

AWS EC2 에 VSCode Server 설치 방법

[1] AWS EC2용 키 페어 생성 : 미국(버지니아 북부) 리전을 사용한다

AWS Console에서 EC2 -> 키 페어로 가서 [키 페어 생성] 버튼을 클릭한다

키 페어 생성 화면에서

이름 : **vscode-ec2-key**

키 페어 유형 : RSA

프라이빗 키 파일 형식 : .pem

키 페어 생성 정보

키 페어
프라이빗 키와 퍼블릭 키로 구성되는 키 페어는 인스턴스에 연결할 때 자격 증명을 증명하는 데 사용하는 보안

이름
vscode-ec2-key
이름에는 최대 255개의 ASCII 문자가 포함됩니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

키 페어 유형 | 정보
☒ RSA ☐ ED25519

프라이빗 키 파일 형식
☒ .pem
OpenSSH와 함께 사용
☐ .ppk
PuTTY와 함께 사용

으로 설정하고 [키 페어 생성] 버튼을 클릭한다

이 때 자동으로 **vscode-ec2-key.pem** 파일이 자동으로 다운로드 되는데 EC2 접속시에 사용되므로 잘 보관해둔다(다운로드 폴더에 저장)

키 페어 생성 완료

키 페어 (1/1) 정보

키 페어를 속성 또는 태그로 찾기

<input checked="" type="checkbox"/>	이름	유형	생성 완료
<input checked="" type="checkbox"/>	vscode-ec2-key	rsa	2025/05/05 17:30 GMT+9

[2] AWS EC2 인스턴스 생성하기

AWS Console에서 EC2 -> 인스턴스로 가서 [인스턴스 시작] 버튼을 클릭한다

인스턴스 시작 입력 화면에서 아래와 같이 설정한다

(아래 없는 내용은 기본 설정 값으로 그대로 두고 진행한다)

이름: `vscode-server-instance`

Amazon Machine Image(AMI):

Ubuntu Server 22.04 LTS (HVM), (SSD) Volume Type

인스턴스 유형 : `t3.large`

키페어 이름 : `vscode-ec2-key` (앞에서 생성한 키페어를 선택해준다)

네트워크 설정

- 방화벽 : 보안그룹 생성

아래와 같이 세 개의 트래픽 허용 모두를 설정해준다

방화벽(보안 그룹) | 정보

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

☒ 보안 그룹 생성

☐ 기존 보안 그룹 선택

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

☒ 다음에서 SSH 트래픽 허용
인스턴스 연결에 도움

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

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

위치 무관
0.0.0.0/0

스토리지 구성:

20 GiB / `gp3` 로 루트 볼륨을 설정한다

모두 설정하였으면 [인스턴스 시작] 버튼을 클릭하면 인스턴스가 생성된다

[모든 인스턴스 보기]를 클릭하고 웹 브라우저 화면을 갱신하면 아래와 같이 보인다

인스턴스 (1) 정보

최종 업데이트 날짜
1 minute 전

연결

인스턴스 상태 ▼

작업 ▼

인스턴스 태그

Q 인스턴스를 속성 또는 (case-sensitive) 태그로 찾기

모든 상태 ▼

<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태 ▼	인스턴스 유형 ▼	상태 검사	경보 상태	가용 영역
<input type="checkbox"/>		i-0242732423964b124	실행 중	t3.large	초기화	경보 보기 +	us-east-1a

인스턴스 ID를 클릭하여 퍼블릭 IPv4 주소를 복사해둔다 (접속 시 사용)

퍼블릭 IPv4 주소

 3.86.45.193 | [개방 주소법](#) 

인스턴스 상태

 **실행 중**

예시) 3.86.45.193

[3] EC2 의 8080 포트 허용을 위한 보안그룹 규칙 추가하기

AWS Console -> EC2-> 인스턴스에서 인스턴스 ID를 클릭하고 보안 탭을 클릭하고

보안 그룹에 보이는 아래 링크를 클릭한다


세부 정보 | 상태 및 경보 | 모니터링 | **보안**

▼ 보안 세부 정보

IAM 역할

-

보안 그룹

 sg-0eb5dd2c4e2607ede (launch-wizard-1)

