssh\config



# JCloud 접속: VSCode

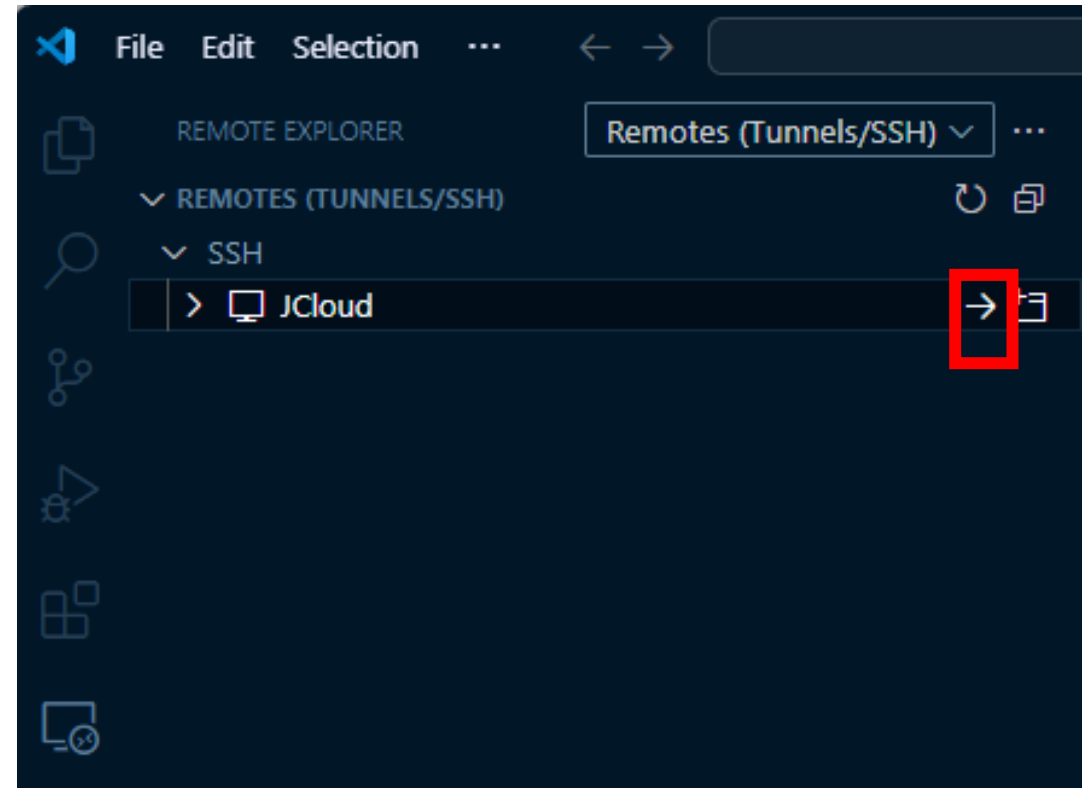
- VSCode 로 SSH 통해 원격 접속
  - Config 파일 수정

```
Host JCloud
  HostName 113.198.66.75
  User ubuntu
  IdentityFile C:\Users\clove\.ssh\hwpark-key.pem
  port 19231
```

- Host: ssh 접속 시 표시될 이름 (원하는 이름으로 정하면 됨)
- HostName: JCloud 접속 주소
- IdentityFile: JCloud 에 등록한 개인 pem 키 경로 (여기 작성한 경로에 pem키 복사필요)
- Port: 포트 주소(SSH 포트 주소)