▼ 인바운드 규칙

[인바운드 규칙 편집] 버튼을 누르고 다시 규칙 추가를 눌러

유형 : 사용자 지정 TCP

포트번호 : 8080

소스 : Anywhere-IPv4 / 0.0.0.0/0

으로 설정하고 [규칙 저장] 버튼을 클릭한다

인바운드 규칙 편집 정보
인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

인바운드 규칙 정보	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	설명 - 선택 사항 정보	
보안 그룹 규칙 ID sgr-015a34680f5d1759a	SSH	TCP	22	사용자...	Q	삭제
sgr-09656dfa03afc322c	HTTPS	TCP	443	사용자...	Q 0.0.0.0/0 X	삭제
sgr-0ee64bf6d1369dbb0	HTTP	TCP	80	사용자...	Q 0.0.0.0/0 X	삭제
-	사용자 지정 TCP	TCP	8080	Anyw...	Q 0.0.0.0/0 X	삭제

[규칙 추가](#)

취소 [변경 사항 미리 보기](#) [규칙 저장](#)

결과 화면 : 아래와 같이 4개의 규칙이 설정되어 있다

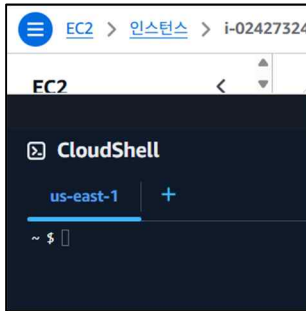
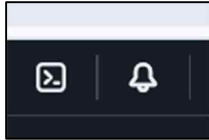
인바운드 규칙 (4) [태그 관리](#) [인바운드 규칙 편집](#)

Q 검색

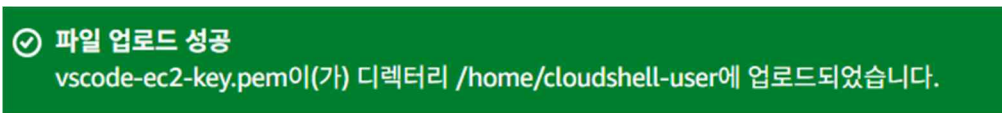
<input type="checkbox"/>	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위
<input type="checkbox"/>	-	sgr-0c5de4b5cc788ef94	IPv4	사용자 지정 TCP	TCP	8080
<input type="checkbox"/>	-	sgr-015a34680f5d1759a	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-09656dfa03afc322c	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-0ee64bf6d1369dbb0	IPv4	HTTP	TCP	80

[4] AWS CloudShell에서 EC2 인스턴스 접속하기

AWS Console에서 우측 상단의 CloudShell 아이콘을 클릭하여 CloudShell을 실행한다

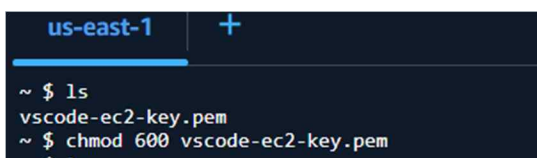


CloudShell 의 우측 상단의 작업 -> 파일 업로드를 클릭하고 다운 받아놓은 vscode-ec2-key.pem 파일을 선택하여 업로드한다



ls 명령을 파일을 확인하고 아래 명령으로 파일 권한을 수정한다

chmod 600 vscode-ec2-key.pem



앞에서 복사해 둔 EC2 인스턴스의 Public IP를 사용하여 아래와 같이 실행한다

ssh -i vscode-ec2-key.pem ubuntu@3.86.45.193

아래 화면에서 "yes"를 입력하고 Enter를 친다

```
~ $ ssh -i vscode-ec2-key.pem ubuntu@3.86.45.193
The authenticity of host '3.86.45.193 (3.86.45.193)' can't be established.
ED25519 key fingerprint is SHA256:92dUZ7Y66jPNMZpbCW9yZFYsd9SHDIKtHQ/s92t0fnw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

EC2 인스턴스 연결 성공 화면

```
us-east-1 +
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.86.45.193' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon May  5 08:58:12 UTC 2025

System load:  0.0          Processes:            103
Usage of /:   8.6% of 19.20GB Users logged in:      0
Memory usage: 3%          IPv4 address for ens5: 172.31.80.33
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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-80-33:~$
```

df -h 명령으로 디스크 용량을 확인해 본다 (20GB)

```
ubuntu@ip-172-31-80-33:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       20G   1.7G   18G   9% /
tmpfs           3.9G   0     3.9G   0% /dev/shm
tmpfs           1.6G  856K   1.6G   1% /run
tmpfs           5.0M   0     5.0M   0% /run/lock
efivarfs        128K   3.6K   120K   3% /sys/firmware/efi/efivars
/dev/nvme0n1p15 105M   6.1M   99M    6% /boot/efi
tmpfs           783M   4.0K   783M   1% /run/user/1000
ubuntu@ip-172-31-80-33:~$
```

[5] EC2에 VSCode Server 설치하기

CloudShell에서 EC2에 연결된 상태에서 다음 명령을 차례로 실행한다

[<https://seokbong.tistory.com/312> 참조]

1. 설치 파일 다운로드 명령

```
curl -fsSL https://code-server.dev/install.sh | sh
```

2. VSCode 서버 실행

```
sudo systemctl enable --now code-server@ubuntu
```

3. config 파일 수정

```
sudo nano ~/.config/code-server/config.yaml
```

아래와 같이 수정한다(빨간색 부분 수정)

```
bind-addr: 0.0.0.0:8080
auth: password
password: mypassword
cert: false
```

Nano 편집기로 수정한 후 **Ctrl-O**를 누르고 엔터를 치면 저장된다

다시 **Ctrl-X**를 누르면 편집기에서 빠져나온다

4. config 변경 값으로 서버 재실행