# JCloud 접속: VSCode

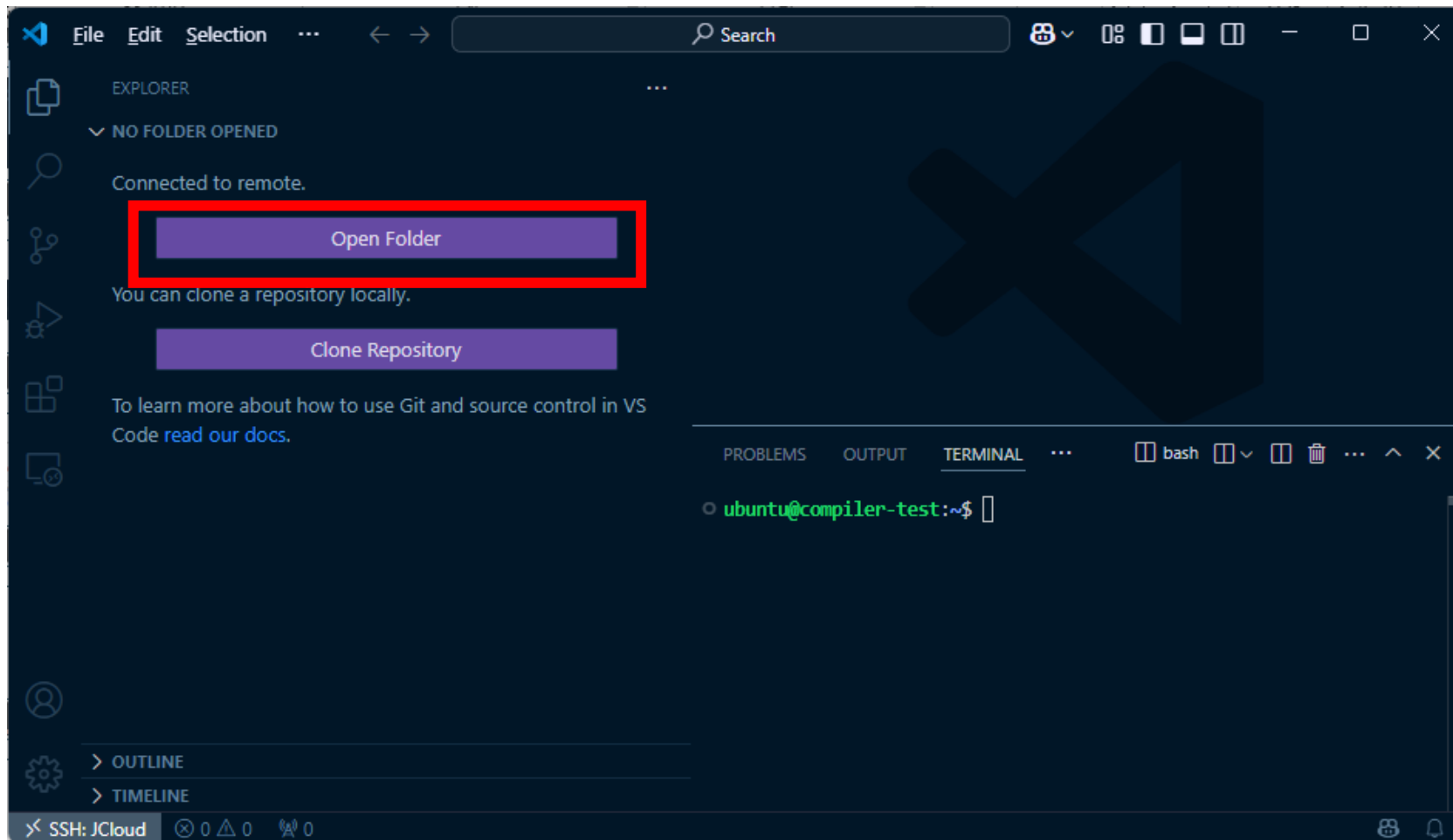
- VSCode 로 SSH 통해 원격 접속
  - JCloud 원격 접속
    - Config 파일에서 설정한 Host 표시
    - Host 옆의 화살표 클릭하여 접속





# JCloud 접속: VSCode

- VSCode 로 SSH 통해 원격 접속



# 개발 환경 요약

---

- JCloud (Ubuntu) 에서 lex 빌드 및 테스트
- 개발 방식
  - Xshell(SSH) 로 JCloud 접속하여 vi 로 개발, 빌드 및 테스트까지 모두 수행
  - 로컬 PC (Window, MAC) 에서 VSCode SSH 접속 통해 JCloud 연결하여 개발, 빌드 및 테스트까지 모두 수행

# Project1

---

[프로젝트 설명]