```
sudo systemctl restart code-server@ubuntu
```

[6] VSCode Server에 접속하기

새로운 웹 브라우저 창을 열고 EC2의 Public IP주소를 사용하여 아래와 같이 입력한다

3.86.45.193:8080

아래 화면에서 비밀번호(mypassword)를 입력하여 서버에 접속한다

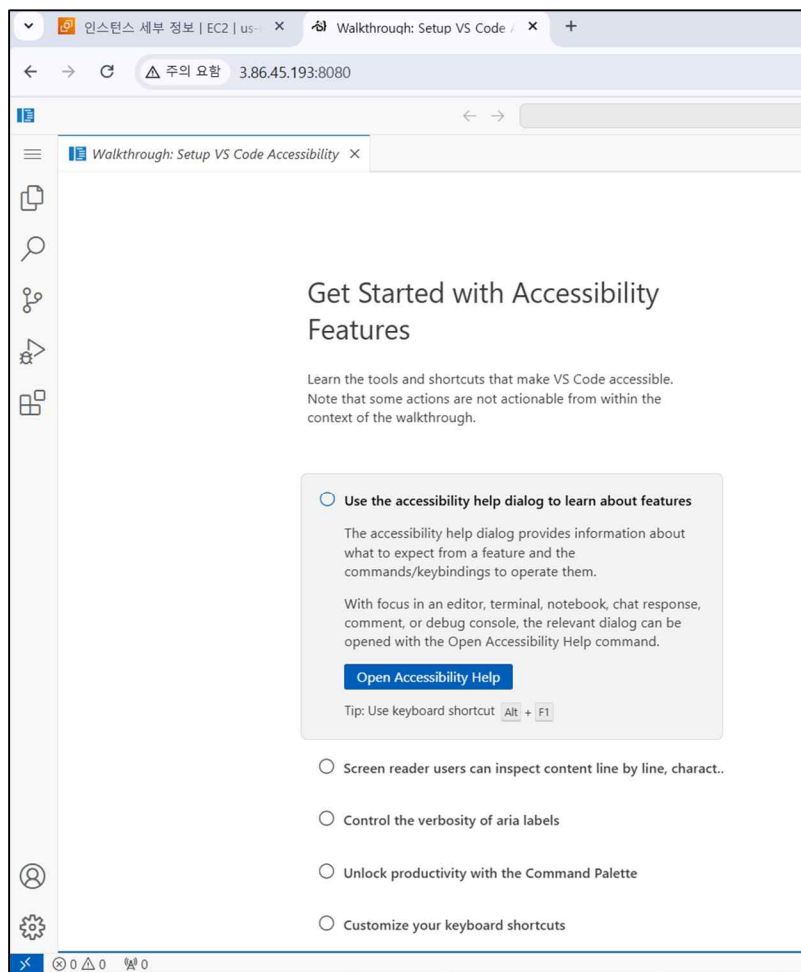


Welcome to code-server

Please log in below. Check the config file at `/home/ubuntu/.config/code-server/config.yaml` for the password.

..... SUBMIT

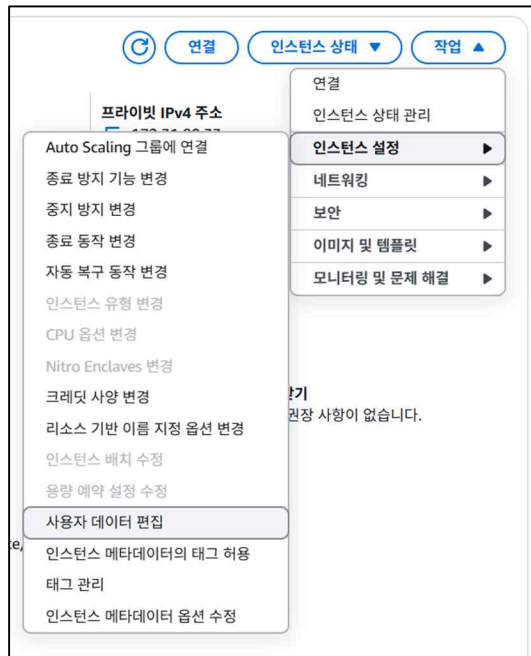
연결 성공 화면



[7] EC2 인스턴스 재 부팅 시 VSCode Server 자동 실행 설정하기

AWS EC2 -> 인스턴스 에서 생성한 인스턴스를 중지시킨다. 중지됨으로 바뀌면

인스턴스의 ID를 클릭하고 우측 상단의 [작업]에서 [인스턴스 설정] -> [사용자 데이터 편집]을 클릭한다



하단에 아래 내용을 붙여 넣고 [저장]버튼을 클릭한다

```
#!/bin/bash
```

```
systemctl enable code-server@ubuntu
```

```
systemctl restart code-server@ubuntu
```

새 사용자 데이터
이 사용자 데이터는 현재 사용자 데이터를 대체함

☒ 사용자 데이터를 텍스트로 수정
아래에 사용자 데이터 추가

```
#!/bin/bash

systemctl enable code-server@ubuntu
systemctl restart code-server@ubuntu
```

☐ 입력이 이미 base64로 인코딩되어 있음

인스턴스를 다시 시작하고 새로 변경당 된 퍼블릭 IPv4 주소를 복사한다음

VSCode Server에 접속한다 (변경된 IP 주소 사용)

54.174.119.207:8080

Welcome to code-server

Please log in below. Check the config file at `/home/ubuntu/.config/code-server/config.yaml` for the password.

연결 성공 화면

Get Started with VS Code for the Web

Customize your editor, learn the basics, and start coding

☒ **Choose your theme**

The right theme helps you focus on your code, is easy on your eyes, and is simply more fun to use.

Tip: Use keyboard shortcut `Ctrl + K` `Ctrl + T`

☐ Just the right amount of UI

☐ Rich support for all your languages

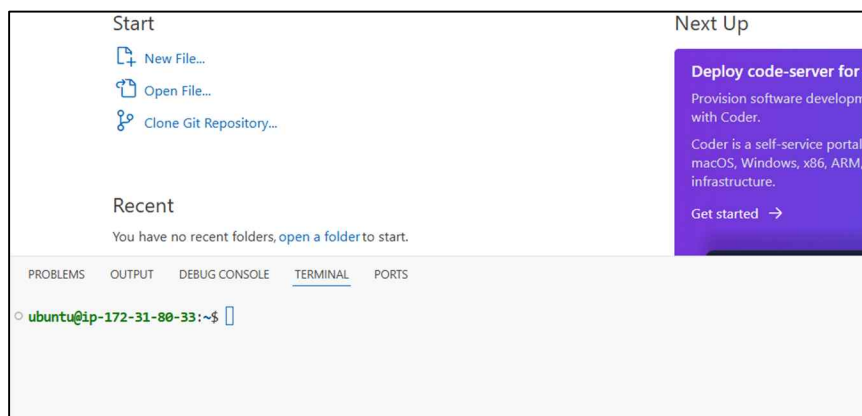
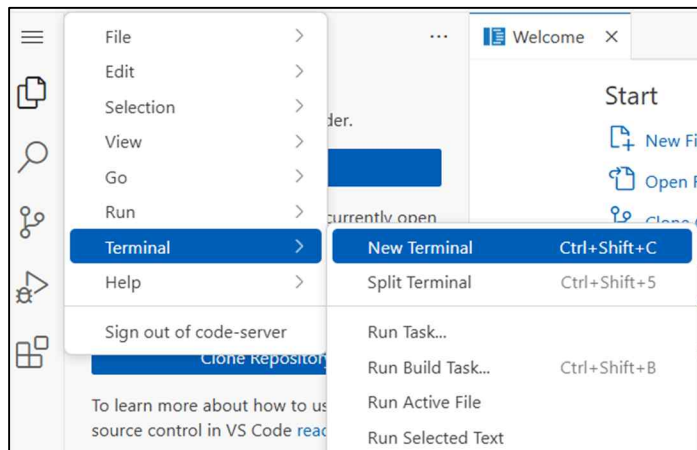
☐ Unlock productivity with the Command Palette

☐ Open up your code

✓ [Mark Done](#) [Next Section](#) →

[7] VSCode Server에 개발 환경 패키지 설치하기

VSCode Server에 접속(혹은 CloudShell 로 직접 EC2 인스턴스에 접속)하여 좌측 첫번째 아이콘을 누르고 아래와 같이 선택해서 새 터미널을 연다



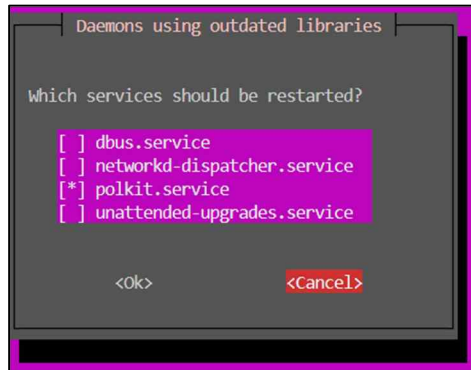
1. 아래 명령을 실행하여 Ubuntu 기본 개발환경 패키지를 설치한다

(VSCode Teminal에서는 마우스 우클릭이 안되고 대신 Ctrl-V로 붙여넣기가 가능하다)

```
sudo apt update
sudo apt install -y \
    python3 python3-pip python3-venv \
    git curl wget unzip \
```