# Project1 – Lex Programming

- Overview

- Lex를 사용해서 간단한 c프로그램을 읽어 들인 후, 각 토큰들을 출력하는 프로그램을 작성
- 입력
  - 중첩된 주석과 .. 연산자를 포함하는 간단한 c프로그램
  - 중첩된 주석
    - 주석 (/ \* ... \*/) 안에 또다른 주석이 중첩 가능
    - 실제 c 언어에서는 중첩된 주석을 지원하지 않고 컴파일 에러 발생
  - .. 연산자
    - E.g., 1..5 는 1.(float) . 5(integer) 가 아닌 1(integer) .. (operator) 2(integer)로 해석되어야 함
    - 1..5 의 의미는 1~5 (range)를 나타냄
- 출력
  - 각 token에 대해서 token의 종류 및 lexeme

# Project1 – Hint

---

- 중첩된 주석

- Lex start condition 이용
- 처음에 normal 모드로 시작 (main 함수에서 BEGIN Normal; 수행)
- “/\*” 읽는 순간 comment mode로 시작 (BEGIN Comment;)
- Comment 모드에서 “/\*” 만나면 depth count 1증가
- Comment 모드에서 “\*/” 만나면 depth count 1감소, depth count 0되면 Normal 모드로 전환
- Comment 모드에서는 “/\*”, “\*/” 외의 모든 문자는 무시

- .. 연산자

- Lookahead operator (/) 이용
- Integer .. Integer 형태로 1 .. 5 는 1~5 범위를 나타냄
- “..”/{integer}

# Sample Input & Output

Input C Code	Output	
<pre>/* **** */ /* nested comments */ /* **** */ struct _point {     float x, y, z;     int color; } point[20]; struct _line {     struct _point *p[2];     int color;     float meter = 0.5; } line[20];  1..50</pre>	KEY	struct
	ID	_point
	OP	{
	ID	float
	ID	x
	OP	,
	ID	y
	OP	,
	ID	z
	OP	;
	...	
	INT	1
	OP	..
	INT	50

출력 시 각 항목 사이는 탭으로  
분리되어야 함  
printf("ID\t%s\n", ...)

# makefile

- lex 이용해 lexer 만들고 최종 실행 파일 생성

```
subc: lex.yy.o
    gcc -o subc lex.yy.o -ll
lex.yy.o: lex.yy.c
    gcc -c -g lex.yy.c
lex.yy.c: subc.l
    flex subc.l
```

- subc.l -> (flex) -> lex.yy.c -> (gcc 컴파일) -> lex.yy.o -> (gcc 링크) -> subc 실행파일
- E.g.,
  - \$ sudo apt install flex // flex 설치
  - \$ make // makefile 통해 빌드
  - \$ ./subc shortex.c // lexer 수행

# 프로젝트 점수 배점

---

- 총 배점: 100점
  - 기본 토큰 처리 (60점)
    - Operators, white-space, keyword 등 토큰 인식 및 출력
  - 중첩된 주석 처리 (20점)
  - .. 연산 처리 (20점)
- JCloud Ubuntu 24.04 환경에서 채점 예정
- 출력 시 각 항목 사이는 탭(\t)으로 분리되어야 함
  - 어길 시 감점
  - `printf("ID\t%s\t\n", ...)`



# 그 외 알아둘 사항

---

- Lex example 및 프로젝트 skeleton 코드 파일 제공
  - Discord 공지 채널 및 lms 수업 공지사항으로 등록
  - Skeleton 코드는 참고용으로, 본인이 설계한 것에 맞게 자유롭게 수정 가능
  - result\_example 폴더 내부 shortex.c 의 lex 결과와 shortex.txt 비교
    - shortex.c 정확히 처리하면 만점 (shortex.c 와 비슷한 코드로 채점 예정)
- Cheating 금지!!
  - Copy Killer 로 코드 검증
  - 코드 내 자세한 주석 작성 추천
  - Discord 로 활발히 질문/답변 추천

# 프로젝트 제출

---

- 제출 기한

- **4월 4일 금요일 23시 59분** (4/4까지)
- 하루 늦을 때마다 20점 감점 (4/8까지 제출 가능)

- 제출 파일

- Zip 으로 압축해서 제출
- **Zip 파일명에 학번 기입** e.g) Compiler\_Project1\_202512345.zip
- Zip 에는 **subc.l 등 소스 코드 파일**과 함께 빌드를 위한 **makefile** 이 있어야 함

- 제출 방법

- lms의 레포트 (Project1) 로 zip 파일 첨부하여 제출