```
build-essential ₩
libssl-dev libffi-dev
```

아래 화면이 나오면 Tab키를 두 번 연속 사용하여 <Cancel>을 선택하고 엔터를 친다



대부분 패키지가 이미 설치되어 있다

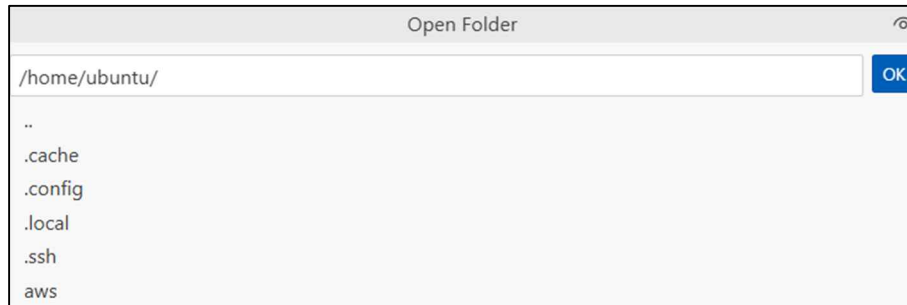
2. AWS CLI 설치

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
aws --version
```

3. AWS IoT Device SDK 설치

```
pip3 install awsiotsdk AWSIoTPythonSDK
sudo apt install -y mosquitto mosquitto-clients
```

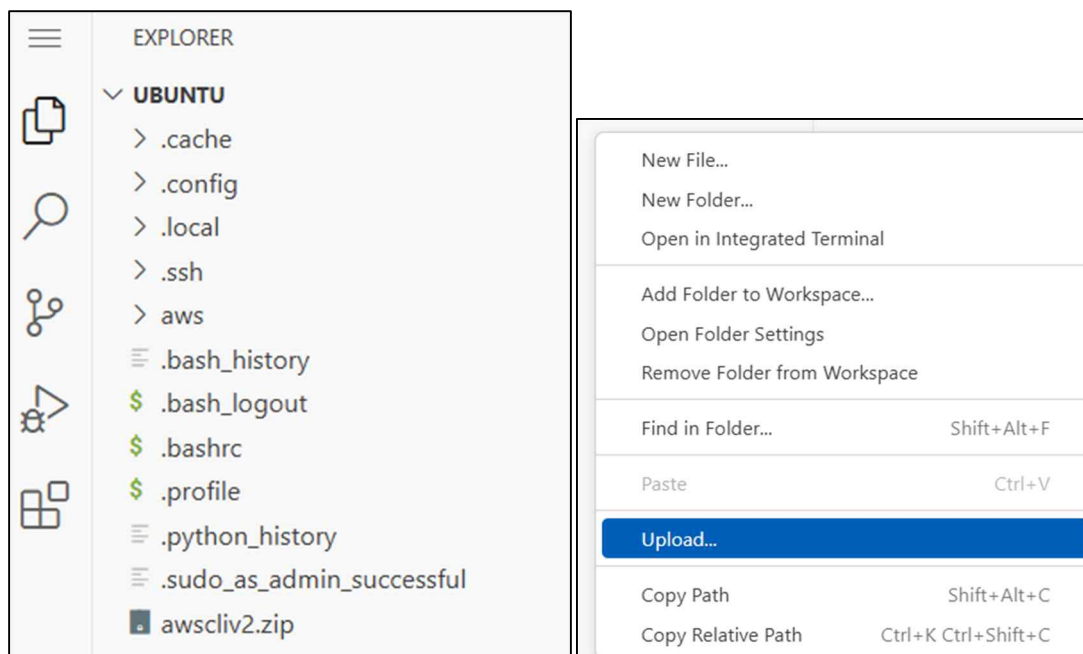
좌측 두번째 Explorer 아이콘을 클릭하고 [Open Folder] 버튼을 클릭하고
/home/ubuntu/ 폴더가 선택되어 있는 상태에서 [OK] 버튼을 클릭한다



이후부터 아래 IoT Core 실습이 가능하다

<https://catalog.us-east-1.prod.workshops.aws/workshops/f87a7c7a-0af8-416a-80ee-7c25c5789307/ko-KR>

인증서 파일 업로드시 EXPLOER 창에서 우측 마우스를 클릭하면 [Upload] 메뉴를 선택할 수 있다



다양한 Extension package를 추가로 설치한다(Python, Korean Language Pack 등)

아래 사이트 참고

<https://wikidocs.net/137959>

[8] EC2 에 IAM Role 부여하기 : AWS IoT Core와 연동할 수 있다

1단계. IAM Role 생성 (EC2용)

1. AWS 콘솔 → IAM 서비스로 이동
2. 왼쪽 메뉴 → 역할 -> 역할 생성
3. Use case: EC2 선택 → 다음 버튼 클릭

신뢰할 수 있는 엔터티 선택 정보

신뢰할 수 있는 엔터티 유형

☒ **AWS 서비스**
EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

☐ **AWS 계정**
사용자 또는 서드 파티에 속한 다른 AWS 계정의 엔터티가 이 계정에서 작업을 수행하도록 허용합니다.

☐ **웹 자격 증명**
지정된 외부 웹 ID 제공업체와 연동된 사용자가 이 역할을 맡아 이 계정에서 작업을 수행하도록 허용합니다.

☐ **SAML 2.0 페더레이션**
기업 디렉터리에서 SAML 2.0과 연동된 사용자가 이 계정에서 작업을 수행할 수 있도록 허용합니다.

☐ **사용자 지정 신뢰 정책**
다른 사용자가 이 계정에서 작업을 수행할 수 있도록 사용자 지정 신뢰 정책을 생성합니다.

사용 사례
EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

서비스 또는 사용 사례

EC2

지정된 서비스에 대한 사용 사례를 선택합니다.

사용 사례

☒ **EC2**
Allows EC2 instances to call AWS services on your behalf.

권한 부여

4. 검색창에 아래와 같이 "iotfull" 검색 하여 선택 : AWSIoTFullAccess

권한 추가 정보

권한 정책 (1/1045) 정보
새 역할에 연결할 정책을 하나 이상 선택합니다.

Q iotfull X

모든 유형

☒ 정책 이름  ☐ 유형

☒  [AWSIoTFullAccess](#) AWS 관리형

선택하고 Next 버튼을 클릭한다

역할 이름 지정

5. 역할 이름: EC2IoTRole

→ 역할 생성 버튼을 클릭한다

2단계. EC2에 Role 부여

1. EC2 콘솔 → 해당 EC2 인스턴스 선택
 2. 위 메뉴 → 작업 -> 보안 -> IAM 역할 수정 클릭
 3. 위에서 만든 EC2IoTRole 선택 → [IAM 역할 업데이트] 버튼 클릭
-

이제 VSCode Server 터미널에서 인증 없이 IoT 관련 명령 바로 동작함:

aws iot create-thing --thing-name MyThing2

aws iot describe-endpoint

- MQTT 테스트 예제 : mqtt-pub-test.py

```
# mqtt-pub-test.py

'''
# MQTT 환경 설정
sudo apt update
sudo apt install mosquitto mosquitto-clients -y
sudo systemctl enable mosquitto
sudo systemctl start mosquitto
pip install paho-mqtt
'''

import time
import json
import paho.mqtt.client as mqtt

broker = "localhost"
port = 1883
topic = "test/topic"

client = mqtt.Client()
client.connect(broker, port)

while True:
    payload = {
        "timestamp": int(time.time()),
        "value": round(25 + time.time() % 5, 2)
    }
    client.publish(topic, json.dumps(payload))
    print("Published:", payload)
    time.sleep(2)
```


- MQTT 테스트 예제 : mqtt-sub-test.py

```
# mqtt-sub-test.py

import json
import paho.mqtt.client as mqtt

broker = "localhost"    # 외부 broker IP
port = 1883
topic = "test/topic"

def on_connect(client, userdata, flags, rc):
    print("Connected with result code", rc)
    client.subscribe(topic)

def on_message(client, userdata, msg):
    print("Received:", json.loads(msg.payload.decode()))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect(broker, port)
client.loop_forever()
```